Statistik aktiv mit R — Textdatei mit Überschriften und Code-Chunks

Hans Peter Wolf, Peter Naeve, Veith Tiemann

April 2006

Inhalt

Vorwort

1 ⟨*Verwendung von R* 1⟩≡
⟨*Sprachelement* 2⟩
⟨*Offenlegung* 3⟩
⟨*Experiment* 4⟩

2 ⟨*Sprachelement* 2⟩≡
```
x <- 1:100; mean(x)
```

3 ⟨*Offenlegung* 3⟩≡
```
length(alter)
```

4 ⟨*Experiment* 4⟩≡
```
exp.mere()
```

Kapitel 1

Datenanalyse? Daten? Statistik?

5 ⟨*5⟩≡
```
sample(6,10,replace=T)
```

6 ⟨*5⟩+≡
```
wuerfel.exp()
```

1.1 Was für Daten gibt es?

1.2 Wo kommen Daten her?

Kapitel 2

Univariate, exploratorische Analyse

7 ⟨*5⟩+≡
```
alter
```

8 ⟨*5⟩+≡
```
length(alter)
```

2.1 Häufigkeitstabellen und deren Darstellung

9 ⟨*5⟩+≡
```
haeufigkeit.diskret(alter)
table(alter)
```

10 ⟨*5⟩+≡
```
plot(table(alter),ylab="abs. H\"aufigkeit")
```

11 ⟨*5⟩+≡
```
pie(table(alter))
```

12 ⟨*5⟩+≡
```
haeuf.stet(alter,anzahl.klassen=6)
```

13 ⟨*5⟩+≡
```
hist(alter,nclass=6,
     prob=FALSE)
```

2.2 Auswertung der Urliste: Lage und Variabilität

14 ⟨* 5⟩+≡
```
x<-sample(buecher.stud,size=20)
halbe.halbe(x)
```

15 ⟨* 5⟩+≡
```
halbe.halbe(sort(x))
```

2.2.1 Zur Lage eines Datensatzes

16 ⟨* 5⟩+≡
```
plot(x,seq(x),xlab="Anzahl Buecher",ylab="i")
abline(v=60)
```

17 ⟨* 5⟩+≡
```
mean(x)
```

18 ⟨* 5⟩+≡
```
mean(x[x<mean(x)])
```

19 ⟨* 5⟩+≡
```
mean(x[x>mean(x)])
```

20 ⟨* 5⟩+≡
```
plot.ohne(x,ohne=2,xlab="Anzahl Buecher",ylab="i")
```

21 ⟨* 5⟩+≡
```
median(x)
```

22 ⟨* 5⟩+≡
```
mean(x,trim=0.05)
```

23 ⟨* 5⟩+≡
```
plot.trim(x)
```

24 ⟨* 5⟩+≡
```
c(mittelwert=mean(x),median=median(x),
  getrimmt_0.1=mean(x,trim=.1),
  modus=modus(x),modus.diskr=modus(x,stetig=F))
```

25 ⟨* 5⟩+≡
```
summary(x)
```

26 ⟨* 5⟩+≡
```
plot(x,1:length(x),xlab="Buecher",ylab="i")
abline(v=summary(x)[-4],lty=2)
```

27 ⟨* 5⟩+≡
```
boxplot(x,range=0,horizontal=T,xlab="Anzahl Buecher")
```

28 ⟨* 5⟩+≡
```
boxplot(x,buecher.stud,range=0,
  names=c("x","buecher.stud"),ylab="Anzahl Buecher")
```

29 ⟨* 5⟩+≡
```
summary(buecher.stud)
```

30 ⟨* 5⟩+≡
```
xx<-wiederholte.stichproben(x=buecher.stud,n=20,wdh=30)
boxplot(xx, range=0, ylab="Anzahl Buecher")
```

2.2.2 Zur Variabilität eines Datensatzes

31 ⟨* 5⟩+≡
```
max(x)-min(x);max(buecher.stud)-min(buecher.stud)
```

32 ⟨* 5⟩+≡
```
IQR(x);IQR(buecher.stud)
```

33 ⟨* 5⟩+≡
```
par(mfrow=c(1,2))
range.plot(x,marker=c(0.25,0.50,0.75,0.95))
range.plot(buecher.stud,marker=c(0.25,0.50,0.75,0.95))
```

34 ⟨* 5⟩+≡
```
sd(buecher.stud)
```

35 ⟨* 5⟩+≡
```
sd(buecher.stud)/mean(buecher.stud)
```

36 ⟨* 5⟩+≡
```
vk.plot(buecher.stud); vk.plot(x,add=T)
```

37 ⟨* 5⟩+≡
```
vk.plot(gewicht.stud); vk.plot(groesse.stud,add=T)
```

38 ⟨* 5⟩+≡
```
mad(buecher.stud)
```

2.3 Die empirische Verteilungsfunktion

39 ⟨* 5⟩+≡
```
plot(sort(x),seq(x),ylab="(i)",
  xlab="Anzahl Buecher")
```

40 ⟨* 5⟩+≡
```
emp.cdf(x,stetig=F)
```

2.4 Besondere Strukturen einer Verteilung

41 ⟨* 5⟩+≡
```
dichte.plot(gewicht.stud,
   fenster=0.125*iqd(gewicht.stud),AXES=FALSE)
```

42 ⟨* 5⟩+≡
```
dichte.manip(gewicht.stud)
```

Symmetrie, Schiefe und Wölbung.

43 ⟨* 5⟩+≡
```
c(modus=modus(gewicht.stud),
      median=median(gewicht.stud),
      mean=mean(gewicht.stud))
```

44 ⟨* 5⟩+≡
```
cbind(modus=modus(groesse.stud),
      median=median(groesse.stud),
      mean=mean(groesse.stud))
```

45 ⟨* 5⟩+≡
```
c(modus=modus(x),median=median(x),mean=mean(x))
```

46 ⟨* 5⟩+≡
```
c(modus=modus(buecher.stud),
      median=median(buecher.stud),
      mean=mean(buecher.stud))
```

47 ⟨* 5⟩+≡
```
c(x=schiefe(x),
  buecher=schiefe(buecher.stud),
  groesse=schiefe(groesse.stud),
  gewicht=schiefe(gewicht.stud))
```

48 ⟨* 5⟩+≡
```
c(x=kurtosis(x)-3,
  buecher=kurtosis(buecher.stud)-3,
  gr=kurtosis(groesse.stud)-3,
  gew=kurtosis(gewicht.stud)-3)
```

49 ⟨* 5⟩+≡
```
box.cox.plot(buecher.stud,interaktiv=T)
```

2.5 Konzentrationsmessung – LORENZ und GINI

50 ⟨* 5⟩+≡
```
eda(klicks.moebel)
```

51 ⟨* 5⟩+≡
```
n<-10; quantile(klicks.moebel,(1:n)/n)
```

52 ⟨* 5⟩+≡
```
lorenz(klicks.moebel)
```

53 ⟨* 5⟩+≡
```
gini(klicks.moebel)
```

2.6 Fallstudie – das 6 aus 49 Lotto

54 ⟨* 5⟩+≡

```
lotto.experiment()
```

Kapitel 3

Bivariate, exploratorische Analyse

Was sind bivariate Daten?

3.1 Korrelation von Merkmalen

55 ⟨* 5⟩+≡
```
pakete
```

56 ⟨* 5⟩+≡
```
rowSums(pakete)
```

57 ⟨* 5⟩+≡
```
colSums(pakete)
```

58 ⟨* 5⟩+≡
```
image.plot(pakete,xlab="Rubrikenklasse",ylab="Paket")
```

59 ⟨* 5⟩+≡
```
zeilenprofil(pakete)
```

60 ⟨* 5⟩+≡
```
zeilenprofil.diff(pakete)
```

61 ⟨* 5⟩+≡
```
erw.unabh(pakete)
```

62 ⟨* 5⟩+≡
```
plot(umsatz,ylab="Umsatz",
     xlab="Mitarbeiterzahl")
```

63 ⟨* 5⟩+≡
```
plot(umsatz,xlab="Mitarbeiterzahl",ylab="Umsatz")
abline(v=mean(umsatz[,1]),h=mean(umsatz[,2]))
```

64 ⟨* 5⟩+≡
```
cor(umsatz[,1],umsatz[,2])
```

65 ⟨* 5⟩+≡
```
cor(umsatz[,1],umsatz[,2],method="spearman")
```

66 ⟨* 5⟩+≡
```
korr.schieber(n=100,korr=0)
```

3.2 Der Vergleich zweier Merkmale

67 ⟨* 5⟩+≡
```
kleiner<-umsatz[umsatz[,1]<50,2]
groesser<-umsatz[umsatz[,1]>=50,2]
print(length(kleiner));print(length(groesser))
```

68 ⟨* 5⟩+≡
```
c(round(mean(kleiner)),round(mean(groesser)))
```

69 ⟨* 5⟩+≡
```
vgl.plots(kleiner,groesser)
```

70 ⟨* 5⟩+≡
```
chal
```

Kapitel 4

Auf zur Modellierung

4.1 Konzepte am Beispiel der Binomialverteilung

4.1.1 Bernoulli-Experimente und Zufallsvariablen

Eine Apothekenuntersuchung.

Zufallsexperimente und Zufallsvariablen.

Anwendung.

Bernoulli-Prozess.

Ziehungsprinzipien.

Simulation einer Apothekenbefragung.

71 ⟨* 5⟩+≡
```
x<-sample(0:1,size=50,replace=T,prob=c(21/50,29/50))
```

72 ⟨* 5⟩+≡
```
sum(x)
```

73 ⟨* 5⟩+≡
```
erfolge.bei.bernoulli.experimenten()
```

4.1.2 Wahrscheinlichkeitsfunktion

Wahrscheinlichkeitsbaum.

Erfolgsanzahl von Bernoulli-Experimenten.

74 ⟨* 5⟩+≡
```
choose(5,2)
```

Berechnung des Binomialkoeffizienten.

4.1.3 Binomialverteilung

75 ⟨* 5⟩+≡
```
n<-50; s<-0:n; p<-29/50
f.s<-dbinom(x=s,n,p=p)
plot(s,f.s,type="h")
```

76 ⟨* 5⟩+≡
```
dbinom(x=29,50,p=29/50)
```

4.1.4 Verteilungsfunktion

77 ⟨* 5⟩+≡
```
pbinom(29,size=50,p=0.70)
```

78 ⟨* 5⟩+≡
```
n<-50; s<-0:n
F.s<-pbinom(0:n,n,p=0.7)
plot(s,F.s,type="s")
```

Beispiel.

79 ⟨* 5⟩+≡
```
print(pbinom(31,50,29/50)-pbinom(26,50,29/50))
print(sum(dbinom(27:31,size=50,p=29/50)))
```

Quantile.

80 ⟨* 5⟩+≡
```
qbinom(0.5258371,50,29/50)
```

Realisationen und Modell.

81 ⟨* 5⟩+≡
```
binomial.experiment()
```

4.1.5 Erwartungswerte

Beispiel Kreditrückzahlungserwartungen.

Intuitive Erwartungen.

Der Erwartungswert.

Beispiel 2-Punkt-Verteilung.

Skalierung.

Beispiel.

4.1.6 Erwartungswert der Binomialverteilung

Anzahl problemloser Kreditverträge.

Anwendung.

82 ⟨* 5⟩+≡

```
n<-10; p<-0.75
f.x<-dbinom(0:n,n,p)
plot(0:n,f.x,type="h")
abline(v=n*p, lty=2)
```

4.1.7 Additivitätseigenschaft des Erwartungswertes

Rückzahlungserwartungen bei Kreditverträgen.

Additivitätssatz.

Verteilung der Summe binomialverteilter Zufallsvariablen.

4.1.8 Binomialverteilung und Variabilität

Variabilitäten von Erfolgsanzahlen.

Anwendung.

Varianz der Rückzahlungen unterschiedlicher Verträge.

Mindestwahrscheinlichkeiten zentraler Schwankungsintervalle.

4.1.9 Verteilung von Mittelwerten

Rückzahlungswahrscheinlichkeit von Krediten.

Anwendung.

Anwendung.

Schwankungsintervall und n.

83 ⟨* 5⟩+≡
```
p.est()
```

4.2 Verschiedene diskrete Verteilungen

4.2.1 Die hypergeometrische Verteilung

Beispiel Schlemmermenü.

Weitere Beispiele.

Berechnungen.

84 ⟨* 5⟩+≡
```
m<-6;n<-19;k<-5;x<-0:k
f.x<-dhyper(x,m,n,k)
plot(x,f.x,type="h")
```

Beispiel Lotto.

85 ⟨* 5⟩+≡
```
f.x<-dhyper(x=6,m=6,n=43,k=6)
1/f.x
```

Kontingenztabellen.

Beispiel Druckerpatronen.

86 ⟨* 5⟩+≡
```
KT.hyper(initial.m.n.k.n11=c(9,10,7,4))
```

Beispiel Verbrechensstatistiken.

87 ⟨* 5⟩+≡
```
phyper(627,m=881,n=5109,k=5637)
```

Eigenschaften der hypergeometrischen Verteilung.

Verteilungsvergleich.

88 ⟨* 5⟩+≡
```
hyper.to.binom()
```

4.2.2 Von der Binomial- zur Poisson-Verteilung

Auslastungsfragen.

Ein Modellierungsschritt.

89 ⟨* 5⟩+≡
```
binom.to.poisson()
```

Anwendung.

90 ⟨* 5⟩+≡
```
lambda<-5; x<-0:max(10,2*lambda)
f.x<-dpois(x,lambda)
plot(x,f.x,type="h")
```

Feuerwehreinsätze.

91 ⟨* 5⟩+≡
```
lambda<-42; k<-2; step<-k*sqrt(lambda)
lim<-floor(c(lambda-step,lambda+step))
ws<-ppois(lim[2],lambda)-ppois(lim[1]-1,lambda)
cat("lambda:",lambda,"k:",k,"Grenzen:",lim,"WS:",ws)
```

4.3 Stetige Modellwelt

4.3.1 Stetige Gleichverteilung

Beispiel Parkplatzsuche.

Erwartungswert und Varianz.

4.3.2 Über Summen zur Normalverteilung

Umsatzplanung.

Vermutung.

Summen gleichverteilter Zufallsvariablen.

92 ⟨* 5⟩+≡
```
sum.zv()
```

Eigenschaften der Normalverteilung.

Standardisierung.

93 ⟨* 5⟩+≡
```
pnorm(500,504,2)
```

94 ⟨* 5⟩+≡
```
mu<-504; sigma<-2; k<-1
cbind(k=k,"Untergrenze"=mu-k*sigma,
      "Obergrenze"=mu+k*sigma,
      "P(-k sigma<X<k sigma)"=pnorm(k)-pnorm(-k))
```

95 ⟨* 5⟩+≡
```
mu<-10; sigma<-2
x<-seq(mu-4*sigma,mu+4*sigma,length=100)
f.x<-dnorm(x,mu,sigma)
plot(x,f.x,type="l")
```

Ein zentrale Grenzwertsatz.

Beispiel Gaststättenumsatz.

96 ⟨* 5⟩+≡
```
mu.x<-30;  var.x<-81; n<-100;  k.fach<-2
mu<-mu.x*n; sd<-(var.x*n)^0.5
cat("P(Summe im zentralen Schwankungsintervall)=")
cat(1-2*pnorm(mu-k.fach*sd,mu,sd),"\n")
```

Beispiel Wahlverhalten.

Approximation der Binomialverteilung.

Approximation von Poisson-Verteilung.

QQ-Plots.

Verteilungsvergleich.

97 ⟨* 5⟩+≡
```
binom.norm.pois()
```

4.3.3 Wartezeitverteilungen

Beispiel Taxifahrer.

98 ⟨* 5⟩+≡
```
p<-0.3; x<-0:20; f.x<-dgeom(x,p)
plot(x,f.x,type="h")
```

Bespiel Überraschungseier.

4.3.4 Von geometrisch zu exponential

99 ⟨* 5⟩+≡
```
geo.to.exp()
```

Eigenschaften der Exponentialverteilung.

100 ⟨* 5⟩+≡
```
x<-(0:100)/5; lambda<-0.5
f.x<-dexp(x,lambda)
plot(x,f.x,type="l")
```

4.3.5 Von Poisson zu Exponential

Beispiel Feuerwehr.

Beispiel Bergwerksunfälle.

101 ⟨* 5⟩+≡
```
w<-diff(coal[,1])
x<-seq(0,7,length=50)
y<-dexp(x,191/111)
hist(w,prob=T,nclass=30)
lines(x,y)
```

4.3.6 Summe exponentialverteilter Zufallsvariablen

Karussell.

102 ⟨* 5⟩+≡
```
n<-3; lambda<-2.5
curve(dgamma(x,n,lambda),0,3*n/lambda)
```

4.3.7 GLIVENKO **und** CANTELLI

Zusammenfassung:

103 ⟨* 5⟩+≡
```
n<-10; wd<-100; mittel<-rep(0,wd)
for(i in 1:wd) mittel[i]<-mean(rexp(n))
mittel
```

104 ⟨* 5⟩+≡
```
mittel.std<-(mittel-mean(mittel))/sd(mittel)
```

Kapitel 5

Casino-Statistik

5.1 Würfelfragen

Eine Würfelfrage von CHEVALIER DE MÉRÉ**.**

105 ⟨* 5⟩+≡
```
exp.mere()
```

5.2 Wahrscheinlichkeit — was ist das?

Wahrscheinlichkeit einer vier beim Spiel: ***Mensch ärgere dich nicht*****.**

Zufallsexperiment und Wahrscheinlichkeiten.

Wahrscheinlichkeiten beim Würfelwurf.

Festlegung von Wahrscheinlichkeiten.

106 ⟨* 5⟩+≡
```
set.seed(9)
table(sample(1:6,6000,replace=T))
```

5.3 Rechnen mit Wahrscheinlichkeiten

Wahrscheinlichkeiten für ***keine sechs*****.**

Gegenwahrscheinlichkeit.

Verknüpfung von Aussagen.

Das sichere bzw. das unmögliche Ereignis.

Wahrscheinlichkeit für zwei Sechsen beim Würfeln.

5.4 Axiome der Wahrscheinlichkeitsrechnung

5.5 Zusammengesetzte Ereignisse

Mindestens eine Sechs bei vier Würfen.

Mindestens eine Sechs bei zwei Würfen.

Vereinigung von Ereignissen.

Mindestens eine Sechs bei zwei Würfen.

Mindestens eine Sechs bei drei Würfen.

Mindestens eine Sechs bei vier Würfen.

107 ⟨* 5⟩+≡

```
p<-1/6; w<-0:20; p.sechs<-1-(1-p)^w
plot(w, p.sechs, type="h", lty=2);  abline(h=0.5)
points(4, 1-(1-p)^4, type="h")
```

Anwendungen.

5.6 Kombinatorik für das Gleichmöglichkeitsmodell

5.7 Wahrscheinlichkeiten und Bedingungen

Auswirkungen von Bedingungen.

Mord im Orientexpress.

Raucher und Geschlecht.

5.8 Abhängigkeit und Unabhängigkeit

Kriminalpolizei.

Glücksspieler.

Multiplikationssatz. Wie wahrscheinlich ist *viermal keine sechs*?

Rauchen und Geschlecht.

5.9 Totale Wahrscheinlichkeit

Impfungen. Lohnt sich eine Impfung?

Allgemeine Ansteckungswahrscheinlichkeit.

5.10 Lernen aus Zusatzinformationen

Vom Symptom zur Ursache.

Ein fiktives Fahndungsbeispiel.

Satz von BAYES**.**

Philosophische Nachbemerkung.

5.11 Zusammengesetzte Zufallsexperimente

Die zweite MÉRÉSCHE **Frage — Pasch sechs bei 24 Würfen.**

Unabhängige Zufallsvorgänge.

Zur zweiten Würfelfrage.

108 ⟨* 5⟩+≡

```
n<-1:40;p<-35/36;y<-1-p^n
plot(n,y,type="h", lty=2)
abline(h=0.5)
points(24,y[24],type="h")
```

Anwendungssituationen.

Kapitel 6

Parameterschätzungen

Beispiel Unfälle.

6.1 Datengrundlage

Stichprobenumfang.

Zufallsstichprobe.

Messfehler.

6.2 Zur Identifikation des Modelltyps

109 ⟨* 5⟩+≡
```
boxplot(zwischen.unfalls.zeiten.02,horizontal=T)
```

110 ⟨* 5⟩+≡
```
plot(unfaelle.pro.woche.02.table)
```

Leitfaden zur Modelltypwahl.

Erkennungsplot für die Poisson-Verteilung.

111 ⟨* 5⟩+≡
```
poisson.erkennungsplot(unfaelle.pro.woche.02)
```

112 ⟨* 5⟩+≡
```
set.seed(7); poisson.erkennungsplot(rpois(50,0.8))
```

Erkennungsplot für die Exponentialverteilung.

113 ⟨* 5⟩+≡
```
exp.erkennungsplot(zwischen.unfalls.zeiten.02)
```

114 ⟨* 5⟩+≡
```
set.seed(7)
exp.erkennungsplot(rexp(50,.1))
```

QQ-Plots.

QQ-Plots zur Normalverteilung.

115 ⟨* 5⟩+≡
```
stpr<-rnorm(n=100,mean=180,sd=10)
qqnorm(stpr); qqline(stpr)
```

Fazit.

6.3 Stichproben- und Schätzfunktionen

6.3.1 Eigenschaften von Schätzfunktionen

Beispiel.

116 ⟨* 5⟩+≡
```
exp.est.fns()
```

6.3.2 Die Stichprobenfunktionen $\overline{X}$ und S^2

Momente von $\overline{X}$.

Beispiel Poisson-Verteilung.

Momente von S^2.

Normalverteilungsfall.

117 ⟨* 5⟩+≡
```
curve(dt(x,3),-5,5)
```

118 ⟨* 5⟩+≡
```
curve(dchisq(x,3),0.5,20)
```

Approximative Verteilung von $\overline{X}$.

CHEBYSHEV. Mit dem Namen CHEBYSHEV

6.3.3 Experimente zur Untersuchung von Stichprobenfunktionen

Ein Experiment zur Normalapproximation von $\overline{X}$.

119 ⟨* 5⟩+≡
```
n<-10; wd<-1000; mittel<-numeric(wd)
for(i in 1:wd) mittel[i]<-mean(rexp(n))
qqnorm(mittel)
```

120 ⟨* 5⟩+≡
```
exp.exp.mittel()
```

Die Verteilung von Stichprobenfunktionen

121 ⟨* 5⟩+≡
```
exp.nv.mischung()
```

Die Verteilung von Stichprobenfunktionen

Bootstrap-Algorithmus.

122 ⟨* 5⟩+≡
```
x<-zwischen.unfalls.zeiten.02; stpr.funktion<-median
B<-1000; result<-numeric(B); set.seed(17)
for(b in 1:B){
  stpr<-sample(x,size=length(x),replace=TRUE)
  result[b]<-stpr.funktion(stpr)
}
plot(density(result),main="Dichtespur")
summary(result)
```

123 ⟨* 5⟩+≡
```
std.err.dach<-sd(result)
```

Fazit.

6.4 Zur Konstruktion von Schätzfunktionen

6.4.1 Parameterschätzung nach der Methode der Momente

Beispiel Poisson-Verteilung.

Beispiel Exponentialverteilung.

124 ⟨* 5⟩+≡
```
1/mean(zwischen.unfalls.zeiten.02)
```

Beispiel Normalverteilung.

125 ⟨* 5⟩+≡
```
x<-dateigroessen
boxplot(log(x))
```

126 ⟨* 5⟩+≡
```
x<-log(dateigroessen)
hist(x, prob=T)
m<-mean(x); sq<-var(x)
curve(dnorm(x,m,sq^0.5),add=T)
c("mu.hat"=m,"sigma.q.hat"=sq)
```

127 ⟨* 5⟩+≡
```
sigma.q.dach.2<-mean(x*x)-mean(x)^2
```

Beispiel Anteilswert schätzen.

Fazit.

6.4.2 Parameterschätzung nach der ML-Methode

Momentenschätzer sind manchmal ungeeignet.

Vampirologie.

Unendlich viele Realisationsmöglichkeiten.

128 ⟨* 5⟩+≡
```
x<-c(3,1,2,2); lambda<-1
prod(dpois(x,lambda))
```

Stetiges Beobachtungsmerkmal.

Stetige Parametermenge.

Graphische Maximierung.

129 ⟨* 5⟩+≡
```
x<-c(3,1,2,2)
lambda<-seq(.5,5,length=100); L<-rep(0,length(lambda))
for(i in seq(lambda)) L[i]<-prod(dpois(x,lambda[i]))
plot(lambda,L,type="l")
```

130 ⟨* 5⟩+≡
```
exp.ml(c(3,1,2,2),"pois")
```

Schätzfunktion.

Bonbonanzahl abschätzen.

Mehrere Parameter.

Fazit.

6.4.3 Fragen an Schätzfunktionen

Annäherung eines Schätzers an den gesuchten Wert.

131 ⟨* 5⟩+≡
```
exp.nv.est()
```

132 ⟨* 5⟩+≡
```
n<-1000; set.seed(13) ;stpr<-rnorm(n,mean=0,sd=1)
mittel<-cumsum(stpr)/(1:n); plot(1:n,mittel,type="l")
```

Variabilität von Schätzern.

133 ⟨* 5⟩+≡
```
# Parameter festlegen
n <- 20; wd <- 100; lambda<-5; set.seed(13)
# Stichproben als Spalten in X ablegen
X <- matrix(rpois(n*wd, lambda),n,wd)
# Schaetzer: apply(X,2,mean) berechnet Spaltenmittel
lambda.dach<-apply(X,2,mean)
# Graphik erstellen
y.j<-2+10*jitter(rep(0,wd))
plot(lambda.dach,y.j,ylim=c(.5,2.5),ylab="",axes=F)
boxplot(lambda.dach,horizontal=T,add=T)
```

Variabilität des abgelieferten Modells.

134 ⟨* 5⟩+≡
```
# Experimentparameter festlegen
n <- 20; wd <-  30; lambda <- 1; set.seed(13)
# Stichprobe ziehen, Berechnungen durchfuehren
lambda.dach<-numeric(wd)
for(j in 1:wd){
  stpr<-rexp(n, lambda)
  lambda.dach[j]<-1/mean(stpr)
}
# Graphik erstellen
x<-seq(0,5,length=100)
plot(x,x,type="n",ylim=c(0,2))
for(j in 1:wd) lines(x,dexp(x,lambda.dach[j]))
```

6.5 Check des gefundenen Modells

6.5.1 Modellcheck

135 ⟨* 5⟩+≡
```
qqnorm(log(dateigroessen))
qqline(log(dateigroessen))
```

Ein falscher Modelltyp.

136 ⟨* 5⟩+≡
```
exp.nv.an.beta()
```

Ausreißerwirkung.

137 ⟨* 5⟩+≡
```
exp.outlier()
```

6.5.2 Beispiel: Unfalldaten

138 ⟨* 5⟩+≡
```
lambda02<-1/mean(zwischen.unfalls.zeiten.02)
lambda04<-1/mean(zwischen.unfalls.zeiten.04)
cat("Jahr 2002: lambda02 =",lambda02,"\n")
cat("Jahr 2004: lambda04 =",lambda04,"\n")
```

139 ⟨* 5⟩+≡
```
q.theo02<-rexp(100,lambda02)
qqplot(zwischen.unfalls.zeiten.02,q.theo02)
```

140 ⟨* 5⟩+≡
```
q.theo04<-rexp(100,lambda04)
qqplot(zwischen.unfalls.zeiten.04,q.theo04)
```

141 ⟨* 5⟩+≡
```
plot(zwischen.unfalls.zeiten.04)
```

142 ⟨* 5⟩+≡
```
z<-zwischen.unfalls.zeiten.04[-(1:14)]
lambda04a<-1/mean(z)
cat("lambda04a =",lambda04a,"\n")
```

143 ⟨* 5⟩+≡
```
q.exp<-rexp(100,lambda04a)
qqplot(z,q.exp)
```

144 ⟨* 5⟩+≡
```
wd<-200; set.seed(13)
n<-length(z); lambda.dach.vec<-numeric(n)
for(i in 1:wd){
  stpr<-sample(z,size=n,replace=T)
  lambda.dach.vec[i]<-1/mean(stpr)
}
hist(lambda.dach.vec,prob=T,main="")
rug(lambda.dach.vec); summary(lambda.dach.vec)
```

Zusammenfassung:

145 ⟨* 5⟩+≡
```
x<-c(44,63,57,33,36,30,36,30,37)
```

146 ⟨* 5⟩+≡
```
daten<-c(.01,.11,.20,.67,.71,.75,.80,.90,1.00,2.06,2.15,2.17
  ,2.18,2.26,2.36,2.38,2.53,2.81,2.98,3.35,3.37,3.47,4.04,4.74
  ,4.82,5.17,5.76,5.98,6.27,6.77,7.14,9.97,10.88,10.98,13.74
  ,18.64,23.16)
```

147 ⟨* 5⟩+≡

```
# Experimentparameter festlegen
n <- 20; wd <- 100; lambda<-5; set.seed(13)
# Stichproben ziehen und Mittel berechnen:
lambda.dach<-rep(0,wd)
for(j in 1:wd)
    lambda.dach[j]<-mean(rpois(n, lambda))
# Darstellung:
boxplot(lambda.dach,horizontal=T)
rug(lambda.dach+rnorm(wd„.1))
# alternativ:      plot(table(lambda.dach))
```

148 ⟨* 5⟩+≡

```
x<-handy; mu.dach<-mean(x); sd.dach<-sd(x)
hist(x,prob=TRUE)
# oder:
curve(dnorm(x,mu.dach,sd.dach),add=TRUE)
```

149 ⟨* 5⟩+≡

```
x<-c(36.92,32.72,60.84,48.38,51.28,77.74,8.28,
     4.98,1.28,4.58)
```

Kapitel 7

Konfidenzintervalle

7.1 Konfidenzintervall für den Median

150 ⟨* 5⟩+≡

```
n<-5;wd<-1000;set.seed(17);x.med<-qexp(.5);anz.in<-0
for(i in 1:wd){
  stpr<-rexp(n)
  anz.in<-anz.in+(min(stpr)<x.med & x.med<max(stpr))
}
cat("realle Ueberdeckungshaeufigkeit:",
    100*anz.in/wd, "%\n")
cat("theoretische Ueberdeckungshaeufigkeit:",
    100*(1-2^(1-n)), "%\n")
```

7.2 Was kostet der Wunsch?

7.2.1 Kann es nicht noch etwas vertrauenswürdiger sein?

7.2.2 Kann es nicht etwas kürzer sein?

7.2.3 Welches k zu vorgegebenem Konfidenzniveau γ?

7.3 Konstruktionsprinzip für Konfidenzintervalle

7.4 Konfidenzintervall für einen Anteil p

151 ⟨* 5⟩+≡

```
p.dach<-0.55; n<-100; alpha<-0.05
SE <- sqrt(p.dach*(1-p.dach)/n)
p.dach +c(-1,1)* qnorm(1-alpha/2) * SE
```

Ein Experiment.

152 ⟨* 5⟩+≡
```
exp.ki.p()
```

Konfidenzintervall für Anzahlen.

7.5 Fragen an Konfidenzintervalle

Das allgemeine Dilemma.

153 ⟨* 5⟩+≡
```
demo.n.alpha.len(p.dach=0.5)
```

Stichprobenumfang ermitteln.

154 ⟨* 5⟩+≡
```
IL<-0.02; alpha<-0.05; p.dach<-0.55
n<-4*p.dach*(1-p.dach)*(qnorm(1-alpha/2)/IL)^2
round(n)
```

155 ⟨* 5⟩+≡
```
IL<-0.02; alpha<-0.05;
p.dach<-0.05
n<-4*p.dach*(1-p.dach)*
   (qnorm(1-alpha/2)/IL)^2
round(n)
```

156 ⟨* 5⟩+≡
⟨*zeichne Zusammenhang n gegen* $1-\alpha$ 259⟩

Genauigkeit ermitteln.

157 ⟨* 5⟩+≡
```
alpha<-0.05; p.dach<-0.55
n<-c( 50, 100,  500, 1000,  5000, 10000)
IL<-2*qnorm(1-alpha/2)* sqrt(p.dach*(1-p.dach)/n)
il<-signif(IL,3)
IL <-format(il,digits=4)
rbind(n,IL)
```

158 ⟨* 5⟩+≡
⟨*zeichne Zusammenhang IL gegen n* 258⟩

Aussagensicherheit ermitteln.

159 ⟨* 5⟩+≡
```
n<-1000; IL<-0.01*c(3,4,5,6,7,8)
z<-IL/2*sqrt(n/(p.dach*(1-p.dach)))
alpha<-2*(1-pnorm(z))
signif(rbind(IL,alpha),2)
```

7.6 Konfidenzintervalle für die Normalverteilung

Ein Konfidenzintervall für μ.

Ein Konfidenzintervall für σ.

160 ⟨* 5⟩+≡
```
alpha<-0.05
chi.q<-qchisq(c(alpha/2,1-alpha/2),n-1)
```

Simultane Konfidenzintervalle.

Fragen an Konfidenzintervalle zur Normalverteilung.

7.7 Anwendung: Raucherrisiken

161 ⟨* 5⟩+≡
⟨*zeige Raucherdemo* 257⟩

7.8 Caveat – Mahnung

Kapitel 8

Statistik und BAYES

8.1 Ein Problem in klassischer Sicht

8.1.1 Euro keine Zufallswährung?

Korrekturen.

8.1.2 Zutreffend oder nicht? 1. Versuch

Eine Simulation.

162 ⟨* 5⟩+≡
```
set.seed(9191)
matrix(rbinom(30,250,0.5),10,3)
```

Theorie.

Anwendung.

163 ⟨* 5⟩+≡
```
pbinom(124,250,0.5)
```

164 ⟨* 5⟩+≡
```
3*pbinom(124,250,0.5)^3
```

Das Ereignis C.

165 ⟨* 5⟩+≡
```
n <- 2:250
m <- floor((n-1)/2)
y <- 3*pbinom(m,n,0.5)^3
plot(n,y,pch=18,bty="n")
```

Fazit.

8.1.3 Zutreffend oder nicht: 2. Versuch

Anwendung der Binomialverteilung.

Fazit.

8.1.4 Welches p?

8.1.5 Ein Modell

8.1.6 Zwei Lösungsvorschläge für das Schätzproblem

1. Methode der Momente.

2. Maximum Likelihood Schätzer.

Anwendung.

166 ⟨* 5⟩+≡
```
n  <- 250
yy <- 141
x  <-seq(0,1,0.005)
y  <-x^yy*(1-x)^(n-yy)
plot(x,y,bty="n",type="l")
points(yy/n,0,pch=6)
```

167 ⟨* 5⟩+≡
```
Lp.diff(250)
```

Freibrief.

Exkurs: Herleitung des ML-Schätzers.

8.1.7 Das p aus den p's

Fazit.

8.2 BAYES und der Euro

8.2.1 Sammlung und Typisierung von Informationen

8.2.2 Beschreibung durch Wahrscheinlichkeitsverteilungen

168 ⟨* 5⟩+≡
```
x <- seq(0,1,length=200)
y <- pbeta(x,250,250)
plot(x,y,type="l",bty="n")
title("beta(250,250)")
```

8.2.3 Per aspera ad astra

8.2.4 Eine Rechtfertigung?

8.2.5 Ein Bayesscher Schätzer

8.2.6 Uniform prior

8.2.7 Glaubwürdiger

8.3 Prior – Sample – Posterior

8.3.1 Beta als Prior

8.3.2 Gut oder schlecht?

8.3.3 Parameterfortschreibung

8.3.4 Der Euro und seine Prior

8.3.5 Play it again, Sam!

169 ⟨* 5⟩+≡
```
exp.bayes()
```

8.4 Beta-Verteilung

8.4.1 Dichte

8.4.2 Porträt

8.4.3 Try yourself

170 ⟨* 5⟩+≡
```
plot.beta()
```

8.4.4 Momente

8.5 Es hilft auch im Weltall

8.5.1 Ärger mit Ariane

8.5.2 Noch einmal Richtung BAYES

Glaubwürdiger.

8.5.3 Truncated uniform prior

Noch glaubwürdiger.

8.5.4 Anwendbarkeit des Bayesschen Ansatzes

Kapitel 9

Testen

9.1 Kochen und Testen

9.1.1 Das Problem: Zwiebelstatistik

9.1.2 Datenbeschaffung

9.1.3 Datensichtung und -reduktion

171 ⟨* 5⟩+≡
```
print(potato.m)
```

172 ⟨* 5⟩+≡
```
plot(potato.v,ylab="weight",xlab="Potato",bty="n")
```

173 ⟨* 5⟩+≡
⟨*simple.box.and.whisker.ordered* 267⟩

174 ⟨* 5⟩+≡
⟨*ordered.group.means* 268⟩

9.1.4 Vermutung

175 ⟨* 5⟩+≡
```
boxplot(mpo,mpo0,mpo1,
names=c("Alle",
"Gruppe M","Gruppe F"),
pars=list(boxwex=.3))[1]
```

9.1.5 Idee eines Tests

Betrachtung von Medianen.

176 ⟨* 5⟩+≡
⟨*simple.binomial.test* 269⟩

Binomialverteilung aktivieren.

177 ⟨* 5⟩+≡
```
dbinom(0:12,12,0.5)
```

Konfidenzintervall interpretieren.

178 ⟨* 5⟩+≡
⟨*simple.confidence.test* 270⟩

Verteilungsfunktionen und ihre Differenzen.

179 ⟨* 5⟩+≡
⟨*jeder.mit.jedem* 272⟩

180 ⟨* 5⟩+≡
```
smirkol(mpo0,mpo1)
```

181 ⟨* 5⟩+≡
```
ks.test(mpo0,mpo1)
```

9.2 Der Aufbau eines Tests

9.2.1 Hypothesen

9.2.2 Entscheidungen und Fehler

9.2.3 Teststatistik

9.2.4 Entscheidungsregel

Orientierung der Entscheidung am Kritischen Wert.

p-Wert (p-value).

9.2.5 Allgemeiner Fahrplan eines Tests mit Demonstration

Binomialtest als Demonstration.

182 ⟨* 5⟩+≡
```
pbinom(1,12,0.5)
```

9.2.6 Binomialtest mit R

183 ⟨* 5⟩+≡
```
binom.test(1,n=12,p=0.5,alternative='less')
```

Interaktive Darstellung des Binomialtests.

184 ⟨* 5⟩+≡
```
exp.binomialtest()
```

9.2.7 Die Gütefunktion eines Tests

185 ⟨* 5⟩+≡
```
exp.nv.guete()
```

9.3 Der χ^2-Test: Ein vielseitiger Geselle

9.3.1 Ist Lotto fair – passt die Gleichverteilung?

Das Datenmaterial.

186 ⟨* 5⟩+≡
⟨*zeige.lotto,tabelle* 265⟩

Operationalisieren der Hypothese.

Formulierung der Teststatistik des χ^2-Test.

Verteilung von X^2 unter H_0.

187 ⟨* 5⟩+≡
```
plot.chi()
```

Entscheidungsregel.

Durchführung mit ăR und Antwort.

188 ⟨* 5⟩+≡
```
chisq.test(lottab)
```

Ist Lotto fair – der χ^2-Test reloaded.

189 ⟨* 5⟩+≡
```
print(kugtab)
```

190 ⟨* 5⟩+≡
⟨*test.for.all* 264⟩

9.3.2 Der χ^2-Anpassungstest

Beispiel Wartezeit.

191 ⟨* 5⟩+≡
```
data<-diff(coal[,1])
n.breaks<-floor(length(data)/10)
breaks<-qexp((0:n.breaks)/n.breaks,191/111)
breaks[length(breaks)]<-1+max(data)
counts<-hist(data,breaks=breaks,plot=F)$counts
chisq.test(counts)
```

192 ⟨* 5⟩+≡
⟨*test.wartezeit* 263⟩

Kontinuierliche Marscherleichterung.

9.3.3 Opfer und Täter – der χ^2-Unabhängigkeitstest

Die Kontingenztabelle.

Besetzung bei Unabhängigkeit.

Wiederbelebung der Idee.

Opfer und Täter.

193 ⟨* 5⟩+≡
⟨*opfer.taeter.test* 262⟩

9.4 Eine kleine Testgalerie

9.4.1 Kolmogorov-Smirnov-Test

194 ⟨* 5⟩+≡
```
mu1<-mean(po1.v); sd1<-sd(po1.v)
ks.test(po0.v, pnorm, mu1, sd1)
```

195 ⟨* 5⟩+≡
```
ks.test(mpo1, mpo0)
```

9.4.2 Normalverteilung: Test auf μ bei bekanntem σ

9.4.3 Normalverteilung: Test auf μ bei unbekanntem σ

196 ⟨* 5⟩+≡
```
mu.1<-mean(po1.v)
t.test(mpo0, mu=mu.1)
```

9.4.4 Test auf Gleichheit der Mittelwerte

197 ⟨* 5⟩+≡
```
t.test(po1.v,po0.v,var.equal=TRUE)
```

198 ⟨* 5⟩+≡
```
t.test(po1.v,po0.v)
```

9.4.5 Vorzeichentest im Einstichprobenfall

199 ⟨* 5⟩+≡
```
mu.1<-mean(po1.v); x<-mpo0[mpo0!=mu.1]
x<-x[!is.na(x)]; n<-length(x)
T.minus<-sum(x<mu.1)
p.value<-pbinom(T.minus,n,0.5)
```

200 ⟨* 5⟩+≡
```
mu.1<-mean(po1.v);
x<-mpo0[mpo0!=mu.1]; x<-x[!is.na(x)]; n<-length(x)
T.minus<-sum(x<mu.1)
binom.test(T.minus,n,alternative="less")
```

9.4.6 Vorzeichentest im Zweistichprobenfall

9.4.7 Wilcoxon-Test für verbundene Stichproben

201 ⟨* 5⟩+≡
```
mu.1<-mean(po1.v); x<-mpo0[mpo0!=mu.1]
x<-x[!is.na(x)]; n<-length(x)
T.minus<-sum(x<mu.1)
p.value<-pbinom(T.minus,n,0.5)
wilcox.test(x,mu=mu.1)
```

202 ⟨* 5⟩+≡
```
x1<-c(89.146,71.874,112.348,73.962,101.136,
      84.570,124.650,82.736,NA,73.270,79.746)
x2<-c(69.800,82.330,79.402,83.088,70.758,87.256,
      66.744,65.894,56.020,82.296,83.312,81.672)
```

203 ⟨* 5⟩+≡
```
IQ<-c(98,100,104,104,102,102,104,94,94,
      103,105,99,102,103)
```

Kapitel 10

Regressionsanalyse

10.1 Eine Reise für den Überblick

Beispiel: Kraftstoffverbrauch.

204 ⟨* 5⟩+≡
```
plot(astra.km,astra.liter,main="Liter und km",
     xlab="km",ylab="Liter",ylim=c(0,50))
```

Modell-Auswahl. Welches Modell ist geeignet?

Modell-Anpassung. Bisweilen geht es auch ohne Formeln wie hier mit der Idee:

205 ⟨* 5⟩+≡
```
x<-astra.km; y<-astra.liter
fbe.fit(x,y)
```

Modell-Check.

206 ⟨* 5⟩+≡
```
residuals <- y - (-0.06779 + 0.07243*x)
plot(x,residuals,type="h")
```

Modell-Interpretation.

10.2 Das lineare Regressionsmodell

10.3 Modell-Schätzung und -Check

10.3.1 Die Methode der kleinsten Quadrate

Eigenschaften.

10.3.2 Ein Anwendungsbeispiel

KQ-Gerade zeichnen.

207 ⟨* 5⟩+≡
```
b.dach<-cov(x,y)/var(x)
a.dach<-mean(y)-b.dach*mean(x)
cat("Modellgerade: y = ",a.dach,"+ ",b.dach,"* x\n")
```

208 ⟨* 5⟩+≡
```
result<-lm(y~x)
plot(y~x); abline(result,col="blue")
cat("Modellgerade: y = ",result$coefficients[1])
cat("+ ",result$coefficients[2],"* x\n")
```

209 ⟨* 5⟩+≡
```
plot(residuals(result))
```

210 ⟨* 5⟩+≡
```
plot(x,result$residuals,type="h")
title("Residualplot")
```

Ausreißer entfernen.

211 ⟨* 5⟩+≡
```
ind<- 30<x & x<800
xx<-x[ind]; yy<-y[ind]
result<-lm(yy~xx)
plot(xx,yy); abline(result)
result$coef
```

212 ⟨* 5⟩+≡
```
xx<-x[ind]; yy<-y[ind]
plot(xx,lm(yy~xx)$resid,
     type="h",ylab="Residuen")
title("Residuen ohne Extremwerte")
```

10.3.3 Residualanalyse

Typische Residualplots.

Nichtlinearitäten.

213 ⟨* 5⟩+≡
```
exp.fit.line.to.poly()
```

Ausreißer.

214 ⟨* 5⟩+≡
```
exp.check.point.influence()
```

Verteilung der Residuen.

215 ⟨* 5⟩+≡
```
result<-lm(y~x)
qqnorm(result$resid); qqline(result$resid)
```

10.4 Modell-Interpretation

10.4.1 Das Bestimmtheitsmaß

216 ⟨* 5⟩+≡
```
x<-astra.km; y<-astra.liter
coefficients<-lm(y~x)$coefficients
y.dach<-coefficients[1]+coefficients[2]*x
Rq<-var(y.dach)/var(y)
```

Eigenschaften des Bestimmtheitsmaßes.

217 ⟨* 5⟩+≡
```
xx<-astra.km[1:10]; yy<-astra.liter[1:10]
show.rq(xx,yy)
```

218 ⟨* 5⟩+≡
```
xx<-astra.km[5:18]
show.rq(xx,xx)
```

219 ⟨* 5⟩+≡
```
print(var(y.dach)/var(y))
print(b.dach^2*var(x)/var(y))
print(cov(x,y)^2/(var(x)*var(y)))
print(cor(x,y)^2)
```

Eine Anpassungsübung.

220 ⟨* 5⟩+≡
```
exp.adjust.Rq()
```

Ausreißer. Wie reagiert R^2 auf Ausreißer?

221 ⟨* 5⟩+≡
```
exp.Rq.outlier()
```

10.4.2 Konfidenzintervalle für Achsenabschnitt und Steigung

222 ⟨* 5⟩+≡
```
x<-astra.km; y<-astra.liter
result<-ki.a.b(x,y,alpha=0.05,plot=TRUE)
```

10.4.3 $E(Y|x_0)$ und Prognose von $Y|x_0$

223 ⟨* 5⟩+≡
```
ki.y.dach(
  cars[,1],cars[,2],
  alpha=c(0.05,0.01),
  x0=seq(0,60,length=100))
```

10.4.4 Test und Modellvergleich

Testfragen.

P-value.

224 ⟨* 5⟩+≡
```
x<-cars[1:10,"speed"]
y<-cars[1:10,"dist"]
plot(x,y,xlab="speed",
  ylab="distance")
abline(lm(y~x),col="blue")
lsfit.b.pvalue(x,y)
```

ANOVA.

225 ⟨* 5⟩+≡
```
x<-cars[1:10,"speed"]; y<-cars[1:10,"dist"]
F.byhand(x,y)
```

226 ⟨* 5⟩+≡
```
x<-cars[1:10,"speed"]; y<-cars[1:10,"dist"]
result<-lm(y~x)
anova(result)
```

10.5 Ausblick

10.5.1 Mehrere erklärende Variablen

227 ⟨* 5⟩+≡
```
pairs(milchprod)
```

228 ⟨* 5⟩+≡
```
result<-lm(BW ~ KH + EW + FE,milchprod)
```

229 ⟨* 5⟩+≡
```
resid<-residuals(result)
plot(milchprod$BW,resid,type="h",
     ylab="Residuen")
```

230 ⟨* 5⟩+≡
```
anova(result)
```

231 ⟨* 5⟩+≡
```
result<-lm(BW ~ KH + FE,milchprod)
```

10.5.2 Nicht lineare Zusammenhänge

232 ⟨* 5⟩+≡
```
x<-cars[,"speed"]; y<-cars[,"dist"]
plot(x,y,xlab="speed",ylab="distance")
abline(lm(y~x),col="blue")
result<-lm(dist~speed+I(speed^2),cars)
coef<-result$coef
xx<-seq(min(x),max(x),length=100)
yy<-coef[1]+coef[2]*xx+coef[3]*xx^2
lines(xx,yy,col="red")
title("lineare und quadratische Anpassung")
```

233 ⟨* 5⟩+≡
```
plot(women)
abline(lm(weight ~ height,women),col="blue",lty=2)
result<-lm(weight ~ height + I(height^2),women)
xx<-58:72; yy<-outer(xx,0:2,"^")%*%result$coefficients
lines(xx,yy,col="red")
```

234 ⟨* 5⟩+≡
```
x<-astra.km; ind<-30<x & x<800
x<-x[ind]; y<-astra.liter[ind]
exp.regr.poly(x,y)
```

10.5.3 Variablentransformationen

235 ⟨* 5⟩+≡
```
library(MASS)
plot(Animals)
abline(lm(brain~body,Animals),col="blue")
```

236 ⟨* 5⟩+≡
```
lAnimals<-log(Animals)
colnames(lAnimals)<-c("log.body","log.brain")
plot(lAnimals,ylab="log(brain)",xlab="log(body)")
abline(lm(log.brain~log.body,lAnimals),col="blue")
```

10.5.4 Polynome und lokale Glätter

237 ⟨* 5⟩+≡
```
plot(lynx,type="b",pch=20)
result<-lowess(lynx,f=10/114)
lines(result,col="red")
title("Entwicklung des Luchsbestands")
```

238 ⟨* 5⟩+≡
```
x<-astra.km; ind<-30<x & x<800
x<-x[ind]; y<-astra.liter[ind]
plot(x,y); lines(lowess(x,y),col="red")
```

239 ⟨* 5⟩+≡
```
exp.regr.smooth(x,y)
```

Zusammenfassung:

240 ⟨* 5⟩+≡
```
anteil<-c(108,83,287,117,92,56,10,2,4,5,13,5,108,24,41,51,26,9)
teuerung<-c(-23,-9,-9,-18,-9,-8,-9,1,-4,12,2,2,-14,50,5,-3,1,16)
plot(anteil, teuerung)
```

241 ⟨* 5⟩+≡
```
ausgaben<-police[,1]; k.rate<-police[,2]
plot(ausgaben,k.rate,xlab="Ausgaben fuer Polizei",
     ylab="Kriminalitaetsrate")
```

Kapitel 11

R-Einführung

11.1 Hintergrund, Installation und erste Schritte mit R

Historie und Installation.

Die erste Sitzung.

242 ⟨* 5⟩+≡
```
123
```

243 ⟨* 5⟩+≡
```
data(co2)
plot(co2,type="l")
summary(co2)
```

Fehler und Tipperleichterung.

244 ⟨* 5⟩+≡
```
mean(co2))
```

245 ⟨* 5⟩+≡
```
mean(co2)
```

11.2 Daten einlesen und Statistiken berechnen

Datenobjekte und Zuweisung.

246 ⟨* 5⟩+≡
```
preis.netto <- 200
steuersatz <- 0.16
umsatzsteuer <- preis.netto*steuersatz
preis.netto * (1.00 + steuersatz)
```

247 ⟨* 5⟩+≡
```
anz.autos <- 0:5
h.anz.autos <- c(119,58,17,4,1,1)
summe.autos<-sum( anz.autos*h.anz.autos )
```

Indexzugriff.

248 ⟨* 5⟩+≡
```
h.anz.autos[2:6]
```

249 ⟨* 5⟩+≡
```
h.anz.autos[-1]
```

Tabellen und Matrizen.

250 ⟨* 5⟩+≡
```
h.tab.autos <- cbind(anz.autos,h.anz.autos)
```

251 ⟨* 5⟩+≡
```
sum(h.tab.autos[,1]*h.tab.autos[,2])
```

Funktionen und Hilfe.

252 ⟨* 5⟩+≡
```
help.start()
```

11.3 Graphiken erstellen

253 ⟨* 5⟩+≡
```
data(co2); x<-co2[1:20]; par(mfrow=2:3)
plot(x,type="l"); plot(x,type="p"); plot(x,type="b")
plot(x,type="h"); plot(x,type="s"); plot(x,type="n")
par(mfrow=c(1,1))
```

11.4 R als Rechnenmaschine

Statistische und mathematische Berechnungen.

Verteilungsmodelle.

Die Funktion ..

254 ⟨* 5⟩+≡

```
KG<-c(171,173,176,170,168,175,198,170,177,198,170,
      173,201,168,205,176,184,183,184,180)
set.seed(13)
stichprobe<-sample(x=KG,size=10,replace=T)
cat("Datensatz - Zusammenfassung\n")
print(summary(KG))
cat("Stichprobe\n")
print(stichprobe)
cat("Stichprobe - Zusammenfassung\n")
summary(stichprobe)
```

Kategoriale Daten.

Listen und Data-Frames.

Häufig verwendete R-Funktionen.

11.5 Bequemes Arbeiten mit diesem Buch in R

Gewährleistung.

11.6 Statistische R-Idioms

Häufigkeitsanalyse

Lage und Variabilität

Univariate Analyse

Bivariate Analyse

Beta-Verteilung –

Binomialverteilung

χ^2**-Verteilung** –

Hypergeometrische Verteilung

Poisson-Verteilung

Weitere Funktionen statistischer Modelle

Tests

Regressionen

Graphische Anweisungen

Experimente und Interaktives

Verwendete Datensätze

11.7 Weitere Infos

FAQ.

Papiere.

Glossar

Kapitel 12

Anhang

12.1 Funktionen, Daten und Code-Chunks

255 ⟨*start* 255⟩≡

```
exp.ki.p<-function(){
  redo<-function(...){
    n<-slider(no=1); alpha<-slider(no=2); wd<-slider(no=3)
    p<-slider(no=4); seed<-slider(no=5)
    set.seed(seed); u<-o<-numeric(wd)
    t.value<-qt(1-alpha/2,n-1)
    for(i in 1:wd){
       p.dach<-mean(rbinom(n,1,p))
       sigma.dach<-sqrt(p.dach*(1-p.dach)/n)
       step<-t.value*sigma.dach
       u[i]<-p.dach-step; o[i]<-p.dach+step
    }
    plot(u,type="n",ylim=0:1,bty="n",xlab="Versuche",
         ylab="realisierte KIs",axes=F)
         axis(2)
    title(paste("n=",n,", alpha=",alpha,
          ", p=",p,", seed=",seed,sep=""))
    segments(1:wd,u,1:wd,o)
    abline(h=p)
  }
  slider(redo,c("n","alpha","wd","p","seed"),
         c(5,.001,5,.01,1),
         c(500,.3,100,.99,100),
         c(1,.001,1,.01,1),
         c(10,.05,10,.5,1))
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

256 ⟨*start* 255⟩+≡
```
demo.n.alpha.len<-function(p.dach){
redo<-function(...){
a<-slider(no=1);b<-slider(no=2)
n.set<-10:100
alpha.set<-seq(.2,.01,length=20)
sigma.dach<-t.value<-matrix(0,length(alpha.set),length(n.set))
for(j in seq(n.set)){
  t.value[,j]<-qt(1-alpha.set/2,n.set[j]-1)
  sigma.dach[,j]<-sqrt(p.dach*(1-p.dach)/n.set[j])
}
length.KI<-t(2*t.value*sigma.dach)
persp(n.set,1-alpha.set,length.KI,
      phi=a,theta=b,ticktype="detailed",
      xlab="n",ylab="1-alpha")
}
slider(redo,c("a","b"), c(-180,-180), c(360,360), c(5,5),c(10,-30))
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

12.1.1 Codechunks

257 ⟨*zeige Raucherdemo* 257⟩≡
```
alpha<-0.05
abk<-c("Ia","Ib","Ic","II","III")
n<-c(21895,44184,25461,17366,33951)
p.dach<-c(32,86,56,106,66)/n
UG<-p.dach-qnorm(1-alpha/2) * sqrt(p.dach*(1-p.dach)/n)
OG<-p.dach+qnorm(1-alpha/2) * sqrt(p.dach*(1-p.dach)/n)
plot(1:5,ylim=c(0,max(1000*OG)),bty="n",type="n",
     axes=F,ylab="Sterberate in 1/1000",xlab="Gruppen")
title("KIs: Bronchialkrebs-WS und Rauchen")
axis(2); axis(1,at=1:5,labels=abk)
segments(1:5,1000*UG,1:5,1000*OG)
```

258 ⟨*zeichne Zusammenhang IL gegen n* 258⟩≡
```
#old<-par(mfrow=1:2)
# 1. Plot:
p.dach<-0.5; n<-c(10,11,12,14,16,20,25,30,40,50,70,100)
n<-c(n,10*n,100*n)
alpha.set<-c(.01,.02,.05,.1)
plot(n,n,type="n",ylim=0:1,log="x",ylab="IL",bty="n")
for(i in seq(alpha.set)){
  alpha<-alpha.set[i]
  IL<-2*qnorm(1-alpha/2)*sqrt(p.dach*(1-p.dach)/n)
  lines(n,IL,lty=i)
}
title(paste("Intervalllaenge gegen n\np.dach=",p.dach))
legend(500,.8,lty=seq(alpha.set),
legend=paste(" (1-alpha) =",1-alpha.set),bty="n")
```

259 ⟨*zeichne Zusammenhang n gegen* $1-\alpha$ 259⟩≡

```
p.dach<-0.5; alpha<-seq(.001,.2,length=100)
IL.set<-c(.01,.02,.04,.08,.13)
plot(1,type="n",xlim=c(.8,1),ylim=c(50,50000),
    log="y",xlab="1-alpha",ylab="n",bty="n")
for(i in seq(IL.set)){
  IL<-IL.set[i]
  n<-4*p.dach*(1-p.dach)*(qnorm(1-alpha/2)/IL)^2
  lines(1-alpha,n,lty=i)
}
title(paste("Stichprobenumfang gegen 1-alpha\np.dach=",p.dach))
legend(.95,150,lty=seq(IL.set),
legend=paste(" IL =",IL.set),bty="n")
```

12.1.2 Erstellung von `kif1.ps`

260 ⟨* 5⟩+≡

```
x<-seq(.1,40,length=200)
f.x<-dchisq(x,15)
alpha.u<-.1; alpha.o<-.1
x.u<-qchisq(alpha.u,15)
x.o<-qchisq(1-alpha.o,15)
plot(x,f.x,type="l",bty="n",ylab="",xlab="",axes=FALSE)
xu<-x[x<=x.u]
f.xu<-dchisq(xu,15)
lines(xu,f.xu,type="h")
xo<-x[x>=x.o]
f.xo<-dchisq(xo,15)
lines(xo,f.xo,type="h")
axis(1,at=c(x.u,x.o),labels=expression(x[u],x[o]))
text((x.u+x.o)/2,.01,cex=2,expression(1-alpha))
text((x.u)/3,.03,cex=2,expression(alpha[u]))
text((x.o+x.u),.03,cex=2,expression(alpha[o]))
arrows((x.u)/3,.025,(x.u)/1.6,0.015)
arrows((x.u+x.o),.025,(x.u+x.o)/1.2,0.015)
```

261 ⟨*do.power* 261⟩≡
```
p <- seq(0,1,.01)
n <- 12
k <- 2
beta1 <- pbinom(k,n,p) + 1 - pbinom(n-k-1,n,p)
plot(p,beta1,type="l",ylab="beta(p)",ylim=c(0,1),bty="n")
#title("Powerfunktionen")
k <- 3
beta2 <- pbinom(k,n,p) + 1 - pbinom(n-k-1,n,p)
lines(p,beta2,lty=2)
k <- 4
beta3 <- pbinom(k,n,p) + 1 - pbinom(n-k-1,n,p)
lines(p,beta3,lty=3)
lines(c(0,0.075),c(0.05,0.05),lty=1)
text(0.125,0.05,"k=2")
lines(c(0,0.075),c(0.1,0.1),lty=2)
text(0.125,0.1,"k=3")
lines(c(0,0.075),c(0.15,0.15),lty=3)
text(0.125,0.15,"k=4")
```

262 ⟨*opfer.taeter.test* 262⟩≡
```
xx <- matrix(c(627,254,5010,99),nc=2,byrow=TRUE)
chisq.test(xx)
```

263 ⟨*test.wartezeit* 263⟩≡
```
w <- diff(coal[,1])
h <- hist(w,nclass=30,plot=FALSE)
br <- h$breaks
th <- pexp(br[-1],191/111)
pth <-c(th,1)-c(0,th)
co <- h$counts
print(chisq.test(c(co,0),p=pth))
```

264 ⟨*test.for.all* 264⟩≡
```
erg <- NULL
for (i in 1:6){
e <- chisq.test(kugtab[,i])
erg <- rbind(erg,c(e$statistic,e$p.value,48))}
kz <- c("X-squared","p-value","df")
ks <- c("kug1","kug2","kug3","kug4","kug5","kug6")
dimnames(erg) <- list(ks,kz)
print(erg)
#print(format(erg[,1:2],digits=1))
```

265 ⟨*zeige.lotto,tabelle* 265⟩≡
```
showlottotabelle(lottab)
```

266 ⟨*num.aus* 266⟩≡
```
nt  <- sum(lottab)
nit <- rep(nt/49,49)
h   <- ((lottab-nit)^2)/nit
ch2 <- sum(h)
h1 <- qchisq(.95,48)
h2 <- pchisq(ch2,48)
h3 <- 1-h2
print(c(h1,h2,h3))
erg <- cbind(lottab,nit,h)
dimnames(erg)<- list(NULL,c("n_i","h_i","d_i^2"))
options(digits=4)
cat("\nArbeitstabelle\n")
print(erg)
cat("\nX^2\n")
print(ch2)
cat("\np-value\n")
print(h3)
options(digits=7)
```

267 ⟨*simple.box.and.whisker.ordered* 267⟩≡
```
h.m <- rbind(po1.m,po0.m)
plot(c(po1.v,po0.v),ylab="weight",bty="n")
title("Potato: Ordered Data + box-and-whisker")
medw <- apply(h.m,1,median)
maxw <- apply(h.m,1,max)
minw <- apply(h.m,1,min)
ii   <- 3+(0:22)*5
dd   <- 1.
segments(ii,minw,ii,maxw,lty=3)
segments(ii-dd,medw,ii+dd,medw,lty=3)
segments(ii-dd,maxw,ii+dd,maxw,lty=3)
segments(ii-dd,minw,ii+dd,minw,lty=3)
abline(v=length(po1.v)+.5,lty=2)
```

268 ⟨*ordered.group.means* 268⟩≡
```
plot(c(mpo1,mpo0),ylab="weight",bty="n",type="n")
title("Potato: Group Means of Ordered Data + C-lines")
abline(h=mmpo1,lty=2)
abline(h=mmpo0,lty=2)
abline(h=mmpo)
lmpo1 <- length(mpo1)
abline(v=lmpo1+.5,lty=3)
text(1:lmpo1,mpo1,"F")
text((lmpo1+1):(lmpo1+length(mpo0)),mpo0,"M")
text(3,112,"M = Mann")
text(3,116,"F = Frau")
```

269 ⟨*simple.binomial.test* 269⟩≡
```
x0    <- mpo0[!is.na(mpo0)]
x1    <- mpo1[!is.na(mpo1)]
medp0 <- median(x0)
medp1 <- median(x1)
v1    <- x1 < medp0
v0    <- x0 < medp1
le0 <- rep(1,length(x0))
le1 <- rep(2,length(x1))
plot(c(x1,x0),c(le1,le0),bty="n",
xlab="mean(potato)",ylab="",ylim=c(0,3),pch=18,yaxt="n")
#title("Simple Test by Plotting and Counting")
mi0 <- min(x0)
mi1 <- min(x1)
ma0 <- max(x0)
ma1 <- max(x1)
y0  <- 1.2
y00 <-  .9
y1  <- 1.8
y11 <- 2.1
segments(c(mi0,mi1),c(y0,y1),c(ma0,ma1),c(y0,y1))
points(medp0,y0,pch=3)
points(medp1,y1,pch=3)
lines(c(medp0,medp0),c(y0,y11),lty=3)
lines(c(medp1,medp1),c(y00,y1),lty=3)
text(medp1,y00,paste(sum(v0),sum(!v0)))
text(medp0,y11,paste(sum(v1),sum(!v1)))
text(mi0-5,y0,"M")
text(ma1+5,y1,"F")
```

270 ⟨*simple.confidence.test* 270⟩≡
```
## data and hypothesis cleared for NA
d1 <- mpo0[!is.na(mpo0)]
h1 <- median(mpo1,na.rm=TRUE)
d2 <- mpo1[!is.na(mpo1)]
h2 <- median(mpo0,na.rm=TRUE)
## doing the test
par(mfrow=1:2)
conf.test1(d1,h1,txt="Median Gr F vs Gr M")
conf.test1(d2,h2,txt="Median Gr M vs Gr F")
par(mfrow=c(1,1))
```

271 ⟨*binomialtabelle* 271⟩≡
```
#Code zur Erzeugung der Binomialtabelle:
for(n in c(3,4,5,6,8,10,12,14,16,20,30)){
 cat("n=",n,"\\\\\n")
 for(j in 0:n){
  cat("&",j,"&")
  cat(paste(signif(pbinom(j,n,(1:5)/10),4),collapse="&"))
  cat("\\\\[0.9ex]\n")
 }
}
```

272 ⟨*jeder.mit.jedem* 272⟩≡
```
par(mfrow=c(3,3))
multiempver(mpo0,txt="Gr M")
multiempver(mpo0,mpo1,txt="Gr M, Gr F")
multiempver(mpo0,mpo,txt="Gr M, All")
multiempver(mpo0,mpo1,mpo,txt="Gr M,Gr F,All")
multiempver(mpo1,txt="Gr F")
multiempver(mpo1,mpo,txt="Gr F, All")
multiempver(mpo0,mpo1,mpo,txt="Gr M,Gr F,All")
multiempver(mpo0,mpo1,mpo,txt="Gr M,Gr F,All")
multiempver(mpo,txt="All")
par(mfrow=c(1,1))
```

12.1.3 `plot.chi`

273 ⟨*start* 255⟩+≡
```
plot.chi <- function(){
refresh.code <- function(...){
par(mfrow=c(1,2))
nu     <- n1 <- slider(no=1)
x      <- seq(0.01,149.99,.05)
ychi   <- y  <- dchisq(x,n1)
yychi  <- yy <- pchisq(x,n1)
yg     <- slider(obj.name="yg")
yyg    <- slider(obj.name="yyg")
ymax   <- max(c(max(y),max(yg)))
plot(x,y,type="l",ylim=c(0,ymax),ylab="",bty="n")
title("Chi: Density")
lines(x,yg,lty=2)
plot(x,yy,type="l",ylim=c(0,1),ylab="",bty="n")
title("Chi: CDF")
lines(x,yyg,lty=2)
par(mfrow=c(1,1)) }

reset.func <- function(...){
par(mfrow=c(1,2))
x   <- seq(0.01,149.99,.05)
nuu  <- 10
yg  <- dchisq(x,nuu)
yyg <- pchisq(x,nuu)
plot(x,yg,type="l",ylab="",bty="n")
title("Chisq: Density")
plot(x,yyg,type="l",ylim=c(0,1),ylab="",bty="n")
title("Chisq: CDF")
slider(obj.name="yg",obj.value=yg)
slider(obj.name="yyg",obj.value=yyg)
par(mfrow=c(1,1)) }
nu<-10
slider(refresh.code,sl.names="nu",1,50,1,
nu,reset.function=reset.func,
title="Chi-Verteilung")

reset.func()
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
 }
```

274 ⟨*start* 255⟩+≡

```
exp.binomialtest<-function(){
if(!exists("slider")) source("http://www.wiwi.uni-bielefeld.de/~wolf/software/R-wtools
slider(obj.name="alternative",obj.value="two.sided")
redo<-function(...){
n<-slider(no=1); p<-slider(no=2); alpha<-slider(no=3)
alternative<-slider(obj.name="alternative")
H<-paste( "H: p=",p)
if(alternative=="less") H<-paste( "H: p>=",p)
if(alternative=="greater") H<-paste( "H: p<=",p)
x<-0:n
if(alternative=="two.sided") alpha.h<-alpha/2 else alpha.h<-alpha
F.x<-pbinom(x,n,p)
plot(x,F.x,bty="n",ylim=0:1)
segments(x[-(n+1)],F.x[-(n+1)],x[-1],F.x[-(n+1)])
if(alternative!="greater"){
 abline(h=alpha.h)#% "red"
 k.stern.ind<-sum(F.x<=alpha.h)
 k.stern1<-x[k.stern.ind]
 segments(k.stern1,0,k.stern1, alpha.h)#% "blue"
 segments(k.stern1,alpha.h,k.stern1,
    if(alternative=="less") 1 else 1-alpha.h,lty=2)#% "blue"
 segments(0,.5,k.stern1,.5)#% "blue"
 text(k.stern1/2,.53,paste("gegen",H))#% "blue"
}
if(alternative!="less"){
 abline(h=1-alpha.h)#% "red"
 k.stern.ind<-length(x)+1-sum(F.x>=(1-alpha.h))
 k.stern2<-x[k.stern.ind]
 segments(k.stern2,1,k.stern2,1-alpha.h)#% "blue"
 segments(k.stern2,
    if(alternative=="greater") 0 else alpha.h,
    k.stern2, 1-alpha.h,lty=2)#% "blue"
 segments(n,.5,k.stern2,.5)#% "blue"
 text((k.stern2+n)/2,.53,paste("gegen",H))#% "blue"
}
 title(paste("n=",n,", p=",p,", alpha=",alpha,
   if(alternative=="less")
     paste("\nK= { 0,...,",k.stern1,"}"),
   if(alternative=="greater")
     paste("\nK= {",k.stern2,",...,",n,"}"),
   if(alternative=="two.sided")
     paste("\nK={0,...,",k.stern1,",",k.stern2,",...,",n,"}")
  ))
}
fbut.two.sided<-function(...){
   slider(obj.name="alternative",obj.value="two.sided")
  redo()
}
fbut.less<-function(...){
   slider(obj.name="alternative",obj.value="less")
  redo()
}
fbut.greater<-function(...){
   slider(obj.name="alternative",obj.value="greater")
  redo()
}
slider(redo,c("n","p.0","alpha"),
```

```
  c(2,.01,.01),  c(100,.99,.3),  c(1,.01,.01),  c(28,.5,.05),
  c(fbut.two.sided,fbut.less,fbut.greater),
  c("G: p!=p.0","G: p<p.0","G: p>p.0"),title="Binomialtest"
)
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

275 ⟨*start* 255⟩+≡

```
exp.nv.guete<-function(){
if(!exists("slider")) source("http://www.wiwi.uni-bielefeld.de/~wolf/software/R-wtools
slider(obj.name="alternative",obj.value="two.sided")
redo<-function(...){
n<-slider(no=1); mu0<-slider(no=2); mu1<-slider(no=3)
alpha<-slider(no=4) # ; beta<-slider(no=5)
agr<-mu1>mu0 # agr==1  <-> greater
alternative<-slider(obj.name="alternative")
delta<-4; sdx<-1/sqrt(n)
x<-seq(min(mu0,mu1)-delta,max(mu0,mu1)+delta,length=200)
fH.x<-dnorm(x,mu0,sdx)
fG.x<-dnorm(x,mu1,sdx)
par(mfrow=2:1)
if(alternative!="two.sided"){
 agr<-alternative=="greater"
 k.stern<-qnorm(if(agr) 1-alpha else alpha,mu0,sdx)
 beta<-if(agr) pnorm(k.stern,mu1,sdx) else 1-pnorm(k.stern,mu1,sdx)
 beta.all<-if(agr) pnorm(k.stern,x,sdx) else 1-pnorm(k.stern,x,sdx)
 plot(x,fH.x,bty="n",type="l",ylim=c(0,max(fH.x)*1.2),
      ylab="Verteilung PG unter H bzw. G",xlab="Parameter")
 title(paste("H: mean =",mu0,
             if(agr)"/  G: mean >" else ",  G: mean <",mu0))
 lines(x,fG.x,col="red")
 ind<-if(agr) x>=k.stern else x<=k.stern
 lines(x[ind],fH.x[ind],type="h",col="black")
 lines(x[!ind],fG.x[!ind],type="h",col="red")
 usr<-par()$usr
 segments(mu1,0,mu1,0.9*usr[4],lty=3,col="red")
 text(mu1,usr[4]*.96,paste("true mean:",mu1),col="red")
 plot(x,x,type="n",bty="n",ylim=c(0,1.2),
    ylab="WS: Beobachtung in K",xlab="Parameter")
 title("power")
segments(mu0,0,mu0,alpha)
 arrows(mu0+1.5,alpha/2,mu0,alpha/2); text(mu0+2,alpha/2+.03,"alpha")
 arrows(mu1+1.5,1-beta/2,mu1,1-beta/2,col="red")
 text(mu1+2,1-beta/2-.03,"beta",col="red")
 segments(mu1,1-beta,mu1,1,col="red")
 lines(x,1-beta.all,lty=1,col="red")
 usr<-par()$usr
 text(mu1,usr[4]*.96,paste("true mean:",mu1),col="red")
} else {
 alpha<-alpha/2
 k.stern<-qnorm(c(1-alpha, alpha),mu0,sdx)
 beta<-pnorm(k.stern[1],mu1,sdx)-pnorm(k.stern[2],mu1,sdx)
 beta.all<-pnorm(k.stern[1],x,sdx)-pnorm(k.stern[2],x,sdx)
 plot(x,fH.x,bty="n",type="l",ylim=c(0,max(fH.x)*1.2),
      ylab="Verteilung PG unter H bzw. G",xlab="Parameter")
 title(paste("H: mean =",mu0,"/  G: mean not equal",mu0))
 lines(x,fG.x,col="red")
 ind<- x>=k.stern[1] | x<=k.stern[2]
 lines(x[ind],fH.x[ind],type="h",col="black")
 lines(x[!ind],fG.x[!ind],type="h",col="red")
 usr<-par()$usr
 segments(mu1,0,mu1,0.9*usr[4],lty=3,col="red")
 text(mu1,usr[4]*.96,paste("true mean",mu1),col="red")
 plot(x,x,type="n",bty="n",ylim=c(0,1.2),
    ylab="WS: Beobachtung in K",xlab="Parameter")
```

```
 title("power")
 segments(mu0,0,mu0,alpha*2)
 arrows(mu0+1.5,alpha,mu0,alpha); text(mu0+2,alpha+.03,"alpha")
 arrows(mu1+1.5,1-beta/2,mu1,1-beta/2,col="red")
 text(mu1+2,1-beta/2-.03,"beta",col="red")
 segments(mu1,1-beta,mu1,1,col="red")
 lines(x,1-beta.all,lty=1,col="red")
 usr<-par()$usr
 text(mu1,usr[4]*.96,paste("true mean",mu1),col="red")
}
par(mfrow=c(1,1))
}
fbut.two.sided<-function(...){
   slider(obj.name="alternative",obj.value="two.sided")
  redo()
}
fbut.less<-function(...){
   slider(obj.name="alternative",obj.value="less")
  redo()
}
fbut.greater<-function(...){
   slider(obj.name="alternative",obj.value="greater")
  redo()
}
slider(redo,c("n","H: mean=mu.0","True mean","alpha"),
  c(2,-5,-5,.01),c(100,5,5,.3),c(1,.1,.1,.01),c(6,1,1.5,.1),
  c(fbut.two.sided,fbut.less,fbut.greater),
  c("G: mean <> mu.0","G: mean < mu.0","G: mean > mu.0"),
title="Guete")
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

276 ⟨*start* 255⟩+≡

```
conf.test1 <- function(data,hypo,txt="Test per Konfidenzintervalle"){
# no NA expected
data <- sort(data)
n  <- length(data)
k  <- floor(n/2)
ju <- 1:k
jo <- n+1-ju
ma <- max(data)
mi <- min(data)
plot(c(mi,ma),c(1,k),type="n",xlab="data",ylab="k",bty="n",las=1,crt=90)
title(txt)
segments(data[ju],ju,data[jo],ju)
abline(v=hypo,lty=2)
invisible() }
```

277 ⟨*start* 255⟩+≡

```
smirkol <- function(dat1,dat2,PLOT=TRUE,PRINT=FALSE){
h  <- CEV.gr(dat1)
x  <- h[,1]
px <- h[,2]
h  <- CEV.gr(dat2)
y  <- h[,1]
py <- h[,2]
nx <- length(x)
ny <- length(y)
n  <- nx+ny
xy <- c(x,y)
gr <- c(rep(1,nx),rep(2,ny))
gr <- gr[order(xy)]
xy <- sort(xy)
lx <- ly <- 0
ix <- iy <- 1
dd <- 0
di <- id <- c(0,0,0)

GO <- TRUE
if(nx==ny&&sum(x==y)==nx&&sum(px==py)==nx) GO <- FALSE

if(GO) {

for (k in (1:n)) {
  if(gr[k]==1) {
    lx <- px[ix]
    wo <- x[ix]
    di <- c(gr[k],ix,iy)
    ix <- ix+1
  }else{
    ly <- py[iy]
    wo <- y[iy]
    di <- c(gr[k],ix,iy)
    iy <- iy+1 }
  dn <- lx-ly
  if(abs(dn) > dd) {
    dd  <- abs(dn)
    pk  <- c(wo,lx,ly)
    id  <- di }
}
erg <- c(dd,id,pk)

}else{
cat("\nVerteilungen gleich\n")
erg <- NA }

if(PLOT) {
  multiempver(dat1,dat2)
  if(GO) lines(c(erg[5],erg[5]),c(erg[6],erg[7]),col="red",lty=3)}

return(erg)
}
```

278 ⟨*start* 255⟩+≡

```
multiempver <- function(d1,d2,d3,txt=""){
if(missing(d3)) d3 <- d1
if(missing(d2)) d2 <- d1
h1   <- CEV.gr(d1)
h2   <- CEV.gr(d2)
h3   <- CEV.gr(d3)
d1x  <- h1[,1]
d2x  <- h2[,1]
d3x  <- h3[,1]
y1   <- h1[,2]
y2   <- h2[,2]
y3   <- h3[,2]
r1  <- range(d1x)
s1  <- (r1[2]-r1[1])/20
r2  <- range(d2x)
s2  <- (r2[2]-r2[1])/20
r3  <- range(d3x)
s3  <- (r3[2]-r3[1])/20
xma <- max(r1[2],r2[2],r3[2])+5*max(s1,s2,s3)
xmi <- min(r1[1],r2[1],r3[1])-5*max(s1,s2,s3)
plot(c(xma,xmi),c(0,1),type="n",ylab="F",xlab="x",bty="n")
title(paste("Emp. Verteil.:",txt))
segments(c(xmi,d1x),c(0,y1),c(d1x,max(d1x)+xma),c(0,y1),lty=1)
segments(c(xmi,d2x),c(0,y2),c(d2x,max(d2x)+xma),c(0,y2),lty=2)
segments(c(xmi,d3x),c(0,y3),c(d3x,max(d3x)+xma),c(0,y3),lty=3)
#points(d1x,y1,pch=18)
#points(d2x,y2,pch=18)
#points(d3x,y3,pch=18)
invisible() }
```

279 ⟨*start* 255⟩+≡

```
CEV.gr <- function(dat) {
# clear for NA
d1  <- dat[!is.na(dat)]
d1x <- sort(d1)
n1  <- length(d1x)
# clear for multiple values
d2  <- c(d1x[-1],2*d1x[n1])
jj  <- d1x!=d2
y   <- ((1:n1)/n1)[jj]
x   <- d1x[jj]
erg <- cbind(x,y)
return(erg) }
```

280 ⟨*start* 255⟩+≡

```
showlottotabelle <- function(tab){
print(tab[1:10])
print(tab[11:20])
print(tab[21:30])
print(tab[31:40])
print(tab[41:49])}
```

12.1.4 Beispiel Lotto

Verteilung Ziffern

281 ⟨*start* 255⟩+≡

```
"lottab" <-
structure(as.integer(c(307, 322, 321, 300, 305, 322, 305, 285,
314, 299, 309, 302, 250, 292, 300, 294, 311, 315, 311, 298, 322,
307, 290, 297, 316, 322, 319, 278, 303, 295, 316, 359, 320, 279,
310, 321, 311, 345, 315, 308, 310, 320, 311, 308, 277, 304, 299,
326, 346)), .Dim = as.integer(49), .Dimnames = structure(list(
    c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
    "11", "12", "13", "14", "15", "16", "17", "18", "19", "20",
    "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
    "31", "32", "33", "34", "35", "36", "37", "38", "39", "40",
    "41", "42", "43", "44", "45", "46", "47", "48", "49")),
.Names = ""), class = "table")
```

Verteilung Lotto alle Ziehungen

282 ⟨*start* 255⟩+≡

```
"kugtab" <-
structure(as.integer(c(52, 53, 57, 45, 49, 45, 39, 49, 53, 46,
53, 51, 47, 44, 51, 49, 61, 54, 56, 53, 51, 48, 52, 57, 42, 59,
57, 40, 42, 52, 58, 58, 55, 42, 51, 57, 61, 61, 48, 57, 55, 54,
46, 44, 54, 59, 57, 41, 51, 54, 59, 48, 55, 48, 41, 60, 53, 49,
56, 67, 49, 49, 41, 46, 57, 53, 46, 40, 47, 45, 50, 44, 45, 47,
49, 49, 44, 48, 36, 55, 79, 70, 45, 46, 49, 49, 55, 61, 47, 50,
60, 44, 56, 42, 50, 58, 63, 62, 53, 51, 53, 57, 46, 70, 50, 43,
44, 48, 36, 57, 42, 48, 48, 46, 54, 55, 58, 55, 53, 46, 42, 46,
44, 53, 52, 51, 53, 53, 60, 57, 64, 50, 54, 54, 36, 64, 58, 48,
48, 63, 54, 59, 30, 47, 47, 59, 57, 47, 57, 68, 44, 52, 44, 61,
47, 44, 48, 46, 50, 39, 46, 33, 50, 55, 49, 60, 44, 56, 56, 43,
53, 60, 50, 59, 53, 51, 51, 50, 59, 49, 54, 48, 55, 49, 55, 50,
61, 46, 39, 62, 63, 54, 52, 36, 46, 72, 56, 48, 45, 45, 51, 64,
43, 48, 58, 61, 48, 43, 28, 55, 61, 40, 47, 50, 47, 53, 64, 56,
56, 49, 60, 58, 52, 55, 57, 51, 45, 54, 43, 43, 63, 53, 52, 45,
45, 48, 68, 54, 60, 47, 44, 41, 55, 59, 48, 45, 54, 50, 54, 59,
58, 52, 45, 66, 40, 59, 52, 45, 58, 61, 52, 41, 61, 50, 46, 53,
51, 53, 47, 63, 53, 50, 35, 52, 52, 48, 52, 39, 45, 48, 53, 64,
65, 53, 47, 43, 50, 45, 39, 53, 55, 46, 58, 56)),
.Dim = as.integer(c(49, 6)),
 .Dimnames = list(c("1", "2", "3", "4", "5", "6", "7", "8",
"9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19",
"20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
"31", "32", "33", "34", "35", "36", "37", "38", "39", "40", "41",
"42", "43", "44", "45", "46", "47", "48", "49"), c("kug1",
"kug2", "kug3", "kug4", "kug5", "kug6")))
```

12.1.5 Beispiel Kohlengruben

283 ⟨*start* 255⟩+≡

```
"coal"<-structure(list(date = c(1851.2, 1851.63, 1851.97, 1851.97, 1852.31,
1852.35, 1852.36, 1852.39, 1852.98, 1853.2, 1853.23, 1853.32,
1853.5, 1854.13, 1856.4, 1856.51, 1856.54, 1856.62, 1857.14,
1857.4, 1857.58, 1858.09, 1858.15, 1858.41, 1858.95, 1860.13,
1860.17, 1860.59, 1860.85, 1860.92, 1860.97, 1861.18, 1861.74,
1861.84, 1862.14, 1862.89, 1862.94, 1863.18, 1863.79, 1863.94,
1863.99, 1865.46, 1865.97, 1866.06, 1866.34, 1866.45, 1866.83,
1866.95, 1866.95, 1867.64, 1867.85, 1867.86, 1868.75, 1868.9,
1868.99, 1869.25, 1869.44, 1869.55, 1869.81, 1869.87, 1870.12,
1870.52, 1870.56, 1870.63, 1871.03, 1871.15, 1871.17, 1871.74,
1871.82, 1872.12, 1872.24, 1872.77, 1873.14, 1874.29, 1874.55,
1874.89, 1874.98, 1875.33, 1875.93, 1875.93, 1875.93, 1876.97,
1877.06, 1877.11, 1877.19, 1877.78, 1877.81, 1878.18, 1878.2,
1878.24, 1878.43, 1878.7, 1879.04, 1879.17, 1879.5, 1880.06,
1880.54, 1880.69, 1880.94, 1881.11, 1881.97, 1882.13, 1882.3,
1882.3, 1882.33, 1882.85, 1883.8, 1883.85, 1884.07, 1884.86,
1885.17, 1885.46, 1885.98, 1886.62, 1886.69, 1886.75, 1886.92,
1887.13, 1887.41, 1888.3, 1889.05, 1889.2, 1889.79, 1890.1, 1890.19,
1891.25, 1891.67, 1892.65, 1893.51, 1894.48, 1895.32, 1896.07,
1896.28, 1896.33, 1899.63, 1901.39, 1902.67, 1905.06, 1905.19,
1905.52, 1906.77, 1908.14, 1908.27, 1908.63, 1909.13, 1909.83,
1910.36, 1910.97, 1912.52, 1913.78, 1914.41, 1916.62, 1918.03,
1922.53, 1922.68, 1923.57, 1927.16, 1928.11, 1930.15, 1930.75,
1931.08, 1931.83, 1931.88, 1932.07, 1932.86, 1932.88, 1933.88,
1934.72, 1935.64, 1935.7, 1936.6, 1937.5, 1938.35, 1939.82, 1940.22,
1940.42, 1941.42, 1941.52, 1941.57, 1942, 1942.13, 1942.48, 1946.95,
1947.02, 1947.62, 1947.64, 1947.69, 1951.41, 1957.88, 1960.49,
1962.22)), .Names = "date", row.names = c("1", "2", "3", "4",
"5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15",
"16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26",
"27", "28", "29", "30", "31", "32", "33", "34", "35", "36", "37",
"38", "39", "40", "41", "42", "43", "44", "45", "46", "47", "48",
"49", "50", "51", "52", "53", "54", "55", "56", "57", "58", "59",
"60", "61", "62", "63", "64", "65", "66", "67", "68", "69", "70",
"71", "72", "73", "74", "75", "76", "77", "78", "79", "80", "81",
"82", "83", "84", "85", "86", "87", "88", "89", "90", "91", "92",
"93", "94", "95", "96", "97", "98", "99", "100", "101", "102",
"103", "104", "105", "106", "107", "108", "109", "110", "111",
"112", "113", "114", "115", "116", "117", "118", "119", "120",
"121", "122", "123", "124", "125", "126", "127", "128", "129",
"130", "131", "132", "133", "134", "135", "136", "137", "138",
"139", "140", "141", "142", "143", "144", "145", "146", "147",
"148", "149", "150", "151", "152", "153", "154", "155", "156",
"157", "158", "159", "160", "161", "162", "163", "164", "165",
"166", "167", "168", "169", "170", "171", "172", "173", "174",
"175", "176", "177", "178", "179", "180", "181", "182", "183",
"184", "185", "186", "187", "188", "189", "190", "191"), class = "data.frame")
```

12.1.6 Daten: Zwiebelbeispiel

Gruppenaufteilung

284 ⟨*start* 255⟩+≡
```
gender <- c(1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1,
1, 1, 0, 0, 1, 0, 0, 0)
ind <- length(gender)
i1  <- (1:ind)[!!gender]
i0  <- (1:ind)[!gender]
```

Kartoffeln

285 ⟨*start* 255⟩+≡
```
potato.v <- c(77.33, 83.93, 71.61, 53.76, 62.37,
 83.56, 82.25, 76.96, 96.34,106.62,
113.71, 84.83, 70.15, 72.11, 70.85,
 86.15, 62.33, 77.77, 69.33, 63.79,
 75.02, 82.21, 82.76, 81.11, 75.91,
 94.90, 81.40, 75.88, 87.45, 75.81,
121.73,108.00,106.87,102.64,122.50,
 79.99, 68.42, 71.30, 66.62, 67.46,
101.26, 71.39, 86.97, 82.46, 94.2,
 78.38, 75.24, 75.02, 81.06, 60.11,
 78.51, 65.68, 65.04, 69.21, 55.28,
 70.13, 64.62, 57.50, 70.69, 66.53,
 91.52, 94.30,119.63, 84.34,115.89,
 96.66, 87.99, 89.47, 73.16, 75.57,
 60.53, 55.12, 50.42, 59.29, 54.74,
 76.83, 85.61, 75.31, 84.04, 89.69,
 87.25, 77.69, 80.87, 92.01, 78.74,
142.65,100.89,143.49, 97.04,139.18,
 66.37,104.26, 98.65, 80.40, 64.00,
 82.23, 98.70, 81.18, 82.06, 64.19,
    NA,    NA,    NA,    NA,    NA,
 90.22, 62.08, 71.07, 72.71, 70.27,
 72.67, 96.29, 82.36, 78.13, 69.28)
potato.m <- matrix(potato.v,23,5 ,byrow=TRUE)
po1.m <- potato.m[i1,]
po0.m <- potato.m[i0,]
po1.v <- as.vector(t(po1.m))
po0.v <- as.vector(t(po0.m))
mpo   <- rowMeans(potato.m)
mpo1  <- rowMeans(po1.m)
mpo0  <- rowMeans(po0.m)
mmpo0 <- mean(mpo0,na.rm=TRUE)
mmpo1 <- mean(mpo1,na.rm=TRUE)
mmpo  <- mean(mpo,na.rm=TRUE)
```

12.1.7 Funktionen

286 ⟨*start* 255⟩+≡

```
exp.ki.p<-function(){
  redo<-function(...){
    n<-slider(no=1); alpha<-slider(no=2); wd<-slider(no=3)
    p<-slider(no=4); seed<-slider(no=5)
    set.seed(seed); u<-o<-numeric(wd)
    t.value<-qt(1-alpha/2,n-1)
    for(i in 1:wd){
       p.dach<-mean(rbinom(n,1,p))
       sigma.dach<-sqrt(p.dach*(1-p.dach)/n)
       step<-t.value*sigma.dach
       u[i]<-p.dach-step; o[i]<-p.dach+step
    }
    plot(u,type="n",ylim=0:1,bty="n",xlab="Versuche",
         ylab="realisierte KIs",axes=F)
         axis(2)
    title(paste("n=",n,", alpha=",alpha,
          ", p=",p,", seed=",seed,sep=""))
    segments(1:wd,u,1:wd,o)
    abline(h=p)
  }
  slider(redo,c("n","alpha","wd","p","seed"),
         c(5,.001,5,.01,1),
         c(500,.3,100,.99,100),
         c(1,.001,1,.01,1),
         c(10,.05,10,.5,1))
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

287 ⟨*start* 255⟩+≡

```
demo.n.alpha.len<-function(p.dach){
redo<-function(...){
a<-slider(no=1);b<-slider(no=2)
n.set<-10:100
alpha.set<-seq(.2,.01,length=20)
sigma.dach<-t.value<-matrix(0,length(alpha.set),length(n.set))
for(j in seq(n.set)){
  t.value[,j]<-qt(1-alpha.set/2,n.set[j]-1)
  sigma.dach[,j]<-sqrt(p.dach*(1-p.dach)/n.set[j])
}
length.KI<-t(2*t.value*sigma.dach)
persp(n.set,1-alpha.set,length.KI,
      phi=a,theta=b,ticktype="detailed",
      xlab="n",ylab="1-alpha")
}
slider(redo,c("a","b"), c(-180,-180), c(360,360), c(5,5),c(10,-30))
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

12.1.8 Codechunks

288 ⟨*zeige Raucherdemo* 257⟩+≡
```
alpha<-0.05
abk<-c("Ia","Ib","Ic","II","III")
n<-c(21895,44184,25461,17366,33951)
p.dach<-c(32,86,56,106,66)/n
UG<-p.dach-qnorm(1-alpha/2) * sqrt(p.dach*(1-p.dach)/n)
OG<-p.dach+qnorm(1-alpha/2) * sqrt(p.dach*(1-p.dach)/n)
plot(1:5,ylim=c(0,max(1000*OG)),bty="n",type="n",
     axes=F,ylab="Sterberate in 1/1000",xlab="Gruppen")
title("KIs: Bronchialkrebs-WS und Rauchen")
axis(2); axis(1,at=1:5,labels=abk)
segments(1:5,1000*UG,1:5,1000*OG)
```

289 ⟨*zeichne Zusammenhang IL gegen n* 258⟩+≡
```
#old<-par(mfrow=1:2)
# 1. Plot:
p.dach<-0.5; n<-c(10,11,12,14,16,20,25,30,40,50,70,100)
n<-c(n,10*n,100*n)
alpha.set<-c(.01,.02,.05,.1)
plot(n,n,type="n",ylim=0:1,log="x",ylab="IL",bty="n")
for(i in seq(alpha.set)){
  alpha<-alpha.set[i]
  IL<-2*qnorm(1-alpha/2)*sqrt(p.dach*(1-p.dach)/n)
  lines(n,IL,lty=i)
}
title(paste("Intervalllaenge gegen n\np.dach=",p.dach))
legend(500,.8,lty=seq(alpha.set),
legend=paste(" (1-alpha) =",1-alpha.set),bty="n")
```

290 ⟨*zeichne Zusammenhang n gegen* $1-\alpha$ 259⟩+≡
```
p.dach<-0.5; alpha<-seq(.001,.2,length=100)
IL.set<-c(.01,.02,.04,.08,.13)
plot(1,type="n",xlim=c(.8,1),ylim=c(50,50000),
    log="y",xlab="1-alpha",ylab="n",bty="n")
for(i in seq(IL.set)){
  IL<-IL.set[i]
  n<-4*p.dach*(1-p.dach)*(qnorm(1-alpha/2)/IL)^2
  lines(1-alpha,n,lty=i)
}
title(paste("Stichprobenumfang gegen 1-alpha\np.dach=",p.dach))
legend(.95,150,lty=seq(IL.set),
legend=paste(" IL =",IL.set),bty="n")
```

12.1.9 Erstellung von `kif1.ps`

291 ⟨* 5⟩+≡
```
x<-seq(.1,40,length=200)
f.x<-dchisq(x,15)
alpha.u<-.1; alpha.o<-.1
x.u<-qchisq(alpha.u,15)
x.o<-qchisq(1-alpha.o,15)
plot(x,f.x,type="l",bty="n",ylab="",xlab="",axes=FALSE)
xu<-x[x<=x.u]
f.xu<-dchisq(xu,15)
lines(xu,f.xu,type="h")
xo<-x[x>=x.o]
f.xo<-dchisq(xo,15)
lines(xo,f.xo,type="h")
axis(1,at=c(x.u,x.o),labels=expression(x[u],x[o]))
text((x.u+x.o)/2,.01,cex=2,expression(1-alpha))
text((x.u)/3,.03,cex=2,expression(alpha[u]))
text((x.o+x.u),.03,cex=2,expression(alpha[o]))
arrows((x.u)/3,.025,(x.u)/1.6,0.015)
arrows((x.u+x.o),.025,(x.u+x.o)/1.2,0.015)
```

292 ⟨*do.power* 261⟩+≡
```
p <- seq(0,1,.01)
n <- 12
k <- 2
beta1 <- pbinom(k,n,p) + 1 - pbinom(n-k-1,n,p)
plot(p,beta1,type="l",ylab="beta(p)",ylim=c(0,1),bty="n")
#title("Powerfunktionen")
k <- 3
beta2 <- pbinom(k,n,p) + 1 - pbinom(n-k-1,n,p)
lines(p,beta2,lty=2)
k <- 4
beta3 <- pbinom(k,n,p) + 1 - pbinom(n-k-1,n,p)
lines(p,beta3,lty=3)
lines(c(0,0.075),c(0.05,0.05),lty=1)
text(0.125,0.05,"k=2")
lines(c(0,0.075),c(0.1,0.1),lty=2)
text(0.125,0.1,"k=3")
lines(c(0,0.075),c(0.15,0.15),lty=3)
text(0.125,0.15,"k=4")
```

293 ⟨*opfer.taeter.test* 262⟩+≡
```
xx <- matrix(c(627,254,5010,99),nc=2,byrow=TRUE)
chisq.test(xx)
```

294 ⟨*test.wartezeit* 263⟩+≡
```
w <- diff(coal[,1])
h <- hist(w,nclass=30,plot=FALSE)
br <- h$breaks
th <- pexp(br[-1],191/111)
pth <-c(th,1)-c(0,th)
co <- h$counts
print(chisq.test(c(co,0),p=pth))
```

295 ⟨*test.for.all* 264⟩+≡
```
erg <- NULL
for (i in 1:6){
e <- chisq.test(kugtab[,i])
erg <- rbind(erg,c(e$statistic,e$p.value,48))}
kz <- c("X-squared","p-value","df")
ks <- c("kug1","kug2","kug3","kug4","kug5","kug6")
dimnames(erg) <- list(ks,kz)
print(erg)
#print(format(erg[,1:2],digits=1))
```

296 ⟨*zeige.lotto,tabelle* 265⟩+≡
```
showlottotabelle(lottab)
```

297 ⟨*num.aus* 266⟩+≡
```
nt  <- sum(lottab)
nit <- rep(nt/49,49)
h   <- ((lottab-nit)^2)/nit
ch2 <- sum(h)
h1 <- qchisq(.95,48)
h2 <- pchisq(ch2,48)
h3 <- 1-h2
print(c(h1,h2,h3))
erg <- cbind(lottab,nit,h)
dimnames(erg)<- list(NULL,c("n_i","h_i","d_i^2"))
options(digits=4)
cat("\nArbeitstabelle\n")
print(erg)
cat("\nX^2\n")
print(ch2)
cat("\np-value\n")
print(h3)
options(digits=7)
```

298 ⟨*simple.box.and.whisker.ordered* 267⟩+≡

```
h.m <- rbind(po1.m,po0.m)
plot(c(po1.v,po0.v),ylab="weight",bty="n")
title("Potato: Ordered Data + box-and-whisker")
medw <- apply(h.m,1,median)
maxw <- apply(h.m,1,max)
minw <- apply(h.m,1,min)
ii   <- 3+(0:22)*5
dd   <- 1.
segments(ii,minw,ii,maxw,lty=3)
segments(ii-dd,medw,ii+dd,medw,lty=3)
segments(ii-dd,maxw,ii+dd,maxw,lty=3)
segments(ii-dd,minw,ii+dd,minw,lty=3)
abline(v=length(po1.v)+.5,lty=2)
```

299 ⟨*ordered.group.means* 268⟩+≡

```
plot(c(mpo1,mpo0),ylab="weight",bty="n",type="n")
title("Potato: Group Means of Ordered Data + C-lines")
abline(h=mmpo1,lty=2)
abline(h=mmpo0,lty=2)
abline(h=mmpo)
lmpo1 <- length(mpo1)
abline(v=lmpo1+.5,lty=3)
text(1:lmpo1,mpo1,"F")
text((lmpo1+1):(lmpo1+length(mpo0)),mpo0,"M")
text(3,112,"M = Mann")
text(3,116,"F = Frau")
```

300 ⟨*simple.binomial.test* 269⟩+≡

```
x0    <- mpo0[!is.na(mpo0)]
x1    <- mpo1[!is.na(mpo1)]
medp0 <- median(x0)
medp1 <- median(x1)
v1    <- x1 < medp0
v0    <- x0 < medp1
le0 <- rep(1,length(x0))
le1 <- rep(2,length(x1))
plot(c(x1,x0),c(le1,le0),bty="n",
xlab="mean(potato)",ylab="",ylim=c(0,3),pch=18,yaxt="n")
#title("Simple Test by Plotting and Counting")
mi0 <- min(x0)
mi1 <- min(x1)
ma0 <- max(x0)
ma1 <- max(x1)
y0  <- 1.2
y00 <-  .9
y1  <- 1.8
y11 <- 2.1
segments(c(mi0,mi1),c(y0,y1),c(ma0,ma1),c(y0,y1))
points(medp0,y0,pch=3)
points(medp1,y1,pch=3)
lines(c(medp0,medp0),c(y0,y11),lty=3)
lines(c(medp1,medp1),c(y00,y1),lty=3)
text(medp1,y00,paste(sum(v0),sum(!v0)))
text(medp0,y11,paste(sum(v1),sum(!v1)))
text(mi0-5,y0,"M")
text(ma1+5,y1,"F")
```

301 ⟨*simple.confidence.test* 270⟩+≡

```
## data and hypothesis cleared for NA
d1 <- mpo0[!is.na(mpo0)]
h1 <- median(mpo1,na.rm=TRUE)
d2 <- mpo1[!is.na(mpo1)]
h2 <- median(mpo0,na.rm=TRUE)
## doing the test
par(mfrow=1:2)
conf.test1(d1,h1,txt="Median Gr F vs Gr M")
conf.test1(d2,h2,txt="Median Gr M vs Gr F")
par(mfrow=c(1,1))
```

302 ⟨*binomialtabelle* 271⟩+≡

```
#Code zur Erzeugung der Binomialtabelle:
for(n in c(3,4,5,6,8,10,12,14,16,20,30)){
 cat("n=",n,"\\\\\n")
 for(j in 0:n){
  cat("&",j,"&")
  cat(paste(signif(pbinom(j,n,(1:5)/10),4),collapse="&"))
  cat("\\\\[0.9ex]\n")
 }
}
```

303 ⟨*jeder.mit.jedem* 272⟩+≡

```
par(mfrow=c(3,3))
multiempver(mpo0,txt="Gr M")
multiempver(mpo0,mpo1,txt="Gr M, Gr F")
multiempver(mpo0,mpo,txt="Gr M, All")
multiempver(mpo0,mpo1,mpo,txt="Gr M,Gr F,All")
multiempver(mpo1,txt="Gr F")
multiempver(mpo1,mpo,txt="Gr F, All")
multiempver(mpo0,mpo1,mpo,txt="Gr M,Gr F,All")
multiempver(mpo0,mpo1,mpo,txt="Gr M,Gr F,All")
multiempver(mpo,txt="All")
par(mfrow=c(1,1))
```

12.1.10 `plot.chi`

304 ⟨*start* 255⟩+≡

```
plot.chi <- function(){
refresh.code <- function(...){
par(mfrow=c(1,2))
nu     <- n1 <- slider(no=1)
x      <- seq(0.01,149.99,.05)
ychi   <- y  <- dchisq(x,n1)
yychi  <- yy <- pchisq(x,n1)
yg     <- slider(obj.name="yg")
yyg    <- slider(obj.name="yyg")
ymax   <- max(c(max(y),max(yg)))
plot(x,y,type="l",ylim=c(0,ymax),ylab="",bty="n")
title("Chi: Density")
lines(x,yg,lty=2)
plot(x,yy,type="l",ylim=c(0,1),ylab="",bty="n")
title("Chi: CDF")
lines(x,yyg,lty=2)
par(mfrow=c(1,1)) }

reset.func <- function(...){
par(mfrow=c(1,2))
x   <- seq(0.01,149.99,.05)
nuu  <- 10
yg  <- dchisq(x,nuu)
yyg <- pchisq(x,nuu)
plot(x,yg,type="l",ylab="",bty="n")
title("Chisq: Density")
plot(x,yyg,type="l",ylim=c(0,1),ylab="",bty="n")
title("Chisq: CDF")
slider(obj.name="yg",obj.value=yg)
slider(obj.name="yyg",obj.value=yyg)
par(mfrow=c(1,1)) }
nu<-10
slider(refresh.code,sl.names="nu",1,50,1,
nu,reset.function=reset.func,
title="Chi-Verteilung")

reset.func()
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
 }
```

305 ⟨*start* 255⟩+≡

```
exp.binomialtest<-function(){
if(!exists("slider")) source("http://www.wiwi.uni-bielefeld.de/~wolf/software/R-wtools/s
slider(obj.name="alternative",obj.value="two.sided")
redo<-function(...){
n<-slider(no=1); p<-slider(no=2); alpha<-slider(no=3)
alternative<-slider(obj.name="alternative")
H<-paste( "H: p=",p)
if(alternative=="less") H<-paste( "H: p>=",p)
if(alternative=="greater") H<-paste( "H: p<=",p)
x<-0:n
if(alternative=="two.sided") alpha.h<-alpha/2 else alpha.h<-alpha
F.x<-pbinom(x,n,p)
plot(x,F.x,bty="n",ylim=0:1)
segments(x[-(n+1)],F.x[-(n+1)],x[-1],F.x[-(n+1)])
if(alternative!="greater"){
 abline(h=alpha.h)#% "red"
 k.stern.ind<-sum(F.x<=alpha.h)
 k.stern1<-x[k.stern.ind]
 segments(k.stern1,0,k.stern1, alpha.h)#% "blue"
 segments(k.stern1,alpha.h,k.stern1,
    if(alternative=="less") 1 else 1-alpha.h,lty=2)#% "blue"
 segments(0,.5,k.stern1,.5)#% "blue"
 text(k.stern1/2,.53,paste("gegen",H))#% "blue"
}
if(alternative!="less"){
 abline(h=1-alpha.h)#% "red"
 k.stern.ind<-length(x)+1-sum(F.x>=(1-alpha.h))
 k.stern2<-x[k.stern.ind]
 segments(k.stern2,1,k.stern2,1-alpha.h)#% "blue"
 segments(k.stern2,
    if(alternative=="greater") 0 else alpha.h,
    k.stern2, 1-alpha.h,lty=2)#% "blue"
 segments(n,.5,k.stern2,.5)#% "blue"
 text((k.stern2+n)/2,.53,paste("gegen",H))#% "blue"
}
 title(paste("n=",n,", p=",p,", alpha=",alpha,
   if(alternative=="less")
     paste("\nK= { 0,...,",k.stern1,"}"),
   if(alternative=="greater")
     paste("\nK= {",k.stern2,",...,",n,"}"),
   if(alternative=="two.sided")
     paste("\nK={0,...,",k.stern1,",",k.stern2,",...,",n,"}")
  ))
}
fbut.two.sided<-function(...){
   slider(obj.name="alternative",obj.value="two.sided")
  redo()
}
fbut.less<-function(...){
   slider(obj.name="alternative",obj.value="less")
  redo()
}
fbut.greater<-function(...){
   slider(obj.name="alternative",obj.value="greater")
  redo()
}
slider(redo,c("n","p.0","alpha"),
```

```
  c(2,.01,.01),  c(100,.99,.3),  c(1,.01,.01),  c(28,.5,.05),
  c(fbut.two.sided,fbut.less,fbut.greater),
  c("G: p!=p.0","G: p<p.0","G: p>p.0"),title="Binomialtest"
)
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

306 ⟨*start* 255⟩+≡

```
exp.nv.guete<-function(){
if(!exists("slider")) source("http://www.wiwi.uni-bielefeld.de/~wolf/software/R-wtools/s
slider(obj.name="alternative",obj.value="two.sided")
redo<-function(...){
n<-slider(no=1); mu0<-slider(no=2); mu1<-slider(no=3)
alpha<-slider(no=4) # ; beta<-slider(no=5)
agr<-mu1>mu0 # agr==1  <-> greater
alternative<-slider(obj.name="alternative")
delta<-4; sdx<-1/sqrt(n)
x<-seq(min(mu0,mu1)-delta,max(mu0,mu1)+delta,length=200)
fH.x<-dnorm(x,mu0,sdx)
fG.x<-dnorm(x,mu1,sdx)
par(mfrow=2:1)
if(alternative!="two.sided"){
 agr<-alternative=="greater"
 k.stern<-qnorm(if(agr) 1-alpha else alpha,mu0,sdx)
 beta<-if(agr) pnorm(k.stern,mu1,sdx) else 1-pnorm(k.stern,mu1,sdx)
 beta.all<-if(agr) pnorm(k.stern,x,sdx) else 1-pnorm(k.stern,x,sdx)
 plot(x,fH.x,bty="n",type="l",ylim=c(0,max(fH.x)*1.2),
      ylab="Verteilung PG unter H bzw. G",xlab="Parameter")
 title(paste("H: mean =",mu0,
             if(agr)"/  G: mean >" else ",  G: mean <",mu0))
 lines(x,fG.x,col="red")
 ind<-if(agr) x>=k.stern else x<=k.stern
 lines(x[ind],fH.x[ind],type="h",col="black")
 lines(x[!ind],fG.x[!ind],type="h",col="red")
 usr<-par()$usr
 segments(mu1,0,mu1,0.9*usr[4],lty=3,col="red")
 text(mu1,usr[4]*.96,paste("true mean:",mu1),col="red")
 plot(x,x,type="n",bty="n",ylim=c(0,1.2),
    ylab="WS: Beobachtung in K",xlab="Parameter")
 title("power")
segments(mu0,0,mu0,alpha)
 arrows(mu0+1.5,alpha/2,mu0,alpha/2); text(mu0+2,alpha/2+.03,"alpha")
 arrows(mu1+1.5,1-beta/2,mu1,1-beta/2,col="red")
 text(mu1+2,1-beta/2-.03,"beta",col="red")
 segments(mu1,1-beta,mu1,1,col="red")
 lines(x,1-beta.all,lty=1,col="red")
 usr<-par()$usr
 text(mu1,usr[4]*.96,paste("true mean:",mu1),col="red")
} else {
 alpha<-alpha/2
 k.stern<-qnorm(c(1-alpha, alpha),mu0,sdx)
 beta<-pnorm(k.stern[1],mu1,sdx)-pnorm(k.stern[2],mu1,sdx)
 beta.all<-pnorm(k.stern[1],x,sdx)-pnorm(k.stern[2],x,sdx)
 plot(x,fH.x,bty="n",type="l",ylim=c(0,max(fH.x)*1.2),
      ylab="Verteilung PG unter H bzw. G",xlab="Parameter")
 title(paste("H: mean =",mu0,"/  G: mean not equal",mu0))
 lines(x,fG.x,col="red")
 ind<- x>=k.stern[1] | x<=k.stern[2]
 lines(x[ind],fH.x[ind],type="h",col="black")
 lines(x[!ind],fG.x[!ind],type="h",col="red")
 usr<-par()$usr
 segments(mu1,0,mu1,0.9*usr[4],lty=3,col="red")
 text(mu1,usr[4]*.96,paste("true mean",mu1),col="red")
 plot(x,x,type="n",bty="n",ylim=c(0,1.2),
    ylab="WS: Beobachtung in K",xlab="Parameter")
```

```
 title("power")
 segments(mu0,0,mu0,alpha*2)
 arrows(mu0+1.5,alpha,mu0,alpha); text(mu0+2,alpha+.03,"alpha")
 arrows(mu1+1.5,1-beta/2,mu1,1-beta/2,col="red")
 text(mu1+2,1-beta/2-.03,"beta",col="red")
 segments(mu1,1-beta,mu1,1,col="red")
 lines(x,1-beta.all,lty=1,col="red")
 usr<-par()$usr
 text(mu1,usr[4]*.96,paste("true mean",mu1),col="red")
}
par(mfrow=c(1,1))
}
fbut.two.sided<-function(...){
   slider(obj.name="alternative",obj.value="two.sided")
  redo()
}
fbut.less<-function(...){
   slider(obj.name="alternative",obj.value="less")
  redo()
}
fbut.greater<-function(...){
   slider(obj.name="alternative",obj.value="greater")
  redo()
}
slider(redo,c("n","H: mean=mu.0","True mean","alpha"),
  c(2,-5,-5,.01),c(100,5,5,.3),c(1,.1,.1,.01),c(6,1,1.5,.1),
  c(fbut.two.sided,fbut.less,fbut.greater),
  c("G: mean <> mu.0","G: mean < mu.0","G: mean > mu.0"),
title="Guete")
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

307 ⟨*start* 255⟩+≡

```
conf.test1 <- function(data,hypo,txt="Test per Konfidenzintervalle"){
# no NA expected
data <- sort(data)
n  <- length(data)
k  <- floor(n/2)
ju <- 1:k
jo <- n+1-ju
ma <- max(data)
mi <- min(data)
plot(c(mi,ma),c(1,k),type="n",xlab="data",ylab="k",bty="n",las=1,crt=90)
title(txt)
segments(data[ju],ju,data[jo],ju)
abline(v=hypo,lty=2)
invisible() }
```

308 ⟨*start* 255⟩+≡

```
smirkol <- function(dat1,dat2,PLOT=TRUE,PRINT=FALSE){
h  <- CEV.gr(dat1)
x  <- h[,1]
px <- h[,2]
h  <- CEV.gr(dat2)
y  <- h[,1]
py <- h[,2]
nx <- length(x)
ny <- length(y)
n  <- nx+ny
xy <- c(x,y)
gr <- c(rep(1,nx),rep(2,ny))
gr <- gr[order(xy)]
xy <- sort(xy)
lx <- ly <- 0
ix <- iy <- 1
dd <- 0
di <- id <- c(0,0,0)

GO <- TRUE
if(nx==ny&&sum(x==y)==nx&&sum(px==py)==nx) GO <- FALSE

if(GO) {

for (k in (1:n)) {
  if(gr[k]==1) {
    lx <- px[ix]
    wo <- x[ix]
    di <- c(gr[k],ix,iy)
    ix <- ix+1
  }else{
    ly <- py[iy]
    wo <- y[iy]
    di <- c(gr[k],ix,iy)
    iy <- iy+1 }
  dn <- lx-ly
  if(abs(dn) > dd) {
    dd  <- abs(dn)
    pk  <- c(wo,lx,ly)
    id  <- di }
}
erg <- c(dd,id,pk)

}else{
cat("\nVerteilungen gleich\n")
erg <- NA }

if(PLOT) {
  multiempver(dat1,dat2)
  if(GO) lines(c(erg[5],erg[5]),c(erg[6],erg[7]),col="red",lty=3)}

return(erg)
}
```

309 ⟨*start* 255⟩+≡

```
multiempver <- function(d1,d2,d3,txt=""){
if(missing(d3)) d3 <- d1
if(missing(d2)) d2 <- d1
h1   <- CEV.gr(d1)
h2   <- CEV.gr(d2)
h3   <- CEV.gr(d3)
d1x  <- h1[,1]
d2x  <- h2[,1]
d3x  <- h3[,1]
y1   <- h1[,2]
y2   <- h2[,2]
y3   <- h3[,2]
r1  <- range(d1x)
s1  <- (r1[2]-r1[1])/20
r2  <- range(d2x)
s2  <- (r2[2]-r2[1])/20
r3  <- range(d3x)
s3  <- (r3[2]-r3[1])/20
xma <- max(r1[2],r2[2],r3[2])+5*max(s1,s2,s3)
xmi <- min(r1[1],r2[1],r3[1])-5*max(s1,s2,s3)
plot(c(xma,xmi),c(0,1),type="n",ylab="F",xlab="x",bty="n")
title(paste("Emp. Verteil.:",txt))
segments(c(xmi,d1x),c(0,y1),c(d1x,max(d1x)+xma),c(0,y1),lty=1)
segments(c(xmi,d2x),c(0,y2),c(d2x,max(d2x)+xma),c(0,y2),lty=2)
segments(c(xmi,d3x),c(0,y3),c(d3x,max(d3x)+xma),c(0,y3),lty=3)
#points(d1x,y1,pch=18)
#points(d2x,y2,pch=18)
#points(d3x,y3,pch=18)
invisible() }
```

310 ⟨*start* 255⟩+≡

```
CEV.gr <- function(dat) {
# clear for NA
d1  <- dat[!is.na(dat)]
d1x <- sort(d1)
n1  <- length(d1x)
# clear for multiple values
d2  <- c(d1x[-1],2*d1x[n1])
jj  <- d1x!=d2
y   <- ((1:n1)/n1)[jj]
x   <- d1x[jj]
erg <- cbind(x,y)
return(erg) }
```

311 ⟨*start* 255⟩+≡
```
showlottotabelle <- function(tab){
print(tab[1:10])
print(tab[11:20])
print(tab[21:30])
print(tab[31:40])
print(tab[41:49])}
```

12.1.11 Beispiel Lotto

Verteilung Ziffern

312 ⟨*start* 255⟩+≡
```
"lottab" <-
structure(as.integer(c(307, 322, 321, 300, 305, 322, 305, 285,
314, 299, 309, 302, 250, 292, 300, 294, 311, 315, 311, 298, 322,
307, 290, 297, 316, 322, 319, 278, 303, 295, 316, 359, 320, 279,
310, 321, 311, 345, 315, 308, 310, 320, 311, 308, 277, 304, 299,
326, 346)), .Dim = as.integer(49), .Dimnames = structure(list(
    c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
    "11", "12", "13", "14", "15", "16", "17", "18", "19", "20",
    "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
    "31", "32", "33", "34", "35", "36", "37", "38", "39", "40",
    "41", "42", "43", "44", "45", "46", "47", "48", "49")),
.Names = ""), class = "table")
```

Verteilung Lotto alle Ziehungen

313 ⟨*start* 255⟩+≡
```
"kugtab" <-
structure(as.integer(c(52, 53, 57, 45, 49, 45, 39, 49, 53, 46,
53, 51, 47, 44, 51, 49, 61, 54, 56, 53, 51, 48, 52, 57, 42, 59,
57, 40, 42, 52, 58, 58, 55, 42, 51, 57, 61, 61, 48, 57, 55, 54,
46, 44, 54, 59, 57, 41, 51, 54, 59, 48, 55, 48, 41, 60, 53, 49,
56, 67, 49, 49, 41, 46, 57, 53, 46, 40, 47, 45, 50, 44, 45, 47,
49, 49, 44, 48, 36, 55, 79, 70, 45, 46, 49, 49, 55, 61, 47, 50,
60, 44, 56, 42, 50, 58, 63, 62, 53, 51, 53, 57, 46, 70, 50, 43,
44, 48, 36, 57, 42, 48, 48, 46, 54, 55, 58, 55, 53, 46, 42, 46,
44, 53, 52, 51, 53, 53, 60, 57, 64, 50, 54, 54, 36, 64, 58, 48,
48, 63, 54, 59, 30, 47, 47, 59, 57, 47, 57, 68, 44, 52, 44, 61,
47, 44, 48, 46, 50, 39, 46, 33, 50, 55, 49, 60, 44, 56, 56, 43,
53, 60, 50, 59, 53, 51, 51, 50, 59, 49, 54, 48, 55, 49, 55, 50,
61, 46, 39, 62, 63, 54, 52, 36, 46, 72, 56, 48, 45, 45, 51, 64,
43, 48, 58, 61, 48, 43, 28, 55, 61, 40, 47, 50, 47, 53, 64, 56,
56, 49, 60, 58, 52, 55, 57, 51, 45, 54, 43, 43, 63, 53, 52, 45,
45, 48, 68, 54, 60, 47, 44, 41, 55, 59, 48, 45, 54, 50, 54, 59,
58, 52, 45, 66, 40, 59, 52, 45, 58, 61, 52, 41, 61, 50, 46, 53,
51, 53, 47, 63, 53, 50, 35, 52, 52, 48, 52, 39, 45, 48, 53, 64,
65, 53, 47, 43, 50, 45, 39, 53, 55, 46, 58, 56)),
.Dim = as.integer(c(49, 6)),
 .Dimnames = list(c("1", "2", "3", "4", "5", "6", "7", "8",
"9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19",
"20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
"31", "32", "33", "34", "35", "36", "37", "38", "39", "40", "41",
"42", "43", "44", "45", "46", "47", "48", "49"), c("kug1",
"kug2", "kug3", "kug4", "kug5", "kug6")))
```

12.1.12 Beispiel Kohlengruben

314 ⟨*start* 255⟩+≡

```
"coal"<-structure(list(date = c(1851.2, 1851.63, 1851.97, 1851.97, 1852.31,
1852.35, 1852.36, 1852.39, 1852.98, 1853.2, 1853.23, 1853.32,
1853.5, 1854.13, 1856.4, 1856.51, 1856.54, 1856.62, 1857.14,
1857.4, 1857.58, 1858.09, 1858.15, 1858.41, 1858.95, 1860.13,
1860.17, 1860.59, 1860.85, 1860.92, 1860.97, 1861.18, 1861.74,
1861.84, 1862.14, 1862.89, 1862.94, 1863.18, 1863.79, 1863.94,
1863.99, 1865.46, 1865.97, 1866.06, 1866.34, 1866.45, 1866.83,
1866.95, 1866.95, 1867.64, 1867.85, 1867.86, 1868.75, 1868.9,
1868.99, 1869.25, 1869.44, 1869.55, 1869.81, 1869.87, 1870.12,
1870.52, 1870.56, 1870.63, 1871.03, 1871.15, 1871.17, 1871.74,
1871.82, 1872.12, 1872.24, 1872.77, 1873.14, 1874.29, 1874.55,
1874.89, 1874.98, 1875.33, 1875.93, 1875.93, 1875.93, 1876.97,
1877.06, 1877.11, 1877.19, 1877.78, 1877.81, 1878.18, 1878.2,
1878.24, 1878.43, 1878.7, 1879.04, 1879.17, 1879.5, 1880.06,
1880.54, 1880.69, 1880.94, 1881.11, 1881.97, 1882.13, 1882.3,
1882.3, 1882.33, 1882.85, 1883.8, 1883.85, 1884.07, 1884.86,
1885.17, 1885.46, 1885.98, 1886.62, 1886.69, 1886.75, 1886.92,
1887.13, 1887.41, 1888.3, 1889.05, 1889.2, 1889.79, 1890.1, 1890.19,
1891.25, 1891.67, 1892.65, 1893.51, 1894.48, 1895.32, 1896.07,
1896.28, 1896.33, 1899.63, 1901.39, 1902.67, 1905.06, 1905.19,
1905.52, 1906.77, 1908.14, 1908.27, 1908.63, 1909.13, 1909.83,
1910.36, 1910.97, 1912.52, 1913.78, 1914.41, 1916.62, 1918.03,
1922.53, 1922.68, 1923.57, 1927.16, 1928.11, 1930.15, 1930.75,
1931.08, 1931.83, 1931.88, 1932.07, 1932.86, 1932.88, 1933.88,
1934.72, 1935.64, 1935.7, 1936.6, 1937.5, 1938.35, 1939.82, 1940.22,
1940.42, 1941.42, 1941.52, 1941.57, 1942, 1942.13, 1942.48, 1946.95,
1947.02, 1947.62, 1947.64, 1947.69, 1951.41, 1957.88, 1960.49,
1962.22)), .Names = "date", row.names = c("1", "2", "3", "4",
"5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15",
"16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26",
"27", "28", "29", "30", "31", "32", "33", "34", "35", "36", "37",
"38", "39", "40", "41", "42", "43", "44", "45", "46", "47", "48",
"49", "50", "51", "52", "53", "54", "55", "56", "57", "58", "59",
"60", "61", "62", "63", "64", "65", "66", "67", "68", "69", "70",
"71", "72", "73", "74", "75", "76", "77", "78", "79", "80", "81",
"82", "83", "84", "85", "86", "87", "88", "89", "90", "91", "92",
"93", "94", "95", "96", "97", "98", "99", "100", "101", "102",
"103", "104", "105", "106", "107", "108", "109", "110", "111",
"112", "113", "114", "115", "116", "117", "118", "119", "120",
"121", "122", "123", "124", "125", "126", "127", "128", "129",
"130", "131", "132", "133", "134", "135", "136", "137", "138",
"139", "140", "141", "142", "143", "144", "145", "146", "147",
"148", "149", "150", "151", "152", "153", "154", "155", "156",
"157", "158", "159", "160", "161", "162", "163", "164", "165",
"166", "167", "168", "169", "170", "171", "172", "173", "174",
"175", "176", "177", "178", "179", "180", "181", "182", "183",
"184", "185", "186", "187", "188", "189", "190", "191"), class = "data.frame")
```

12.1.13 Daten: Zwiebelbeispiel

Gruppenaufteilung

315 ⟨*start* 255⟩+≡
```
gender <- c(1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1,
1, 1, 0, 0, 1, 0, 0, 0)
ind <- length(gender)
i1  <- (1:ind)[!!gender]
i0  <- (1:ind)[!gender]
```

Kartoffeln

316 ⟨*start* 255⟩+≡
```
potato.v <- c(77.33, 83.93, 71.61, 53.76, 62.37,
 83.56, 82.25, 76.96, 96.34,106.62,
113.71, 84.83, 70.15, 72.11, 70.85,
 86.15, 62.33, 77.77, 69.33, 63.79,
 75.02, 82.21, 82.76, 81.11, 75.91,
 94.90, 81.40, 75.88, 87.45, 75.81,
121.73,108.00,106.87,102.64,122.50,
 79.99, 68.42, 71.30, 66.62, 67.46,
101.26, 71.39, 86.97, 82.46, 94.2,
 78.38, 75.24, 75.02, 81.06, 60.11,
 78.51, 65.68, 65.04, 69.21, 55.28,
 70.13, 64.62, 57.50, 70.69, 66.53,
 91.52, 94.30,119.63, 84.34,115.89,
 96.66, 87.99, 89.47, 73.16, 75.57,
 60.53, 55.12, 50.42, 59.29, 54.74,
 76.83, 85.61, 75.31, 84.04, 89.69,
 87.25, 77.69, 80.87, 92.01, 78.74,
142.65,100.89,143.49, 97.04,139.18,
 66.37,104.26, 98.65, 80.40, 64.00,
 82.23, 98.70, 81.18, 82.06, 64.19,
    NA,    NA,    NA,    NA,    NA,
 90.22, 62.08, 71.07, 72.71, 70.27,
 72.67, 96.29, 82.36, 78.13, 69.28)
potato.m <- matrix(potato.v,23,5 ,byrow=TRUE)
po1.m <- potato.m[i1,]
po0.m <- potato.m[i0,]
po1.v <- as.vector(t(po1.m))
po0.v <- as.vector(t(po0.m))
mpo   <- rowMeans(potato.m)
mpo1  <- rowMeans(po1.m)
mpo0  <- rowMeans(po0.m)
mmpo0 <- mean(mpo0,na.rm=TRUE)
mmpo1 <- mean(mpo1,na.rm=TRUE)
mmpo  <- mean(mpo,na.rm=TRUE)
```

317 ⟨*start* 255⟩+≡
⟨*stelle datensaetze bereit* 318⟩
⟨*andere fkt* 319⟩
⟨*lotto* 320⟩
```
library(boot)
```

318 ⟨*stelle datensaetze bereit* 318⟩≡

```
x.buch<-c(150 ,60 ,10 ,70 ,100 ,100 ,40 ,40 ,800 ,100
        ,60 ,40 ,70 ,200 ,5 ,60 ,300 ,80 ,20 ,10)
alter<-
c(23, 21, 22, 19, 20, 21, 21, 22, 20, 20, 22, 21, 20, 20, 19, 26, 21, 20, 25,
        26, 22, 19, 21, 20, 20, 19, 23, 20, 21, 22, 20, 21, 18, 21, 20, 24, 24,
        19, 23, 24, 20, 20, 20, 21, 19, 20, 23, 20, 20, 21, 20, 20, 24, 19, 21,
        20, 28, 24, 20, 20, 23, 21, 20, 21, 19, 21, 21, 20, 23, 20, 22, 21, 23,
        19, 20, 23, 21, 21, 21, 20, 21, 23, 20, 22, 21, 28, 21, 22, 23, 22, 22,
        20, 22, 21, 19, 19, 19, 20, 20, 21, 24, 19, 22, 20, 23, 20, 21, 22, 23,
        20, 23, 20, 18, 21, 21, 24, 23, 21, 21, 20, 20, 24, 19, 23, 22, 21, 20,
        24, 21, 19, 21, 20, 23, 20, 20, 20, 22, 20, 20, 20, 20, 21, 20, 21, 21,
        20, 20, 22, 23, 19, 20, 20, 19, 23, 27, 21, 21, 24, 27, 20, 21, 21, 20,
        19, 19, 19, 21, 19, 22, 19, 20, 24, 21, 20, 23, 21, 21, 27, 20, 18, 19,
        20, 24, 20, 29, 26, 25, 22, 24, 26, 30, 20, 20, 23, 21, 20, 22, 22, 21,
        25, 22, 20, 21, 22, 20, 19, 19, 22, 23, 20, 19, 19, 20, 20, 19, 22, 20,
        27, 27, 20, 24, 21, 20, 21, 20, 24, 22, 23, 23, 20, 20, 21, 21, 21, 20,
        22, 19, 19, 19, 23, 20, 23, 21, 23, 21, 20, 20, 19, 21, 24, 20, 20, 20,
        20, 21, 20, 20, 20, 21, 19, 22, 21, 20, 20, 22)
"umsatz" <-
structure(list(mitarb = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11,
12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27,
28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43,
44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59,
60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75,
76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91,
92, 93, 94, 95, 96, 97, 98, 99, 100, 80, 73, 30, 47, 3, 14, 22,
76, 86, 66), ums = c(2164, 172, 2458, 2611, 542, 983, 1147, 1133,
3475, 1750, 3069, 1567, 2975, 3206, 4625, 4089, 1275, 1350, 1636,
2611, 3575, 3894, 3003, 6200, 2153, 3972, 5825, 2100, 4175, 5917,
7081, 7244, 2842, 5950, 6569, 8300, 6419, 6228, 2925, 9000, 3986,
5617, 5136, 6278, 6875, 9028, 7181, 9333, 5308, 5972, 7792, 7367,
7742, 6450, 9181, 11178, 9975, 8283, 10358, 10500, 7286, 8028,
6725, 8933, 9764, 12083, 10236, 9367, 5175, 10694, 10481, 11000,
7908, 12128, 8458, 11922, 8631, 7583, 10558, 10444, 12575, 12528,
10836, 10033, 8375, 9317, 9492, 14422, 10653, 8750, 12892, 12011,
10075, 13228, 10292, 16667, 7275, 10439, 11825, 10611, 16103,
15132, 8612, 13026, 6911, 9341, 8247, 17035, 18736, 14565)), .Names = c("mitarb",
"ums"), row.names = c("1", "2", "3", "4", "5", "6", "7", "8",
"9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19",
"20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
"31", "32", "33", "34", "35", "36", "37", "38", "39", "40", "41",
"42", "43", "44", "45", "46", "47", "48", "49", "50", "51", "52",
"53", "54", "55", "56", "57", "58", "59", "60", "61", "62", "63",
"64", "65", "66", "67", "68", "69", "70", "71", "72", "73", "74",
"75", "76", "77", "78", "79", "80", "81", "82", "83", "84", "85",
"86", "87", "88", "89", "90", "91", "92", "93", "94", "95", "96",
"97", "98", "99", "100", "101", "102", "103", "104", "105", "106",
"107", "108", "109", "110"), class = "data.frame")

pakete<-matrix(c(927,746,600,448,152,94,55,
                 1249,1355,1097,897,398,196,125,
                 422,542,544,419,202,163,95)
        ,ncol=7,nrow=3,byrow=T)
dimnames(pakete)<-list(paste("P",1:3,sep=""),
        paste("R",c(3,5,10,15,20,50,100),sep=""))
buecher.stud<-
c(80, 180, 200, 300, 0, 30, 150, 80, 300, 150, 20, 10, 60, 30, 40, 30, 150, 150,
```

```
        400, 0, 200, 0, 800, 20, 100, 100, 50, 5, 400, 200, 40, 150, 30, 30, 0,
        100, 30, 80, 25, 100, 30, 12, 80, 10, 30, 100, 21, 100, 100, 30, 500,
        40, 50, 63, 70, 30, 70, 50, 40, 45, 15, 350, 220, 50, 300, 25, 120, 100,
        60, 60, 40, 40, 0, 300, 50, 500, 200, 100, 25, 20, 200, 300, 50, 30, 0,
        40, 50, 180, 25, 200, 20, 20, 120, 3000, 20, 400, 50, 50, 30, 50, 100,
        152, 130, 328, 100, 50, 30, 120, 20, 40, 50, 30, 20, 40, 0, 100, 80, 50,
        20, 100, 20, 200, 30, 150, 10, 100, 50, 100, 20, 10, 1500, 40, 100, 50,
        250, 60, 10, 0, 0, 70, 80, 60, 120, 150, 300, 0, 300, 20, 100, 20, 60,
        100, 30, 60, 80, 75, 600, 200, 20, 300, 200, 50, 30, 100, 60, 20, 50,
        50, 70, 50, 400, 120, 100, 200, 2, 50, 50, 200, 50, 30, 50, 100, 50, 6,
        25, 20, 100, 15, 5, 30, 0, 100, 50, 100, 80)
cd<-cd.stud<-
c(100, 40, 86, 54, 80, 3, 300, 12, 50, 80, 40, 20, 30, 40, 40, 50, 30, 30, 20,
        180, 25, 190, 30, 32, 50, 300, 160, 40, 80, 50, 350, 20, 25, 10, 30, 50,
        40, 150, 40, 50, 15, 3, 30, 70, 5, 20, 50, 100, 20, 10, 120, 35, 15,
        233, 450, 220, 20, 30, 35, 15, 20, 200, 250, 10, 90, 15, 160, 40, 30,
        500, 15, 15, 91, 300, 50, 150, 25, 50, 50, 30, 30, 130, 160, 0, 6, 15,
        40, 40, 20, 120, 50, 100, 68, 100, 20, 100, 25, 50, 40, 30, 60, 100,
        152, 40, 25, 150, 40, 20, 0, 40, 30, 25, 0, 35, 0, 40, 0, 150, 50, 200,
        50, 120, 80, 300, 10, 20, 200, 10, 30, 70, 60, 15, 20, 150, 90, 40, 40,
        0, 50, 10, 60, 40, 180, 100, 63, 0, 50, 30, 20, 35, 30, 50, 10, 100, 30,
        100, 55, 25, 4, 40, 30, 150, 20, 20, 25, 50, 280, 8, 20, 100, 100, 30,
        0, 60, 81, 50, 20, 20, 20, 10, 100, 30, 10, 12, 30, 60, 30, 20, 20, 0,
        1, 1, 100, 100, 60)
groesse.stud<-
structure(.Data = c(171, 173, 176, 170, 168, 175, 198, 170, 177, 198, 170, 173,
        201, 168, 205, 176, 184, 183, 184, 180, 190, 180, 172, 182, 176, 168,
        176, 180, 184, 190, 188, 177, 172, 160, 186, 184, 168, 168, 183, 180,
        166, 180, 171, 170, 175, 175, 180, 180, 191, 176, 165, 184, 186, 180,
        164, 180, 182, 178, 183, 178, 178, 188, 170, 179, 180, 182, 165, 182,
        197, 193, 181, 178, 180, 183, 175, 179, 182, 188, 183, 173, 175, 176,
        182, 178, 171, 173, 190, 180, 178, 180, 187, 170, 186, 181, 165, 190,
        174, 177, 180, 192, 175, 169, 187, 165, 193, 181, 180, 182, 163, 169,
        183, 182, 170, 175, 184, 159, 178, 176, 180, 190, 175, 180, 170, 173,
        185, 183, 187, 186, 168, 170, 175, 181, 180, 182, 180, 175, 186, 170,
        185, 168, 185, 191, 192, 160, 182, 185, 161, 185, 182, 170, 170, 180,
        160, 173, 164, 170, 187, 190, 180, 192, 187, 186, 168, 167, 163, 175,
        194, 164, 170, 177, 190, 185, 176, 175, 167, 179, 195, 180, 180, 166,
        174, 180, 190, 160, 186, 184, 182, 185, 180, 180, 181, 182, 172, 166,
        185, 178, 183, 183, 180, 177, 191, 196, 187, 183, 196, 170, 164, 196,
        167, 192, 178, 176, 169, 182, 196, 170, 180, 169, 184, 182, 185, 183,
        177, 185, 179, 175, 176, 180, 183, 188, 188, 175, 172, 180, 170, 185,
        172, 165, 183, 175, 185, 160, 188, 186, 171, 178, 182, 185, 188, 194,
        180, 165, 161, 180, 185, 188, 178, 176, 178, 175, 192, 171, 190, 171,
        169), .Names = c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
        "11", "12", "13", "14", "15", "16", "17", "18", "19", "20", "21", "22",
        "23", "24", "25", "26", "27", "28", "29", "30", "31", "32", "33", "34",
        "35", "36", "37", "38", "39", "40", "41", "42", "43", "44", "45", "46",
        "47", "48", "49", "50", "51", "52", "53", "54", "55", "56", "57", "58",
        "59", "60", "61", "62", "63", "64", "65", "66", "67", "68", "69", "70",
        "71", "72", "73", "74", "75", "76", "77", "78", "79", "80", "81", "82",
        "83", "84", "85", "86", "87", "88", "89", "90", "91", "92", "93", "94",
        "95", "96", "97", "98", "99", "100", "101", "102", "103", "104", "105",
        "106", "107", "108", "109", "110", "111", "112", "113", "114", "115",
        "116", "117", "118", "119", "120", "121", "122", "123", "124", "125",
        "126", "127", "128", "129", "130", "131", "132", "133", "134", "135",
        "136", "137", "138", "139", "140", "141", "142", "143", "144", "145",
        "146", "147", "148", "149", "150", "151", "152", "153", "154", "155",
```

```
        "156", "157", "158", "159", "160", "161", "162", "163", "164", "165",
        "166", "167", "168", "169", "170", "171", "172", "173", "174", "175",
        "176", "177", "178", "179", "180", "181", "182", "183", "184", "185",
        "186", "187", "188", "189", "190", "191", "192", "193", "194", "195",
        "196", "197", "198", "199", "200", "201", "202", "203", "204", "205",
        "206", "207", "208", "209", "210", "211", "212", "213", "214", "215",
        "216", "217", "218", "219", "220", "221", "222", "223", "224", "225",
        "226", "227", "228", "229", "230", "231", "232", "233", "234", "235",
        "236", "237", "238", "239", "240", "241", "242", "243", "244", "245",
        "246", "247", "248", "249", "250", "251", "252", "253", "254", "255",
        "256", "257", "258", "259", "260", "261", "262", "263", "264", "265"))
mathe<-mathe.stud<-
structure(.Data = c(2, 2, 4, 1.3, 3, 1, 4, 3.3, 3, 3, 3.3, 2, 2, 1, 3, 2.5, 5,
        5, 4, 3, 4, 3, 4, 5, 1, 1, 4, 4.3, 3.3, 2, 4, 3, 4, 3.3, 5, 5, 1.3, 3,
        1, 2, 1, 3, 4, 4, 1.7, 3, 3, 1, 4, 4, 3, 5, 2, 2, 3, 2, 3.3, 2, 3, 2.3,
        4, 5, 2, 3, 1, 3, 3, 6, 3, 4, 1, 2.7, 2, 4, 2, 3, 3, 3, 3, 4, 1, 2.7, 3,
        3, 3, 2, 3, 1.7, 4, 4, 2, 1.3, 2.2, 1, 4, 3, 1.7, 3, 4, 3, 3, 4.3, 3, 3,
        4, 3, 3, 4, 3, 4.7, 4, 4, 3, 2, 5, 2, 3.3, 2, 3.7, 2.7, 4, 3, 1, 1.7, 4,
        4, 3, 1.3, 4, 2, 6, 2.3, 3, 4, 3, 3.7, 3, 4, 3, 3, 3, 4, 2, 3, 3, 2.7,
        2.3, 2, 3, 3, 3, 1.3, 2.7, 2.3, 1.3, 2.3, 2.3, 1.7, 3, 2, 5, 4, 3, 1, 1,
        1, 2, 1, 3, 1.5, 2, 3.7, 3, 3.3, 4, 2, 3, 4, 3, 4, 2, 2, 3, 3, 3, 3, 4,
        3.7, 6, 1, 2, 4.7, 2.3, 6, 4, 1.3, 5, 2.7, 1.3, 3.3, 4, 4, 3, 4, 3, 3,
        1.7, 5, 3, 2, 3.7, 3.3, 1.7, 2, 3, 3, 1, 3, 4.3, 4, 2.3, 3, 4, 2.3, 3,
        2.7, 2.7, 1.3, 2.7, 3.7, 4, 3, 5, 3, 3, 5, 3), .Names = c("1", "2", "3",
        "4", "5", "7", "8", "9", "10", "11", "12", "13", "14", "15", "17", "18",
        "19", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30", "31",
        "32", "33", "34", "36", "37", "38", "39", "40", "41", "42", "43", "44",
        "45", "48", "49", "50", "51", "52", "53", "54", "55", "56", "57", "58",
        "59", "60", "61", "62", "63", "64", "65", "66", "67", "68", "69", "70",
        "71", "72", "73", "74", "75", "78", "79", "80", "81", "82", "84", "85",
        "87", "88", "89", "90", "91", "92", "93", "94", "95", "96", "97", "98",
        "99", "100", "101", "102", "103", "104", "105", "106", "107", "108",
        "109", "110", "111", "112", "113", "114", "115", "116", "117", "118",
        "119", "120", "121", "123", "124", "125", "126", "127", "129", "130",
        "131", "132", "134", "135", "136", "137", "138", "139", "140", "141",
        "142", "143", "144", "145", "146", "147", "148", "150", "151", "152",
        "153", "154", "156", "157", "158", "159", "160", "161", "162", "163",
        "164", "165", "167", "168", "169", "170", "171", "172", "173", "174",
        "175", "176", "177", "178", "181", "182", "184", "185", "186", "187",
        "188", "190", "192", "193", "194", "195", "196", "197", "199", "200",
        "201", "202", "204", "205", "206", "207", "209", "210", "211", "212",
        "213", "214", "215", "216", "217", "220", "221", "222", "223", "224",
        "225", "226", "227", "229", "231", "232", "233", "234", "235", "236",
        "237", "238", "239", "240", "241", "242", "243", "244", "245", "246",
        "247", "248", "249", "250", "251", "252", "253", "254", "255", "256",
        "257", "258", "259", "260", "261", "262", "263", "264", "265"))
gewicht.stud<-
c(68, 75, 63, 85, 47, 63, 70, 85, 98, 50, 71, 67, 63, 56, 56, 75, 53, 52, 76,
        70, 59, 100, 56, 86, 46, 79, 83, 50, 70, 48, 100, 69, 65.5, 60, 70, 80,
        72, 70, 68, 73, 73, 83, 70, 78, 57, 66, 72, 57, 54, 61, 60, 56, 57, 75,
        118, 80, 54, 50, 72, 70, 77, 72, 90, 73, 75, 75, 52, 70, 85, 67, 70, 52,
        94, 79, 87, 75, 60, 67, 69, 69, 62, 53, 75, 53, 56, 53, 60, 60, 58, 82,
        98, 70, 74, 57, 74, 65, 90, 78, 73, 83, 80, 77, 49, 65, 83, 80, 70, 65,
        75, 65, 74, 55, 53, 76, 49, 80, 75, 65, 78, 63, 60, 65, 74, 62, 57, 66,
        68, 80, 72, 76, 49, 76, 55, 73, 78, 75, 57, 63, 60, 72, 78, 67, 71, 70,
        65, 78, 73, 85, 68, 79, 45, 95, 83, 65, 65, 53, 58, 70, 70, 70, 58, 68,
        50, 67, 65, 68, 85, 85, 85, 70, 78, 75, 110, 94, 70, 60, 56, 76, 57, 60,
        70, 80, 65, 64, 53, 55, 90, 58, 72, 69, 67, 60, 90)
```

```
set.seed(2)
x<-sample(buecher.stud,size=20)

klicks.moebel<-c(207,
31,
173,
229,
125,
179,
105,
51,
160,
78,
1299,
226,
158,
32,
65,
237,
27,
80,
143,
153,
131,
2169,
77,
184,
146,
34,
79,
56,
66,
850,
177,
696,
545,
8,
14,
92,
130,
7,
36,
251,
55,
167,
112,
74,
59,
381,
32,
22,
106,
48,
46,
69,
572,
97,
```

```
279,
137,
35,
43,
61,
238,
36,
120,
24,
35,
11,
35,
78,
650,
613,
130,
66,
12,
60,
152,
130,
86,
113,
121,
192,
186,
237,
175,
131,
58,
135,
165,
276,
45,
262,
54,
83,
857,
63,
73,
58,
41,
257,
118,
227,
157,
133,
13,
105,
7,
9,
1059,
18,
31,
34,
178,
13,
38,
```

```
64,
81,
31,
42,
10,
124,
1746,
381,
23,
95,
32,
19,
31,
787,
20,
104,
138,
131,
66,
153,
448,
759,
60,
12,
35,
35,
146,
144,
16,
98,
73,
39,
12,
296,
276,
375,
41,
34,
22,
417,
70,
18,
221,
173,
10,
46,
)
"chal"<-structure(.Data = list(temp = c(53, 57, 58, 63, 66, 67, 67, 67, 68, 69, 70, 70
"blaetter"<-structure(.Data = c(6.0999999999999996, 5.9000000000000004, 6.099999999999
"hua"<-structure(.Data = c(688, 326, 343, 98, 116, 38, 84, 48, 584, 241, 909, 403, 188
"shosho"<-c(0.69299999999999995, 0.66200000000000003, 0.68999999999999995,      0.6059
"stat1.2"<-structure(.Data = c(3, 4, 3, 3, 2, 4, 3, 4, 3, 2, 3, 1, 4, 2, 2, 3, 4, 4, 2
"pi.vec"<-
c(3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3, 2, 3, 8, 4, 6, 2, 6, 4, 3, 3,
        8, 3, 2, 7, 9, 5, 0, 2, 8, 8, 4, 1, 9, 7, 1, 6, 9, 3, 9, 9, 3, 7, 5, 1,
        0, 5, 8, 2, 0, 9, 7, 4, 9, 4, 4, 5, 9, 2, 3, 0, 7, 8, 1, 6, 4, 0, 6, 2,
        8, 6, 2, 0, 8, 9, 9, 8, 6, 2, 8, 0, 3, 4, 8, 2, 5, 3, 4, 2, 1, 1, 7, 0,
        6, 7, 9, 8, 2, 1, 4, 8, 0, 8, 6, 5, 1, 3, 2, 8, 2, 3, 0, 6, 6, 4, 7, 0,
```

```
9, 3, 8, 4, 4, 6, 0, 9, 5, 5, 0, 5, 8, 2, 2, 3, 1, 7, 2, 5, 3, 5, 9, 4,
0, 8, 1, 2, 8, 4, 8, 1, 1, 1, 7, 4, 5, 0, 2, 8, 4, 1, 0, 2, 7, 0, 1, 9,
3, 8, 5, 2, 1, 1, 0, 5, 5, 5, 9, 6, 4, 4, 6, 2, 2, 9, 4, 8, 9, 5, 4, 9,
3, 0, 3, 8, 1, 9, 6, 4, 4, 2, 8, 8, 1, 0, 9, 7, 5, 6, 6, 5, 9, 3, 3, 4,
4, 6, 1, 2, 8, 4, 7, 5, 6, 4, 8, 2, 3, 3, 7, 8, 6, 7, 8, 3, 1, 6, 5, 2,
7, 1, 2, 0, 1, 9, 0, 9, 1, 4, 5, 6, 4, 8, 5, 6, 6, 9, 2, 3, 4, 6, 0, 3,
4, 8, 6, 1, 0, 4, 5, 4, 3, 2, 6, 6, 4, 8, 2, 1, 3, 3, 9, 3, 6, 0, 7, 2,
6, 0, 2, 4, 9, 1, 4, 1, 2, 7, 3, 7, 2, 4, 5, 8, 7, 0, 0, 6, 6, 0, 6, 3,
1, 5, 5, 8, 8, 1, 7, 4, 8, 8, 1, 5, 2, 0, 9, 2, 0, 9, 6, 2, 8, 2, 9, 2,
5, 4, 0, 9, 1, 7, 1, 5, 3, 6, 4, 3, 6, 7, 8, 9, 2, 5, 9, 0, 3, 6, 0, 0,
1, 1, 3, 3, 0, 5, 3, 0, 5, 4, 8, 8, 2, 0, 4, 6, 6, 5, 2, 1, 3, 8, 4, 1,
4, 6, 9, 5, 1, 9, 4, 1, 5, 1, 1, 6, 0, 9, 4, 3, 3, 0, 5, 7, 2, 7, 0, 3,
6, 5, 7, 5, 9, 5, 9, 1, 9, 5, 3, 0, 9, 2, 1, 8, 6, 1, 1, 7, 3, 8, 1, 9,
3, 2, 6, 1, 1, 7, 9, 3, 1, 0, 5, 1, 1, 8, 5, 4, 8, 0, 7, 4, 4, 6, 2, 3,
7, 9, 9, 6, 2, 7, 4, 9, 5, 6, 7, 3, 5, 1, 8, 8, 5, 7, 5, 2, 7, 2, 4, 8,
9, 1, 2, 2, 7, 9, 3, 8, 1, 8, 3, 0, 1, 1, 9, 4, 9, 1, 2, 9, 8, 3, 3, 6,
7, 3, 3, 6, 2, 4, 4, 0, 6, 5, 6, 6, 4, 3, 0, 8, 6, 0, 2, 1, 3, 9, 4, 9,
4, 6, 3, 9, 5, 2, 2, 4, 7, 3, 7, 1, 9, 0, 7, 0, 2, 1, 7, 9, 8, 6, 0, 9,
4, 3, 7, 0, 2, 7, 7, 0, 5, 3, 9, 2, 1, 7, 1, 7, 6, 2, 9, 3, 1, 7, 6, 7,
5, 2, 3, 8, 4, 6, 7, 4, 8, 1, 8, 4, 6, 7, 6, 6, 9, 4, 0, 5, 1, 3, 2, 0,
0, 0, 5, 6, 8, 1, 2, 7, 1, 4, 5, 2, 6, 3, 5, 6, 0, 8, 2, 7, 7, 8, 5, 7,
7, 1, 3, 4, 2, 7, 5, 7, 7, 8, 9, 6, 0, 9, 1, 7, 3, 6, 3, 7, 1, 7, 8, 7,
2, 1, 4, 6, 8, 4, 4, 0, 9, 0, 1, 2, 2, 4, 9, 5, 3, 4, 3, 0, 1, 4, 6, 5,
4, 9, 5, 8, 5, 3, 7, 1, 0, 5, 0, 7, 9, 2, 2, 7, 9, 6, 8, 9, 2, 5, 8, 9,
2, 3, 5, 4, 2, 0, 1, 9, 9, 5, 6, 1, 1, 2, 1, 2, 9, 0, 2, 1, 9, 6, 0, 8,
6, 4, 0, 3, 4, 4, 1, 8, 1, 5, 9, 8, 1, 3, 6, 2, 9, 7, 7, 4, 7, 7, 1, 3,
0, 9, 9, 6, 0, 5, 1, 8, 7, 0, 7, 2, 1, 1, 3, 4, 9, 9, 9, 9, 9, 9, 8, 3,
7, 2, 9, 7, 8, 0, 4, 9, 9, 5, 1, 0, 5, 9, 7, 3, 1, 7, 3, 2, 8, 1, 6, 0,
9, 6, 3, 1, 8, 5, 9, 5, 0, 2, 4, 4, 5, 9, 4, 5, 5, 3, 4, 6, 9, 0, 8, 3,
0, 2, 6, 4, 2, 5, 2, 2, 3, 0, 8, 2, 5, 3, 3, 4, 4, 6, 8, 5, 0, 3, 5, 2,
6, 1, 9, 3, 1, 1, 8, 8, 1, 7, 1, 0, 1, 0, 0, 0, 3, 1, 3, 7, 8, 3, 8, 7,
5, 2, 8, 8, 6, 5, 8, 7, 5, 3, 3, 2, 0, 8, 3, 8, 1, 4, 2, 0, 6, 1, 7, 1,
7, 7, 6, 6, 9, 1, 4, 7, 3, 0, 3, 5, 9, 8, 2, 5, 3, 4, 9, 0, 4, 2, 8, 7,
5, 5, 4, 6, 8, 7, 3, 1, 1, 5, 9, 5, 6, 2, 8, 6, 3, 8, 8, 2, 3, 5, 3, 7,
8, 7, 5, 9, 3, 7, 5, 1, 9, 5, 7, 7, 8, 1, 8, 5, 7, 7, 8, 0, 5, 3, 2, 1,
7, 1, 2, 2, 6, 8, 0, 6, 6, 1, 3, 0, 0, 1, 9, 2, 7, 8, 7, 6, 6, 1, 1, 1,
9, 5, 9, 0, 9, 2, 1, 6, 4, 2, 0, 1, 9, 8, 9, 3, 8, 0, 9, 5, 2, 5, 7, 2,
0, 1, 0, 6, 5, 4, 8, 5, 8, 6, 3, 2, 7, 8, 8, 6, 5, 9, 3, 6, 1, 5, 3, 3,
8, 1, 8, 2, 7, 9, 6, 8, 2, 3, 0, 3, 0, 1, 9, 5, 2, 0, 3, 5, 3, 0, 1, 8,
5, 2, 9, 6, 8, 9, 9, 5, 7, 7, 3, 6, 2, 2, 5, 9, 9, 4, 1, 3, 8, 9, 1, 2,
4, 9, 7, 2, 1, 7, 7, 5, 2, 8, 3, 4, 7, 9, 1, 3, 1, 5, 1, 5, 5, 7, 4, 8,
5, 7, 2, 4, 2, 4, 5, 4, 1, 5, 0, 6, 9, 5, 9, 5, 0, 8, 2, 9, 5, 3, 3, 1,
1, 6, 8, 6, 1, 7, 2, 7, 8, 5, 5, 8, 8, 9, 0, 7, 5, 0, 9, 8, 3, 8, 1, 7,
5, 4, 6, 3, 7, 4, 6, 4, 9, 3, 9, 3, 1, 9, 2, 5, 5, 0, 6, 0, 4, 0, 0, 9,
2, 7, 7, 0, 1, 6, 7, 1, 1, 3, 9, 0, 0, 9, 8, 4, 8, 8, 2, 4, 0, 1, 2, 8,
5, 8, 3, 6, 1, 6, 0, 3, 5, 6, 3, 7, 0, 7, 6, 6, 0, 1, 0, 4, 7, 1, 0, 1,
8, 1, 9, 4, 2, 9, 5, 5, 5, 9, 6, 1, 9, 8, 9, 4, 6, 7, 6, 7, 8, 3, 7, 4,
4, 9, 4, 4, 8, 2, 5, 5, 3, 7, 9, 7, 7, 4, 7, 2, 6, 8, 4, 7, 1, 0, 4, 0,
4, 7, 5, 3, 4, 6, 4, 6, 2, 0, 8, 0, 4, 6, 6, 8, 4, 2, 5, 9, 0, 6, 9, 4,
9, 1, 2, 9, 3, 3, 1, 3, 6, 7, 7, 0, 2, 8, 9, 8, 9, 1, 5, 2, 1, 0, 4, 7,
5, 2, 1, 6, 2, 0, 5, 6, 9, 6, 6, 0, 2, 4, 0, 5, 8, 0, 3, 8, 1, 5, 0, 1,
9, 3, 5, 1, 1, 2, 5, 3, 3, 8, 2, 4, 3, 0, 0, 3, 5, 5, 8, 7, 6, 4, 0, 2,
4, 7, 4, 9, 6, 4, 7, 3, 2, 6, 3, 9, 1, 4, 1, 9, 9, 2, 7, 2, 6, 0, 4, 2,
6, 9, 9, 2, 2, 7, 9, 6, 7, 8, 2, 3, 5, 4, 7, 8, 1, 6, 3, 6, 0, 0, 9, 3,
4, 1, 7, 2, 1, 6, 4, 1, 2, 1, 9, 9, 2, 4, 5, 8, 6, 3, 1, 5, 0, 3, 0, 2,
8, 6, 1, 8, 2, 9, 7, 4, 5, 5, 5, 7, 0, 6, 7, 4, 9, 8, 3, 8, 5, 0, 5, 4,
9, 4, 5, 8, 8, 5, 8, 6, 9, 2, 6, 9, 9, 5, 6, 9, 0, 9, 2, 7, 2, 1, 0, 7,
9, 7, 5, 0, 9, 3, 0, 2, 9, 5, 5, 3, 2, 1, 1, 6, 5, 3, 4, 4, 9, 8, 7, 2,
```

```
0, 2, 7, 5, 5, 9, 6, 0, 2, 3, 6, 4, 8, 0, 6, 6, 5, 4, 9, 9, 1, 1, 9, 8,
8, 1, 8, 3, 4, 7, 9, 7, 7, 5, 3, 5, 6, 6, 3, 6, 9, 8, 0, 7, 4, 2, 6, 5,
4, 2, 5, 2, 7, 8, 6, 2, 5, 5, 1, 8, 1, 8, 4, 1, 7, 5, 7, 4, 6, 7, 2, 8,
9, 0, 9, 7, 7, 7, 7, 2, 7, 9, 3, 8, 0, 0, 0, 8, 1, 6, 4, 7, 0, 6, 0, 0,
1, 6, 1, 4, 5, 2, 4, 9, 1, 9, 2, 1, 7, 3, 2, 1, 7, 2, 1, 4, 7, 7, 2, 3,
5, 0, 1, 4, 1, 4, 4, 1, 9, 7, 3, 5, 6, 8, 5, 4, 8, 1, 6, 1, 3, 6, 1, 1,
5, 7, 3, 5, 2, 5, 5, 2, 1, 3, 3, 4, 7, 5, 7, 4, 1, 8, 4, 9, 4, 6, 8, 4,
3, 8, 5, 2, 3, 3, 2, 3, 9, 0, 7, 3, 9, 4, 1, 4, 3, 3, 3, 4, 5, 4, 7, 7,
6, 2, 4, 1, 6, 8, 6, 2, 5, 1, 8, 9, 8, 3, 5, 6, 9, 4, 8, 5, 5, 6, 2, 0,
9, 9, 2, 1, 9, 2, 2, 2, 1, 8, 4, 2, 7, 2, 5, 5, 0, 2, 5, 4, 2, 5, 6, 8,
8, 7, 6, 7, 1, 7, 9, 0, 4, 9, 4, 6, 0, 1, 6, 5, 3, 4, 6, 6, 8, 0, 4, 9,
8, 8, 6, 2, 7, 2, 3, 2, 7, 9, 1, 7, 8, 6, 0, 8, 5, 7, 8, 4, 3, 8, 3, 8,
2, 7, 9, 6, 7, 9, 7, 6, 6, 8, 1, 4, 5, 4, 1, 0, 0, 9, 5, 3, 8, 8, 3, 7,
8, 6, 3, 6, 0, 9, 5, 0, 6, 8, 0, 0, 6, 4, 2, 2, 5, 1, 2, 5, 2, 0, 5, 1,
1, 7, 3, 9, 2, 9, 8, 4, 8, 9, 6, 0, 8, 4, 1, 2, 8, 4, 8, 8, 6, 2, 6, 9,
4, 5, 6, 0, 4, 2, 4, 1, 9, 6, 5, 2, 8, 5, 0, 2, 2, 2, 1, 0, 6, 6, 1, 1,
8, 6, 3, 0, 6, 7, 4, 4, 2, 7, 8, 6, 2, 2, 0, 3, 9, 1, 9, 4, 9, 4, 5, 0,
4, 7, 1, 2, 3, 7, 1, 3, 7, 8, 6, 9, 6, 0, 9, 5, 6, 3, 6, 4, 3, 7, 1, 9,
1, 7, 2, 8, 7, 4, 6, 7, 7, 6, 4, 6, 5, 7, 5, 7, 3, 9, 6, 2, 4, 1, 3, 8,
9, 0, 8, 6, 5, 8, 3, 2, 6, 4, 5, 9, 9, 5, 8, 1, 3, 3, 9, 0, 4, 7, 8, 0,
2, 7, 5, 9, 0, 0, 9, 9, 4, 6, 5, 7, 6, 4, 0, 7, 8, 9, 5, 1, 2, 6, 9, 4,
6, 8, 3, 9, 8, 3, 5, 2, 5, 9, 5, 7, 0, 9, 8, 2, 5, 8, 2, 2, 6, 2, 0, 5,
2, 2, 4, 8, 9, 4, 0, 7, 7, 2, 6, 7, 1, 9, 4, 7, 8, 2, 6, 8, 4, 8, 2, 6,
0, 1, 4, 7, 6, 9, 9, 0, 9, 0, 2, 6, 4, 0, 1, 3, 6, 3, 9, 4, 4, 3, 7, 4,
5, 5, 3, 0, 5, 0, 6, 8, 2, 0, 3, 4, 9, 6, 2, 5, 2, 4, 5, 1, 7, 4, 9, 3,
9, 9, 6, 5, 1, 4, 3, 1, 4, 2, 9, 8, 0, 9, 1, 9, 0, 6, 5, 9, 2, 5, 0, 9,
3, 7, 2, 2, 1, 6, 9, 6, 4, 6, 1, 5, 1, 5, 7, 0, 9, 8, 5, 8, 3, 8, 7, 4,
1, 0, 5, 9, 7, 8, 8, 5, 9, 5, 9, 7, 7, 2, 9, 7, 5, 4, 9, 8, 9, 3, 0, 1,
6, 1, 7, 5, 3, 9, 2, 8, 4, 6, 8, 1, 3, 8, 2, 6, 8, 6, 8, 3, 8, 6, 8, 9,
4, 2, 7, 7, 4, 1, 5, 5, 9, 9, 1, 8, 5, 5, 9, 2, 5, 2, 4, 5, 9, 5, 3, 9,
5, 9, 4, 3, 1, 0, 4, 9, 9, 7, 2, 5, 2, 4, 6, 8, 0, 8, 4, 5, 9, 8, 7, 2,
7, 3, 6, 4, 4, 6, 9, 5, 8, 4, 8, 6, 5, 3, 8, 3, 6, 7, 3, 6, 2, 2, 2, 6,
2, 6, 0, 9, 9, 1, 2, 4, 6, 0, 8, 0, 5, 1, 2, 4, 3, 8, 8, 4, 3, 9, 0, 4,
5, 1, 2, 4, 4, 1, 3, 6, 5, 4, 9, 7, 6, 2, 7, 8, 0, 7, 9, 7, 7, 1, 5, 6,
9, 1, 4, 3, 5, 9, 9, 7, 7, 0, 0, 1, 2, 9, 6, 1, 6, 0, 8, 9, 4, 4, 1, 6,
9, 4, 8, 6, 8, 5, 5, 5, 8, 4, 8, 4, 0, 6, 3, 5, 3, 4, 2, 2, 0, 7, 2, 2,
2, 5, 8, 2, 8, 4, 8, 8, 6, 4, 8, 1, 5, 8, 4, 5, 6, 0, 2, 8, 5, 0, 6, 0,
1, 6, 8, 4, 2, 7, 3, 9, 4, 5, 2, 2, 6, 7, 4, 6, 7, 6, 7, 8, 8, 9, 5, 2,
5, 2, 1, 3, 8, 5, 2, 2, 5, 4, 9, 9, 5, 4, 6, 6, 6, 7, 2, 7, 8, 2, 3, 9,
8, 6, 4, 5, 6, 5, 9, 6, 1, 1, 6, 3, 5, 4, 8, 8, 6, 2, 3, 0, 5, 7, 7, 4,
5, 6, 4, 9, 8, 0, 3, 5, 5, 9, 3, 6, 3, 4, 5, 6, 8, 1, 7, 4, 3, 2, 4, 1,
1, 2, 5, 1, 5, 0, 7, 6, 0, 6, 9, 4, 7, 9, 4, 5, 1, 0, 9, 6, 5, 9, 6, 0,
9, 4, 0, 2, 5, 2, 2, 8, 8, 7, 9, 7, 1, 0, 8, 9, 3, 1, 4, 5, 6, 6, 9, 1,
3, 6, 8, 6, 7, 2, 2, 8, 7, 4, 8, 9, 4, 0, 5, 6, 0, 1, 0, 1, 5, 0, 3, 3,
0, 8, 6, 1, 7, 9, 2, 8, 6, 8, 0, 9, 2, 0, 8, 7, 4, 7, 6, 0, 9, 1, 7, 8,
2, 4, 9, 3, 8, 5, 8, 9, 0, 0, 9, 7, 1, 4, 9, 0, 9, 6, 7, 5, 9, 8, 5, 2,
6, 1, 3, 6, 5, 5, 4, 9, 7, 8, 1, 8, 9, 3, 1, 2,)
```

319 ⟨*andere fkt* 319⟩≡

```
dichte.plot<-
function(x, fenster = 2 * iqd(x), xlim = F, Main = deparse(substitute(x)),
        Xlab = paste("Fensterbreite:", fenster), AXES = T, typ = "g", add = F, lty = 2)
{
        x.dens <- density(x, width= fenster, window = typ)
        if(typ == "r") {
                x.dens$y <- c(0, 0, x.dens$y, 0, 0)
                x.dens$x <- c(min(x.dens$x) - 1, min(x.dens$x), x.dens$x, max(x.dens$x),
        }
        if(!add) {
                if(is.logical(xlim[1])) {
                        plot(x.dens, ylab = "", type = "l", xlab = Xlab, main = Main, ax
                }
                else {
                        plot(x.dens, ylab = "", type = "l", xlab = Xlab, main = Main, ax
                }
                if(!AXES){
                        axis(1)
                        axis(2)
                }
        }
        else {
                lines(x.dens, lty = lty)
        }
}
iqd<-
function(x)
return(summary(x)[5] - summary(x)[2])

emp.cdf<-
function(x, stetig = T, MAIN = deparse(substitute(x)), cx = 0.9, Print.ret = T, xlab = "
{
        n <- length(x)
        if(stetig) {
                x.haeuf <- haeuf.stet(x,breaks,anzahl.klassen)
                x.ret <- cbind(x.haeuf, rel.H = x.haeuf[, 3]/n, rel.cum.H = cumsum(x.hae
                plot.x <- c(x.haeuf[, 1], x.haeuf[dim(x.haeuf)[1], 2])
                range.x <- diff(c(plot.x[1], plot.x[length(plot.x)]))
                plot.y <- c(0, cumsum(x.haeuf[, 3]))/n
                plot.x <- c(plot.x[1] - 0.1 * range.x, plot.x, plot.x[length(plot.x)] +
                plot.type <- "l"
        }
        else {
                x.haeuf <- haeuf(x)
                x.range<-diff(range(x))
                x.ret <- cbind(x.haeuf, rel.H = x.haeuf[, 2]/n, rel.cum.H = cumsum(x.hae
                plot.x <- x.haeuf[, 1]
                plot.y <- cumsum(x.haeuf[, 2])/n
                plot.x <- c(plot.x[1]-0.05*x.range, plot.x, plot.x[length(plot.x)] + 0.0
                plot.type <- "s"
        }
        plot.y <- c(0, plot.y, 1)
        plot(plot.x, plot.y, type = plot.type, axes = F, xlab = xlab, ylab = "kum. rel.
        title(main = MAIN, cex = cx)
        axis(1)
        axis(2)
        ## print(MAIN)
```

```
        if(Print.ret) return(x.ret)
}


haeuf.stet<-
function(x, breaks = numeric(0), anzahl.klassen = numeric(0))
{
        if(length(breaks) > 1) {
                x.hist <- hist(x, breaks = breaks, plot = F)
                breaks <- x.hist$breaks
                counts <- x.hist$counts
        }
        else {
                if(length(anzahl.klassen) > 0) {
                        x.hist <- hist(x, breaks = seq(min(x), max(x), length = anzah
                        breaks <- x.hist$breaks
                        counts <- x.hist$counts
                }
                else {
                        x.hist <- hist(x, plot = F)
                        breaks <- x.hist$breaks
                        counts <- x.hist$counts
                }
        }
        ug <- breaks[ - length(breaks)]
        og <- breaks[-1]
        h <- cbind(ug.i = ug, og.i = og, n.i = counts, h.i = counts/length(x), F.i = c
        dimnames(h)[[1]] <- 1:length(ug)
        h
}

range.plot<-function(x,marker=c(0.5,0.9),...)
{
p<-seq(0,1,by=0.005)
q.p<-quantile(x,p)
y.u<-q.p[1:(length(q.p)/2)]
y.o<-rev(q.p)[1:(length(q.p)/2)]
y<-y.o-y.u
xx.u<-p[1:(length(q.p)/2)]
xx.o<-rev(p)[1:(length(q.p)/2)]
xx<-xx.o-xx.u
xx<-round(xx,2)
names(xx)<-NULL
names(y)<-NULL
plot(xx,y,type="l",xlab="p%",ylab="Spannweiten",
        axes=F,...)
axis(1,at=seq(0,1,by=.2),labels=seq(0,100,by=20))
axis(2)
for (i in 1:length(marker))
        {
        lines(x=rep(marker[i],2),
                y=c(par("usr")[3],y[xx==marker[i]]),
                lty=3)
        lines(x=c(marker[i],par("usr")[1]),
                y=rep(y[xx==marker[i]],2),
                lty=3)
        }
invisible(cbind(p=xx,s=y))
```

```
}

msd<-function(x)
{
        n<-length(x)
        var(x)*((n-1)/n)
}



vk.plot<-function(x,add=F,...)
{
x<-sort(x)
s.x<-numeric(0)
nn<-n<-length(x)
repeat{
        s.x<-c(s.x,sqrt(var(x))/mean(x))
        x<-x[-c(1,n)]
        n<-length(x)
        nn<-c(nn,n)
        if(n==1){
                s.x<-c(s.x,0)
                break
                }
        if (n==0){
                nn<-nn[-length(nn)]
                break
                }
        }
if(!add)
{
        plot(nn/nn[1],s.x,type="l",axes=F,xlab="p%",ylab="vk",...)
        axis(1,at=seq(0,1,by=0.2),labels=seq(0,100,by=20))
        axis(2)
        }else {
        lines(nn/nn[1],s.x,lty=2)
        }
invisible(cbind(nn,vk=s.x))
}

schiefe<-
function(x)
{
        sum((x - mean(x))^3)/length(x)/msd(x)^(3/2)
}

kurtosis<-
function(x,NV.korrekt=F)
{
if(!NV.korrekt)
        (sum((x - mean(x))^4)/length(x))/msd(x)^2
else    (sum((x - mean(x))^4)/length(x))/msd(x)^2 -3
}

box.cox<-
function(x, lambda = 0)
{
        if(!all(x > 0))
```

```
                x <- x - min(x) + 1
        if(lambda == 0)
                return(log(x))
        else return((x^lambda - 1)/lambda)
}

box.cox.plot<-function(x,lambda=seq(-2,2,by=0.1),NV.korrekt=T,interaktiv=F,...)
{
n.l<-length(lambda)
schiefe.l<-numeric(n.l)
kurtosis.l<-numeric(n.l)
for (i in 1:n.l){
        xx<-box.cox(x,lambda[i])
        schiefe.l[i]<-schiefe(xx)
        kurtosis.l[i]<-kurtosis(xx,NV.korrekt)
        }
plot(schiefe.l,kurtosis.l,xlab="Schiefe",ylab="Kurtosis",...)
if(interaktiv){
cat("!!Achtung:\n
  Bitte Punkt im rechten Bild anklicken\nAbbruch mit rechter Maustaste!!\n")
title("Bitte Punkt anklicken-> lambda Wert\nAbbruch mit rechter Maustaste",cex.main=.8
repeat{
        id<-identify(cbind(schiefe.l,kurtosis.l),plot=F,n=1)
        if(length(id)>0){
                text(schiefe.l[id],kurtosis.l[id],
                        labels=paste("l=",lambda[id]))
                } else {
                        break
                }
        }
}
invisible(cbind(schiefe.l,kurtosis.l,lambda))
}
box.cox.plot<-function(x,lambda=seq(-2,2,by=0.1),NV.korrekt=T,
  interaktiv=F,tit=""){
if(interaktiv==F){
  n.l<-length(lambda); schiefe.l<-numeric(n.l)
  kurtosis.l<-numeric(n.l)
  for (i in 1:n.l){
        xx<-box.cox(x,lambda[i])
        schiefe.l[i]<-schiefe(xx)
        kurtosis.l[i]<-kurtosis(xx,NV.korrekt)
  }
  plot(schiefe.l,kurtosis.l,xlab="Schiefe",ylab="Kurtosis")
  title(tit)
  invisible(cbind(schiefe.l,kurtosis.l,lambda))
} else {
  #### interaktiv==TRUE
new.graph<-function(){
    if(exists("win.graph")) win.graph()
    if(exists("x11")) x11()
    "relax"
}
new.graph()
redo<-function(...){
  n.l<-length(lambda); schiefe.l<-kurtosis.l<-numeric(n.l)
  for (i in 1:n.l){
        xx<-box.cox(x,lambda[i])
```

```
        schiefe.l[i]<-schiefe(xx)
        kurtosis.l[i]<-kurtosis(xx,NV.korrekt)
  }
  plot(schiefe.l,kurtosis.l,xlab="Schiefe",ylab="Kurtosis")
  lambda.mark<-slider(no=1); xx<-box.cox(x,lambda.mark)
  sch<-schiefe(xx); kur<-kurtosis(xx,NV.korrekt)
  points(sch,kur,pch=19,cex=2)
  title(paste(tit,"\nlambda=",signif(lambda.mark,3)))
  cat("Schiefe:",sch,", kurtosis:",kur,"\n")
}
  slider(redo, "lambda",min(lambda),max(lambda),
       diff(lambda[1:2]), lambda[1]     )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
}

match.all<-
function(such, daten)
{
        n <- length(daten)
        a <- aa <- rep(0, n)
        a[match(such, daten)] <- 1
        aa[match(such, rev(daten))] <- 1
        return(cumsum(a) == ((1 + sum(aa)) - rev(cumsum(aa))))
}

haeufigkeit.diskret<-function(x)
{
x.tab<-table(x)
erg<-cbind(
        i=1:length(x.tab),
        x.i=names(x.tab),
        n.i=x.tab,
        h.i=round(x.tab/sum(x.tab),3),
        F.i=round(cumsum(x.tab/sum(x.tab)),3))


dimnames(erg)<-list(NULL,c("i","x.i","n.i","h.i","F.i"))
as.data.frame(erg)
}


halbe.halbe<-function(x)
{
n<-length(x)
if((n%%2==0)) {
        cat(x[1:(n/2)],
                "\n",
                x[(n/2+1):n],"\n"
        )
        } else {
        cat(x[1:ceiling(n/2)],
                "\n",
                x[(ceiling(n/2)+1):n],"\n"
        )
}
```

```
}

plot.ohne<-function(x,ohne=1,...)
{
x.lim<-range(x)
x.sort<-rev(sort(x))
if(ohne>0) x.cur<-x[x<x.sort[ohne]] else x.cur<-x
plot(x.cur,1:length(x.cur),bty="n",xlim=x.lim,...)
abline(v=mean(x.cur))
mtext(round(mean(x.cur),1),at=mean(x.cur),side=1,line=1,cex=.8)
}
plot.trim<-function(x,alpha=seq(0,0.5,by=0.05))
{
x.alpha<-rep(0,length(alpha))
cat("zeilenweise: alpha, getrimmtes Mittel\n")
for (i in 1:length(alpha))
        cat(alpha[i],round(x.alpha[i]<-mean(x,trim=alpha[i]),2),"\n")
par(mfrow=c(1,1))
plot(alpha,x.alpha,type="b", ylab="getrimmtes Mittel",bty="n")
abline(h=c(mean(x),median(x)),lty=3)
title("Verschiedene getrimmte arithmetische Mittel")
NULL
}
modus<-
function(x, stetig = T)
{
        if(stetig) {
                x.hist <- hist(x, plot = F)
                wo <- (1:length(x.hist$counts))[max(x.hist$counts) == x.hist$counts]
                if(length(wo) == 1)
                        return(mean(x.hist$breaks[wo:(wo + 1)]))
                else return(NA)
        }
        else {
                x.haeuf <- haeuf(x)
                wo <- (1:dim(x.haeuf)[1])[max(x.haeuf[, 2]) == x.haeuf[, 2]]
                if(length(wo) == 1)
                        return(as.vector(x.haeuf[wo, 1]))
                else return(NA)
        }
}

haeuf<-
function(x)
{
        s <- sort(unique(x))
        n <- length(s)
        h <- rep(0, n)
        for(i in 1:n)
                h[i] <- sum(s[i] == x)
        cbind(s, h)
}

wiederholte.stichproben<-function(x,n,wdh,split=T)
{
xx<-matrix(0,nrow=n,ncol=wdh)
for (i in 1:wdh)
        xx[,i]<-sample(x,n)
```

```
if(split)
        return(split(xx,col(xx)))
else return(xx)
}
dichte.manip<-function(x,fenster=iqd(x))
{
###Vorbereitung

refresh.code<-function(...){
fenster<-slider(no=1)
dichte<-density(x,width=fenster)
plot(dichte,axes=F,main="Dichtespur",ylab="")
axis(1)
}
  slider(refresh.code,
  # names of sliders
       c("Fensterbreite"),
  # min of sliders
       c(0.01),
  # max of sliders
       c(3*iqd(x)),
  # step of sliders
       c(0.01),
  # initial values
       c(fenster)
        )
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
wuerfel.exp<-function(wdh=100,p.sechs=1/6,wdh.max=2000)
{
###Vorbereitung

refresh.code<-function(...){
wdh<-slider(no=1)
p.sechs<-slider(no=2)
stichprobe<-sample(1:6,wdh,replace=T,
        prob=c(rep((1-p.sechs)/5,5),p.sechs))
par(mfrow=c(2,2))
erg<-barplot(table(stichprobe),ylim=c(0,wdh.max/6)*1.5)
text(erg,table(stichprobe)-0.03*par()$usr[4],table(stichprobe),
        cex=.8)
erg<-barplot(table(stichprobe)/wdh,ylim=c(0,1/6)*2)
text(erg,table(stichprobe)/wdh-0.03*par()$usr[4],
        round(table(stichprobe)/wdh,3),
        cex=.8)
plot(1:wdh,cumsum(stichprobe==6)/(1:wdh),type="l",
        ylim=c(0,1),ylab="rel. H\"aufigkeit 6",xlab="Anzahl W\"urfe",bty="n")
abline(h=p.sechs,col=2,lty=3)
plot(1:wdh,cumsum(stichprobe==6),type="l",ylab="Anzahl 6",xlab="Anzahl W\"urfe",bty="n",
abline(h=p.sechs*wdh,col=2,lty=3)
par(mfrow=c(1,1))
}
  slider(refresh.code,
  # names of sliders
       c("Umfang","WS f\"ur eine 6"),
  # min of sliders
       c(1,0),
  # max of sliders
```

```
        c(wdh.max,1),
    # step of sliders
        c(1,0.001),
    # initial values
        c(wdh,p.sechs)
         )
par(mfrow=c(1,1))
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
eda<-function(x)
{
par(mfrow=c(2,2))
boxplot(x,axes=F,horizontal=T,xlab=deparse(substitute(x)))
axis(1)
hist(x,main="",xlab=deparse(substitute(x)))
plot(x,1:length(x),bty="n",axes=F,ylab="",xlab=deparse(substitute(x)))
axis(1)
title(paste("Mittel:",round(mean(x),0),
        " Median:",round(median(x),0),
        "\nStreuung:",round(sqrt(var(x)),0),
        " Anzahl:",length(x)))
plot(density(x,width=iqd(x)),bty="n",main="Fenster: IQR")
par(mfrow=c(1,1))
}

vert<-function(x,anzahl=10)
{
erg<-numeric(anzahl)
beob<-numeric(anzahl)
zwischen<-c(0,quantile(x,seq(1/anzahl,1,by=1/anzahl)))
for (i in 1:(length(zwischen)-1)){
        erg[i]<-sum(x[x>zwischen[i] & x<=zwischen[i+1]])
        beob[i]<-sum(x>zwischen[i] & x<=zwischen[i+1])
        }
return(list(erg,erg/sum(x),beobachtungen=beob))
}



lorenz<-
function(x,anzahl=10,abs = F, sub = "",titel=F)
{
vert.merkmal<-rep(1/anzahl,anzahl)
vert.andere<-vert(x,anzahl=anzahl)[[2]]
print(sum(vert.merkmal))
        if(abs) {
                vert.merkmal <- sum(vert.merkmal)/length(vert.merkmal)
                vert.andere <- sum(vert.andere)/length(vert.andere)
        }
        vert.merkmal <- c(0, vert.merkmal)
        vert.andere <- c(0, vert.andere)
        plot(cumsum(vert.merkmal), cumsum(vert.andere), type = "l", axes = F,
                xlim = c(0, 1), ylim = c(0, 1), xlab = "%", ylab = "%")
        lines(x = c(0, 1), y = c(0, 1))
        axis(1, at = seq(0, 1, by = 0.1), labels = seq(0, 100, by = 10))
        axis(2, at = seq(0, 1, by = 0.1), labels = seq(0, 100, by = 10))
        axis(3, at = seq(0, 1, by = 0.1), labels = seq(0, 100, by = 10))
        axis(4, at = seq(0, 1, by = 0.1), labels = seq(0, 100, by = 10))
```

```
        print(cbind(vert.merkmal,vert.andere))
        print(cbind(cumsum(vert.merkmal), cumsum(vert.andere)))
        for(i in 2:length(vert.merkmal))
                lines(x = rep(cumsum(vert.merkmal)[i], 2), y = c(cumsum(
                        vert.andere)[i], cumsum(vert.merkmal)[i]), lty = 2)
if(titel)       title(paste(main = "Lorenzkurve (Gini=", round(gini.coef.klass(
                vert.merkmal[-1], vert.andere[-1]), 4), ")", sep = ""), sub =
                sub, cex.main = 0.9)
        print("Die Flaeche (Konzentrationsflaeche):")
        kf <- gini(x,anzahl)
        print(kf/2)
        print("Der Gini-Koeffizient ist:")
        kf
}
gini<-
function(x,anzahl=10, abs = F)
{
vert.merkmal<-rep(1/anzahl,anzahl)
vert.andere<-vert(x,anzahl=anzahl)[[2]]
        if(abs) {
                vert.merkmal <- sum(vert.merkmal)/length(vert.merkmal)
                vert.andere <- sum(vert.andere)/length(vert.andere)
        }
        F.merkmal <- c(0, cumsum(vert.merkmal))
        p <- vert.andere
        F.sum <- F.merkmal[-1] + F.merkmal[ - length(F.merkmal)]
        2 * (0.5 * sum(p * F.sum) - 0.5)
}



image.plot<-function(m,...)
{
image(x=1:dim(m)[2],
        y=1:dim(m)[1],z=t(m),axes=F,col=rev(gray.colors(100)),...)
axis(1,1:dim(m)[2],labels=dimnames(m)[[2]])

axis(2,at=1:dim(m)[1],labels=dimnames(m)[[1]])

for(i in 1:dim(m)[1])
        text(1:dim(m)[2],rep(i,dim(m)[2]),
                paste(round(m[i,]/sum(m)*100,1),"%",sep=""),cex=.8)
}
zeilenprofil<-function(m)
{
par(mfrow=c(2,2))
for (i in 1:dim(m)[1]){
        barplot(m[i,]/sum(m[i,]),ylim=c(0,0.4))
        title(paste(dimnames(m)[[1]][i],sum(m[i,])))
        }
barplot(apply(m,2,sum)/sum(m),ylim=c(0,0.4))
title(paste("Gesamt",sum(m)))
par(mfrow=c(1,1))
cat("Plot erstellt\n")
NULL
}
zeilenprofil.diff<-function(m)
```

```
{
par(mfrow=c(2,2))
minus<-apply(m,2,sum)/sum(m)
for (i in 1:dim(m)[1]){
        barplot((m[i,]/sum(m[i,])-minus)/(m[i,]/sum(m[i,])),
        ylim=c(-.5,.5),ylab="rel. Unterschied")
        title(paste(dimnames(m)[[1]][i],sum(m[i,]),
        "\nim Vergleich zum mittleren Profil"))
        abline(h=0)
        }
par(mfrow=c(1,1))
cat("Plot erstellt\n")
NULL
}
erw.unabh<-function(m){
        mm<-m[dim(m)[1]:1,]
        round(outer(apply(mm,1,sum),apply(m,2,sum),"*")/sum(mm),0)
}
korr.schieber<-function(n=100,korr=0)
{
###Vorbereitung

refresh.code<-function(...){
n<-slider(no=1)
korr<-slider(no=2)
zz<-rmultnorm(n,c(0,0),matrix(c(1,korr,korr,1),byrow=T,ncol=2))
plot(zz,bty="n",ylim=c(-3,3),xlim=c(-3,3),xlab="",ylab="")
title(paste("Anzahl Punkte:",n,"\neingestellte Korrelation",korr))
}
  slider(refresh.code,
  # names of sliders
       c("Anzahl Punkte","Korrelation"),
  # min of sliders
       c(2,-1),
  # max of sliders
       c(1000,1),
  # step of sliders
       c(1,0.01),
  # initial values
       c(n,korr)
        )
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
rmultnorm<-
function(n, mu, vmat, tol = 9.9999999999999982e-08)
{
        p <- ncol(vmat)
        if(length(mu) != p)
                stop("mu vector is the wrong length")
        if(max(abs(vmat - t(vmat))) > tol)
                stop("vmat not symmetric")
        vs <- svd(vmat)
        vsqrt <- t(vs$v %*% (t(vs$u) * sqrt(vs$d)))
        ans <- matrix(rnorm(n * p), nrow = n) %*% vsqrt
        ans <- sweep(ans, 2, mu, "+")
        dimnames(ans) <- list(NULL, dimnames(vmat)[[2]])
        ans
}
```

```
vgl.plots<-function(x,y){
par(mfrow=c(2,2))
gesamt<-c(x,y)
boxplot(x,y,
        names=c(deparse(substitute(x)),deparse(substitute(y))),
        horizontal=T,main="Vergleichende Boxplots\nmit Gesamtmittel",
        bty="n")
abline(v=mean(gesamt),lty=2)



plot(density(y),bty="n",xlim=range(gesamt)*1.1,col=1,
        main="Dichtesch\"atzer",xlab="")
lines(density(x),lty=2)

abline(v=c(mean(x),mean(y)))


hist(y,col=1,density=0,xlim=range(gesamt)*1.1,
        main="Histogramm",xlab="")
hist(x,add=T,density=0,lty=2)

qqplot(x,y,xlim=range(gesamt),ylim=range(gesamt),bty="n",
        xlab=deparse(substitute(x)),ylab=deparse(substitute(y)))
title("QQ-Plot")
abline(0,1)
par(mfrow=c(1,1))
}
lese.ein<-function(frage="bitte Wert eingeben und RETURN",
                   default="hallo"){
  require(tcltk)
  tt<-tktoplevel(); tkwm.geometry(tt,"+0+0")
  tkwm.title(tt,frage)
  Name <- tclVar(default)
  entry.Name <-tkentry(tt,width="75",textvariable=Name)
  tkpack(tklabel(tt,text="please enter number."))
  tkpack(entry.Name)
  tkbind(tt,"<Return>",function(...){
    tkdestroy(tt) })
  tkwait.window(tt)
  eingabe<-tclvalue(Name)
}
lotto.experiment<-function()
{
  win.graph<-function(){
    if(exists("win.graph")) return(win.graph)
    if(exists("x11")) return(x11)
    "relax"
  }
tcltkmenu<-function(choices, graphics=FALSE, title=""){
    TopN<-tktoplevel()
    tkwm.geometry(TopN,"+0+0"); if(title!="")tkwm.title(TopN, title)
    nc<-length(choices<- c(choices, "EXIT"))
    mch<-max(chars<-nchar(choices)); leer<-paste(rep(" ",mch),collapse="")
    choices<-paste(choices, substring(leer,1,mch-chars) )
    escape.brace<-function(text){text<-gsub("\\\\","\\\\\\\\}",text)
      text<-gsub("\\{","\\\\{",text);text<-gsub("\\}","\\\\}",text);text
    }
```

```
    choices<-as.character(escape.brace(choices))
    bw<-as.character(max(mch,10+nchar(title)))
    for(item in choices) {
      tkpack(tkradiobutton(TopN,width=bw,anchor="sw",
                           text=item,variable="tvchoice",value=item))
    }
    tkwait.variable("tvchoice")
    ind <- match(tclvalue("tvchoice"), choices)
    choice <- if(ind==nc) "0" else ind
    tkdestroy(TopN)
    return(as.numeric(choice))
}

choice<-tcltkmenu(c("Verteilung der historischen Lottozahlen",
                "Verteilung bei den 6 Ziehungen",
                "Verteilung der simulierten Lottozahlen",
                "Warten",
                "Zus\"atzliches Warten",
                "Nach einem Tip in den historischen Daten suchen",
                "Nach einem Tip in den simulierten Daten suchen"))
switch(choice,
        {
        par(mfrow=c(1,1))
        erg<-table(lotto)
        plot(table(lotto)/sum(table(lotto)),bty="n",ylab="rel. H\"aufigkeit",
                xlab="Kugeln")
        abline(h=1/49)

        text(names(erg)[c(13,32)],erg[c(13,32)]/sum(erg)+0.02*par()$usr[4],
                round(erg[c(13,32)]/sum(erg),3),cex=.8)
        title(paste(dim(lotto)[2],"Ziehungen\n",
                sum(table(lotto)),"gezogene Kugeln"))
        },{
        par(mfrow=c(3,2))
        for(i in 1:6)
        {
                erg<-table(lotto[i,])
                plot(erg/sum(erg),bty="n",ylab="rel. H\"aufigkeit",
                        xlab="Kugeln")
                abline(h=1/49)
                title(paste(i,". Kugel",sep=""))
        }
        par(mfrow=c(1,1))
        },{
        par(mfrow=c(1,1))
        set.seed(2)
        wieviele<-as.numeric(lese.ein(frage="Wie viele Ziehungen?",default=13000))
        lotto.z<-matrix(0,nrow=6,ncol=wieviele)
        for (i in 1:wieviele)
            lotto.z[,i]<-sample(1:49,6,replace=F)
        erg<-table(lotto.z)
        plot(erg/sum(erg),bty="n",ylab="rel. H\"aufigkeit",
                xlab="Kugeln")
        abline(h=1/49)
        title(paste(dim(lotto.z)[2],"Ziehungen\n",
                sum(table(lotto.z)),"gezogene Kugeln"))
        },{
        zahl<-apply(lotto,2,FUN=function(x) sum(x==13))
```

```
        wo.zahl<-(1:length(zahl))[zahl==1]
        warten<-c(wo.zahl[1],diff(wo.zahl))-1

        erg<-table(warten)
        plot(erg/sum(erg),bty="n",xlab="Wartezeiten",
                ylab="rel. H\"aufigkeit")
        title(paste("durchschn. Wartezeit auf die 13:",round(mean(warten),2)))

        zahl<-apply(lotto,2,FUN=function(x) sum(x==25))
        wo.zahl<-(1:length(zahl))[zahl==1]
        warten<-c(wo.zahl[1],diff(wo.zahl))-1
        erg<-table(warten)
        if(length(dev.list())<2) win.graph() else dev.set(dev.next())
        plot(erg/sum(erg),bty="n",xlab="Wartezeiten",
                ylab="rel. H\"aufigkeit")
        title(paste("durchschn. Wartezeit auf die 25:",round(mean(warten),2)))
        },{
        zahl<-apply(lotto,2,FUN=function(x) sum(x==13))
        wo.zahl<-(1:length(zahl))[zahl==1]
        warten<-c(wo.zahl[1],diff(wo.zahl))-1
        zus.warten<-(warten[warten>=8]-8)

        erg<-table(zus.warten)
        plot(erg/sum(erg),bty="n",xlab="Wartezeiten",
                ylab="rel. H\"aufigkeit")
        title(paste("durchschn. zus\"atzliche Wartezeit auf die 13:\n",
                        round(mean(zus.warten),2)))
        zahl<-apply(lotto,2,FUN=function(x) sum(x==25))
        wo.zahl<-(1:length(zahl))[zahl==1]
        warten<-c(wo.zahl[1],diff(wo.zahl))-1
        zus.warten<-(warten[warten>=8]-8)

        erg<-table(zus.warten)
        if(length(dev.list())<2) win.graph() else dev.set(dev.next())
        plot(erg/sum(erg),bty="n",xlab="Wartezeiten",
                ylab="rel. H\"aufigkeit")
        title(paste("durchschn. zus\"atzliche Wartezeit auf die 25:\n",
                        round(mean(zus.warten),2)))
        },{
tip<-lese.ein(frage="Bitte einen Lotto Tip eingeben <RETURN>",
                default=c(9,13,24,27,40,44))
tip<-as.numeric(unlist(strsplit(tip," ")))

anz.richtige<-apply(lotto,2,FUN=function(x) sum(!is.na(match(x,tip))))
erg<-table(anz.richtige)
plot(erg/sum(erg),bty="n",ylab="rel. H\"aufigkeit",
        xlab="Anzahl Richtige")

text(names(erg),erg/sum(erg)+0.02*par()$usr[4],erg)
title(c("Tip:\n",paste(tip,collapse=",")))
        if(length(dev.list())<2) win.graph() else dev.set(dev.next())

plot(1:length(anz.richtige),anz.richtige,pch="*",bty="n",xlab="Ziehung")

        if(length(dev.list())<3) win.graph() else dev.set(dev.next())
plot(1:length(anz.richtige),
        cumsum(anz.richtige==0)/(1:length(anz.richtige)),bty="n",type="l",
        ylim=c(0,1),xlim=c(1,length(anz.richtige)*1.05),
```

```
        ylab="rel. Gewinnh\"aufigkeit",xlab="Ziehung")

text(length(anz.richtige)*1.02,sum(anz.richtige==0)/length(anz.richtige),
        "0",cex=.8)
for(i in 1:4){
        lines(1:length(anz.richtige),
        cumsum(anz.richtige==i)/(1:length(anz.richtige)))
        text(length(anz.richtige)*1.02,sum(anz.richtige==i)/length(anz.richtige),
                i,cex=.8)
}


        },{
        set.seed(2)
wieviele<-as.numeric(lese.ein(frage="Wie viele Ziehungen? <RETURN>",default=13000))
lotto.z<-matrix(0,nrow=6,ncol=wieviele)
for (i in 1:wieviele)
    lotto.z[,i]<-sample(1:49,6,replace=F)

par(mfrow=c(1,1))
tip<-lese.ein(frage="Bitte einen Lotto Tip eingeben <RETURN>",
                default=c(9,13,24,27,40,44))
tip<-as.numeric(unlist(strsplit(tip," ")))
anz.richtige<-apply(lotto.z,2,FUN=function(x) sum(!is.na(match(x,tip))))
erg<-table(anz.richtige)
plot(erg/sum(erg),bty="n",ylab="rel. H\"aufigkeit",
        xlab="Anzahl Richtige")

title(c("Tip:\n",paste(tip,collapse=",")))
text(names(erg),erg/sum(erg)+0.02*par()$usr[4],erg)
erg

        if(length(dev.list())<2) win.graph() else dev.set(dev.next())

plot(1:length(anz.richtige),anz.richtige,pch="*",bty="n",xlab="Ziehung")

}
)
cat("lotto.experiment beendet\n"); NULL
}
```

320 ⟨*lotto* 320⟩≡

```
lotto <-
structure(c(13, 41, 3, 23, 12, 16, 12, 32, 30, 49, 3, 18, 34,
12, 14, 24, 36, 23, 30, 13, 36, 44, 23, 4, 31, 44, 6, 49, 5,
39, 29, 44, 6, 37, 22, 18, 27, 44, 8, 16, 12, 4, 26, 37, 9, 44,
46, 24, 47, 20, 31, 26, 21, 5, 10, 26, 37, 8, 31, 22, 26, 8,
42, 39, 36, 14, 34, 38, 42, 39, 9, 5, 38, 22, 46, 4, 27, 36,
37, 6, 41, 22, 21, 42, 29, 1, 19, 9, 49, 16, 23, 40, 35, 12,
17, 27, 18, 5, 22, 6, 38, 43, 26, 42, 43, 12, 35, 36, 14, 1,
23, 27, 33, 30, 26, 35, 38, 20, 42, 8, 30, 42, 1, 48, 32, 45,
41, 6, 4, 47, 17, 19, 6, 28, 9, 26, 20, 22, 26, 10, 3, 49, 24,
38, 42, 16, 18, 3, 37, 19, 25, 10, 32, 7, 43, 22, 32, 39, 19,
25, 2, 30, 46, 32, 14, 31, 12, 1, 19, 33, 21, 10, 24, 35, 18,
4, 40, 36, 47, 34, 45, 39, 41, 49, 3, 1, 43, 31, 39, 32, 4, 34,
22, 12, 37, 31, 6, 42, 36, 33, 38, 43, 35, 5, 36, 34, 35, 11,
33, 17, 23, 40, 14, 46, 10, 16, 25, 41, 5, 46, 38, 19, 37, 18,
43, 32, 14, 16, 44, 13, 35, 21, 25, 5, 47, 3, 43, 20, 48, 42,
48, 12, 19, 17, 34, 23, 38, 48, 5, 10, 18, 8, 27, 35, 9, 17,
23, 10, 40, 9, 23, 25, 31, 46, 10, 31, 4, 11, 26, 47, 5, 34,
33, 14, 6, 40, 36, 3, 16, 29, 25, 11, 49, 48, 40, 31, 26, 14,
42, 47, 15, 28, 27, 22, 42, 23, 17, 8, 4, 18, 31, 9, 5, 22, 21,
6, 9, 10, 39, 36, 27, 24, 33, 44, 29, 36, 9, 6, 32, 10, 11, 25,
12, 42, 41, 22, 1, 2, 48, 36, 17, 5, 49, 39, 29, 47, 38, 7, 16,
19, 39, 33, 29, 36, 7, 34, 49, 14, 37, 40, 39, 28, 16, 14, 16,
36, 44, 25, 6, 9, 15, 9, 47, 10, 27, 21, 32, 11, 13, 7, 43, 6,
17, 20, 23, 15, 31, 35, 49, 45, 13, 47, 1, 5, 23, 31, 17, 11,
26, 38, 20, 4, 16, 19, 17, 15, 6, 47, 7, 33, 3, 12, 2, 39, 6,
9, 28, 49, 12, 6, 39, 30, 43, 34, 31, 49, 18, 9, 17, 19, 1, 31,
42, 35, 9, 15, 15, 16, 28, 43, 23, 2, 36, 9, 34, 4, 11, 45, 2,
36, 45, 7, 35, 18, 3, 14, 35, 24, 44, 32, 18, 48, 22, 45, 31,
14, 47, 20, 30, 44, 27, 26, 2, 40, 31, 14, 48, 18, 39, 18, 6,
28, 44, 15, 35, 44, 11, 39, 27, 45, 18, 43, 23, 37, 20, 33, 31,
41, 4, 2, 27, 22, 15, 21, 31, 17, 24, 40, 28, 16, 44, 30, 20,
29, 24, 35, 3, 45, 40, 1, 4, 32, 37, 14, 3, 29, 7, 47, 1, 32,
8, 35, 21, 38, 14, 44, 34, 30, 10, 12, 19, 4, 25, 8, 23, 31,
15, 36, 37, 46, 26, 11, 5, 22, 6, 14, 41, 44, 40, 34, 28, 30,
42, 1, 2, 23, 44, 3, 9, 25, 34, 20, 1, 27, 48, 17, 42, 8, 43,
6, 13, 39, 29, 33, 36, 26, 3, 40, 48, 8, 41, 14, 36, 31, 24,
30, 42, 38, 27, 47, 7, 40, 2, 4, 27, 38, 21, 12, 20, 40, 24,
37, 19, 28, 34, 12, 25, 10, 33, 9, 35, 7, 21, 22, 13, 26, 42,
20, 9, 26, 25, 31, 11, 6, 49, 1, 14, 41, 29, 30, 35, 36, 24,
2, 27, 42, 22, 4, 18, 9, 23, 2, 49, 26, 46, 45, 3, 11, 24, 45,
46, 16, 30, 20, 20, 24, 49, 34, 8, 22, 23, 25, 3, 26, 1, 14,
46, 36, 9, 27, 26, 11, 45, 26, 3, 32, 47, 43, 27, 11, 22, 7,
39, 18, 33, 22, 28, 23, 4, 7, 46, 14, 34, 45, 47, 36, 36, 15,
21, 44, 10, 6, 19, 36, 9, 37, 23, 22, 30, 32, 12, 34, 14, 3,
7, 4, 23, 41, 6, 27, 34, 46, 4, 48, 38, 39, 30, 45, 40, 43, 25,
37, 36, 37, 8, 45, 15, 21, 19, 2, 26, 3, 17, 12, 24, 9, 11, 27,
48, 21, 11, 20, 49, 5, 10, 36, 18, 6, 2, 8, 17, 35, 39, 20, 27,
5, 22, 43, 30, 38, 46, 28, 22, 34, 44, 36, 29, 49, 24, 23, 16,
10, 29, 13, 28, 47, 27, 25, 5, 3, 22, 49, 1, 21, 38, 30, 25,
9, 31, 10, 48, 32, 38, 15, 2, 7, 32, 36, 25, 38, 49, 26, 18,
34, 41, 21, 37, 29, 3, 8, 38, 6, 25, 33, 24, 43, 30, 31, 34,
8, 32, 46, 48, 9, 46, 13, 33, 30, 32, 24, 17, 24, 26, 32, 30,
37, 9, 8, 36, 27, 10, 15, 28, 33, 6, 31, 46, 39, 19, 18, 17,
36, 21, 49, 4, 29, 44, 3, 49, 30, 42, 48, 41, 47, 17, 35, 46,
48, 24, 10, 14, 18, 14, 9, 22, 36, 26, 34, 1, 39, 20, 17, 29,
10, 18, 28, 35, 45, 42, 37, 45, 19, 21, 14, 31, 32, 47, 13, 21,
```

```
37, 45, 44, 33, 35, 40, 1, 5, 9, 20, 32, 14, 13, 7, 1, 46, 39,
27, 9, 35, 22, 4, 38, 8, 2, 30, 44, 33, 16, 25, 19, 6, 20, 46,
4, 6, 49, 48, 45, 33, 30, 6, 34, 21, 36, 47, 8, 13, 7, 29, 49,
46, 11, 18, 44, 8, 12, 22, 21, 45, 11, 47, 25, 22, 18, 19, 49,
4, 25, 11, 44, 46, 49, 19, 18, 28, 13, 1, 17, 27, 4, 11, 48,
31, 9, 26, 13, 6, 45, 15, 17, 40, 9, 44, 41, 12, 14, 49, 19,
41, 46, 2, 13, 34, 27, 46, 45, 44, 28, 7, 18, 2, 43, 4, 6, 41,
40, 28, 41, 47, 44, 27, 5, 25, 26, 33, 1, 41, 6, 13, 25, 33,
6, 11, 15, 11, 25, 6, 33, 29, 8, 18, 49, 29, 44, 47, 5, 33, 18,
46, 21, 37, 2, 11, 16, 10, 1, 36, 39, 14, 29, 37, 40, 48, 21,
49, 26, 19, 34, 5, 43, 49, 16, 35, 10, 7, 23, 23, 19, 35, 18,
2, 39, 2, 42, 39, 48, 33, 28, 16, 28, 20, 4, 46, 22, 21, 40,
1, 14, 8, 15, 30, 31, 35, 46, 7, 5, 44, 43, 42, 14, 22, 19, 49,
7, 27, 41, 46, 1, 28, 19, 41, 4, 1, 5, 8, 25, 22, 49, 44, 17,
10, 42, 44, 23, 21, 11, 30, 19, 41, 28, 32, 17, 42, 10, 24, 28,
23, 35, 34, 42, 3, 4, 43, 21, 2, 36, 49, 44, 18, 9, 35, 25, 31,
23, 46, 29, 9, 46, 2, 26, 40, 13, 31, 30, 14, 48, 34, 38, 24,
6, 48, 45, 37, 14, 26, 9, 35, 36, 33, 11, 31, 16, 13, 28, 15,
9, 2, 13, 19, 42, 41, 31, 36, 33, 14, 16, 35, 11, 8, 3, 29, 9,
18, 26, 15, 16, 44, 1, 9, 23, 48, 45, 26, 46, 11, 31, 1, 9, 30,
24, 22, 32, 15, 21, 19, 16, 29, 17, 39, 34, 6, 5, 17, 31, 19,
11, 45, 40, 2, 38, 39, 26, 19, 18, 6, 29, 12, 18, 33, 29, 25,
37, 20, 31, 38, 19, 36, 45, 29, 23, 35, 3, 7, 37, 15, 23, 43,
38, 47, 4, 4, 34, 3, 5, 14, 49, 2, 20, 1, 37, 21, 29, 5, 43,
14, 32, 45, 20, 40, 20, 42, 25, 27, 22, 31, 47, 48, 25, 8, 17,
9, 42, 48, 40, 43, 39, 33, 41, 28, 22, 27, 36, 14, 24, 49, 33,
15, 7, 45, 42, 22, 43, 37, 48, 30, 41, 49, 29, 4, 40, 48, 13,
20, 41, 33, 8, 10, 12, 28, 45, 26, 22, 12, 45, 1, 26, 25, 14,
12, 13, 4, 3, 24, 20, 34, 44, 25, 48, 6, 2, 28, 19, 46, 41, 32,
43, 26, 48, 47, 33, 43, 45, 4, 32, 42, 22, 19, 5, 47, 10, 25,
9, 30, 22, 7, 47, 43, 39, 15, 3, 17, 36, 48, 39, 44, 3, 17, 42,
40, 4, 25, 33, 45, 35, 12, 14, 22, 39, 28, 9, 3, 33, 41, 32,
41, 36, 14, 22, 46, 7, 28, 6, 36, 8, 14, 32, 26, 32, 28, 39,
29, 2, 32, 44, 31, 24, 40, 48, 10, 24, 34, 27, 15, 31, 24, 41,
14, 19, 31, 40, 8, 39, 27, 10, 49, 13, 23, 4, 49, 34, 21, 17,
16, 43, 23, 35, 21, 15, 22, 29, 30, 49, 12, 20, 22, 29, 13, 44,
4, 23, 32, 18, 38, 5, 25, 33, 43, 34, 27, 13, 48, 35, 14, 4,
38, 25, 30, 7, 11, 43, 6, 19, 10, 12, 31, 39, 37, 28, 12, 24,
45, 48, 46, 11, 22, 27, 46, 41, 23, 25, 32, 42, 19, 22, 3, 29,
25, 45, 17, 48, 33, 18, 7, 8, 4, 16, 12, 31, 19, 3, 38, 24, 20,
12, 30, 45, 25, 22, 29, 5, 14, 38, 5, 48, 29, 31, 6, 18, 8, 47,
35, 24, 41, 49, 19, 9, 10, 3, 18, 24, 30, 15, 47, 41, 20, 4,
36, 13, 33, 19, 14, 42, 27, 14, 1, 47, 20, 46, 22, 18, 5, 13,
4, 3, 34, 12, 23, 24, 32, 26, 46, 5, 47, 45, 1, 8, 42, 40, 8,
36, 47, 20, 15, 8, 27, 49, 11, 42, 20, 10, 30, 13, 39, 38, 10,
35, 14, 16, 37, 23, 15, 47, 6, 4, 49, 38, 18, 45, 10, 6, 47,
37, 15, 16, 18, 25, 37, 43, 41, 38, 5, 32, 16, 42, 5, 15, 13,
49, 44, 36, 26, 13, 6, 36, 35, 18, 1, 25, 9, 28, 2, 8, 24, 28,
21, 13, 26, 8, 48, 41, 34, 32, 2, 40, 6, 7, 40, 9, 15, 12, 32,
38, 9, 40, 6, 33, 20, 3, 31, 17, 26, 21, 4, 23, 28, 27, 38, 29,
30, 38, 17, 16, 15, 40, 1, 19, 23, 40, 46, 8, 27, 39, 18, 22,
3, 34, 7, 5, 22, 43, 18, 21, 7, 21, 9, 45, 35, 31, 45, 24, 14,
48, 43, 41, 15, 1, 41, 36, 2, 11, 47, 39, 5, 25, 29, 2, 47, 5,
34, 9, 22, 21, 23, 29, 36, 40, 28, 14, 37, 42, 1, 25, 29, 17,
1, 11, 26, 33, 25, 18, 2, 48, 42, 1, 32, 19, 5, 28, 3, 43, 42,
15, 12, 49, 35, 2, 41, 47, 28, 15, 12, 32, 25, 43, 9, 31, 12,
23, 28, 38, 25, 48, 33, 34, 31, 9, 18, 17, 44, 15, 11, 33, 20,
32, 15, 35, 10, 2, 13, 15, 29, 16, 5, 11, 20, 1, 10, 49, 27,
```

```
37, 7, 13, 10, 16, 5, 18, 33, 4, 29, 14, 23, 36, 1, 33, 48, 4,
35, 31, 44, 17, 7, 14, 19, 12, 31, 25, 48, 45, 40, 47, 22, 35,
4, 40, 16, 33, 9, 33, 5, 6, 44, 41, 33, 11, 18, 49, 12, 27, 44,
8, 13, 14, 30, 32, 40, 20, 39, 2, 37, 1, 27, 17, 7, 16, 25, 8,
23, 5, 29, 10, 47, 39, 5, 32, 21, 49, 13, 25, 36, 4, 31, 22,
42, 32, 9, 2, 1, 27, 24, 26, 1, 33, 35, 20, 11, 37, 12, 22, 9,
19, 26, 27, 33, 45, 37, 5, 47, 3, 37, 20, 41, 5, 49, 48, 17,
44, 33, 7, 25, 26, 29, 23, 40, 18, 27, 9, 39, 31, 33, 28, 42,
14, 32, 11, 36, 38, 9, 29, 2, 45, 48, 17, 25, 19, 21, 33, 43,
4, 19, 45, 16, 22, 1, 21, 12, 14, 27, 44, 24, 38, 25, 45, 35,
28, 32, 10, 26, 12, 10, 32, 27, 25, 9, 4, 44, 39, 37, 46, 24,
4, 42, 39, 36, 7, 29, 35, 48, 45, 21, 27, 34, 49, 17, 28, 13,
44, 47, 12, 40, 34, 3, 4, 10, 47, 27, 32, 1, 19, 26, 48, 27,
2, 3, 35, 25, 11, 33, 15, 31, 19, 25, 22, 36, 23, 38, 13, 29,
19, 23, 13, 34, 25, 2, 27, 18, 10, 30, 6, 48, 15, 17, 10, 32,
33, 14, 36, 15, 16, 10, 46, 24, 13, 9, 13, 28, 47, 35, 46, 40,
2, 11, 38, 12, 6, 4, 44, 26, 1, 48, 14, 47, 35, 6, 39, 45, 33,
19, 48, 43, 3, 17, 8, 44, 11, 10, 41, 23, 30, 48, 26, 32, 6,
35, 18, 7, 23, 44, 8, 12, 39, 47, 14, 28, 8, 9, 37, 15, 5, 21,
39, 36, 22, 40, 14, 34, 22, 38, 48, 7, 13, 16, 6, 36, 48, 32,
2, 30, 38, 25, 49, 15, 7, 9, 25, 6, 20, 1, 11, 41, 27, 2, 46,
24, 16, 47, 49, 10, 25, 18, 31, 19, 26, 41, 14, 6, 9, 46, 22,
41, 10, 4, 1, 35, 19, 11, 43, 28, 48, 44, 18, 49, 22, 29, 15,
17, 42, 32, 16, 47, 46, 3, 21, 37, 42, 6, 7, 15, 42, 33, 47,
27, 32, 48, 20, 37, 43, 24, 37, 1, 21, 23, 15, 4, 16, 6, 20,
23, 30, 14, 25, 26, 38, 1, 41, 13, 4, 46, 19, 43, 33, 48, 11,
46, 14, 37, 30, 16, 31, 27, 36, 18, 15, 11, 26, 13, 25, 30, 10,
43, 35, 42, 44, 46, 9, 13, 41, 17, 45, 15, 11, 39, 11, 9, 22,
26, 35, 41, 27, 19, 26, 47, 8, 11, 39, 21, 12, 41, 44, 10, 25,
48, 49, 45, 18, 27, 29, 25, 10, 18, 1, 23, 38, 6, 1, 44, 2, 45,
32, 25, 40, 18, 5, 2, 20, 31, 37, 27, 49, 28, 46, 36, 34, 43,
48, 11, 7, 17, 36, 26, 19, 20, 27, 35, 30, 6, 3, 5, 21, 4, 10,
34, 11, 31, 28, 46, 34, 42, 37, 44, 12, 38, 6, 20, 49, 27, 11,
15, 31, 39, 30, 35, 23, 38, 5, 19, 22, 43, 7, 47, 24, 35, 36,
40, 46, 19, 3, 17, 29, 6, 48, 22, 12, 31, 49, 30, 23, 38, 1,
37, 24, 24, 7, 19, 32, 31, 12, 8, 4, 44, 30, 28, 40, 38, 48,
30, 23, 31, 17, 19, 2, 20, 37, 8, 23, 45, 37, 6, 36, 11, 9, 4,
24, 26, 8, 22, 38, 26, 7, 30, 40, 38, 34, 26, 23, 30, 34, 44,
48, 28, 19, 14, 44, 21, 46, 14, 17, 29, 45, 23, 11, 2, 3, 29,
46, 11, 37, 39, 28, 40, 7, 1, 11, 32, 19, 34, 18, 43, 35, 3,
32, 12, 30, 4, 26, 3, 21, 10, 26, 43, 11, 34, 23, 17, 5, 33,
38, 4, 14, 38, 18, 25, 31, 28, 13, 40, 24, 31, 32, 10, 30, 17,
36, 40, 7, 15, 27, 32, 14, 45, 37, 20, 33, 22, 7, 37, 24, 9,
31, 1, 36, 7, 19, 2, 29, 16, 37, 12, 21, 5, 6, 36, 16, 9, 29,
34, 42, 38, 16, 40, 43, 8, 42, 38, 21, 1, 28, 28, 32, 48, 6,
47, 34, 30, 7, 36, 14, 17, 43, 26, 11, 46, 2, 3, 21, 29, 10,
25, 18, 7, 32, 28, 22, 47, 21, 7, 39, 21, 16, 47, 32, 26, 2,
23, 37, 44, 3, 35, 15, 26, 32, 1, 16, 41, 35, 10, 31, 43, 16,
25, 17, 2, 13, 10, 39, 24, 9, 36, 25, 41, 22, 18, 47, 1, 30,
42, 23, 24, 28, 25, 32, 48, 15, 13, 42, 37, 21, 10, 39, 41, 23,
38, 46, 28, 19, 40, 39, 49, 4, 43, 5, 9, 47, 32, 48, 34, 19,
11, 12, 5, 35, 42, 33, 48, 9, 6, 34, 31, 43, 11, 13, 49, 38,
21, 22, 45, 1, 45, 32, 5, 9, 37, 4, 22, 39, 1, 29, 23, 49, 27,
14, 33, 37, 39, 48, 37, 22, 10, 5, 11, 27, 12, 23, 13, 49, 10,
22, 47, 1, 49, 36, 35, 18, 3, 38, 24, 37, 46, 12, 31, 34, 41,
44, 37, 48, 16, 36, 45, 40, 34, 47, 41, 16, 38, 36, 34, 9, 46,
23, 36, 6, 14, 45, 45, 20, 32, 12, 16, 37, 36, 4, 35, 40, 17,
31, 36, 41, 38, 43, 45, 34, 27, 20, 4, 6, 24, 1, 18, 32, 39,
```

```
42, 43, 21, 42, 26, 41, 9, 29, 22, 35, 25, 46, 47, 20, 8, 2,
46, 42, 34, 23, 38, 1, 34, 17, 44, 33, 6, 14, 19, 29, 18, 39,
16, 22, 30, 40, 1, 36, 14, 41, 18, 32, 21, 2, 26, 24, 23, 38,
9, 33, 3, 22, 33, 39, 46, 9, 5, 17, 32, 31, 5, 40, 11, 33, 2,
45, 8, 23, 4, 36, 17, 10, 30, 28, 49, 30, 39, 29, 46, 15, 20,
16, 15, 47, 37, 21, 35, 23, 33, 43, 7, 13, 25, 35, 8, 15, 4,
10, 12, 27, 9, 18, 47, 17, 7, 46, 38, 18, 13, 31, 24, 29, 34,
32, 42, 26, 10, 29, 23, 49, 13, 8, 5, 33, 42, 3, 13, 47, 37,
46, 36, 28, 49, 6, 9, 35, 8, 18, 30, 27, 15, 18, 38, 42, 33,
14, 27, 3, 49, 4, 19, 43, 34, 39, 45, 9, 33, 43, 29, 13, 37,
36, 9, 34, 16, 46, 34, 15, 28, 6, 27, 26, 4, 16, 23, 2, 30, 2,
1, 48, 3, 5, 47, 22, 42, 14, 30, 1, 27, 34, 45, 49, 23, 10, 31,
37, 36, 31, 29, 49, 4, 47, 41, 12, 37, 21, 11, 45, 23, 2, 39,
31, 27, 31, 11, 4, 24, 9, 29, 38, 33, 1, 15, 29, 5, 47, 23, 12,
6, 27, 15, 21, 20, 32, 3, 29, 16, 3, 47, 49, 24, 38, 37, 8, 12,
19, 17, 21, 25, 47, 33, 17, 9, 10, 46, 46, 38, 7, 9, 25, 15,
19, 32, 20, 40, 27, 4, 13, 31, 44, 8, 41, 17, 32, 22, 44, 19,
31, 48, 32, 48, 3, 5, 12, 25, 21, 6, 43, 3, 12, 4, 38, 44, 39,
2, 16, 17, 35, 36, 21, 13, 45, 46, 37, 25, 19, 31, 13, 21, 42,
8, 43, 31, 37, 38, 4, 15, 8, 44, 48, 17, 11, 34, 33, 38, 17,
2, 24, 12, 22, 32, 15, 34, 36, 22, 35, 31, 6, 9, 32, 44, 2, 15,
24, 39, 26, 9, 6, 18, 48, 46, 2, 5, 23, 47, 40, 18, 43, 44, 14,
46, 35, 6, 17, 36, 6, 48, 20, 45, 42, 21, 44, 24, 8, 27, 1, 14,
12, 13, 48, 33, 32, 42, 41, 10, 15, 48, 11, 48, 23, 20, 33, 29,
5, 48, 25, 35, 21, 36, 47, 30, 6, 24, 20, 39, 5, 46, 37, 20,
9, 10, 28, 8, 40, 19, 39, 36, 49, 38, 25, 30, 5, 36, 47, 8, 5,
39, 29, 32, 39, 8, 2, 19, 1, 26, 42, 46, 34, 33, 19, 41, 44,
7, 47, 39, 18, 3, 34, 29, 9, 24, 30, 45, 10, 24, 7, 44, 29, 16,
46, 28, 30, 44, 34, 29, 38, 8, 19, 14, 41, 43, 40, 5, 30, 22,
27, 44, 21, 29, 2, 39, 37, 6, 41, 15, 28, 26, 39, 25, 15, 30,
28, 26, 6, 20, 26, 40, 36, 19, 42, 41, 34, 39, 8, 43, 35, 21,
8, 19, 31, 39, 1, 4, 9, 22, 20, 49, 36, 5, 45, 35, 49, 17, 30,
5, 37, 40, 45, 9, 31, 36, 31, 8, 23, 45, 4, 48, 6, 19, 3, 22,
12, 2, 4, 20, 45, 5, 33, 27, 17, 18, 21, 24, 33, 34, 19, 38,
46, 28, 33, 36, 36, 42, 37, 15, 28, 21, 39, 11, 22, 41, 49, 43,
5, 10, 26, 33, 47, 12, 37, 32, 41, 12, 10, 4, 38, 40, 7, 1, 23,
49, 34, 27, 3, 40, 24, 23, 38, 45, 30, 10, 47, 5, 40, 18, 25,
12, 48, 5, 33, 12, 43, 42, 6, 21, 34, 16, 28, 27, 49, 21, 14,
38, 48, 28, 21, 24, 1, 44, 14, 26, 16, 35, 16, 23, 10, 45, 21,
14, 34, 13, 28, 33, 31, 35, 4, 44, 35, 40, 28, 20, 42, 13, 6,
17, 44, 1, 35, 29, 42, 37, 39, 44, 46, 25, 42, 11, 18, 43, 27,
13, 16, 1, 49, 33, 45, 8, 27, 29, 22, 18, 43, 33, 42, 17, 9,
30, 12, 5, 41, 17, 10, 11, 29, 5, 45, 34, 8, 49, 21, 40, 38,
18, 39, 41, 32, 15, 21, 10, 16, 19, 2, 43, 20, 33, 40, 41, 4,
32, 31, 18, 21, 12, 47, 6, 5, 2, 22, 7, 4, 10, 44, 17, 36, 31,
39, 6, 35, 7, 21, 18, 11, 7, 34, 42, 26, 8, 22, 47, 45, 29, 25,
3, 12, 7, 44, 3, 33, 49, 42, 27, 3, 22, 32, 16, 47, 27, 23, 45,
19, 49, 9, 19, 36, 39, 5, 42, 21, 22, 4, 44, 23, 7, 48, 7, 20,
42, 29, 27, 48, 21, 31, 24, 14, 34, 2, 18, 26, 45, 43, 1, 48,
34, 44, 12, 22, 11, 6, 12, 49, 29, 18, 16, 19, 23, 7, 33, 42,
38, 18, 44, 30, 41, 8, 2, 49, 30, 26, 29, 21, 39, 43, 5, 34,
20, 29, 18, 39, 35, 32, 46, 2, 49, 42, 2, 20, 37, 18, 3, 30,
45, 49, 17, 21, 44, 23, 40, 48, 43, 37, 7, 16, 18, 31, 8, 43,
3, 45, 1, 20, 13, 30, 34, 49, 12, 19, 30, 9, 33, 7, 49, 22, 24,
1, 32, 4, 9, 10, 30, 43, 19, 40, 32, 41, 3, 2, 30, 14, 26, 21,
2, 45, 41, 8, 9, 3, 38, 34, 13, 23, 2, 13, 5, 19, 22, 40, 38,
13, 21, 22, 48, 22, 25, 12, 43, 44, 38, 36, 7, 35, 16, 15, 32,
21, 3, 36, 20, 15, 9, 36, 26, 43, 3, 19, 45, 35, 2, 49, 20, 37,
```

```
36, 14, 1, 42, 19, 32, 29, 45, 22, 36, 24, 28, 44, 22, 26, 1,
20, 23, 15, 9, 2, 48, 25, 13, 47, 39, 40, 6, 11, 21, 9, 41, 38,
29, 11, 42, 22, 19, 2, 46, 9, 38, 28, 3, 10, 24, 31, 48, 9, 46,
13, 33, 6, 41, 17, 20, 2, 43, 33, 12, 32, 27, 8, 6, 21, 36, 30,
29, 11, 18, 12, 38, 39, 11, 26, 44, 30, 41, 14, 14, 15, 34, 11,
49, 19, 43, 29, 14, 30, 18, 39, 16, 2, 24, 47, 14, 30, 19, 27,
28, 8, 5, 40, 25, 39, 36, 10, 4, 8, 23, 11, 44, 34, 6, 48, 6,
37, 25, 38, 39, 42, 17, 30, 38, 7, 12, 24, 23, 1, 21, 7, 30,
10, 44, 21, 2, 40, 22, 31, 43, 29, 13, 11, 26, 22, 18, 15, 11,
28, 5, 41, 7, 17, 46, 40, 49, 23, 35, 39, 47, 16, 48, 32, 42,
16, 15, 2, 46, 29, 47, 31, 13, 38, 32, 2, 11, 45, 26, 43, 46,
12, 15, 32, 9, 7, 33, 10, 21, 14, 39, 8, 15, 27, 1, 18, 39, 41,
26, 32, 9, 43, 31, 29, 21, 37, 21, 47, 40, 48, 43, 34, 24, 16,
19, 12, 39, 25, 20, 36, 12, 33, 41, 46, 11, 31, 39, 43, 20, 18,
41, 46, 47, 28, 43, 25, 24, 8, 21, 49, 46, 48, 19, 17, 41, 31,
8, 11, 46, 26, 45, 24, 27, 43, 32, 29, 2, 9, 11, 35, 6, 47, 22,
26, 28, 16, 32, 3, 28, 40, 49, 14, 42, 46, 14, 31, 1, 40, 9,
12, 45, 25, 28, 26, 39, 10, 3, 42, 32, 17, 35, 33, 40, 36, 8,
6, 34, 47, 21, 17, 38, 45, 26, 29, 28, 11, 14, 5, 22, 42, 37,
21, 16, 49, 13, 24, 35, 19, 20, 3, 27, 44, 28, 30, 39, 47, 12,
39, 40, 29, 10, 17, 40, 10, 8, 30, 43, 21, 25, 9, 4, 8, 16, 22,
45, 35, 18, 10, 14, 40, 41, 19, 28, 2, 23, 17, 27, 12, 30, 41,
8, 1, 39, 44, 28, 36, 48, 31, 36, 41, 4, 10, 33, 32, 48, 39,
30, 40, 4, 5, 8, 35, 2, 38, 3, 34, 2, 37, 26, 17, 18, 16, 3,
16, 36, 38, 12, 39, 20, 2, 39, 40, 15, 33, 37, 15, 35, 14, 40,
36, 38, 40, 9, 49, 10, 5, 35, 9, 11, 10, 2, 24, 24, 22, 42, 33,
5, 38, 15, 21, 38, 19, 14, 33, 20, 45, 26, 27, 40, 37, 21, 43,
29, 33, 6, 45, 42, 36, 48, 41, 9, 21, 1, 28, 4, 30, 41, 15, 9,
49, 45, 11, 41, 32, 41, 29, 27, 17, 15, 30, 35, 46, 32, 16, 27,
25, 15, 43, 40, 47, 31, 16, 43, 9, 19, 8, 24, 34, 2, 49, 34,
40, 45, 23, 25, 48, 18, 26, 1, 46, 34, 21, 48, 49, 25, 14, 3,
10, 48, 46, 27, 13, 12, 13, 32, 48, 4, 11, 29, 17, 26, 47, 43,
40, 45, 22, 49, 12, 30, 3, 24, 2, 48, 18, 36, 15, 20, 8, 47,
31, 40, 25, 3, 17, 46, 11, 20, 41, 22, 21, 40, 43, 45, 9, 26,
20, 10, 22, 1, 40, 32, 34, 46, 44, 14, 11, 26, 13, 21, 48, 49,
44, 7, 49, 33, 14, 24, 43, 22, 8, 5, 49, 25, 31, 14, 17, 33,
38, 1, 47, 33, 40, 36, 23, 10, 6, 4, 15, 3, 28, 49, 25, 40, 12,
11, 36, 33, 39, 22, 40, 12, 26, 41, 27, 5, 38, 9, 18, 24, 25,
39, 46, 43, 48, 33, 10, 29, 11, 15, 3, 21, 37, 2, 7, 20, 6, 39,
4, 8, 27, 17, 14, 36, 45, 4, 12, 19, 36, 14, 7, 32, 10, 19, 28,
7, 38, 10, 18, 21, 46, 3, 15, 4, 12, 18, 42, 38, 21, 49, 37,
36, 29, 38, 39, 32, 1, 6, 23, 29, 38, 34, 41, 9, 22, 45, 38,
10, 31, 45, 44, 39, 7, 15, 49, 48, 13, 8, 6, 29, 35, 12, 49,
9, 3, 24, 2, 42, 17, 43, 40, 13, 21, 40, 43, 22, 25, 45, 44,
35, 21, 17, 15, 48, 5, 45, 2, 24, 16, 38, 36, 35, 44, 11, 6,
41, 44, 38, 3, 24, 19, 46, 12, 33, 6, 19, 23, 31, 3, 29, 14,
49, 6, 16, 46, 7, 38, 32, 48, 14, 38, 36, 25, 24, 23, 6, 7, 28,
3, 10, 25, 38, 22, 44, 28, 21, 5, 34, 18, 2, 11, 46, 14, 38,
5, 17, 29, 39, 28, 32, 49, 6, 24, 8, 29, 36, 12, 4, 17, 35, 7,
12, 5, 11, 25, 15, 48, 12, 48, 44, 14, 24, 18, 5, 39, 43, 3,
36, 14, 39, 32, 3, 5, 6, 38, 42, 14, 10, 48, 46, 26, 33, 19,
44, 46, 39, 24, 47, 31, 46, 18, 2, 30, 1, 19, 27, 2, 49, 9, 21,
15, 23, 49, 44, 38, 19, 13, 10, 29, 30, 4, 49, 14, 1, 21, 45,
11, 30, 35, 44, 48, 41, 9, 25, 22, 20, 49, 32, 44, 22, 37, 40,
16, 24, 2, 41, 5, 36, 18, 14, 21, 27, 18, 3, 17, 49, 16, 9, 48,
17, 7, 12, 16, 1, 32, 17, 49, 43, 44, 17, 26, 20, 11, 49, 13,
7, 48, 30, 24, 45, 10, 15, 24, 5, 46, 29, 40, 46, 4, 34, 41,
27, 29, 46, 20, 17, 15, 4, 47, 47, 49, 20, 3, 22, 35, 20, 37,
```

```
17, 40, 8, 36, 39, 34, 7, 44, 2, 43, 40, 8, 1, 37, 26, 47, 19,
11, 38, 21, 18, 2, 24, 32, 36, 35, 40, 15, 2, 42, 40, 1, 37,
21, 43, 3, 6, 27, 40, 1, 12, 42, 49, 16, 28, 14, 41, 23, 33,
34, 16, 14, 10, 14, 34, 27, 7, 42, 21, 1, 6, 30, 45, 31, 17,
7, 8, 20, 26, 2, 42, 48, 40, 44, 6, 19, 8, 48, 16, 45, 41, 26,
30, 31, 40, 23, 17, 19, 2, 13, 6, 38, 27, 24, 12, 25, 3, 4, 8,
1, 49, 8, 2, 37, 30, 40, 21, 32, 34, 49, 42, 45, 24, 43, 26,
39, 17, 49, 40, 26, 42, 46, 17, 25, 40, 48, 15, 45, 3, 36, 20,
33, 3, 42, 25, 4, 31, 19, 17, 18, 8, 41, 3, 20, 36, 41, 13, 27,
46, 32, 30, 2, 14, 44, 14, 37, 36, 35, 21, 26, 38, 37, 22, 46,
48, 4, 12, 4, 20, 16, 43, 19, 39, 25, 18, 2, 28, 12, 13, 40,
35, 16, 22, 21, 44, 9, 36, 25, 7, 46, 6, 21, 46, 11, 43, 16,
6, 48, 34, 22, 35, 49, 39, 43, 5, 4, 20, 1, 44, 41, 17, 2, 5,
38, 16, 15, 17, 1, 31, 27, 19, 16, 30, 13, 45, 46, 41, 21, 9,
49, 7, 29, 21, 4, 36, 41, 34, 37, 30, 5, 42, 3, 9, 38, 48, 33,
47, 49, 28, 46, 31, 3, 28, 40, 37, 12, 36, 17, 16, 39, 14, 2,
17, 35, 19, 28, 12, 18, 12, 33, 25, 4, 28, 49, 36, 47, 44, 13,
26, 10, 31, 10, 25, 1, 19, 34, 18, 1, 25, 8, 2, 24, 16, 48, 37,
32, 7, 49, 25, 11, 37, 34, 10, 22, 7, 1, 22, 32, 20, 37, 45,
6, 17, 34, 1, 3, 41, 30, 21, 13, 31, 10, 11, 8, 5, 46, 45, 32,
42, 48, 26, 3, 22, 16, 43, 20, 5, 28, 45, 39, 49, 6, 31, 34,
42, 44, 26, 24, 29, 43, 8, 28, 30, 36, 31, 20, 35, 26, 16, 46,
1, 4, 43, 48, 27, 32, 41, 42, 47, 45, 39, 1, 43, 2, 47, 49, 30,
49, 7, 10, 6, 40, 19, 12, 26, 21, 9, 47, 29, 14, 42, 3, 34, 6,
21, 6, 38, 48, 35, 22, 35, 34, 2, 21, 41, 31, 1, 48, 39, 49,
11, 46, 44, 33, 25, 22, 2, 9, 28, 14, 31, 38, 30, 21, 48, 11,
41, 38, 19, 4, 41, 9, 40, 2, 21, 23, 20, 36, 27, 33, 23, 41,
19, 14, 1, 21, 23, 36, 1, 41, 9, 24, 15, 23, 26, 34, 43, 17,
32, 22, 23, 26, 14, 19, 33, 29, 14, 39, 24, 8, 37, 4, 25, 42,
14, 17, 47, 18, 6, 41, 26, 49, 27, 32, 34, 25, 8, 28, 39, 13,
43, 17, 2, 15, 21, 45, 19, 49, 15, 18, 20, 42, 23, 17, 33, 32,
31, 40, 37, 11, 7, 10, 48, 17, 16, 29, 27, 5, 9, 18, 29, 47,
19, 20, 7, 3, 2, 27, 34, 36, 14, 30, 29, 16, 28, 33, 37, 42,
24, 27, 3, 1, 19, 4, 18, 1, 6, 3, 28, 7, 35, 40, 27, 13, 46,
15, 12, 41, 42, 28, 21, 33, 3, 2, 38, 40, 19, 12, 26, 15, 29,
49, 47, 39, 20, 5, 30, 23, 48, 41, 29, 10, 7, 25, 16, 23, 6,
46, 42, 23, 41, 47, 30, 40, 8, 27, 46, 49, 37, 20, 46, 38, 23,
45, 43, 25, 17, 42, 39, 34, 40, 1, 4, 11, 24, 5, 9, 31, 13, 25,
44, 24, 49, 39, 28, 43, 14, 19, 17, 38, 30, 3, 16, 4, 19, 5,
28, 46, 45, 18, 18, 23, 47, 16, 38, 29, 22, 32, 9, 17, 21, 33,
27, 3, 11, 36, 42, 6, 45, 35, 46, 27, 10, 43, 39, 2, 21, 47,
42, 29, 12, 25, 40, 11, 2, 14, 28, 42, 31, 7, 18, 15, 36, 31,
49, 38, 23, 40, 25, 23, 10, 29, 26, 49, 19, 49, 12, 30, 26, 8,
13, 17, 15, 44, 7, 46, 41, 19, 4, 7, 29, 8, 3, 17, 35, 33, 23,
6, 1, 17, 42, 9, 6, 20, 26, 48, 5, 24, 16, 35, 13, 46, 1, 37,
26, 5, 43, 33, 48, 4, 13, 3, 8, 32, 20, 33, 49, 43, 19, 27, 3,
31, 24, 14, 49, 34, 39, 21, 20, 42, 21, 8, 13, 29, 26, 9, 39,
6, 22, 34, 20, 10, 25, 30, 1, 43, 39, 26, 11, 40, 43, 22, 28,
37, 48, 37, 28, 19, 25, 17, 47, 1, 42, 48, 34, 3, 26, 31, 1,
45, 12, 28, 31, 3, 21, 46, 22, 23, 1, 18, 23, 38, 30, 8, 46,
25, 28, 14, 11, 30, 25, 33, 22, 43, 48, 8, 15, 25, 18, 48, 36,
10, 9, 7, 18, 40, 32, 26, 9, 46, 17, 6, 23, 28, 13, 11, 35, 16,
30, 48, 26, 37, 6, 27, 38, 12, 43, 6, 4, 24, 23, 1, 18, 7, 26,
31, 39, 13, 24, 21, 31, 30, 46, 20, 41, 22, 11, 28, 49, 16, 31,
20, 18, 43, 3, 49, 17, 10, 16, 42, 30, 39, 39, 44, 12, 35, 36,
38, 5, 4, 31, 35, 45, 1, 45, 25, 34, 9, 8, 26, 46, 11, 24, 39,
3, 21, 49, 22, 47, 19, 37, 39, 40, 20, 18, 2, 9, 24, 35, 41,
45, 29, 34, 7, 36, 1, 48, 23, 31, 24, 46, 19, 39, 37, 6, 34,
```

```
8, 25, 33, 31, 15, 32, 15, 33, 9, 40, 21, 20, 17, 27, 11, 45,
34, 20, 19, 33, 31, 17, 49, 3, 41, 7, 39, 48, 11, 2, 24, 40,
42, 39, 1, 25, 20, 30, 43, 33, 5, 37, 21, 32, 3, 16, 39, 34,
33, 32, 4, 31, 39, 36, 31, 11, 24, 25, 1, 45, 20, 32, 21, 24,
25, 2, 17, 38, 8, 37, 31, 22, 22, 37, 13, 15, 28, 48, 44, 4,
41, 29, 21, 15, 12, 26, 44, 35, 7, 14, 1, 20, 38, 12, 39, 30,
14, 10, 17, 32, 19, 31, 26, 46, 2, 3, 33, 25, 8, 41, 7, 3, 20,
15, 49, 22, 17, 8, 3, 2, 30, 40, 38, 32, 20, 39, 31, 8, 26, 7,
5, 40, 45, 12, 18, 17, 13, 32, 37, 4, 7, 24, 20, 47, 21, 24,
48, 29, 4, 22, 39, 7, 49, 32, 1, 47, 41, 42, 47, 45, 30, 23,
27, 31, 24, 26, 45, 38, 12, 26, 3, 45, 33, 6, 19, 6, 26, 44,
45, 15, 41, 48, 43, 37, 36, 20, 48, 29, 11, 35, 28, 10, 41, 16,
38, 48, 6, 44, 36, 25, 27, 21, 22, 16, 24, 39, 20, 4, 16, 3,
20, 8, 33, 5, 30, 49, 13, 37, 49, 30, 43, 28, 1, 40, 36, 2, 5,
21, 35, 2, 25, 26, 22, 7, 2, 26, 33, 41, 1, 27, 20, 39, 29, 31,
19, 27, 12, 45, 2, 16, 29, 21, 10, 7, 42, 19, 4, 12, 47, 31,
30, 27, 36, 38, 36, 45, 6, 12, 21, 47, 31, 17, 11, 44, 14, 30,
3, 8, 22, 27, 15, 23, 13, 32, 3, 26, 37, 24, 17, 37, 31, 39,
12, 36, 46, 4, 6, 3, 23, 16, 16, 24, 44, 49, 35, 3, 22, 5, 1,
28, 27, 24, 31, 33, 21, 15, 22, 13, 46, 31, 20, 12, 48, 42, 45,
38, 9, 35, 6, 39, 32, 27, 47, 2, 45, 29, 22, 46, 48, 43, 28,
29, 41, 42, 12, 32, 27, 28, 40, 29, 32, 8, 22, 19, 14, 27, 32,
22, 10, 49, 48, 6, 14, 9, 42, 45, 47, 20, 10, 22, 17, 9, 42,
23, 6, 22, 7, 46, 41, 24, 28, 3, 40, 45, 13, 38, 43, 1, 42, 37,
33, 14, 4, 1, 27, 30, 23, 42, 1, 32, 48, 6, 26, 18, 33, 36, 20,
47, 7, 47, 27, 30, 37, 25, 49, 40, 18, 34, 28, 4, 1, 38, 33,
4, 20, 14, 38, 49, 32, 12, 19, 3, 23, 20, 8, 2, 49, 41, 42, 21,
32, 18, 7, 23, 24, 12, 15, 47, 49, 11, 29, 9, 4, 7, 16, 26, 10,
5, 13, 16, 31, 22, 45, 10, 12, 40, 35, 19, 33, 39, 19, 1, 4,
38, 27, 35, 7, 37, 33, 18, 23, 31, 6, 34, 40, 25, 27, 7, 33,
14, 23, 36, 10, 11, 35, 46, 9, 27, 17, 28, 30, 43, 12, 25, 38,
43, 12, 31, 10, 49, 31, 18, 35, 22, 45, 33, 39, 31, 12, 7, 24,
30, 31, 26, 35, 25, 38, 19, 1, 3, 13, 11, 44, 9, 3, 38, 25, 29,
30, 9, 26, 15, 17, 29, 42, 9, 47, 4, 2, 13, 49, 26, 48, 28, 36,
40, 39, 3, 36, 49, 28, 19, 48, 17, 33, 25, 38, 15, 9, 49, 17,
2, 32, 45, 3, 46, 31, 43, 27, 32, 7, 3, 8, 38, 32, 39, 27, 12,
7, 16, 32, 46, 13, 40, 35, 42, 2, 19, 18, 5, 44, 32, 4, 12, 30,
14, 16, 17, 33, 40, 26, 31, 35, 18, 15, 7, 41, 44, 8, 1, 43,
10, 45, 6, 30, 17, 47, 40, 32, 37, 32, 8, 38, 40, 27, 21, 3,
32, 27, 34, 33, 16, 12, 44, 35, 39, 11, 4, 13, 44, 16, 7, 26,
46, 16, 4, 5, 44, 18, 24, 19, 14, 20, 43, 2, 32, 22, 8, 14, 36,
2, 5, 39, 35, 31, 37, 46, 29, 37, 3, 23, 7, 44, 25, 13, 45, 25,
2, 35, 46, 21, 25, 9, 37, 44, 11, 28, 39, 16, 11, 3, 49, 9, 47,
46, 10, 41, 29, 22, 6, 45, 38, 41, 13, 24, 4, 29, 43, 34, 32,
3, 1, 7, 41, 32, 46, 13, 30, 11, 18, 15, 10, 17, 23, 20, 9, 24,
31, 30, 8, 44, 32, 48, 40, 10, 1, 26, 38, 46, 13, 12, 45, 41,
1, 9, 30, 32, 26, 29, 43, 8, 4, 29, 44, 17, 25, 23, 42, 40, 2,
41, 13, 11, 10, 21, 48, 8, 12, 15, 5, 41, 42, 23, 36, 20, 1,
21, 41, 5, 46, 23, 19, 29, 26, 48, 45, 42, 20, 28, 9, 10, 1,
49, 46, 45, 5, 43, 2, 22, 44, 37, 17, 48, 43, 9, 18, 12, 15,
32, 17, 41, 25, 20, 6, 1, 45, 46, 33, 5, 11, 47, 29, 13, 35,
23, 41, 34, 7, 2, 47, 24, 25, 20, 35, 7, 19, 15, 3, 47, 17, 32,
14, 6, 32, 15, 33, 12, 5, 38, 46, 26, 9, 5, 12, 25, 5, 20, 26,
41, 27, 43, 31, 26, 44, 42, 28, 40, 35, 16, 33, 21, 5, 2, 39,
6, 30, 16, 26, 11, 7, 37, 26, 19, 20, 8, 43, 4, 6, 9, 29, 17,
19, 6, 30, 28, 16, 49, 12, 9, 20, 37, 11, 32, 40, 6, 24, 3, 11,
46, 32, 44, 28, 23, 9, 39, 3, 48, 16, 37, 47, 19, 35, 24, 39,
6, 9, 47, 2, 33, 1, 23, 34, 15, 40, 30, 38, 8, 22, 38, 2, 43,
```

```
7, 47, 16, 3, 30, 15, 8, 17, 10, 45, 25, 17, 8, 2, 11, 7, 36,
14, 49, 29, 25, 18, 13, 40, 43, 35, 3, 38, 29, 6, 35, 47, 3,
32, 2, 44, 15, 1, 28, 23, 46, 38, 2, 45, 25, 16, 44, 12, 43,
38, 25, 23, 31, 24, 25, 39, 36, 31, 13, 33, 7, 2, 30, 16, 44,
39, 38, 46, 24, 16, 12, 38, 3, 19, 2, 2, 40, 20, 9, 7, 16, 23,
8, 42, 37, 4, 9, 23, 43, 19, 25, 24, 37, 31, 9, 5, 16, 15, 10,
34, 49, 37, 2, 44, 20, 40, 34, 49, 9, 39, 10, 9, 10, 33, 46,
16, 8, 31, 49, 15, 19, 21, 22, 36, 13, 20, 14, 9, 44, 18, 47,
5, 3, 17, 34, 49, 41, 29, 3, 27, 45, 48, 5, 25, 20, 49, 39, 43,
16, 17, 26, 30, 41, 8, 15, 1, 27, 12, 34, 5, 16, 24, 20, 35,
17, 31, 36, 37, 25, 15, 41, 7, 27, 21, 5, 36, 9, 12, 13, 22,
39, 48, 45, 11, 32, 48, 36, 30, 29, 32, 12, 46, 40, 14, 3, 29,
17, 20, 40, 9, 18, 20, 41, 38, 6, 32, 40, 34, 11, 47, 18, 20,
27, 3, 30, 36, 7, 19, 24, 44, 39, 47, 49, 37, 2, 24, 37, 6, 13,
22, 42, 40, 24, 36, 15, 11, 42, 36, 37, 42, 28, 15, 2, 21, 20,
30, 9, 10, 34, 13, 37, 16, 19, 27, 32, 22, 7, 9, 3, 14, 35, 29,
24, 28, 21, 3, 36, 40, 23, 3, 44, 25, 48, 1, 32, 22, 40, 33,
37, 13, 12, 18, 10, 32, 11, 45, 18, 2, 47, 42, 27, 24, 45, 26,
44, 12, 7, 43, 47, 8, 36, 32, 38, 48, 3, 6, 26, 2, 1, 28, 34,
49, 31, 35, 8, 19, 17, 42, 26, 43, 10, 18, 29, 33, 16, 48, 34,
35, 21, 43, 36, 19, 47, 42, 5, 47, 30, 27, 46, 13, 26, 47, 25,
23, 16, 7, 27, 24, 34, 31, 44, 12, 1, 35, 41, 26, 7, 23, 18,
8, 25, 38, 42, 29, 44, 18, 4, 14, 15, 38, 29, 47, 5, 21, 43,
34, 10, 25, 36, 18, 26, 17, 1, 4, 33, 31, 49, 18, 9, 38, 8, 29,
46, 37, 25, 17, 43, 2, 26, 2, 23, 14, 41, 8, 40, 23, 29, 25,
3, 35, 36, 1, 3, 4, 39, 41, 43, 35, 49, 34, 7, 47, 11, 16, 4,
41, 1, 44, 9, 10, 29, 23, 6, 20, 45, 16, 6, 31, 45, 9, 13, 28,
47, 23, 48, 15, 25, 16, 3, 5, 9, 41, 34, 42, 34, 43, 26, 32,
21, 35, 46, 42, 3, 26, 16, 7, 35, 46, 37, 1, 18, 8, 2, 19, 22,
30, 7, 44, 45, 30, 31, 17, 15, 35, 43, 33, 39, 32, 26, 3, 26,
32, 48, 24, 5, 7, 36, 34, 33, 22, 45, 23, 31, 30, 18, 48, 8,
43, 32, 16, 10, 26, 29, 24, 36, 21, 33, 43, 28, 38, 32, 14, 40,
44, 16, 41, 8, 31, 14, 17, 2, 4, 5, 28, 11, 6, 10, 47, 39, 44,
17, 49, 48, 49, 29, 11, 23, 20, 27, 33, 31, 4, 45, 11, 23, 33,
34, 2, 20, 16, 15, 17, 4, 42, 36, 35, 7, 3, 26, 31, 21, 40, 46,
26, 40, 43, 19, 42, 36, 2, 20, 38, 14, 8, 23, 38, 14, 33, 35,
15, 4, 4, 3, 8, 32, 5, 49, 37, 15, 4, 32, 42, 5, 38, 20, 6, 49,
46, 26, 35, 34, 8, 13, 32, 22, 30, 2, 7, 29, 11, 15, 13, 43,
32, 16, 7, 40, 18, 31, 13, 45, 49, 46, 38, 21, 8, 5, 35, 40,
3, 18, 24, 43, 40, 23, 35, 49, 22, 25, 21, 10, 3, 22, 31, 1,
37, 39, 47, 24, 18, 39, 38, 35, 46, 32, 19, 18, 28, 13, 32, 15,
21, 30, 48, 12, 35, 38, 27, 46, 6, 11, 37, 39, 41, 43, 8, 10,
8, 9, 12, 16, 45, 5, 32, 6, 43, 11, 20, 1, 26, 38, 33, 2, 31,
20, 29, 23, 48, 33, 5, 18, 39, 14, 27, 19, 1, 2, 32, 19, 21,
47, 5, 17, 14, 41, 42, 23, 24, 9, 32, 24, 17, 42, 1, 44, 21,
39, 43, 25, 41, 42, 45, 16, 27, 49, 21, 38, 42, 39, 11, 4, 8,
7, 9, 16, 26, 7, 5, 43, 27, 16, 12, 45, 15, 20, 25, 11, 26, 21,
41, 7, 23, 18, 24, 45, 35, 6, 38, 33, 41, 6, 20, 37, 35, 30,
16, 45, 44, 6, 32, 4, 46, 2, 10, 35, 32, 15, 18, 48, 49, 39,
33, 7, 41, 23, 21, 45, 48, 20, 47, 14, 30, 43, 17, 34, 30, 35,
25, 41, 25, 33, 13, 15, 34, 17, 30, 40, 41, 17, 43, 1, 37, 40,
33, 46, 47, 9, 8, 43, 37, 18, 35, 16, 6, 16, 2, 7, 39, 44, 32,
48, 46, 27, 43, 45, 40, 37, 27, 29, 48, 4, 42, 24, 20, 2, 31,
48, 30, 2, 22, 9, 26, 5, 43, 31, 28, 49, 44, 45, 20, 42, 40,
48, 14, 17, 27, 5, 38, 19, 24, 28, 10, 2, 40, 16, 21, 12, 37,
20, 5, 7, 4, 32, 39, 7, 42, 10, 21, 36, 23, 29, 44, 11, 12, 15,
18, 4, 19, 17, 42, 16, 11, 43, 45, 22, 26, 18, 22, 38, 34, 25,
23, 24, 12, 17, 1, 20, 38, 19, 10, 37, 14, 29, 9, 48, 49, 1,
```

```
2, 41, 36, 22, 40, 5, 14, 22, 36, 1, 26, 14, 6, 5, 16, 11, 28,
32, 15, 34, 40, 8, 15, 28, 22, 14, 23, 20, 14, 26, 3, 43, 45,
31, 23, 5, 46, 16, 35, 8, 7, 35, 26, 48, 8, 9, 35, 16, 19, 11,
18, 5, 49, 28, 31, 30, 23, 19, 30, 31, 39, 3, 1, 18, 18, 49,
2, 27, 45, 25, 27, 13, 44, 21, 26, 47, 37, 29, 36, 8, 16, 17,
5, 33, 35, 1, 37, 30, 26, 1, 49, 43, 21, 13, 40, 48, 39, 28,
21, 16, 14, 43, 40, 35, 15, 46, 41, 1, 33, 43, 5, 9, 19, 30,
33, 14, 11, 27, 40, 19, 27, 16, 42, 43, 36, 34, 32, 14, 41, 17,
13, 21, 20, 19, 3, 15, 10, 3, 41, 11, 25, 24, 13, 11, 49, 42,
5, 39, 9, 19, 47, 18, 10, 27, 14, 21, 46, 17, 43, 26, 21, 33,
16, 2, 36, 26, 38, 1, 11, 14, 33, 5, 33, 17, 48, 2, 3, 25, 45,
7, 28, 43, 42, 10, 47, 41, 42, 28, 45, 9, 11, 29, 41, 20, 25,
47, 24, 13, 10, 3, 44, 5, 32, 16, 13, 26, 15, 25, 1, 31, 25,
11, 4, 15, 32, 40, 43, 21, 28, 15, 49, 20, 29, 44, 11, 38, 39,
8, 34, 41, 47, 3, 34, 19, 46, 29, 32, 27, 8, 23, 21, 22, 42,
43, 17, 38, 16, 4, 42, 28, 6, 28, 35, 11, 14, 25, 36, 47, 23,
25, 42, 12, 34, 37, 12, 41, 19, 11, 32, 2, 12, 6, 43, 23, 21,
22, 45, 33, 13, 48, 10, 37, 12, 46, 30, 29, 35, 18, 31, 28, 22,
49, 31, 48, 17, 8, 4, 37, 40, 8, 23, 21, 5, 7, 26, 27, 32, 22,
19, 21, 15, 36, 12, 16, 30, 13, 21, 16, 9, 25, 44, 1, 30, 5,
45, 21, 26, 31, 25, 2, 38, 10, 30, 41, 14, 12, 42, 38, 49, 36,
18, 49, 8, 33, 4, 47, 6, 40, 33, 32, 3, 8, 43, 16, 27, 4, 21,
14, 19, 40, 12, 6, 22, 31, 27, 41, 45, 32, 20, 29, 34, 28, 46,
23, 38, 29, 10, 1, 39, 37, 21, 45, 11, 24, 42, 12, 31, 48, 31,
4, 48, 47, 34, 12, 34, 33, 20, 45, 17, 22, 44, 33, 4, 46, 48,
18, 22, 20, 7, 26, 9, 29, 7, 35, 30, 2, 42, 34, 8, 48, 27, 21,
11, 10, 3, 27, 31, 9, 2, 21, 23, 18, 36, 15, 6, 47, 11, 17, 28,
8, 33, 19, 37, 27, 39, 19, 2, 28, 37, 49, 28, 42, 6, 46, 14,
2, 27, 43, 26, 47, 40, 49, 23, 7, 44, 21, 27, 6, 29, 32, 18,
5, 35, 16, 9, 27, 25, 39, 30, 48, 41, 35, 25, 44, 5, 49, 22,
38, 41, 7, 33, 45, 13, 29, 12, 18, 10, 6, 18, 29, 15, 20, 34,
43, 36, 10, 7, 23, 21, 5, 47, 35, 26, 30, 35, 7, 11, 48, 30,
29, 4, 44, 7, 9, 34, 40, 15, 46, 25, 7, 32, 23, 49, 38, 1, 3,
37, 42, 48, 1, 18, 5, 14, 16, 4, 48, 11, 42, 9, 7, 17, 35, 21,
24, 32, 23, 15, 41, 34, 42, 25, 30, 2, 36, 12, 34, 48, 14, 25,
27, 45, 49, 28, 34, 30, 33, 15, 1, 34, 2, 36, 37, 16, 20, 33,
1, 33, 8, 4, 30, 45, 48, 42, 32, 19, 21, 36, 4, 38, 48, 18, 33,
15, 11, 37, 11, 18, 40, 17, 5, 3, 49, 38, 37, 43, 10, 10, 27,
12, 17, 32, 24, 37, 10, 1, 26, 20, 2, 31, 7, 35, 46, 5, 20, 14,
46, 17, 45, 35, 25, 4, 31, 21, 15, 18, 10, 47, 42, 17, 23, 28,
48, 40, 25, 44, 14, 35, 30, 23, 47, 30, 22, 21, 1, 6, 49, 11,
19, 4, 20, 2, 34, 15, 24, 37, 21, 46, 26, 33, 44, 22, 28, 19,
48, 34, 21, 25, 8, 8, 27, 26, 39, 45, 12, 31, 19, 24, 14, 23,
25, 14, 47, 7, 41, 40, 1, 39, 21, 30, 28, 4, 36, 17, 2, 33, 12,
49, 16, 47, 46, 24, 8, 32, 11, 24, 27, 36, 31, 28, 35, 25, 48,
9, 41, 28, 11, 14, 36, 48, 7, 10, 40, 47, 41, 38, 25, 27, 39,
26, 32, 48, 42, 6, 37, 5, 42, 41, 21, 26, 18, 13, 35, 39, 6,
23, 37, 38, 5, 42, 35, 22, 12, 3, 13, 19, 40, 17, 22, 24, 33,
12, 39, 18, 5, 14, 32, 31, 38, 7, 18, 10, 37, 22, 17, 38, 14,
33, 10, 23, 4, 3, 12, 3, 21, 13, 27, 11, 5, 12, 37, 3, 20, 33,
1, 1, 22, 6, 2, 11, 48, 7, 11, 14, 21, 13, 35, 16, 7, 1, 37,
39, 13, 46, 14, 29, 1, 12, 21, 39, 21, 28, 27, 37, 44, 5, 31,
13, 26, 33, 38, 18, 28, 19, 34, 38, 21, 49, 41, 38, 2, 14, 46,
17, 24, 2, 37, 28, 8, 6, 34, 39, 35, 17, 21, 41, 16, 11, 28,
4, 27, 28, 10, 30, 35, 34, 1, 1, 42, 16, 31, 20, 4, 21, 7, 48,
6, 25, 33, 26, 42, 48, 10, 16, 44, 3, 33, 47, 5, 37, 20, 18,
2, 38, 36, 34, 44, 47, 40, 28, 2, 19, 25, 45, 15, 3, 27, 10,
42, 38, 23, 44, 9, 22, 14, 35, 11, 10, 25, 20, 47, 20, 39, 19,
```

```
27, 9, 30, 48, 13, 28, 29, 32, 6, 22, 36, 2, 45, 17, 13, 27,
3, 46, 30, 44, 38, 27, 18, 15, 33, 35, 31, 46, 17, 24, 36, 11,
25, 20, 22, 16, 21, 10, 44, 48, 7, 29, 12, 6, 9, 5, 16, 41, 20,
6, 28, 4, 34, 37, 24, 2, 42, 38, 4, 10, 1, 48, 9, 10, 43, 19,
6, 47, 30, 38, 47, 2, 19, 44, 43, 17, 44, 13, 33, 6, 19, 12,
39, 22, 26, 29, 3, 36, 5, 17, 13, 44, 48, 18, 39, 37, 26, 20,
27, 1, 32, 44, 31, 15, 29, 37, 48, 22, 34, 9, 45, 42, 1, 27,
26, 11, 7, 46, 15, 39, 4, 5, 32, 40, 8, 35, 42, 18, 46, 6, 44,
10, 32, 12, 15, 30, 12, 34, 39, 32, 4, 38, 47, 33, 2, 39, 18,
6, 9, 21, 49, 44, 38, 44, 5, 43, 13, 20, 42, 32, 1, 35, 2, 23,
20, 9, 43, 24, 11, 17, 35, 9, 44, 1, 35, 15, 38, 46, 40, 8, 38,
26, 36, 48, 2, 27, 16, 37, 21, 5, 33, 4, 48, 32, 31, 24, 34,
8, 27, 2, 14, 32, 11, 44, 26, 21, 33, 37, 39, 46, 8, 48, 41,
3, 16, 48, 44, 27, 46, 49, 38, 25, 41, 42, 8, 47, 15, 45, 32,
31, 14, 41, 47, 18, 49, 3, 9, 49, 36, 29, 8, 47, 10, 18, 3, 41,
47, 27, 19, 24, 4, 37, 17, 35, 13, 31, 41, 46, 27, 45, 6, 25,
34, 8, 49, 41, 2, 23, 30, 13, 7, 11, 19, 1, 16, 20, 33, 11, 15,
27, 33, 20, 47, 3, 19, 7, 9, 12, 46, 20, 33, 30, 21, 13, 6, 1,
37, 46, 44, 19, 29, 21, 9, 47, 24, 30, 13, 15, 43, 27, 2, 23,
30, 46, 48, 16, 2, 3, 40, 11, 38, 2, 30, 26, 17, 33, 16, 49,
33, 9, 42, 37, 15, 28, 43, 38, 20, 19, 3, 7, 36, 11, 8, 15, 1,
31, 32, 47, 40, 15, 17, 45, 17, 27, 21, 9, 2, 22, 40, 19, 46,
6, 41, 33, 32, 10, 23, 27, 15, 24, 28, 46, 3, 7, 16, 31, 12,
22, 7, 25, 3, 37, 32, 39, 44, 10, 36, 49, 33, 38, 15, 2, 14,
41, 46, 19, 3, 39, 7, 1, 31, 30, 22, 6, 36, 32, 21, 44, 25, 45,
28, 28, 13, 42, 9, 32, 47, 45, 47, 6, 24, 32, 22, 15, 17, 44,
6, 35, 23, 31, 21, 43, 8, 13, 14, 43, 5, 32, 48, 29, 11, 20,
16, 21, 1, 47, 32, 37, 36, 24, 30, 26, 9, 18, 12, 5, 37, 47,
33, 7, 35, 6, 40, 1, 2, 47, 24, 23, 26, 4, 31, 13, 38, 32, 17,
29, 11, 35, 1, 27, 21, 41, 34, 23, 32, 36, 8, 43, 10, 41, 14,
7, 1, 5, 45, 39, 36, 27, 47, 22, 37, 25, 36, 22, 35, 6, 2, 43,
30, 22, 6, 10, 4, 46, 18, 10, 48, 35, 26, 40, 20, 33, 47, 3,
10, 16, 10, 4, 35, 14, 46, 18, 8, 16, 5, 47, 21, 37, 31, 39,
12, 29, 45, 11, 15, 38, 49, 22, 19, 1, 17, 41, 25, 4, 6, 3, 11,
33, 5, 42, 13, 40, 48, 47, 5, 22, 11, 4, 38, 40, 28, 43, 2, 31,
48, 21, 7, 42, 5, 38, 5, 48, 12, 14, 44, 5, 23, 46, 32, 9, 12,
47, 36, 15, 34, 29, 12, 20, 40, 45, 14, 1, 22, 9, 16, 44, 7,
38, 49, 33, 34, 32, 19, 40, 49, 4, 42, 30, 33, 10, 13, 11, 36,
19, 12, 21, 49, 7, 19, 9, 10, 43, 45, 13, 21, 48, 17, 10, 46,
33, 17, 23, 42, 40, 37, 36, 37, 6, 3, 49, 38, 26, 29, 39, 49,
36, 2, 18, 15, 17, 24, 44, 31, 43, 23, 7, 20, 13, 17, 21, 34,
27, 10, 40, 37, 39, 11, 1, 38, 14, 37, 49, 48, 21, 44, 41, 40,
38, 39, 29, 34, 1, 20, 10, 42, 17, 43, 16, 31, 6, 15, 27, 7,
34, 20, 8, 17, 2, 11, 19, 39, 32, 33, 29, 14, 5, 23, 14, 47,
40, 6, 1, 24, 13, 46, 30, 5, 1, 41, 42, 7, 34, 5, 46, 1, 37,
12, 22, 5, 23, 33, 36, 48, 17, 30, 16, 14, 42, 46, 17, 47, 14,
45, 26, 6, 22, 19, 35, 36, 40, 28, 19, 13, 34, 22, 43, 26, 20,
10, 18, 46, 39, 6, 13, 18, 38, 3, 32, 3, 4, 16, 22, 11, 45, 44,
38, 27, 29, 1, 33, 37, 49, 28, 17, 32, 30, 22, 41, 24, 14, 45,
45, 23, 47, 30, 12, 28, 16, 46, 6, 7, 48, 24, 9, 7, 49, 29, 46,
42, 8, 4, 15, 39, 36, 5, 10, 4, 11, 20, 15, 25, 3, 32, 42, 10,
31, 2, 2, 15, 30, 22, 16, 35, 20, 33, 39, 26, 40, 4, 38, 1, 21,
30, 23, 2, 13, 29, 39, 22, 46, 20, 5, 9, 18, 19, 11, 48, 30,
2, 42, 26, 38, 24, 1, 21, 3, 24, 13, 49, 18, 3, 7, 43, 38, 26,
35, 14, 34, 7, 33, 45, 24, 19, 38, 12, 9, 37, 34, 5, 48, 3, 7,
16, 38, 45, 37, 30, 31, 48, 35, 15, 12, 49, 10, 34, 13, 19, 43,
25, 15, 38, 28, 38, 39, 15, 46, 44, 42, 24, 9, 34, 2, 5, 3, 14,
20, 36, 19, 46, 1, 48, 10, 15, 11, 38, 39, 8, 31, 48, 38, 15,
```

```
22, 48, 3, 30, 36, 15, 23, 28, 48, 33, 6, 12, 17, 19, 20, 8,
29, 23, 36, 44, 37, 7, 9, 17, 37, 35, 14, 1, 49, 30, 46, 31,
32, 8, 43, 33, 31, 7, 10, 38, 8, 32, 3, 36, 41, 28, 19, 29, 24,
20, 31, 27, 45, 48, 6, 3, 37, 43, 42, 33, 30, 31, 29, 49, 48,
11, 17, 33, 29, 48, 44, 14, 41, 27, 48, 10, 18, 6, 42, 24, 41,
11, 26, 13, 43, 33, 5, 37, 17, 14, 42, 23, 28, 35, 20, 36, 5,
44, 26, 19, 18, 6, 3, 29, 41, 45, 31, 6, 45, 49, 27, 24, 21,
1, 47, 11, 46, 24, 19, 42, 21, 24, 1, 23, 44, 36, 33, 44, 24,
35, 49, 3, 8, 33, 49, 15, 38, 37, 37, 14, 41, 42, 3, 8, 5, 22,
49, 36, 12, 23, 5, 26, 42, 12, 18, 39, 12, 49, 31, 30, 15, 6,
13, 27, 38, 14, 3, 39, 18, 22, 17, 28, 5, 47, 49, 32, 29, 10,
21, 27, 3, 40, 31, 45, 1, 37, 2, 6, 18, 15, 27, 5, 10, 23, 40,
34, 30, 22, 6, 28, 18, 45, 4, 29, 27, 29, 28, 22, 16, 8, 6, 44,
33, 47, 35, 32, 8, 38, 17, 43, 14, 34, 28, 49, 31, 2, 3, 42,
32, 34, 18, 23, 26, 45, 1, 19, 48, 32, 39, 37, 31, 49, 16, 37,
15, 32, 6, 28, 14, 16, 23, 10, 8, 38, 23, 35, 3, 13, 17, 47,
27, 39, 49, 20, 9, 4, 35, 20, 21, 25, 22, 27, 26, 7, 17, 35,
5, 25, 31, 30, 38, 41, 14, 42, 19, 44, 10, 46, 15, 20, 34, 41,
17, 11, 8, 3, 5, 23, 48, 43, 23, 28, 16, 45, 26, 46, 12, 17,
6, 16, 11, 27, 18, 2, 5, 25, 6, 29, 9, 33, 31, 5, 14, 28, 15,
13, 35, 10, 1, 45, 15, 25, 48, 27, 42, 30, 28, 25, 21, 15, 7,
34, 7, 30, 40, 10, 4, 21, 24, 10, 31, 49, 47, 6, 31, 45, 39,
29, 40, 19, 16, 2, 32, 39, 27, 46, 40, 42, 38, 49, 7, 19, 26,
37, 9, 1, 10, 23, 31, 30, 13, 44, 42, 15, 24, 33, 3, 47, 10,
46, 31, 23, 20, 43, 4, 5, 29, 48, 6, 18, 40, 49, 15, 28, 33,
32, 34, 14, 49, 41, 39, 14, 22, 10, 18, 10, 42, 41, 13, 43, 47,
27, 16, 49, 18, 13, 15, 49, 1, 14, 9, 22, 37, 27, 30, 4, 44,
12, 36, 16, 45, 38, 8, 18, 37, 22, 33, 7, 21, 39, 23, 14, 12,
30, 19, 2, 47, 19, 16, 42, 39, 15, 26, 45, 21, 3, 8, 11, 44,
27, 41, 7, 12, 5, 36, 15, 10, 26, 22, 23, 15, 16, 20, 27, 19,
13, 24, 15, 40, 27, 20, 4, 25, 33, 26, 20, 40, 31, 27, 42, 29,
1, 6, 17, 4, 23, 14, 2, 16, 22, 25, 3, 43, 38, 42, 33, 46, 1,
20, 45, 6, 7, 48, 33, 1, 43, 49, 46, 19, 9, 45, 23, 49, 38, 31,
17, 21, 11, 28, 23, 32, 34, 22, 40, 1, 8, 25, 36, 9, 35, 1, 46,
28, 11, 47, 27, 41, 7, 45, 1, 40, 21, 41, 5, 20, 41, 18, 32,
43, 24, 45, 33, 8, 19, 43, 12, 35, 42, 48, 46, 16, 47, 17, 15,
41, 46, 49, 11, 7, 3, 33, 12, 27, 4, 48, 33, 20, 11, 18, 19,
9, 46, 39, 37, 48, 20, 21, 22, 23, 14, 25, 42, 30, 6, 23, 19,
36, 45, 20, 31, 38, 23, 25, 18, 44, 16, 10, 11, 9, 35, 34, 42,
17, 22, 6, 4, 14, 41, 16, 19, 25, 2, 38, 21, 31, 32, 29, 46,
12, 41, 31, 18, 9, 38, 25, 45, 19, 32, 22, 12, 30, 26, 37, 1,
10, 31, 44, 37, 6, 17, 36, 42, 30, 26, 31, 25, 32, 24, 37, 7,
27, 49, 40, 44, 11, 35, 7, 45, 41, 14, 19, 36, 21, 45, 2, 6,
44, 11, 39, 2, 22, 4, 22, 10, 44, 45, 27, 31, 43, 33, 12, 27,
47, 34, 11, 46, 18, 9, 36, 30, 32, 11, 30, 12, 39, 14, 10, 11,
47, 38, 28, 43, 43, 33, 6, 38, 19, 3, 12, 49, 15, 22, 42, 35,
9, 30, 42, 26, 16, 35, 33, 44, 43, 12, 10, 40, 31, 8, 12, 29,
15, 40, 5, 19, 20, 40, 47, 2, 22, 41, 7, 46, 4, 26, 41, 3, 19,
20, 15, 40, 2, 10, 42, 34, 44, 23, 20, 22, 18, 40, 19, 35, 35,
29, 26, 41, 31, 36, 37, 1, 10, 44, 47, 43, 26, 11, 7, 47, 36,
30, 19, 29, 38, 39, 4, 11, 19, 18, 42, 8, 29, 37, 41, 45, 3,
23, 30, 28, 46, 12, 27, 3, 42, 7, 40, 4, 49, 7, 25, 39, 23, 38,
6, 41, 4, 29, 44, 16, 2, 37, 9, 14, 22, 36, 43, 48, 47, 26, 17,
9, 21, 28, 22, 32, 17, 23, 1, 3, 8, 30, 20, 28, 15, 5, 14, 45,
19, 38, 43, 34, 20, 22, 21, 47, 23, 3, 27, 25, 8, 5, 34, 28,
11, 19, 8, 39, 3, 13, 9, 15, 43, 32, 4, 49, 14, 21, 9, 30, 4,
29, 41, 24, 20, 47, 12, 27, 34, 36, 17, 21, 15, 34, 10, 38, 26,
22, 27, 29, 37, 33, 24, 29, 44, 32, 5, 37, 11, 46, 10, 28, 24,
```

```
20, 42, 9, 11, 45, 47, 31, 29, 26, 15, 38, 42, 9, 48, 34, 47,
49, 39, 35, 1, 25, 16, 4, 48, 31, 49, 33, 24, 12, 41, 8, 26,
6, 2, 32, 45, 13, 35, 7, 23, 12, 6, 36, 40, 16, 13, 12, 17, 26,
17, 5, 37, 49, 1, 9, 19, 11, 25, 27, 43, 37, 43, 40, 14, 41,
44, 36, 38, 45, 16, 37, 29, 1, 16, 41, 42, 21, 9, 12, 14, 17,
33, 18, 35, 41, 20, 42, 49, 43, 44, 19, 22, 10, 38, 5, 7, 6,
44, 38, 48, 19, 37, 39, 35, 10, 18, 32, 28, 38, 6, 43, 31, 24,
12, 48, 41, 2, 42, 33, 3, 7, 31, 4, 22, 34, 9, 16, 22, 42, 19,
7, 5, 48, 25, 12, 4, 43, 16, 11, 26, 18, 24, 5, 17, 41, 36, 9,
6, 47, 35, 5, 21, 8, 29, 17, 25, 4, 27, 7, 23, 19, 22, 17, 6,
10, 16, 8, 21, 34, 41, 40, 4, 32, 12, 8, 42, 46, 38, 16, 4, 14,
21, 24, 16, 19, 44, 4, 2, 7, 36, 3, 25, 4, 38, 15, 5, 24, 17,
49, 12, 26, 31, 43, 13, 28, 3, 12, 30, 32, 15, 2, 42, 47, 20,
38, 9, 11, 11, 37, 17, 47, 16, 8, 14, 13, 29, 27, 11, 18, 1,
46, 22, 40, 20, 13, 25, 39, 26, 2, 19, 21, 42, 46, 39, 26, 13,
47, 38, 43, 23, 21, 47, 9, 6, 39, 12, 45, 7, 32, 35, 29, 17,
10, 4, 28, 4, 8, 49, 2, 40, 23, 37, 9, 3, 35, 27, 5, 8, 20, 4,
36, 13, 21, 21, 34, 24, 3, 25, 32, 27, 37, 32, 9, 47, 44, 48,
20, 4, 26, 29, 32, 33, 36, 32, 2, 27, 3, 42, 46, 21, 7, 5, 8,
46, 27, 47, 23, 19, 48, 2, 46, 37, 15, 24, 33, 31, 26, 33, 28,
3, 18, 9, 38, 28, 7, 39, 23, 49, 16, 44, 46, 15, 2, 44, 45, 48,
12, 46, 30, 37, 33, 12, 6, 42, 15, 8, 42, 40, 1, 17, 18, 12,
10, 42, 6, 9, 37, 29, 46, 48, 42, 35, 20, 49, 46, 10, 20, 21,
38, 7, 25, 1, 49, 32, 15, 12, 17, 14, 27, 8, 46, 45, 42, 43,
17, 4, 11, 12, 39, 24, 4, 29, 21, 7, 36, 48, 34, 46, 4, 43, 49,
14, 8, 33, 38, 3, 7, 10, 9, 32, 6, 42, 3, 49, 12, 31, 26, 4,
28, 9, 34, 23, 24, 26, 13, 38, 18, 25, 3, 20, 47, 5, 17, 43,
19, 20, 7, 8, 23, 35, 42, 19, 21, 3, 4, 41, 29, 3, 25, 14, 6,
41, 40, 46, 49, 1, 12, 22, 18, 11, 49, 35, 34, 1, 26, 36, 9,
28, 3, 37, 32, 17, 39, 6, 18, 26, 35, 36, 40, 1, 44, 19, 25,
28, 2, 4, 35, 5, 27, 43, 20, 48, 42, 10, 4, 37, 34, 18, 2, 10,
22, 20, 27, 34, 43, 24, 47, 37, 49, 28, 3, 24, 23, 30, 10, 25,
29, 14, 12, 36, 25, 7, 21, 29, 5, 37, 35, 1, 17, 48, 4, 16, 17,
33, 40, 41, 35, 20, 11, 34, 39, 24, 9, 29, 19, 35, 22, 9, 7,
24, 2, 32, 21, 26, 42, 38, 3, 17, 2, 21, 46, 22, 47, 31, 37,
16, 46, 1, 27, 31, 7, 41, 32, 13, 9, 46, 40, 22, 11, 43, 46,
15, 24, 34, 25, 25, 5, 11, 26, 19, 23, 33, 27, 42, 49, 3, 25,
12, 42, 3, 37, 18, 24, 49, 34, 25, 17, 14, 15, 15, 30, 24, 43,
46, 32, 33, 36, 2, 47, 6, 18, 34, 13, 49, 42, 18, 6, 40, 43,
26, 7, 8, 37, 40, 37, 19, 3, 25, 24, 13, 18, 32, 5, 29, 35, 17,
47, 23, 5, 8, 40, 2, 15, 22, 39, 41, 37, 8, 16, 10, 2, 5, 6,
47, 9, 19, 34, 26, 43, 33, 15, 49, 43, 5, 48, 34, 49, 4, 31,
38, 8, 11, 42, 37, 5, 44, 38, 5, 33, 30, 15, 6, 19, 2, 33, 36,
6, 40, 4, 21, 13, 27, 16, 43, 22, 49, 20, 36, 24, 3, 5, 37, 24,
18, 49, 31, 43, 5, 42, 20, 4, 47, 49, 11, 26, 49, 27, 20, 38,
31, 26, 33, 10, 23, 47, 17, 49, 38, 10, 12, 37, 46, 35, 34, 24,
14, 43, 19, 30, 35, 21, 36, 43, 22, 10, 7, 44, 28, 31, 46, 16,
4, 21, 22, 36, 38, 1, 42, 40, 20, 41, 40, 11, 9, 13, 7, 32, 4,
28, 36, 16, 22, 30, 34, 28, 26, 1, 48, 25, 32, 42, 39, 3, 21,
48, 5, 4, 27, 47, 24, 42, 3, 41, 5, 10, 13, 38, 19, 1, 15, 31,
32, 43, 1, 33, 4, 26, 47, 22, 12, 46, 39, 41, 5, 37, 48, 2, 27,
31, 40, 38, 23, 10, 41, 22, 25, 7, 8, 47, 7, 29, 10, 39, 31,
26, 36, 41, 32, 7, 40, 9, 7, 32, 38, 28, 32, 44, 49, 48, 34,
45, 33, 35, 37, 40, 24, 32, 18, 9, 19, 46, 40, 17, 18, 3, 12,
23, 35, 43, 23, 17, 16, 12, 31, 36, 9, 14, 13, 21, 8, 41, 24,
39, 38, 4, 34, 33, 17, 41, 25, 4, 8, 2, 4, 24, 45, 23, 32, 18,
26, 9, 18, 2, 14, 21, 32, 31, 2, 8, 43, 29, 26, 11, 40, 36, 3,
9, 15, 6, 12, 28, 16, 48, 6, 49, 16, 42, 41, 17, 37, 43, 1, 44,
```

```
39, 47, 49, 32, 30, 45, 12, 8, 46, 17, 43, 45, 35, 36, 27, 11,
18, 16, 32, 39, 45, 39, 1, 3, 21, 18, 4, 32, 9, 48, 31, 14, 31,
40, 20, 41, 22, 47, 14, 35, 47, 30, 22, 16, 41, 16, 25, 40, 6,
37, 41, 32, 48, 17, 35, 4, 46, 4, 3, 39, 40, 11, 30, 11, 19,
27, 13, 25, 42, 9, 31, 15, 28, 43, 45, 36, 2, 18, 16, 46, 19,
21, 20, 47, 14, 35, 1, 42, 36, 37, 5, 21, 1, 5, 27, 3, 17, 30,
11, 29, 49, 37, 14, 31, 49, 10, 41, 40, 24, 31, 22, 25, 39, 5,
4, 15, 29, 39, 2, 46, 5, 22, 19, 32, 3, 4, 15, 41, 7, 44, 17,
1, 28, 49, 9, 21, 38, 36, 26, 29, 21, 33, 19, 32, 8, 36, 36,
11, 26, 29, 24, 6, 43, 17, 2, 35, 23, 46, 27, 41, 2, 46, 28,
44, 14, 19, 42, 3, 15, 24, 39, 16, 32, 12, 35, 28, 24, 5, 19,
1, 48, 16, 36, 37, 25, 20, 8, 30, 48, 1, 5, 45, 35, 18, 15, 10,
6, 25, 41, 17, 10, 5, 26, 25, 9, 45, 11, 43, 14, 35, 37, 12,
5, 45, 39, 32, 40, 36, 25, 47, 49, 10, 41, 12, 12, 32, 8, 38,
35, 30, 14, 25, 19, 20, 33, 15, 5, 42, 46, 3, 35, 24, 18, 28,
12, 19, 37, 13, 8, 7, 18, 19, 32, 49, 33, 41, 42, 11, 22, 12,
26, 39, 16, 24, 45, 28, 7, 45, 18, 22, 5, 16, 5, 46, 24, 34,
14, 48, 19, 28, 27, 12, 48, 15, 46, 17, 18, 48, 19, 34, 15, 3,
43, 33, 38, 28, 40, 34, 7, 38, 18, 49, 45, 28, 4, 12, 5, 17,
9, 5, 12, 19, 28, 21, 10, 5, 11, 8, 18, 31, 15, 17, 43, 39, 2,
7, 1, 16, 15, 21, 24, 32, 30, 22, 11, 7, 42, 1, 3, 18, 25, 11,
10, 1, 21, 14, 18, 29, 45, 2, 42, 30, 8, 18, 15, 9, 6, 42, 25,
20, 32, 4, 11, 18, 6, 24, 19, 39, 36, 24, 49, 10, 9, 23, 22,
14, 4, 38, 7, 9, 17, 30, 12, 18, 32, 6, 37, 10, 34, 41, 40, 7,
21, 12, 49, 44, 34, 10, 6, 8, 4, 31, 10, 15, 44, 26, 16, 46,
22, 9, 30, 49, 44, 5, 28, 24, 36, 27, 32, 18, 23, 22, 34, 49,
26, 15, 18, 17, 34, 6, 30, 13, 2, 1, 12, 37, 44, 31, 21, 9, 8,
48, 31, 30, 26, 22, 42, 6, 1, 29, 28, 7, 29, 14, 40, 24, 26,
30, 25, 33, 5, 43, 20, 41, 43, 22, 10, 34, 18, 35, 24, 37, 30,
33, 8, 7, 49, 41, 11, 14, 6, 8, 3, 27, 29, 32, 36, 18, 41, 17,
15, 40, 27, 26, 13, 4, 11, 15, 46, 47, 32, 6, 46, 25, 18, 41,
2, 3, 21, 22, 44, 6, 20, 25, 17, 46, 41, 13, 44, 20, 35, 15,
38, 17, 44, 31, 6, 8, 29, 9, 20, 14, 5, 26, 41, 30, 26, 19, 2,
25, 39, 14, 1, 26, 44, 8, 27, 29, 23, 47, 5, 22, 14, 37, 40,
24, 36, 12, 32, 42, 6, 34, 39, 30, 25, 27, 34, 49, 5, 27, 35,
37, 6, 14, 20, 34, 46, 12, 45, 12, 31, 36, 47, 25, 28, 27, 20,
18, 12, 1, 19, 48, 29, 39, 49, 42, 21, 3, 46, 38, 1, 26, 10,
37, 21, 4, 31, 35, 24, 6, 5, 22, 8, 2, 24, 9, 39, 29, 49, 26,
36, 33, 30, 9, 48, 27, 36, 30, 39, 18, 47, 2, 18, 34, 32, 22,
29, 44, 38, 15, 49, 42, 41, 5, 43, 10, 32, 35, 19, 13, 32, 22,
39, 43, 41, 13, 30, 13, 3, 17, 44, 37, 48, 10, 37, 20, 28, 46,
5, 10, 34, 29, 18, 21, 40, 16, 17, 2, 34, 39, 12, 31, 2, 39,
15, 24, 44, 7, 39, 9, 48, 31, 46, 32, 4, 49, 11, 7, 42, 37, 29,
46, 7, 19, 14, 24, 32, 42, 36, 11, 44, 40, 27, 4, 25, 33, 37,
49, 19, 48, 21, 32, 39, 47, 37, 3, 25, 9, 6, 17, 21, 46, 1, 24,
35, 11, 16, 44, 17, 41, 17, 44, 36, 2, 29, 14, 6, 47, 8, 16,
33, 20, 39, 11, 8, 4, 17, 38, 17, 39, 47, 44, 26, 6, 19, 2, 38,
49, 16, 46, 46, 27, 32, 25, 15, 48, 9, 29, 49, 16, 4, 39, 26,
20, 45, 13, 42, 44, 10, 1, 2, 35, 30, 29, 25, 8, 35, 22, 49,
3, 17, 36, 28, 27, 18, 25, 2, 22, 15, 28, 24, 35, 5, 45, 7, 48,
20, 40, 42, 49, 18, 35, 40, 20, 20, 42, 39, 27, 44, 7, 40, 16,
17, 43, 26, 3, 41, 4, 33, 36, 40, 27, 9, 24, 38, 10, 5, 8, 30,
17, 16, 5, 42, 24, 42, 18, 34, 3, 45, 37, 37, 27, 15, 48, 41,
14, 47, 23, 44, 6, 7, 16, 38, 1, 32, 8, 28, 9, 3, 44, 12, 35,
36, 48, 5, 7, 41, 18, 22, 38, 46, 19, 45, 31, 27, 22, 15, 13,
33, 23, 29, 7, 9, 22, 24, 6, 30, 33, 41, 29, 31, 19, 36, 48,
4, 47, 46, 3, 6, 18, 13, 28, 10, 3, 23, 14, 48, 32, 27, 36, 3,
23, 5, 13, 43, 3, 8, 2, 35, 37, 19, 24, 38, 36, 1, 13, 43, 16,
```

```
30, 6, 37, 15, 41, 32, 7, 26, 2, 48, 12, 47, 41, 19, 8, 45, 42,
40, 28, 23, 44, 27, 21, 12, 41, 18, 17, 33, 49, 45, 39, 37, 8,
2, 4, 31, 6, 44, 41, 13, 11, 38, 42, 35, 9, 1, 38, 12, 24, 31,
47, 37, 3, 34, 26, 6, 36, 37, 49, 41, 47, 27, 4, 31, 25, 21,
26, 18, 13, 33, 7, 43, 8, 32, 5, 43, 32, 28, 13, 10, 4, 11, 33,
48, 32, 20, 24, 32, 49, 36, 15, 13, 38, 10, 33, 23, 45, 26, 25,
22, 20, 26, 33, 37, 13, 21, 31, 4, 25, 26, 27, 39, 2, 31, 18,
38, 34, 16, 28, 30, 48, 1, 16, 39, 20, 12, 41, 6, 28, 1, 13,
25, 30, 18, 11, 29, 10, 26, 32, 45, 18, 3, 20, 21, 40, 30, 11,
40, 42, 14, 48, 25, 30, 28, 33, 3, 37, 7, 23, 47, 29, 44, 28,
2, 21, 44, 24, 22, 10, 9, 44, 3, 46, 5, 34, 25, 12, 47, 32, 26,
13, 16, 41, 45, 12, 43, 11, 39, 47, 9, 23, 17, 36, 5, 49, 20,
48, 26, 44, 13, 32, 36, 23, 49, 42, 4, 20, 13, 27, 17, 48, 33,
11, 37, 6, 25, 32, 42, 7, 23, 16, 41, 37, 42, 40, 29, 19, 43,
21, 37, 40, 28, 38, 24, 27, 20, 14, 11, 15, 25, 6, 46, 37, 3,
9, 18, 39, 10, 31, 5, 42, 21, 47, 3, 37, 15, 10, 32, 39, 36,
11, 13, 36, 2, 40, 20, 33, 44, 29, 37, 18, 23, 21, 38, 33, 24,
15, 39, 1, 43, 32, 23, 13, 25, 42, 32, 49, 44, 39, 48, 37, 39,
2, 25, 16, 26, 12, 39, 2, 38, 9, 15, 33, 21, 47, 39, 28, 13,
44, 47, 35, 6, 20, 7, 24, 8, 49, 32, 21, 43, 14, 3, 6, 22, 32,
19, 29, 7, 3, 42, 20, 26, 25, 43, 9, 42, 17, 8, 49, 36, 33, 21,
10, 3, 38, 5, 40, 13, 20, 25, 36, 48, 41, 34, 47, 26, 20, 43,
4, 1, 30, 47, 23, 4, 31, 29, 43, 37, 9, 38, 35, 3, 34, 20, 3,
19, 24, 38, 14, 18, 15, 35, 12, 17, 36, 25, 24, 46, 7, 31, 35,
25, 45, 8, 35, 12, 2, 20, 44, 43, 19, 30, 24, 41, 11, 10, 9,
7, 23, 18, 3, 32, 11, 36, 28, 13, 2, 42, 38, 27, 28, 25, 10,
42, 21, 11, 6, 48, 41, 49, 18, 4, 6, 39, 17, 48, 6, 44, 35, 29,
6, 28, 27, 18, 36, 21, 43, 47, 29, 8, 23, 27, 16, 12, 28, 20,
41, 8, 32, 37, 36, 30, 15, 33, 17, 12, 34, 11, 23, 41, 11, 2,
26, 47, 1, 3, 33, 9, 15, 18, 11, 46, 40, 37, 10, 41, 33, 28,
13, 48, 47, 33, 45, 18, 6, 2, 13, 26, 32, 34, 43, 23, 39, 45,
2, 37, 17, 23, 2, 19, 38, 47, 3, 17, 5, 47, 37, 49, 46, 49, 18,
1, 34, 29, 39, 30, 8, 34, 35, 29, 45, 16, 8, 14, 30, 29, 16,
43, 14, 7, 3, 45, 49, 8, 44, 5, 15, 40, 43, 2, 48, 7, 24, 16,
3, 7, 34, 24, 1, 43, 4, 37, 23, 35, 17, 24, 6, 25, 32, 22, 24,
10, 21, 35, 24, 20, 3, 42, 10, 32, 47, 36, 20, 9, 16, 4, 48,
18, 28, 7, 26, 21, 20, 2, 36, 23, 20, 44, 23, 38, 36, 39, 5,
4, 39, 24, 30, 32, 42, 27, 8, 10, 2, 14, 33, 44, 22, 35, 6, 41,
11, 49, 13, 32, 10, 39, 4, 49, 44, 18, 46, 32, 48, 6, 10, 31,
44, 46, 35, 33, 28, 29, 3, 40, 19, 44, 36, 28, 47, 10, 36, 1,
11, 29, 7, 14, 22, 26, 39, 31, 10, 24, 10, 33, 26, 49, 15, 5,
27, 47, 38, 34, 16, 30, 15, 30, 8, 39, 2, 27, 20, 5, 7, 44, 22,
13, 3, 43, 31, 49, 46, 9, 7, 31, 2, 8, 41, 27, 32, 15, 8, 6,
31, 11, 28, 26, 23, 18, 44, 49, 38, 26, 33, 27, 46, 40, 18, 7,
36, 13, 22, 26, 24, 41, 46, 25, 26, 2, 27, 14, 9, 36, 24, 33,
13, 3, 21, 14, 31, 15, 47, 19, 11, 1, 49, 8, 3, 49, 35, 4, 29,
38, 46, 26, 14, 23, 6, 8, 32, 2, 26, 10, 15, 37, 47, 4, 41, 45,
9, 14, 9, 16, 37, 44, 29, 17, 27, 43, 15, 18, 29, 2, 13, 30,
29, 27, 47, 4, 36, 14, 26, 3, 46, 23, 15, 47, 29, 6, 30, 38,
8, 45, 11, 18, 10, 23, 4, 6, 15, 25, 35, 49, 33, 3, 26, 21, 8,
42, 16, 30, 12, 39, 10, 42, 48, 14, 39, 43, 45, 40, 11, 45, 6,
27, 24, 31, 29, 49, 8, 13, 18, 35, 47, 44, 7, 24, 15, 6, 35,
37, 44, 16, 36, 10, 37, 12, 36, 6, 48, 16, 11, 34, 40, 43, 15,
16, 38, 9, 13, 23, 40, 27, 5, 6, 32, 29, 30, 45, 10, 42, 30,
29, 2, 25, 16, 48, 47, 46, 21, 31, 28, 33, 49, 9, 18, 4, 28,
27, 15, 32, 1, 26, 43, 11, 12, 10, 35, 7, 47, 10, 36, 33, 39,
48, 36, 1, 11, 14, 48, 16, 40, 15, 13, 34, 1, 49, 9, 23, 22,
40, 3, 41, 14, 35, 34, 39, 19, 42, 4, 47, 17, 42, 41, 12, 20,
```

```
16, 15, 45, 29, 12, 19, 49, 29, 2, 36, 43, 41, 31, 32, 15, 30,
48, 44, 35, 14, 22, 48, 25, 23, 47, 27, 17, 6, 5, 12, 48, 44,
42, 40, 36, 17, 49, 32, 36, 10, 11, 2, 49, 37, 29, 47, 13, 45,
33, 44, 49, 36, 24, 23, 25, 40, 44, 18, 12, 12, 39, 22, 1, 45,
31, 16, 38, 5, 9, 40, 41, 22, 11, 8, 35, 10, 2, 48, 39, 17, 42,
3, 41, 23, 39, 8, 31, 33, 35, 45, 46, 24, 11, 3, 49, 47, 33,
8, 49, 38, 44, 40, 38, 48, 18, 32, 34, 47, 21, 22, 1, 14, 3,
37, 22, 43, 5, 36, 15, 33, 22, 13, 48, 11, 27, 15, 4, 9, 38,
22, 45, 15, 35, 32, 21, 5, 34, 20, 14, 8, 27, 6, 34, 19, 29,
27, 44, 31, 38, 12, 6, 1, 11, 20, 16, 16, 14, 30, 25, 23, 36,
2, 10, 49, 41, 38, 14, 37, 2, 42, 30, 19, 31, 9, 49, 34, 14,
38, 13, 3, 39, 20, 21, 43, 38, 7, 11, 15, 22, 37, 39, 30, 20,
19, 46, 6, 48, 24, 47, 36, 38, 5, 40, 42, 3, 7, 11, 43, 26, 8,
41, 32, 48, 15, 45, 49, 9, 28, 46, 38, 29, 27, 32, 48, 20, 26,
9, 13, 12, 36, 7, 5, 1, 32, 23, 29, 44, 18, 40, 41, 29, 39, 31,
27, 4, 35, 28, 36, 32, 1, 5, 1, 26, 7, 19, 31, 38, 2, 44, 24,
26, 5, 6, 31, 32, 23, 4, 1, 19, 33, 7, 32, 41, 27, 25, 31, 2,
21, 36, 41, 11, 34, 10, 17, 38, 33, 28, 43, 22, 9, 31, 35, 48,
10, 25, 7, 41, 28, 17, 33, 7, 29, 32, 14, 19, 19, 24, 45, 29,
43, 20, 27, 4, 12, 41, 19, 45, 11, 17, 6, 48, 20, 4, 40, 46,
26, 18, 30, 32, 15, 32, 10, 27, 46, 25, 18, 4, 20, 22, 5, 41,
38, 45, 29, 48, 13, 37, 3, 43, 38, 35, 29, 49, 46, 13, 48, 1,
23, 9, 20, 22, 26, 32, 12, 15, 49, 38, 44, 27, 3, 18, 24, 19,
20, 11, 1, 34, 25, 29, 44, 15, 12, 30, 32, 29, 18, 38, 1, 8,
20, 46, 35, 2, 29, 11, 3, 6, 4, 2, 5, 26, 46, 7, 39, 36, 42,
6, 36, 4, 29, 18, 25, 34, 26, 12, 24, 13, 9, 48, 17, 47, 39,
16, 23, 3, 28, 11, 40, 49, 42, 46, 33, 39, 35, 14, 41, 18, 11,
43, 12, 42, 1, 13, 19, 48, 5, 21, 43, 31, 13, 42, 24, 25, 12,
41, 31, 12, 49, 35, 30, 34, 16, 14, 18, 48, 1, 49, 25, 6, 32,
24, 23, 15, 15, 30, 3, 5, 44, 38, 3, 22, 36, 11, 43, 44, 34,
27, 49, 31, 35, 28, 16, 3, 21, 44, 33, 29, 27, 42, 9, 33, 20,
21, 22, 49, 42, 40, 33, 19, 9, 41, 4, 24, 20, 23, 14, 1, 28,
43, 21, 47, 24, 35, 30, 34, 17, 19, 35, 14, 34, 1, 36, 19, 45,
35, 44, 28, 24, 21, 25, 18, 42, 22, 29, 12, 19, 42, 10, 27, 45,
35, 19, 17, 20, 44, 29, 6, 30, 12, 7, 42, 4, 3, 27, 34, 5, 35,
22, 3, 45, 49, 14, 23, 1, 10, 25, 7, 12, 5, 49, 34, 47, 27, 19,
30, 23, 14, 1, 8, 7, 41, 24, 9, 5, 24, 43, 48, 2, 37, 44, 28,
43, 19, 4, 46, 19, 22, 46, 6, 47, 7, 6, 20, 38, 17, 2, 25, 33,
24, 18, 30, 5, 23, 47, 2, 1, 12, 40, 30, 37, 20, 4, 6, 29, 31,
38, 22, 33, 35, 31, 32, 40, 30, 15, 28, 10, 25, 45, 21, 34, 41,
19, 30, 26, 27, 43, 23, 12, 31, 4, 24, 33, 11, 20, 48, 17, 9,
15, 40, 10, 20, 27, 11, 5, 3, 36, 32, 29, 13, 22, 5, 14, 32,
42, 7, 44, 26, 12, 49, 29, 2, 8, 4, 5, 42, 39, 20, 43, 33, 36,
38, 48, 11, 25, 1, 42, 17, 5, 41, 22, 44, 16, 45, 11, 13, 1,
7, 25, 15, 21, 15, 33, 32, 16, 25, 14, 39, 30, 7, 36, 38, 9,
38, 16, 34, 4, 40, 5, 25, 37, 43, 29, 6, 31, 49, 19, 36, 40,
32, 9, 29, 35, 40, 16, 26, 21, 39, 27, 25, 9, 40, 25, 35, 44,
37, 1, 33, 40, 32, 49, 46, 28, 24, 29, 43, 6, 13, 42, 28, 16,
44, 18, 24, 29, 36, 11, 8, 15, 17, 2, 35, 9, 28, 3, 31, 38, 24,
40, 2, 14, 44, 9, 38, 33, 13, 5, 32, 28, 27, 26, 18, 47, 21,
32, 16, 38, 26, 39, 16, 19, 32, 1, 31, 32, 20, 25, 39, 46, 1,
43, 27, 4, 28, 22, 32, 25, 23, 36, 35, 16, 47, 41, 40, 29, 39,
18, 43, 25, 4, 10, 20, 35, 18, 26, 39, 9, 45, 10, 11, 35, 1,
7, 2, 26, 38, 3, 32, 1, 48, 39, 47, 37, 38, 2, 26, 21, 32, 24,
30, 10, 33, 35, 29, 13, 49, 11, 5, 37, 46, 22, 48, 43, 17, 39,
49, 24, 26, 27, 41, 8, 1, 23, 37, 47, 6, 48, 4, 10, 37, 43, 14,
46, 4, 45, 32, 42, 43, 40, 4, 26, 15, 11, 33, 24, 27, 31, 11,
48, 40, 28, 11, 20, 33, 18, 13, 17, 47, 2, 27, 5, 7, 17, 42,
```

```
12, 8, 37, 26, 46, 28, 31, 37, 4, 6, 49, 48, 1, 28, 3, 47, 40,
38, 43, 13, 2, 12, 28, 12, 15, 3, 37, 35, 41, 39, 42, 44, 34,
32, 16, 12, 49, 13, 33, 21, 29, 21, 48, 4, 47, 23, 41, 9, 34,
43, 46, 42, 15, 21, 25, 38, 10, 48, 44, 49, 5, 25, 41, 33, 45,
36, 30, 49, 11, 9, 10, 2, 15, 44, 14, 28, 37, 38, 3, 9, 33, 36,
9, 21, 2, 24, 42, 11, 48, 18, 2, 27, 25, 7, 3, 14, 7, 35, 33,
38, 41, 42, 49, 36, 44, 29, 46, 1, 7, 29, 28, 31, 15, 11, 34,
1, 10, 9, 13, 42, 2, 22, 8, 43, 8, 35, 13, 23, 39, 34, 6, 15,
2, 31, 48, 49, 17, 4, 25, 32, 36, 22, 42, 18, 6, 40, 35, 29,
16, 12, 14, 27, 43, 6, 39, 40, 22, 11, 37, 42, 18, 6, 23, 31,
38, 22, 33, 21, 25, 38, 44, 37, 18, 27, 6, 23, 2, 22, 10, 35,
29, 47, 17, 6, 14, 34, 24, 28, 22, 2, 26, 7, 44, 12, 29, 6, 30,
15, 32, 41, 28, 2, 27, 35, 6, 25, 1, 38, 28, 12, 40, 2, 6, 4,
38, 33, 27, 19, 26, 12, 33, 28, 12, 36, 32, 48, 23, 20, 8, 28,
27, 41, 42, 41, 25, 6, 9, 30, 38, 25, 2, 15, 10, 9, 3, 8, 40,
11, 42, 37, 27, 18, 37, 22, 23, 4, 26, 7, 42, 5, 24, 3, 2, 14,
16, 20, 26, 15, 2, 4, 40, 38, 30, 1, 26, 42, 40, 20, 30, 7, 33,
10, 40, 37, 2, 41, 29, 1, 33, 20, 4, 35, 27, 19, 40, 39, 31,
44, 18, 47, 25, 49, 48, 12, 44, 11, 8, 48, 46, 2, 43, 24, 49,
27, 29, 12, 37, 11, 30, 13, 25, 3, 2, 46, 20, 16, 8, 42, 11,
1, 42, 12, 47, 26, 47, 3, 27, 31, 18, 36, 45, 31, 7, 10, 41,
21, 1, 48, 29, 33, 8, 9, 23, 22, 38, 47, 49, 7, 47, 28, 4, 20,
30, 49, 38, 40, 43, 44, 5, 4, 7, 44, 39, 25, 2, 27, 18, 47, 15,
17, 10, 48, 43, 4, 8, 38, 37, 44, 24, 17, 7, 46, 5, 23, 18, 36,
29, 27, 9, 33, 11, 49, 39, 42, 6, 7, 42, 8, 6, 21, 10, 1, 38,
32, 39, 34, 47, 49, 16, 29, 14, 43, 46, 40, 43, 38, 9, 44, 34,
15, 7, 31, 22, 5, 1, 19, 7, 48, 18, 3, 36, 11, 11, 38, 6, 28,
34, 46, 36, 23, 5, 16, 39, 37, 8, 7, 49, 10, 18, 27, 29, 44,
30, 25, 6, 5, 19, 45, 6, 39, 13, 16, 2, 43, 1, 21, 36, 30, 3,
34, 24, 36, 11, 31, 39, 8, 10, 41, 40, 26, 18, 27, 26, 44, 7,
5, 26, 9, 10, 31, 34, 13, 20, 4, 7, 5, 29, 44, 3, 27, 25, 19,
34, 16, 39, 40, 41, 21, 43, 35, 29, 17, 16, 37, 18, 19, 23, 27,
39, 8, 30, 20, 8, 11, 14, 36, 48, 37, 29, 10, 12, 40, 6, 28,
30, 33, 26, 39, 11, 1, 17, 6, 16, 35, 7, 12, 42, 28, 21, 43,
34, 1, 43, 30, 16, 34, 15, 41, 30, 17, 34, 5, 7, 44, 38, 24,
32, 27, 43, 1, 44, 24, 14, 7, 3, 13, 32, 44, 16, 34, 43, 12,
11, 9, 10, 7, 43, 47, 34, 24, 6, 40, 8, 9, 13, 3, 22, 6, 17,
38, 22, 41, 12, 19, 48, 11, 35, 42, 24, 17, 13, 27, 24, 19, 17,
44, 6, 18, 48, 38, 32, 9, 12, 43, 17, 41, 6, 49, 35, 39, 13,
18, 47, 37, 9, 6, 38, 43, 36, 47, 44, 20, 8, 32, 33, 38, 48,
25, 4, 18, 30, 12, 6, 24, 8, 22, 49, 31, 23, 35, 18, 45, 14,
10, 40, 30, 39, 12, 46, 45, 14, 17, 47, 7, 33, 46, 42, 13, 45,
36, 47, 28, 17, 44, 14, 22, 43, 46, 16, 18, 9, 3, 1, 19, 47,
42, 27, 32, 34, 40, 14, 13, 10, 31, 4, 46, 3, 16, 45, 26, 17,
31, 8, 24, 41, 32, 5, 14, 48, 23, 9, 20, 13, 38, 11, 7, 22, 10,
44, 37, 35, 1, 33, 3, 28, 46, 14, 15, 38, 24, 26, 12, 21, 29,
44, 2, 37, 34, 38, 41, 5, 2, 21, 48, 38, 23, 19, 17, 34, 23,
47, 26, 16, 28, 14, 9, 13, 47, 2, 8, 10, 45, 46, 39, 5, 47, 28,
21, 4, 25, 44, 18, 24, 28, 15, 30, 18, 31, 7, 36, 37, 6, 21,
12, 2, 15, 1, 3, 12, 2, 1, 31, 34, 11, 12, 45, 38, 17, 4, 14,
34, 31, 15, 2, 7, 6, 33, 7, 25, 19, 49, 42, 15, 8, 16, 4, 26,
18, 11, 18, 25, 22, 19, 42, 34, 11, 26, 47, 3, 27, 44, 17, 40,
35, 7, 39, 30, 49, 16, 19, 25, 11, 47, 33, 7, 35, 28, 31, 4,
35, 37, 32, 41, 15, 27, 24, 35, 49, 25, 38, 32, 18, 4, 22, 43,
21, 16, 47, 44, 41, 31, 27, 38, 22, 4, 33, 36, 19, 5, 39, 8,
31, 37, 25, 31, 8, 4, 49, 28, 48, 41, 33, 24, 20, 35, 43, 29,
44, 11, 5, 30, 26, 30, 24, 31, 37, 43, 2, 36, 24, 35, 7, 17,
40), .Dim = c(6, 2516))
```

321 ⟨*start* 255⟩+≡

```
slider<-function (sl.functions, sl.names, sl.mins, sl.maxs, sl.deltas,
    sl.defaults, but.functions, but.names, no, set.no.value,
    obj.name, obj.value, reset.function, title)
{
    if (!missing(no))
        return(as.numeric(tclvalue(get(paste("slider", no, sep = ""),
            env = slider.env))))
    if (!missing(set.no.value)) {
        try(eval(parse(text = paste("tclvalue(slider", set.no.value[1],
            ")<-", set.no.value[2], sep = "")), env = slider.env))
        return(set.no.value[2])
    }
    if (!exists("slider.env"))
        slider.env «- new.env()
    if (!missing(obj.name)) {
        if (!missing(obj.value))
            assign(obj.name, obj.value, env = slider.env)
        else obj.value <- get(obj.name, env = slider.env)
        return(obj.value)
    }
    if (missing(title))
        title <- "Steuerungsfenster"
    require(tcltk)
    nt <- tktoplevel()
    tkwm.title(nt, title)
    tkwm.geometry(nt, "+0+0")
    if (missing(sl.names))
        sl.names <- NULL
    if (missing(sl.functions))
        sl.functions <- function(...) {
        }
    for (i in seq(sl.names)) {
        eval(parse(text = paste("assign('slider", i, "',tclVar(sl.defaults[i]),env=sl
            sep = "")))
        tkpack(fr <- tkframe(nt))
        lab <- tklabel(fr, text = sl.names[i], width = "25")
        sc <- tkscale(fr, from = sl.mins[i], to = sl.maxs[i],
            showvalue = T, resolution = sl.deltas[i], orient = "horiz")
        tkpack(lab, sc, side = "right")
        assign("sc", sc, env = slider.env)
        eval(parse(text = paste("tkconfigure(sc,variable=slider",
            i, ")", sep = "")), env = slider.env)
        sl.fun <- if (length(sl.functions) > 1)
            sl.functions[[i]]
        else sl.functions
        if (!is.function(sl.fun))
            sl.fun <- eval(parse(text = paste("function(...){",
                sl.fun, "}")))
        tkconfigure(sc, command = sl.fun)
    }
    assign("slider.values.old", sl.defaults, env = slider.env)
    tkpack(f.but <- tkframe(nt), fill = "x")
    tkpack(tkbutton(f.but, text = "Exit", command = function() tkdestroy(nt)),
        side = "right")
    if (missing(reset.function))
        reset.function <- function(...) print("relax")
    if (!is.function(reset.function))
```

```
        reset.function <- eval(parse(text = paste("function(...){",
            reset.function, "}")))
    tkpack(tkbutton(f.but, text = "Reset", command = function() {
        for (i in seq(sl.names)) eval(parse(text = paste("tclvalue(slider",
            i, ")<-", sl.defaults[i], sep = "")), env = slider.env)
        reset.function()
    }), side = "right")
    if (missing(but.names))
        but.names <- NULL
    for (i in seq(but.names)) {
        but.fun <- if (length(but.functions) > 1)
            but.functions[[i]]
        else but.functions
        if (!is.function(but.fun))
            but.fun <- eval(parse(text = paste("function(...){",
                but.fun, "}")))
        tkpack(tkbutton(f.but, text = but.names[i], command = but.fun),
            side = "left")
    }
    invisible(nt)
}
```

322 ⟨*simuliere Erfolgsanzahlen* 322⟩≡

```
# Modell
f.ok<-29/50; n.stpr<-50
# Experimentparameter
wd<-5000; seed<-19; set.seed(seed)
# Umsetzung
stpr<-sample(c(1,0), size=n.stpr*wd, prob=c(f.ok, 1-f.ok),replace=T)
result.tab<-table(result<-apply(matrix(stpr,nrow=n.stpr),2,sum))/wd
# Ergebnisdarstellung
plot(as.character(names(result.tab)), result.tab,
     xlab="Erfolgsanzahl",ylab="relative H\344aufigkeit",type="h")
title(paste("Erfolgsanzahlen bei Stichprobenumfang",n.stpr,"mit",
    "Erfolgschance",f.ok,"\n",wd,"Wiederholungen mit Zufallsstart", seed))
summary(result)
```

323 ⟨*start* 255⟩+≡

```
erfolge.bei.bernoulli.experimenten<-function(){
  refresh.code<-function(...){
  # Modellierung
     prozent.ok<-slider(no=1); f.ok<-prozent.ok/100; h.ok<-21000*f.ok
     gg<-c(rep(1,h.ok),rep(0,21000-h.ok))
  # Experimentparameter
     n.stpr<-slider(no=2); wd<-slider(no=3); set.seed(zz<-slider(no=4))
  # Umsetzung
     result<-unlist(lapply(1:wd,function(x) { stpr<-sample(gg,size=n.stpr); sum(stpr)
  # Ergebnisdarstellung
     f.anz<-table(result)/wd
     plot(as.character(names(f.anz)),f.anz,
        xlab="Erfolgsanzahl",ylab="relative H\344aufigkeit",
        xlim=c(0,n.stpr),ylim=c(0,1.5*dbinom(round(n.stpr*.5),n.stpr,.5)),
        main=paste("H\344ufigkeiten verschiedener Erfolgsanzahlen\n","% Erfolge in GG=
                          prozent.ok,", n.stpr=",n.stpr,", WD=",wd,", Zufall=",zz,sep
    print(summary(result))
  }
  slider(refresh.code,
       c("GG: Erfolgsprozentsatz","Stichprobenumfang","Experiment-Wiederholungen","Zu
       c(1,10,100,1),c(99,100,3000,30),c(1,5,200,1),c(58,50,500,13)  )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

324 ⟨*start* 255⟩+≡

```
binomial.experiment<-function(){
  refresh.code<-function(...){
  # Modellierung
     prozent.ok<-slider(no=1); n<-slider(no=2); s0<-slider(no=3)
     p<-prozent.ok/100
  # Experimentparameter
     wd<-slider(no=4); set.seed(zz<-slider(no=5))
  # Umsetzung
     stpr<-sample(c(1,0), size=n*wd, prob=c(p, 1-p),replace=T)
     result.tab<-table(result<-apply(matrix(stpr,nrow=n),2,sum))/wd
     s<-0:n;f.s<-dbinom(s,n,p);F.s<-pbinom(s,n,p)
     f.s0<-dbinom(s0,n,p);F.s0<-pbinom(s0,n,p); F.quer.s0<-1-pbinom(s0,n,p)
  # Ergebnisdarstellung
     par(mfrow=c(2,1))
     plot(as.character(names(result.tab)),result.tab,
        xlab="s",ylab="f(s),  f.dach(s)",  xlim=c(0,n),
        # ylim=c(0,2.0*dbinom(round(n*.5),n,.5)),
        col="blue",
        lty=2,type="p")
        points(s,f.s,type="h")
        nice<-function(x)paste(floor(x),".",substring(round(10000*(1+x)),2,5),sep="")
        x<-0; y<-0.90*par()$usr[4]; if(p<0.5){ x<-n/2 }
        text(x,y,paste("P(S=",s0,")=",nice(f.s0),sep=""),pos=4)
        lines(c(s0,s0,0),c(0,f.s0,f.s0),lty=2,col="red")
        x<-as.character(names(result.tab));y<-cumsum(result.tab)
     plot(x,y, xlab="s",ylab="F(s), F.dach(s)",
                  xlim=c(0,n),ylim=c(0,1.1),col="blue",type="p",lty=2)
        points(c(0,s,n+1),c(0,F.s,1),type="s")
        text(n/2,1.05,"o  Simulationsergebnis",col="blue")
        x<-0; y<-0.95; if(p<0.5){ x<-n/2; y<-0.15 }
        text(x,y,paste("P(S<=",s0,")=",nice(F.s0),sep=""),pos=4)
        text(x,y-0.1,paste("P(S> ",s0,")=",nice(F.quer.s0),sep=""),pos=4)
        lines(c(s0,s0,0),c(0,F.s0,F.s0),lty=2,col="red")
    par(mfrow=c(1,1))
    title(paste("Binomialverteilung,",
          "p*100=", prozent.ok,", n=",n,"\nSimulation: WD=",wd,", Zufall=",zz,sep=""))
  }
  slider(refresh.code,
       c("p*100   Erfolgsprozentsatz","n   Experimentanzahl","s0  spezielle Realisation"
          "wd   Simulationsanzahl","Zufallsstart"),
       c(1,1,0,100,1),c(99,100,100,3000,30),c(1,1,1,100,1),c(50,10,5,50,13)  )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

325 ⟨*start* 255⟩+≡

```
binomial.zentrales.schwankungsintervall<-function(){
  refresh.code<-function(...){
  # Vorbereitung
     pp<-slider(no=1)/100; nn<-slider(no=2); kk<-slider(no=3)
     par(mfrow=c(1,3))
     ZSI<-function(n,p,k){ EW<-n*p; sigma<-(EW*(1-p))^0.5
                           ug<-ceiling(EW-k*sigma); og<-floor(EW+k*sigma)
                           pbinom(og,n,p)-pbinom(ug-1,n,p) }
  # p freie Variable
     p<-seq(0,1,length=50)
     Pp<-unlist(sapply(p,function(pp) ZSI(nn,pp,kk)))
     plot(p,Pp,type="l",ylim=0:1)
     text(mean(p),0.05,paste("n=",nn,", k=",kk,", p variabel",sep=""))
  # n freie Variable
     n<-1:100
     Pn<-unlist(sapply(n,function(nn)ZSI(nn,pp,kk)))
     plot(n,Pn,type="l",ylim=0:1)
           text(mean(n),0.05,paste("p=",pp,", k=",kk,", n variabel",sep=""))
  # k freie Variable
     k<-seq(0.5,4,length=50)
     Pk<-unlist(sapply(k,function(kk)ZSI(nn,pp,kk)))
     plot(k,Pk,type="l",ylim=0:1)
           text(mean(k),0.05,paste("n=",nn,", p=",pp,", k variabel",sep=""))
           par(mfrow=c(1,1)); title("Binomialverteilung: P(-k*sigma <= S_n <= k*sigma
  }
  slider(refresh.code,
       c("p*100   Erfolgsprozentsatz","n   Experimentanzahl","k  Intervallbreite"),
       c(1,1,0.5),c(99,200,5),c(1,1,0.1),c(50,50,1)  )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

326 ⟨*start* 255⟩+≡

```
p.est<-function(){
  refresh.code<-function(...){
  # Vorbereitung
    p<-slider(no=1)/100; n<-slider(no=2); kk<-slider(no=3); zz<-slider(no=4)
    set.seed(zz)
    x<-1:n; res<-rbinom(n,1,p)
    f.x<-cumsum(res)/x
    k<-1/(1-kk)^0.5
    delta<-k*0.5/sqrt(1:n)
    ug<-f.x-delta; og<-f.x+delta
 # plot
    plot(x,f.x,xlim=c(1,n),ylim=0:1,xlab="n",ylab="Intervall",type="l",log="x")
    title(paste("p mit Mittel einfangen\np=",p,", Sicherheit=",kk,", Zufall=",zz,sep=""))
    abline(h=p,col="red")
    lines(x,ug); lines(x,og)
  }
  slider(refresh.code,
       c("p*100   Erfolgsprozentsatz","n   Experimentumfang","1-1/k^2  Sicherheit",
         "Zufallsstart"),
       c(1,1,0.01,1),c(99,5001,.99,999),c(1,50,0.01,1),c(50,2000,.9,7)  )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

327 ⟨*start* 255⟩+≡

```
hyper.to.binom<-function(){
  refresh.code<-function(...){
  # Vorbereitung
    m<-slider(no=1); n<-slider(no=2); k<-slider(no=3); x.0<-slider(no=4)
    if(k>(m+n)) k<-slider(set.no.value=c(3,m+n))
 # Berechnung
    x<-0:k;     f.hyper<-dhyper(x,m,n,k)
    p<-m/(m+n); f.binom<-dbinom(x,k,p)
  # Plot
   if(is.nan(f.hyper[1])){
     plot(1,type="n");text(1,1,"nicht definiert")
   }else{
    par(mfrow=c(3,1))
    plot(x,f.hyper,type="h",bty="l",
         main=paste("hypergeometrisch: m=",m,",n=",n,",k=",k,sep=""))
    abline(v=x.0, col="red", lty=2)
    ymax<-par()$usr[4]
    nice<-function(x)paste(floor(x),".",substring(round(10000*(1+x)),
                           2,5),sep="")
    if(m>=n){pos<-4;xt<-0}else{pos<-2;xt<-k}
         text(xt,ymax*0.9,pos=pos,
         paste("P(X= ",x.0,")=",nice(dhyper(x.0,m,n,k)),sep=""))
         text(xt,ymax*0.8,pos=pos,
         paste("P(X<=",x.0,")=",nice(phyper(x.0,m,n,k)),sep=""))
         text(xt,ymax*0.7,pos=pos,
         paste("P(X> ",x.0,")=",nice(1-phyper(x.0,m,n,k)),sep=""))
    plot(x,f.binom,type="h",bty="l",ylim=c(0,ymax),
         main=paste("binomial: n=",k,",p=",nice(p),sep=""))
    plot(x,f.hyper-f.binom,type="h",bty="l",main="Unterschiede")
    abline(h=0)
    par(mfrow=c(1,1))
   }
  }
  slider(refresh.code,
       c("m weisse Kugeln","n schwarze Kugeln","k Stichprobenumfang",
         "x.0 spezielle Stelle"),
       c(1,1,1,1),c(99,99,99,99),c(1,1,1,1),c(10,10,10,5)  )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

328 ⟨*start* 255⟩+≡

```
tab.compute<-function(tab,theo=F,chisquare=F,K=F,K.norm=F){
   if(min(tab)<0) return(0)
   h.j<-apply(tab,2,sum); hi.<-apply(tab,1,sum)
   h.theo<-outer(hi.,h.j,"*")/(n<-sum(hi.))
   if(theo) return(h.theo)
   chiq<-sum((tab-h.theo)^2/h.theo)
   if(chisquare) return(chiq)
   K<-(chiq/(n+chiq))^0.5
   if(K) return(K)
   if(K.norm) return(K<-K/(2/3))
   tab
}
mchi<-function(tab){
   h.1<-sum(tab[,1])
   hi.<-tab[,1]+tab[,2]
   tab.mat<-cbind(0:h.1,h.1:0,hi.[1]-(0:h.1),hi.[2]-(h.1:0))
   # tab.mat<-tab.mat[apply(tab.mat>=0,1,all),]
   k<-length(tab.mat[,1]); ws<-result<-1:k
   for(i in seq(result)){
     tab<-matrix(tab.mat[i,],2,2)
     result[i]<-tab.compute(tab,chisquare=T)
   }
   ws<-dhyper(tab.mat[,1],sum(tab[1,]),sum(tab[2,]),sum(tab[,1]))
   cumws<-phyper(tab.mat[,1],sum(tab[1,]),sum(tab[2,]),sum(tab[,1]))
   cbind(tab.mat[,1],result,ws)
}
KT.hyper<-function(initial.m.n.k.n11=c(10,10,10,5)){
  refresh.code<-function(...){
  # Vorbereitung
    m<-slider(no=1); n<-slider(no=2); k<-slider(no=3)
    n.11<-slider(no=4)
  # check
    if(k>(m+n)) k<-slider(set.no.value=c(3,m+n))
    if(n.11>k) n.11<-slider(set.no.value=c(4,k))
    if(n.11>m) n.11<-slider(set.no.value=c(4,m))
   if(n.11<k-n) n.11<-slider(set.no.value=c(4,k-n))
  # berechnung
    tab<-matrix(c(n.11,k-n.11,m-n.11,n-(k-n.11)),2,2)
    chi.n.11<-tab.compute(tab,chisquare=T)
    vertlg<-mchi(tab)
  # Plot
    par(mfrow=2:1)
    plot(vertlg[,1],vertlg[,3],type="h",bty="l",xlab="n.11",ylab="f(n.11)",
         main=paste("hypergeometrisch: m=",m,",n=",n,",k=",k,sep=""))
    abline(v=n.11,col="red",lty=2)
    ind<-order(vertlg[,2]); x<-vertlg[ind,2]; y<-cumsum(vertlg[ind,3])
    plot(c(0,x),c(0,y),type="s",bty="l",main="Verteilung Chisquare",
         xlab="chisquare",ylab="F.chisquare")
    vvv<-rev(y[chi.n.11>=x])[1] # phyper(n.11,m,n,k)
    segments(chi.n.11,0,chi.n.11,vvv,col="red",lty=2)
    segments(chi.n.11,vvv,0,vvv,col="red",lty=2)
       xymax<-par()$usr[c(2,4)];x<-xymax[1]*0.7
      text(x,0.5,"STPR"); text(x,0.4,n.11); text(x,0.3,k-n.11);
      text(x,0.2,k); x<-x*0.8/0.7
      text(x,0.5,"Rest"); text(x,0.4,m-n.11); text(x,0.3,n-(k-n.11));
      text(x,0.2,n+m-k); x<-x*0.9/0.8
      text(x,0.5,"GG"); text(x,0.4,m); text(x,0.3,n); text(x,0.2,m+n)
```

```
    par(mfrow=c(1,1))
  }
  slider(refresh.code,
       c("m Summe erste Zeile","n Summe zweite Zeile",
         "k Summe erste Spalte","n.11 Eintrag oben links"),
       c(1,1,1,1),c(99,99,99,99),c(1,1,1,1),initial.m.n.k.n11  )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

329 ⟨*start* 255⟩+≡

```
binom.to.poisson<-function(){
  refresh.code<-function(...){
  # Vorbereitung
    n<-slider(no=1); lambda<-slider(no=2); x.0<-slider(no=3)
    if(lambda>n)n<-slider(set.no.value=c(1,ceiling(lambda)))
 # Berechnung
    p<-lambda/n
    x<-max(0,round(n*p-3*lambda^0.5)):round(n*p+3*lambda^0.5);
    f.poisson<-dpois(x,lambda); f.binom<-dbinom(x,n,p)
    if(x.0<x[1])x.0<-slider(set.no.value=c(3,x[1]))
    if(x.0>max(x))x.0<-slider(set.no.value=c(3,max(x)))
  # Plot
    nice<-function(x,st=4)paste(floor(x),".",substring(round(1E4*(1+x%%1)),2,1+st),se
    par(mfrow=c(3,1))
    plot(x,f.poisson,type="h",bty="l",main=paste("poisson: lambda=",nice(lambda,1),se
    abline(v=x.0, col="red", lty=2)
    ymax<-par()$usr[4]; xdel<-diff(par()$usr[1:2])
    if(lambda>5){pos<-4;xt<-par()$usr[1]+xdel/30}else{pos<-2;xt<-par()$usr[2]-xdel/6}
    text(xt,ymax*0.9,pos=pos,paste("P(X= ",x.0,")=",nice(dpois(x.0,lambda)),sep=""))
    text(xt,ymax*0.8,pos=pos,paste("P(X<=",x.0,")=",nice(ppois(x.0,lambda)),sep=""))
    text(xt,ymax*0.7,pos=pos,paste("P(X> ",x.0,")=",nice(1-ppois(x.0,lambda)),sep="")
    plot(x,f.binom,type="h",bty="l",ylim=c(0,ymax),
         main=paste("binomial: n=",n,",p=",nice(p),sep=""))
    plot(x,f.poisson-f.binom,type="h",bty="l",main="Unterschiede")
    abline(h=0)
    par(mfrow=c(1,1))
  }
  slider(refresh.code,
       c("n: Anzahl Experimente",
          "lambda: E(X) aus Poisson","x.0: spezielle Stelle"),
       c(1,1,1),c(10000,99,99),c(1,.1,1),c(100,10,5)  )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

330 ⟨*start* 255⟩+≡

```
sum.zv<-function(){
  refresh.code<-function(...){
  # Vorbereitung
    n<-slider(no=1); k<-slider(no=2); zz<-slider(no=3); wd<-slider(no=4)
  # Berechnung
    set.seed(zz); result<-apply(matrix(runif(n*wd),n,wd),2,sum)
  # Plot
    par(mfrow=c(2,1))
    plot(sort(result),seq(result)/length(result),type="s",
         xlab="Summe",ylab="F.dach.summe")
    title(paste("Summation",n,"gleichverteilter ZV,",wd,"Wiederholungen"))
    ds<-density(result,width=0.02*k*diff(range(result))/wd^0.3)
    plot(ds,type="l",main="Dichtesch\344tzung",ylab="f.dach.summe")
    par(mfrow=c(1,1))
  }
  slider(refresh.code,
         c("n: Anzahl Zufallsvariablen","k: Glattheit",
           "zz: Zufallsstart","wd: Anzahl der Wiederholungen"),
         c(1,1,1,100),c(50,99,999,10000),c(1,1,1,100),c(1,50,16,100)  )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

331 ⟨*start* 255⟩+≡

```
geo.to.exp<-function(){
  refresh.code<-function(...){
  # Vorbereitung
    n<-slider(no=1); p<-slider(no=2)/100; x.0<-slider(no=3)
  # Berechnung
     pn<-p/n; x<-1:(1/pn*3);xn<-x/n; lambda<-pn
     f.geo<-dgeom(x-1,pn); f.exp<-dexp(x-1,lambda)
     F.geo<-pgeom(x-1,pn); F.exp<-pexp(x-1,lambda)
     if(x.0>max(xn)) x.0<-slider(set.no.value=c(3,ceiling(max(xn))))
# Plot
    par(mfrow=c(2,1))
    plot(xn,f.geo,type="h",bty="l",xlim=c(0,max(xn)),
        xlab="Untersuchungszeitpunkte",ylab="f.geo | f.exp",
         main=paste("Anzahl Versuche, p=",round(p,2),
         ",\nTeilungen n=",n,", lambda/n=",round(lambda*n,2),sep=""))
    lines(xn-xn[1],f.exp,lty=2); abline(v=x.0,col="red",lty=2)

    plot(c(0,xn),c(0,F.geo),type="s",bty="l",
        xlab="Untersuchungszeitpunkte",ylab="F.geo | F.exp",
        xlim=c(0,max(xn)),ylim=0:1)
    lines(xn-xn[1],F.exp,lty=2); abline(v=x.0,col="red",lty=2)
    xt<-0.7*par()$usr[2]
    text(xt,0.4,pos=4,paste("P(X.exp<=",x.0,")=",
                            round(pexp(x.0,lambda),4),sep=""))
  par(mfrow=c(1,1))
  }
  slider(refresh.code,
       c("n: Anzahl Teilungen",
         "p: Event-WS in %",
          "x.0: x-Wert"),
       c(1,1,.02),c(25,99,300),c(1,1,.2),c(1,50,2)  )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

332 ⟨*start* 255⟩+≡

```
binom.norm.pois<-function(){
  refresh.code<-function(...){
  # Vorbereitung
    n<-slider(no=1); p<-slider(no=2)/100; m<-slider(no=3)
  # Berechnung
    p.werte<-((1:m)-0.5)/m
    xnorm<-qnorm(p.werte)
    xpois<-qpois(p.werte,n*p)
    xbinom<-qbinom(p.werte,n,p)
  # Plot
    nice<-function(x,st=4)paste(floor(x),".",
          substring(round(10000*(1+x%%1)),2,1+st),sep="")
    par(mfrow=c(1,3))
    plot(xnorm,xbinom,type="p",xlab="q.norm",ylab="q.binom",
        main=paste("Binomial gegen Normal\nn=",n,
                   ", p=",nice(p,2),sep=""))
    abline(n*p,(n*p*(1-p))^0.5)
    plot(xnorm,xpois,type="p",xlab="q.norm",ylab="q.pois",
        main=paste("Poisson gegen Normal\nlambda=",nice(n*p,2),sep=""))
    abline(n*p,(n*p)^0.5)
    plot(xbinom,xpois,type="p",xlab="q.binom",ylab="q.pois",
        main=paste("Poisson gegen Binomial\nn=",n,
                   ", p=",nice(p,2),"lambda=",nice(n*p,2),sep=""))
    abline(0,1)
    par(mfrow=c(1,1))
  }
  slider(refresh.code,
       c("n: Versuchsanzahl",
         "p: Erfolgs-WS in %",
          "m: Punkteanzahl"),
       c(1,1,5),c(500,99,100),c(1,1,1),c(50,10,50)  )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

333 ⟨*Gamma-Verteilungen* 333⟩≡

```
par(mfrow=c(2,2)); z<-seq(0,1,length=100)
for(n in c(1,3,20,60)){
  x<-z*3*n;y<-dgamma(x,n,1)
  plot(x,y,type="l",xlab="t",ylab="f(t)")
  title(paste("Gamma-Verteilung, n=",n,", lambda=1",sep=""))
}
par(mfrow=c(1,1))
```

334 ⟨* 5⟩+≡
```
n<-50;   p<-0.7;   s.0<-29
s<-0:n;   F.s<-pbinom(0:n,size=n,p=p)
plot(s,F.s,type="h",lty=3)
lines(s,F.s,type="s")
title(paste("Binomialverteilung\nn=",n,", p=",p,", s.0=",s.0,sep=""))
lines(cbind(c(0,s.0),F.s[s.0+1]),col="red",lty=2)
text(0,.95,paste("P(X<=",s.0,")=",round(F.s[s.0+1],3),sep=""),pos=4)
```

335 ⟨* 5⟩+≡
```
par(mfrow=c(3,3))
for(n in c(5,10,50))
  for(p in c(5/50,25/50,45/50)){
   s<-0:n;   f.s<-dbinom(s,size=n,p=p)
    plot(s,f.s,type="h")
    title(paste("Binomialverteilung\nn=",n,", p=",p,sep=""))
  }
par(mfrow=c(1,1))
```

336 ⟨* 5⟩+≡
```
par(mfrow=c(2,2))
for(lambda in c(0.5,2.5,5,25)){
  x<-0:max(10,2*lambda)
  f.x<-dpois(x,lambda)
  plot(x,f.x,type="h",
     main=paste("pois(lambda=",lambda,")",sep=""))
}
par(mfrow=c(1,1))
```

337 ⟨* 5⟩+≡
```
par(mfrow=c(3,1))
f1.x<-dnorm(x1<-seq(-4,4,length=100))
plot(x1,f1.x,type="l",main="Standardnormalverteilung: N(0,1)")
f2.x<-dnorm(x2<-seq(10-4*2,10+4*2,length=100),10,2)
plot(x2,f2.x,type="l",main="Normalverteilung: N(10,2)")
f3.x<-dnorm(x3<-seq(-5-4*.7,1-5+4*.7,length=100),-5,.7)
plot(x1,f1.x,type="l",main="verschiedene Normalverteilungen",
     xlim=c(-7,16),ylim=c(0,1.2*dnorm(0,0,.7)))
points(x2,f2.x,type="l",lty=2)
points(x3,f3.x,type="l",lty=3)
legend(12,0.6,lty=1:3,c("N( 0,1)","N(10,2)","N(-5,0.7)"))
par(mfrow=c(1,1))
```

338 ⟨* 5⟩+≡
```
x<-0:20
par(mfrow=c(2,2))
for(p in c(.05,.2,.5,.8)){
  y<-dgeom(x,p)
  plot(x,y,type="h",xlab="x : Fehlversuche",ylab="f(x)",
     main=paste("geometrische Verteilung p=",p,sep=""))
}
par(mfrow=c(1,1))
```

339 ⟨* 5⟩+≡
```
x<-(0:100)/5
par(mfrow=c(2,2))
for(lambda in c(0.1,0.5,1,5)){
  y<-dexp(x,lambda)
  plot(x,y,type="l",xlab="t",ylab="f(t)",
     main=paste("Exponentialverteilung lambda=",lambda,sep=""))
}
par(mfrow=c(1,1))
```

340 ⟨*start* 255⟩+≡

```
"coal"<-structure(list(date = c(1851.2, 1851.63, 1851.97, 1851.97, 1852.31,
1852.35, 1852.36, 1852.39, 1852.98, 1853.2, 1853.23, 1853.32,
1853.5, 1854.13, 1856.4, 1856.51, 1856.54, 1856.62, 1857.14,
1857.4, 1857.58, 1858.09, 1858.15, 1858.41, 1858.95, 1860.13,
1860.17, 1860.59, 1860.85, 1860.92, 1860.97, 1861.18, 1861.74,
1861.84, 1862.14, 1862.89, 1862.94, 1863.18, 1863.79, 1863.94,
1863.99, 1865.46, 1865.97, 1866.06, 1866.34, 1866.45, 1866.83,
1866.95, 1866.95, 1867.64, 1867.85, 1867.86, 1868.75, 1868.9,
1868.99, 1869.25, 1869.44, 1869.55, 1869.81, 1869.87, 1870.12,
1870.52, 1870.56, 1870.63, 1871.03, 1871.15, 1871.17, 1871.74,
1871.82, 1872.12, 1872.24, 1872.77, 1873.14, 1874.29, 1874.55,
1874.89, 1874.98, 1875.33, 1875.93, 1875.93, 1875.93, 1876.97,
1877.06, 1877.11, 1877.19, 1877.78, 1877.81, 1878.18, 1878.2,
1878.24, 1878.43, 1878.7, 1879.04, 1879.17, 1879.5, 1880.06,
1880.54, 1880.69, 1880.94, 1881.11, 1881.97, 1882.13, 1882.3,
1882.3, 1882.33, 1882.85, 1883.8, 1883.85, 1884.07, 1884.86,
1885.17, 1885.46, 1885.98, 1886.62, 1886.69, 1886.75, 1886.92,
1887.13, 1887.41, 1888.3, 1889.05, 1889.2, 1889.79, 1890.1, 1890.19,
1891.25, 1891.67, 1892.65, 1893.51, 1894.48, 1895.32, 1896.07,
1896.28, 1896.33, 1899.63, 1901.39, 1902.67, 1905.06, 1905.19,
1905.52, 1906.77, 1908.14, 1908.27, 1908.63, 1909.13, 1909.83,
1910.36, 1910.97, 1912.52, 1913.78, 1914.41, 1916.62, 1918.03,
1922.53, 1922.68, 1923.57, 1927.16, 1928.11, 1930.15, 1930.75,
1931.08, 1931.83, 1931.88, 1932.07, 1932.86, 1932.88, 1933.88,
1934.72, 1935.64, 1935.7, 1936.6, 1937.5, 1938.35, 1939.82, 1940.22,
1940.42, 1941.42, 1941.52, 1941.57, 1942, 1942.13, 1942.48, 1946.95,
1947.02, 1947.62, 1947.64, 1947.69, 1951.41, 1957.88, 1960.49,
1962.22)), .Names = "date", row.names = c("1", "2", "3", "4",
"5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15",
"16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26",
"27", "28", "29", "30", "31", "32", "33", "34", "35", "36", "37",
"38", "39", "40", "41", "42", "43", "44", "45", "46", "47", "48",
"49", "50", "51", "52", "53", "54", "55", "56", "57", "58", "59",
"60", "61", "62", "63", "64", "65", "66", "67", "68", "69", "70",
"71", "72", "73", "74", "75", "76", "77", "78", "79", "80", "81",
"82", "83", "84", "85", "86", "87", "88", "89", "90", "91", "92",
"93", "94", "95", "96", "97", "98", "99", "100", "101", "102",
"103", "104", "105", "106", "107", "108", "109", "110", "111",
"112", "113", "114", "115", "116", "117", "118", "119", "120",
"121", "122", "123", "124", "125", "126", "127", "128", "129",
"130", "131", "132", "133", "134", "135", "136", "137", "138",
"139", "140", "141", "142", "143", "144", "145", "146", "147",
"148", "149", "150", "151", "152", "153", "154", "155", "156",
"157", "158", "159", "160", "161", "162", "163", "164", "165",
"166", "167", "168", "169", "170", "171", "172", "173", "174",
"175", "176", "177", "178", "179", "180", "181", "182", "183",
"184", "185", "186", "187", "188", "189", "190", "191"), class = "data.frame")
```

341 ⟨* 5⟩+≡
```
par(mfrow=c(2,2)); z<-seq(0,1,length=100)
lambda<-25
for(n in c(1,3,20,60)){
  x<-z*max(10,2*n);y<-dgamma(x,n,1)
  plot(x,y,type="l",xlab="t",ylab="f(t)")
  title(paste("Erlang-Verteilung\nn=",n,", lambda=",lambda,sep=""))
}
par(mfrow=c(1,1))
```

342 ⟨*start* 255⟩+≡
```
exp.mere<-function(){
  restart<-function(...){
    set.seed(slider(no=2))
    versuche<-slider(no=1); n<-versuche*4
    x<-matrix(sample(1:6,n,T),versuche,4)
    sechs.in.4<-apply(x==6,1,any)
    freq.6.in.4<-cumsum(sechs.in.4)/(1:versuche)
    n<-versuche*48
    x<-matrix(sample(1:6,n,T),versuche,48)
    doppel.6.in.24<-apply(x==6,1,function(x)any(x[1:24]&x[25:48]))
    freq.doppel.6.in.24<-cumsum(doppel.6.in.24)/(1:versuche)
    plot(freq.6.in.4,ylim=0:1,log="x",type="l",bty="n", # xlim=c(1,1000),
         ylab="relatvie H\344ufigkeit",xlab="Versuche")
    lines(1:versuche,freq.doppel.6.in.24,lty=2,col=2)
    abline(h=0.5)
    legend(100,0.95,bty="n",lty=1:2,col=1:2,legend=c("6 bei 4 W\374rfen",
              "Doppel-6 bei 24 W\374rfen mit 2 W\374rfeln"))
    title("Wahrscheinlichkeitsfrage von Chevalier de M\351r\351")
  }
  slider(restart,c("Anzahl Versuche","Zufallsstart"),
    c(100,1), c(10000,100), c(100,1), c(100,59)
  )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

12.1.14 reale Chunks

343 ⟨* 5⟩+≡
```
curve(dchisq(x,1),0.5,40,lty=1)
for(k in h<-c(99,8,3,1))
    curve(dt(x,k),-5,5,add=k<99,lty=k)
legend(-4,.3,paste("k=",h),lty=h)
title("t-Verteilungen")
```

344 ⟨* 5⟩+≡
```
curve(dchisq(x,1),0.5,40,lty=1)
for(k in h<-c(3,9,27))
    curve(dchisq(x,k),0,40,add=k>1,lty=k)
legend(30,.3,paste("k=",h),lty=h)
title("Chi-Quadrat-Verteilungen")
```

345 ⟨* 5⟩+≡
```
x<-c(3,1,2,2)
for(lambda in 1:3)
  cat("L(",lambda,") = ",prod(dpois(x,lambda)),"\n",sep="")
```

346 ⟨* 5⟩+≡
```
x<-log(dateigroessen)
n<-length(x); p<-seq(x)/n-n/2
x.empirisch<-sort(x)
x.theoretisch<-qnorm(p,mean(x),sd(x))
plot(x.theoretisch,x.empirisch); abline(0,1)
title(" log(Dateigroessen) gegen NV-Quantile")
```

347 ⟨* 5⟩+≡
```
par(mfrow=c(1,2))
n<-length(zwischen.unfalls.zeiten.02)
q.theo02<-qexp(((1:n)-0.5)/n,lambda02)
qqplot(zwischen.unfalls.zeiten.02,q.theo02)
n<-length(zwischen.unfalls.zeiten.04)
q.theo04<-qexp(((1:n)-0.5)/n,lambda04)
qqplot(zwischen.unfalls.zeiten.04,q.theo04)
par(mfrow=c(1,1))
```

348 ⟨*stelle Unfalldaten für 2002 bereit* 348⟩≡

```
zwischen.unfalls.zeiten.02<-
c(16.100,16.600,3.340,22.900,1.350,0.968,8.720,1.060,15.800,3,
10.100,1.920,37.800,0.160,2.050,1.880,11.900,7.870,1.120,13.200,
8.010,3.830,2.180,5.240,7.570,14.700,42.500,4.690,4.180,3.750,
10.200,3.020,23.300,2.900,0.710,11.200,29.300,6.380,0.238,0.222,
1.080,4.510)
unfaelle.pro.woche.02<-rep(0:4,c(24,18,7,2,1))
unfaelle.pro.woche.02.table<-table(unfaelle.pro.woche.02)
zwischen.unfalls.zeiten.04<-c( 50.5,18.6,10.9,39.6,47.8,13.7,32.3,5.76,0.997,3.37,2.81,
23.2,11,0.747,2.15,4.04,0.805,2.26,3.47,2.98,5.98,2.38,0.111,2.17,6.77,2.06,
2.53,0.198,2.36,4.82,6.27,5.17,4.74,0.712,7.14,9.97,0.667,2.18,0.905,3.35,0.0104 )
unfaelle.pro.woche.04a<-rep(0:1,c(20,6))
unfaelle.pro.woche.04a.table<-table(unfaelle.pro.woche.04a)
unfaelle.pro.woche.04b<-rep(0:5,c(9,6,6,4,0,1))
unfaelle.pro.woche.04b.table<-table(unfaelle.pro.woche.04b)
```

349 ⟨*start* 255⟩+≡

⟨*stelle Unfalldaten für 2002 bereit* 348⟩
⟨*stelle Funktion* `g.dach.x()` *bereit* 350⟩

350 ⟨*stelle Funktion* `g.dach.x()` *bereit* 350⟩≡

```
g.dach.x<-function(data,main,...){
  data<-sort(data)
  freq.x<-table(factor(data,x<-min(data):max(data)))
  g.dach<-x[-1]*freq.x[-1]/freq.x[-length(freq.x)]
  cex<-sqrt(freq.x[-1]*freq.x[-length(freq.x)]); cex<-1+cex/max(cex)*4
  plot(x[-1],g.dach,cex=cex,main=main,...)
  title(paste("\n\nMittel:",signif(mean(data),4),
              "/ Stichprobenvarianz:",signif(var(data),4)))
}
```

351 ⟨*start* 255⟩+≡

```
exp.g.dach<-function(){
  refresh<-function(...){
  g.dach.plot<-function(stpr,typ){
    mittel<-mean(stpr); s.q<-var(stpr); N<-length(stpr)
    x.max<-max(stpr);x.min<-min(stpr)
    freq<-table(factor(stpr,x.min:x.max))
    g.x<-((x.min+1):x.max)*freq[-1]/freq[-length(freq)]
    ok<-(!is.nan(g.x))& g.x<Inf
    cex<-sqrt(freq[-1]*freq[-length(freq)])[ok]; cex<-cex/max(cex)*5
    plot(((x.min+1):x.max)[ok],g.x[ok],cex=cex,
    ylab="g.dach(x)",xlab="x",ylim=c(0,max(g.x[ok])),
    main=paste(typ,"\nN=",N,", s^2/Mittel:",signif(s.q/mittel,3)))
  }
  par(mfrow=c(2,2))
    N<-slider(no=1)
    n.binom<-slider(no=2); p.binom<-slider(no=3)
    lambda<-slider(no=4)
    p.geom<-slider(no=5)
    n.unif<-slider(no=6)
    seed<-slider(no=7); set.seed(seed)
    stpr<-rbinom(N,n.binom,p.binom)
    g.dach.plot(stpr,paste("binom(",n.binom,",",p.binom,"), seed=",seed))
    stpr<-rpois(N,lambda)
    g.dach.plot(stpr,paste("pois(",lambda,"), seed=",seed))
    stpr<-rgeom(N,p.geom)
    g.dach.plot(stpr,paste("geom(",p.geom,"), seed=",seed))
    stpr<-ceiling(runif(N,1,n.unif))
    g.dach.plot(stpr,paste("unif( 1",n.unif,"), seed=",seed))
    par(mfrow=c(1,1))
  }
  slider(refresh,c("N","n.binom","p.binom","lambda","p.geom","n.unif","zufall"),
       c(20,4,.05,0.5,.05,3,1),c(2000,100,.95,20,.95,20,999),
       c(10,1,0.05,.01,.05,1,1),c(100,10,.5,2.5,.5,10,13))
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

352 ⟨*start* 255⟩+≡

```
"dateigroessen" <-
c(6076, 41, 39, 1124, 819, 156, 128, 48, 153, 8, 804, 356, 632,
130, 52, 542, 8, 888, 955, 6658, 653, 1062, 9030, 3473, 9030,
3473, 9030, 3473, 9030, 3473, 345, 215, 2784, 73, 176, 128, 392,
1406290, 37584, 608, 392, 2692348, 578, 1544, 196, 136, 136,
152, 104, 168, 208, 176, 192, 80, 224, 104, 1118, 1588, 2132,
645, 43585, 333, 289, 6017, 504, 501, 1027, 1027, 58, 5105, 10549,
2496, 2642, 1090, 6043, 11482, 3274, 3311, 4960, 10127, 2614,
2759, 310, 30783, 18009, 1640, 30800, 106496, 2786, 1224, 5960,
102400, 2762, 1214, 237568, 2786, 1266, 24576, 13026, 24576,
2806, 28672, 2806, 24576, 2806, 24576, 2830, 24576, 2806, 24576,
7912, 24576, 2806, 20480, 2806, 24576, 2786, 28672, 2806, 1206,
28672, 2806, 24576, 2806, 49152, 2830, 20480, 2952, 20480, 2806,
24576, 2806, 57344, 2786, 1224, 6446, 132, 227741, 206554, 27956,
765956, 217091, 252034, 31977, 1376, 152, 144, 176, 96, 220598,
2646, 4956, 11059, 104, 192, 224, 104, 1027, 1027, 310, 30783,
18009, 1640, 30800, 106496, 2786, 1224, 5960, 102400, 2762, 1214,
237568, 2786, 1266, 24576, 13026, 24576, 2806, 28672, 2806, 24576,
2806, 24576, 2830, 24576, 2806, 24576, 7912, 24576, 2806, 20480,
2806, 24576, 2786, 28672, 2806, 1206, 28672, 2806, 24576, 2806,
49152, 2830, 20480, 2952, 20480, 2806, 24576, 2806, 57344, 2786,
1224, 132, 227741, 206554, 27956, 765956, 217091, 252034, 31977,
4976, 10117, 2418, 2504, 128, 712, 1584, 1044, 218, 63, 299,
20, 153600, 287644, 40960, 28275, 40960, 1322, 1230, 1699, 9,
46, 9, 1384, 726, 27003, 1007, 231, 6838, 6390, 743, 7948, 321,
216, 96, 128, 72, 112, 192, 72, 776, 112, 1584, 1044, 218, 403,
30783, 30800, 19185, 762, 132, 27956, 261186, 72, 1640, 106496,
2786, 1224, 5960, 102400, 2762, 1214, 237568, 2786, 1266, 24576,
13026, 24576, 2806, 28672, 2806, 24576, 2806, 24576, 2830, 24576,
2806, 24576, 7912, 24576, 2806, 20480, 2806, 24576, 2786, 28672,
2806, 1206, 28672, 2806, 24576, 2806, 49152, 2830, 20480, 2952,
20480, 2806, 24576, 2806, 57344, 2786, 1224, 63, 299, 20, 153600,
287644, 40960, 28275, 40960, 1319, 1322, 1230, 1699, 9, 46, 9,
1384, 726, 27003, 1007, 231, 6838, 6390, 743, 7948, 130, 214294,
210489, 151522, 189040, 941, 608, 392, 9368, 13966, 2328, 648,
30668, 13127, 20542, 578, 1502, 196, 136, 136, 152, 104, 168,
208, 176, 192, 80, 224, 104, 1118, 1628, 2077, 645, 43585, 333,
289, 6017, 504, 501, 1027, 1027, 58, 5105, 10549, 2496, 2642,
1090, 6043, 11482, 3274, 3311, 4960, 10127, 2614, 2759, 310,
30783, 18009, 1640, 30800, 106496, 2786, 1224, 5960, 102400,
2762, 1214, 237568, 2786, 1266, 24576, 13026, 24576, 2806, 28672,
2806, 24576, 2806, 24576, 2830, 24576, 2806, 24576, 7912, 24576,
2806, 20480, 2806, 24576, 2786, 28672, 2806, 1206, 28672, 2806,
24576, 2806, 49152, 2830, 20480, 2952, 20480, 2806, 24576, 2806,
57344, 2786, 1224, 6446, 132, 227741, 206554, 27956, 765956,
217091, 252034, 31977)
"handy" <-
c(78, 30, 32, 45, 25, 30, 15, 26, 31, 35, 30, 53, 15, 0, 25,
8, 10, 45, 25, 40, 30, 29, 10, 20, 15, 50, 300, 60, 5, 10, 50,
10, 17, 28, 0, 27, 67, 30, 40, 85, 60, 1, 30, 25, 30, 160, 143,
20, 25, 0, 20, 29, 15, 15, 12, 7.8, 14, 15, 20, 90, 50, 20, 15,
40, 5, 20, 28, 25, 70, 15, 18, 25, 10, 35, 15, 20, 105, 8, 46,
30, 50, 10, 92, 30, 15, 45, 30, 30, 10, 42, 15, 10, 55, 40, 20,
60, 30, 0, 25, 60, 15, 20, 70, 15, 32.62, 40, 40, 40, 30, 60,
20, 0, 50, 0, 30, 25, 15, 15, 30, 60, 50, 140, 23, 50, 45, 20,
23, 25, 10, 28, 20, 20, 15, 25, 15, 12, 100, 25, 15, 0, 65, 50,
35, 0, 35, 75, 40, 20, 55, 37, 42, 15, 15, 15, 66, 40, 60, 35,
```

```
23, 50, 30, 25, 15, 12, 50, 50, 0, 30, 36, 106, 25, 15, 15, 60,
50, 2, 15, 15, 160, 50, 25, 90, 40, 25, 96, 10, 50, 0, 50, 30,
40, 46, 22, 50, 50.74, 40, 80, 15, 50, 15, 5, 15, 40, 65, 10,
15, 15, 40, 50, 60, 27, 100, 30, 15, 40, 17, 18, 0, 12, 30, 12,
25, 25, 35, 0, 65, 55, 0, 20, 30, 0, 0, 48, 30, 15, 45, 25, 50,
10, 15, 0, 50, 35, 60, 20, 15, 100, 25, 60, 32, 10, 17, 20, 41,
15, 15, 90, 30, 40, 25, 30, 85, 30, 150, 16, 22, 0.8, 40, 33,
70, 90, 73, 15, 12, 30, 22, 12, 15, 20, 38.5, 20, 18, 97, 21,
30, 45, 15, 30, 60, 10, 28, 85, 150, 15, 50, 10, 35, 30, 0, 15,
30, 98, 25, 20, 15, 28)
```

353 ⟨*start* 255⟩+≡

```
zeige.varianzreduktion.bei.wachsendem.n <- function(
    t.dach, model="norm", par1, par2, hline=0, vline=1){
  cat("Graphik zum wachsenden Stichprobenumfang\n")
  t.dach<-cbind(t.dach); n<-nrow(t.dach)
  if(!missing(par2)) par(mfrow=c(2,2))
  lab<-names(formals(paste("r",model,sep="")))[-1]
  plot(1:n, t.dach[,1], type="l", ylab=lab[1], xlab="m", log="x")
  abline(h=hline)
  if(!missing(par2)){
    plot(t.dach[,2:1]  , type="l", ylab=lab[1], xlab=lab[2])
    abline(h=hline);abline(v=1)
    plot(1,type="n",xlab="",ylab="",axes=F)
    text(1,1,paste("Entwicklung von", lab,
                   "\nbei wachsendem Stichprobenumfang"))
    plot(t.dach[,2],1:n, type="l", ylab="m", xlab=lab[2], log="y")
    abline(v=vline)
    par(mfrow=c(1,1))
  } else title(paste("Entwicklung von", lab,
                     "\nbei wachsendem Stichprobenumfang"))
}
```

354 ⟨*start* 255⟩+≡

```
exp.ml<-function(stpr,model){
 n<-length(stpr); stpr<-sort(stpr);  stpr.freq<-table(stpr)
 stpr.x<-as.numeric(names(stpr.freq))
 ind<-match(model,c("pois","exp","bernoulli","normal"))
 mitte<-c(1.5*mean(x),.5/mean(x), .3, 1.5*mean(x))[ind]
 breite<-c(5*mean(x),5/mean(x), 1, 5*sd(x))[ind]
 fktn<-c("dpois(x,theta)","dexp(x,theta)","dbinom(x,1,theta)",
         "dnorm(x,theta,var(stpr)^0.5)")[ind]
 doml<-function(...){
  center<-slider(no=1); hspread<-slider(no=2)/2;   anz.theta<-slider(no=3)
  theta.vec<-seq(center-hspread,center+hspread,length=anz.theta)
  theta<-rep(theta.vec,length(stpr.x))
  x<-as.vector(matrix(rep(stpr.x,anz.theta),anz.theta,length(stpr.x),T))
  p.x.th<-matrix(eval(parse(text=fktn)),anz.theta,length(stpr.x))
  p.x.th[is.nan(p.x.th)]<-0
  prod.vec<-apply(p.x.th[,rep(1:length(stpr.x),stpr.freq)],1,prod)
  indizes<-(1:anz.theta)
  ind.max<-indizes[prod.vec==(mprod<-max(prod.vec))][1]
  prod.vec<-prod.vec*10^-(potenz<-floor(log(mprod,base=10)))
  plot(theta.vec,prod.vec,
      type="l",ylim=c(0,max(prod.vec)),
      xlab="untersuchter Parameterbereich",
      ylab=paste("* 10^",potenz,sep=""),
      main="Likelihood")
  segments(theta.vec,0,theta.vec,prod.vec)
 }
  slider(doml,c("Mitte","Intervallbreite","Anzahl Punkte"),
   c(mitte-breite/2,breite/1000,5),
   c(mitte+breite/2,breite,100),
   c(breite/1000,breite/1000,1),
   c(mitte,breite,10))
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

355 ⟨*start* 255⟩+≡

```
poisson.erkennungsplot<-function(stpr){
 stpr<-sort(stpr)
 freq<-table(stpr)
 x<-as.numeric(names(freq))
 cex<-1+ceiling(3*(freq-min(freq))/(max(freq)-min(freq))^2)
 y<-log(freq/length(stpr))+log(gamma(x+1))
 plot(x,y,cex=cex,
      ylab="log(rel.freq)+log(x!)",
      main="Erkennungsplot Poissonverteilung")
  abline(res<-lsfit(rep(x,freq),rep(y,freq))$coef)
  names(res)[1]<-"Achsenabschnitt"
  res[1]
}
exp.erkennungsplot<-function(stpr){
  stpr<-sort(stpr)
  F.dach<-(seq(stpr)-0.5)/length(stpr)
  plot(stpr,-log(1-F.dach),
       xlab="x",ylab="-log(1-F.dach(x))")
  res<-lsfit(stpr,-log(1-F.dach),intercept=FALSE)$coef
  abline(c(0,res)); names(res)<-"Steigung"
  res
}
```

356 ⟨*start* 255⟩+≡

```
exp.est.fns<-function(){
 refresh<-function(...){
    n<-slider(no=1)
    wd<-slider(no=2)
    lambda<-slider(no=3)
    set.seed(seed<-slider(no=4))
    result<-matrix(0,5,wd)
    for(i in 1:wd){
       stpr<-rpois(n,lambda)
       theta1<-1
       theta2<-stpr[1]
       theta3<-0.05*mean(stpr)-3.33
       theta4<-mean(stpr)
       theta5<-median(stpr)
       result[,i]<-c(theta1,theta2,theta3,theta4,theta5)
    }
    boxplot(split(result,row(result)),horizontal=TRUE,
    names=c("theta=1","X_1","mean/20-3.33","mean","median"))
    title(paste("n=",n,"wd=",wd,"lambda=",lambda,"seed=",seed))
    abline(h=lambda,lty=2)
 }
 slider(refresh,c("n","Wiederholungen","lambda","seed"),
      c(2,5,.2,1),c(1000,100,30,1000),c(1,10,.1,1),c(20,20,5,7))
 cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

357 ⟨*start* 255⟩+≡

```
exp.nv.mischung<-function(){
ziehe.stichprobe<-function(n){
  x<-c(rnorm(n*8,0,1),rnorm(n*2,0,sqrt(9)));sample(x,n,T)
}
titel<-"GG: 80% N(0,1), 20% N(0,9),"
n<-100;wd<-1000;seed<-13; set.seed(seed)
result.mean<-result.median<-rep(0,wd)
for(i in 1:wd){
  stpr<-ziehe.stichprobe(n)
  result.mean[i]<-mean(stpr);result.median[i]<-median(stpr)
}
xy1<-density(result.mean);
xy2<-density(result.median)
ymax<-max(xy1$y,xy2$y)
plot(xy1,type="l",main="",ylab="f.dach",ylim=c(0,ymax))
lines(xy2$x,xy2$y,lty=2)
legend(.2,.8*ymax,c("Mittelwert","Median"),lty=1:2)
title(paste("Verteilung von Stichprobenfunktionen\n",
            titel," wd=",wd,", n=",n,", seed=",seed,sep=""))
cat("Plot erstellt\n"); NULL
}
```

358 ⟨*start* 255⟩+≡

```
exp.exp.mittel<-function(){
 redo<-function(...){
  n<-slider(no=1);wd<-slider(no=2);seed<-slider(no=3)
  set.seed(seed);mittel<-numeric(wd)
  for(i in 1:wd) mittel[i]<-mean(rexp(n))
  old<-par(mfrow=1:2)
  plot(ecdf(mittel),main=paste("Mittel aus exp(1),n=",n,
        ", wd=",wd,"\necdf(mittel)",sep=""))

  mx<-mean(mittel); sdx<-sd(mittel)
  x<-seq(mx-4*sdx,mx+4*sdx,length=100)
  F.x<-pnorm(x,mx,sdx)
  lines(x,F.x,col="red")
  qqnorm(mittel); qqline(mittel)
  par(old)
 }
 slider(redo,c("Stichprobenumfang","Wiederholungen","Zufall"),
        c(2,2,2,),c(50,1000,1000),c(1,1,1),c(10,100,2))
 cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

359 ⟨*start* 255⟩+≡

```
exp.nv.est<-function(mue=0,sd=1){
  redo<-function(...){
    n.max<-slider(no=1);zoom<-slider(no=2);set.seed(slider(no=3))
    X <- rnorm(n.max, mue, sd)
    n.vec<-1:n.max
    mue.dach<-(cumsum(X)/n.vec)[-1]
    sigma.q.dach<-(cumsum(X^2)/(n.vec-1))[-1]+
                  -mue.dach^2*(n.vec/(n.vec-1))[-1]
    plot(mue.dach,sigma.q.dach,type="l",
      xlim=c(mue-2*sd/zoom,mue+2*sd/zoom),
      ylim=c(sd-sd/zoom,sd+sd/zoom))
    abline(v=mue,h=sd)
    title(paste("Stichprobenumfang bis",n.max))
    return(c(mean(X),var(X)))
  }
  slider(redo,c("n.max","Zoom Faktor","seed"),
        c(3,1,1),c(2000,20,1000),c(1,1,1),c(20,.3,1))
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

360 ⟨*start* 255⟩+≡

```
exp.nv.an.beta<-function(){
  redo<-function(...){
    n<-slider(no=1); wd<-slider(no=2)
    a<-slider(no=3); b<-slider(no=4); set.seed(seed<-slider(no=5))
    stpr<-matrix(rbeta(n*wd,a,b),n,wd)
    stpr.mittel<-apply(stpr,2,mean)
    stpr.sd<-apply(stpr,2,var)^0.5
    x<-seq(0.001,.999,length=100)
    y<-dbeta(x,a,b)
    plot(x,y,type="p",col="red",
         xlim=c(-1,2),ylim=c(0,2*max(y)))
    title(paste("GG: Beta(a=",a,", b=",b,")\nn=",n,", wd=",wd,", seed=",seed,sep=""))
    for(i in 1:wd)
        curve(dnorm(x,stpr.mittel[i],stpr.sd[i]),-1,2,add=T)
  }
  slider(redo,c("n","wd","beta: Par.1","beta: Par.2","seed"),
          c(3,2,0.05,0.05,1),c(50,20,3,3,100),
          c(1,1,0.05,0.05,1),c(5,3,1,1,7)
  )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

361 ⟨*start* 255⟩+≡

```
exp.outlier<-function(){
  redo<-function(...){
    n<-slider(no=1); wd<-slider(no=2)
    anteil.outlier<-slider(no=3)/100
    lage.outlier<-slider(no=4); sd.outlier<-slider(no=5)
    set.seed(seed<-slider(no=6))
    n.normal<-ceiling(n*(1-anteil.outlier))
    stpr<-c(rnorm(wd*n.normal),
                rnorm(wd*(n-n.normal), lage.outlier,sd.outlier))
    stpr<-matrix(stpr,n,wd,byrow=TRUE)
    stpr.mittel<-apply(stpr,2,mean)
    stpr.sd<-apply(stpr,2,sd)
    xmin<-min(-3.5,lage.outlier-3.5*sd.outlier)
    xmax<-max(3.5,lage.outlier+3.5*sd.outlier)
    ymax<-max(dnorm(0), dnorm(lage.outlier,lage.outlier,sd.outlier))
        x<-seq(xmin,xmax,length=100)
    f.x<-dnorm(x)*(1-anteil.outlier)+
         dnorm(x,lage.outlier,sd.outlier)*anteil.outlier
    plot(x,f.x,type="b",xlim=c(xmin,xmax), ylim=c(0,ymax))
        title(paste(100*(1-anteil.outlier),"% norm(0,1) und ",
                100*anteil.outlier, "% norm(",
                     lage.outlier,",",sd.outlier,")\n",
                     "n=",n,", wd=",wd,"\n", sep=""))
    for(i in 1:wd)curve(dnorm(x,stpr.mittel[i],stpr.sd[i]),xmin,xmax,add=T)
  }
  slider(redo,c("n","wd","Prozent Ausreisser",
        "Lage Ausrei{\ss}er","Std.-Abw. Ausreisser","seed"),
      c(10,2,0,-20,0.5,1),c(200,20,50,20,10,100),
      c(1,1,0,0,0.5,1),c(20,10,10,7,1,1)
  )
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

12.1.15 Funktionen

362 ⟨*start* 255⟩+≡

```
exp.ki.p<-function(){
  redo<-function(...){
    n<-slider(no=1); alpha<-slider(no=2); wd<-slider(no=3)
    p<-slider(no=4); seed<-slider(no=5)
    set.seed(seed); u<-o<-numeric(wd)
    t.value<-qt(1-alpha/2,n-1)
    for(i in 1:wd){
       p.dach<-mean(rbinom(n,1,p))
       sigma.dach<-sqrt(p.dach*(1-p.dach)/n)
       step<-t.value*sigma.dach
       u[i]<-p.dach-step; o[i]<-p.dach+step
    }
    plot(u,type="n",ylim=0:1,bty="n",xlab="Versuche",
         ylab="realisierte KIs",axes=F)
         axis(2)
    title(paste("n=",n,", alpha=",alpha,
          ", p=",p,", seed=",seed,sep=""))
    segments(1:wd,u,1:wd,o)
    abline(h=p)
  }
  slider(redo,c("n","alpha","wd","p","seed"),
         c(5,.001,5,.01,1),
         c(500,.3,100,.99,100),
         c(1,.001,1,.01,1),
         c(10,.05,10,.5,1))
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

363 ⟨*start* 255⟩+≡

```
demo.n.alpha.len<-function(p.dach){
redo<-function(...){
a<-slider(no=1);b<-slider(no=2)
n.set<-10:100
alpha.set<-seq(.2,.01,length=20)
sigma.dach<-t.value<-matrix(0,length(alpha.set),length(n.set))
for(j in seq(n.set)){
  t.value[,j]<-qt(1-alpha.set/2,n.set[j]-1)
  sigma.dach[,j]<-sqrt(p.dach*(1-p.dach)/n.set[j])
}
length.KI<-t(2*t.value*sigma.dach)
persp(n.set,1-alpha.set,length.KI,
      phi=a,theta=b,ticktype="detailed",
      xlab="n",ylab="1-alpha")
}
slider(redo,c("a","b"), c(-180,-180), c(360,360), c(5,5),c(10,-30))
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

12.1.16 Codechunks

364 ⟨*zeige Raucherdemo* 257⟩+≡
```
alpha<-0.05
abk<-c("Ia","Ib","Ic","II","III")
n<-c(21895,44184,25461,17366,33951)
p.dach<-c(32,86,56,106,66)/n
UG<-p.dach-qnorm(1-alpha/2) * sqrt(p.dach*(1-p.dach)/n)
OG<-p.dach+qnorm(1-alpha/2) * sqrt(p.dach*(1-p.dach)/n)
plot(1:5,ylim=c(0,max(1000*OG)),bty="n",type="n",
     axes=F,ylab="Sterberate in 1/1000",xlab="Gruppen")
title("KIs: Bronchialkrebs-WS und Rauchen")
axis(2); axis(1,at=1:5,labels=abk)
segments(1:5,1000*UG,1:5,1000*OG)
```

365 ⟨*zeichne Zusammenhang IL gegen n* 258⟩+≡
```
#old<-par(mfrow=1:2)
# 1. Plot:
p.dach<-0.5; n<-c(10,11,12,14,16,20,25,30,40,50,70,100)
n<-c(n,10*n,100*n)
alpha.set<-c(.01,.02,.05,.1)
plot(n,n,type="n",ylim=0:1,log="x",ylab="IL",bty="n")
for(i in seq(alpha.set)){
  alpha<-alpha.set[i]
  IL<-2*qnorm(1-alpha/2)*sqrt(p.dach*(1-p.dach)/n)
  lines(n,IL,lty=i)
}
title(paste("Intervalllaenge gegen n\np.dach=",p.dach))
legend(500,.8,lty=seq(alpha.set),
legend=paste(" (1-alpha) =",1-alpha.set),bty="n")
```

366 ⟨*zeichne Zusammenhang n gegen* $1-\alpha$ 259⟩+≡
```
p.dach<-0.5; alpha<-seq(.001,.2,length=100)
IL.set<-c(.01,.02,.04,.08,.13)
plot(1,type="n",xlim=c(.8,1),ylim=c(50,50000),
    log="y",xlab="1-alpha",ylab="n",bty="n")
for(i in seq(IL.set)){
  IL<-IL.set[i]
  n<-4*p.dach*(1-p.dach)*(qnorm(1-alpha/2)/IL)^2
  lines(1-alpha,n,lty=i)
}
title(paste("Stichprobenumfang gegen 1-alpha\np.dach=",p.dach))
legend(.95,150,lty=seq(IL.set),
legend=paste(" IL =",IL.set),bty="n")
```

12.1.17 Erstellung von `kif1.ps`

367 ⟨* 5⟩+≡
```
x<-seq(.1,40,length=200)
f.x<-dchisq(x,15)
alpha.u<-.1; alpha.o<-.1
x.u<-qchisq(alpha.u,15)
x.o<-qchisq(1-alpha.o,15)
plot(x,f.x,type="l",bty="n",ylab="",xlab="",axes=FALSE)
xu<-x[x<=x.u]
f.xu<-dchisq(xu,15)
lines(xu,f.xu,type="h")
xo<-x[x>=x.o]
f.xo<-dchisq(xo,15)
lines(xo,f.xo,type="h")
axis(1,at=c(x.u,x.o),labels=expression(x[u],x[o]))
text((x.u+x.o)/2,.01,cex=2,expression(1-alpha))
text((x.u)/3,.03,cex=2,expression(alpha[u]))
text((x.o+x.u),.03,cex=2,expression(alpha[o]))
arrows((x.u)/3,.025,(x.u)/1.6,0.015)
arrows((x.u+x.o),.025,(x.u+x.o)/1.2,0.015)
```

368 ⟨*start* 255⟩+≡
```
k1 <-  600
z1 <- 400
n1 <- k1+z1
k2 <- 141
z2 <- 109
n2 <- k2+z2
k3 <- 136
z3 <- 114
n3 <- k3+z3
k4 <- 123
z4 <- 137
n4 <- k4+z4
p1 <- k1/n1
p2 <- k2/n2
p3 <- k3/n3
p4 <- k4/n4
```

12.1.18 Modellierung der Anfangsmeinung

Dichte und Verteilungsfunktion I.

369 ⟨* 5⟩+≡
```
x <- seq(0,0.4,0.01)
x <- c(x,seq(0.4,0.6,0.0005))
x <- c(x,seq(0.6,1.0,0.01))
par(mfrow=c(2,2))
y <- dbeta(x,250,250)
plot(x,y,type="l",bty="n")
title("beta(250,250)")
y <- pbeta(x,250,250)
plot(x,y,type="l",bty="n")
title("beta(250,250)")
y <- dbeta(x,500,500)
plot(x,y,type="l",bty="n")
title("beta(500,500)")
y <- pbeta(x,500,500)
plot(x,y,type="l",bty="n")
title("beta(500,500)")
par(mfrow=c(1,1))
```

Dichte und Verteilungsfunktion II.

370 ⟨* 5⟩+≡
```
x <- seq(0,0.4,0.01)
x <- c(x,seq(0.4,0.6,0.0005))
x <- c(x,seq(0.6,1.0,0.01))
par(mfrow=c(2,2))
y <- dbeta(x,1000,1000)
plot(x,y,type="l",bty="n")
title("beta(1000,1000)")
y <- pbeta(x,1000,1000)
plot(x,y,type="l",bty="n")
title("beta(1000,1000)")
y <- dbeta(x,5000,5000)
plot(x,y,type="l",bty="n")
title("beta(5000,5000)")
y <- pbeta(x,5000,5000)
plot(x,y,type="l",bty="n")
title("beta(5000,5000)")
par(mfrow=c(1,1))
```

12.1.19 Profile der Beta-Verteilung

371 ⟨* 5⟩+≡
```
par(mfrow=c(3,3))
x <- seq(0,1,0.01)
y <- dbeta(x,1,1)
plot(x,y,type="l",bty="n")
title("(a,b)=(1,1)")
y <- dbeta(x,1,2)
plot(x,y,type="l",bty="n")
title("(a,b)=(1,2)")
y <- dbeta(x,2,1)
plot(x,y,type="l",bty="n")
title("(a,b)=(2,1)")
y <- dbeta(x,5,5)
plot(x,y,type="l",bty="n")
title("(a,b)=(5,5)")
y <- dbeta(x,2,5)
plot(x,y,type="l",bty="n")
title("(a,b)=(2,5)")
y <- dbeta(x,5,2)
plot(x,y,type="l",bty="n")
title("(a,b)=(5,2)")
y <- dbeta(x,1/2,1/2)
plot(x,y,type="l",bty="n")
title("(a,b)=(1/2,1/2)")
y <- dbeta(x,1/2,1)
plot(x,y,type="l",bty="n")
title("(a,b)=(1/2,1)")
y <- dbeta(x,1,1/2)
plot(x,y,type="l",bty="n")
title("(a,b)=(1,1/2)")
par(mfrow=c(1,1))
```

12.1.20 Funktion ...

372 ⟨*start* 255⟩+≡
```
Lp.diff <- function(n) {
x <- seq(0,1,0.005)
y <- 0:n
xp <- yp <- NULL
for (i in y){
 yp <- c(yp,max(x^i*(1-x)^(n-i)))
 xp <- c(xp,(i/n)^(i)*(1-i/n)^(n-i)) }
#ii <- round(0.4*n):round(0.6*n)
plot(xp,yp,xlab="Lp(y/n;y)",ylab="max(L(p;y)",log="xy",bty="n")
#plot(xp[ii],yp[ii],xlab="Lp(y/n;y)",ylab="max(L(p;y)")
title(paste("Lp: Differenz der Funktionswerte, n =",n))
invisible() }
```

12.1.21 `exp.bayes`

373 ⟨*start* 255⟩+≡

```
exp.bayes <- function(){
### begin refresh.code
refresh.code <- function(...){
p     <- slider(no=1)
nsamp <- slider(no=2)
slider(obj.name="nsamp",obj.value=nsamp)
slider(obj.name="p",obj.value=p)
}
### end refresh.code

### begin go.funct
go.funct <- function(...){
## begin loeschen
 posterior1  <- slider(obj.name="posterior1")
 prior1      <- slider(obj.name="prior1")
 ynorm       <- slider(obj.name="ynorm")
 p.prior     <- slider(obj.name="p.prior")
 p.samp      <- slider(obj.name="p.samp")
 p.posterior <- slider(obj.name="p.posterior")
# prior1      <- prior*ynorm
# posterior1  <- posterior*ynorm
 par(col="white")
 lines(x,prior1,lty=2)
 lines(x,posterior1)
 lines(c(p.prior,p.prior),c(0.1,1),lty=2)
 points(p.prior,0.05,pch=18)
 lines(c(p.posterior,p.posterior),c(0.1,1),lty=1,)
 points(p.posterior,0.05,pch=15)
 lines(c(p.samp,p.samp),c(0.1,1),lty=3)
 points(p.samp,0.05,pch=17)
### end loeschen

### begin new
 alpha   <- slider(obj.name="alpha")
 beta    <- slider(obj.name="beta")
 p       <- slider(obj.name="p")
 nsamp   <- slider(obj.name="nsamp")
 p.prior <- alpha/(alpha+beta)
 a       <- rbinom(1,nsamp,p)
 p.samp  <- a/nsamp
 alpha   <- alpha + a
 beta    <- beta  + nsamp - a
 slider(obj.name="alpha",obj.value=alpha)
 slider(obj.name="beta",obj.value=beta)
 p.posterior <- alpha/(alpha+beta)
 posterior <- slider(obj.name="posterior")
 prior      <- posterior
 posterior  <- dbeta(x,alpha,beta)
 ynorm      <- 1/max(c(prior,posterior))
 prior1     <- prior*ynorm
 posterior1 <- posterior*ynorm
 slider(obj.name="posterior",obj.value=posterior)
 slider(obj.name="posterior1",obj.value=posterior1)
 slider(obj.name="prior1",obj.value=prior1)
 slider(obj.name="ynorm",obj.value=ynorm)
 slider(obj.name="p.prior",obj.value=p.prior)
 slider(obj.name="p.samp",obj.value=p.samp)
```

```
 slider(obj.name="p.posterior",obj.value=p.posterior)
 par(col="black")
 lines(x,prior1,lty=2)
 lines(x,posterior1)
 lines(c(p.prior,p.prior),c(0.1,1),lty=2)
 points(p.prior,0.05,pch=18)
 lines(c(p.posterior,p.posterior),c(0.1,1),lty=1)
 points(p.posterior,0.05,pch=15)
 lines(c(p.samp,p.samp),c(0.1,1),lty=3)
 points(p.samp,0.05,pch=17) }
### end go.funct

### begin reset.funct
reset.funct <- function(...){
plot(c(0,1),c(0,1),type="n",col="black",bty="n",ylab="",xlab="p")
title("Prior - Sample - Posterior")
alpha     <- alpha.def <- 1
slider(obj.name="alpha",obj.value=alpha)
slider(obj.name="alpha.def",obj.value=alpha.def)
beta      <- beta.def <- 1
slider(obj.name="beta",obj.value=beta)
slider(obj.name="beta.def",obj.value=beta.def)
prior     <- dbeta(x,alpha.def,beta.def)
posterior <- prior
ynorm     <- max(prior)
prior1    <- prior*ynorm
p.prior   <- alpha/(alpha+beta)
slider(obj.name="prior1",obj.value=prior1)
slider(obj.name="posterior1",obj.value=prior1)
slider(obj.name="posterior",obj.value=prior)
slider(obj.name="ynorm",obj.value=ynorm)
slider(obj.name="p.prior",obj.value=p.prior)
slider(obj.name="p.posterior",obj.value=p.prior)
slider(obj.name="p.samp",obj.value=p.prior)
lines(c(p.prior,p.prior),c(0.1,1),lty=2)
lines(x,prior1,lty=2)
points(p.prior,0.05,pch=18)}
### end reset.funct

### install sliders
slider(refresh.code,sl.names=c("p","nsamp"),sl.mins=c(0,5),
sl.maxs=c(1,100),
sl.deltas=c(0.05,1),sl.defaults=c(0.5,10),
but.names="go",
but.functions=go.funct,
reset.function=reset.funct,
title = "prior - sample -posterior")

### start
x         <- seq(0.01,.99,0.005)
reset.funct()
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

12.1.22 `plot.beta`

374 ⟨*start* 255⟩+≡

```
plot.beta <- function(){
refresh.code <- function(...){
par(mfrow=c(1,2))
alpha <- a1 <- slider(no=1)
beta  <- a2 <- slider(no=2)
x      <- seq(0.01,0.99,.01)
ybeta  <- y  <- dbeta(x,a1,a2)
yybeta <- yy <- pbeta(x,a1,a2)
yg     <- slider(obj.name="yg")
yyg    <- slider(obj.name="yyg")
ymax   <- max(c(max(yg),max(y)))
plot(x,y,type="l",xlim=c(0,1),ylim=c(0,ymax),ylab=" ",bty="n")
title("Beta: Density")
lines(x,yg,lty=2)
plot(x,yy,type="l",xlim=c(0,1),ylim=c(0,1),ylab="",bty="n")
title("Beta: CDF")
lines(x,yyg,lty=2)
par(mfrow=c(1,1)) }

reset.func <- function(...){
par(mfrow=c(1,2))
x   <- seq(0.01,0.99,.01)
yg  <- dbeta(x,1,1)
yyg <- pbeta(x,1,1)
plot(x,yg,type="l",xlim=c(0,1),ylab="",bty="n")
title("Beta: Density")
plot(x,yyg,type="l",xlim=c(0,1),ylim=c(0,1),ylab="",bty="n")
title("Beta: CDF")
slider(obj.name="yg",obj.value=yg)
slider(obj.name="yyg",obj.value=yyg)
par(mfrow=c(1,1)) }

slider(refresh.code,sl.names=c("alpha","beta"),c(0,0),c(3,3),c(.05,.05),
c(1,1),reset.function=reset.func,
title="Beta-Verteilung")

reset.func()
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

375 ⟨*do.power* 261⟩+≡
```
p <- seq(0,1,.01)
n <- 12
k <- 2
beta1 <- pbinom(k,n,p) + 1 - pbinom(n-k-1,n,p)
plot(p,beta1,type="l",ylab="beta(p)",ylim=c(0,1),bty="n")
#title("Powerfunktionen")
k <- 3
beta2 <- pbinom(k,n,p) + 1 - pbinom(n-k-1,n,p)
lines(p,beta2,lty=2)
k <- 4
beta3 <- pbinom(k,n,p) + 1 - pbinom(n-k-1,n,p)
lines(p,beta3,lty=3)
lines(c(0,0.075),c(0.05,0.05),lty=1)
text(0.125,0.05,"k=2")
lines(c(0,0.075),c(0.1,0.1),lty=2)
text(0.125,0.1,"k=3")
lines(c(0,0.075),c(0.15,0.15),lty=3)
text(0.125,0.15,"k=4")
```

376 ⟨*opfer.taeter.test* 262⟩+≡
```
xx <- matrix(c(627,254,5010,99),nc=2,byrow=TRUE)
chisq.test(xx)
```

377 ⟨*test.wartezeit* 263⟩+≡
```
w <- diff(coal[,1])
h <- hist(w,nclass=30,plot=FALSE)
br <- h$breaks
th <- pexp(br[-1],191/111)
pth <-c(th,1)-c(0,th)
co <- h$counts
print(chisq.test(c(co,0),p=pth))
```

378 ⟨*test.for.all* 264⟩+≡
```
erg <- NULL
for (i in 1:6){
e <- chisq.test(kugtab[,i])
erg <- rbind(erg,c(e$statistic,e$p.value,48))}
kz <- c("X-squared","p-value","df")
ks <- c("kug1","kug2","kug3","kug4","kug5","kug6")
dimnames(erg) <- list(ks,kz)
print(erg)
#print(format(erg[,1:2],digits=1))
```

379 ⟨*zeige.lotto,tabelle* 265⟩+≡
```
showlottotabelle(lottab)
```

380 ⟨*num.aus* 266⟩+≡
```
nt  <- sum(lottab)
nit <- rep(nt/49,49)
h   <- ((lottab-nit)^2)/nit
ch2 <- sum(h)
h1 <- qchisq(.95,48)
h2 <- pchisq(ch2,48)
h3 <- 1-h2
print(c(h1,h2,h3))
erg <- cbind(lottab,nit,h)
dimnames(erg)<- list(NULL,c("n_i","h_i","d_i^2"))
options(digits=4)
cat("\nArbeitstabelle\n")
print(erg)
cat("\nX^2\n")
print(ch2)
cat("\np-value\n")
print(h3)
options(digits=7)
```

381 ⟨*simple.box.and.whisker.ordered* 267⟩+≡
```
h.m <- rbind(po1.m,po0.m)
plot(c(po1.v,po0.v),ylab="weight",bty="n")
title("Potato: Ordered Data + box-and-whisker")
medw <- apply(h.m,1,median)
maxw <- apply(h.m,1,max)
minw <- apply(h.m,1,min)
ii   <- 3+(0:22)*5
dd   <- 1.
segments(ii,minw,ii,maxw,lty=3)
segments(ii-dd,medw,ii+dd,medw,lty=3)
segments(ii-dd,maxw,ii+dd,maxw,lty=3)
segments(ii-dd,minw,ii+dd,minw,lty=3)
abline(v=length(po1.v)+.5,lty=2)
```

382 ⟨*ordered.group.means* 268⟩+≡

```
plot(c(mpo1,mpo0),ylab="weight",bty="n",type="n")
title("Potato: Group Means of Ordered Data + C-lines")
abline(h=mmpo1,lty=2)
abline(h=mmpo0,lty=2)
abline(h=mmpo)
lmpo1 <- length(mpo1)
abline(v=lmpo1+.5,lty=3)
text(1:lmpo1,mpo1,"F")
text((lmpo1+1):(lmpo1+length(mpo0)),mpo0,"M")
text(3,112,"M = Mann")
text(3,116,"F = Frau")
```

383 ⟨*simple.binomial.test* 269⟩+≡

```
x0    <- mpo0[!is.na(mpo0)]
x1    <- mpo1[!is.na(mpo1)]
medp0 <- median(x0)
medp1 <- median(x1)
v1    <- x1 < medp0
v0    <- x0 < medp1
le0 <- rep(1,length(x0))
le1 <- rep(2,length(x1))
plot(c(x1,x0),c(le1,le0),bty="n",
xlab="mean(potato)",ylab="",ylim=c(0,3),pch=18,yaxt="n")
#title("Simple Test by Plotting and Counting")
mi0 <- min(x0)
mi1 <- min(x1)
ma0 <- max(x0)
ma1 <- max(x1)
y0  <- 1.2
y00 <-  .9
y1  <- 1.8
y11 <- 2.1
segments(c(mi0,mi1),c(y0,y1),c(ma0,ma1),c(y0,y1))
points(medp0,y0,pch=3)
points(medp1,y1,pch=3)
lines(c(medp0,medp0),c(y0,y11),lty=3)
lines(c(medp1,medp1),c(y00,y1),lty=3)
text(medp1,y00,paste(sum(v0),sum(!v0)))
text(medp0,y11,paste(sum(v1),sum(!v1)))
text(mi0-5,y0,"M")
text(ma1+5,y1,"F")
```

384 ⟨*simple.confidence.test* 270⟩+≡
```
## data and hypothesis cleared for NA
d1 <- mpo0[!is.na(mpo0)]
h1 <- median(mpo1,na.rm=TRUE)
d2 <- mpo1[!is.na(mpo1)]
h2 <- median(mpo0,na.rm=TRUE)
## doing the test
par(mfrow=1:2)
conf.test1(d1,h1,txt="Median Gr F vs Gr M")
conf.test1(d2,h2,txt="Median Gr M vs Gr F")
par(mfrow=c(1,1))
```

385 ⟨*binomialtabelle* 271⟩+≡
```
#Code zur Erzeugung der Binomialtabelle:
for(n in c(3,4,5,6,8,10,12,14,16,20,30)){
 cat("n=",n,"\\\\\n")
 for(j in 0:n){
  cat("&",j,"&")
  cat(paste(signif(pbinom(j,n,(1:5)/10),4),collapse="&"))
  cat("\\\\[0.9ex]\n")
 }
}
```

386 ⟨*jeder.mit.jedem* 272⟩+≡
```
par(mfrow=c(3,3))
multiempver(mpo0,txt="Gr M")
multiempver(mpo0,mpo1,txt="Gr M, Gr F")
multiempver(mpo0,mpo,txt="Gr M, All")
multiempver(mpo0,mpo1,mpo,txt="Gr M,Gr F,All")
multiempver(mpo1,txt="Gr F")
multiempver(mpo1,mpo,txt="Gr F, All")
multiempver(mpo0,mpo1,mpo,txt="Gr M,Gr F,All")
multiempver(mpo0,mpo1,mpo,txt="Gr M,Gr F,All")
multiempver(mpo,txt="All")
par(mfrow=c(1,1))
```

12.1.23 `plot.chi`

387 ⟨*start* 255⟩+≡

```
plot.chi <- function(){
refresh.code <- function(...){
par(mfrow=c(1,2))
nu     <- n1 <- slider(no=1)
x      <- seq(0.01,149.99,.05)
ychi   <- y  <- dchisq(x,n1)
yychi  <- yy <- pchisq(x,n1)
yg     <- slider(obj.name="yg")
yyg    <- slider(obj.name="yyg")
ymax   <- max(c(max(y),max(yg)))
plot(x,y,type="l",ylim=c(0,ymax),ylab="",bty="n")
title("Chi: Density")
lines(x,yg,lty=2)
plot(x,yy,type="l",ylim=c(0,1),ylab="",bty="n")
title("Chi: CDF")
lines(x,yyg,lty=2)
par(mfrow=c(1,1)) }

reset.func <- function(...){
par(mfrow=c(1,2))
x   <- seq(0.01,149.99,.05)
nuu  <- 10
yg  <- dchisq(x,nuu)
yyg <- pchisq(x,nuu)
plot(x,yg,type="l",ylab="",bty="n")
title("Chisq: Density")
plot(x,yyg,type="l",ylim=c(0,1),ylab="",bty="n")
title("Chisq: CDF")
slider(obj.name="yg",obj.value=yg)
slider(obj.name="yyg",obj.value=yyg)
par(mfrow=c(1,1)) }
nu<-10
slider(refresh.code,sl.names="nu",1,50,1,
nu,reset.function=reset.func,
title="Chi-Verteilung")

reset.func()
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
 }
```

388 ⟨*start* 255⟩+≡

```
exp.binomialtest<-function(){
if(!exists("slider")) source("http://www.wiwi.uni-bielefeld.de/~wolf/software/R-wtools
slider(obj.name="alternative",obj.value="two.sided")
redo<-function(...){
n<-slider(no=1); p<-slider(no=2); alpha<-slider(no=3)
alternative<-slider(obj.name="alternative")
H<-paste( "H: p=",p)
if(alternative=="less") H<-paste( "H: p>=",p)
if(alternative=="greater") H<-paste( "H: p<=",p)
x<-0:n
if(alternative=="two.sided") alpha.h<-alpha/2 else alpha.h<-alpha
F.x<-pbinom(x,n,p)
plot(x,F.x,bty="n",ylim=0:1)
segments(x[-(n+1)],F.x[-(n+1)],x[-1],F.x[-(n+1)])
if(alternative!="greater"){
 abline(h=alpha.h)#% "red"
 k.stern.ind<-sum(F.x<=alpha.h)
 k.stern1<-x[k.stern.ind]
 segments(k.stern1,0,k.stern1, alpha.h)#% "blue"
 segments(k.stern1,alpha.h,k.stern1,
    if(alternative=="less") 1 else 1-alpha.h,lty=2)#% "blue"
 segments(0,.5,k.stern1,.5)#% "blue"
 text(k.stern1/2,.53,paste("gegen",H))#% "blue"
}
if(alternative!="less"){
 abline(h=1-alpha.h)#% "red"
 k.stern.ind<-length(x)+1-sum(F.x>=(1-alpha.h))
 k.stern2<-x[k.stern.ind]
 segments(k.stern2,1,k.stern2,1-alpha.h)#% "blue"
 segments(k.stern2,
    if(alternative=="greater") 0 else alpha.h,
    k.stern2, 1-alpha.h,lty=2)#% "blue"
 segments(n,.5,k.stern2,.5)#% "blue"
 text((k.stern2+n)/2,.53,paste("gegen",H))#% "blue"
}
 title(paste("n=",n,", p=",p,", alpha=",alpha,
   if(alternative=="less")
     paste("\nK= { 0,...,",k.stern1,"}"),
   if(alternative=="greater")
     paste("\nK= {",k.stern2,",...,",n,"}"),
   if(alternative=="two.sided")
     paste("\nK={0,...,",k.stern1,",",k.stern2,",...,",n,"}")
  ))
}
fbut.two.sided<-function(...){
   slider(obj.name="alternative",obj.value="two.sided")
  redo()
}
fbut.less<-function(...){
   slider(obj.name="alternative",obj.value="less")
  redo()
}
fbut.greater<-function(...){
   slider(obj.name="alternative",obj.value="greater")
  redo()
}
slider(redo,c("n","p.0","alpha"),
```

```
  c(2,.01,.01),  c(100,.99,.3),  c(1,.01,.01),  c(28,.5,.05),
  c(fbut.two.sided,fbut.less,fbut.greater),
  c("G: p!=p.0","G: p<p.0","G: p>p.0"),title="Binomialtest"
)
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

389 ⟨*start* 255⟩+≡

```
exp.nv.guete<-function(){
if(!exists("slider")) source("http://www.wiwi.uni-bielefeld.de/~wolf/software/R-wtools
slider(obj.name="alternative",obj.value="two.sided")
redo<-function(...){
n<-slider(no=1); mu0<-slider(no=2); mu1<-slider(no=3)
alpha<-slider(no=4) # ; beta<-slider(no=5)
agr<-mu1>mu0 # agr==1  <-> greater
alternative<-slider(obj.name="alternative")
delta<-4; sdx<-1/sqrt(n)
x<-seq(min(mu0,mu1)-delta,max(mu0,mu1)+delta,length=200)
fH.x<-dnorm(x,mu0,sdx)
fG.x<-dnorm(x,mu1,sdx)
par(mfrow=2:1)
if(alternative!="two.sided"){
 agr<-alternative=="greater"
 k.stern<-qnorm(if(agr) 1-alpha else alpha,mu0,sdx)
 beta<-if(agr) pnorm(k.stern,mu1,sdx) else 1-pnorm(k.stern,mu1,sdx)
 beta.all<-if(agr) pnorm(k.stern,x,sdx) else 1-pnorm(k.stern,x,sdx)
 plot(x,fH.x,bty="n",type="l",ylim=c(0,max(fH.x)*1.2),
      ylab="Verteilung PG unter H bzw. G",xlab="Parameter")
 title(paste("H: mean =",mu0,
             if(agr)"/  G: mean >" else ",  G: mean <",mu0))
 lines(x,fG.x,col="red")
 ind<-if(agr) x>=k.stern else x<=k.stern
 lines(x[ind],fH.x[ind],type="h",col="black")
 lines(x[!ind],fG.x[!ind],type="h",col="red")
 usr<-par()$usr
 segments(mu1,0,mu1,0.9*usr[4],lty=3,col="red")
 text(mu1,usr[4]*.96,paste("true mean:",mu1),col="red")
 plot(x,x,type="n",bty="n",ylim=c(0,1.2),
    ylab="WS: Beobachtung in K",xlab="Parameter")
 title("power")
segments(mu0,0,mu0,alpha)
 arrows(mu0+1.5,alpha/2,mu0,alpha/2); text(mu0+2,alpha/2+.03,"alpha")
 arrows(mu1+1.5,1-beta/2,mu1,1-beta/2,col="red")
 text(mu1+2,1-beta/2-.03,"beta",col="red")
 segments(mu1,1-beta,mu1,1,col="red")
 lines(x,1-beta.all,lty=1,col="red")
 usr<-par()$usr
 text(mu1,usr[4]*.96,paste("true mean:",mu1),col="red")
} else {
 alpha<-alpha/2
 k.stern<-qnorm(c(1-alpha, alpha),mu0,sdx)
 beta<-pnorm(k.stern[1],mu1,sdx)-pnorm(k.stern[2],mu1,sdx)
 beta.all<-pnorm(k.stern[1],x,sdx)-pnorm(k.stern[2],x,sdx)
 plot(x,fH.x,bty="n",type="l",ylim=c(0,max(fH.x)*1.2),
      ylab="Verteilung PG unter H bzw. G",xlab="Parameter")
 title(paste("H: mean =",mu0,"/  G: mean not equal",mu0))
 lines(x,fG.x,col="red")
 ind<- x>=k.stern[1] | x<=k.stern[2]
 lines(x[ind],fH.x[ind],type="h",col="black")
 lines(x[!ind],fG.x[!ind],type="h",col="red")
 usr<-par()$usr
 segments(mu1,0,mu1,0.9*usr[4],lty=3,col="red")
 text(mu1,usr[4]*.96,paste("true mean",mu1),col="red")
 plot(x,x,type="n",bty="n",ylim=c(0,1.2),
    ylab="WS: Beobachtung in K",xlab="Parameter")
```

```
 title("power")
 segments(mu0,0,mu0,alpha*2)
 arrows(mu0+1.5,alpha,mu0,alpha); text(mu0+2,alpha+.03,"alpha")
 arrows(mu1+1.5,1-beta/2,mu1,1-beta/2,col="red")
 text(mu1+2,1-beta/2-.03,"beta",col="red")
 segments(mu1,1-beta,mu1,1,col="red")
 lines(x,1-beta.all,lty=1,col="red")
 usr<-par()$usr
 text(mu1,usr[4]*.96,paste("true mean",mu1),col="red")
}
par(mfrow=c(1,1))
}
fbut.two.sided<-function(...){
   slider(obj.name="alternative",obj.value="two.sided")
  redo()
}
fbut.less<-function(...){
   slider(obj.name="alternative",obj.value="less")
  redo()
}
fbut.greater<-function(...){
   slider(obj.name="alternative",obj.value="greater")
  redo()
}
slider(redo,c("n","H: mean=mu.0","True mean","alpha"),
  c(2,-5,-5,.01),c(100,5,5,.3),c(1,.1,.1,.01),c(6,1,1.5,.1),
  c(fbut.two.sided,fbut.less,fbut.greater),
  c("G: mean <> mu.0","G: mean < mu.0","G: mean > mu.0"),
title="Guete")
cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

390 ⟨*start* 255⟩+≡

```
conf.test1 <- function(data,hypo,txt="Test per Konfidenzintervalle"){
# no NA expected
data <- sort(data)
n  <- length(data)
k  <- floor(n/2)
ju <- 1:k
jo <- n+1-ju
ma <- max(data)
mi <- min(data)
plot(c(mi,ma),c(1,k),type="n",xlab="data",ylab="k",bty="n",las=1,crt=90)
title(txt)
segments(data[ju],ju,data[jo],ju)
abline(v=hypo,lty=2)
invisible() }
```

391 ⟨*start* 255⟩+≡

```
smirkol <- function(dat1,dat2,PLOT=TRUE,PRINT=FALSE){
h  <- CEV.gr(dat1)
x  <- h[,1]
px <- h[,2]
h  <- CEV.gr(dat2)
y  <- h[,1]
py <- h[,2]
nx <- length(x)
ny <- length(y)
n  <- nx+ny
xy <- c(x,y)
gr <- c(rep(1,nx),rep(2,ny))
gr <- gr[order(xy)]
xy <- sort(xy)
lx <- ly <- 0
ix <- iy <- 1
dd <- 0
di <- id <- c(0,0,0)

GO <- TRUE
if(nx==ny&&sum(x==y)==nx&&sum(px==py)==nx) GO <- FALSE

if(GO) {

for (k in (1:n)) {
  if(gr[k]==1) {
    lx <- px[ix]
    wo <- x[ix]
    di <- c(gr[k],ix,iy)
    ix <- ix+1
  }else{
    ly <- py[iy]
    wo <- y[iy]
    di <- c(gr[k],ix,iy)
    iy <- iy+1 }
  dn <- lx-ly
  if(abs(dn) > dd) {
    dd  <- abs(dn)
    pk  <- c(wo,lx,ly)
    id  <- di }
}
erg <- c(dd,id,pk)

}else{
cat("\nVerteilungen gleich\n")
erg <- NA }

if(PLOT) {
  multiempver(dat1,dat2)
  if(GO) lines(c(erg[5],erg[5]),c(erg[6],erg[7]),col="red",lty=3)}

return(erg)
}
```

392 ⟨*start* 255⟩+≡

```
multiempver <- function(d1,d2,d3,txt=""){
if(missing(d3)) d3 <- d1
if(missing(d2)) d2 <- d1
h1   <- CEV.gr(d1)
h2   <- CEV.gr(d2)
h3   <- CEV.gr(d3)
d1x  <- h1[,1]
d2x  <- h2[,1]
d3x  <- h3[,1]
y1   <- h1[,2]
y2   <- h2[,2]
y3   <- h3[,2]
r1  <- range(d1x)
s1  <- (r1[2]-r1[1])/20
r2  <- range(d2x)
s2  <- (r2[2]-r2[1])/20
r3  <- range(d3x)
s3  <- (r3[2]-r3[1])/20
xma <- max(r1[2],r2[2],r3[2])+5*max(s1,s2,s3)
xmi <- min(r1[1],r2[1],r3[1])-5*max(s1,s2,s3)
plot(c(xma,xmi),c(0,1),type="n",ylab="F",xlab="x",bty="n")
title(paste("Emp. Verteil.:",txt))
segments(c(xmi,d1x),c(0,y1),c(d1x,max(d1x)+xma),c(0,y1),lty=1)
segments(c(xmi,d2x),c(0,y2),c(d2x,max(d2x)+xma),c(0,y2),lty=2)
segments(c(xmi,d3x),c(0,y3),c(d3x,max(d3x)+xma),c(0,y3),lty=3)
#points(d1x,y1,pch=18)
#points(d2x,y2,pch=18)
#points(d3x,y3,pch=18)
invisible() }
```

393 ⟨*start* 255⟩+≡

```
CEV.gr <- function(dat) {
# clear for NA
d1  <- dat[!is.na(dat)]
d1x <- sort(d1)
n1  <- length(d1x)
# clear for multiple values
d2  <- c(d1x[-1],2*d1x[n1])
jj  <- d1x!=d2
y   <- ((1:n1)/n1)[jj]
x   <- d1x[jj]
erg <- cbind(x,y)
return(erg) }
```

394 ⟨*start* 255⟩+≡
```
showlottotabelle <- function(tab){
print(tab[1:10])
print(tab[11:20])
print(tab[21:30])
print(tab[31:40])
print(tab[41:49])}
```

12.1.24 Beispiel Lotto

Verteilung Ziffern

395 ⟨*start* 255⟩+≡
```
"lottab" <-
structure(as.integer(c(307, 322, 321, 300, 305, 322, 305, 285,
314, 299, 309, 302, 250, 292, 300, 294, 311, 315, 311, 298, 322,
307, 290, 297, 316, 322, 319, 278, 303, 295, 316, 359, 320, 279,
310, 321, 311, 345, 315, 308, 310, 320, 311, 308, 277, 304, 299,
326, 346)), .Dim = as.integer(49), .Dimnames = structure(list(
    c("1", "2", "3", "4", "5", "6", "7", "8", "9", "10",
    "11", "12", "13", "14", "15", "16", "17", "18", "19", "20",
    "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
    "31", "32", "33", "34", "35", "36", "37", "38", "39", "40",
    "41", "42", "43", "44", "45", "46", "47", "48", "49")),
.Names = ""), class = "table")
```

Verteilung Lotto alle Ziehungen

396 ⟨*start* 255⟩+≡
```
"kugtab" <-
structure(as.integer(c(52, 53, 57, 45, 49, 45, 39, 49, 53, 46,
53, 51, 47, 44, 51, 49, 61, 54, 56, 53, 51, 48, 52, 57, 42, 59,
57, 40, 42, 52, 58, 58, 55, 42, 51, 57, 61, 61, 48, 57, 55, 54,
46, 44, 54, 59, 57, 41, 51, 54, 59, 48, 55, 48, 41, 60, 53, 49,
56, 67, 49, 49, 41, 46, 57, 53, 46, 40, 47, 45, 50, 44, 45, 47,
49, 49, 44, 48, 36, 55, 79, 70, 45, 46, 49, 49, 55, 61, 47, 50,
60, 44, 56, 42, 50, 58, 63, 62, 53, 51, 53, 57, 46, 70, 50, 43,
44, 48, 36, 57, 42, 48, 48, 46, 54, 55, 58, 55, 53, 46, 42, 46,
44, 53, 52, 51, 53, 53, 60, 57, 64, 50, 54, 54, 36, 64, 58, 48,
48, 63, 54, 59, 30, 47, 47, 59, 57, 47, 57, 68, 44, 52, 44, 61,
47, 44, 48, 46, 50, 39, 46, 33, 50, 55, 49, 60, 44, 56, 56, 43,
53, 60, 50, 59, 53, 51, 51, 50, 59, 49, 54, 48, 55, 49, 55, 50,
61, 46, 39, 62, 63, 54, 52, 36, 46, 72, 56, 48, 45, 45, 51, 64,
43, 48, 58, 61, 48, 43, 28, 55, 61, 40, 47, 50, 47, 53, 64, 56,
56, 49, 60, 58, 52, 55, 57, 51, 45, 54, 43, 43, 63, 53, 52, 45,
45, 48, 68, 54, 60, 47, 44, 41, 55, 59, 48, 45, 54, 50, 54, 59,
58, 52, 45, 66, 40, 59, 52, 45, 58, 61, 52, 41, 61, 50, 46, 53,
51, 53, 47, 63, 53, 50, 35, 52, 52, 48, 52, 39, 45, 48, 53, 64,
65, 53, 47, 43, 50, 45, 39, 53, 55, 46, 58, 56)),
.Dim = as.integer(c(49, 6)),
 .Dimnames = list(c("1", "2", "3", "4", "5", "6", "7", "8",
"9", "10", "11", "12", "13", "14", "15", "16", "17", "18", "19",
"20", "21", "22", "23", "24", "25", "26", "27", "28", "29", "30",
"31", "32", "33", "34", "35", "36", "37", "38", "39", "40", "41",
"42", "43", "44", "45", "46", "47", "48", "49"), c("kug1",
"kug2", "kug3", "kug4", "kug5", "kug6")))
```

12.1.25 Beispiel Kohlengruben

397 ⟨*start* 255⟩+≡

```
"coal"<-structure(list(date = c(1851.2, 1851.63, 1851.97, 1851.97, 1852.31,
1852.35, 1852.36, 1852.39, 1852.98, 1853.2, 1853.23, 1853.32,
1853.5, 1854.13, 1856.4, 1856.51, 1856.54, 1856.62, 1857.14,
1857.4, 1857.58, 1858.09, 1858.15, 1858.41, 1858.95, 1860.13,
1860.17, 1860.59, 1860.85, 1860.92, 1860.97, 1861.18, 1861.74,
1861.84, 1862.14, 1862.89, 1862.94, 1863.18, 1863.79, 1863.94,
1863.99, 1865.46, 1865.97, 1866.06, 1866.34, 1866.45, 1866.83,
1866.95, 1866.95, 1867.64, 1867.85, 1867.86, 1868.75, 1868.9,
1868.99, 1869.25, 1869.44, 1869.55, 1869.81, 1869.87, 1870.12,
1870.52, 1870.56, 1870.63, 1871.03, 1871.15, 1871.17, 1871.74,
1871.82, 1872.12, 1872.24, 1872.77, 1873.14, 1874.29, 1874.55,
1874.89, 1874.98, 1875.33, 1875.93, 1875.93, 1875.93, 1876.97,
1877.06, 1877.11, 1877.19, 1877.78, 1877.81, 1878.18, 1878.2,
1878.24, 1878.43, 1878.7, 1879.04, 1879.17, 1879.5, 1880.06,
1880.54, 1880.69, 1880.94, 1881.11, 1881.97, 1882.13, 1882.3,
1882.3, 1882.33, 1882.85, 1883.8, 1883.85, 1884.07, 1884.86,
1885.17, 1885.46, 1885.98, 1886.62, 1886.69, 1886.75, 1886.92,
1887.13, 1887.41, 1888.3, 1889.05, 1889.2, 1889.79, 1890.1, 1890.19,
1891.25, 1891.67, 1892.65, 1893.51, 1894.48, 1895.32, 1896.07,
1896.28, 1896.33, 1899.63, 1901.39, 1902.67, 1905.06, 1905.19,
1905.52, 1906.77, 1908.14, 1908.27, 1908.63, 1909.13, 1909.83,
1910.36, 1910.97, 1912.52, 1913.78, 1914.41, 1916.62, 1918.03,
1922.53, 1922.68, 1923.57, 1927.16, 1928.11, 1930.15, 1930.75,
1931.08, 1931.83, 1931.88, 1932.07, 1932.86, 1932.88, 1933.88,
1934.72, 1935.64, 1935.7, 1936.6, 1937.5, 1938.35, 1939.82, 1940.22,
1940.42, 1941.42, 1941.52, 1941.57, 1942, 1942.13, 1942.48, 1946.95,
1947.02, 1947.62, 1947.64, 1947.69, 1951.41, 1957.88, 1960.49,
1962.22)), .Names = "date", row.names = c("1", "2", "3", "4",
"5", "6", "7", "8", "9", "10", "11", "12", "13", "14", "15",
"16", "17", "18", "19", "20", "21", "22", "23", "24", "25", "26",
"27", "28", "29", "30", "31", "32", "33", "34", "35", "36", "37",
"38", "39", "40", "41", "42", "43", "44", "45", "46", "47", "48",
"49", "50", "51", "52", "53", "54", "55", "56", "57", "58", "59",
"60", "61", "62", "63", "64", "65", "66", "67", "68", "69", "70",
"71", "72", "73", "74", "75", "76", "77", "78", "79", "80", "81",
"82", "83", "84", "85", "86", "87", "88", "89", "90", "91", "92",
"93", "94", "95", "96", "97", "98", "99", "100", "101", "102",
"103", "104", "105", "106", "107", "108", "109", "110", "111",
"112", "113", "114", "115", "116", "117", "118", "119", "120",
"121", "122", "123", "124", "125", "126", "127", "128", "129",
"130", "131", "132", "133", "134", "135", "136", "137", "138",
"139", "140", "141", "142", "143", "144", "145", "146", "147",
"148", "149", "150", "151", "152", "153", "154", "155", "156",
"157", "158", "159", "160", "161", "162", "163", "164", "165",
"166", "167", "168", "169", "170", "171", "172", "173", "174",
"175", "176", "177", "178", "179", "180", "181", "182", "183",
"184", "185", "186", "187", "188", "189", "190", "191"), class = "data.frame")
```

12.1.26 Daten: Zwiebelbeispiel

Gruppenaufteilung

398 ⟨*start* 255⟩+≡
```
gender <- c(1, 0, 1, 0, 1, 1, 0, 1, 1, 0, 1, 1, 0, 0, 1,
1, 1, 0, 0, 1, 0, 0, 0)
ind <- length(gender)
i1  <- (1:ind)[!!gender]
i0  <- (1:ind)[!gender]
```

Kartoffeln

399 ⟨*start* 255⟩+≡
```
potato.v <- c(77.33, 83.93, 71.61, 53.76, 62.37,
 83.56, 82.25, 76.96, 96.34,106.62,
113.71, 84.83, 70.15, 72.11, 70.85,
 86.15, 62.33, 77.77, 69.33, 63.79,
 75.02, 82.21, 82.76, 81.11, 75.91,
 94.90, 81.40, 75.88, 87.45, 75.81,
121.73,108.00,106.87,102.64,122.50,
 79.99, 68.42, 71.30, 66.62, 67.46,
101.26, 71.39, 86.97, 82.46, 94.2,
 78.38, 75.24, 75.02, 81.06, 60.11,
 78.51, 65.68, 65.04, 69.21, 55.28,
 70.13, 64.62, 57.50, 70.69, 66.53,
 91.52, 94.30,119.63, 84.34,115.89,
 96.66, 87.99, 89.47, 73.16, 75.57,
 60.53, 55.12, 50.42, 59.29, 54.74,
 76.83, 85.61, 75.31, 84.04, 89.69,
 87.25, 77.69, 80.87, 92.01, 78.74,
142.65,100.89,143.49, 97.04,139.18,
 66.37,104.26, 98.65, 80.40, 64.00,
 82.23, 98.70, 81.18, 82.06, 64.19,
    NA,    NA,    NA,    NA,    NA,
 90.22, 62.08, 71.07, 72.71, 70.27,
 72.67, 96.29, 82.36, 78.13, 69.28)
potato.m <- matrix(potato.v,23,5 ,byrow=TRUE)
po1.m <- potato.m[i1,]
po0.m <- potato.m[i0,]
po1.v <- as.vector(t(po1.m))
po0.v <- as.vector(t(po0.m))
mpo   <- rowMeans(potato.m)
mpo1  <- rowMeans(po1.m)
mpo0  <- rowMeans(po0.m)
mmpo0 <- mean(mpo0,na.rm=TRUE)
mmpo1 <- mean(mpo1,na.rm=TRUE)
mmpo  <- mean(mpo,na.rm=TRUE)
```

400 ⟨*start* 255⟩+≡

```
police <-
structure(c(79.1, 163.5, 57.8, 196.9, 123.4, 68.2, 96.3, 155.5,
85.6, 70.5, 167.4, 84.9, 51.1, 66.4, 79.8, 94.6, 53.9, 92.9,
75, 122.5, 74.2, 43.9, 121.6, 96.8, 52.3, 199.3, 34.2, 121.6,
104.3, 69.6, 37.3, 75.4, 107.2, 92.3, 65.3, 127.2, 83.1, 56.6,
82.6, 115.1, 88, 54.2, 82.3, 103, 45.5, 50.8, 84.9, 58, 103,
45, 149, 109, 118, 82, 115, 65, 71, 121, 75, 67, 62, 57, 81,
66, 123, 128, 113, 74, 47, 87, 78, 63, 160, 69, 82, 166, 58,
55, 90, 63, 97, 97, 109, 58, 51, 61, 82, 72, 56, 75, 95, 46,
106, 90), .Dim = as.integer(c(47, 2)))

bekleidung <-
structure(c(32, 12, 81, 28, 287, 51, 345, 56, 181, 62, 5, 46,
34, 10, 219, 71, 102, 126, 61, 151, 118, 124, 131, 310, 80, 328,
213, 305, 423, 7, 45, 140, 24, 133, 89, 172, 81, 44, 84, 71,
81, 155, 18, 29, 17, 93, 29, 45, 35, 56, 68, 31, 34, 16, 15,
29, 7, 20, 18, 14, 51, 21, 117, 174, 201, 287, 88, 122, 57, 41,
4, 11, 16, 62, 62, 15, 5, 16, 9, 15, -1, 1, -4, 13, 17, 16, -19,
-2, 17, 23, 20, 25, 4, 0, 0, 3, 17, 8, 13, -1, -5, -6, 34, 27,
21, 22, 9, 36, 31, 30, -1, 2, -2, -16, 0, 75, 17, -13, 1, 3,
8, 7, 23, 5, 9, 13, 19, 20, 36, 35, 50, 51, 46, 42, 24, 22, 29,
30, 15, 17, 27, 31, 2, 30, 54, 57), .Dim = as.integer(c(73, 2
)))
```

401 ⟨*old/defekt* 401⟩≡

```
exp.regr.poly<-function(x,y){
  replot<-function(...){
    pg<-slider(no=1)
    formula<-paste(c(
          "x","+I(x^2)","+I(x^3)","+I(x^4)","+I(x^5)"
             ,"+I(x^6)","+I(x^7)","+I(x^8)","+I(x^9)"
          )[1:pg],collapse="")
    xy<-as.data.frame(cbind(x=x,y=y))
    code<-paste("lm(y~",formula,",xy)")
    result<-eval(parse(text=code))
    plot(x,y)
    xx<-seq(min(x),max(x),length=100)
    yy<-cbind(1,xx,xx^2,xx^3,xx^4,xx^5,xx^6,xx^7,xx^8,xx^9
             )[,1:(1+pg)]%*% result$coef
    lines(xx,yy,col="red")
    title(paste("Polynomgrad:",pg))
  }
  slider(replot,"Polynomgrad",1,9,1,1)
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
exp.regr.poly<-function(x,y){
  replot<-function(...){
    pg<-slider(no=1)
    formula<-paste(c("x","+I(x^2)","+I(x^3)","+I(x^4)","+I(x^5)"
                            ,"+I(x^6)","+I(x^7)","+I(x^8)","+I(x^9)"
                              )[1:pg],collapse="")
    #xy<-as.data.frame(cbind(x=x,y=y))
    #code<-paste("lm(y~",formula,",xy)")
    result<-eval(parse(text=paste("lm(",formula,")")))
    plot(x,y)
    xx<-seq(min(x),max(x),length=100)
    yy<-cbind(1,xx,xx^2,xx^3,xx^4,xx^5)[,1+gp]%*% result$coef
    lines(xx,yy,col="red")
    title(paste("Polynomgrad:",pg))
  }
  slider(replot,"Polynomgrad",1,10,1,1)
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

402 ⟨*start* 255⟩+≡

```
exp.regr.poly<-function(x,y){
  replot<-function(...){
    pg<-slider(no=1)
    formula<-paste(c("y~x","+I(x^2)","+I(x^3)","+I(x^4)","+I(x^5)"
                             ,"+I(x^6)","+I(x^7)","+I(x^8)","+I(x^9)"
                               )[1:pg],collapse="")
    result<-eval(parse(text=paste("lm(",formula,")")))
    plot(x,y)
    xx<-seq(min(x),max(x),length=100)
    yy<-cbind(1,xx,xx^2,xx^3,xx^4,xx^5,xx^6,xx^7,
          xx^8,xx^9)[,1:(1+pg)]%*% result$coef
    lines(xx,yy,col="red")
    title(paste("Polynomgrad:",pg))
  }
  slider(replot,"Polynomgrad",1,9,1,1)
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}

exp.regr.smooth<-function(x,y){
  replot<-function(...){
    f<-slider(no=1)
    plot(x,y)
    lines(lowess(x,y,f),col="red")
       title(paste("Fensterbreite:",f))
  }
  slider(replot,"Fensterbreite",0,1,.01,.1)
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
plot.lsline<-function(x,y,main=""){
  result<-lm(y~x)
  a.dach<-result$coef[1]; b.dach<-result$coef[2]
  plot(c(x,0),c(y,0),type="n",main=paste("\n\n",main))
  points(x[ind],y[ind])
  abline(a.dach,b.dach,col="blue")
  ⟨notiere im Titel â und b̂ 407⟩
  cat("Plot erstellt\n"); NULL
}
```

403 ⟨*berechne Maßzahlen zur Regression von* y *auf* x 403⟩≡

```
# Summen
sumx <-sum(x); sumy<-sum(y);sumxx<-sum(x*x);sumyy<-sum(y*y);sumxy<-sum(x*y)
meanx<-mean(x);meany<-mean(y);meanxx<-mean(x*x);meanxy<-mean(x*y)
sxx  <-var(x);   syy  <-var(y);  sxy <-(sum((x-meanx)*(y-meany)))/(n-1)
# Regressionsgroessen
r <-sxy/sqrt(sxx * syy); rq <-r * r
bdach<-(meanxy - meany * meanx)/(meanxx - meanx * meanx)
adach<- meany -  bdach * meanx;
ydach <-adach + bdach * x; udach <-y - ydach; sumuu <-sum(udach * udach);
# Varianzen
sigdach<-(sigqdach<-(sumuu)/(n - 2))^0.5
sig.a.dach<-(sigq.a.dach <-(sigqdach * sumxx)/(n * (n - 1) * sxx))^0.5
sig.b.dach<-(sigq.b.dach <- sigqdach         /(    (n - 1) * sxx))^0.5
```

404 ⟨*start* 255⟩+≡

```
F.byhand<-function(x,y){
  if(is.matrix(x)) {y<-x[,2];x<-x[,1]}
  if((n<-length(x))!= length(y)) return("ERROR: Laengen passen nicht!")
  ⟨berechne Maßzahlen zur Regression von y auf x 403⟩
  mse.regr<- sum((ydach-meany)^2)/(1)
  mse.resid<-sum((y-ydach)^2)/(n-2)
  F<-mse.regr/mse.resid
  pvalue<-1-pf(F,1,n-2)
  return(list(mse.regr=mse.regr,mse.resid=mse.resid,F=F,pvalue=pvalue))
}
ki.y.dach<- function(x, y, alpha = 1/20, x0) { ###alt
  if(is.matrix(x)) {y<-x[,2];x<-x[,1]}
  if((n<-length(x))!= length(y)) return("ERROR: Laengen passen nicht!")
  ⟨berechne Maßzahlen zur Regression von y auf x 403⟩
# t-Wert
  twert <-qt(1 - alpha/2, n - 2)
  twertsim <-qt(1 - alpha/4, n - 2)
# KIs
  if(!missing(x0)){
        ki.g.x0 <-adach+bdach*x0+c(-1,1)*twert*
                  sqrt(sigqdach*(1/n+(meanx-x0)^2/((n-1)*var(x))))
  } else ki.g.x0<-NULL
  x0<-seq(min(x), max(x), length=100)
  step<-twert*sqrt(sigqdach*(1+1/n+(meanx-x0)^2/((n-1)*var(x))))
  ki.y.x0.u <-adach+bdach*x0 - step
  ki.y.x0.o <-adach+bdach*x0 + step
  plot(x,y)
  abline(adach,bdach,col="blue")
  lines(x0, ki.y.x0.u, col="red" )
  lines(x0, ki.y.x0.o, col="red" )
  ki.g.x0
}
ki.y.dach<- function(x, y, alpha = 1/20, x0) {
  if(is.matrix(x)) {y<-x[,2];x<-x[,1]}
  if((n<-length(x))!= length(y)) return("ERROR: Laengen passen nicht!")
  ⟨berechne Maßzahlen zur Regression von y auf x 403⟩
# t-Wert
  twert <-qt(1 - alpha/2, n - 2)
  twertsim <-qt(1 - alpha/4, n - 2)
# KIs
#  if(!missing(x0)){
#        ki.g.x0 <-adach+bdach*x0+c(-1,1)*twert*
#                  sqrt(sigqdach*(1/n+(meanx-x0)^2/((n-1)*var(x))))
#  } else ki.g.x0<-NULL
  xlim<-range(c(x,x0))
  ylim<-range(c(adach+bdach*xlim,y))
  plot(x,y,xlim=xlim,ylim=ylim)
  abline(adach,bdach,col="blue")
  if(missing(x0)) x0<-seq(min(x), max(x), length=100)
  for(i in seq(twert)){
    step<-twert[i]*sqrt(sigqdach*(1+1/n+(meanx-x0)^2/((n-1)*var(x))))
    ki.y.x0.u <-adach+bdach*x0 - step
    ki.y.x0.o <-adach+bdach*x0 + step
    lines(x0, ki.y.x0.u, col="red" )
    lines(x0, ki.y.x0.o, col="red" )
  }
#  ki.g.x0
```

```
}
ki.a.b<-function(x,y,alpha=0.05,plot=TRUE){
  if(is.matrix(x)) {y<-x[,2];x<-x[,1]}
  if((n<-length(x))!= length(y)) return("ERROR: Laengen passen nicht!")
  ⟨berechne Maßzahlen zur Regression von y auf x 403⟩
# t-Wert
  twert <-qt(1 - alpha/2, n - 2)
  twertsim <-qt(1 - alpha/4, n - 2)
# KIs
  ki.a <-adach + c(-1, 1) * twert * sqrt(sigq.a.dach)
  ki.b <-bdach + c(-1, 1) * twert * sqrt(sigq.b.dach)
  ki.a.sim <-adach + c(-1, 1) * twertsim * sqrt(sigq.a.dach)
  ki.b.sim <-bdach + c(-1, 1) * twertsim * sqrt(sigq.b.dach)
  result<-rbind(ki.a,ki.b,ki.a.sim,ki.b.sim)
  dimnames(result)<-
         list(c(paste("  ",(1-alpha)*100,"%-KI->",
              c("a","b","a simultan","b simultan"),sep="")),
              c("Untergrenze:","Obergrenze:"))

  if(plot==TRUE){
    plot(x,y)
    abline(lm(y~x)$coef,col="blue"); h<-(mean(x)<0)
    abline(a=result[3,1],b=result[4,1+h],col="red")
    abline(a=result[3,2],b=result[4,2-h],col="red")
    result
  }
}
lsfit.b.pvalue<-function(x,y){
  if(is.matrix(x)) {y<-x[,2];x<-x[,1]}
  if((n<-length(x))!= length(y)) return("ERROR: Laengen passen nicht!")
  ⟨berechne Maßzahlen zur Regression von y auf x 403⟩
# p-value
  if(bdach>0){
    p.value<-2*pt(-bdach/sigq.b.dach^0.5,n-2)
  } else {
    p.value<-2*(1-pt(-bdach/sigq.b.dach^0.5,n-2))
  }
  p.value
}
exp.Rq.outlier<-function(){
 replot<-function(...){
  n<-20; a<-1; b<-1
  set.seed(13)
  x<-runif(n-1,-1,1); y<-runif(n-1,-1,1)
  distance<-slider(no=1); alpha<-(slider(no=2)-3)*2*pi/12
  x<-c(distance*cos(alpha),x)
  y<-c(distance*sin(alpha),y)
  lim<-max(c(x,distance))
  xlim<-c(-lim,lim)
  plot(x,y,xlim=xlim,ylim=xlim);
  abline(lm(y~x),col="blue")
  points(x[1],y[1],col="red")
  title("R^2: Ausreisserempfindlichkeit")
  title(paste("\n\nR^2=",signif(cor(x,y)^2,4)))
 }
 slider(replot,c("Entfernung","Richtung"),
              c(0,-6),c(50,18),c(.2,0.25),c(5,1.5))
 cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
```

```
}
exp.adjust.Rq<-function(){
  replot<-function(...){
    n<-slider(no=6); seed<-slider(no=7); set.seed(seed)
    mu1<-slider(no=1);    mu2<-slider(no=2)
    sd1<-slider(no=3);    sd2<-slider(no=4)
    sdxy<-slider(no=5)/10*(sd1*sd2)^0.5
    x<-rnorm(n); y<-rnorm(n)
    xy<-cbind(x,y)%*%matrix(c(sd1,sdxy,sdxy,sd2),2,2)
    x<-xy[,1]+mu1; y<-xy[,2]+mu2
    plot(x,y); abline(lm(y~x),col="blue")
    title(paste("R^2 =",signif(cor(x,y)^2,4)))
  }
  #  ... 7.3, 100, 13 -> .90
  slider(replot,
         c("Parameter 1","Parameter 2","Parameter 3",
           "Parameter 4","Parameter 5","n","seed"),
         c(-10,-10,0,0,-10,10,1),c(10,10,10,10,10,200,100),
         c(.1,.1,.1,.1,.1,1,1),c(1,1,1,1,0,50,13))
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

405 ⟨*start* 255⟩+≡
⟨*definiere* `show.rq` 406⟩

406 ⟨*definiere* `show.rq` 406⟩≡

```
show.rq<-function(x,y){
  plot(x,y)
  result<-lm(y~x); abline(result,col="blue")
  a.dach<-result$coef[1]; b.dach<-result$coef[2]
  y.dach<-a.dach+b.dach*x
  xmin<-par()$usr[1]; xmax<-par()$usr[2]
  arrows(x,y,xmin,y,col="red",lty=2)
  segments(x,y,x,y.dach,col="black",lty=3)
  arrows(x,y.dach,xmax,y.dach,col="blue",lty=1)
  Rq<-var(y.dach)/var(y)
  title(paste("R^2 =",signif(Rq,3),
            "\nvar(y.dach) =",signif(var(y.dach),3),
            ";  var(y)=",signif(var(y),3)))
}
```

407 ⟨*notiere im Titel $\hat{a}$ und $\hat{b}$* 407⟩≡

```
tit<-substitute(paste(hat(a),"=",value0,
                ",  ",hat(b),"=",value1),
  list(value0=signif(a.dach,3),value1=signif(b.dach,3) )
)
title(tit)
```

408 ⟨*start* 255⟩+≡

```
exp.fit.line.to.poly<-function(){
  replot<-function(...){
    n<-slider(no=1);
    sigma.u<-slider(no=2)
    set.seed(13)
    p.coef<-c(slider(no=3),slider(no=4),
              slider(no=5),slider(no=6))
    x<-runif(n,0,100); u<-rnorm(n,0,sigma.u)
    y<-p.coef[1]+p.coef[2]*x+p.coef[3]*x^2+p.coef[4]*x^3+u
    result<-lm(y~x)
    par(mfrow=c(2,1))
    plot(x,y)
    abline(result,col="blue")
    title(paste(  "a.dach=",signif(result$coef[1],5),
                ", b.dach=",signif(result$coef[2],5),
               ", sigma.U=",sigma.u,
    "\ny=",p.coef[1],"*x^0+",p.coef[2],"*x^1+",
           p.coef[3],"*x^2+",p.coef[4],"*x^3+e",sep=""))
     plot(x,result$resid,type="h",main="Residualplot")
    par(mfrow=c(1,1))
  }
  slider(replot,c("n","sigma.U","a0","a1","a2","a3"),
            c(20,50,-20000,-1000,-100,-1),
            c(200,10000,20000,1000,100,1),
            c(10,50,1000,100,10,.1),
            c(20,50,20000,-14,-50,.5))
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
exp.check.point.influence<-function(x,y,seed=13){
  if(missing(x)) {
   n<-10; set.seed(seed)
   x<-runif(n,0,10); u<-rnorm(n,0,1)
   y<-2+x+u
  }
  n<-length(x)
  delta.x<-diff(range(x));   delta.y<-diff(range(y))
  xmin<-min(x)-delta.x; ymin<-min(y)-delta.y
  xmax<-max(x)+delta.x; ymax<-max(y)+delta.y
  slider(obj.name="j",obj.value=1)
  slider(obj.name="x",obj.value=x)
  slider(obj.name="y",obj.value=y)
  replot<-function(...){
    j.alt<-slider(obj.name="j"); j<-slider(no=1)
    x<-slider(obj.name="x");  y<-slider(obj.name="y")
    if(j.alt!=j){
      slider(set.no.value=c(2,x[j]))
      slider(set.no.value=c(3,y[j]))
      slider(obj.name="j",obj.value=j)
    }
    xneu<-slider(no=2); yneu<-slider(no=3)
    x[j]<-xneu; y[j]<-yneu
    slider(obj.name="x",obj.value=x)
    slider(obj.name="y",obj.value=y)
    result<-lm(y~x)
    par(mfrow=c(1,2))
    plot(x,y,type="n",
         xlim=c(xmin,xmax),ylim=c(ymin,ymax))
```

```
    abline(result,col="blue")
    text(x,y,as.character(1:n))
    text(x[j],y[j],as.character(j),col="red")
    title(paste(  "a.dach=",signif(result$coef[1],2),
                ", b.dach=",signif(result$coef[2],2),sep=""))
    plot(x,result$resid,type="h",main="Residualplot")
    points(x[j],result$resid[j],type="h",col="red")
    par(mfrow=c(1,1))
  }
  slider(replot,c("Punkt?","neuer x-Wert?","neuer y-Wert?"),
           c(1, xmin,ymin),
           c(n, max(x)+delta.x,max(y)+delta.y),
           c(1, xmin,ymin),
           c(1,x[1], y[1]))
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
# exp.check.point.influence()
```

409 ⟨* 5⟩+≡

```
plot(1); b<-2
a<-quote(hat(a)==substitute(expression(b)))
a<-substitute(paste(b, "\n ", Delta, " values"),
              list(b =8 ))
tit<-substitute(paste(hat(beta)[0],"=",value0,
              ",  ",hat(b),"=",value1),
              list(value0=signif(result$coef[1],3),
                   value1=signif(result$coef[2],3) ))
#title(quote(hat(a)==0))#title(quote(hat(a)==0))
title(a)
```

410 ⟨* 5⟩+≡

```
set.seed(72); ind<-sample(seq(x),size=8)
xx<-x[ind]; yy<-y[ind]
xx<-(xx-min(xx))/(max(xx)-min(xx))/2+.25
yy<-(yy-min(yy))/(max(yy)-min(yy))/2+.25
b.dach<-sum((xx-mean(xx))*(xx-mean(xx))) / sum((xx-mean(xx))^2)
a.dach<-mean(yy)-b.dach*mean(xx)
plot(xx,yy,axes=F,xlim=0:1,ylim=0:1,xlab="",ylab="")
abline(a.dach,b.dach,col="blue")
res<-yy-(a.dach+b.dach*xx)
for(i in seq(xx))
  polygon(c(xx[i],xx[i],xx[i]+abs(res[i]),xx[i]+abs(res[i])),
          c(yy[i],yy[i]+(-res[i]),yy[i]+(-res[i]),yy[i]),col="red")
```

411 ⟨*start* 255⟩+≡

```
milchprod<-structure(c(94, 86, 91, 91, 240, 292, 68, 238, 64, 333, 244,
276, 347, 200, 290, 138, 130, 124, 142, 34, 32, 45, 52, 48, 20,
40, 33, 17, 30, 31, 33, 32, 35, 32, 27, 24, 41, 10, 34, 25, 31,
26, 18, 95, 65, 28, 24, 28, 24, 240, 300, 35, 237, 35, 250, 240,
280, 210, 180, 200), .Dim = as.integer(c(15, 4)), .Dimnames = list(
    c("bauer-vanille","bauer-himbeer","bauer-zitrone","bauer-pfirsich-maracuya",
    "schmand", "semi-h-schlagsahne", "weihenstephan-joghurt-mild",
    "hollandaise", "vollmilch-lippe", "mibell-pizzakaese", "creme-frischli",
    "ringo-sahne", "pfannenkaese", "buko-frischkaese-kraeuter",
    "philadelphia-doppelrahmstufe"), c("kcal", "kohlehydrate",
    "eiweiss", "fett")))
milchprod<-as.data.frame(milchprod)
colnames(milchprod)<-c("BW","KH","EW","FE")
astra.liter <-
c(38.4, 32, 26.1, 41, 41.9, 38, 34.2, 23.2, 43.2, 41.9, 36.8,
20.6, 30, 43.2, 31.4, 30, 42.6, 29.2, 41.2, 37.8, 20.5, 42.5,
38.5, 36, 41, 27.4, 41.8, 37.1, 42, 42.2, 28.6, 43.8, 38, 25.6,
37.4, 30.7, 33, 40.3, 29.5, 36, 31.1, 42.5, 41.7, 30.3, 28.9,
35.6, 38.3, 38.4, 31.1, 34.1, 38.4, 29.4, 27, 21, 42, 40.3, 24.8,
31.5, 26.4, 40, 35, 22.6, 14, 41.5, 42.6, 37.6, 37.5, 39.1, 39,
39.1, 30.9, 22.9, 20, 36, 40.2, 31.3, 40.1, 33.3, 41.6, 39.1,
38.8, 33.5, 40, 30.9, 22, 41, 41.9, 42.2, 40, 22.1, 16.5, 33,
35.4, 15.3, 11.7, 21.1, 24.4, 39.3, 37.7, 41.2, 34, 38, 41, 41.2,
38.5, 27.9, 40.8)
astra.km <-
c(12, 454, 370, 520, 514, 577, 428, 411, 694, 607, 569, 298,
614, 460, 401, 582, 372, 306, 532, 441, 260, 538, 955, 469, 491,
345, 615, 519, 651, 620, 391, 597, 552, 358, 456, 471, 540, 452,
383, 483, 433, 558, 475, 410, 255, 600, 416, 538, 433, 502, 492,
563, 324, 419, 513, 527, 313, 470, 379, 672, 384, 295, 194, 519,
494, 462, 465, 556, 516, 574, 408, 352, 331, 501, 567, 441, 603,
452, 529, 551, 490, 407, 556, 473, 311, 639, 597, 573, 612, 278,
264, 489, 652, 216, 180, 328, 484, 511, 464, 544, 396, 487, 529,
524, 523, 384, 630)
fbe.fit<-function(x,y){
  replot<-function(...){
    x1<-slider(no=1);    y1<-slider(no=2)
    x2<-slider(no=3);    y2<-slider(no=4)
    b<-(y2-y1)/(x2-x1); a<-y1-b*x1
    xlim<-range(c(xlim,0)); ylim<-range(c(ylim,0))
    plot(x,y,xlim=xlim,ylim=ylim,
         main=paste("Achsenabschnitt:",signif(a,4),
                    "\nSteigung:",signif(b,4)))
    abline(a,b,col="red")
    points(x1,y1,col="red",pch=16,cex=2)
    points(x2,y2,col="red",pch=16,cex=2)
  }
  xlim<-range(x); ylim<-range(y)
  dx<-diff(xlim);dy<-diff(ylim)
  xlim<-xlim+c(-.5,.5)*dx
  ylim<-ylim+c(-.5,.5)*dy
  slider(replot,c("x1 : x-Wert Punkt 1","y1 : y-Wert Punkt 1",
                  "x2 : x-Wert Punkt 2","y2 : y-Wert Punkt 2"),
         c(xlim[1],ylim[1],xlim[1],ylim[1]),
         c(xlim[2],ylim[2],xlim[2],ylim[2]),
         0.001*c(dx,dy,dx,dy),
         c(xlim[1],ylim[1],xlim[2],ylim[2]),
```

```
        title="fbe-fit: by fixing two points")
  replot()
  cat("Demo gestartet, siehe Steuerungsfenster\n"); NULL
}
```

412 ⟨*verschiedene Residualplots* 412⟩≡

```
set.seed(13)
n<-100
x<-sort(runif(n)); y<-rnorm(n)
par(mfrow=c(2,2))
yy<-y
resid<-lm(yy~x)$resid
plot(x,resid,type="h",axes=F,xlab="",ylab="",
 main="strukturloser Residualplot")
yy<-y;yy[4]<-y-20
resid<-lm(yy~x)$resid
plot(x,resid,type="h",axes=F,xlab="",ylab="",
 main="Ausrei\337er")
yy<-y*20*x
resid<-lm(yy~x)$resid
plot(x,resid,type="h",axes=F,ylab="",
 main="var(u) mit x steigend")
yy<-y+20*(x-.5)^2
resid<-lm(yy~x)$resid
plot(x,resid,type="h",axes=F,ylab="",
 main="x<->y nicht linear")
```

413 ⟨* 5⟩+≡

```
cat("Zahl zwischen 1 und 5 eingeben!")
input<-readline()
input<-as.numeric(input)
if(is.na(input)) cat("war keine Zahl!\n")
```

414 ⟨* 5⟩+≡

```
x<-c(67,45,54,62,36,51,27,116,0,29,48,16,83,24,39,
23,79,0,107,18,49,59,9,43,34,43,56,15,28,67,27,
25,42,49,39,64,42,36,42,0,98,0,44,37,58,54,38,49,58,51)
```

415 ⟨* 5⟩+≡

```
tsu<-cbind(c(14.0,.2,10,26.2,10,.1,9.5,7.3,2.9,7.7,5.1,15,2.5,6.6,4.0),
   c(103,1,169,2080,330,7,223,87,9,108,12,2182,150,5,96))
#tsu<-rbind(tsu,cbind(xy[!(xy[,11] %in% tsu[,1]),11],0))
#plot(log(tsu))
#abline(lsfit(log(tsu)[,1],log(tsu)[,2]))
plot(tsu);abline(lsfit(tsu[,1],tsu[,2]))
ind<-tsu[,2]<1000
#ind<-seq(tsu[,1])
x<-tsu[ind,1]; y<-tsu[ind,2]
y<-y; x<-(x)
plot(x,y); abline(lsfit(x,y))
coef<-(lm(y~x+I(x^2)#+I(x^3)#+I(x^4)
))$coef
#coef<-(lm(y~x+I(x^2)+I(x^3)+I(x^4)))$coef
xx<-seq(min(x),max(x),length=100)
yy<-coef[1]+coef[2]*xx+coef[3]*xx^2#+coef[4]*xx^3+coef[5]*xx^4
lines(xx,yy)
res<-lsfit(x,y)
names(res)
```