R PROGRAMMING FOR MIPS

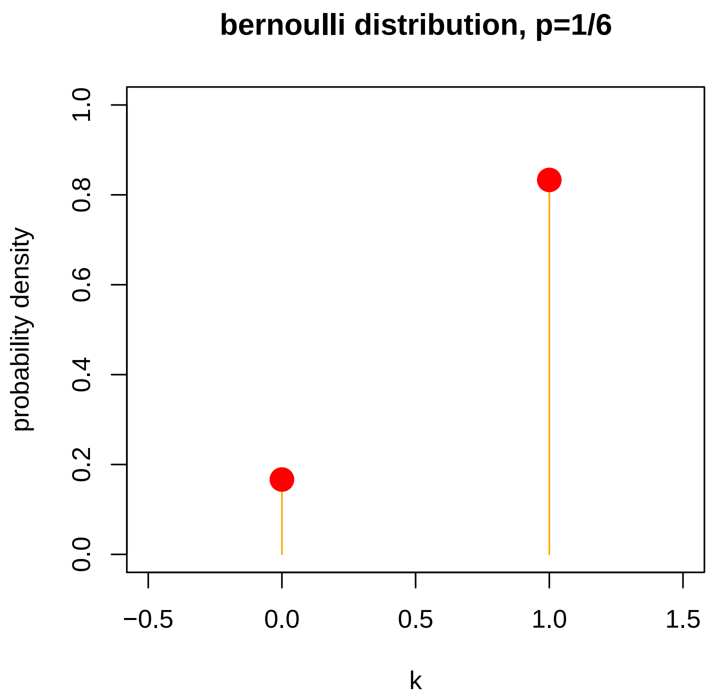
CHRIS PARRISH

ABSTRACT. These notes illustrate the use of the R programming language and programming environment for constructing short demos in support of a class on statistical modeling. The examples track discussions in the text *A Modern Introduction to Probability and Statistics, Understanding Why and How*, by F.M. Dekking, et al., published in the series "Springer Texts in Statistics" by Springer-Verlag, 2005, ISBN 1-852-33896-2, and they make use of the data sets which accompany that text.

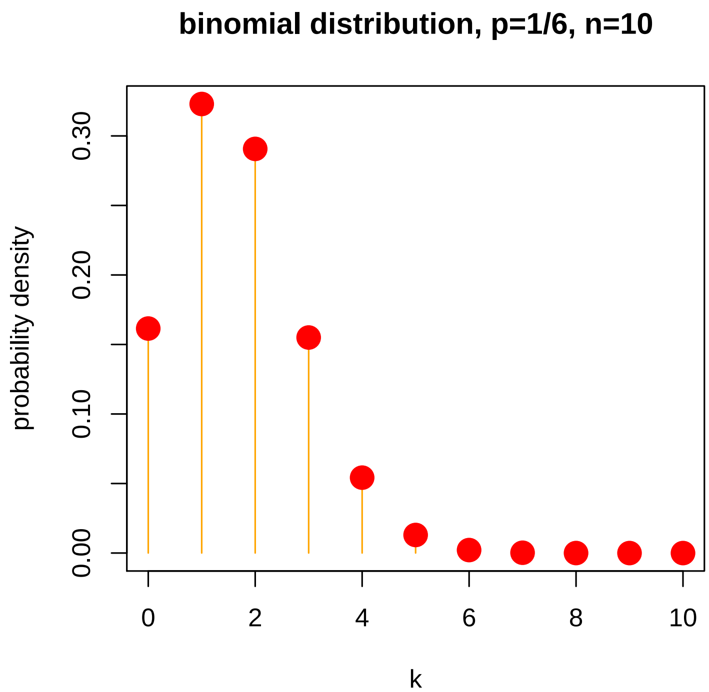
Date: August 21, 2006.

Chapter 4: Discrete random variables

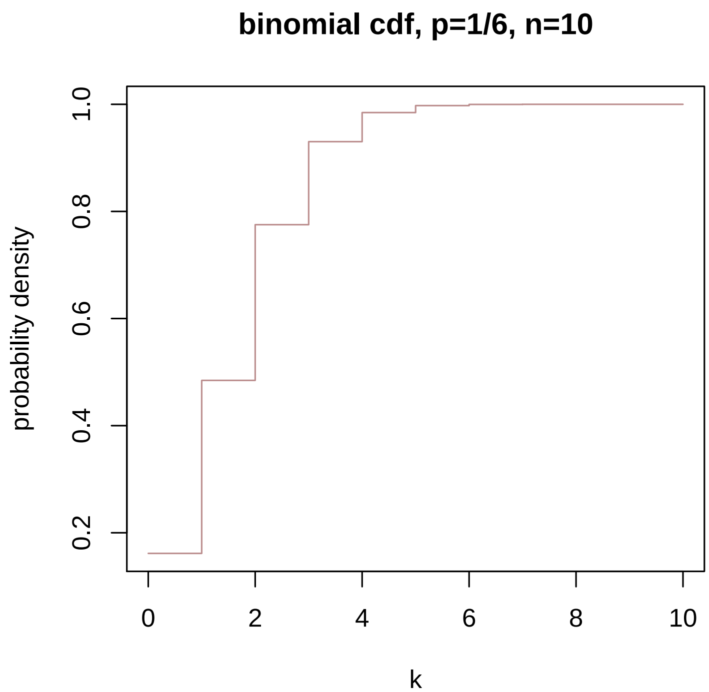
```
> p=1/6
> heights=c(p,q=1-p)
> plot(0:1,heights,type="h",
       xlim=c(-.5,1.5),ylim=c(0,1),
       main="bernoulli distribution, p=1/6",col="orange",
       xlab="k",ylab="probability density")
> points(0:1,heights,
         pch=16,cex=2,col="red")
```



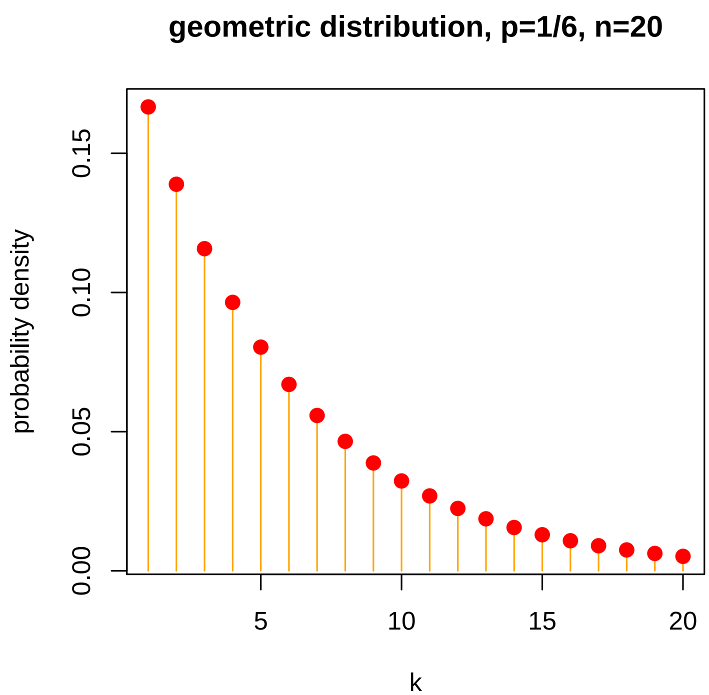
```
// binomial distribution

> p=1/6; n=10
> heights=dbinom(0:10,size=n,p)
> plot(0:10,heights,type="h",
       main="binomial distribution, p=1/6, n=10",col="orange",
       xlab="k",ylab="probability density")
> points(0:10,heights,
         pch=16,cex=2,col="red")
```



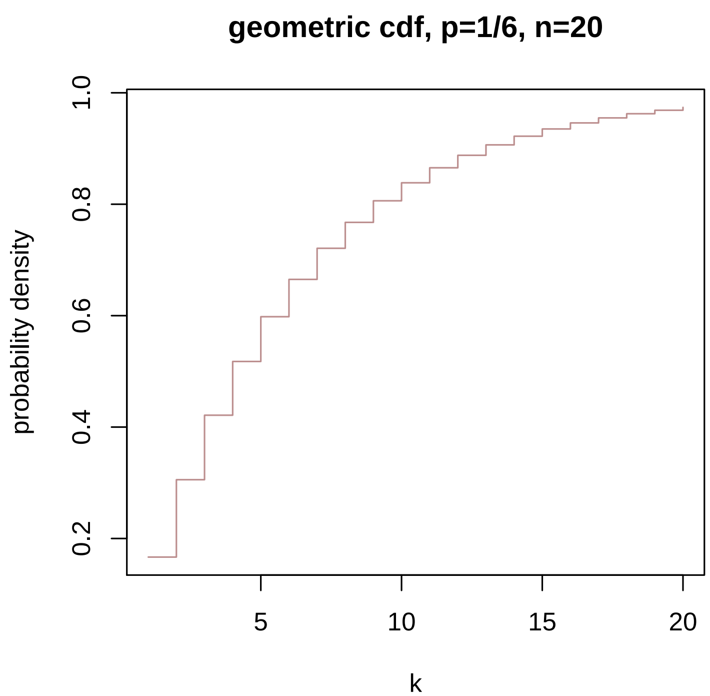
```
// binomial cdf

> heights=pbinom(0:10,size=n,p)
> plot(0:10,heights,type="s",
       main="binomial cdf, p=1/6, n=10",col="rosybrown",
       xlab="k",ylab="probability density")
```



```
// geometric distribution

> p=1/6; n=20
> dgeo <- function(k,p){(1-p)^(k-1)*p}
> heights= dgeo(1:n,p)
> plot(1:n,heights,type="h",
      main="geometric distribution, p=1/6, n=20",col="orange",
      xlab="k",ylab="probability density")
> points(1:n,heights,
        pch=16,cex=1.2,col="red")
```

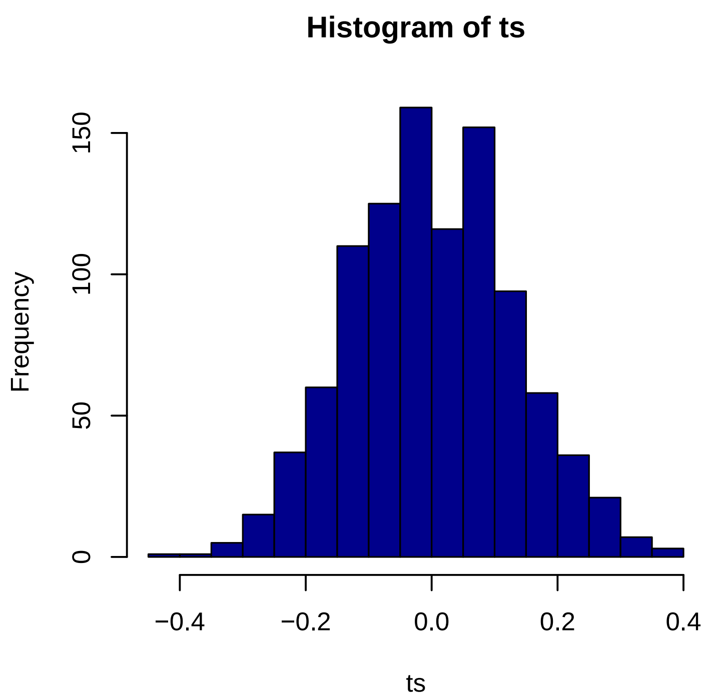


```
// geometric cdf

> pgeo <- function(k){sum(dgeo(1:k,p))}
> heights= lapply(1:n,pgeo)
> plot(1:n,heights,type="s",
       main="geometric cdf, p=1/6, n=20",col="rosybrown",
       xlab="k",ylab="probability density")
```

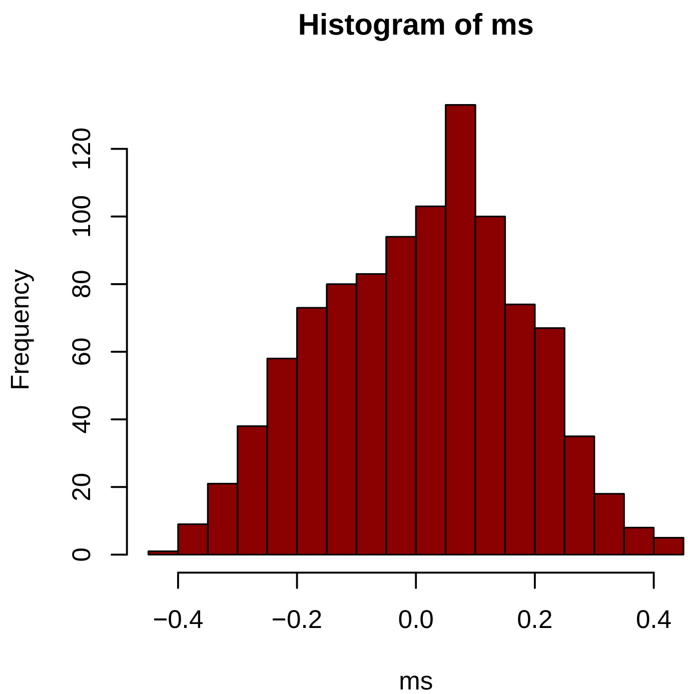


Chapter 6: Simulation

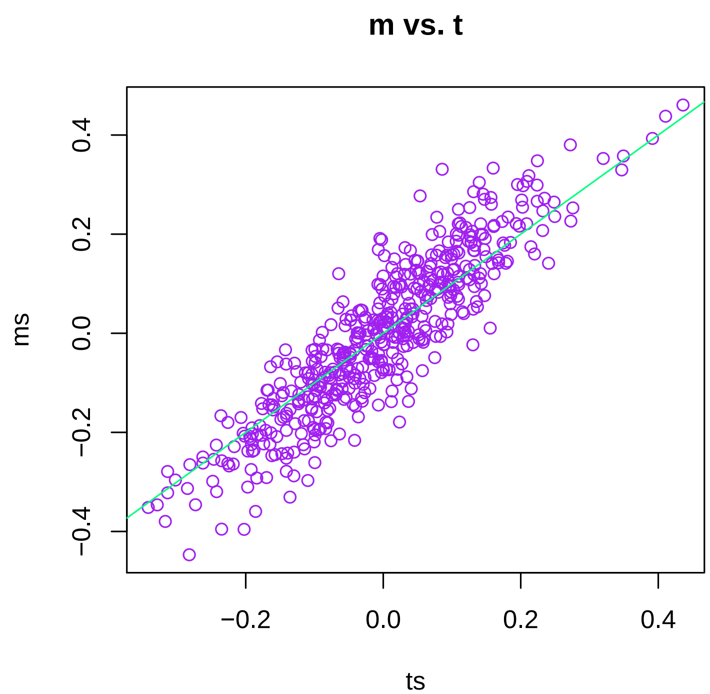
```
// two jury rules: t

> a=-0.5; b=0.5
> t <- function(samplesize=7,nsamples=1000){
          ts=c();
          for (i in 1:nsamples)
             ts[i]=mean(sort(runif(samplesize,a,b))[2:(samplesize-1)]);
          hist(ts,breaks=16,lwd=1.2,col="dark blue")
       }
> t()
```



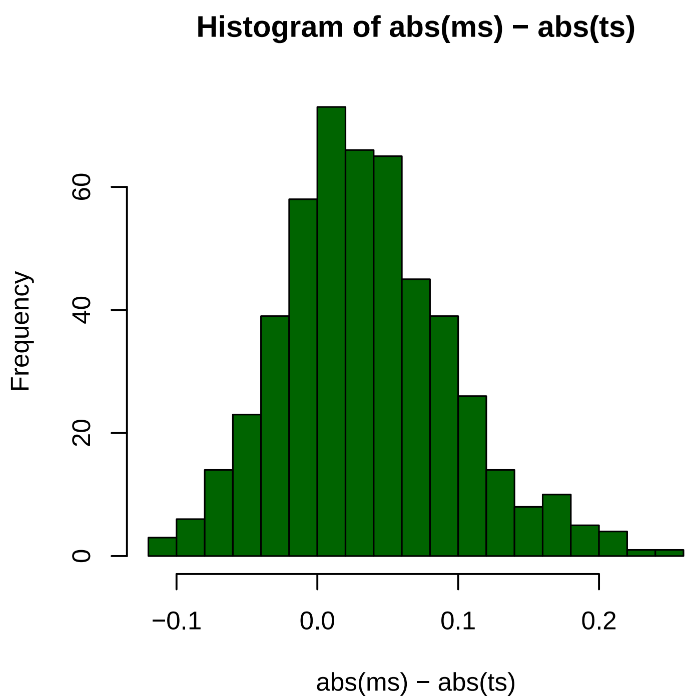
```
// two jury rules: m

> m <- function(samplesize=7,nsamples=1000){
          ms=c();
          for (i in 1:nsamples)
            ms[i]=median(runif(samplesize,a,b));
          hist(ms,breaks=16,lwd=1.2,col="dark red")
       }
> m()
```



```
// two jury rules: m vs. t

> mtplot <- function(samplesize=7,nsamples=500){
               ts=c(); ms=c();
               for (i in 1:nsamples) {
                  samp=runif(samplesize,a,b);
                  ts[i]=mean(sort(samp)[2:(samplesize-1)]);
                  ms[i]=median(samp);
               }
               tm=data.frame(ts,ms);
               plot(tm,col="purple",main="m vs. t");
               abline(0,1,col="springgreen")
            }
> mtplot()
```

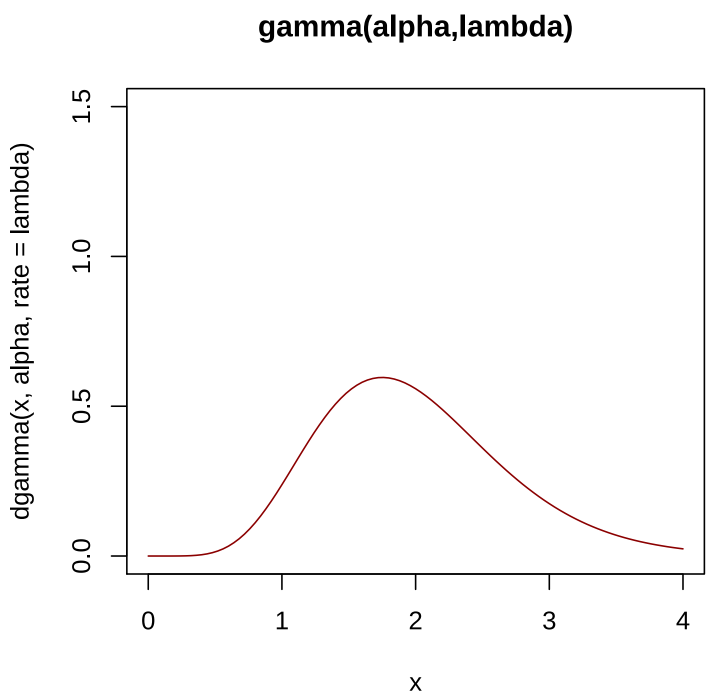


```
// two jury rules: abs(ms)-abs(ts)

> absplot <- function(samplesize=7,nsamples=500){
                ts=c(); ms=c();
                for (i in 1:nsamples) {
                   samp=runif(samplesize,a,b);
                   ts[i]=mean(sort(samp)[2:(samplesize-1)]);
                   ms[i]=median(samp);
                }
                hist(abs(ms)-abs(ts),
                breaks=16,lwd=1.2,col="dark green")
             }
> absplot()
```



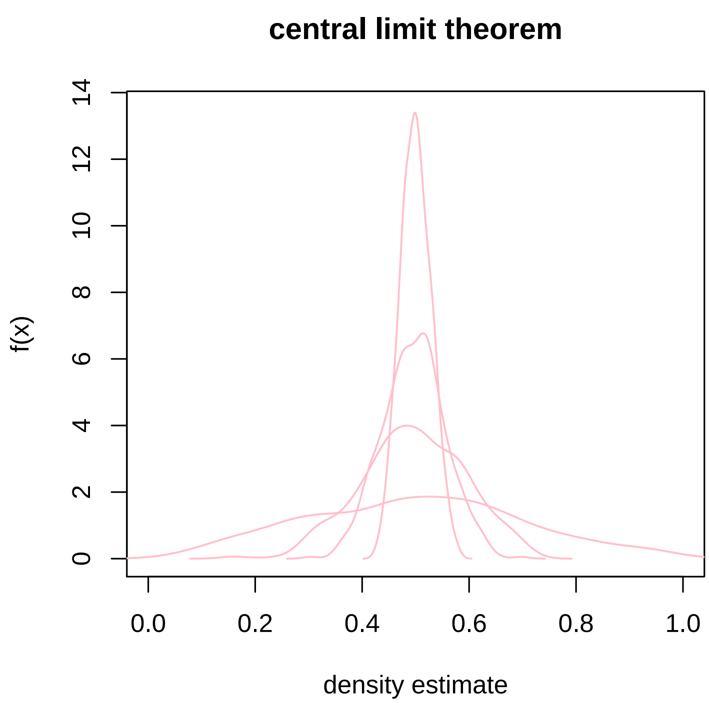
Chapter 13: The law of large numbers

```
// density function of a gamma distribution

// the gamma(alpha,lambda) distribution of Dekking, et al.
// is modeled by the gamma distribution of R with parameters
// shape = alpha, rate = lambda, scale = 1/lambda

> n=4; alpha=2*n; lambda=n
> curve(dgamma(x,alpha,lambda),
        0,4,ylim=c(0,1.5),
        col="dark red",
        main="gamma(alpha,lambda)")
```



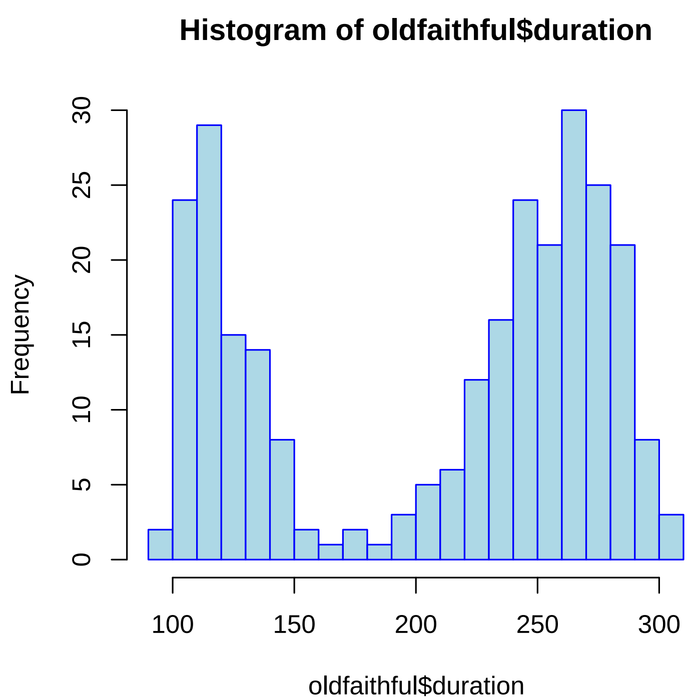
Chapter 14: The central limit theorem

```
// central limit theorem
// cf. Verzani, fig. 6.3, p168

> plot(0,0,type="n",xlim=c(0,1),ylim=c(0,13.5),
       xlab="density estimate",ylab="f(x)",
       main="central limit theorem")
> a=0; b=1
> f <- function(samplesize=100,nsamples=500){
          res=c();
          for (i in 1:nsamples)
             res[i]=mean(runif(samplesize,a,b));
          lines(density(res),lwd=1.2,col="pink")
       }
> lapply(c(2,10,25,100),f)
```

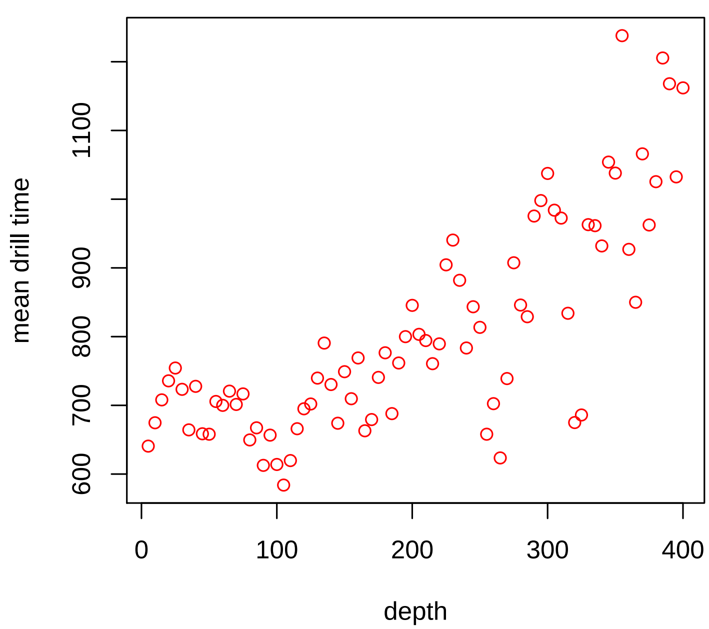


Chapter 15: Exploratory data analysis: graphical summaries

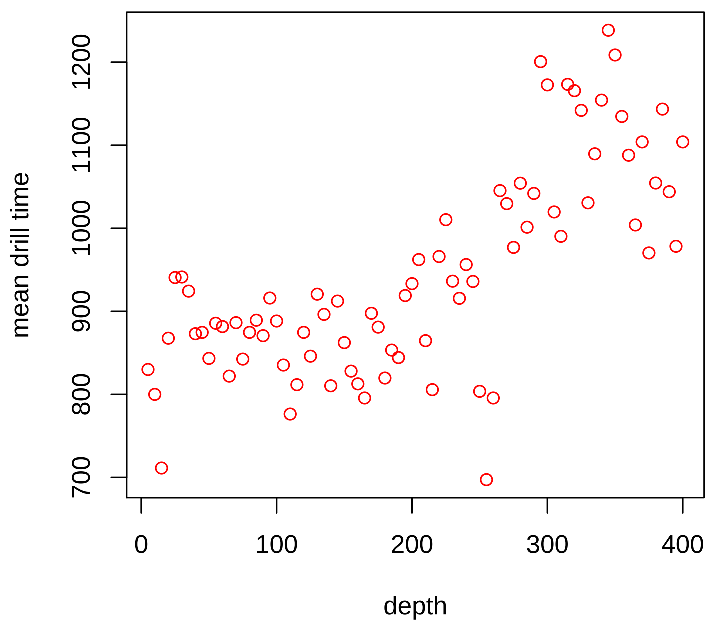
```
// Old Faithful histogram

> oldfaithful <- read.delim("oldfaithful.txt",
                             header=FALSE,
                             col.names=c("duration"))
> oldfaithful
    duration
1        216
2        108
3        200
4        137
5        272
   [etc]
> hist(oldfaithful$duration,
       breaks=20,
       col="lightblue",
       border="blue")
```



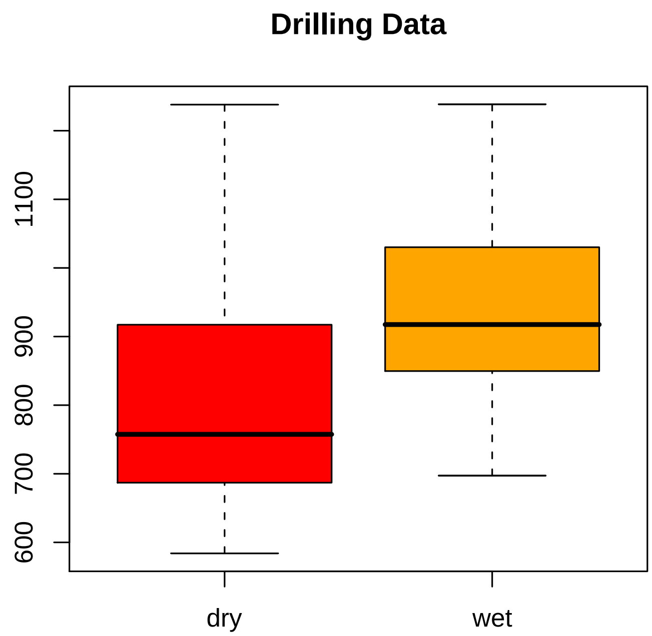
```
// scatterplot for dry drilling data

> drilling <- read.delim("drilling.txt",
                         header=FALSE,
                         col.names=c("depth","dry","wet"))
> drilling
   depth    dry    wet
1      5 640.67 830.00
2     10 674.67 800.00
3     15 708.00 711.33
4     20 735.67 867.67
5     25 754.33 940.67
 [etc]
> d=subset(drilling,
           select=c(depth,dry))
> plot(d,
       xlab="depth",
       ylab="mean drill time",
       col="red")
```



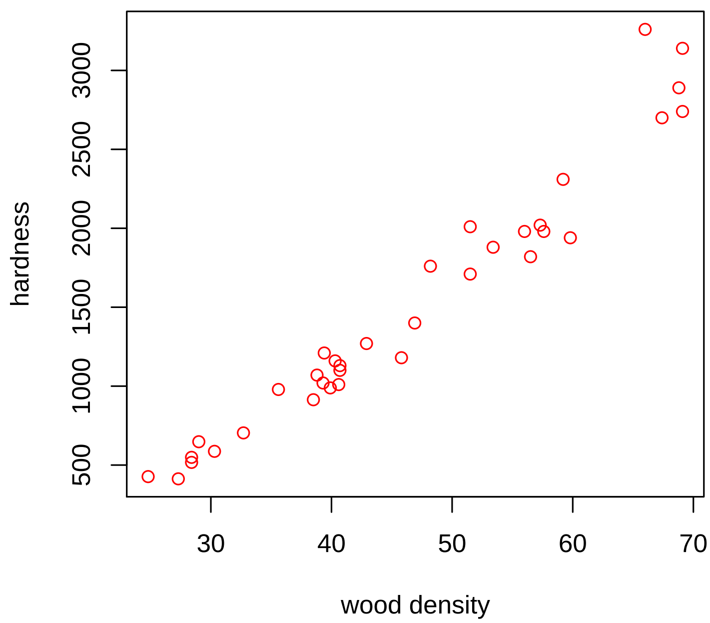
```
// scatterplot for wet drilling data

> w=subset(drilling,
           select=c(depth,wet))
> plot(w,
       xlab="depth",
       ylab="mean drill time",
       col="red")
```



```
// boxplots for drilling data

> boxplot(d$dry, w$wet,
          names=c("dry","wet"),
          col=c("red","orange"))
> title("Drilling Data")
```

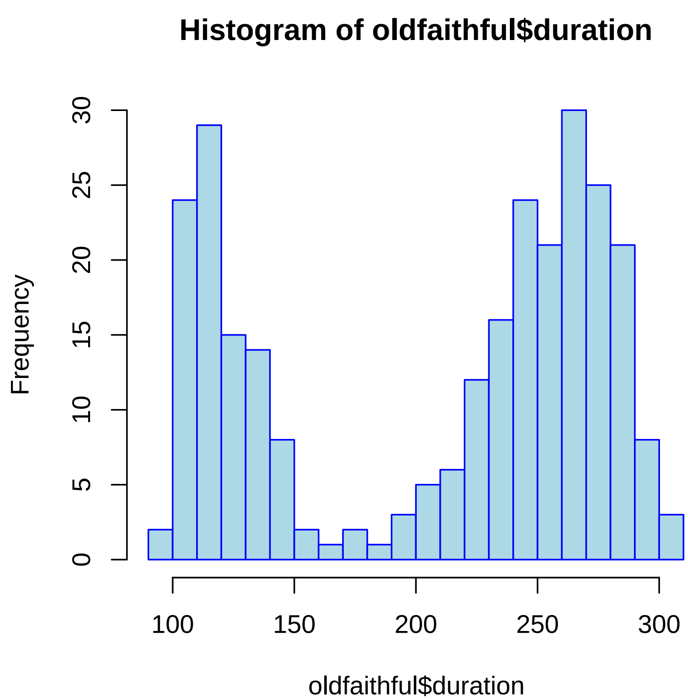


```
// scatterplot for jankahardness

> wood <- read.delim("jankahardness.txt",
                     header=FALSE,
                     col.names=c("density","hardness"))
> wood
  density hardness
1    24.7      484
2    24.8      427
3    27.3      413
4    28.4      517
5    28.4      549
   [etc]
> plot(wood,
      xlab="wood density",
      ylab="hardness",
      col="red")
```

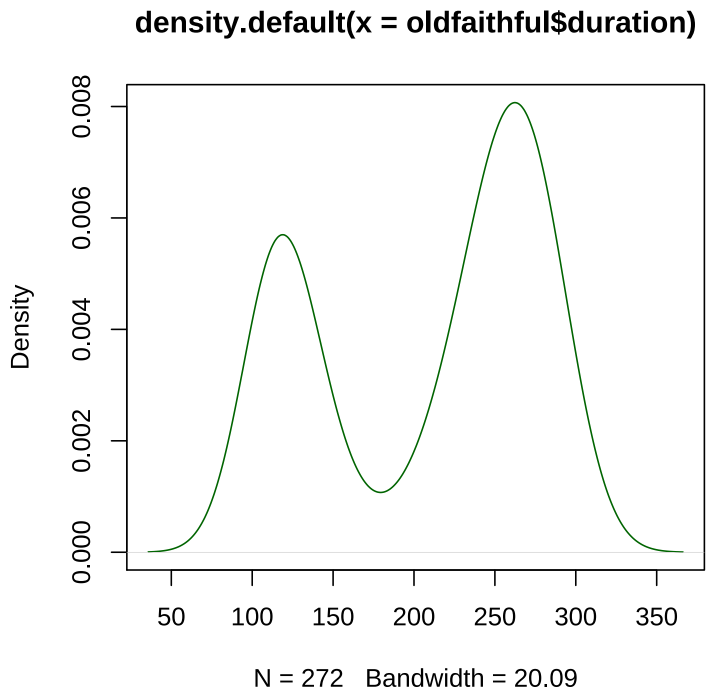


Chapter 17: Basic statistical models

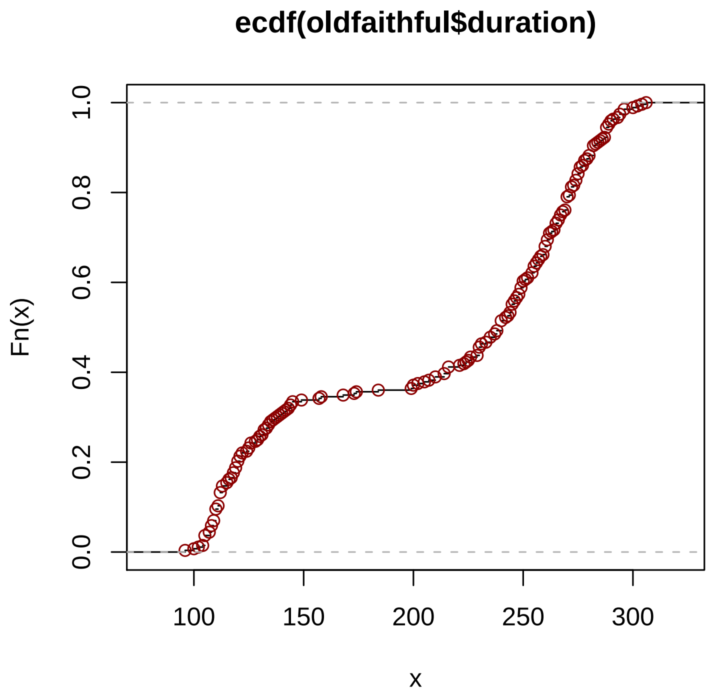
```
// Old Faithful histogram

> oldfaithful <- read.delim("oldfaithful.txt",
                            header=FALSE,
                            col.names=c("duration"))
> oldfaithful
    duration
1        216
2        108
3        200
4        137
5        272
   [etc]
> hist(oldfaithful$duration,
       breaks=20,
       col="lightblue",
       border="blue")
```



```
// Old Faithful kernel density plot

> plot(density(oldfaithful$duration),
       col="dark green")
```

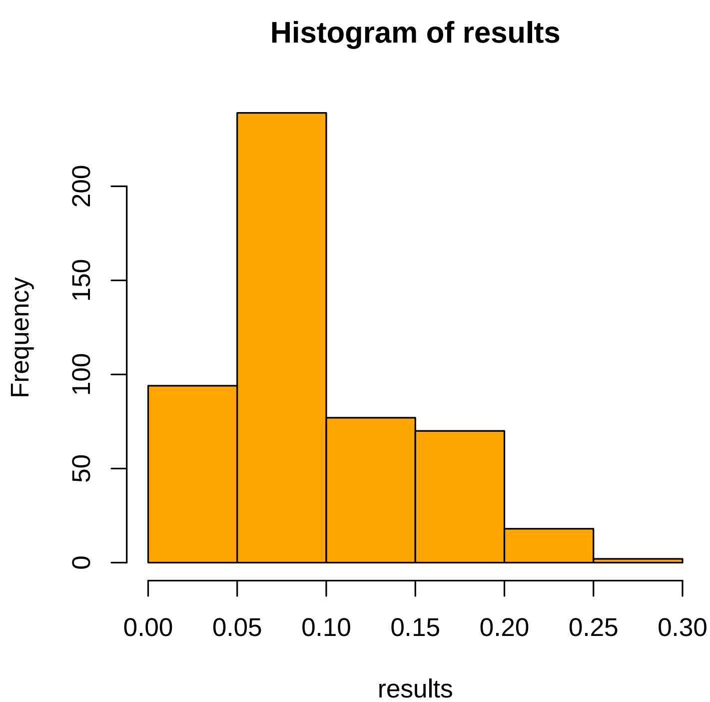


```
// Old Faithful empirical cumulative density plot

> plot(ecdf(oldfaithful$duration),
       col.points="dark red")
```

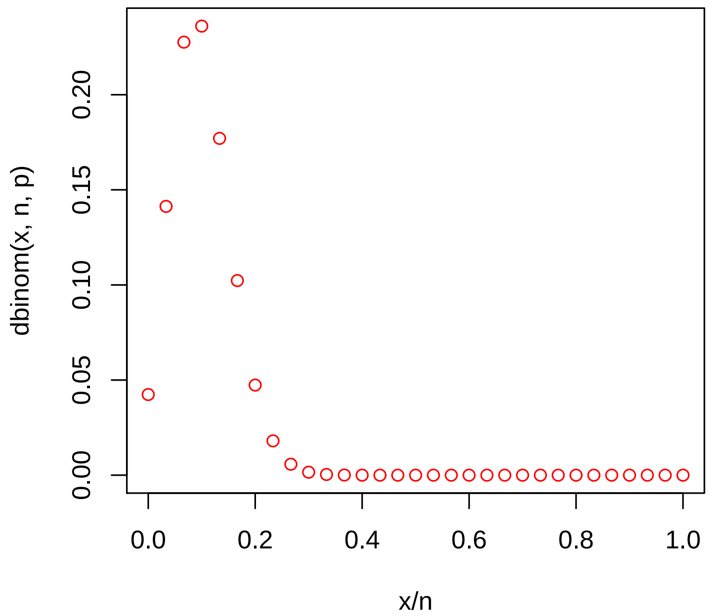


Chapter 19: Unbiased estimators

```
// sample distribution of an unbiased estimator

> mu=log(10); n=30; samp=rpois(n,mu); samp
 [1] 3 2 4 0 8 3 6 4 3 1 2 5 0 2 3 2 2 1 3 0 5 2 2 4 4 2 0 3 0 2

// determine the number and frequency of zeros in a sample

> nzeros <- function(samp){length(samp[samp==0])}
> fzeros <- function(samp){length(samp[samp==0])/length(samp)}

> samp=rpois(n,mu); samp
 [1] 2 2 5 1 3 3 4 6 3 2 4 5 6 1 3 4 0 1 3 4 5 2 5 1 2 4 2 0 1 3
> nzeros(samp)
[1] 2
> fzeros(samp)
[1] 0.06666667

// plot a frequency histogram for the estimator fzeros
// source sampling.r for Kaplan's program repeattrials

> results=repeattrials(stat=fzeros,
                       sf=function(){rpois(n,mu)},
                       nt=500)
> hist(results,
       breaks=9,
       col="orange")
```

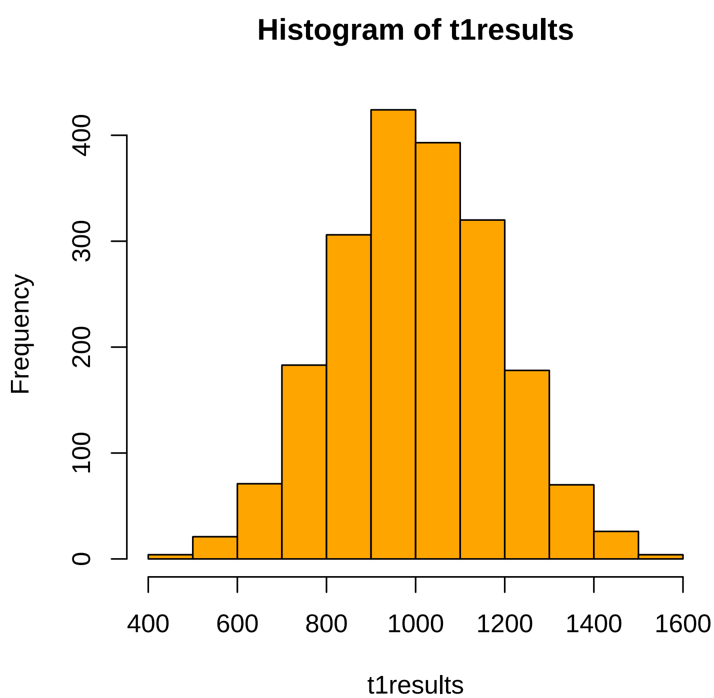


```
    // compare with a binomial distribution

> p=.1; n=30; x=0:n
> plot(x/n,dbinom(x,n,p),col="red")
```

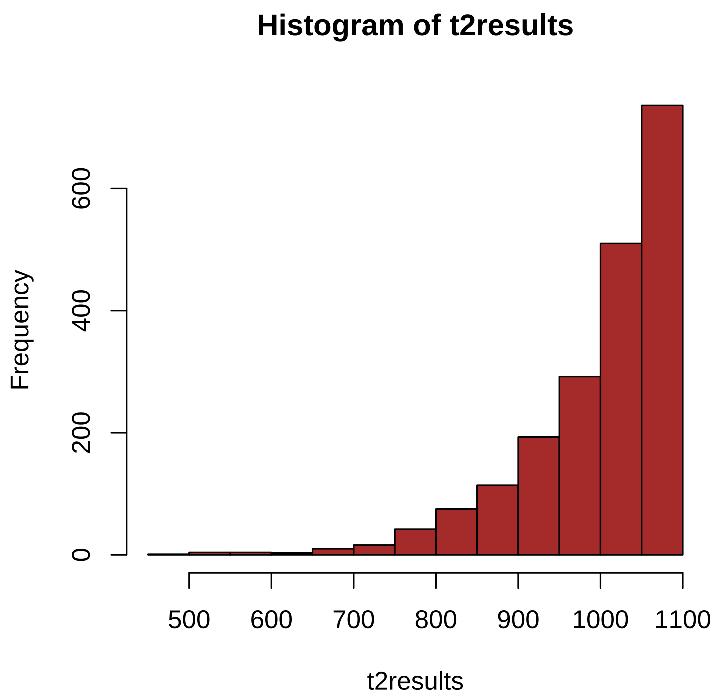


Chapter 20: Efficiency and mean-squared error

```
// sample distributions for two estimators

> N=1000; n=10
> samp=sample(1:N,n,replace=FALSE)
> samp
 [1] 105 648 524 461 700 356 501 829 780 347

// construct two unbiased estimators

> t1 <- function(samp){2*mean(samp)-1}
> t2 <- function(samp){((n+1)/n)*max(samp)-1}
> t1(samp)
[1] 1049.2
> t2(samp)
[1] 910.9

// plot frequency histograms for the estimators
// source sampling.r for Kaplan's program repeattrials

> t1results=repeattrials(stat=t1,
                         sf=function(){sample(1:N,n,replace=FALSE)},
                         nt=2000)
> hist(t1results,
       breaks=9,
       col="orange")
```

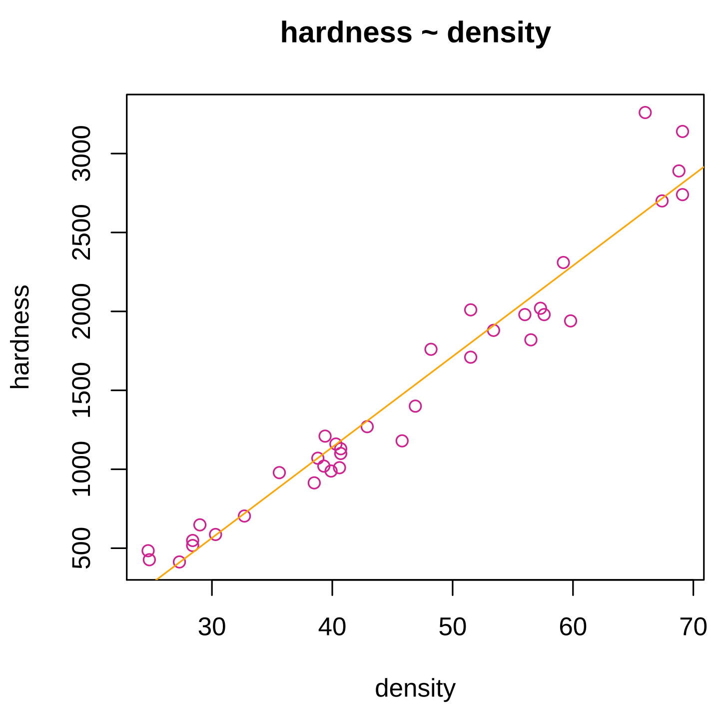


```
> t2results=repeattrials(stat=t2,
                         sf=function(){sample(1:N,n,replace=FALSE)},
                         nt=2000)
> hist(t2results,
       breaks=9,
       col="brown")
```

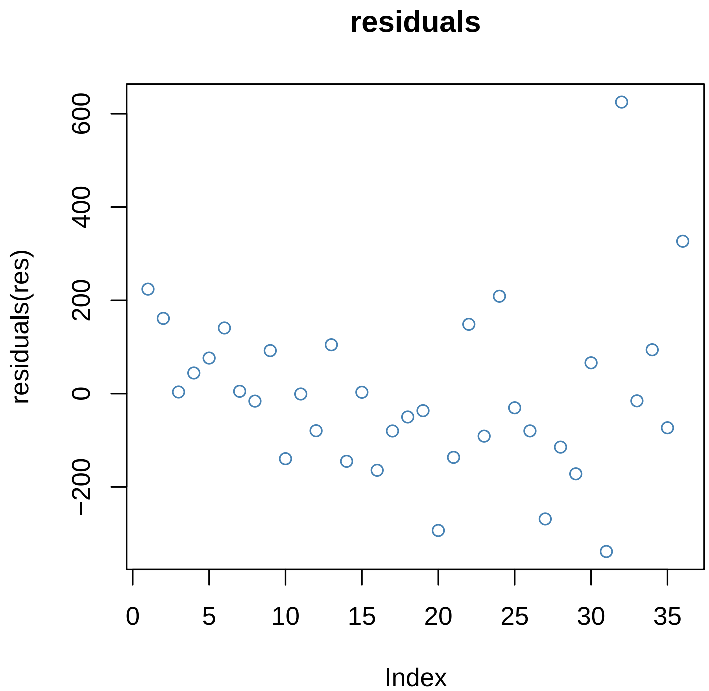


Chapter 22: The method of least squares

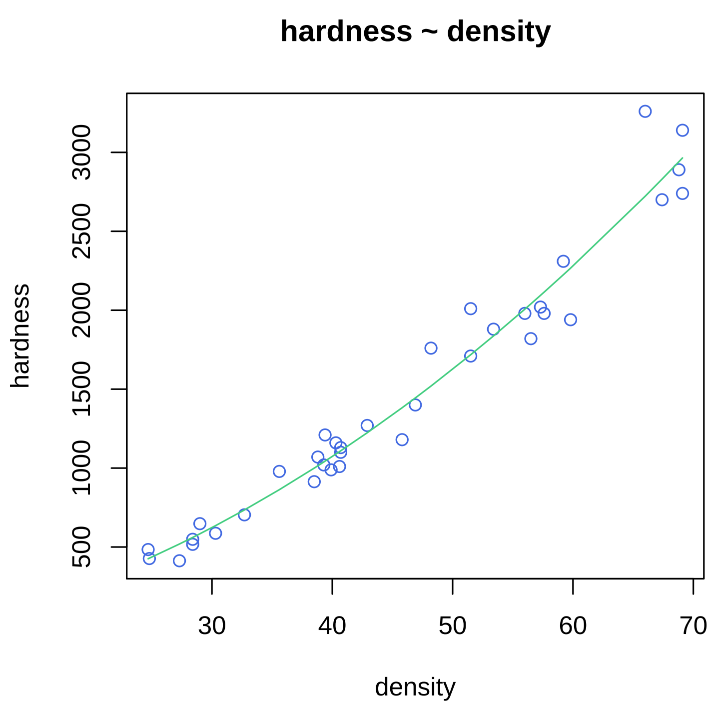
```
// regression line

> wood <- read.delim("jankahardness.txt",
                     header=FALSE,
                     col.names=c("density","hardness"))
> wood
   density hardness
1     24.7      484
2     24.8      427
3     27.3      413
   [etc]
> attach(wood)
> plot(hardness ~ density,
       main="hardness ~ density",
       col="violetred")
> res = lm(hardness ~ density)
> res

Call:
lm(formula = hardness ~ density)

Coefficients:
(Intercept)      density
   -1160.50        57.51

> abline(res,col="orange")
> detach(wood)
```



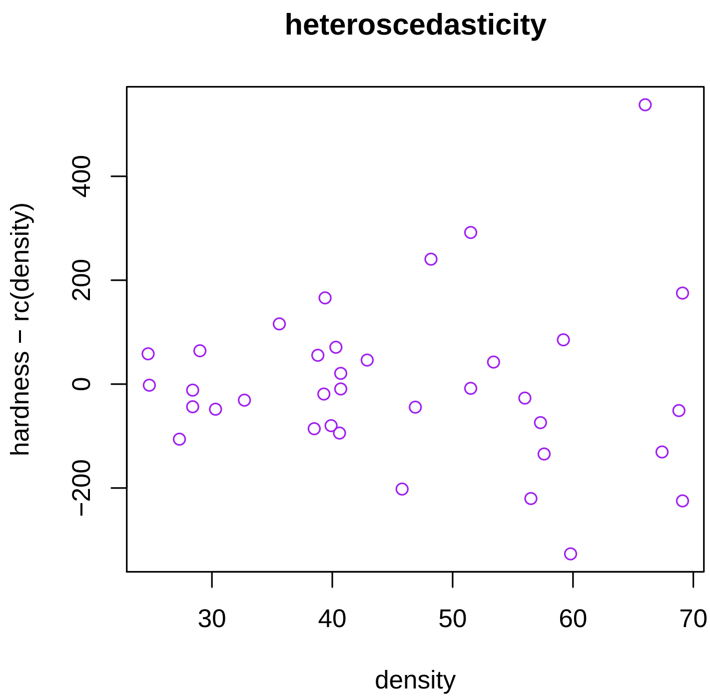
```
// residuals

> plot(residuals(res),
       main="residuals",
       col="steelblue")
```



```
// linear model with two explanatory variables

> densitysquared <- density^2
> lm(hardness ~ density + densitysquared)

Call:
lm(formula = hardness ~ density + densitysquared)

Coefficients:
   (Intercept)         density  densitysquared
     -118.0074          9.4340          0.5091

> rcurve = -118.0074 +
            9.4340*density +
            0.5091*density*density
> plot(hardness ~ density,
       main="hardness ~ density",
       col="royalblue")
> lines(density,rcurve,col="seagreen3")
```

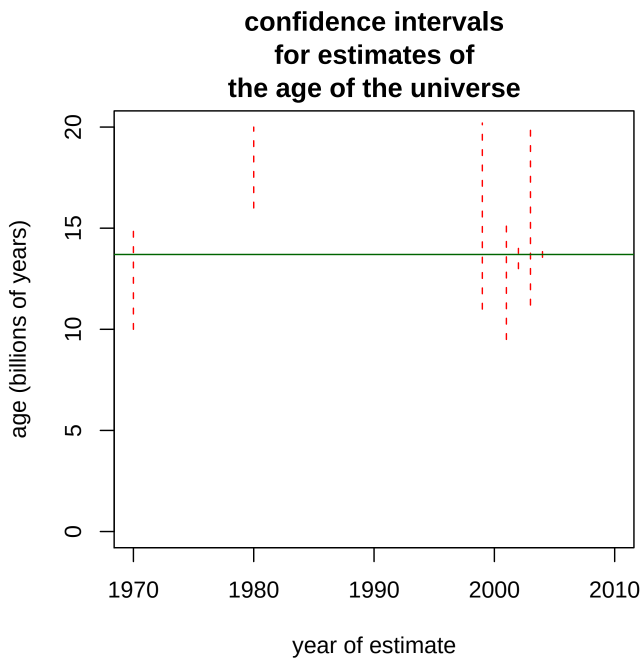


```
     // heteroscedasticity
     // what a word! say it quickly and sound confident

> rc <- function(x){-118.0074 + 9.4340*x + 0.5091*x*x}
> plot(density,
       hardness-rc(density),
       main="heteroscedasticity",
       col="purple")
> detach(wood)
```

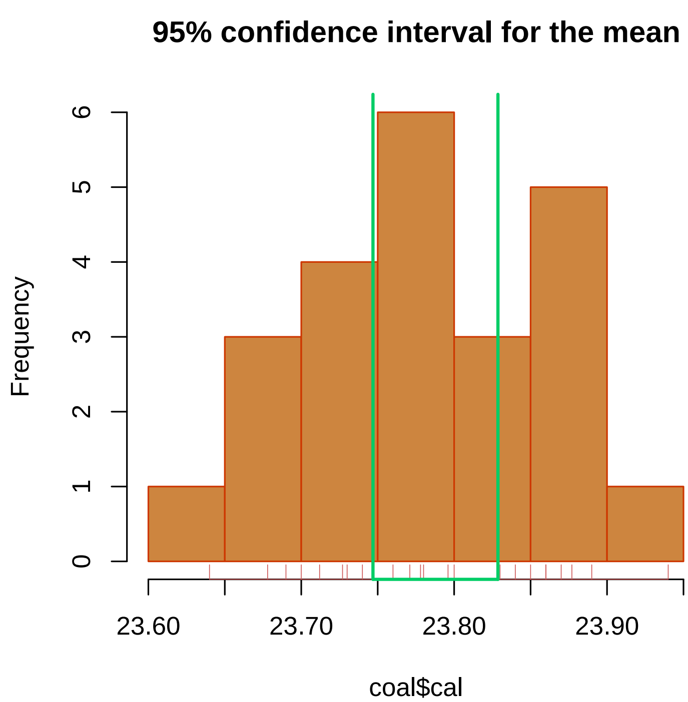


Chapter 23: Confidence intervals for the mean

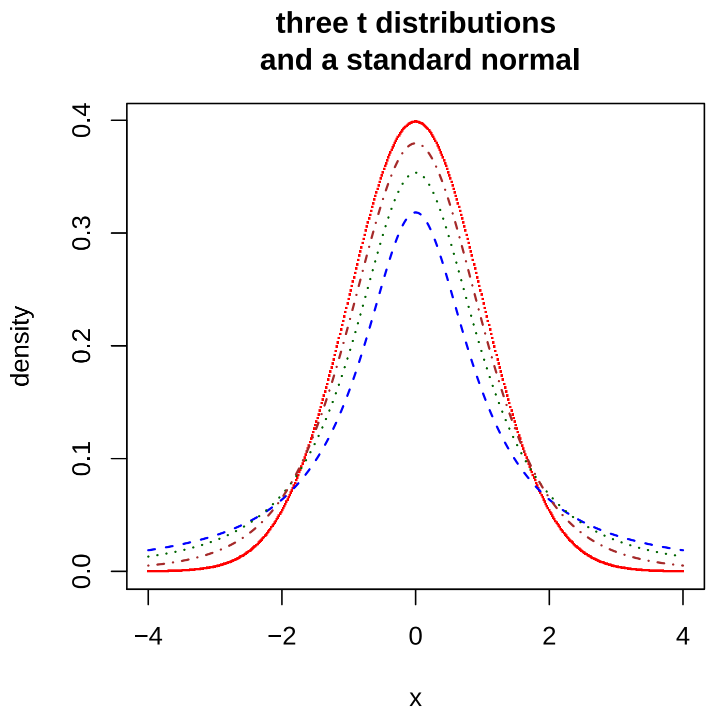
```
// age of the universe
// cf. Verzani, fig. 7.1, p182

> library(UsingR)
> age.universe
       lower upper year                                                source
7  13.000000 14.00 2002                                       Hubble telescope
3  11.200000 20.00 2003                           Krauss, Chaboyer, Science, 2003
12 13.560000 13.84 2003                         WMAP satellite. Microwave radiation
   [etc]
> estimates=list(
    c(2004,13.56,13.84),
    c(2003,11.20,20.00),
    c(2002,13.00,14.00),
    c(2001, 9.50,15.50),
    c(1999,11.00,20.20),
    c(1980,16.00,20.00),
    c(1970,10.00,15.00)
  )
> plot(0,0,type="n",
       xlim=c(1970,2010),ylim=c(0,20),
       main="confidence intervals\nfor estimates of\nthe age of the universe",
       xlab="year of estimate",ylab="age (billions of years)")
> f <- function(ls){
          x0=ls[1];y0=ls[2];x1=ls[1];y1=ls[3];
          segments(x0,y0,x1,y1,lty=2,col="red")
       }
> lapply(estimates,f)
> abline(h=13.7,col="dark green")
```



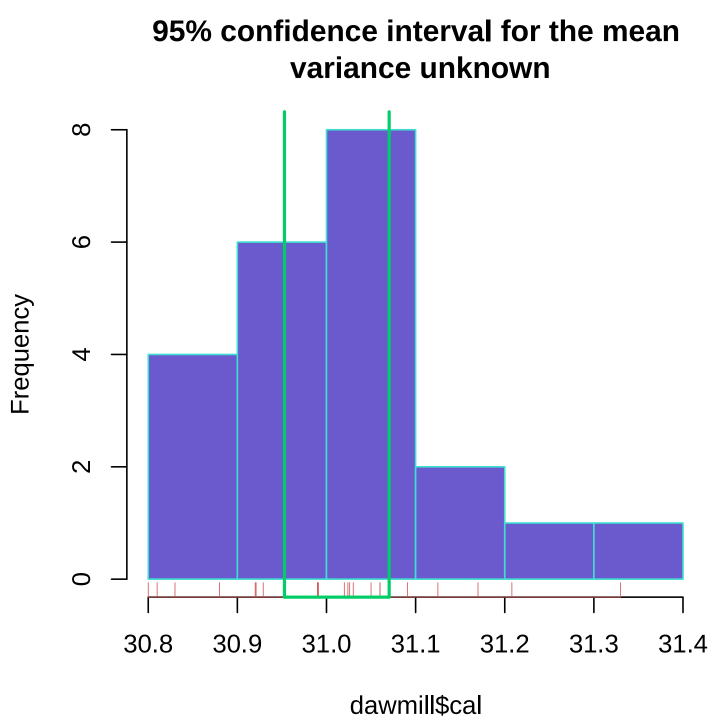
```
// confidence interval for the mean
// known variance

> coal <- read.delim("grosscalOsterfeld.txt",
                     header=FALSE,
                     col.names=c("cal"))
> coal
      cal
1  23.870
2  23.730
3  23.712
4  23.760
5  23.640
   [etc]
> m <- mean(coal$cal)
> sigma=.1; n=length(coal$cal)
> alpha=.05; zalpha=1.96
> ci=c(m-zalpha*sigma/sqrt(n),m+zalpha*sigma/sqrt(n))
> ci
[1] 23.74691 23.82865

> hist(coal$cal,
       col="peru",
       border="orangered3",
       main="95% confidence interval for the mean")
> rug(coal$cal,col="indianred3")
> rug(ci,ticksize=1,lwd=2,col="springgreen3")
```



```
// t distributions

> x <- seq(-4,4,length=1000)
> plot(x,dnorm(x,0,1),
       pch=".",ylab="density",
       main="three t distributions\n and a standard normal",
       col="red")
> lines(x,dt(x,1),lty=2,lwd=1.4,col="blue")
> lines(x,dt(x,2),lty=3,lwd=1.4,col="darkgreen")
> lines(x,dt(x,5),lty=4,lwd=1.4,col="brown")
```



```
// confidence interval for the mean
// unknown variance

> dawmill <- read.delim("grosscalDawMill.txt",
                        header=FALSE,
                        col.names=c("cal"))
> dawmill
      cal
1  30.990
2  31.030
3  31.060
4  30.921
5  30.920
   [etc]
> m=mean(dawmill$cal)
> sn=sd(dawmill$cal)
> n=length(dawmill$cal)
> alpha=.05; talpha=2.080
> ci=c(m-talpha*sn/sqrt(n-1),m+talpha*sn/sqrt(n-1))
> ci
[1] 30.95286 31.07032

> hist(dawmill$cal,
       col="slateblue",
       border="turquoise",
       main="95% confidence interval for the mean\n variance unknown")
> rug(dawmill$cal,col="indianred3")
> rug(ci,ticksize=1,lwd=2,col="springgreen3")
```



Chapter 24: More on confidence intervals

```
// confidence interval for a proportion

> a=1.0307; b=-1.2787; c=0.3894
> l=(-b-sqrt(b^2-4*a*c))/(2*a)
> u=(-b+sqrt(b^2-4*a*c))/(2*a)
> ci=c(l,u)
> ci
[1] 0.5367676 0.7038456

> x=seq(0.4,0.8,length=1000)
> plot(x,a*x^2+b*x+c,
       main="confidence interval for a proportion",
       col="green")
> lines(x,0*x,col="lightgreen")
> points(c(l,u),c(0,0),col="red")
> rug(ci,ticksize=1,lwd=2,col="lightblue")
```