```
#  draw the figures illustrating maximum likelihood

x<-c(1,3,4,6,8,9,12)
y<-c(5,8,6,10,9,13,12)
windows(14,6)
par(mfrow=c(1,3))
plot(x,y,pch=21,bg="blue",ylim=c(0,15))
abline(0,0.6793,col="red")
plot(x,y,pch=21,bg="blue",ylim=c(0,15))
abline(8,0.6793,col="red")
plot(x,y,pch=21,bg="blue",ylim=c(0,15))
abline(lm(y~x))
abline(lm(y~x),col="blue")



plot(x,y,pch=21,bg="blue",ylim=c(0,15))
abline(4.8273,1.5,col="red")
plot(x,y,pch=21,bg="blue",ylim=c(0,15))
abline(4.8273,0.2,col="red")
plot(x,y,pch=21,bg="blue",ylim=c(0,15))
abline(lm(y~x))
abline(lm(y~x),col="blue")


# randomizing treatments for experimental design

treatments <- c("aloprin","vitex","formixin","panto","allclear")

#   use sample to shuffle them for the active insects in dishes 1 to 5

sample(treatments)


# this produces a warning message because the same variable name
# appears in two attached dataframes

first.frame <- read.csv("c:\\temp\\test.pollute.csv")
second.frame <- read.csv("c:\\temp\\ozone.data.csv")
attach(first.frame)
attach(second.frame)

# this is how you should avoid this kind of problem

first.frame <- read.csv("c:\\temp\\test.pollute.csv")
second.frame <- read.csv("c:\\temp\\ozone.data.csv")
attach(first.frame)

# ........
#  this is where you work on the information from first.frame.
#  Then when you are finished
# ........

detach(first.frame)
attach(second.frame)
```

```
# read data from a file called worms.csv to create a
# dataframe called worms

worms <- read.csv("c:\\temp\\worms.csv")
names(worms)

attach(worms)
worms

# all rows, just columns 1 to 3

worms[,1:3]

#  all columns just rows 5 to 15

worms[5:15,]

# all columns but only selected rows (area > 3 and slope < 3)

worms[Area>3 & Slope <3,]


# sort the rows by increasing area


worms[order(Area),]


# only the columns with numeric data


worms[order(Area),c(2,3,5,7)]


# sort into decending order with just two columns


worms[rev(order(worms[,5])),c(5,7)]

# using tapply with a specified dataframe

with(worms,tapply(Worm.density,Vegetation,mean))

# using aggregate to summarise multiple variables by facor levels

aggregate(worms[,c(2,3,5,7)],list(Vegetation),mean)

aggregate(worms[,c(2,3,5,7)],list(Community=Vegetation),mean)

# multiple explanatory variables

aggregate(worms[,c(2,3,5,7)],
      list(Moisture=Damp,Community=Vegetation),mean)

# aggregate and tapply compared

with(worms,tapply(Slope,list(Damp,Vegetation),mean))

# plotting your data

data <- read.csv("c:\\temp\\das.csv")
attach(data)
```

```
head(data)

# finding the identity of the outlier

which(y > 10)
y[50]

# plots with categorical explanatory variables

yields <- read.csv("c:\\temp\\fertyield.csv")
attach(yields)
head(yields)
table(treatment)
which(treatment == "nitogen")

# scatterplots

data <- read.csv("c:\\temp\\scatter.csv")
attach(data)
head(data)

plot(x,y,pch=21,bg="red")


# box and whisker plots

data <- read.csv("c:\\temp\\weather.data.csv")
attach(data)
head(data)

plot(factor(month),upper)

# data for coplot

data <- read.csv("c:\\temp\\coplot.csv")
attach(data)
head(data)

# scale the plotting area to accommodate two plots side by side

windows(7,4)
par(mfrow=c(1,2))
plot(x,y)
plot(z,y)

# using coplot

windows(7,7)
coplot(y~x|z,pch=16,panel=panel.smooth)

# factorial data

data <- read.csv("c:\\temp\\np.csv")
attach(data)
head(data)

windows(7,4)
par(mfrow=c(1,2))
plot(nitrogen,yield,main="N")
plot(phosphorus,yield,main="P")
```

```
tapply(yield,list(nitrogen,phosphorus),mean)

barplot(tapply(yield,list(nitrogen,phosphorus),mean),
                                        beside=TRUE,xlab="phosphorus")
legend(locator(1),legend=c("no","yes"),title="nitrogen",
                                         fill=c("black","lightgrey"))
```

```
#  Chapter 3

yvals <- read.csv("c:\\temp\\yvalues.csv")
attach(yvals)
hist(y)


# arithmetic mean

total <- sum(y)
n <- length(y)

( ybar <- total/n )

arithmetic.mean <- function(x) sum(x)/length(x)

data <- c(3,4,6,7)
arithmetic.mean(data)

arithmetic.mean(y)

mean(y)


# median

sorted <- sort(y)
length(y)/2
ceiling(length(y)/2)
sorted[20]
sorted[ceiling(length(y)/2)]


sort(y)[ceiling(length(y)/2)]

y.even <- y[-1]
length(y.even)

sort(y.even)[19]
sort(y.even)[20]


(sort(y.even)[19]+sort(y.even)[20])/2

38%%2

39%%2


med <- function(x) {
modulo <- length(x)%%2
if (modulo == 0)
(sort(x)[ceiling(length(x)/2)]+sort(x)[ceiling(1+length(x)/2)])/2
else  sort(x)[ceiling(length(x)/2)]
}

med(y)
med(y.even)

median(y)
median(y.even)
```

```
#   geometric mean

100000^0.2
insects <- c(1,10,1000,10,1)
mean(insects)

exp(mean(log(insects)))


# harmonic mean

v <- c(1,2,4,1)
length(v)/sum(1/v)

1/mean(1/v)
```

```
#   Variance

y <- c(13,7,5,12,9,15,6,11,9,7,12)
plot(y,ylim=c(0,20))

# range

range(y)

plot(1:11,y,ylim=c(0,20),pch=16,col="blue")
lines(c(4.5,4.5),c(5,15),col="brown")
lines(c(4.5,3.5),c(5,5),col="brown",lty=2)
lines(c(4.5,5.5),c(15,15),col="brown",lty=2)

# residuals

plot(1:11,y,ylim=c(0,20),pch=16,col="blue")
abline(h=mean(y),col="green")
for (i in 1:11) lines(c(i,i),c(mean(y),y[i]),col="red")


# sum of squares

y - mean(y)
y - mean(y))^2
sum((y - mean(y))^2)

# variance

variance <- function (x)   sum((x-mean(x))^2)/(length(x)-1)

variance(y)
var(y)

# ozone example

ozone <- read.csv("c:\\temp\\gardens.csv")
attach(ozone)
ozone

mean(gardenA)

gardenA - mean(gardenA)

(gardenA - mean(gardenA))^2

sum((gardenA - mean(gardenA))^2)

sum((gardenA - mean(gardenA))^2)/9

mean(gardenB)

gardenB - mean(gardenB)

(gardenB - mean(gardenB))^2

sum((gardenB - mean(gardenB))^2)

sum((gardenB - mean(gardenB))^2)/9

mean(gardenC)
```

```
gardenC - mean(gardenC)

(gardenC - mean(gardenC))^2

sum((gardenC - mean(gardenC))^2)

sum((gardenC - mean(gardenC))^2)/9

var(gardenC)/var(gardenB)


# critical value of Fisher's F

2*(1 - pf(10.667,9,9))

var.test(gardenB,gardenC)

# sample size

plot(c(0,32),c(0,15),type="n",xlab="Sample size",ylab="Variance")

for (df in seq(3,31,2)) {
for( i in 1:30){
x <- rnorm(df,mean=10,sd=2)
points(df,var(x)) }}

# standard error of a mean

sqrt(var(gardenA)/10)

sqrt(var(gardenB)/10)

sqrt(var(gardenC)/10)

# quantiles of the t distribution

qt(.025,9)

qt(.975,9)

qt(.995,9)

qt(.9975,9)

qt(.975,9)*sqrt(1.33333/10)

# bootstrap intervals


data <- read.csv("c:\\temp\\skewdata.csv")
attach(data)
names(data)

plot(c(0,30),c(0,60),type="n",xlab="Sample size",
ylab="Confidence interval")
for (k in seq(5,30,3)){
a <- numeric(10000)
for (i in 1:10000){
a[i] <- mean(sample(values,k,replace=T))
}
```

```
points(c(k,k),quantile(a,c(.025,.975)),type="b",pch=21,bg="red")
}

quantile(a,c(.025,.975))

xv <- seq(5,30,0.1)
yv <- mean(values)+1.96*sqrt(var(values)/xv)
lines(xv,yv,col="blue")
yv <- mean(values)-1.96*sqrt(var(values)/xv)
lines(xv,yv,col="blue")

yv <- mean(values)-qt(.975,xv)*sqrt(var(values)/xv)
lines(xv,yv,lty=2,col="green")
yv <- mean(values)+qt(.975,xv)*sqrt(var(values)/xv)
lines(xv,yv,lty=2,col="green")
```

```
# single samples

data <- read.csv("c:\\temp\\example.csv")
attach(data)
names(data)

summary(y)

boxplot(y)

hist(y)

# rug plot

length(table(y))
plot(range(y),c(0,10),type="n",xlab="y values",ylab="")
for (i in 1:100) lines(c(y[i],y[i]),c(0,1),col="blue")

# designing a histogram

(max(y)-min(y))/10

diff(range(y))/11

# the game of craps

score <- 2:12

ways <- c(1,2,3,4,5,6,5,4,3,2,1)

( game <- rep(score,ways) )

sample(game,1)

outcome <- numeric(10000)
for (i in 1:10000) outcome[i] <- sample(game,1)
hist(outcome,breaks=(1.5:12.5))

mean.score <- numeric(10000)
for (i in 1:10000) mean.score[i] <- mean(sample(game,3))
hist(mean.score,breaks=(1.5:12.5))

mean(mean.score)
sd(mean.score)

xv <- seq(2,12,0.1)
yv <- 10000*dnorm(xv,mean(mean.score),sd(mean.score))
hist(mean.score,breaks=(1.5:12.5),ylim=c(0,3000),col="yellow", main="")
lines(xv,yv,col="red")

# standard normal disribution

standard.deviations <- seq(-3,3,0.01)
pd <- dnorm(standard.deviations)
plot(standard.deviations,pd,type="l",col="blue")

pnorm(-2)
pnorm(-1)
1-pnorm(3)

qnorm(c(0.025,0.975))
```

```
# shading the tails of the standard normal distribution

xv<-seq(-3,3,0.01)
yv<-dnorm(xv)
plot(c(-3,3),c(0,0.3),xlim=c(-
3,3),ylim=c(0,0.4),type="n",ylab="pd",xlab="standard deviations")
polygon(c(1.96,1.96,-1.96,-
1.96,xv[105:496]),c(yv[496],0,0,yv[105],yv[105:496]),col="green")
polygon(c(-1.96,-1.96,xv[1],xv[1:104]),c(yv[104],0,0,yv[1:104]),col="red")
polygon(c(xv[601],xv[601],1.96,1.96,xv[497:601]),c(yv[601],0,0,yv[496:601])
,col="red")
text(0,0.2,"95%",cex=2)
lines(xv,yv,col="blue")


# calculations with the sandard normal distribution


ht <- seq(150,190,0.01)
plot(ht,dnorm(ht,170,8),type="l",col="brown",
ylab="Probability density",xlab="Height")

pnorm(-1.25)

pnorm(1.875)
1 - pnorm(1.875)

pnorm(1.25) - pnorm(-0.625)

# drawing a panel of four normal distributions

par(mfrow=c(2,2))

ht <- seq(150,190,0.01)
pd <- dnorm(ht,170,8)

plot(ht,dnorm(ht,170,8),type="l",col="brown",
ylab="Probability density",xlab="Height")

plot(ht,dnorm(ht,170,8),type="l",col="brown",
ylab="Probability density",xlab="Height")
yv <- pd[ht<=160]
xv <- ht[ht<=160]
xv <- c(xv,160,150)
yv <- c(yv,yv[1],yv[1])
polygon(xv,yv,col="orange")


plot(ht,dnorm(ht,170,8),type="l",col="brown",
ylab="Probability density",xlab="Height")
xv <- ht[ht>=185]
yv <- pd[ht>=185]
xv <- c(xv,190,185)
yv <- c(yv,yv[501],yv[501])
polygon(xv,yv,col="blue")

plot(ht,dnorm(ht,170,8),type="l",col="brown",
ylab="Probability density",xlab="Height")
xv <- ht[ht>=160 & ht <= 180]
yv <- pd[ht>=160 & ht <= 180]
```

```
xv <- c(xv,180,160)
yv <- c(yv,pd[1],pd[1])
polygon(xv,yv,col="green")

# plots for skewness

data <- read.csv("c:\\temp\\skewdata.csv")
attach(data)
qqnorm(values)
qqline(values,lty=2)

# speed of light data

light <- read.csv("c:\\temp\\light.csv")
attach(light)
names(light)
hist(speed)
summary(speed)

wilcox.test(speed,mu=990)

a <- numeric(10000)
for(i in 1:10000)  a[i] <- mean(sample(speed,replace=T))
hist(a)


# student's t distribution

plot(c(0,30),c(0,10),type="n",
xlab="Degrees of freedom",ylab="Students t value")
lines(1:30,qt(0.975,df=1:30),col="red")
abline(h=1.96,lty=2,col="green")

xvs <- seq(-4,4,0.01)
plot(xvs,dnorm(xvs),type="l",
ylab="Probability density",xlab="Deviates")
lines(xvs,dt(xvs,df=5),col="red")

qt(0.975,5)


# skewness

skew <- function(x){
m3 <- sum((x-mean(x))^3)/length(x)
s3 <- sqrt(var(x))^3
m3/s3  }

hist(values,main="",col="green")

skew(values)
skew(values)/sqrt(6/length(values))
1 - pt(2.949,28)

skew(sqrt(values))/sqrt(6/length(values))
skew(log(values))/sqrt(6/length(values))


# kurtosis

kurtosis <- function(x) {
```

```
m4 <- sum((x-mean(x))^4)/length(x)
s4 <- var(x)^2
m4/s4 - 3  }

kurtosis(values)
kurtosis(values)/sqrt(24/length(values))
```

```
# code 7   Regression


# text figure

plot(c(0,10),c(0,100),xlab="",ylab="",type="n")
lines(c(0,10),c(80,10),lwd=2)

# intercept = 80

lines(c(0,0),c(0,80),col="green")
lines(c(0,-10),c(80,80),col="red")

# slope = -7

lines(c(2,8),c(24,24),col="brown")
lines(c(2,2),c(66,24),col="blue")

# tannin example

reg.data <- read.csv("c:\\temp\\tannin.csv")
attach(reg.data)
names(reg.data)

plot(tannin,growth,pch=21,bg="blue")

lm(growth~tannin)
abline(lm(growth~tannin),col="green")

fitted <- predict(lm(growth~tannin))
fitted
lines(c(0,0),c(12,11.7555555))

# residuals

for (i in 1:9)
lines (c(tannin[i],tannin[i]),c(growth[i],fitted[i]),col="red")

# estimating the maximum likelihood slope

b <-  seq(-1.43,-1,0.002)
sse <- numeric(length(b))
for (i in 1:length(b)) {
a <- mean(growth)-b[i]*mean(tannin)
residual <- growth - a - b[i]*tannin
sse[i] <- sum(residual^2)
}

plot(b,sse,type="l",ylim=c(19,24))
arrows(-1.216,20.07225,-1.216,19,col="red")
abline(h=20.07225,col="green",lty=2)
lines(b,sse)

b[which(sse==min(sse))]


# corrected sums of squares

SSX <- sum(tannin^2)-sum(tannin)^2/length(tannin)
SSY <- sum(growth^2)-sum(growth)^2/length(growth)
SSXY <- sum(tannin*growth)-sum(tannin)*sum(growth)/length(tannin)
```

```
# box 7.5 figure

plot(c(0,10),c(0,10),xlab="",ylab="",type="n")
abline(h=5,lty=2)
lines(c(0,10),c(8,2))
text(2,6.2,expression(hat(y) - bar(y)))
text(2,8.45,expression(y - hat(y)))
arrows(7,5,7,9.5,code=3,length=0.1)
arrows(1,5,1,7.4,code=3,length=0.1)
arrows(1,9.5,1,7.4,code=3,length=0.1)
points(1,9.5,pch=16)
text(8,7.4,expression(y - bar(y)))
text(0.2,5,expression(bar(y)))
text(.2,7.4,expression(hat(y)))
text(.2,9.5,"y")


# regreesion model in R


model <- lm(growth~tannin)
summary(model)
summary.aov(model)

par(mfrow=c(2,2))
plot(model)


# a non-linear relationship

par(mfrow=c(1,1))
data <- read.csv("c:\\temp\\decay.csv")
attach(data)
names(data)

plot(time,amount,pch=21,col="blue",bg="green")

abline(lm(amount~time),col="red")
summary(lm(amount~time))

plot(time,log(amount),pch=21,col="blue",bg="red")
abline(lm(log(amount)~time),col="blue")

model <- lm(log(amount)~time)
summary(model)

par(mfrow=c(1,1))
plot(time,amount,pch=21,col="blue",bg="green")
xv <- seq(0,30,0.25)
yv <- 94.38536 * exp(-0.068528 * xv)
lines(xv,yv,col="red")

# shapes of quadratic relationships

par(mfrow=c(2,2))
curve(4+2*x-0.1*x^2,0,10,col="red",ylab="y")
curve(4+2*x-0.2*x^2,0,10,col="red",ylab="y")
curve(12-4*x+0.3*x^2,0,10,col="red",ylab="y")
curve(4+0.5*x+0.1*x^2,0,10,col="red",ylab="y")
```

```
model2 <- lm(amount~time)
model3 <- lm(amount~time+I(time^2))
summary(model3)
AIC(model2,model3)
anova(model2,model3)


# non-linear regression using nls

deer <- read.csv("c:\\temp\\jaws.csv")
attach(deer)
names(deer)
par(mfrow=c(1,1))
plot(age,bone,pch=21,bg="lightgrey")

model <- nls(bone~a-b*exp(-c*age),start=list(a=120,b=110,c=0.064))
summary(model)

model2 <- nls(bone~a*(1-exp(-c*age)),start=list(a=120,c=0.064))
anova(model,model2)

av <- seq(0,50,0.1)
bv <- predict(model2,list(age=av))
lines(av,bv,col="blue")
summary(model2)

null.model <- lm(bone ~ 1)
summary.aov(null.model)


#  generalized additive models GAM

library(mgcv)
hump <- read.csv("c:\\temp\\hump.csv")
attach(hump)
names(hump)

model <- gam(y~s(x))

plot(model,col="blue")
points(x,y-mean(y),pch=21,bg="red")

summary(model)
```

```
# one-way anova

oneway <- read.csv("c:\\temp\\oneway.csv")
attach(oneway)
names(oneway)

plot(1:20,ozone,ylim=c(0,8),ylab="y",xlab="order",pch=21,bg="red")

abline(h=mean(ozone),col="blue")
for(i in 1:20) lines(c(i,i),c(mean(ozone),ozone[i]),col="green")

plot(1:20,ozone,ylim=c(0,8),ylab="y",xlab="order",
pch=21,bg=as.numeric(garden))
abline(h=mean(ozone[garden=="A"]))
abline(h=mean(ozone[garden=="B"]),col="red")

index <- 1:length(ozone)
for (i in 1:length(index)){
if (garden[i] == "A" )
     lines(c(index[i],index[i]),c(mean(ozone[garden=="A"]),ozone[i]))
else
     lines(c(index[i],index[i]),c(mean(ozone[garden=="B"]),ozone[i]),
col="red")
}

SSY <- sum((ozone-mean(ozone))^2)
SSY

sum((ozone[garden=="A"]-mean(ozone[garden=="A"]))^2)
sum((ozone[garden=="B"]-mean(ozone[garden=="B"]))^2)

qf(0.95,1,18)
1-pf(15.0,1,18)

summary(aov(ozone~garden))

plot(aov(ozone~garden))

cbind(ozone[garden=="A"],ozone[garden=="B"])
tapply(ozone,garden,sum)

mean(ozone[garden=="A"])-mean(ozone)
mean(ozone[garden=="B"])-mean(ozone)
mean(ozone[garden=="A"])
mean(ozone[garden=="B"])-mean(ozone[garden=="A"])


# plots for anova

comp <- read.csv("c:\\temp\\competition.csv")
attach(comp)
names(comp)

plot(clipping,biomass,xlab="Competition treatment",
ylab="Biomass",col="lightgrey")

heights <- tapply(biomass,clipping,mean)
barplot(heights,col="green",ylim=c(0,700),
ylab="mean biomass",xlab="competition treatment")
```

```
# error bars

error.bars <- function(y,z) {
x <- barplot(y,plot=F)
n <- length(y)
for (i in 1:n)
arrows(x[i],y[i]-z,x[i],y[i]+z,code=3,angle=90,length=0.15)
}

model <- aov(biomass~clipping)
summary(model)
table(clipping)

se <- rep(28.75,5)

error.bars(heights,se)

ci <- se*qt(.975,5)
barplot(heights,col="green",ylim=c(0,700),
ylab="mean biomass",xlab="competition treatment")
error.bars(heights,ci)

lsd <- qt(0.975,10)*sqrt(2*4961/6)
lsdbars <- rep(lsd,5)/2

barplot(heights,col="green",ylim=c(0,700),
ylab="mean biomass",xlab="competition treatment")
error.bars(heights,lsdbars)


# fatorial experiments

weights <- read.csv("c:\\temp\\growth.csv")
attach(weights)

barplot(tapply(gain,list(diet,supplement),mean),beside=T)

labels <- levels(diet)
shade <- c(0.2,0.6,0.9)

barplot(tapply(gain,list(diet,supplement),mean),beside=T,
ylab="weight gain",xlab="supplement",ylim=c(0,30))

legend(locator(1),labels,gray(shade))

tapply(gain,list(diet,supplement),mean)
model <- aov(gain~diet*supplement)
summary(model)
tapply(gain,list(diet,supplement),length)

x <- as.vector(barplot(tapply(gain,list(diet,supplement),mean),
beside=T,ylim=c(0,30)))
y <- as.vector(tapply(gain,list(diet,supplement),mean))
z <- rep(0.656,length(x))
for( i in 1:length(x) )
arrows(x[i],y[i]-z[i],x[i],y[i]+z[i],length=0.05,code=3,angle=90)

legend(locator(1),labels,gray(shade))

model <- lm(gain~diet+supplement)
```

```
summary(model)

supp2 <- factor(supplement)
levels(supp2)

model2 <- lm(gain~diet+supp2)
anova(model,model2)

# split plot experiments

yields <- read.csv("c:\\temp\\splityield.csv")
attach(yields)
names(yields)

model <-
aov(yield~irrigation*density*fertilizer+Error(block/irrigation/density))
summary(model)

interaction.plot(fertilizer,irrigation,yield)
interaction.plot(density,irrigation,yield)


# random effects and pseudoreplication

rats <- read.csv("c:\\temp\\rats.csv")
attach(rats)
names(rats)

Treatment <- factor(Treatment)
Rat <- factor(Rat)
Liver <- factor(Liver)

# this is the wrong way to do the analysis

model <- aov(Glycogen~Treatment)
summary(model)

# this is the right way to do the analysis

yv <- tapply(Glycogen,list(Treatment,Rat),mean)
( yv <- as.vector(yv) )

treatment <- factor(c(1,2,3,1,2,3))
model <- aov(yv~treatment)
summary(model)


# variance components analysis

model2 <- aov(Glycogen~Treatment+Error(Treatment/Rat/Liver))
summary(model2)
```

```
# analysis of covariance

compensation <- read.csv("c:\\temp\\ipomopsis.csv")
attach(compensation)
names(compensation)

plot(Root,Fruit,pch=16,col="blue")
plot(Grazing,Fruit,col="lightgreen")

# the wrong analysis (not controlling for initial size)

summary(aov(Fruit~Grazing))

# the correct anocova

model <- lm(Fruit~Root*Grazing)
summary.aov(model)

model <- lm(Fruit~Grazing*Root)
summary.aov(model)

model2 <- lm(Fruit~Grazing+Root)
anova(model,model2)
summary.lm(model2)

plot(Root,Fruit,pch=21,bg=(1+as.numeric(Grazing)))
legend(locator(1),c("grazed","ungrazed"),col=c(2,3),pch=16)
abline(-127.829,23.56,col="blue")
abline(-127.829+36.103,23.56,col="blue")
```

```
# multiple regression

ozone.pollution <- read.csv("c:\\temp\\ozone.data.csv")
attach(ozone.pollution)
names(ozone.pollution)

pairs(ozone.pollution,panel=panel.smooth)

library(mgcv)
par(mfrow=c(2,2))
model <- gam(ozone~s(rad)+s(temp)+s(wind))
plot(model,col= "blue")

par(mfrow=c(1,1))
library(tree)
model <- tree(ozone~.,data=ozone.pollution)
plot(model)
text(model)

model1 <- lm(ozone~temp*wind*rad+I(rad^2)+I(temp^2)+I(wind^2))
summary(model1)

model2 <- update(model1,~. - temp:wind:rad)
summary(model2)

model3 <- update(model2,~. - wind:rad)
summary(model3)

model4 <- update(model3,~. - temp:wind)
summary(model4)

model5 <- update(model4,~. - I(rad^2))
summary(model5)

model6 <- update(model5,~. - temp:rad)
summary(model6)

plot(model6)


# start all over again with a new transfrmation of the response

model7 <- lm(log(ozone)~temp*wind*rad+I(rad^2)+I(temp^2)+I(wind^2))

model8 <- step(model7)
summary(model8)
plot(model8)


# a more tricky example

pollute <- read.csv("c:\\temp\\sulphur.dioxide.csv")
attach(pollute)
names(pollute)

pairs(pollute,panel=panel.smooth)

par(mfrow=c(1,1))
library(tree)
model <- tree(Pollution~.,data=pollute)
plot(model)
```

```
text(model)

model1 <-
lm(Pollution~Temp+I(Temp^2)+Industry+I(Industry^2)+Population+I(Population^
2)+Wind+I(Wind^2)+Rain+I(Rain^2)+Wet.days+I(Wet.days^2))
summary(model1)

model2 <- step(model1)
summary(model2)

model3 <- update(model2, ~.- Rain-I(Wind^2))
summary(model3)

interactions <- c("ti","tp","tw","tr","td","ip","iw",
"ir","id","pw","pr","pd","wr","wd","rd")

sample(interactions)

model4 <-
lm(Pollution~Temp+Industry+Population+Wind+Rain+Wet.days+Wind:Rain+Wind:Wet
.days+Industry:Wet.days+Industry:Rain+Rain:Wet.days)
model5 <-
lm(Pollution~Temp+Industry+Population+Wind+Rain+Wet.days+Population:Rain+Te
mp:Population+Population:Wind+Temp:Industry+Industry:Wind)
model6 <-
lm(Pollution~Temp+Industry+Population+Wind+Rain+Wet.days+Temp:Wind+Populati
on:Wet.days+Temp:Rain+Temp:Wet.days+Industry:Population)

model7 <-
lm(Pollution~Temp+Industry+Population+Wind+Rain+Wet.days+Wind:Rain+Wind:Wet
.days+Population:Wind+Temp:Rain)
summary(model7)

model8 <- update(model7,~.-Temp:Rain)
summary(model8)

model9 <- update(model8,~.-Population:Wind)
summary(model9)

plot(model9)

model10 <- update(model9,~. + Wind:Rain:Wet.days)
summary(model10)
```

```
# generalised linear models

# there is no R code in this chapter
```

```
#count data

# regression

clusters <- read.csv("c:\\temp\\clusters.csv")
attach(clusters)
names(clusters)

plot(Distance,Cancers,pch=21,bg="lightblue")

model1 <- glm(Cancers~Distance,poisson)
summary(model1)

model2 <- glm(Cancers~Distance,quasipoisson)
summary(model2)

xv <- 0:100
yv <- 0.186865-0.006138*xv
y <- exp(yv)
lines(xv,y,col="red")

y <- predict(model2,list(Distance=xv), type="response")
lines(xv,y,col="red")

# categorical explanatory vaariables

count <- read.csv("c:\\temp\\cells.csv")
attach(count)
names(count)

table(cells)


tapply(cells,smoker,mean)

tapply(cells,weight,mean)

tapply(cells,sex,mean)

tapply(cells,age,mean)

model1 <- glm(cells~smoker*sex*age*weight,poisson)
summary(model1)

model2 <- glm(cells~smoker*sex*age*weight,quasipoisson)
summary(model2)

model3 <- update(model2, ~. -smoker:sex:age:weight)
model4 <- update(model3, ~. -sex:age:weight)
anova(model4,model3,test="F")

model5 <- update(model4, ~. -smoker:sex:age)
anova(model5,model4,test="F")

model6 <- update(model5, ~. -smoker:age:weight)
anova(model6,model5,test="F")

Despite 1-star significance for one of the interaction terms, this was not
significant either, so we leave it out.

model7 <- update(model6, ~. -smoker:sex:weight)
```

```
anova(model7,model6,test="F")

model8 <- update(model7, ~. -smoker:age)
anova(model8,model7,test="F")

model9 <- update(model8, ~. -sex:weight)
anova(model9,model8,test="F")

model10 <- update(model9, ~. -age:weight)
anova(model10,model9,test="F")

model11 <- update(model10, ~. -smoker:sex)
anova(model11,model10,test="F")

model12 <- update(model11, ~. -sex:age)
anova(model12,model11,test="F")

model13 <- update(model11, ~. -smoker:weight)
anova(model13,model11,test="F")

tapply(cells,list(smoker,weight),mean)
tapply(cells,list(sex,age),mean)

barplot(tapply(cells,list(smoker,weight),mean),beside=T)

weight <- factor(weight,c("normal","over","obese"))
barplot(tapply(cells,list(smoker,weight),mean),beside=T)

barplot(tapply(cells,list(smoker,weight),mean),beside=T)
legend(locator(1),c("non smoker","smoker"),fill=gray(c(0.2,0.8)))

# comlex contingency tables

induced <- read.csv("c:\\temp\\induced.csv")
attach(induced)
names(induced)

model <- glm(Count~Tree*Aphid*Caterpillar,family=poisson)

model2 <- update(model , ~ . - Tree:Aphid:Caterpillar)
anova(model,model2,test="Chi")

model3 <- update(model2 , ~ . - Aphid:Caterpillar)
anova(model3,model2,test="Chi")

# the wrong way of doing it

wrong <- glm(Count~Aphid*Caterpillar,family=poisson)
wrong1 <- update(wrong,~. - Aphid:Caterpillar)
anova(wrong,wrong1,test="Chi")

tapply(Count,list(Tree,Caterpillar),sum)

# ancova with count data

species <- read.csv("c:\\temp\\species.csv")
attach(species)
names(species)

plot(Biomass,Species,pch=21,bg=(1+as.numeric(pH)))
```

```
model <- lm(Species~Biomass*pH)
summary(model)

abline(40.60407,-2.80045,col="red")
abline(40.60407-22.75667,-2.80045-0.02733,col="green")
abline(40.60407-11.57307,-2.80045+0.23535,col="blue")

model <- glm(Species~Biomass*pH,poisson)
summary(model)

model2 <- glm(Species~Biomass+pH,poisson)
anova(model,model2,test="Chi")

plot(Biomass,Species,pch=21,bg=(1+as.numeric(pH)))
xv <- seq(0,10,0.1)
length(xv)

acidity <- rep("low",101)

yv <- predict(model,list(Biomass=xv,pH=acidity),type="response")
lines(xv,yv,col="green")

acidity <- rep("mid",101)

yv <- predict(model,list(Biomass=xv,pH=acidity),type="response")
lines(xv,yv,col="blue")

acidity <- rep("high",101)

yv <- predict(model,list(Biomass=xv,pH=acidity),type="response")
lines(xv,yv,col="red")

# frquency distributions

case.book <- read.csv("c:\\temp\\cases.csv")
attach(case.book)
names(case.book)

frequencies <- table(cases)
frequencies

mean(cases)

windows(7,4)
par(mfrow=c(1,2))

barplot(frequencies,ylab="Frequency",xlab="Cases",
col="red",main="observed")

barplot(dpois(0:10,1.775)*80,names=as.character(0:10),
ylab="Frequency",xlab="Cases",col="blue",main="expected")

var(cases)/mean(cases)

negbin <- function(x,u,k)
    (1+u/k)^(-k)*(u/(u+k))^x*gamma(k+x)/(factorial(x)*gamma(k))


xf <- numeric(11)
for (i in 0:10) xf[i+1] <- negbin(i,0.8,0.2)
barplot(xf)
```

```
mean(cases)^2/(var(cases)-mean(cases))

expected <- dnbinom(0:10,size=0.8898,mu=1.775)*80

both <- numeric(22)
both[1:22 %% 2 != 0] <- frequencies
both[1:22 %% 2 == 0] <- expected

labels <- character(22)
labels[1:22 %% 2 == 0] <- as.character(0:10)

barplot(both,col=rep(c("lightgray","darkgray"),11),names=labels,ylab="Frequ
ency",xlab="Cases")

legend(locator(1),c("Observed","Expected"), fill=c("lightgray","darkgray"))

cs <- factor(0:10)
levels(cs)[6:11] <- "5+"
levels(cs)

ef <- as.vector(tapply(expected,cs,sum))
of <- as.vector(tapply(frequencies,cs,sum))

sum((of-ef)^2/ef)

1 - pchisq(2.581842,3)
```

```
# proportion data


# logistic regression

numbers <- read.csv("c:\\temp\\sexratio.csv")
numbers

attach(numbers)
windows(7,4)
par(mfrow=c(1,2))
p <- males/(males+females)
plot(density,p,ylab="Proportion male")
plot(log(density),p,ylab="Proportion male")

y <- cbind(males,females)

model <- glm(y~density,binomial)
summary(model)

model <- glm(y~log(density),binomial)
summary(model)

xv <- seq(0,6,0.1)
plot(log(density),p,ylab="Proportion male",pch=21,bg="blue")
lines(xv,predict(model,list(density=exp(xv)),
type="response"),col="brown")

# catagorical explanatory variables

germination <- read.csv("c:\\temp\\germination.csv")
attach(germination)
names(germination)

y <- cbind(count , sample-count)

model <- glm(y ~ Orobanche * extract, binomial)
summary(model)

model <- glm(y ~ Orobanche * extract, quasibinomial)

model2 <- update(model, ~ . - Orobanche:extract)

anova(model,model2,test="F")

anova(model2,test="F")

model3 <- update(model2, ~ . - Orobanche)
anova(model2,model3,test="F")

coef(model3)

1/(1+1/(exp(-0.5122)))
1/(1+1/(exp(-0.5122+1.0574)))

tapply(predict(model3,type="response"),extract,mean)

p <- count/sample
tapply(p,extract,mean)

as.vector(tapply(count,extract,sum))/
```

```
as.vector(tapply(sample,extract,sum))

# ancova with proportion data

props <- read.csv("c:\\temp\\flowering.csv")
attach(props)
names(props)

y <- cbind(flowered,number-flowered)
pf <- flowered/number
pfc <- split(pf,variety)
dc <- split(dose,variety)

plot(dose,pf,type="n",ylab="Proportion flowered")
points(jitter(dc[[1]]),jitter(pfc[[1]]),pch=21,bg="red")
points(jitter(dc[[2]]),jitter(pfc[[2]]),pch=22,bg="blue")
points(jitter(dc[[3]]),jitter(pfc[[3]]),pch=23,bg="gray")
points(jitter(dc[[4]]),jitter(pfc[[4]]),pch=24,bg="green")
points(jitter(dc[[5]]),jitter(pfc[[5]]),pch=25,bg="yellow")

model1 <- glm(y~dose*variety,binomial)
summary(model1)

model2 <- glm(y~dose*variety,quasibinomial)
summary(model2)

model3 <- glm(y~dose+variety,quasibinomial)
anova(model2,model3,test="F")


xv <- seq(0,32,0.25)
length(xv)

yv <- predict(model3,list(dose=xv,variety=rep("A",129)),type="response")
lines(xv,yv,col="red")
yv <- predict(model3,list(dose=xv,variety=rep("B",129)),type="response")
lines(xv,yv,col="blue")
yv <- predict(model3,list(dose=xv,variety=rep("C",129)),type="response")
lines(xv,yv,col="gray")
yv <- predict(model3,list(dose=xv,variety=rep("D",129)),type="response")
lines(xv,yv,col="green")
yv <- predict(model3,list(dose=xv,variety=rep("E",129)),type="response")
lines(xv,yv,col="yellow")
```

```
# binary response variables

island <- read.csv("c:\\temp\\isolation.csv")
attach(island)
names(island)

[1] "incidence" "area"      "isolation"

model1 <- glm(incidence~area*isolation,binomial)

model2 <- glm(incidence~area+isolation,binomial)

anova(model1,model2,test="Chi")
summary(model2)

windows(7,4)
par(mfrow=c(1,2))
xv <- seq(0,9,0.01)

modela <- glm(incidence~area,binomial)
modeli <- glm(incidence~isolation,binomial)

yv <- predict(modela,list(area=xv),type="response")
plot(area,incidence,pch=21,bg="yellow")
lines(xv,yv,col="blue")

xv2 <- seq(0,10,0.1)
yv2 <- predict(modeli,list(isolation=xv2),type="response")
plot(isolation,incidence,pch=21,bg="yellow")
lines(xv2,yv2,col="red")

ac <- cut(area,3)
ic <- cut(isolation,3)
tapply(incidence,ac,sum)
tapply(incidence,ic,sum)

table(ac)
table(ic)

tapply(incidence,ac,sum)/ table(ac)
tapply(incidence,ic,sum)/ table(ic)

xv <- seq(0,9,0.01)
yv <- predict(modela,list(area=xv),type="response")
plot(area,incidence,pch=21,bg="yellow")
lines(xv,yv,col="blue")

d <- (max(area)-min(area))/3
left <- min(area)+d/2
mid <- left+d
right <- mid+d
xva <- c(left,mid,right)
pa <- as.vector(tapply(incidence,ac,sum)/ table(ac))
se <- sqrt(pa*(1-pa)/table(ac))

xv <- seq(0,9,0.01)
yv <- predict(modela,list(area=xv),type="response")
lines(xv,yv,col="blue")

points(xva,pa,pch=16,col="red")
for (i in 1:3) lines(c(xva[i],xva[i]),
```

```
c(pa[i]+se[i],pa[i]-se[i]),col="red" )

xv2 <- seq(0,10,0.1)
yv2 <- predict(modeli,list(isolation=xv2),type="response")
plot(isolation,incidence,pch=21,bg="yellow")
lines(xv2,yv2,col="red")

d <- (max(isolation)-min(isolation))/3
left <- min(isolation)+d/2
mid <- left+d
right <- mid+d
xvi <- c(left,mid,right)
pi <- as.vector(tapply(incidence,ic,sum)/ table(ic))
se <- sqrt(pi*(1-pi)/table(ic))

points(xvi,pi,pch=16,col="blue")
for (i in 1:3) lines(c(xvi[i],xvi[i]),
c(pi[i]+se[i],pi[i]-se[i]),col="blue" )

# binary ancova

infection <- read.csv("c:\\temp\\infection.csv")
attach(infection)
names(infection)

windows(7,4)
par(mfrow=c(1,2))
plot(infected,weight,xlab="Infection",ylab="Weight",col="lightblue")
plot(infected,age,xlab="Infection",ylab="Age", col="lightgreen")

table(infected,sex)

model <- glm(infected~age*weight*sex,family=binomial)
summary(model)

model2 <- step(model)

summary(model2)

model3 <- update(model2,~.-age:weight)
anova(model2,model3,test="Chi")

model4 <- update(model2,~.-age:sex)
anova(model2,model4,test="Chi")

model5 <- glm(infected~age+weight+sex,family=binomial)
summary(model5)

model6 <- glm(infected~age+weight+sex+I(weight^2)+I(age^2),family=binomial)
summary(model6)


library(mgcv)
model7 <- gam(infected~sex+s(age)+s(weight),family=binomial)
plot.gam(model7)

model8 <- glm(infected~sex+age+I(age^2)+
I((weight-12)*(weight>12)),family=binomial)
summary(model8)

model9 <- update(model8,~.-sex)
```

```
anova(model8,model9,test="Chi")

model10 <- update(model8,~.-I(age^2))
anova(model8,model10,test="Chi")

summary(model9)
```

```
#  age at death data

mortality <- read.csv("c:\\temp\\deaths.csv")
attach(mortality)
names(mortality)

tapply(death,treatment,mean)

tapply(death,treatment,var)

model <- glm(death~treatment,Gamma)
summary(model)

detach(mortality)

# survival analysis with censoring

library(survival)
sheep <- read.csv("c:\\temp\\sheep.deaths.csv")
attach(sheep)
names(sheep)

plot(survfit(Surv(death,status)~group),col=c(2,3,4),
xlab="Age at death (months)")

model <- survreg(Surv(death,status)~weight*group,dist="exponential")
summary(model)

model2 <- survreg(Surv(death,status)~weight+group,dist="exponential")
anova(model,model2,test="Chi")

model3 <- survreg(Surv(death,status)~group,dist="exponential")
anova(model2,model3,test="Chi")

model4 <- survreg(Surv(death,status)~1,dist="exponential")
anova(model3,model4,test="Chi")

summary(model3)

model3 <- survreg(Surv(death,status)~group,dist="exponential")
model4 <- survreg(Surv(death,status)~group,dist="extreme")
model5 <- survreg(Surv(death,status)~group,dist="gaussian")
model6 <- survreg(Surv(death,status)~group,dist="logistic")
anova(model3,model4,model5,model6)

tapply(predict(model3,type="response"),group,mean)

tapply(death,group,mean)
```