#2.3 Lab Intro to R ¶

#2.3.1

In [133]:

```
x = c(1,3,2,5)
x
```

```
1  3  2  5
```

In [134]:

```
x = c(1, 6, 2)
x
```

```
1  6  2
```

In [135]:

```
y = c(1, 4, 3)
y
```

```
1  4  3
```

In [136]:

```
length(x)
```

```
3
```

In [137]:

```
length(y)
```

```
3
```

In [138]:

```
x + y
```

```
2  10  5
```

In [139]:

```
#ls():get a list of all objects in the current environment
#rm()Objects can be removed from the environment with this fucn.
ls()
```

```
'x'  'y'
```

In [140]:

```
rm(x, y)
ls()

```

In [141]:

```
#To remove all objects from the environment, we first get
#the list of all objects with the ls() function, and pass this lis
rm(list = ls())
```

In [142]:

```
#to get help
?matrix
```

In [143]:

```
x=matrix(data=c(1,2,3,4), nrow=2, ncol=2)  #create a matrix
x
```

```
1  3
2  4
```

In [144]:

```
#following matrix creation with data in row-order.
matrix(c(1, 2, 3, 4), 2, 2, byrow = TRUE)
```

```
1  2
3  4
```

In [145]:

```
sqrt(x) #sqrt of matrix
```

```
1.000000  1.732051
1.414214  2.000000
```

In [146]:

```
x^2 # ^ operator raise each element of the matrix to a power
```

```
1   9
4  16
```

In [147]:

```
#The rnorm() can be used to generate random no.s
#rnorm() creates standard normal random variables
#with a mean of 0 and a standard deviation of 1.
x = rnorm(50)
y = x + rnorm(50, mean = 50, sd = 0.1)
cor(x, y)
```

```
0.995528952267613
```

In [148]:

```
#to get  consistent results
set.seed(1303)
rnorm(50)
```

```
-1.14397631447974   1.34212936561501   2.18539047574276   0.536392517923731
0.0631929664685468   0.502234482468979   -0.000416724686432643
0.565819840539162   -0.572522688962623   -1.11022500727696   -0.0486871233624514
-0.695656217619366   0.828917480303335   0.206652855081802   -0.235674509102427
-0.556310491381104   -0.364754357080585   0.862355034263622   -0.63077153536771
0.313602125215739   -0.931495317661393   0.823867618473952   0.523370702077482
0.706921411979056   0.420204325601679   -0.269052154682033   -1.51031729990999
-0.69021247657504   -0.143471952443572   -1.0135274099044   1.57327373614751
0.0127465054882014   0.872647049887217   0.422066190530336   -0.0188157916578866
2.61574896890584   -0.693140174826871   -0.266321780991085   -0.720636441231524
1.36773420645149   0.264007332160512   0.632186807367191   -1.33065098578719
0.0268888182209596   1.0406363207788   1.31202379854711   -0.0300020766733214
-0.250025712488174   0.0234144856913592   1.65987065574227
```

In [149]:

```
set.seed(3)
y = rnorm(100)
mean(y)

#mean() sd() var() can be used to calculate the
#mean,standard dev,variance of a vector
```

```
0.0110355710943715
```

In [150]:

```
var(y)
```

```
0.732867501277449
```

In [151]:

```
sqrt(var(y))
```

```
0.856076808047881
```

In [152]:

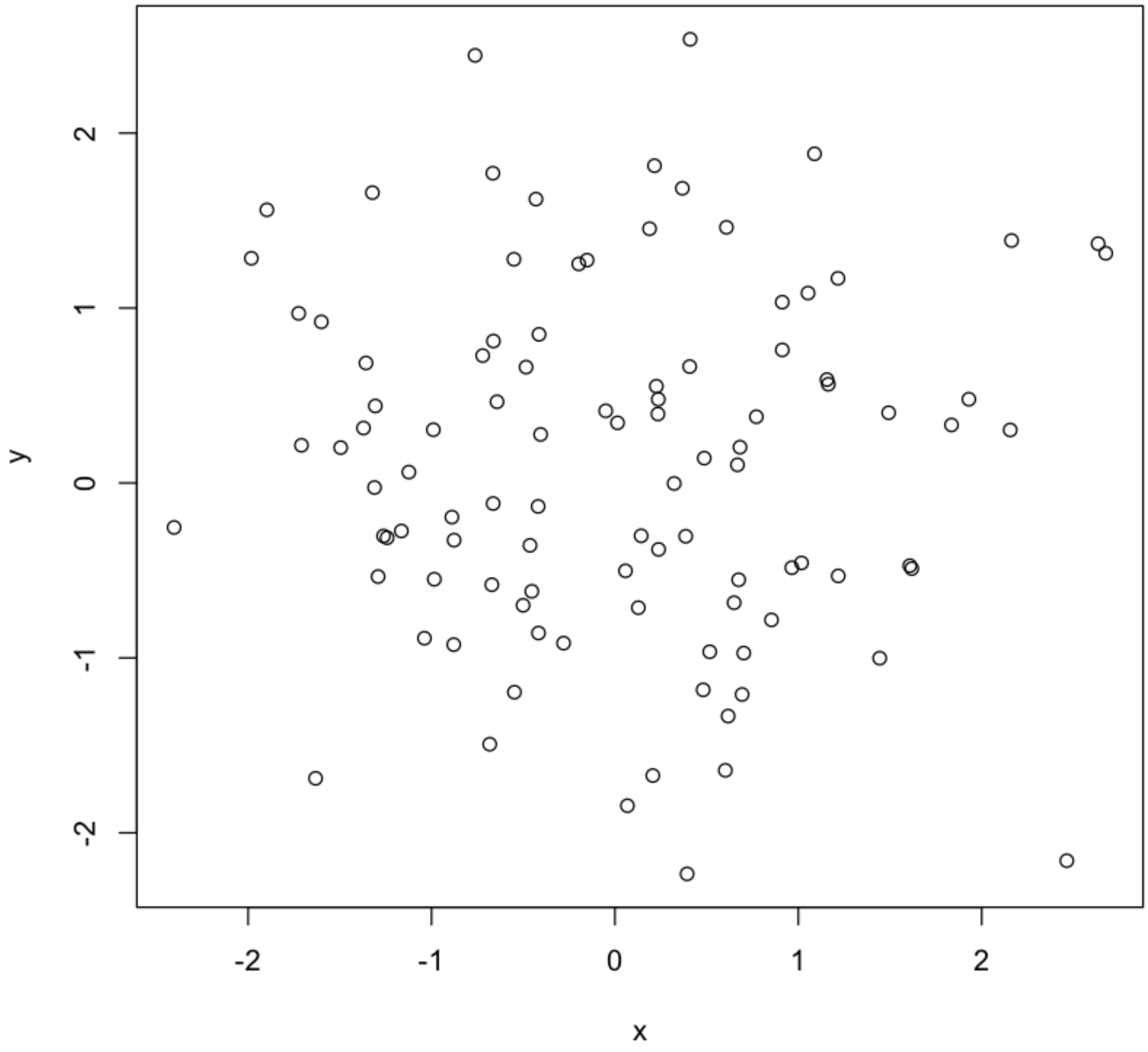
```
sd(y)
```

```
0.856076808047881
```

# 2.3.2 Graphics

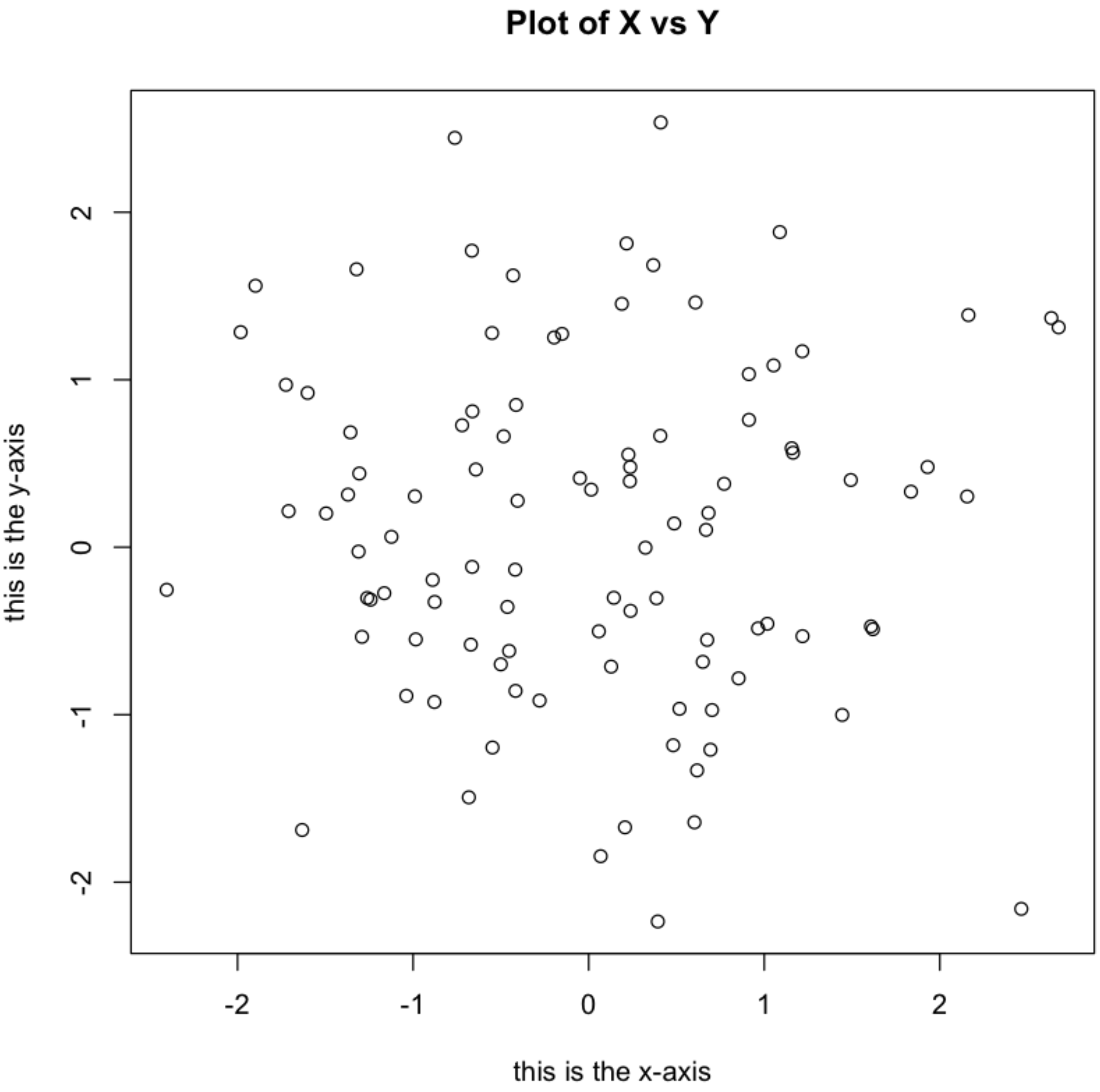
In [153]:

```
x = rnorm(100)
y = rnorm(100)
plot(x, y)
```



In [154]:

```
plot(x, y, xlab = "this is the x-axis", ylab = "this is the y-axis
```



In [155]:

```
pdf("Figure.pdf")    #save figures
plot(x, y, col = "green")
dev.off()
```

pdf: 2

In [156]:

```
x = seq(1, 10)  #to generate seq of numbers
x
```

```
1  2  3  4  5  6  7  8  9  10
```

In [157]:

```
x = 1:10
x
```

```
1  2  3  4  5  6  7  8  9  10
```

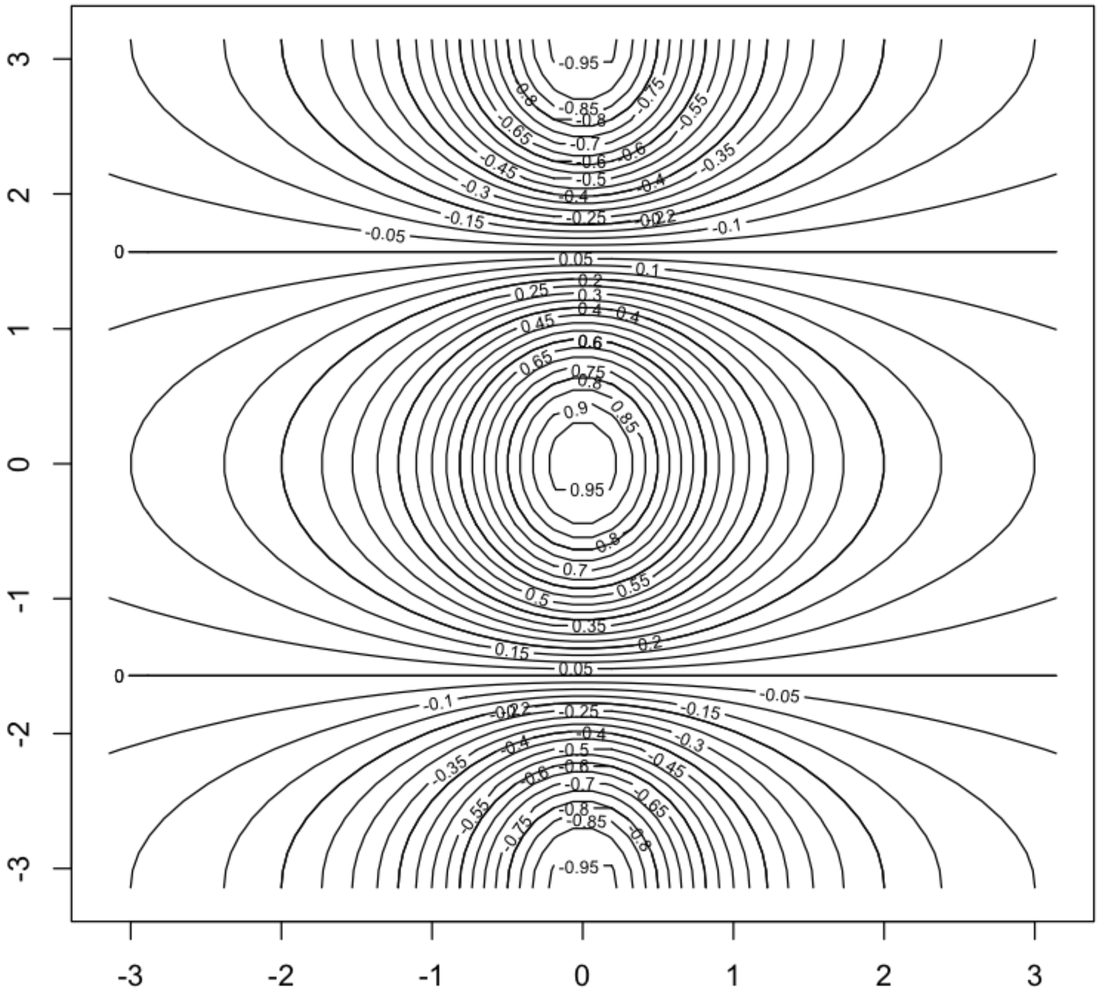
In [158]:

```
x = seq(-pi, pi, length = 50)
x
```

```
-3.14159265358979   -3.0133643820147   -2.88513611043961   -2.75690783886451
-2.62867956728942   -2.50045129571433   -2.37222302413923   -2.24399475256414
-2.11576648098904   -1.98753820941395   -1.85930993783886   -1.73108166626376
-1.60285339468867   -1.47462512311358   -1.34639685153848   -1.21816857996339
-1.0899403083883   -0.961712036813202   -0.833483765238109   -0.705255493663015
-0.577027222087922   -0.448798950512828   -0.320570678937734
-0.192342407362641   -0.064114135787547   0.0641141357875465   0.19234240736264
0.320570678937734   0.448798950512828   0.577027222087921   0.705255493663015
0.833483765238108   0.961712036813202   1.0899403083883   1.21816857996339
1.34639685153848   1.47462512311358   1.60285339468867   1.73108166626376
1.85930993783886   1.98753820941395   2.11576648098904   2.24399475256414
2.37222302413923   2.50045129571433   2.62867956728942   2.75690783886451
2.88513611043961   3.0133643820147   3.14159265358979
```

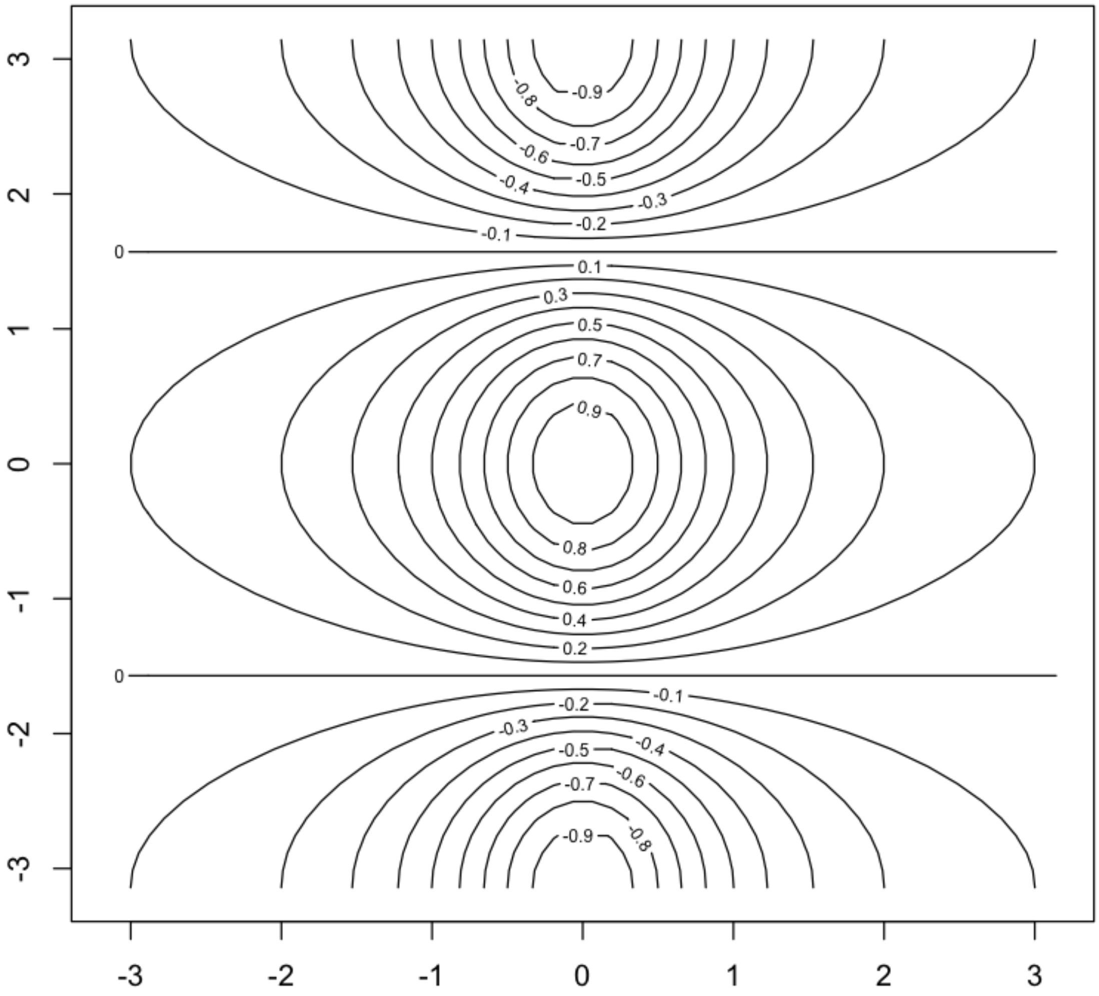
In [159]:

```
#contour plots
y = x
f = outer(x, y, function(x, y) cos(y)/(1 + x^2))   #outer product
contour(x, y, f)
contour(x, y, f, nlevels = 45, add = T)   #compute f at every poir
```



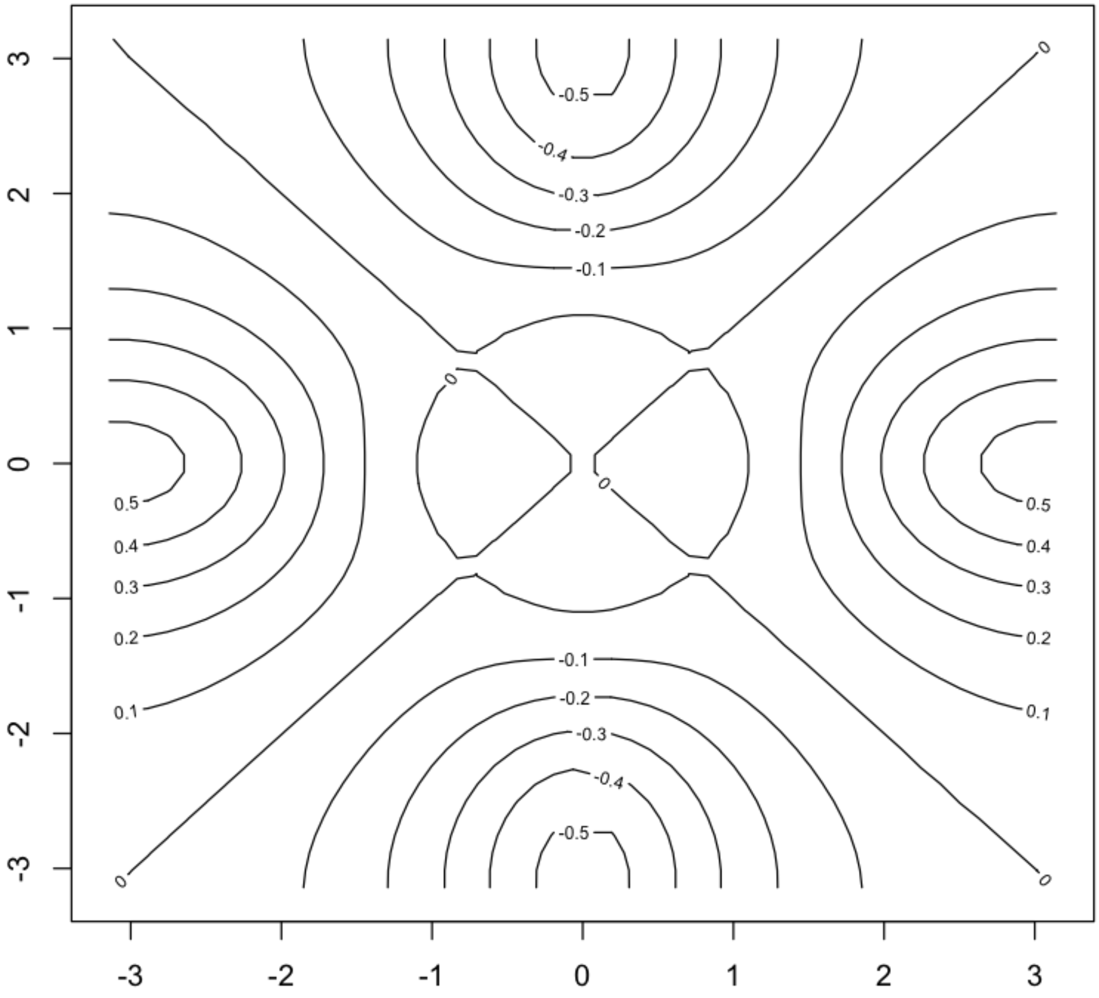
In [160]:

```
contour(x, y, f, nlevels = 15)
```



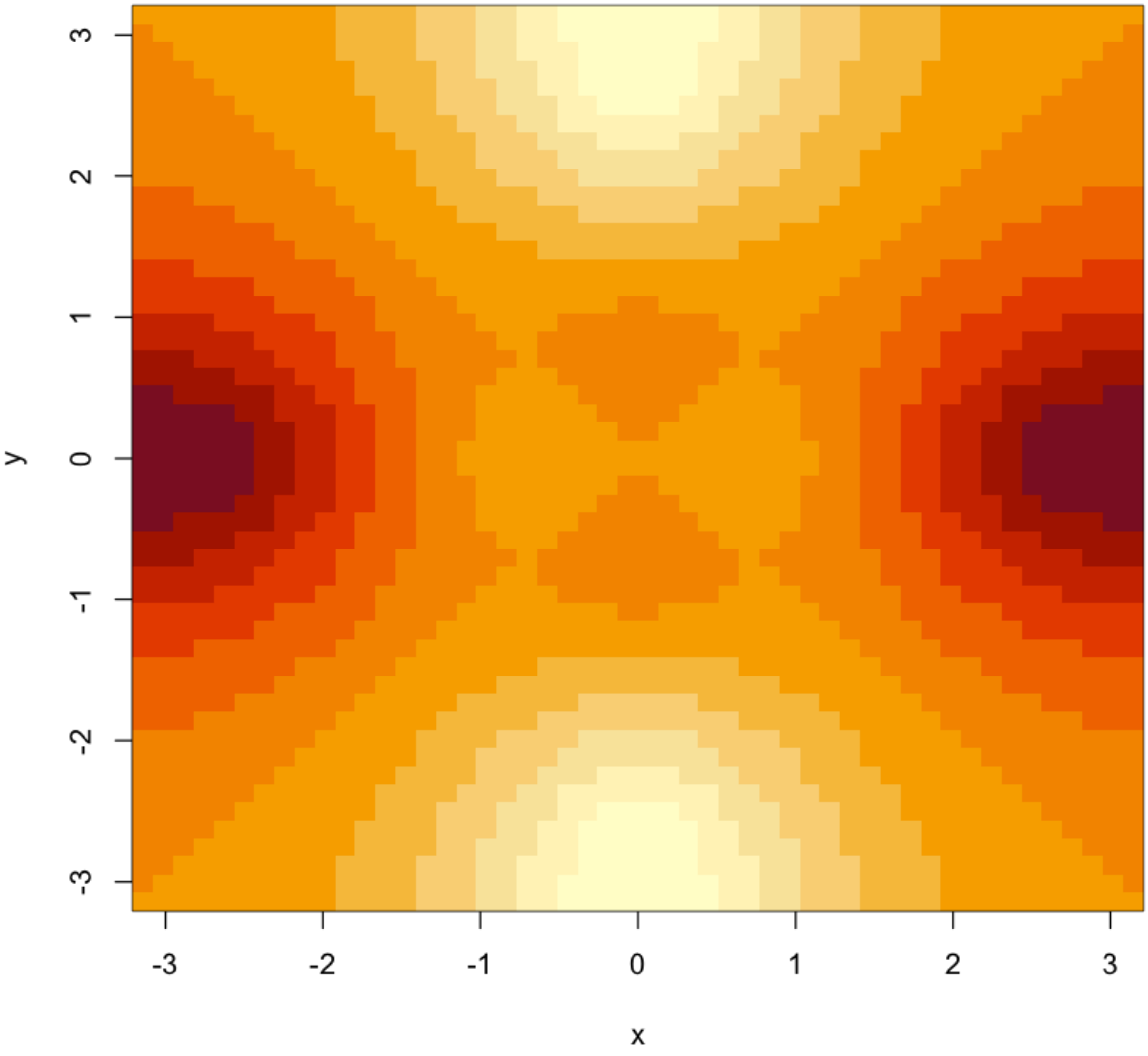
In [161]:

```
fa = (f - t(f))/2   #skew sym matrix
contour(x, y, fa, nlevels = 15)
```



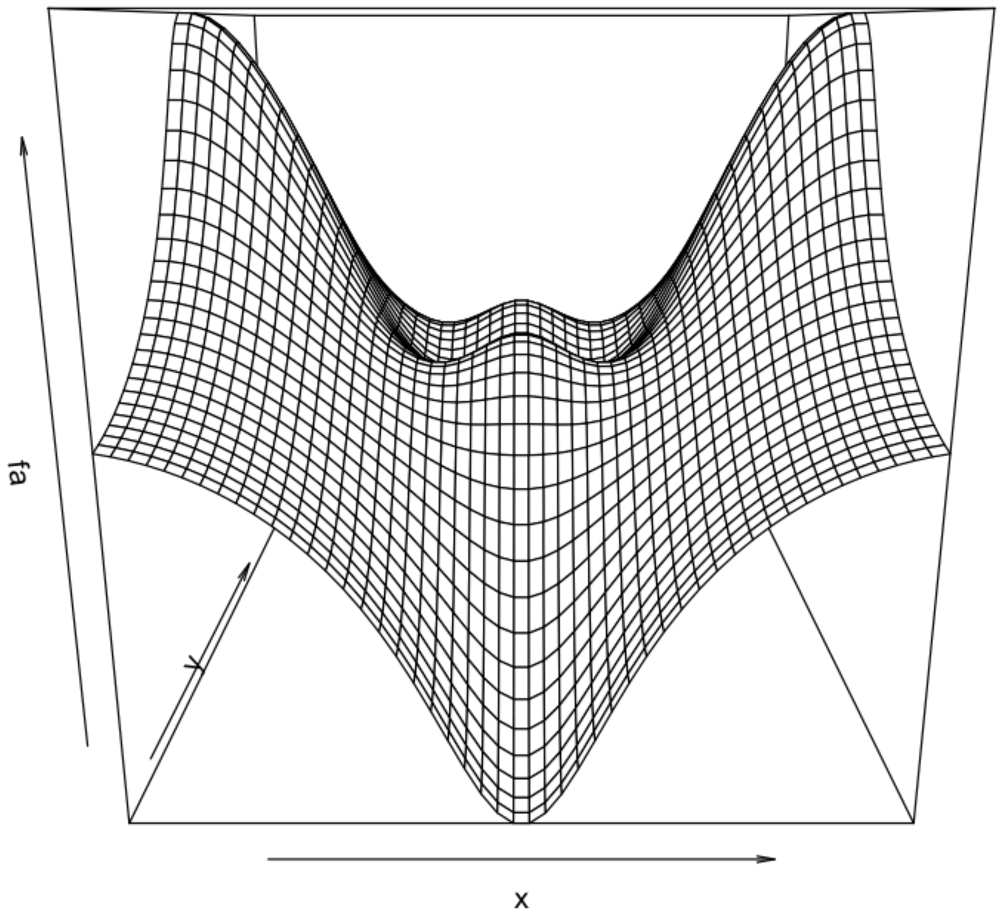
In [162]:

```
image(x, y, fa)
```



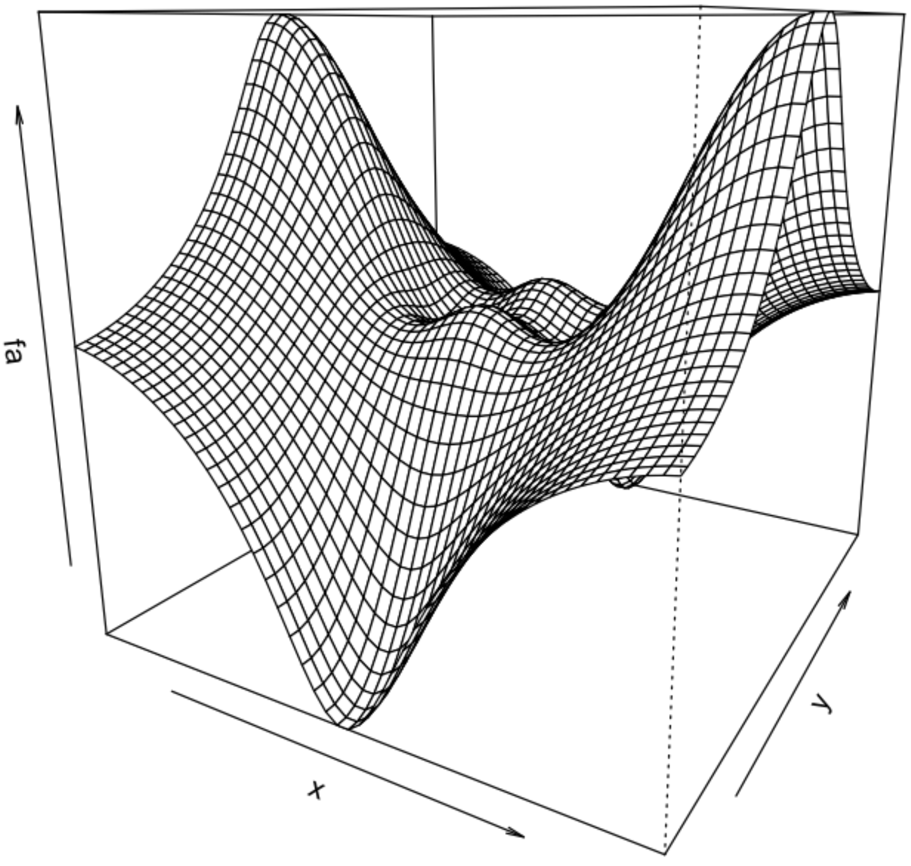
In [163]:

```
persp(x,y,fa)  #produce a three-dimensional plot
```



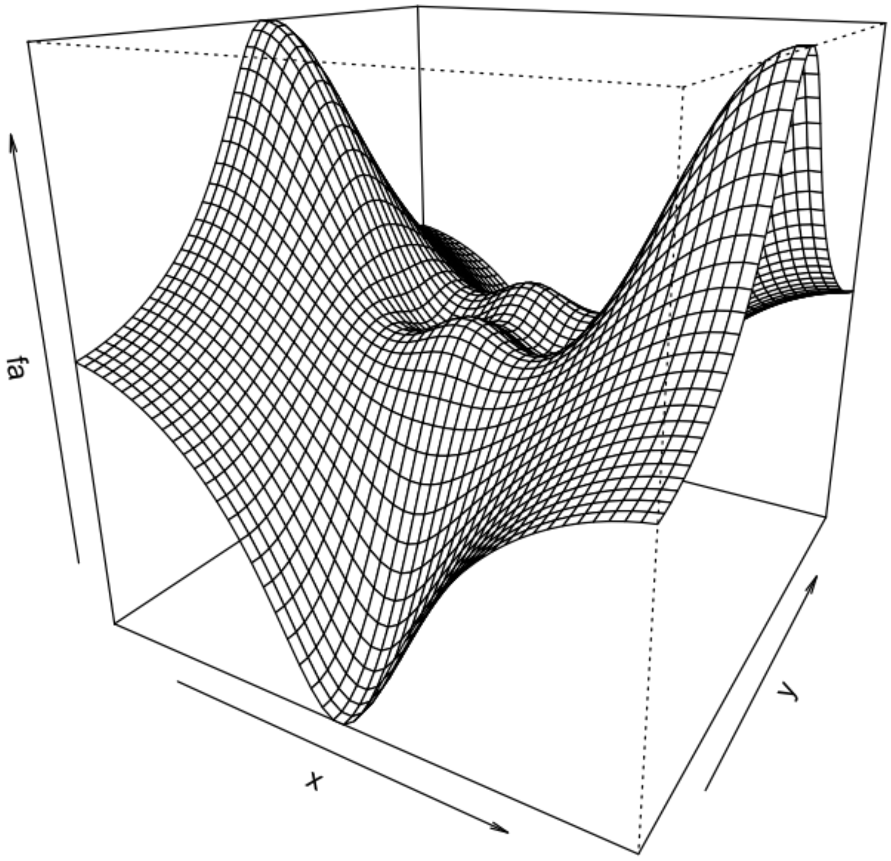
In [164]:

```
persp(x,y,fa,theta=30)
```



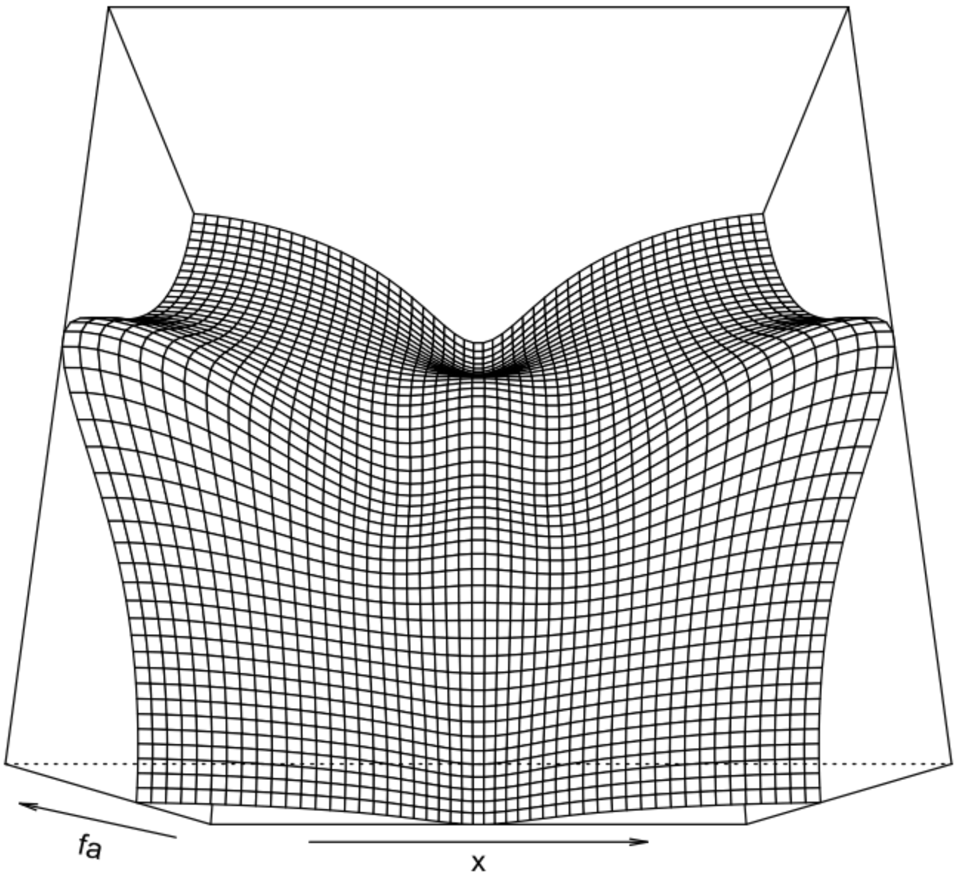
In [165]:

```
persp(x,y,fa,theta=30,phi=20)
```



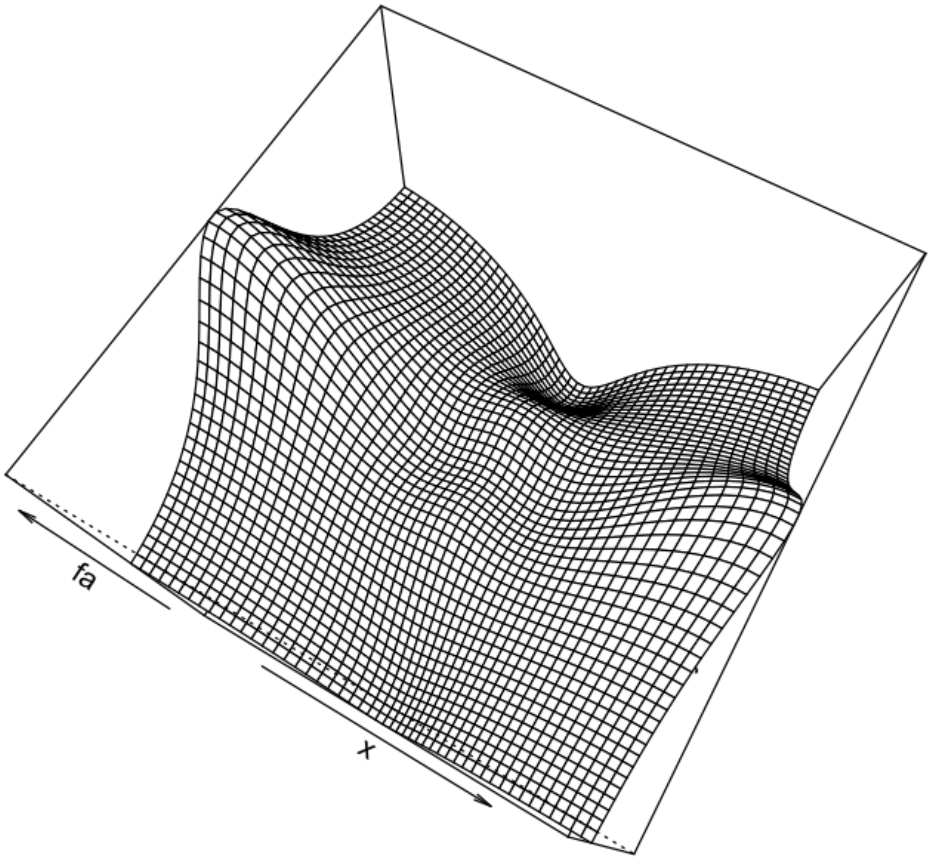
In [166]:

```
persp(x,y,fa,theta=0,phi=70)
```



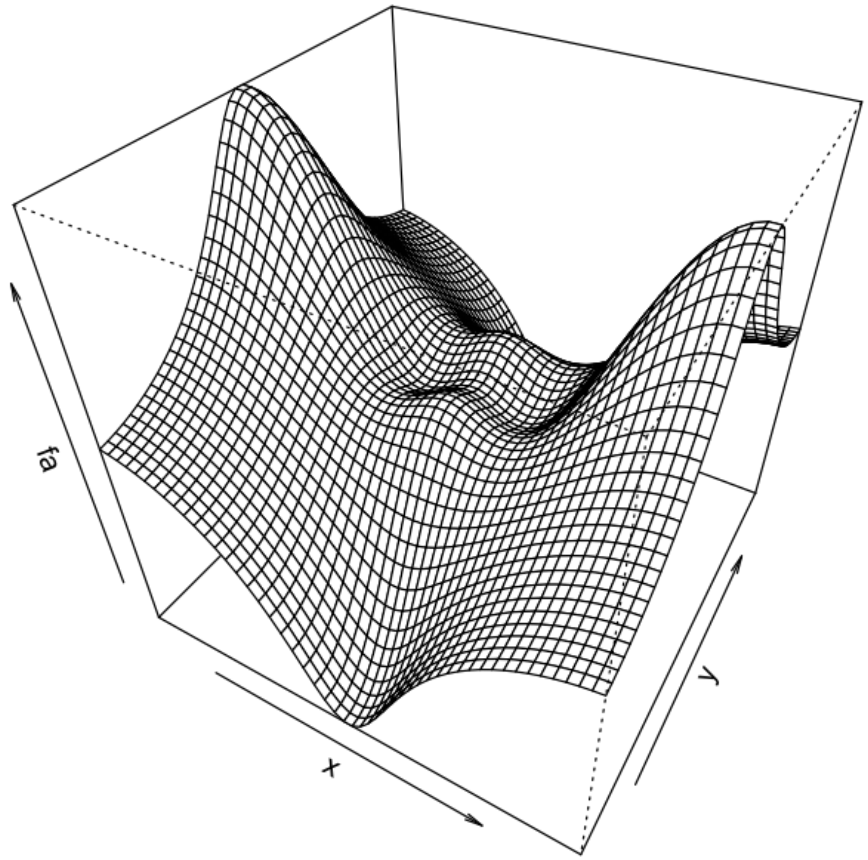
In [167]:

```
persp(x,y,fa,theta=30,phi=70)
```



In [168]:

```
persp(x,y,fa,theta=30,phi=40)
```



In [169]:

```
A=matrix(1:16,4,4)
A
```

```
1  5   9  13
2  6  10  14
3  7  11  15
4  8  12  16
```

In [170]:

```
A[2,3]
```

```
10
```

In [171]:

```
A[c(1,3),c(2,4)]
```

```
5  13
7  15
```

In [172]:

```
 A[1:3,2:4]
```

```
5   9  13
6  10  14
7  11  15
```

In [173]:

```
A [1:2 ,]
```

```
1  5   9  13
2  6  10  14
```

In [174]:

```
A [ ,1:2]
```

```
1  5
2  6
3  7
4  8
```

In [175]:

```
 A[1,]
```

```
1  5  9  13
```

In [176]:

```
A[-c(1,3),]
```

```
2  6  10  14
4  8  12  16
```

In [177]:

```
dim(A)
```

```
4  4
```

#2.3.4 Loading data

In [178]:
```
Auto = read.csv("/Users/priyanka/desktop/Auto.csv")
```

In [179]:
```
dim(Auto)
```

```
397  9
```

In [180]:
```
head(Auto)
```

mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name
18	8	307	130	3504	12.0	70	1	chevrolet chevelle malibu
15	8	350	165	3693	11.5	70	1	buick skylark 320
18	8	318	150	3436	11.0	70	1	plymouth satellite
16	8	304	150	3433	12.0	70	1	amc rebel sst
17	8	302	140	3449	10.5	70	1	ford torino
15	8	429	198	4341	10.0	70	1	ford galaxie 500

In [181]:
```
Auto = read.csv("/Users/priyanka/desktop/Auto.csv", header=T,na.st
Auto=na.omit(Auto)
```

In [182]:
```
dim(Auto)
```

```
392  9
```

In [183]:
```
Auto [1:4 ,]
```

mpg	cylinders	displacement	horsepower	weight	acceleration	year	origin	name
18	8	307	130	3504	12.0	70	1	chevrolet chevelle malibu
15	8	350	165	3693	11.5	70	1	buick skylark 320
18	8	318	150	3436	11.0	70	1	plymouth satellite
16	8	304	150	3433	12.0	70	1	amc rebel sst

In [184]:

```
Auto=na.omit(Auto)
```

In [185]:

```
dim(Auto)
```

```
392  9
```

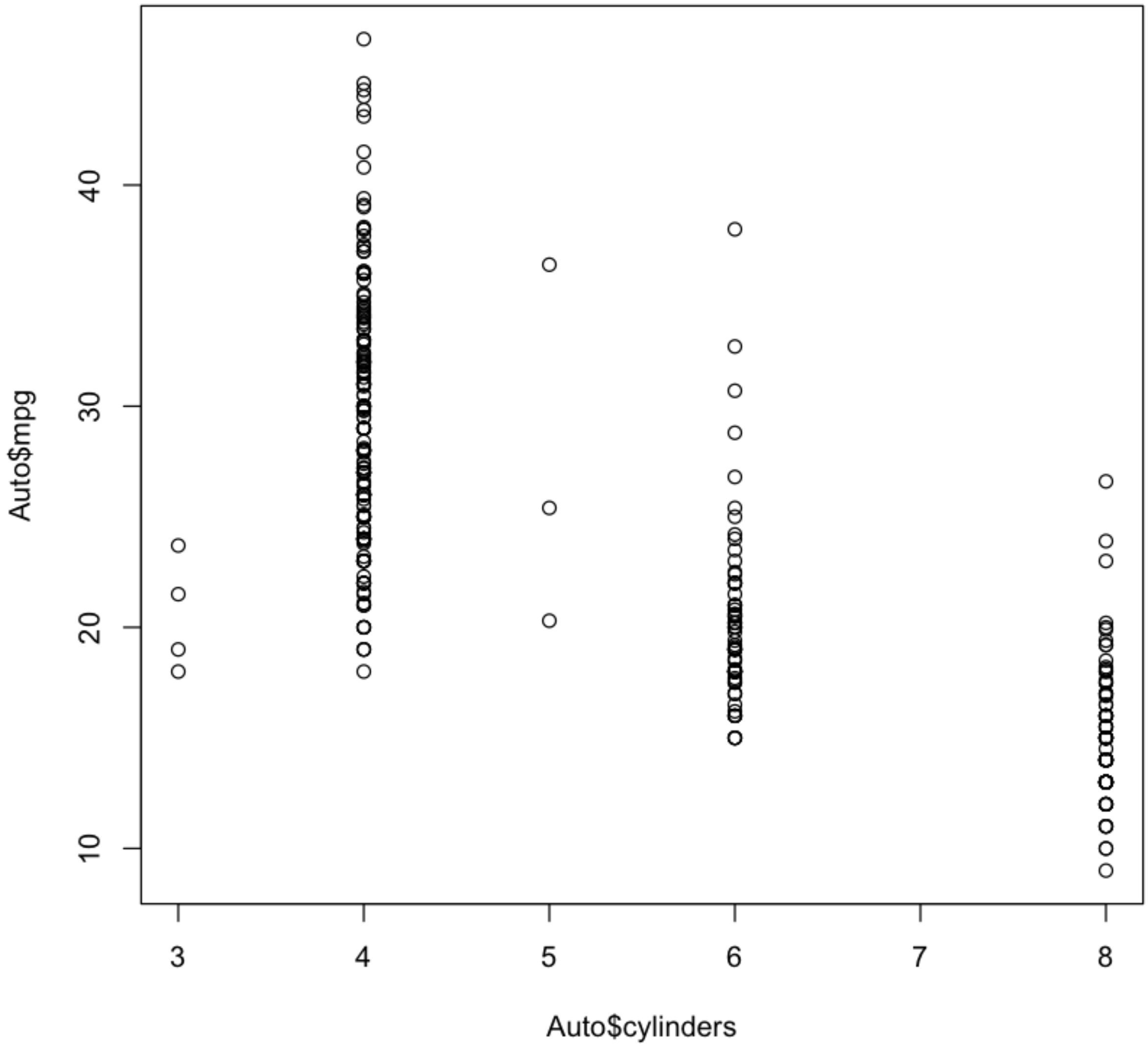
In [186]:

```
names(Auto)
```

```
'mpg'   'cylinders'   'displacement'   'horsepower'   'weight'   'acceleration'   'year'
'origin'   'name'
```

In [187]:

```
plot(Auto$cylinders , Auto$mpg )
```

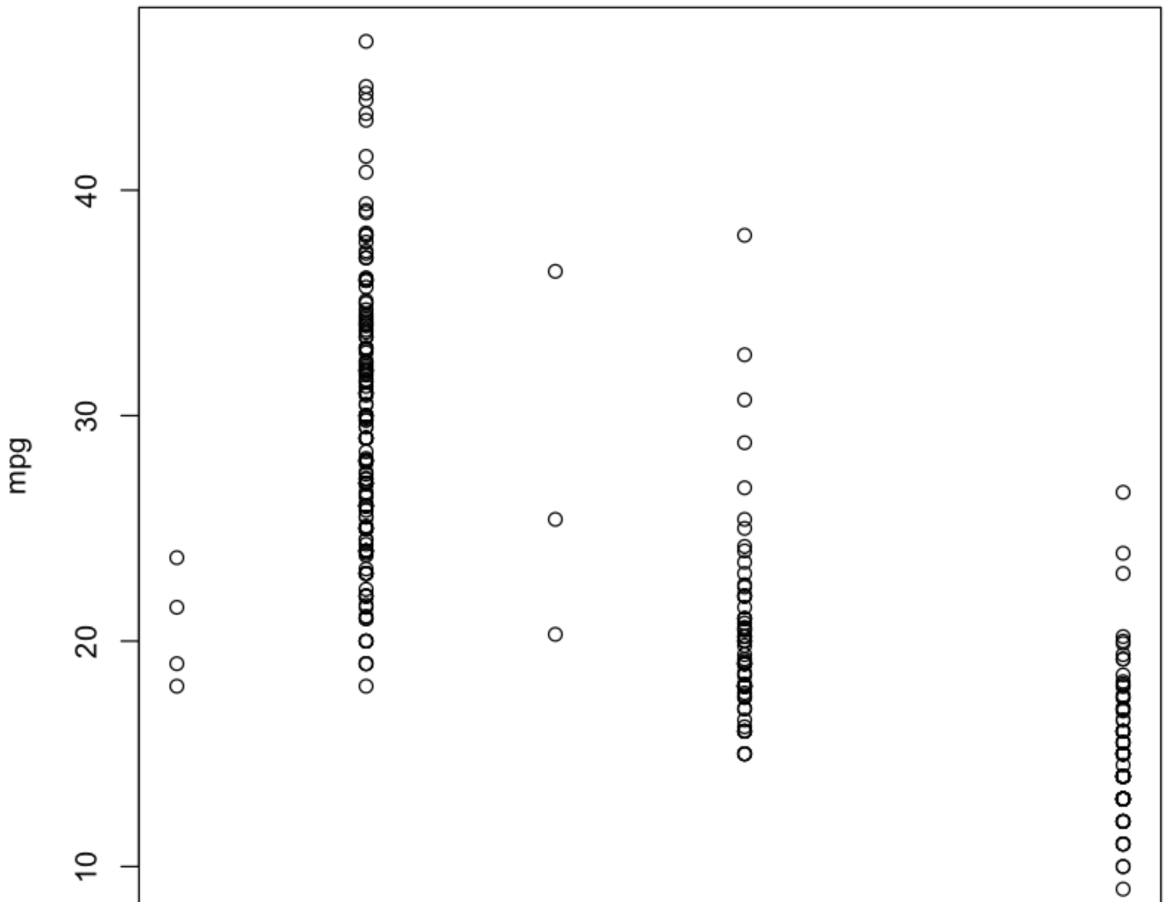


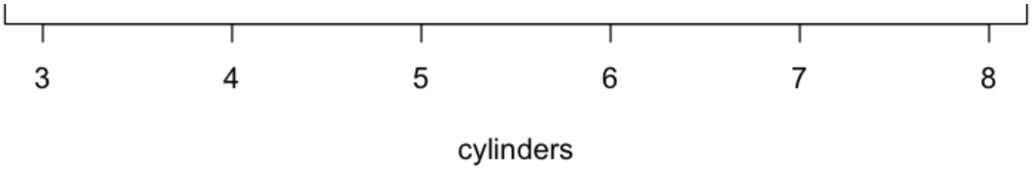
In [188]:

```
attach(Auto)
plot(cylinders,mpg)
```

```
The following objects are masked from Auto (pos = 3):

    acceleration, cylinders, displacement, horsepower, mpg, name,
    origin, weight, year

The following objects are masked from Auto (pos = 4):

    acceleration, cylinders, displacement, horsepower, mpg, name,
    origin, weight, year
```

```
The following objects are masked from Auto (pos = 5):

    acceleration, cylinders, displacement, horsepower, mpg, name,
    origin, weight, year

The following objects are masked from Auto (pos = 6):

    acceleration, cylinders, displacement, horsepower, mpg, name,
    origin, weight, year

The following objects are masked from Auto (pos = 7):

    acceleration, cylinders, displacement, horsepower, mpg, name,
    origin, weight, year

The following objects are masked from Auto (pos = 8):

    acceleration, cylinders, displacement, horsepower, mpg, name,
    origin, weight, year
```





In [189]:

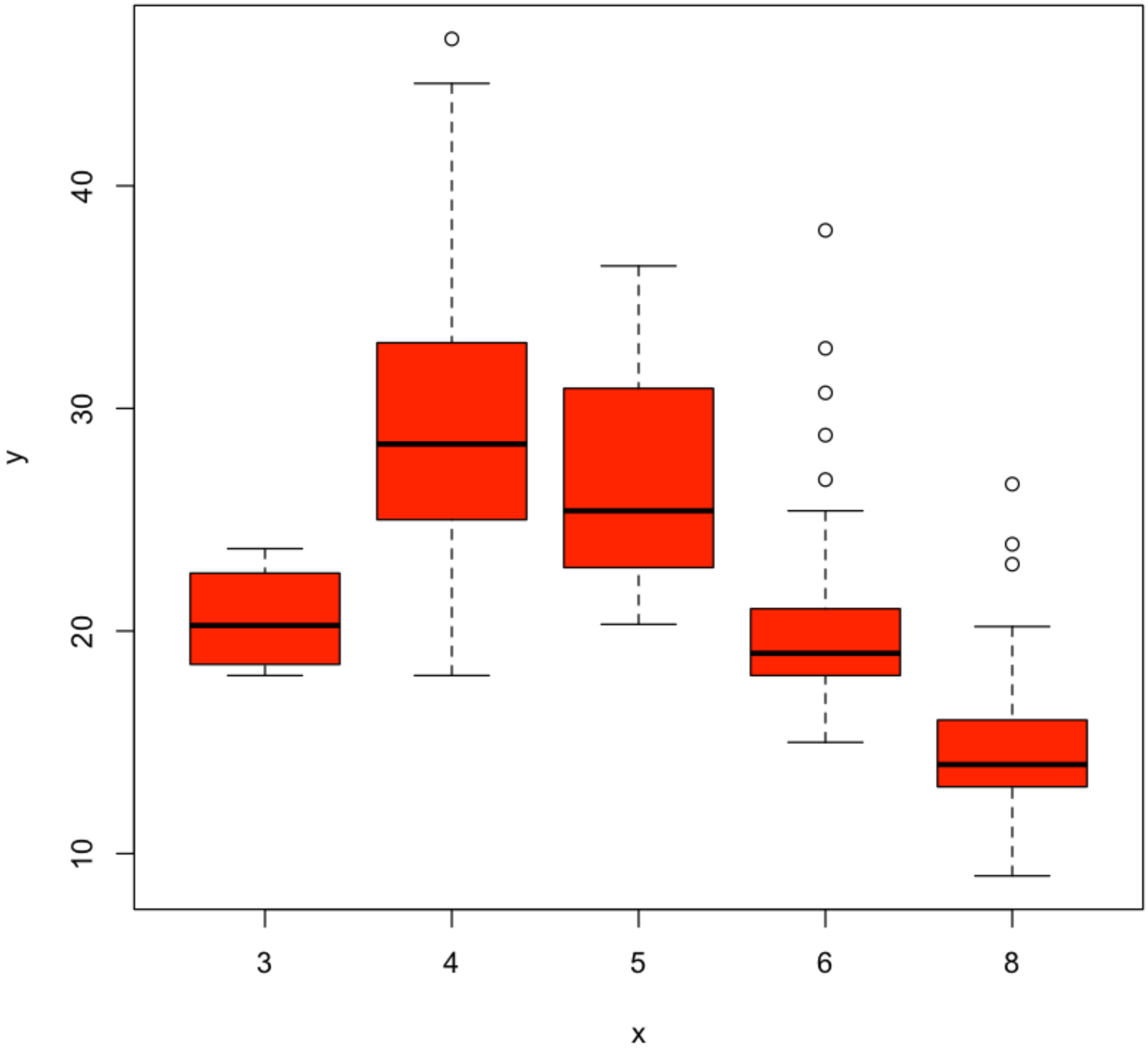
```
cylinders=as.factor(cylinders)
#converts quantitative variables into qualitative variables
```

In [190]:

```
?plot
```

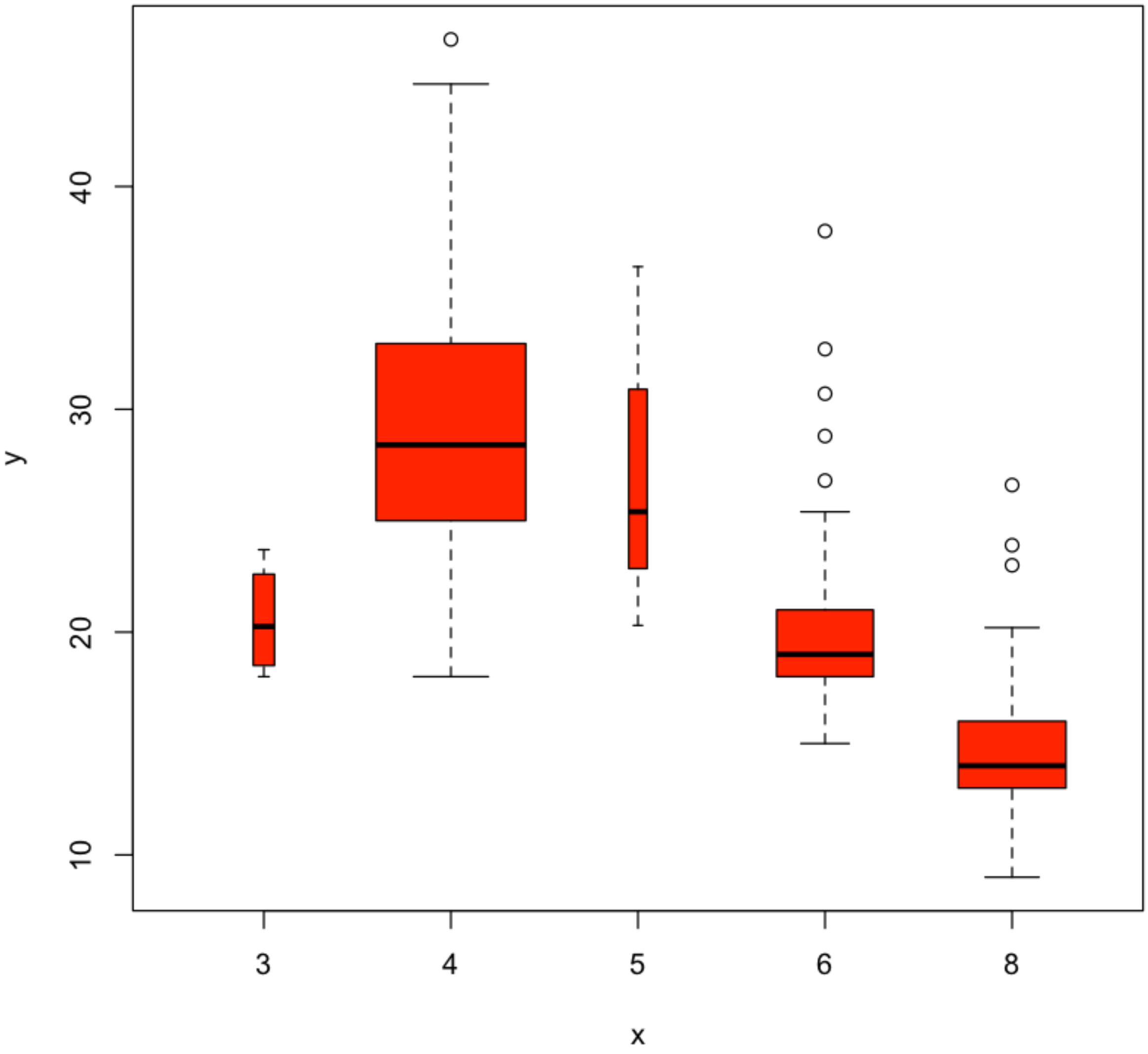
In [191]:

```
plot(cylinders,mpg, col="red")
```



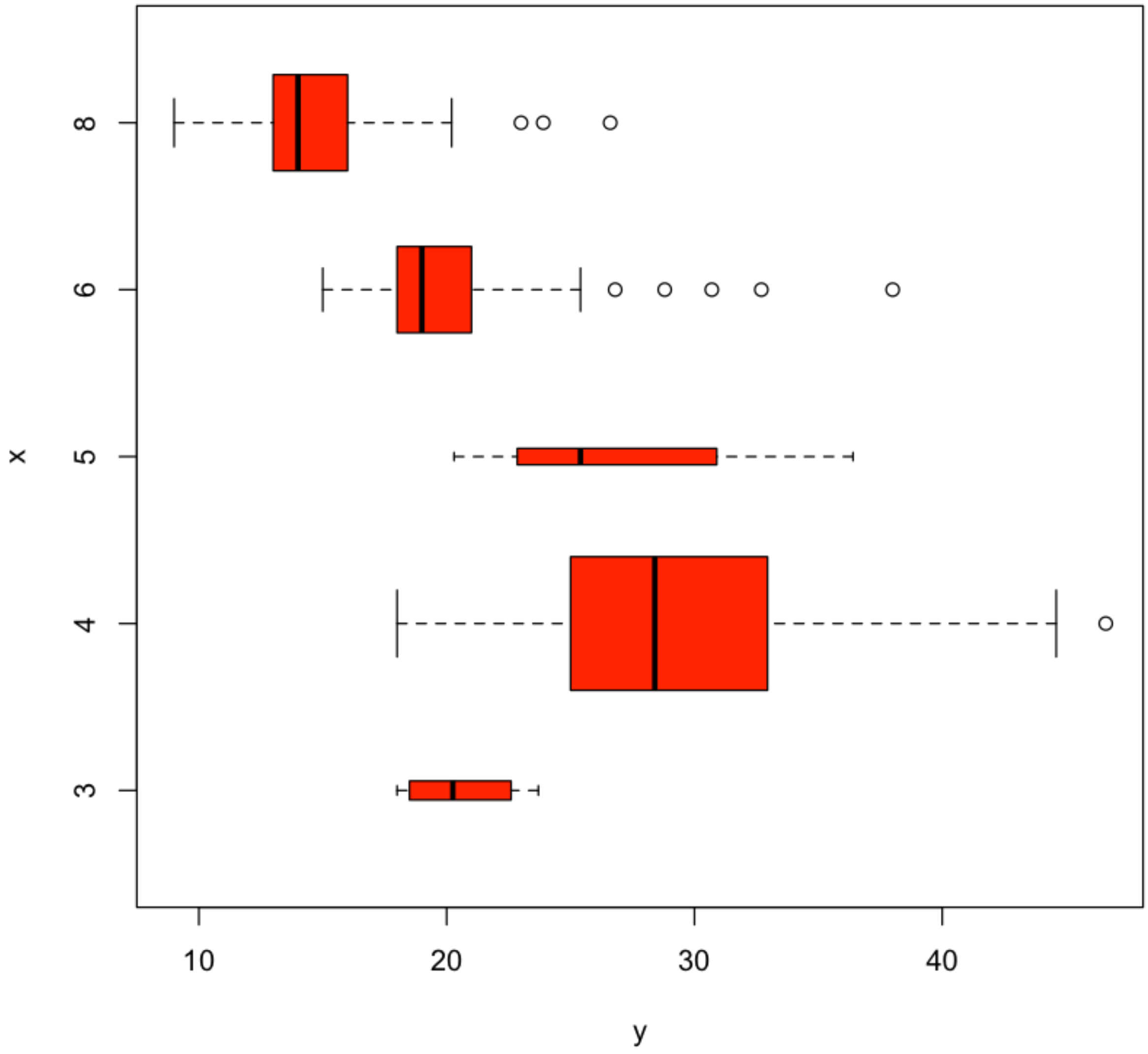
In [192]:

```
plot(cylinders, mpg, col = "red", varwidth = T)
```



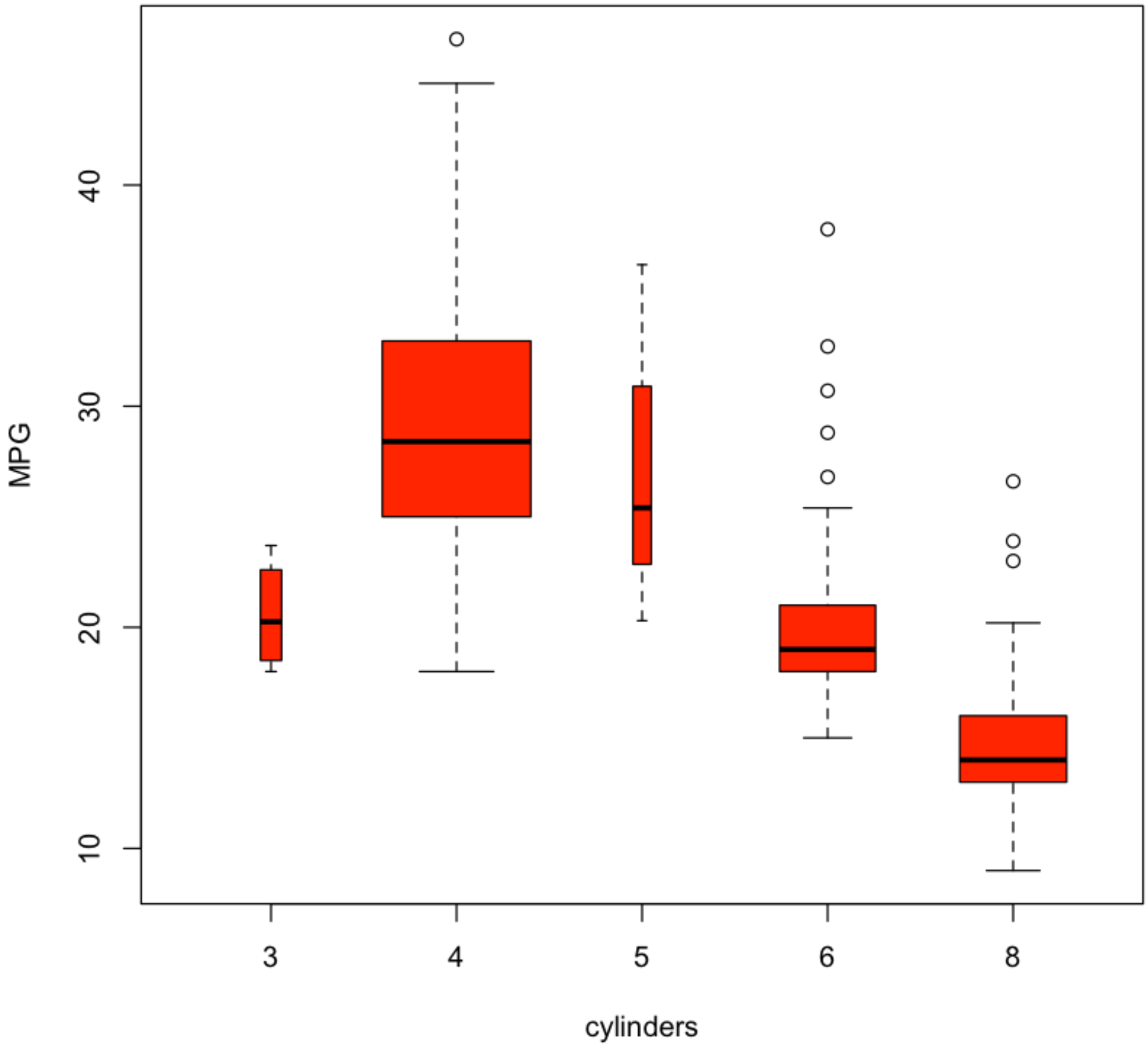
In [193]:

```
plot(cylinders, mpg, col = "red", varwidth = T, horizontal = T)
```

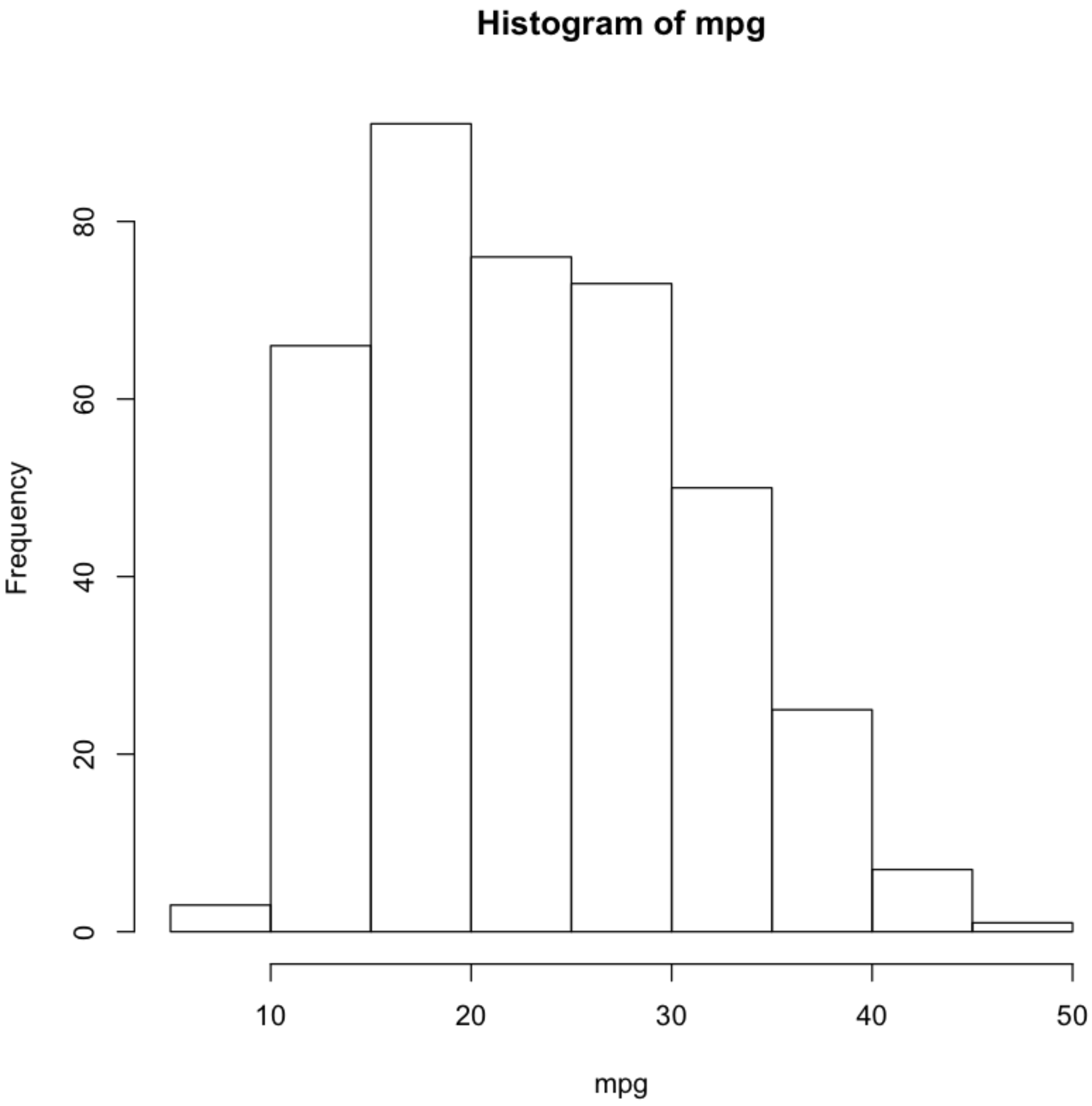


In [194]:

```
plot(cylinders, mpg, col = "red", varwidth = T, xlab = "cylinders"
```

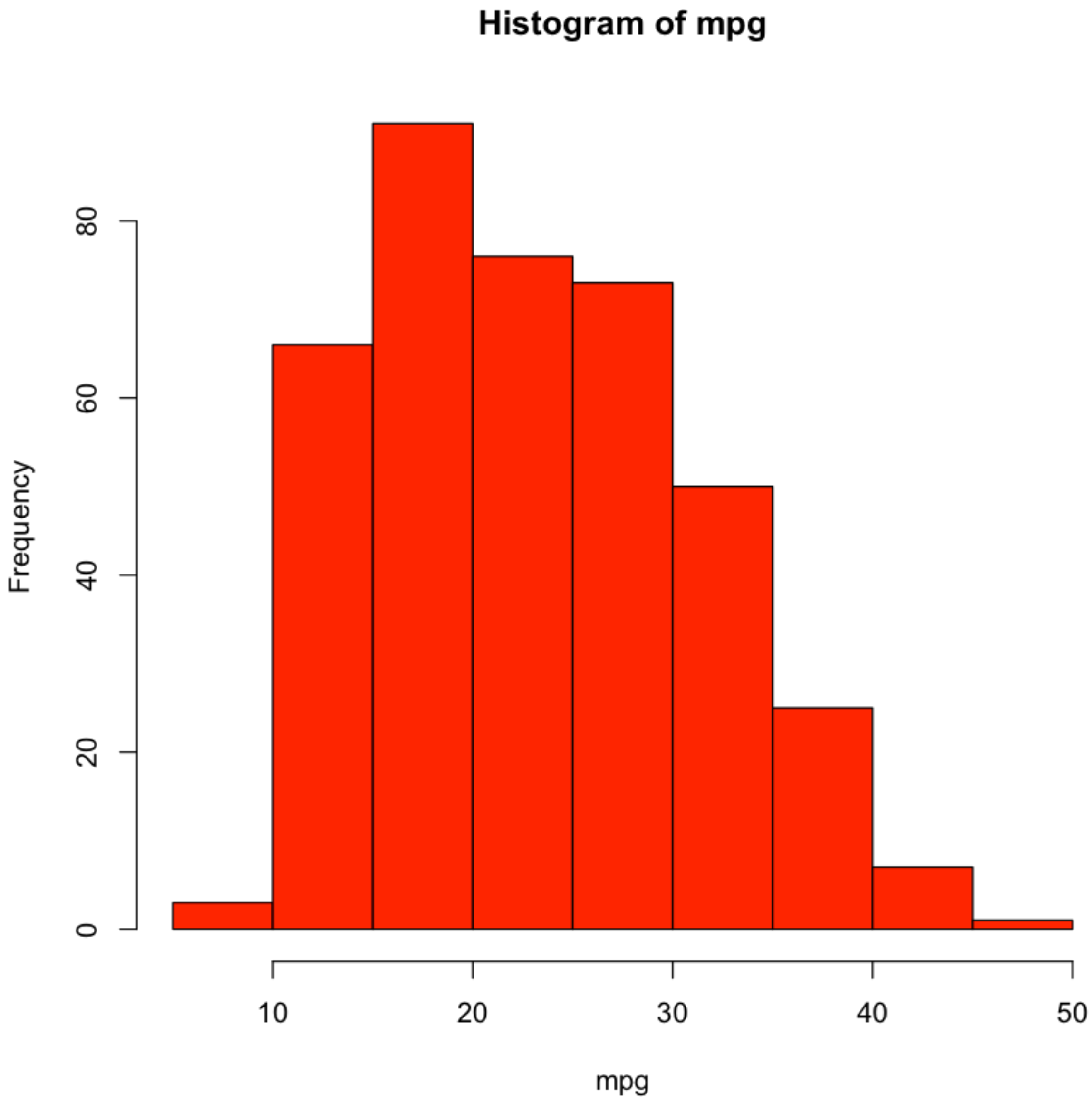


In [195]:
```
hist(mpg)
```



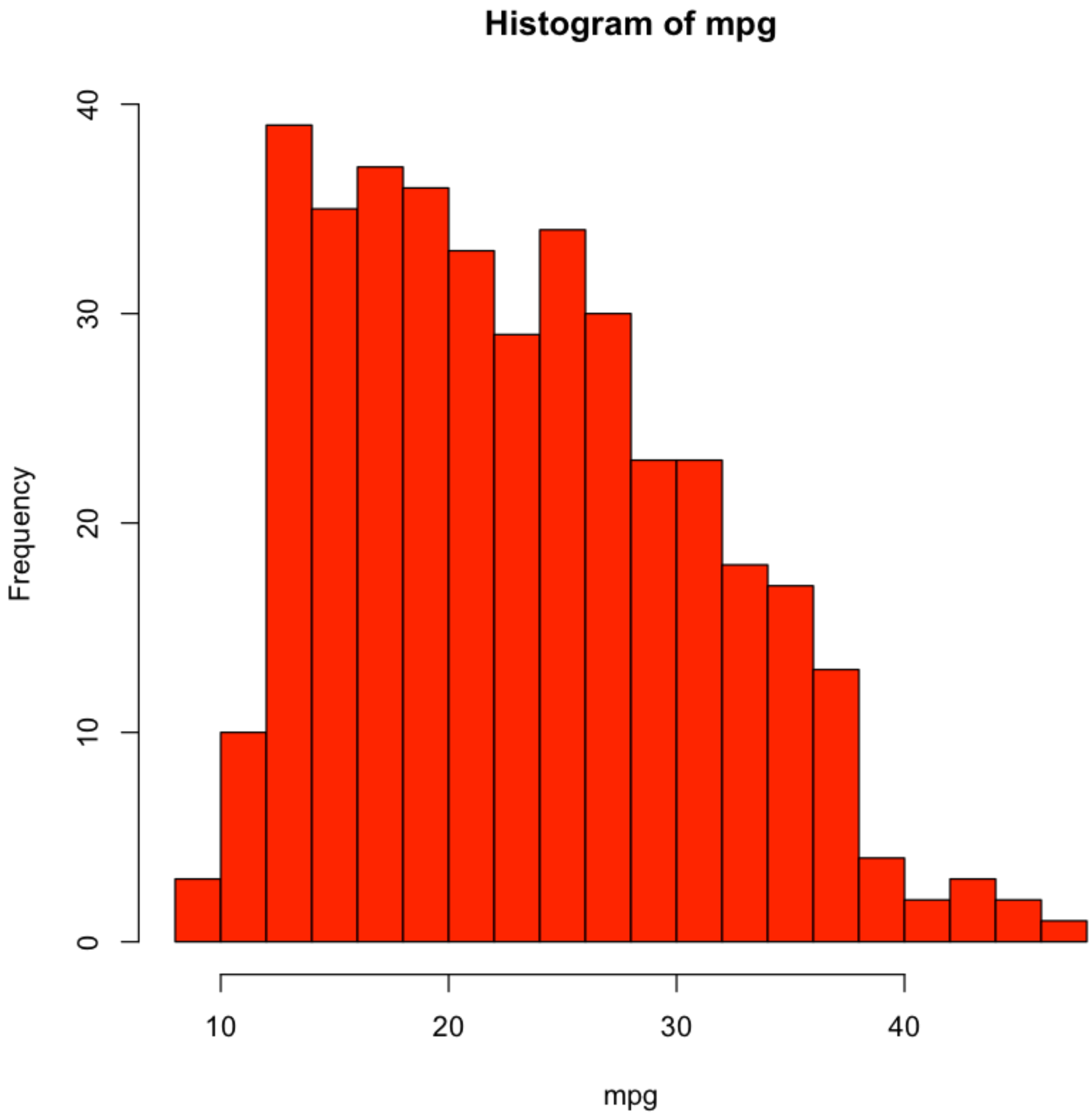
In [196]:

```
hist(mpg, col = 2)
```



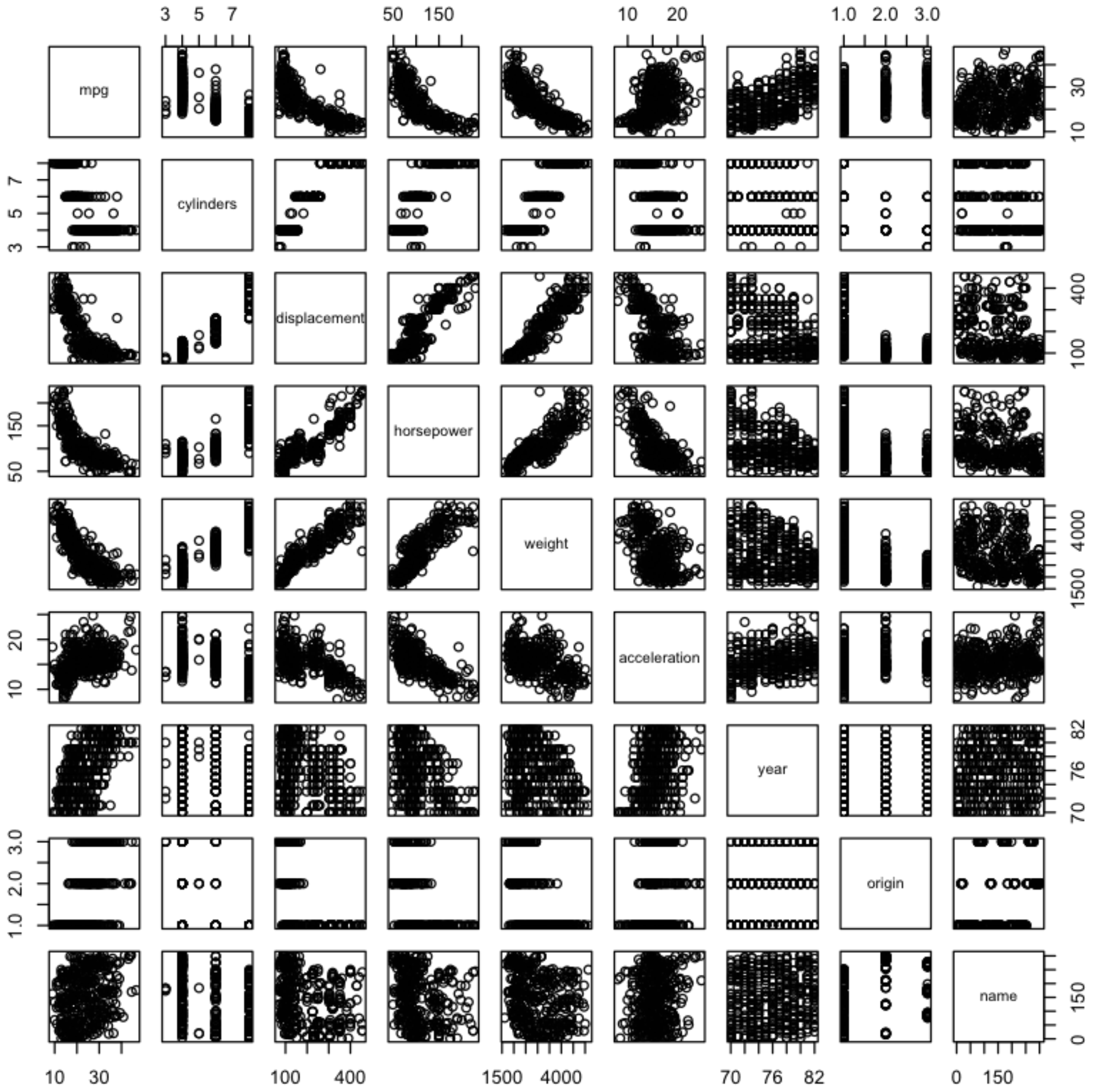
In [197]:

```
hist(mpg, col = 2, breaks = 15)
```

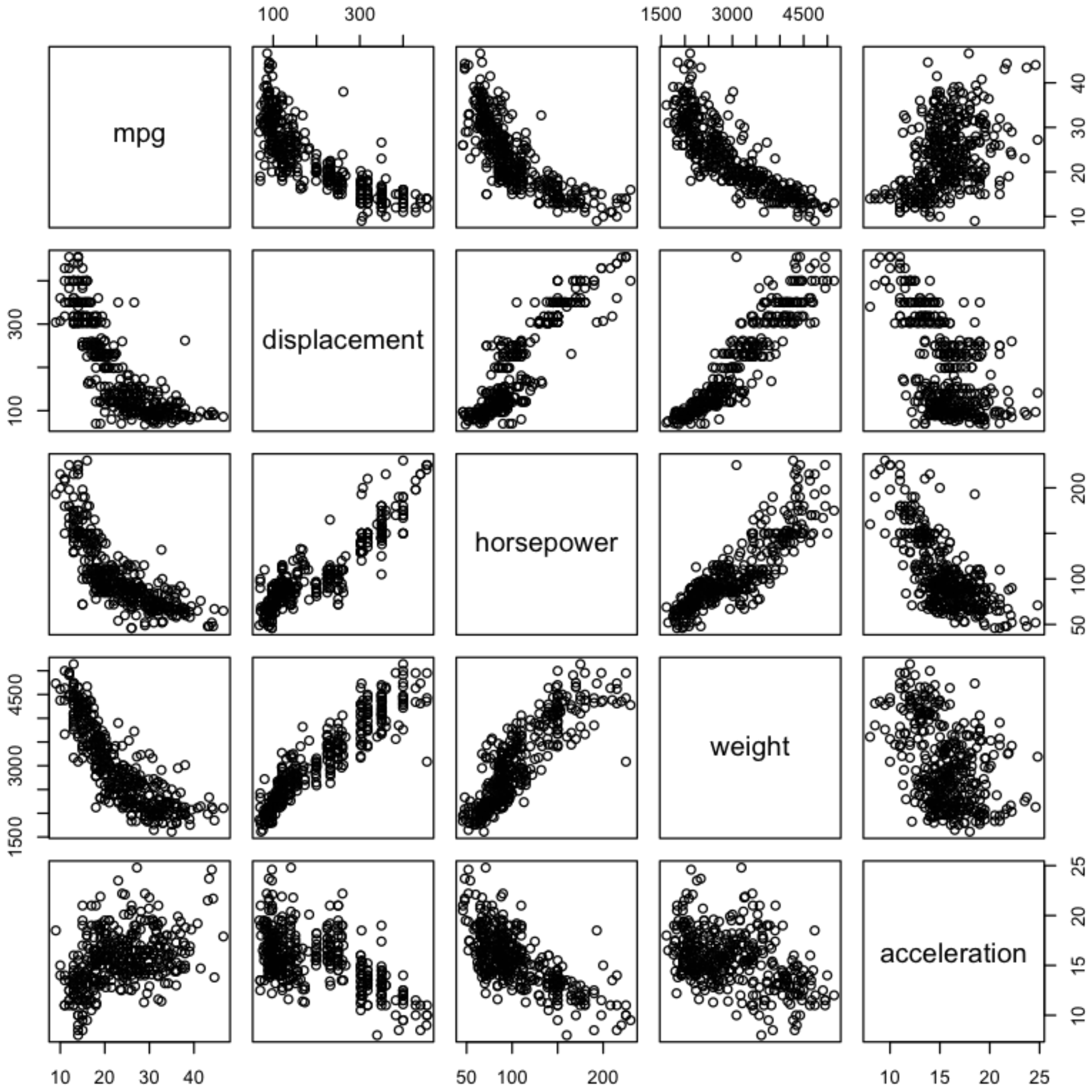


In [198]:

```
pairs(Auto)
```

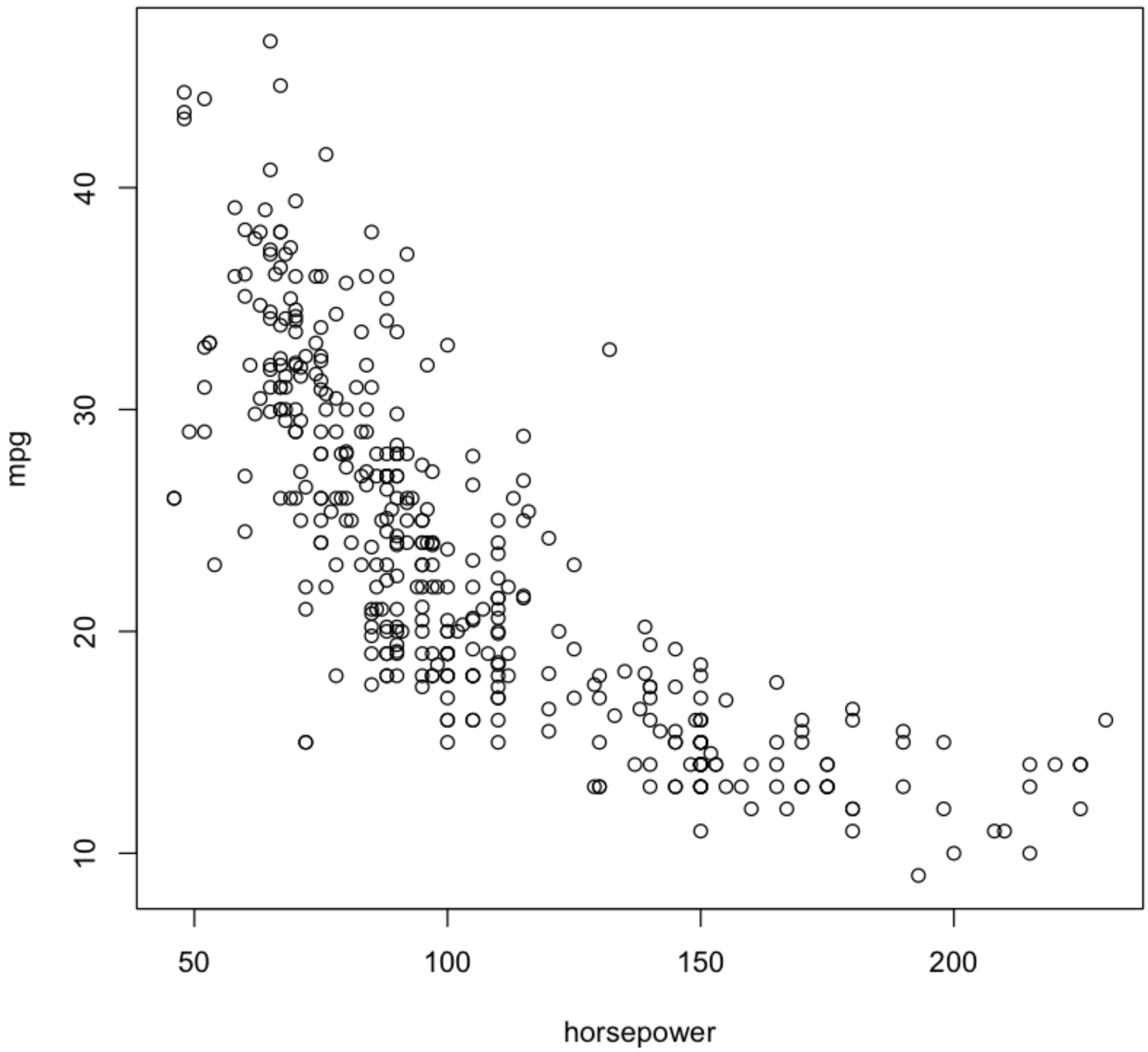


In [199]:
```
pairs(~mpg + displacement + horsepower + weight + acceleration, Au
```



In [200]:

```
plot(horsepower, mpg)
identify(horsepower, mpg, name)
#identify() provides a useful interactive method for identifying t
#value for a particular variable for points on a plot.
```



In [201]:

```
summary(Auto)
```

```
      mpg          cylinders      displacement     horsepower    
weight
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   
:1613
 1st Qu.:17.00   1st Qu.:4.000   1st Qu.:105.0   1st Qu.: 75.0   1st 
Qu.:2225
 Median :22.75   Median :4.000   Median :151.0   Median : 93.5   Medi
an :2804
 Mean   :23.45   Mean   :5.472   Mean   :194.4   Mean   :104.5   Mean   
:2978
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:275.8   3rd Qu.:126.0   3rd 
Qu.:3615
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   
:5140

  acceleration        year           origin                      name
 Min.   : 8.00   Min.   :70.00   Min.   :1.000   amc matador       :
5
 1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   ford pinto        :
5
 Median :15.50   Median :76.00   Median :1.000   toyota corolla    :
5
 Mean   :15.54   Mean   :75.98   Mean   :1.577   amc gremlin       :
4
 3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000   amc hornet        :
4
 Max.   :24.80   Max.   :82.00   Max.   :3.000   chevrolet chevette:
4
                                                 (Other)           :3
65
```

In [202]:

```
summary(mpg)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   9.00   17.00   22.75   23.45   29.00   46.60 
```