ISLR Lab 2.3

August 29, 2025

ISLR Lab 2.3

[1]:
```
x <- c(1,3,2,5)
print(x)
```

```
[1] 1 3 2 5
```

[2]:
```
x = c(1,6,2)
print(x)
y = c(1,4,3)
```

```
[1] 1 6 2
```

[3]:
```
print(length(x))
print(length(y))
print(x+y)
```

```
[1] 3
[1] 3
[1]  2 10  5
```

[4]:
```
print(ls())
rm(x,y)
print(ls())
```

```
[1] "x" "y"
character(0)
```

[5]:
```
rm(list=ls())
```

[ ]:
```
?matrix
```

[7]:
```
x=matrix(data=c(1,2,3,4), nrow=2, ncol=2)
print(x)
```

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

[8]:
```
x=matrix(c(1,2,3,4) ,2,2)
```

[9]:
```
print(matrix (c(1,2,3,4) ,2,2,byrow=TRUE))
```

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

[10]:
```
print(sqrt(x))
```

```
         [,1]     [,2]
[1,] 1.000000 1.732051
[2,] 1.414214 2.000000
```

[11]:
```
print(x^2)
```

```
     [,1] [,2]
[1,]    1    9
[2,]    4   16
```

[12]:
```
x=rnorm(50)
y=x+rnorm(50,mean=50,sd=.1)
print(cor(x,y))
```

```
[1] 0.996258
```

[13]:
```
set.seed(1303)
print(rnorm(50))
```

```
 [1] -1.1439763145  1.3421293656  2.1853904757  0.5363925179  0.0631929665
 [6]  0.5022344825 -0.0004167247  0.5658198405 -0.5725226890 -1.1102250073
[11] -0.0486871234 -0.6956562176  0.8289174803  0.2066528551 -0.2356745091
[16] -0.5563104914 -0.3647543571  0.8623550343 -0.6307715354  0.3136021252
[21] -0.9314953177  0.8238676185  0.5233707021  0.7069214120  0.4202043256
[26] -0.2690521547 -1.5103172999 -0.6902124766 -0.1434719524 -1.0135274099
[31]  1.5732737361  0.0127465055  0.8726470499  0.4220661905 -0.0188157917
[36]  2.6157489689 -0.6931401748 -0.2663217810 -0.7206364412  1.3677342065
[41]  0.2640073322  0.6321868074 -1.3306509858  0.0268888182  1.0406363208
[46]  1.3120237985 -0.0300020767 -0.2500257125  0.0234144857  1.6598706557
```

[14]:
```
set.seed(3)
y=rnorm(100)
print(mean(y))
```

```
[1] 0.01103557
```

[15]:
```
print(var(y))
```

```
[1] 0.7328675
```

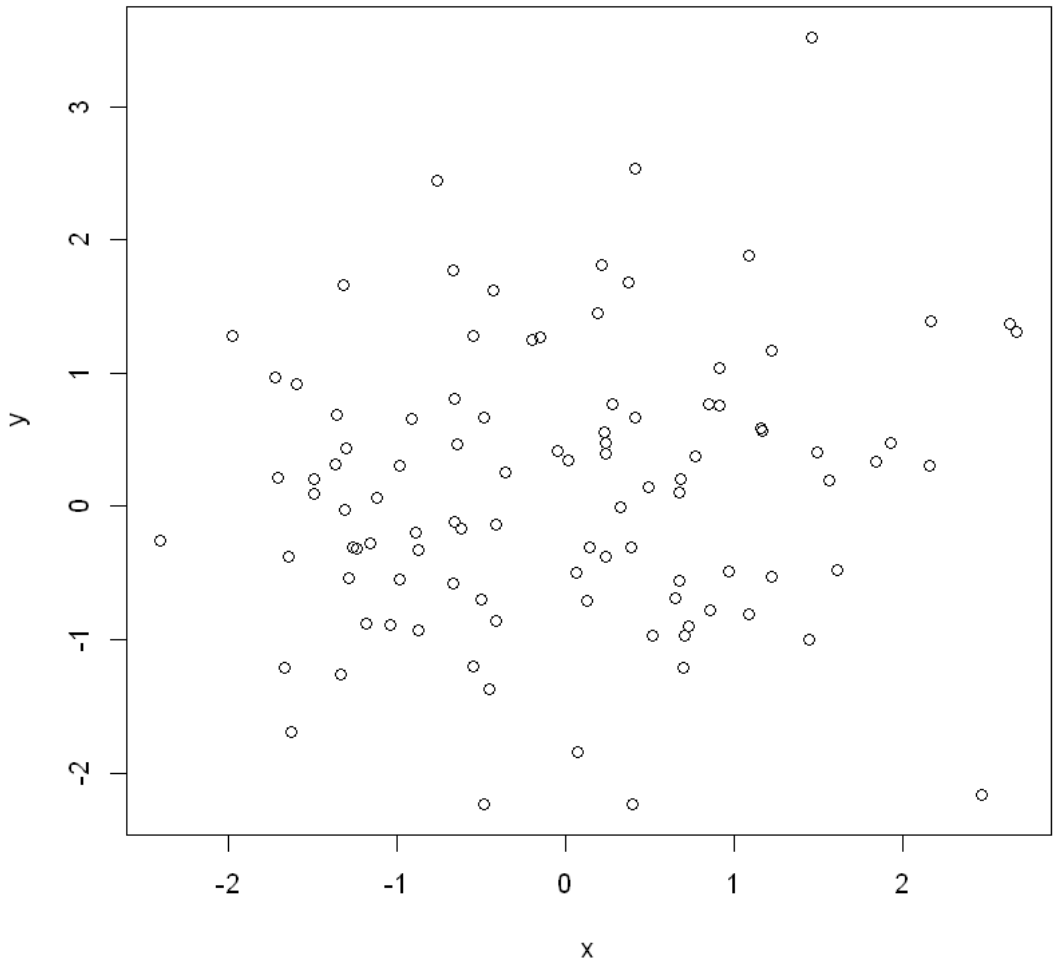
[16]:
```
print(sqrt(var(y)))
```

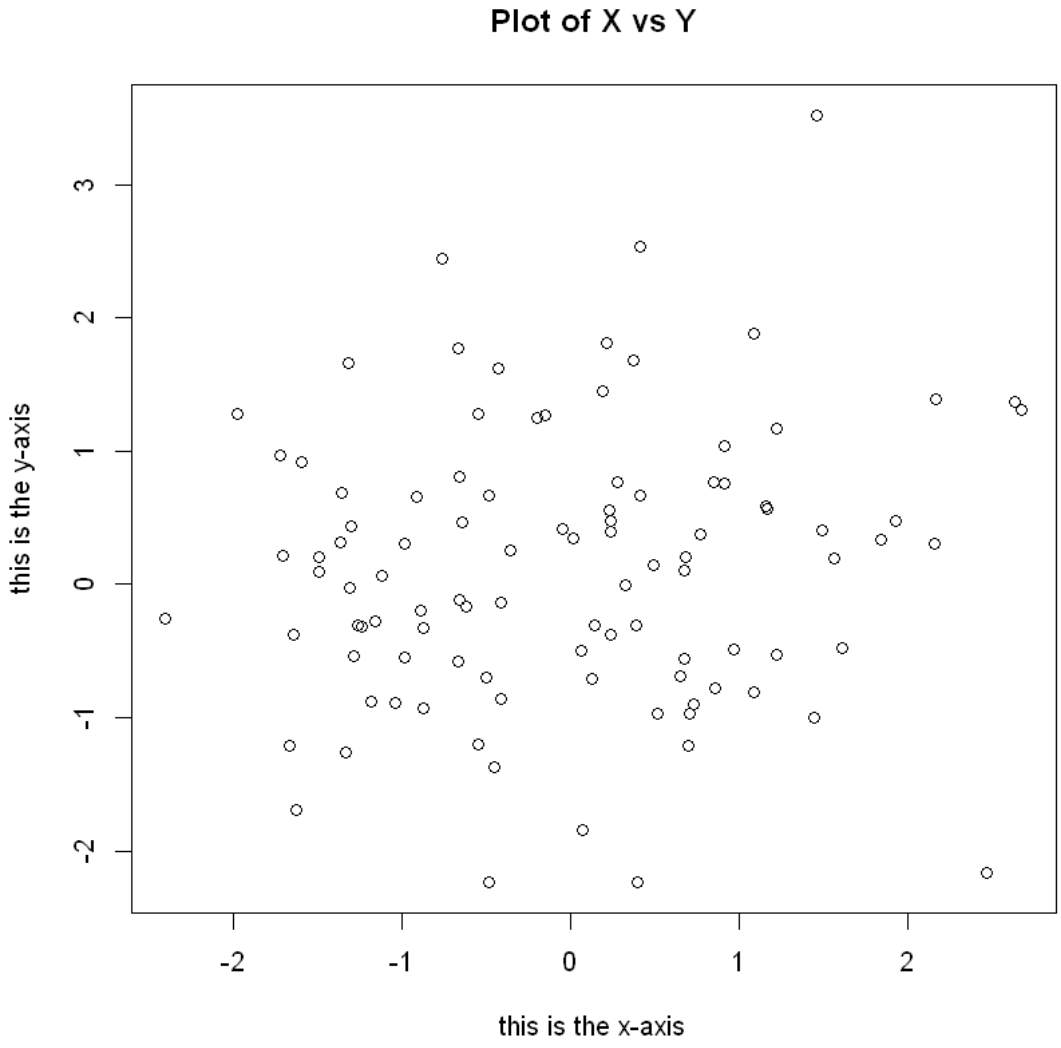
```
[1] 0.8560768
```

[17]:
```
print(sd(y))
```

```
[1] 0.8560768
```

[18]:
```
x=rnorm(100)
y=rnorm(100)
plot(x,y)
plot(x,y,xlab="this is the x-axis",ylab="this is the y-axis", main="Plot of X␣
↪vs Y")
```





[19]:
```
pdf("Figure.pdf")
plot(x,y,col="green")
dev.off()
```

agg__record__61171926: 2

[20]:
```
x=seq(1,10)
print(x)
```

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

[21]:
```
 x=1:10
print(x)
```

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

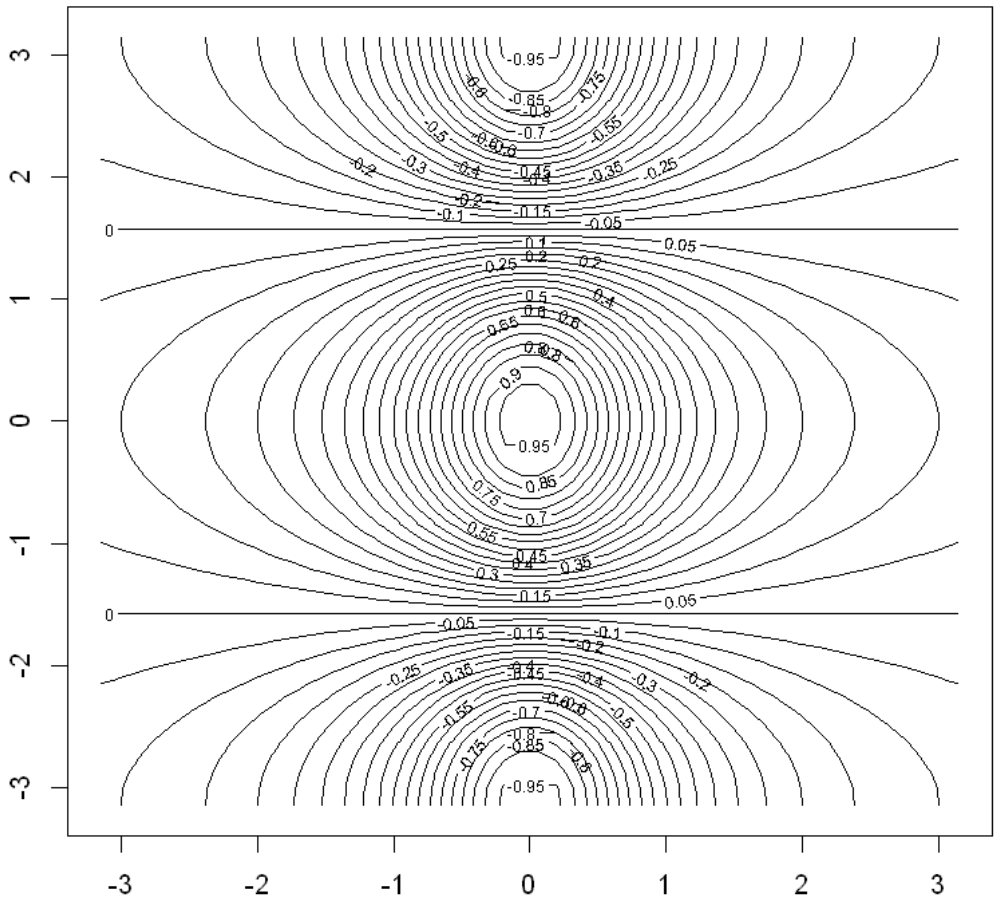
[22]:
```
x=seq(-pi,pi,length =50)
print(x)
```

```
 [1] -3.14159265 -3.01336438 -2.88513611 -2.75690784 -2.62867957 -2.50045130
 [7] -2.37222302 -2.24399475 -2.11576648 -1.98753821 -1.85930994 -1.73108167
[13] -1.60285339 -1.47462512 -1.34639685 -1.21816858 -1.08994031 -0.96171204
[19] -0.83348377 -0.70525549 -0.57702722 -0.44879895 -0.32057068 -0.19234241
[25] -0.06411414  0.06411414  0.19234241  0.32057068  0.44879895  0.57702722
[31]  0.70525549  0.83348377  0.96171204  1.08994031  1.21816858  1.34639685
[37]  1.47462512  1.60285339  1.73108167  1.85930994  1.98753821  2.11576648
[43]  2.24399475  2.37222302  2.50045130  2.62867957  2.75690784  2.88513611
[49]  3.01336438  3.14159265
```

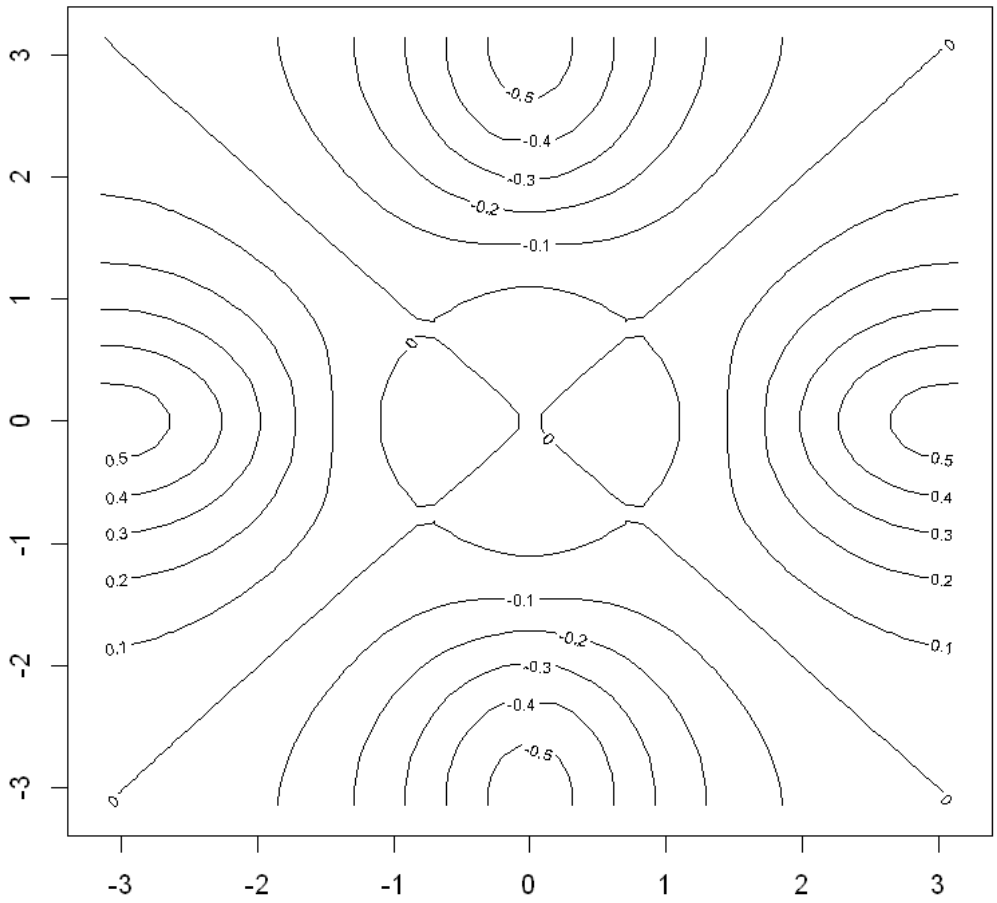
[23]:
```
y = x
f=outer(x,y,function (x,y)cos(y)/(1+x^2))
```

[ ]:
```
?contour
```

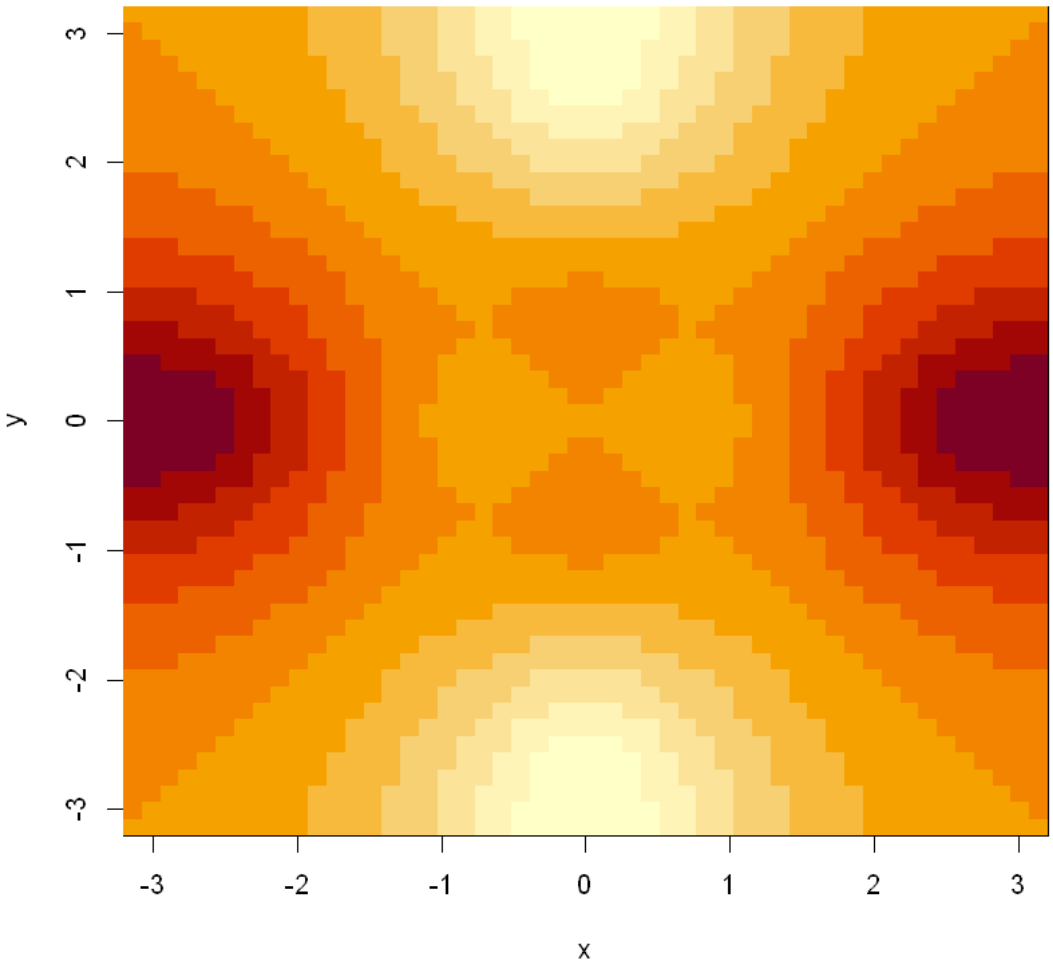
[25]:
```
contour(x,y,f)
contour(x,y,f,nlevels=45,add=T)
```



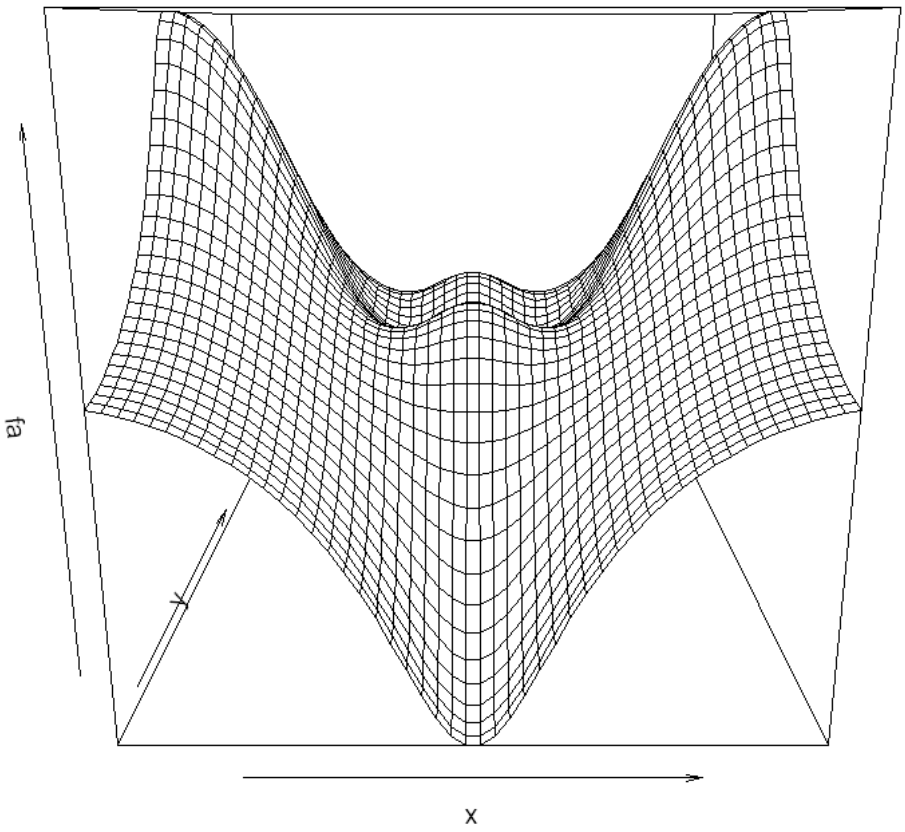
[26]:
```
fa=(f-t(f))/2
contour (x,y,fa,nlevels =15)
```

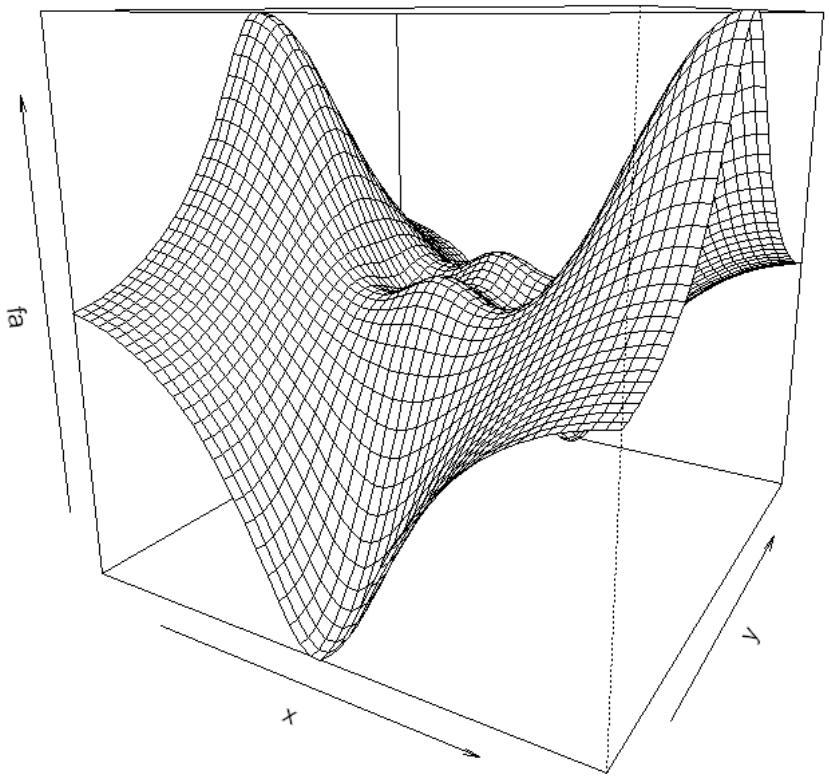


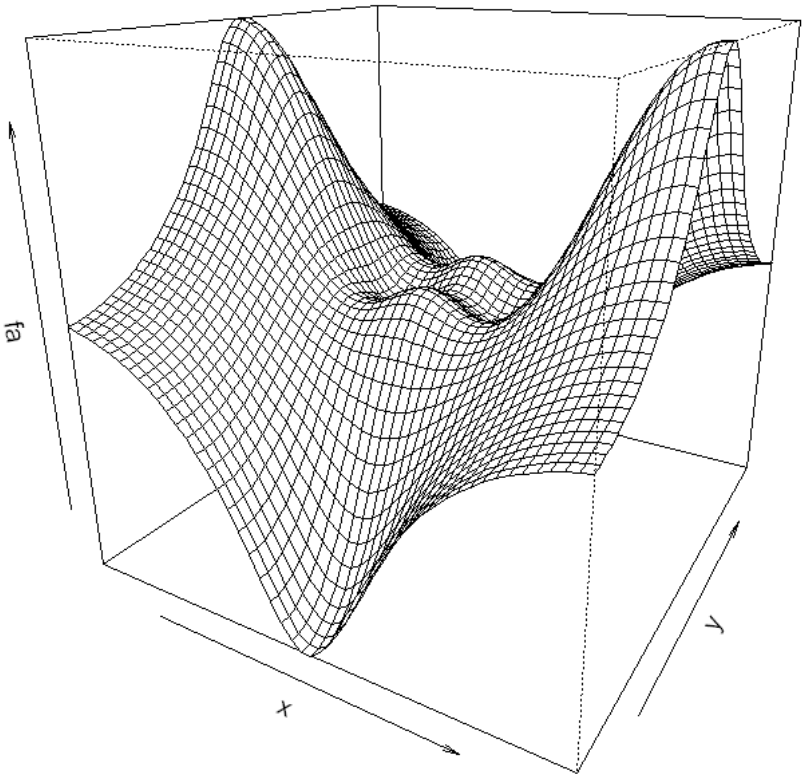
[27]:
```
image(x,y,fa)
```

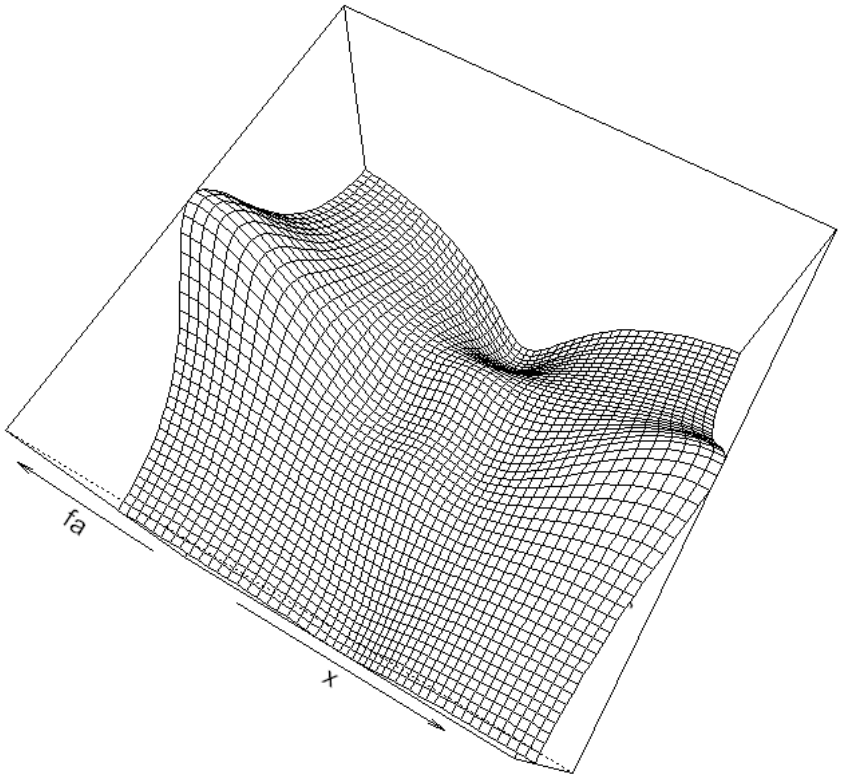


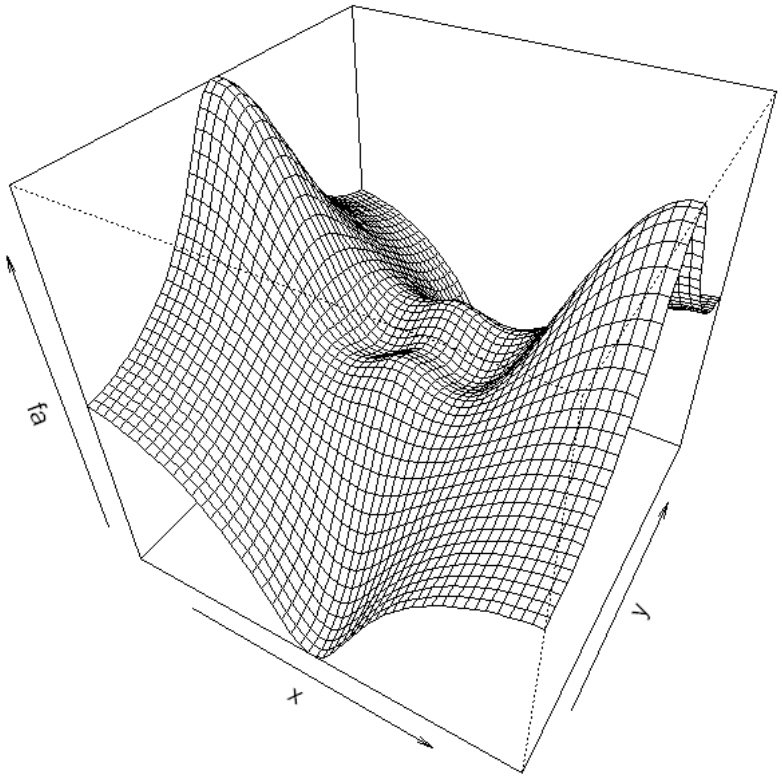
[28]:
```
persp(x,y,fa)
persp(x,y,fa,theta =30)
persp(x,y,fa,theta =30,phi =20)
persp(x,y,fa,theta =30,phi =70)
persp(x,y,fa,theta =30,phi =40)
```


fa
y
x


fa
x
y


fa
x
y


fa
x



[29]:
```
A=matrix (1:16,4,4)
print(A)
```

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

[30]:
```
print(A[2,3])
```

```
[1] 10
```

[31]:
```
print(A[c(1,3),c(2,4)])
```

```
     [,1] [,2]
[1,]    5   13
[2,]    7   15
```

[32]:
```
print(A[1:3,2:4])
```

```
     [,1] [,2] [,3]
[1,]    5    9   13
[2,]    6   10   14
[3,]    7   11   15
```

[33]:
```
print(A[1:2,])
```

```
     [,1] [,2] [,3] [,4]
[1,]    1    5    9   13
[2,]    2    6   10   14
```

[34]:
```
print(A[,1:2])
```

```
     [,1] [,2]
[1,]    1    5
[2,]    2    6
[3,]    3    7
[4,]    4    8
```

[35]:
```
print(A[1,])
```

```
[1]  1  5  9 13
```

[36]:
```
print(A[-c(1,3) ,])
```

```
     [,1] [,2] [,3] [,4]
[1,]    2    6   10   14
[2,]    4    8   12   16
```

[37]:
```
print(dim(A))
```

```
[1] 4 4
```

[38]:
```
Auto=read.table("Auto.data")
fix(Auto)
```

[39]:
```
Auto=read.table("Auto.data",header=T,na.strings ="?")
fix(Auto)
```

[40]:
```
Auto=read.csv("Auto.csv",header=T,na.strings ="?")
fix(Auto)
print(dim(Auto))
```

```
[1] 397   9
```

[41]:
```
Auto=na.omit(Auto)
print(dim(Auto))
```

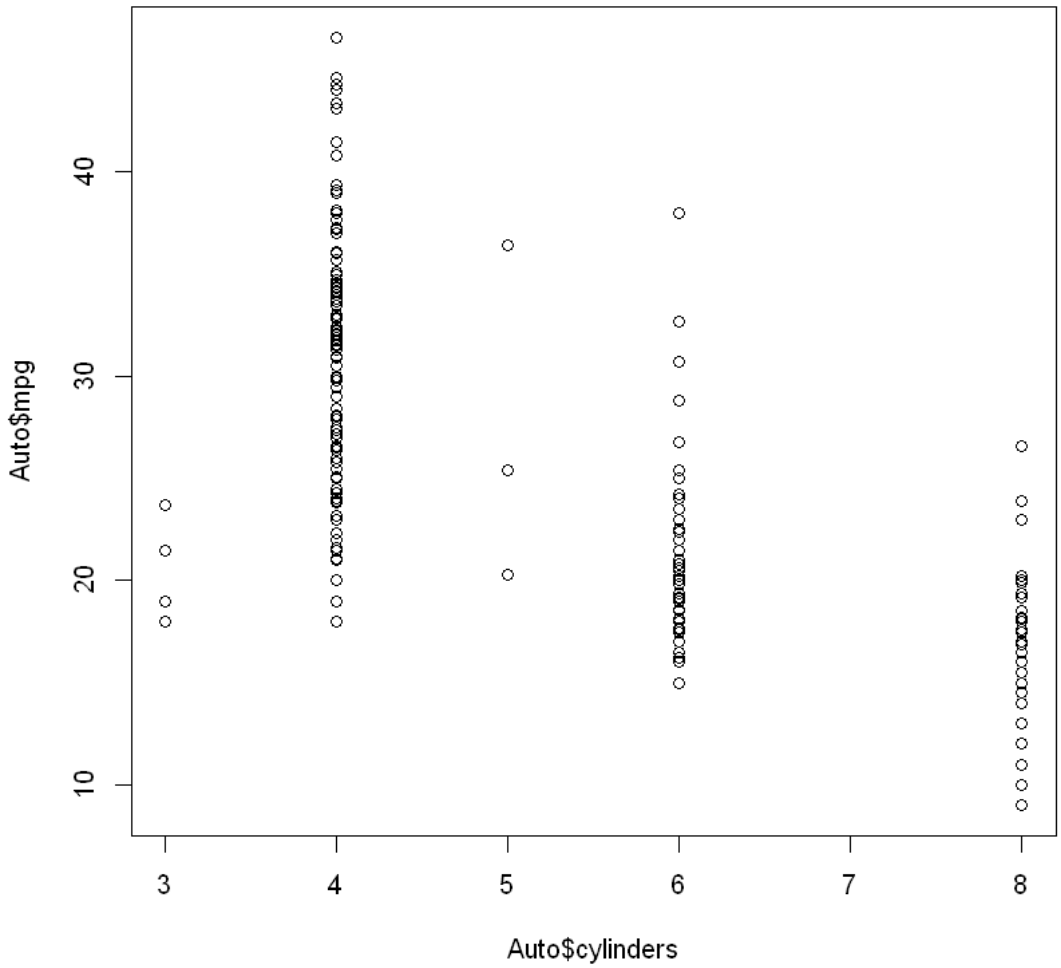
```
[1] 392   9
```

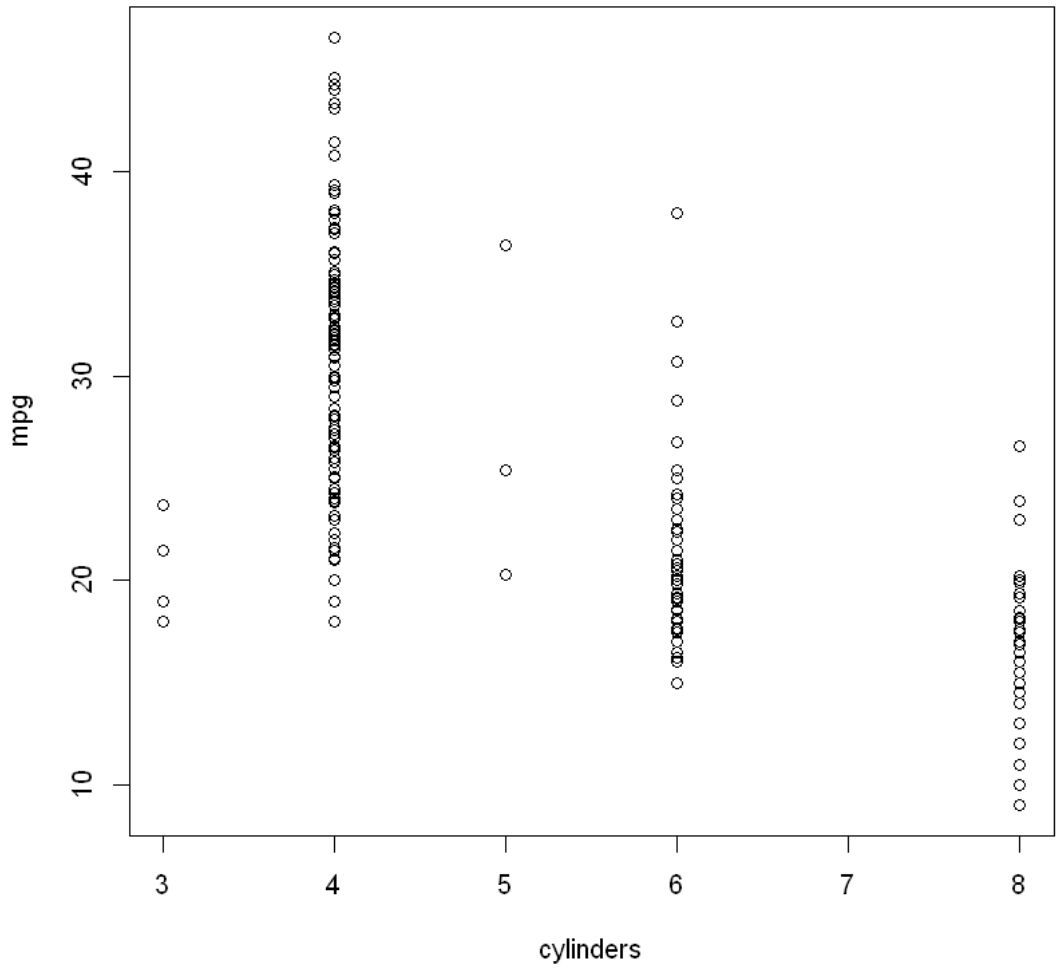
[42]:
```
plot(cylinders , mpg)
```

```
Error: object 'cylinders' not found
Traceback:

1. .handleSimpleError(function (cnd)
 . {
 .     watcher$capture_plot_and_output()
 .     cnd <- sanitize_call(cnd)
 .     watcher$push(cnd)
 .     switch(on_error, continue = invokeRestart("eval_continue"),
 .         stop = invokeRestart("eval_stop"), error = NULL)
 . }, "object 'cylinders' not found", base::quote(eval(expr, envir)))
```

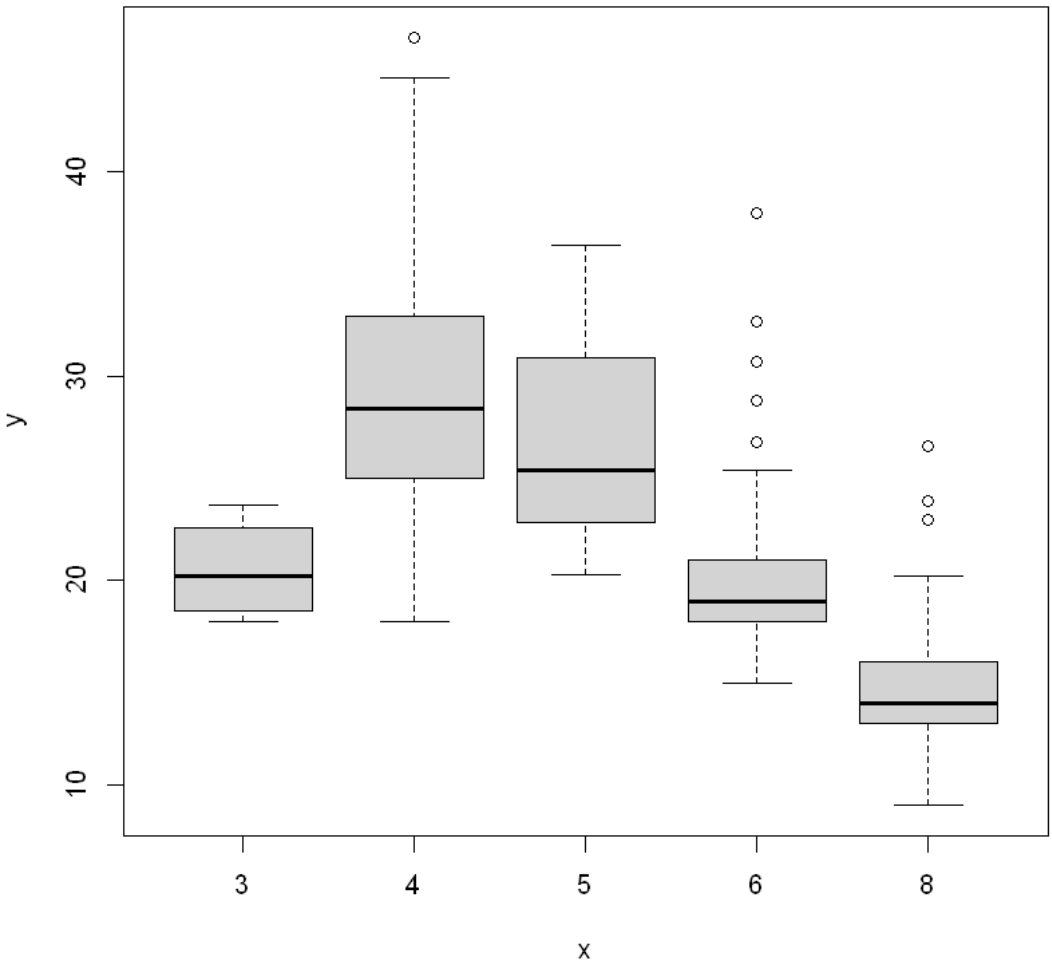
[43]:
```
plot(Auto$cylinders, Auto$mpg )
attach(Auto)
plot(cylinders, mpg )
```

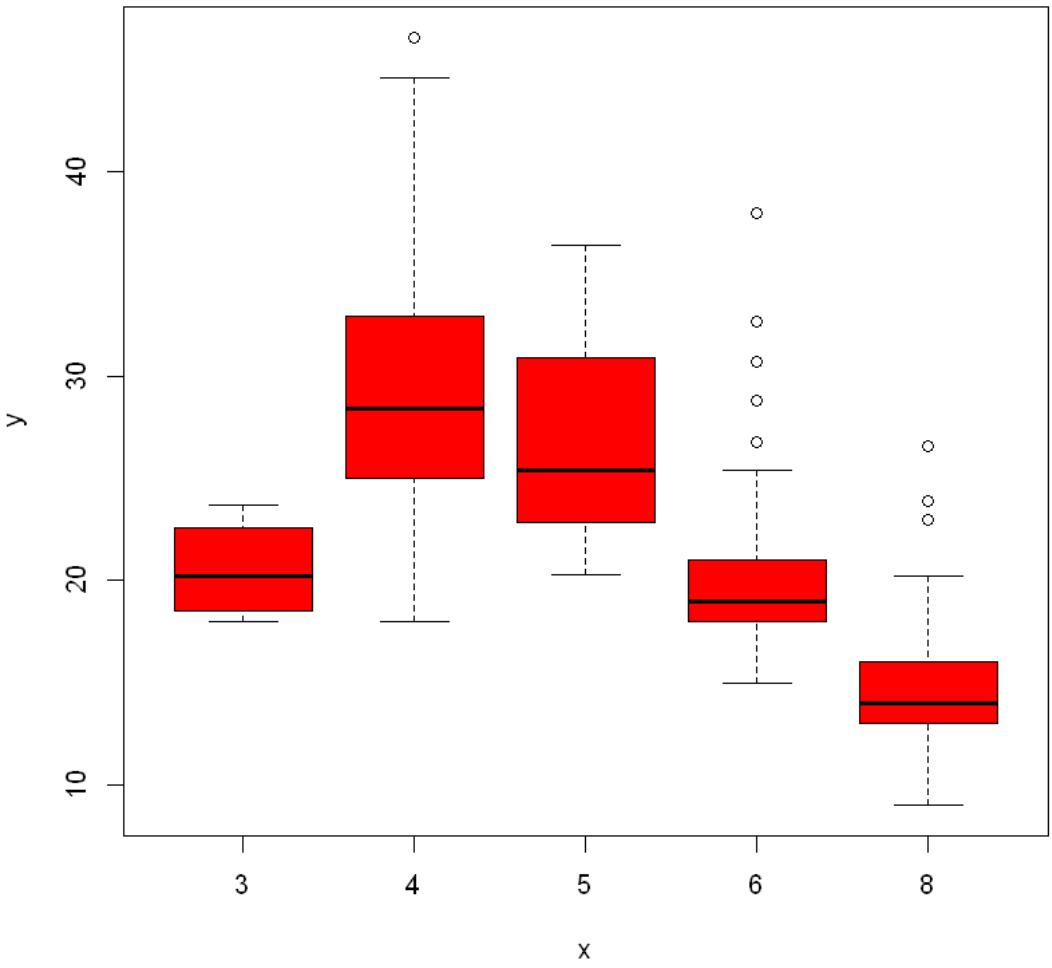

Auto$mpg
10
20
30
40
3
4
5
6
7
8
Auto$cylinders

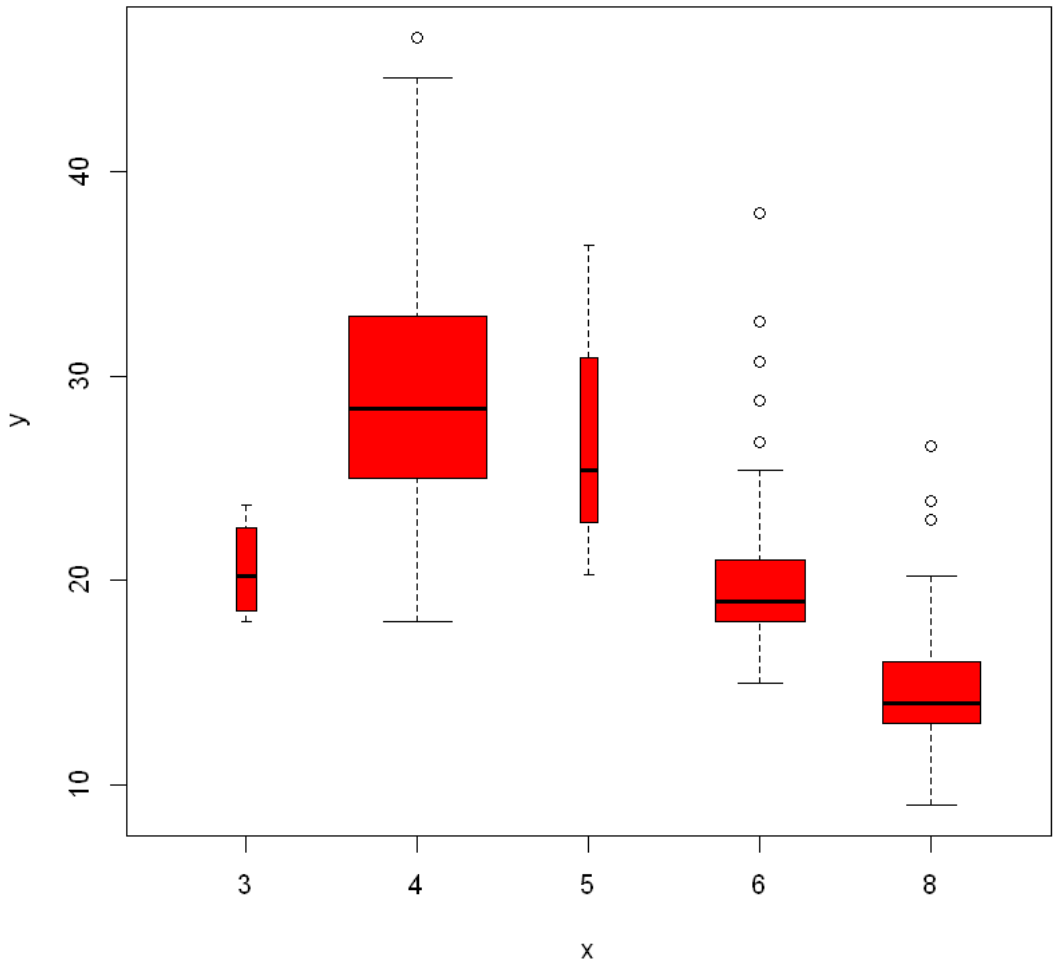


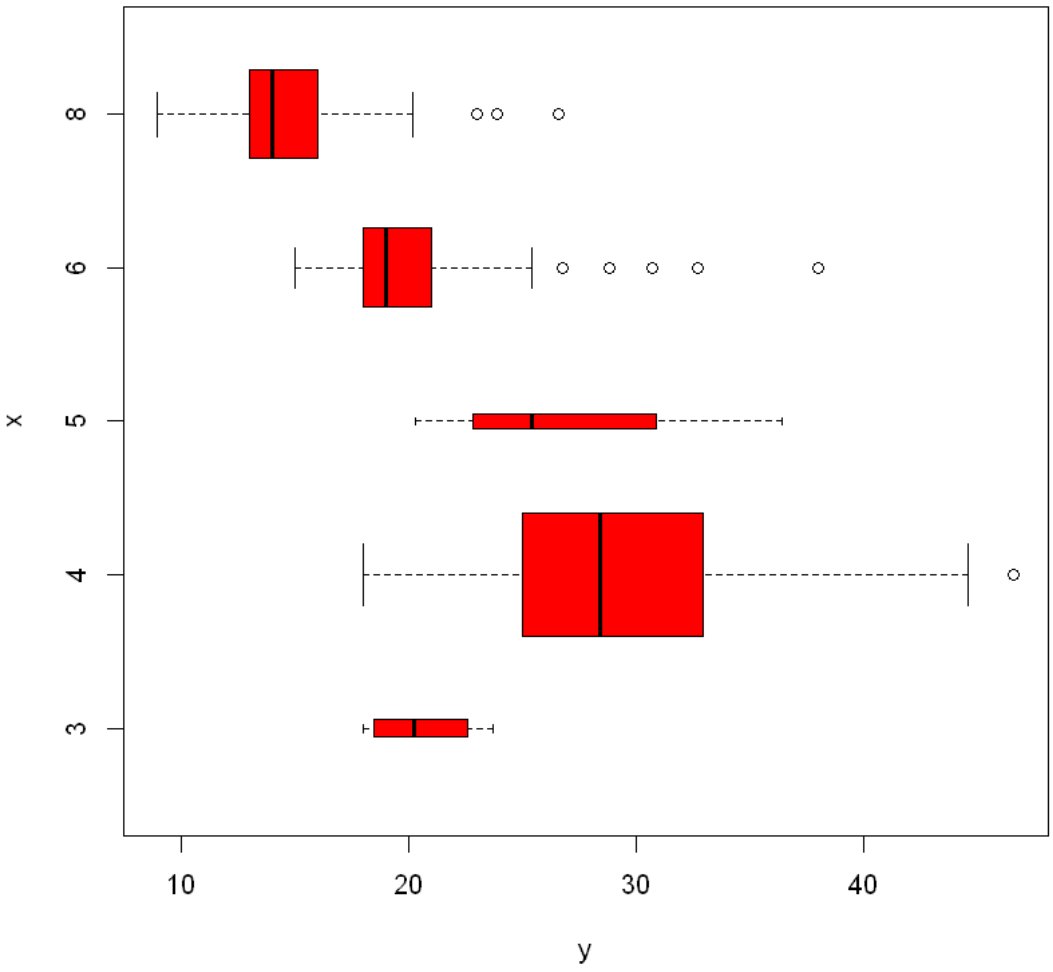
[44]:
```
cylinders =as.factor(cylinders )
```

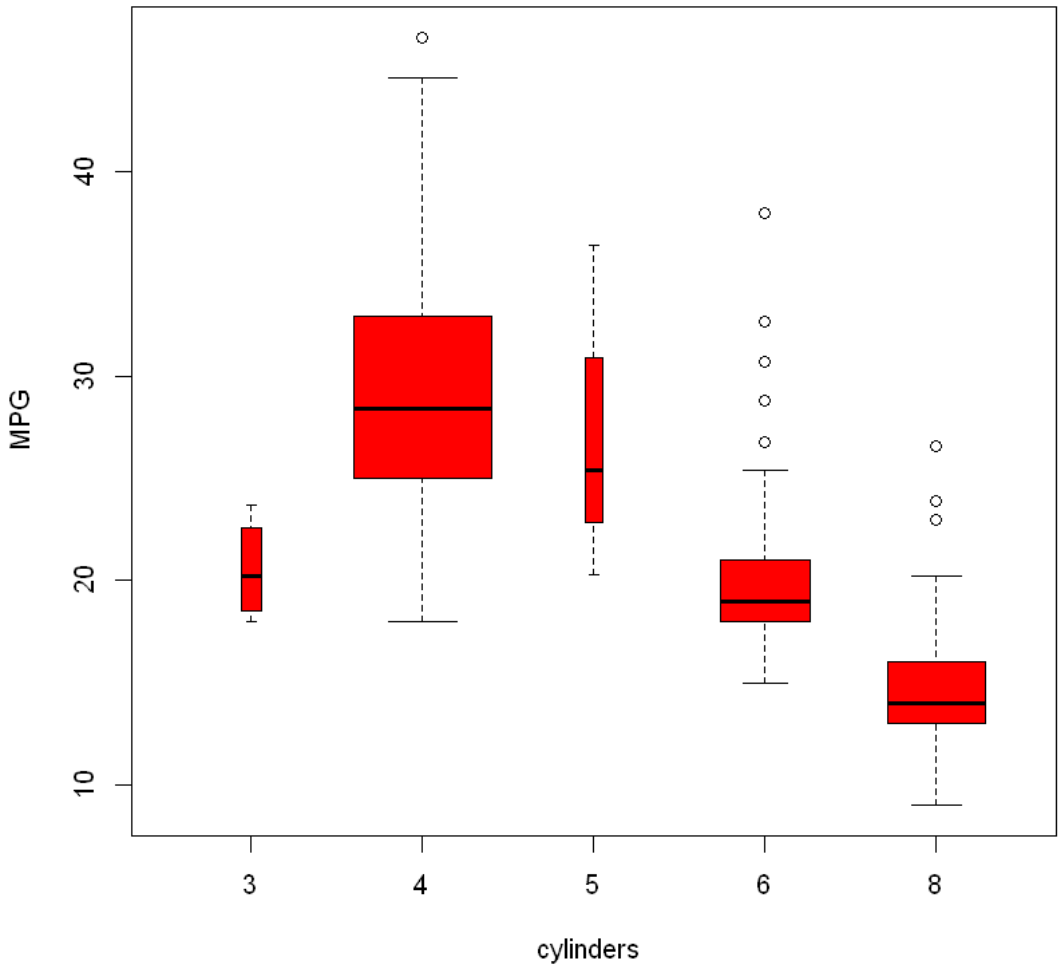
[45]:
```
plot(cylinders , mpg)
plot(cylinders , mpg , col ="red ")
plot(cylinders , mpg , col ="red", varwidth =T)
plot(cylinders , mpg , col ="red", varwidth =T,horizontal =T)
plot(cylinders , mpg , col ="red", varwidth =T, xlab="cylinders ",ylab="MPG")
```


y
10
20
30
40
3
4
5
6
8
x

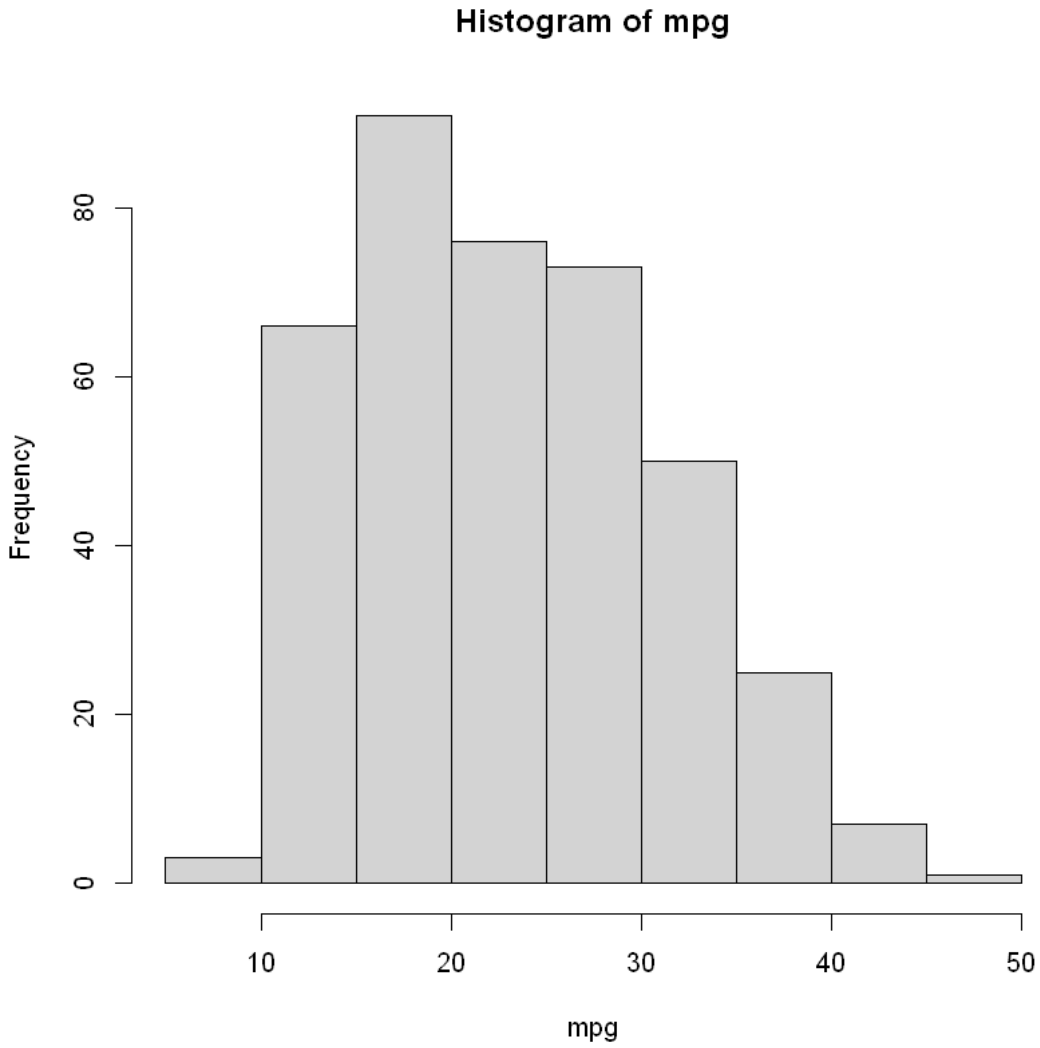

y
10
20
30
40
3
4
5
6
8
x

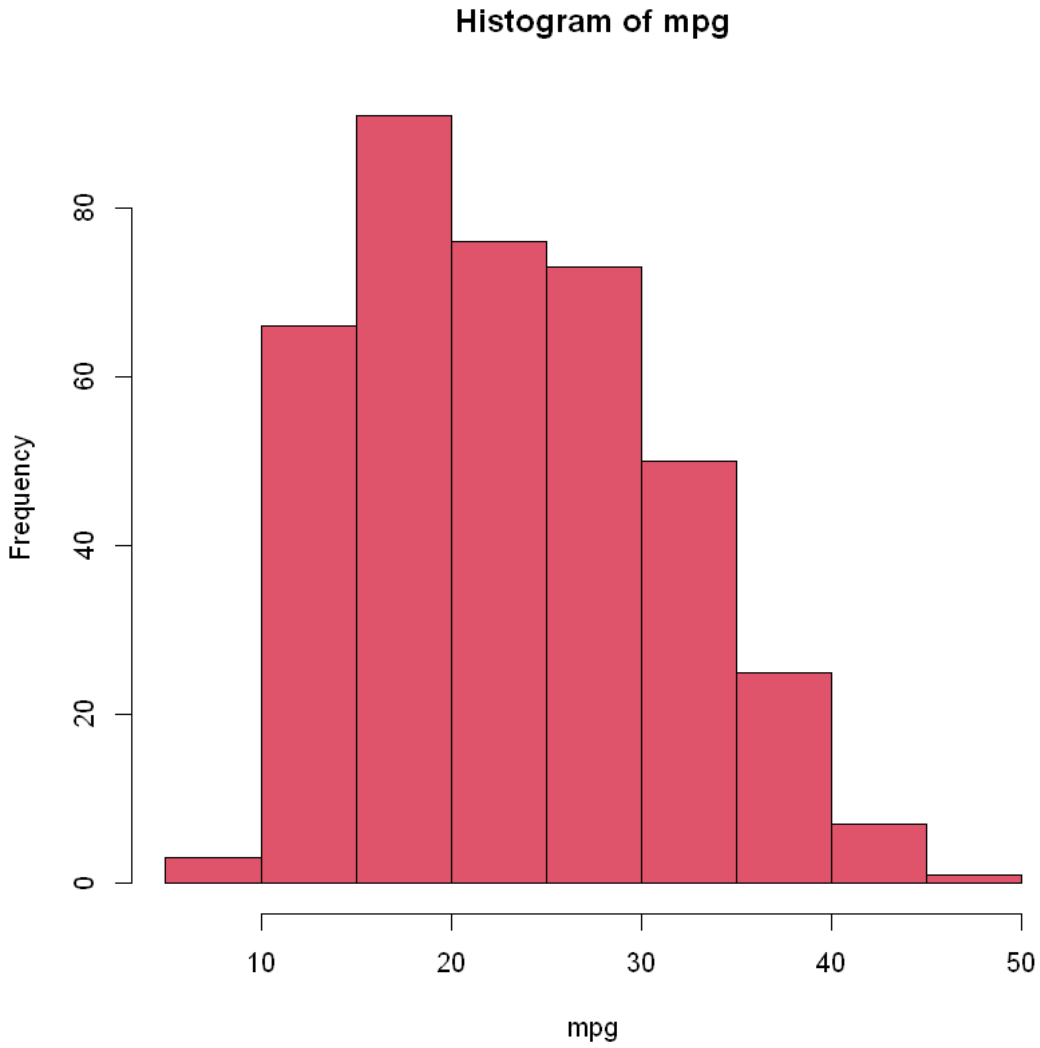

y
10
20
30
40
3
4
5
6
8
x

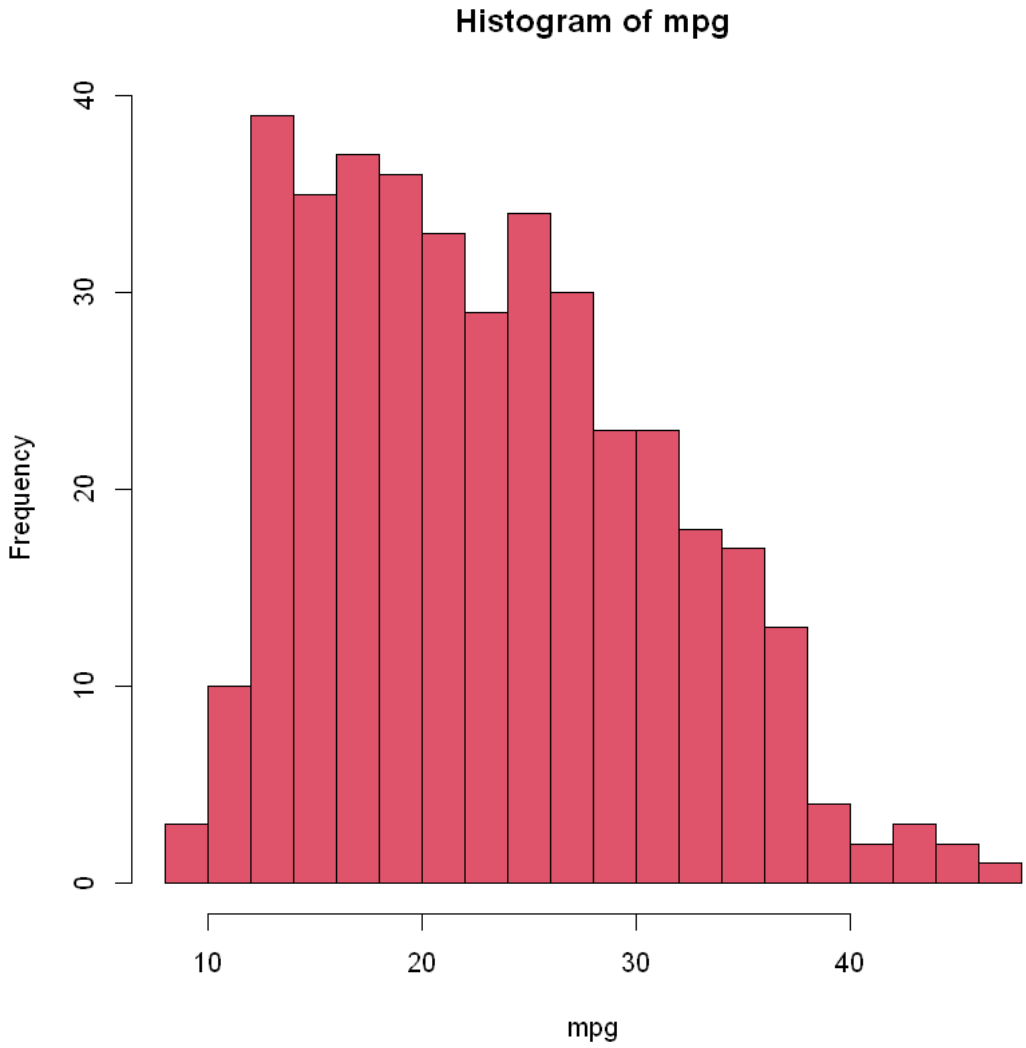

x
8
6
5
4
3
10
20
30
40
y



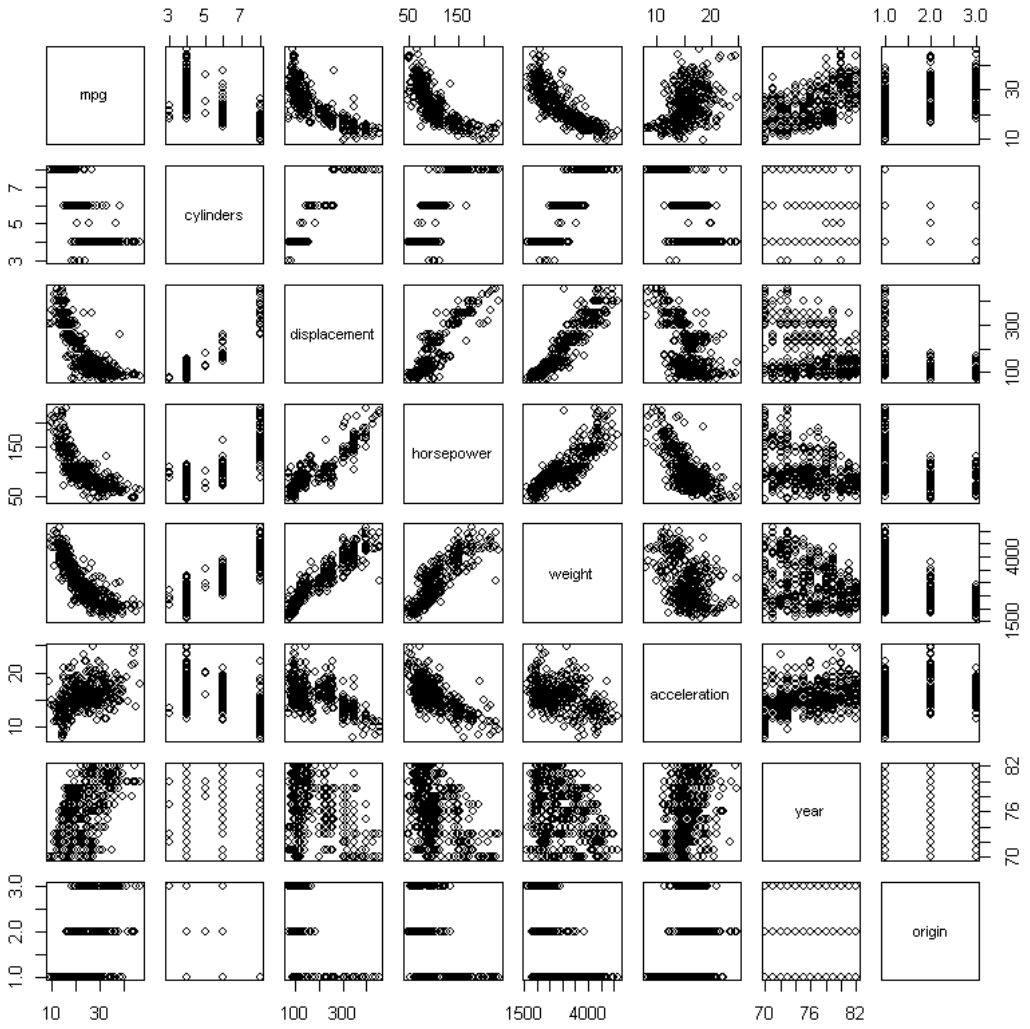
[46]:
```
hist(mpg)
hist(mpg ,col =2)
hist(mpg ,col=2, breaks =15)
```

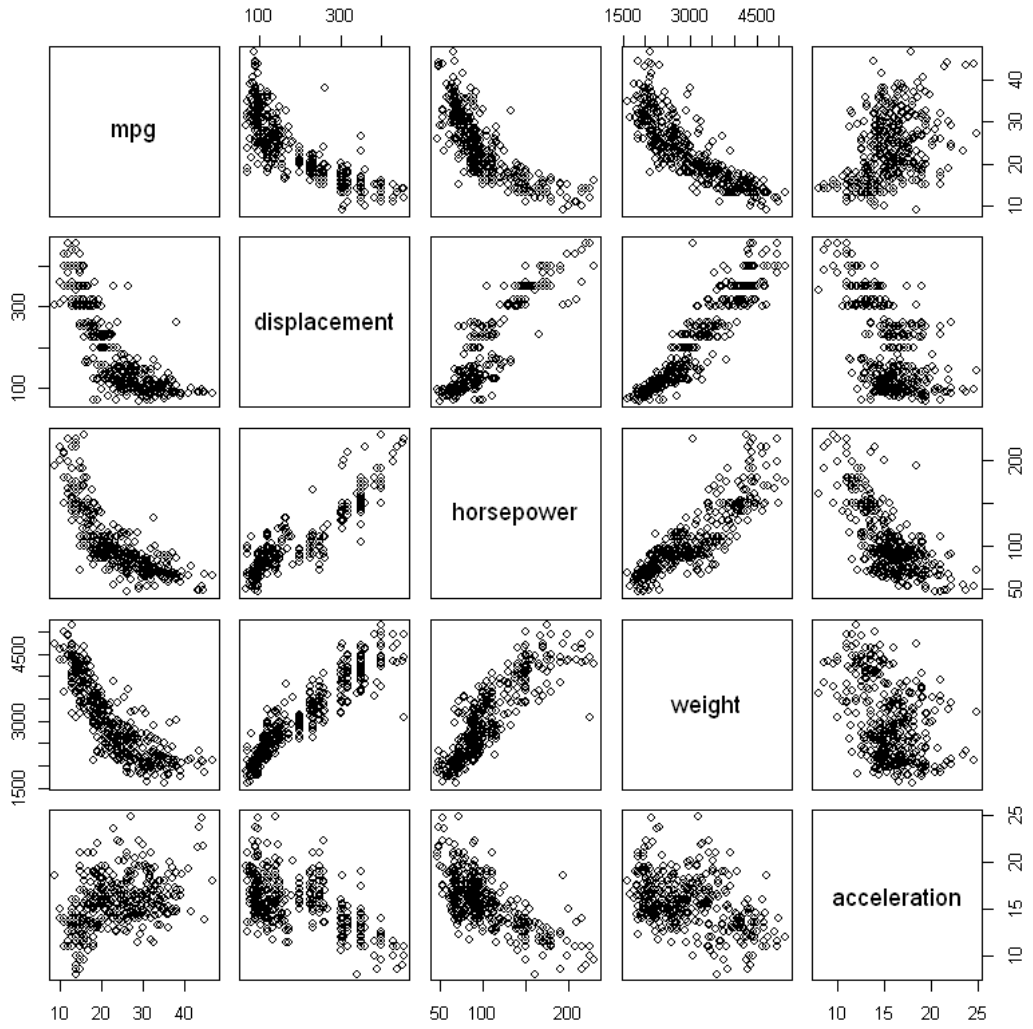

Histogram of mpg
Frequency
0
20
40
60
80
10
20
30
40
50
mpg


Histogram of mpg
Frequency
0
20
40
60
80
10
20
30
40
50
mpg

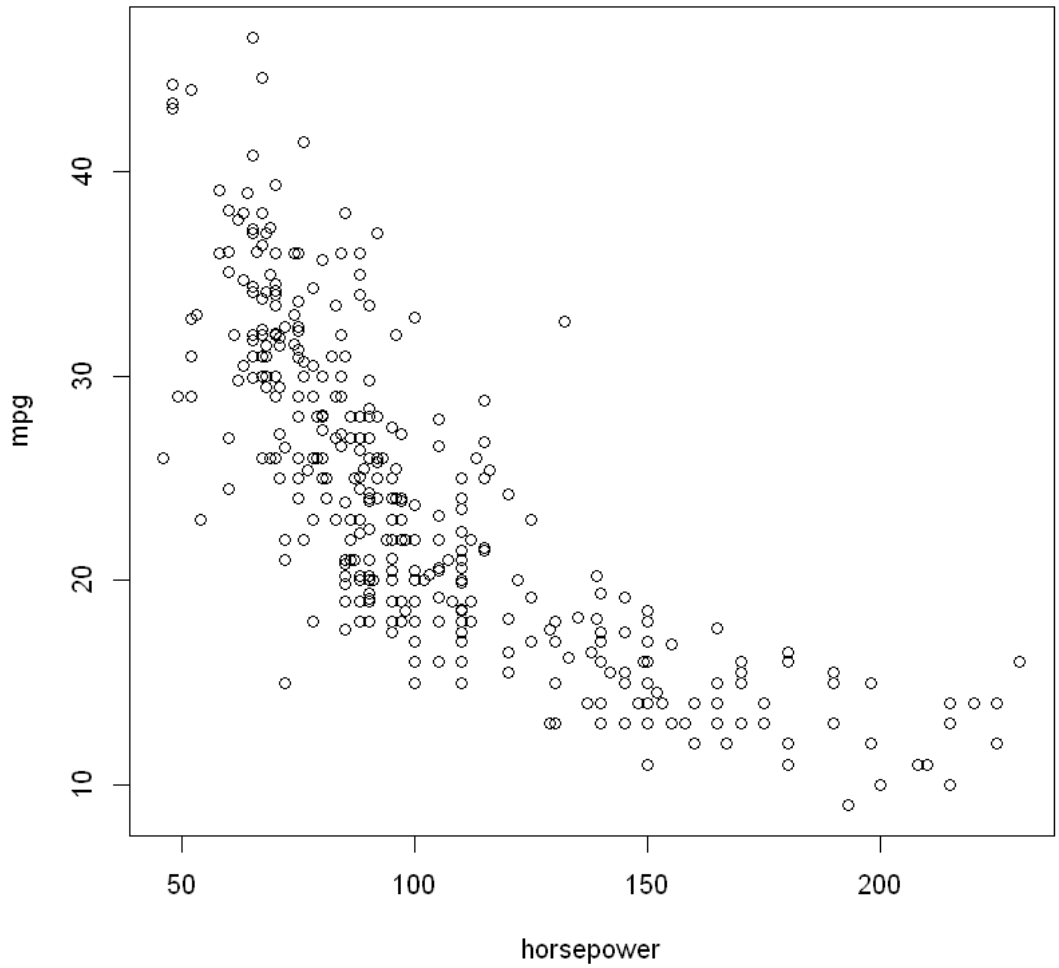


[47]:
```
pairs(Auto[,1:8])
pairs(~mpg + displacement + horsepower + weight +
acceleration, Auto)
```


mpg
cylinders
displacement
horsepower
weight
acceleration
year
origin



[48]:
```
#x11() # Makes plot appear in separate window so that the interactive function␣
 ↪identify will work
plot(horsepower ,mpg)
identify(horsepower ,mpg ,name)
```



[49]:
```
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   Median :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   Length:392
 1st Qu.:13.78   1st Qu.:73.00   1st Qu.:1.000   Class :character
 Median :15.50   Median :76.00   Median :1.000   Mode  :character
```

```
Mean   :15.54   Mean   :75.98   Mean   :1.577
3rd Qu.:17.02   3rd Qu.:79.00   3rd Qu.:2.000
Max.   :24.80   Max.   :82.00   Max.   :3.000
```

[50]:
```
summary (mpg)
```

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

[ ]:
```

```