chapter2_hw+labs

```
install.packages("tidyverse")
```

```
Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.4'
(as 'lib' is unspecified)
```

```
library(tidyverse)# load the core tidyverse packages
```

```
-- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
v dplyr     1.1.4     v readr     2.1.5
v forcats   1.0.0     v stringr   1.5.1
v ggplot2   3.5.1     v tibble    3.2.1
v lubridate 1.9.3     v tidyr     1.3.1
v purrr     1.0.2
```

```
-- Conflicts ------------------------------------------ tidyverse_conflicts() --
x dplyr::filter() masks stats::filter()
x dplyr::lag()    masks stats::lag()
i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becor
```

```
library(dplyr) # load dplyr
library(ggplot2)# gg plot
library(readxl) # need to load from tidyverse
library(readr) # to import data
library(png)
library(grid)
library(ggpubr)
library(ISLR2)
```

1.For each of parts (a) through (d), indicate whether we would generally expect the performance of a fexible statistical learning method to be better or worse than an infexible method. Justify your answer. (a) The sample size n is extremely large, and the number of predictors p is small. #fexible method better (b) The number of predictors p is extremely large, and the number of observations n is small. #infexible method better;overftting in highly fexible methods (c) The relationship between the predictors and response is highly non-linear. #a fexible statistical learning better (d) The variance of the error terms, i.e. 2 = Var( ), is extremely high. #infexible method better; if a method has high variance then small changes in the training data can result in large changes in f^. In general, more fexible statistical methods have higher variance

2.Explain whether each scenario is a classification or regression problem, and indicate whether we are most interested in inference or prediction. Finally, provide n and p.

(a) We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in understanding which factors affect CEO salary.

#regression; inference; n = 500; p = 3(profit, number of employees, industry salary)

(b) We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price,and ten other variables.

#classification; prediction; n = 20; p = 13(price charged for the product, marketing budget, competition price,and ten other variables.)

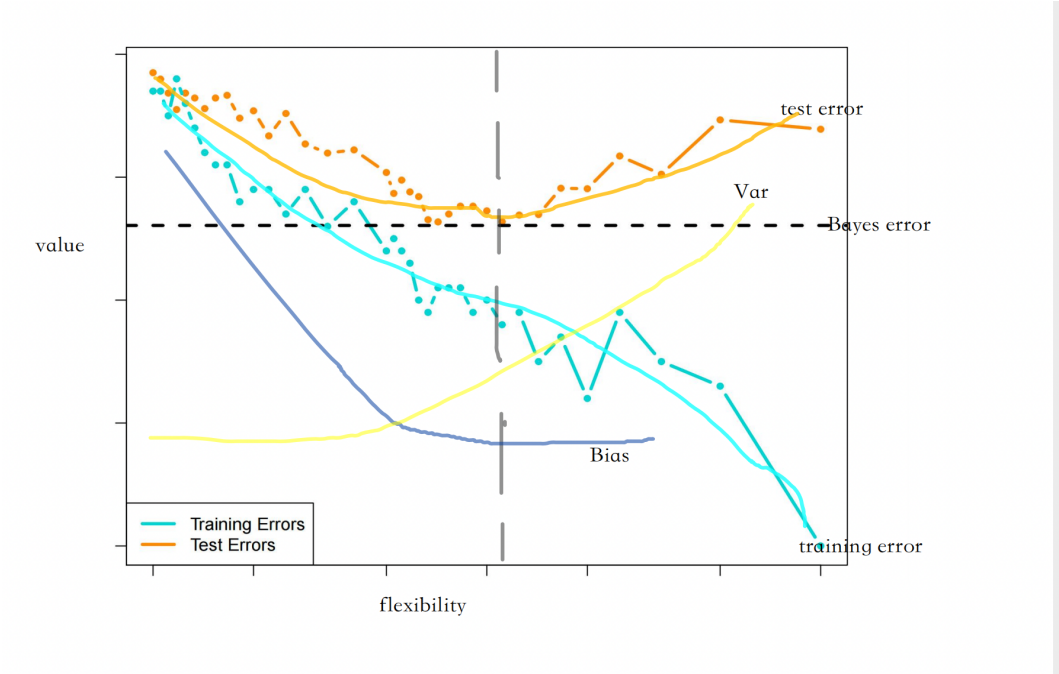
(c) We are interested in predicting the % change in the USD/Euro exchange rate in relation to the weekly changes in the world stock markets. Hence we collect weekly data for all of 2012. For each week we record the % change in the USD/Euro, the % change in the US market, the % change in the British market, and the % change in the German market.

#regression; prediction; n = 52(weeks in 2012); p = 3(the % change in the US market, the % change in the British market, and the % change in the German market)

3. We now revisit the bias-variance decomposition.

(a) Provide a sketch of typical (squared) bias, variance, training error, test error, and Bayes (or irreducible) error curves, on a single plot, as we go from less flexible statistical learning methods towards more flexible approaches. The x-axis should represent the amount of flexibility in the method, and the y-axis should represent the values for each curve. There should be five curves. Make sure to label each one.

```
rm(list=ls())
answer3 <- readPNG("bias-variance.png")
p1<-ggplot()+background_image(answer3)+theme_void()
p1
```



(b) Explain why each of the five curves has the shape displayed in part (a).

#As in the regression setting, the training error rate consistently declines as the fexibility increases.

#the test error exhibits a characteristic U-shape, declining at frst (with a minimum at approximately K = 10) before increasing again when the method becomes excessively flexible and overfits.

#The Bayes classifer produces the lowest possible test error rate

5. What are the advantages and disadvantages of a very flexible (versus a less flexible) approach for regression or classification? Under what circumstances might a more flexible approach be preferred to a less flexible approach? When might a less flexible approach be preferred?

#Generally, use more fexible methods, the variance will increase and the bias will decrease.
#when the true f is substantially non-linear, a more flexible approach would be preferred
#when the true f is close to linear, a less flexible approach would be preferred

6. Describe the differences between a parametric and a non-parametric statistical learning approach. What are the advantages of a parametric approach to regression or classification (as opposed to a nonparametric approach)? What are its disadvantages?

#a parametric form for f simplifes the problem of estimating f(generally much easier to estimate a set of parameters) #The potential disadvantage of a parametric approach is that the model we choose will usually not match the true unknown form of f.

7. The table below provides a training data set containing six observations, three predictors, and one qualitative response variable.

Suppose we wish to use this data set to make a prediction for Y when X1 = X2 = X3 = 0 using K-nearest neighbors. (a) Compute the Euclidean distance between each observation and the test point, X1 = X2 = X3 = 0.

3;2;3.162278;2.236068;1.414214;1.732051

```
sqrt(((0-0)^2+(3-0)^2+(0-0)^2))
```

```
[1] 3
```

```
sqrt(((2-0)^2+(0-0)^2+(0-0)^2))
```

```
[1] 2
```

```
sqrt(((0-0)^2+(1-0)^2+(3-0)^2))
```

```
[1] 3.162278
```

```
sqrt(((0-0)^2+(1-0)^2+(2-0)^2))
```

```
[1] 2.236068
```

```
sqrt(((-1-0)^2+(0-0)^2+(1-0)^2))
```

```
[1] 1.414214
```

```
sqrt(((1-0)^2+(1-0)^2+(1-0)^2))
```

```
[1] 1.732051
```

(b) What is our prediction with K = 1? Why? # Green (Closest:1.414214) (Obs:5)

(c) What is our prediction with K = 3? Why? #Red (1.414214;1.732051;2)(Obs:2,5,6)(Red:2/3;Green:1/3)

(d) If the Bayes decision boundary in this problem is highly nonlinear, then would we expect the best value for K to be large or small? Why? #small; Since this problem is highly nonlinear, a smaller K results in an overly flexible decision boundary

8.This exercise relates to the College data set, which can be found in the file College.csv on the book website. It contains a number of variables for 777 different universities and colleges in the US.

```
college <- read_csv("College.csv")
```

```
New names:
Rows: 777 Columns: 19
-- Column specification
-------------------------------------------------------- Delimiter: "," chr
(2): ...1, Private dbl (17): Apps, Accept, Enroll, Top10perc, Top25perc,
F.Undergrad, P.Undergr...
i Use `spec()` to retrieve the full column specification for this data. i
Specify the column types or set `show_col_types = FALSE` to quiet this message.
* `` -> `...1`
```

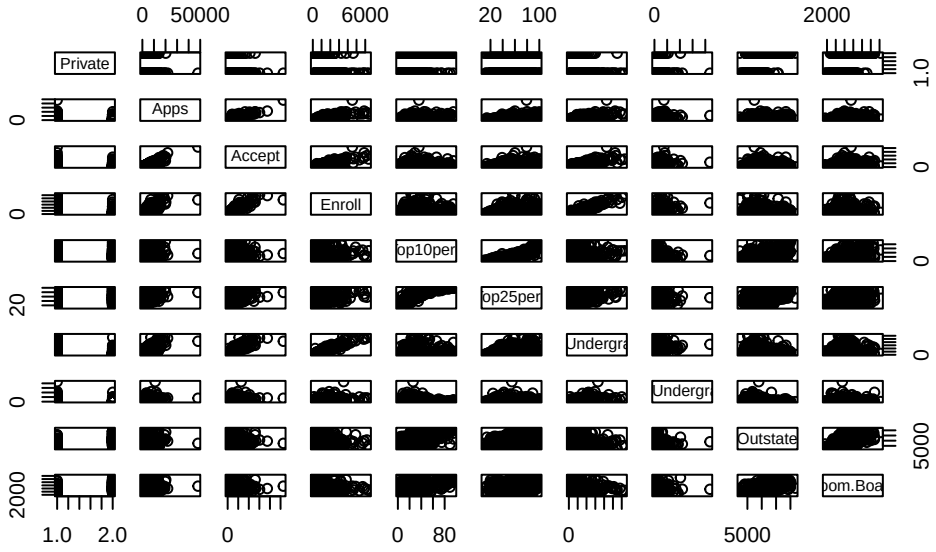
```
#change df to matrix to avoid "Error in `.rowNamesDF<-`(x, value = value)"
college <- as.matrix(college)
rownames(college) <- college[, 1]

#delete the first col
college <- college[, -1]
```

```
#change back to df and the correct variables' types
college <- as.data.frame(college)
college[, 2:ncol(college)] <- lapply(college[, 2:ncol(college)], function(x) as.numeric(as.cl
college$Private <- as.factor(college$Private)
summary(college)
```

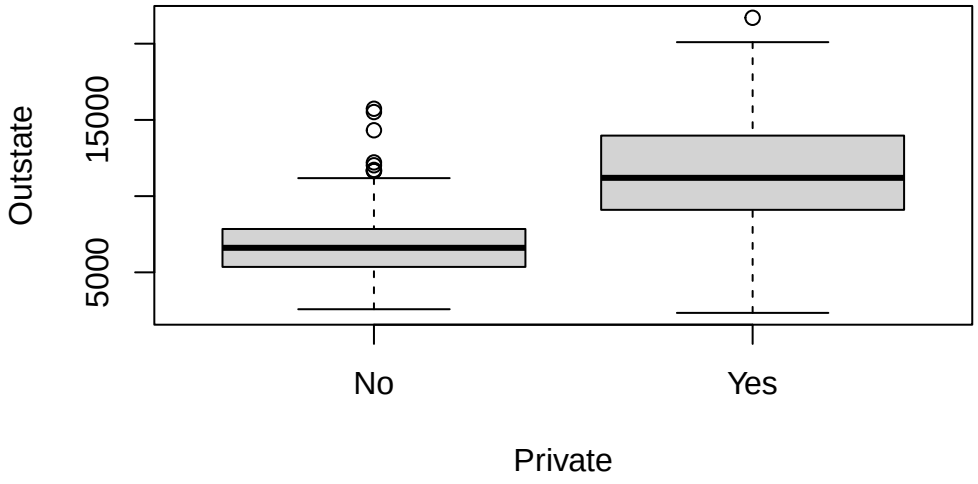
```
 Private        Apps           Accept          Enroll       Top10perc
 No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
 Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
```

```
          Median : 1558   Median : 1110   Median : 434   Median :23.00
          Mean   : 3002   Mean   : 2019   Mean   : 780   Mean   :27.56
          3rd Qu.: 3624   3rd Qu.: 2424   3rd Qu.: 902   3rd Qu.:35.00
          Max.   :48094   Max.   :26330   Max.   :6392   Max.   :96.00
  Top25perc      F.Undergrad     P.Undergrad         Outstate
Min.   :  9.0   Min.   :  139   Min.   :    1.0   Min.   : 2340
1st Qu.: 41.0   1st Qu.:  992   1st Qu.:   95.0   1st Qu.: 7320
Median : 54.0   Median : 1707   Median :  353.0   Median : 9990
Mean   : 55.8   Mean   : 3700   Mean   :  855.3   Mean   :10441
3rd Qu.: 69.0   3rd Qu.: 4005   3rd Qu.:  967.0   3rd Qu.:12925
Max.   :100.0   Max.   :31643   Max.   :21836.0   Max.   :21700
  Room.Board       Books           Personal         PhD
Min.   :1780   Min.   :  96.0   Min.   : 250   Min.   :  8.00
1st Qu.:3597   1st Qu.: 470.0   1st Qu.: 850   1st Qu.: 62.00
Median :4200   Median : 500.0   Median :1200   Median : 75.00
Mean   :4358   Mean   : 549.4   Mean   :1341   Mean   : 72.66
3rd Qu.:5050   3rd Qu.: 600.0   3rd Qu.:1700   3rd Qu.: 85.00
Max.   :8124   Max.   :2340.0   Max.   :6800   Max.   :103.00
   Terminal       S.F.Ratio      perc.alumni        Expend
Min.   : 24.0   Min.   : 2.50   Min.   : 0.00   Min.   : 3186
1st Qu.: 71.0   1st Qu.:11.50   1st Qu.:13.00   1st Qu.: 6751
Median : 82.0   Median :13.60   Median :21.00   Median : 8377
Mean   : 79.7   Mean   :14.09   Mean   :22.74   Mean   : 9660
3rd Qu.: 92.0   3rd Qu.:16.50   3rd Qu.:31.00   3rd Qu.:10830
Max.   :100.0   Max.   :39.80   Max.   :64.00   Max.   :56233
  Grad.Rate
Min.   : 10.00
1st Qu.: 53.00
Median : 65.00
Mean   : 65.46
3rd Qu.: 78.00
Max.   :118.00
```

```r
pairs(college[,1:10])
```



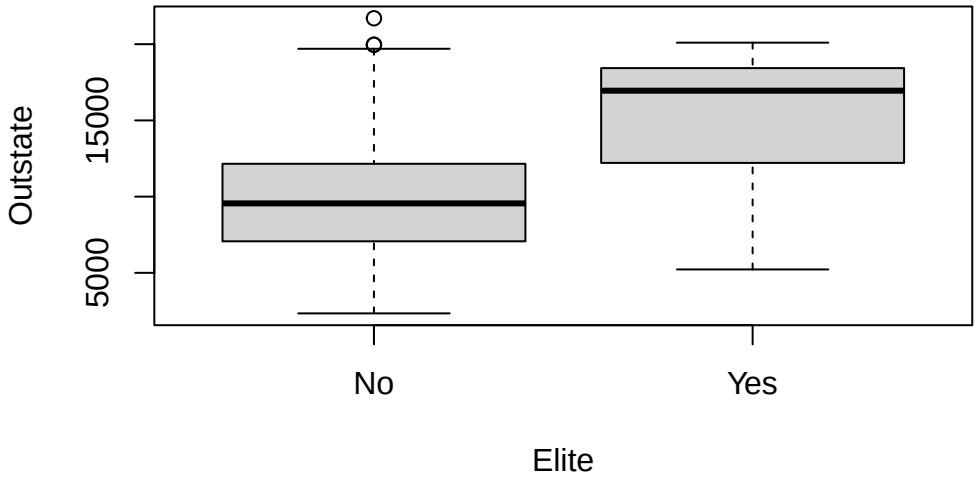

```
plot(college$Private, college$Outstate, xlab = "Private", ylab = "Outstate")
```



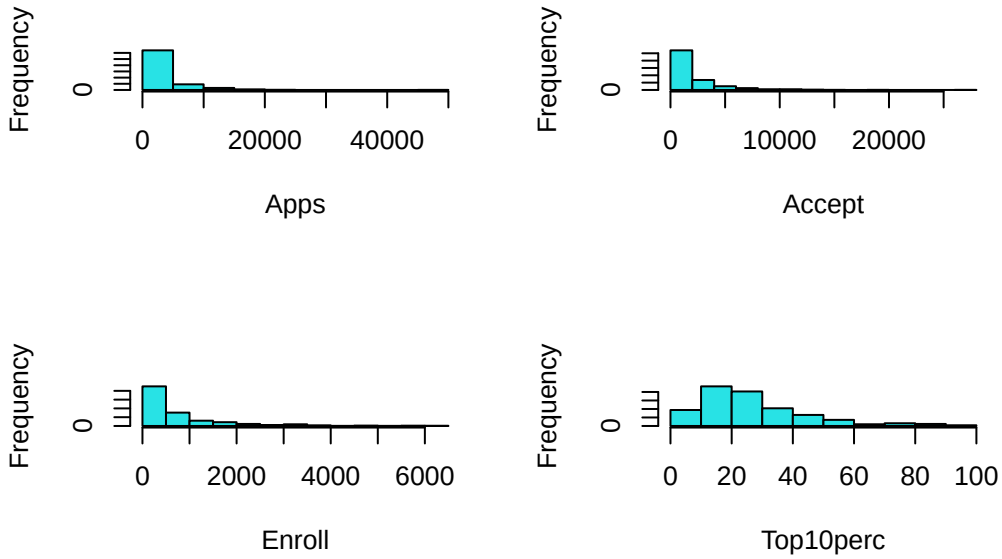

```
Elite <- rep("No", nrow(college))
Elite[college$Top10perc > 50] <- "Yes"
Elite <- as.factor(Elite)
college <- data.frame(college, Elite)
# 78 elite universities
summary(college$Elite)
```

```
 No Yes
699  78
```

```
plot(college$Elite, college$Outstate, xlab = "Elite", ylab = "Outstate")
```




```
par(mfrow = c(2, 2))
hist(college$Apps, col =5, xlab = "Apps", main = NULL)
hist(college$Accept, col =5, xlab = "Accept", main = NULL)
hist(college$Enroll, col =5, xlab = "Enroll", main = NULL)
hist(college$Top10perc, col =5, xlab = "Top10perc", main = NULL)
```



9.This exercise involves the Auto data set studied in the lab. Make sure that the missing values have been removed from the data.

```
Auto <- read_csv("Auto.csv")
```

```
Rows: 397 Columns: 9
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): horsepower, name
dbl (7): mpg, cylinders, displacement, weight, acceleration, year, origin

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
Auto$horsepower <- as.numeric(Auto$horsepower)
```

```
Warning: NAs introduced by coercion
```

```
Auto <- na.omit(Auto)
#quantitative:mpg; cylinders; displacement; horsepower;weight; acceleration;year;
#qualitative:origin; name
#mpg
range(Auto[, 1])
```

```
[1]  9.0 46.6
```

```
#cylinders
range(Auto[, 2])
```

```
[1] 3 8
```

```
#displacement
range(Auto[, 3])
```

```
[1]  68 455
```

```
#horsepower
range(Auto[, 4])
```

```
[1]  46 230
```

```
#weight
range(Auto[, 5])
```

```
[1] 1613 5140
```

```
#acceleration
range(Auto[, 6])
```

```
[1]  8.0 24.8
```

```
#year
range(Auto[, 7])
```

```
[1] 70 82
```

(c) What is the mean and standard deviation of each quantitative predictor?

```
sapply(Auto[, 1:7], mean)
```

```
       mpg    cylinders displacement   horsepower       weight acceleration
  23.445918     5.471939   194.411990   104.469388  2977.584184    15.541327
      year
  75.979592
```

```
sapply(Auto[, 1:7], sd)
```

```
       mpg    cylinders displacement   horsepower       weight acceleration
  7.805007     1.705783   104.644004    38.491160   849.402560     2.758864
      year
  3.683737
```

(d) Now remove the 10th through 85th observations. What is the range, mean, and standard deviation of each predictor in the subset of the data that remains?

```
Auto_sub <- Auto[-(10:85), ]
sapply(Auto_sub[, 1:8], range)
```

```
     mpg cylinders displacement horsepower weight acceleration year origin
[1,] 11.0         3           68         46   1649          8.5   70      1
[2,] 46.6         8          455        230   4997         24.8   82      3
```

```
sapply(Auto_sub[, 1:8], mean)
```

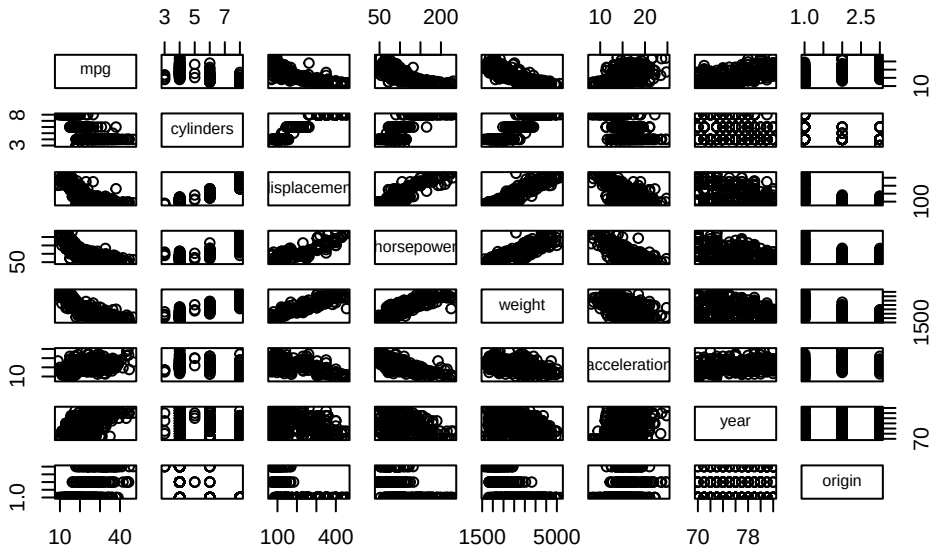
```
        mpg    cylinders displacement   horsepower       weight acceleration
  24.404430     5.373418   187.240506   100.721519  2935.971519    15.726899
       year       origin
  77.145570     1.601266
```

```
sapply(Auto_sub[, 1:8], sd)
```

```
        mpg    cylinders displacement   horsepower       weight acceleration
   7.867283     1.654179    99.678367    35.708853   811.300208     2.693721
       year       origin
   3.106217     0.819910
```

(e) Using the full data set, investigate the predictors graphically, using scatterplots or other tools of your choice. Create some plots highlighting the relationships among the predictors. Comment on your findings.

```
pairs(Auto[, 1:8])#feel like name can't be plotted
```



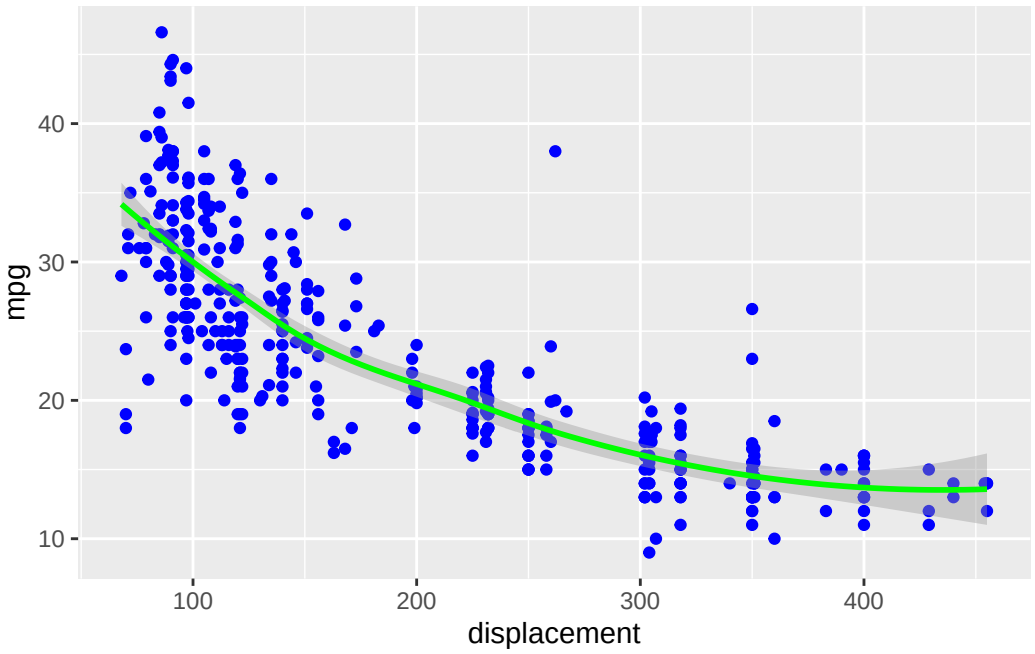
(f) Suppose that we wish to predict gas mileage (mpg) on the basis of the other variables. Do your plots suggest that any of the other variables might be useful in predicting mpg? Justify your answer. #displacement; horsepower;weight

```
cor(Auto[,1:7])
```

```
                     mpg  cylinders displacement horsepower     weight
mpg           1.0000000 -0.7776175   -0.8051269 -0.7784268 -0.8322442
cylinders    -0.7776175  1.0000000    0.9508233  0.8429834  0.8975273
displacement -0.8051269  0.9508233    1.0000000  0.8972570  0.9329944
horsepower   -0.7784268  0.8429834    0.8972570  1.0000000  0.8645377
weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
             acceleration       year
mpg             0.4233285  0.5805410
cylinders      -0.5046834 -0.3456474
displacement   -0.5438005 -0.3698552
horsepower     -0.6891955 -0.4163615
weight         -0.4168392 -0.3091199
acceleration    1.0000000  0.2903161
year            0.2903161  1.0000000
```

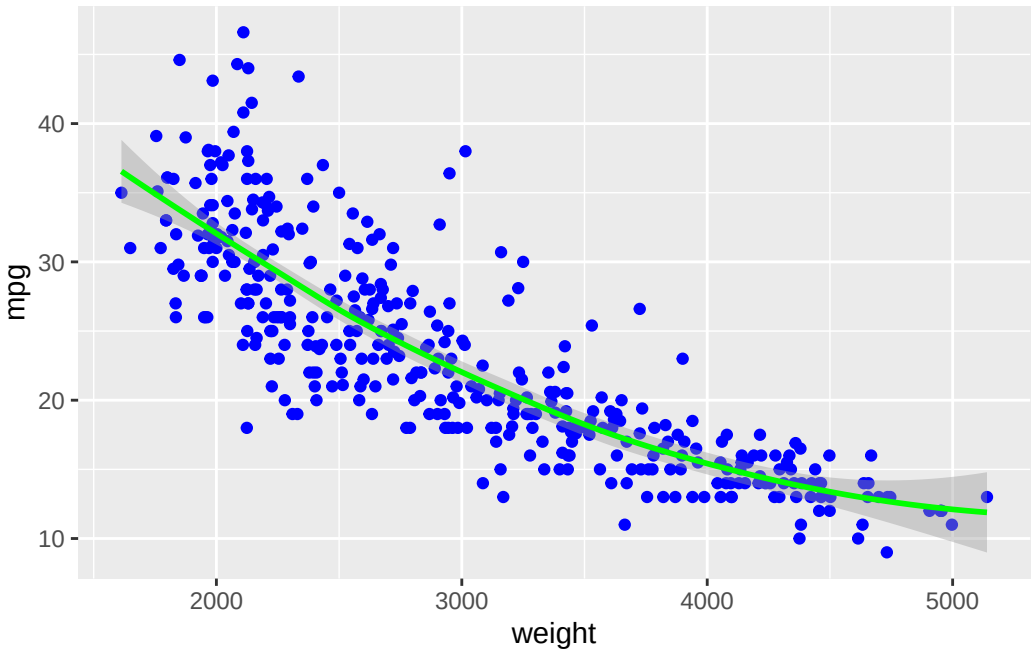
```
Auto %>%
  ggplot(aes(x = displacement, y = mpg)) +
  geom_point(color = "blue")+
  geom_smooth(color = "green")
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```




```
Auto %>%
  ggplot(aes(x = weight, y = mpg)) +
  geom_point(color = "blue")+
  geom_smooth(color = "green")
```

```
`geom_smooth()` using method = 'loess' and formula = 'y ~ x'
```



10. This exercise involves the Boston housing data set.

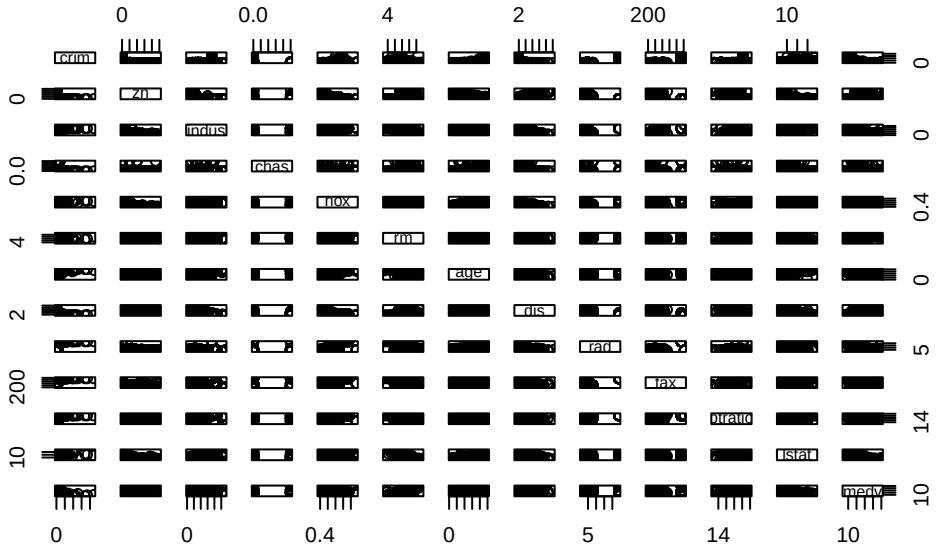
(a) To begin, load in the Boston data set. The Boston data set is part of the ISLR2 library How many rows are in this data set? How many columns? What do the rows and columns represent?

```
head(Boston) # 506*13; rows the size of the sample; cols the number of independent variable/p
```

```
     crim zn indus chas   nox    rm  age    dis rad tax ptratio lstat medv
1 0.00632 18  2.31    0 0.538 6.575 65.2 4.0900   1 296    15.3  4.98 24.0
2 0.02731  0  7.07    0 0.469 6.421 78.9 4.9671   2 242    17.8  9.14 21.6
3 0.02729  0  7.07    0 0.469 7.185 61.1 4.9671   2 242    17.8  4.03 34.7
4 0.03237  0  2.18    0 0.458 6.998 45.8 6.0622   3 222    18.7  2.94 33.4
5 0.06905  0  2.18    0 0.458 7.147 54.2 6.0622   3 222    18.7  5.33 36.2
6 0.02985  0  2.18    0 0.458 6.430 58.7 6.0622   3 222    18.7  5.21 28.7
```

(b) Make some pairwise scatterplots of the predictors (columns) in this data set. Describe your fndings.

```
pairs(Boston)
```



(c) Are any of the predictors associated with per capital crime rate? If so, explain the relationship.

```
cor(Boston[,1:13])
```

```
                crim          zn       indus         chas         nox
crim     1.00000000 -0.20046922  0.40658341 -0.055891582  0.42097171
zn      -0.20046922  1.00000000 -0.53382819 -0.042696719 -0.51660371
indus    0.40658341 -0.53382819  1.00000000  0.062938027  0.76365145
chas    -0.05589158 -0.04269672  0.06293803  1.000000000  0.09120281
nox      0.42097171 -0.51660371  0.76365145  0.091202807  1.00000000
rm      -0.21924670  0.31199059 -0.39167585  0.091251225 -0.30218819
age      0.35273425 -0.56953734  0.64477851  0.086517774  0.73147010
dis     -0.37967009  0.66440822 -0.70802699 -0.099175780 -0.76923011
rad      0.62550515 -0.31194783  0.59512927 -0.007368241  0.61144056
tax      0.58276431 -0.31456332  0.72076018 -0.035586518  0.66802320
ptratio  0.28994558 -0.39167855  0.38324756 -0.121515174  0.18893268
lstat    0.45562148 -0.41299457  0.60379972 -0.053929298  0.59087892
medv    -0.38830461  0.36044534 -0.48372516  0.175260177 -0.42732077
                rm         age         dis          rad         tax    ptratio
crim    -0.21924670  0.35273425 -0.37967009  0.625505145  0.58276431  0.2899456
zn       0.31199059 -0.56953734  0.66440822 -0.311947826 -0.31456332 -0.3916785
indus   -0.39167585  0.64477851 -0.70802699  0.595129275  0.72076018  0.3832476
chas     0.09125123  0.08651777 -0.09917578 -0.007368241 -0.03558652 -0.1215152
nox     -0.30218819  0.73147010 -0.76923011  0.611440563  0.66802320  0.1889327
rm       1.00000000 -0.24026493  0.20524621 -0.209846668 -0.29204783 -0.3555015
```

```
age     -0.24026493  1.00000000 -0.74788054  0.456022452  0.50645559  0.2615150
dis      0.20524621 -0.74788054  1.00000000 -0.494587930 -0.53443158 -0.2324705
rad     -0.20984667  0.45602245 -0.49458793  1.000000000  0.91022819  0.4647412
tax     -0.29204783  0.50645559 -0.53443158  0.910228189  1.00000000  0.4608530
ptratio -0.35550149  0.26151501 -0.23247054  0.464741179  0.46085304  1.0000000
lstat   -0.61380827  0.60233853 -0.49699583  0.488676335  0.54399341  0.3740443
medv     0.69535995 -0.37695457  0.24992873 -0.381626231 -0.46853593 -0.5077867
              lstat       medv
crim     0.4556215 -0.3883046
zn      -0.4129946  0.3604453
indus    0.6037997 -0.4837252
chas    -0.0539293  0.1752602
nox      0.5908789 -0.4273208
rm      -0.6138083  0.6953599
age      0.6023385 -0.3769546
dis     -0.4969958  0.2499287
rad      0.4886763 -0.3816262
tax      0.5439934 -0.4685359
ptratio  0.3740443 -0.5077867
lstat    1.0000000 -0.7376627
medv    -0.7376627  1.0000000
```

```
#rad; tax
```

(d) Do any of the census tracts of Boston appear to have particularly high crime rates? Tax rates? Pupil-teacher ratios? Comment on the range of each predictor.

```
range(Boston$crim) #crime rates
```

```
[1]  0.00632 88.97620
```

```
range(Boston$tax) #Tax rates
```

```
[1] 187 711
```

```
range(Boston$ptratio)
```

```
[1] 12.6 22.0
```

(e) How many of the census tracts in this data set bound the Charles river?

```
Boston %>%
  count(chas == 1) #35
```

```
  chas == 1   n
1     FALSE 471
2      TRUE  35
```

(f) What is the median pupil-teacher ratio among the towns in this data set?

```
summary(Boston$ptratio) #19.05
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
  12.60   17.40   19.05   18.46   20.20   22.00
```

(g) Which census tract of Boston has lowest median value of owner occupied homes? What are the values of the other predictors for that census tract, and how do those values compare to the overall ranges for those predictors? Comment on your findings.

```
Boston %>%
  filter(medv == min(medv)) # tax rate; ptratio high in the range
```

```
     crim zn indus chas   nox    rm age    dis rad tax ptratio lstat medv
1 38.3518  0  18.1    0 0.693 5.453 100 1.4896  24 666    20.2 30.59    5
2 67.9208  0  18.1    0 0.693 5.683 100 1.4254  24 666    20.2 22.98    5
```

```
range(Boston$crim) #crime rates
```

```
[1]  0.00632 88.97620
```

```
range(Boston$tax) #Tax rates
```

```
[1] 187 711
```

```
range(Boston$ptratio)
```

```
[1] 12.6 22.0
```

(h) In this data set, how many of the census tracts average more than seven rooms per dwelling? More than eight rooms per dwelling? Comment on the census tracts that average more than eight rooms per dwelling

```
Boston %>%
  filter(rm > 7) %>%
  count() #64
```

```
  n
1 64
```

```
Boston %>%
  filter(rm > 8) %>%
  count() #13 Census tracts with an average of more than eight rooms per dwelling are signif
```

```
  n
1 13
```

##########labs ##basics

```
x <- c(1,2,3) #vector
ls()# a list of all of the objects;like data and functions
```

```
[1] "answer3"  "Auto"     "Auto_sub" "college"  "Elite"    "p1"       "x"
```

```
rm(x)#delete
rm(list = ls())# remove list of all objects
```

##matrix

```
     [,1] [,2]
[1,]    2    2
[2,]    3    6
```

```
     [,1] [,2]
[1,]    2    3
[2,]    2    6
```

#The sqrt() function returns the square root of each element of a vector or matrix. The command x2raises each element of x sqrt() to the power 2

```
sqrt(y)
```

```
         [,1]     [,2]
[1,] 1.414214 1.414214
[2,] 1.732051 2.449490
```

```
y^2
```

```
     [,1] [,2]
[1,]    4    4
[2,]    9   36
```

#The rnorm() function generates a vector of random normal variables, with frst argument n the sample size. #By default, 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 = .1)
cor(x,y)
```

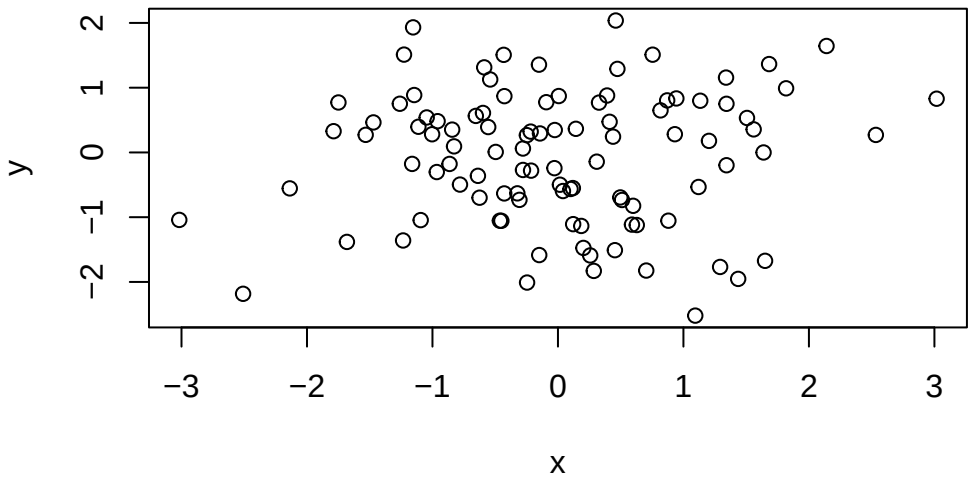
```
[1] 0.9961631
```

```
set.seed(1008)
rnorm(50)
```

```
 [1]  0.43318704  0.81822111 -2.02763858  0.69069198 -0.34684495  2.27165870
 [7]  1.60450776 -0.49576719  0.11615780  0.33592741  1.26685176  0.03692722
[13]  0.64930389  0.05461035  0.67520367  1.92052240  0.04463261  0.36282407
[19] -0.09014921 -0.32928758  0.21442911 -0.15480663 -0.28005953  1.40621827
[25]  1.27670624 -0.64265551 -1.08470109 -0.52066460 -2.00970361 -0.59796924
[31] -0.35687791  1.18058964  0.44783843  0.34094338 -1.20904130  0.96376536
[37] -1.80910015 -0.03602382 -0.68290703  0.86683673 -0.91495798 -1.66660652
[43]  0.57028117 -0.50309439 -0.46862812 -0.94754647  0.72386078 -0.98830071
[49]  1.14527957 -1.10987353
```

#The function dev.off() indicates to R that we are done creating the plot.

```
x <- rnorm(100)
y <- rnorm(100)
plot(x,y)
```



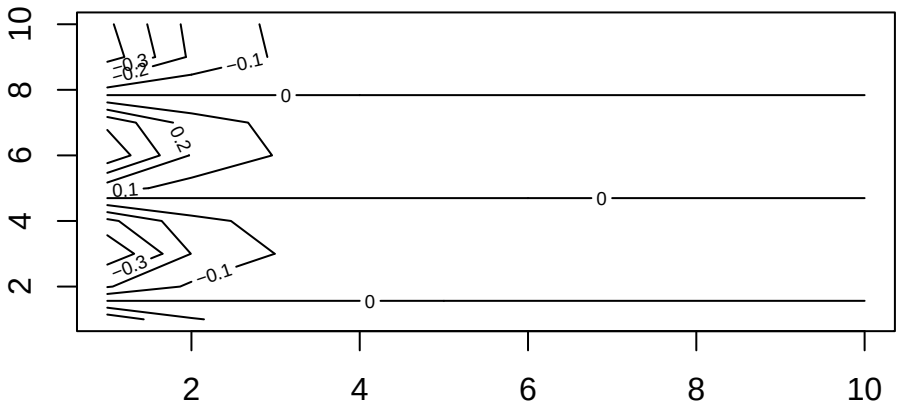
#run the 3 lines together

```
pdf("Figure.pdf")
plot(x, y, col = "green")

dev.off()
```

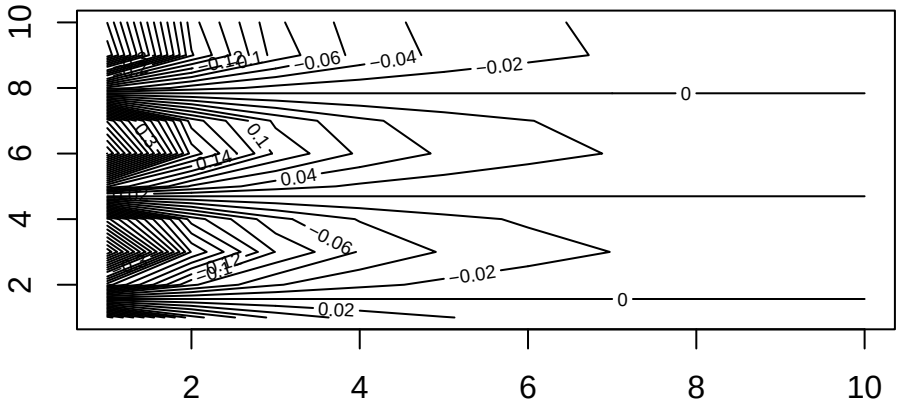
```
pdf
  2
```

#The function seq() can be used to create a sequence of numbers. #The contour() function produces a contour plot in order to represent three-dimensional data; contour plot it is like a topographical map

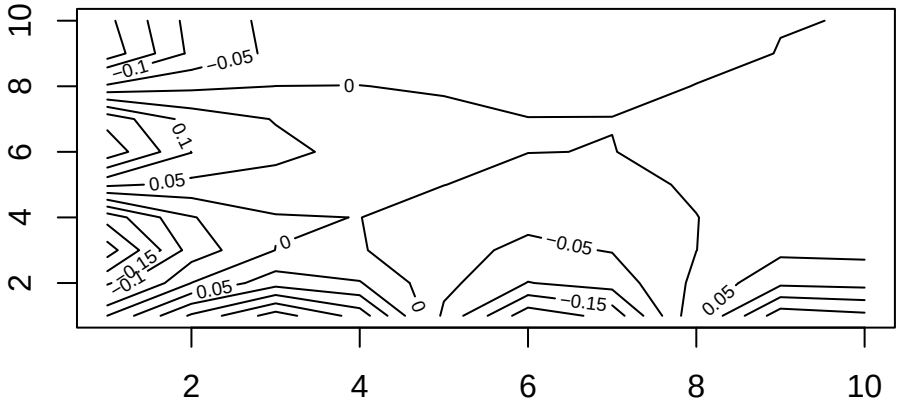
```
x <- seq(1, 10)
y <- x

f <- outer(x, y, function(x, y) cos(y) / (1 + x^2))


contour(x, y, f)
```



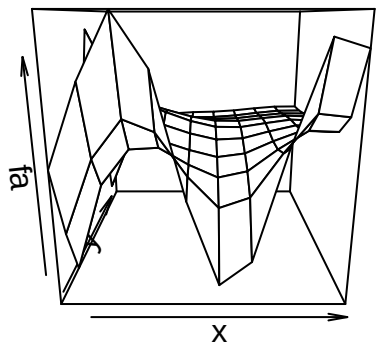

```
contour(x, y, f, nlevels = 45)
```



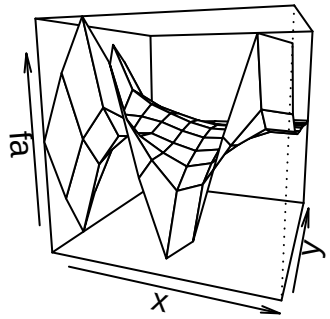

```
fa <- (f - t(f)) / 2
contour(x, y, fa, nlevels = 15)
```



```
persp(x, y, fa)
```




```
persp(x, y, fa, theta = 20, phi = 10)
```




```
A <- matrix(1:16,4,4)
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
```

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

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

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

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

#attach(Auto) use the attach() function in attach() order to tell R to make the variables in this data frame available by name. #The as.factor() function converts quantitative variables into qualitative as.factor() variables.

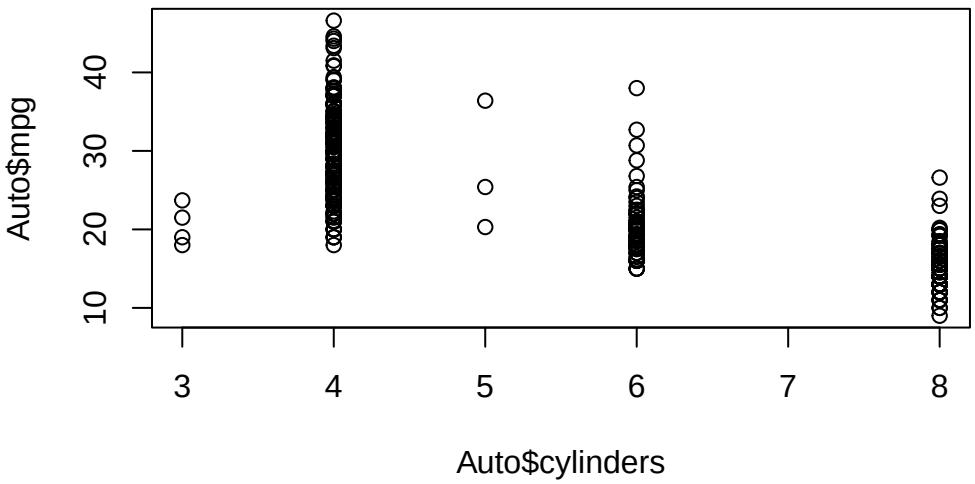
```
rm(cylinders)
```

```
Warning in rm(cylinders): object 'cylinders' not found
```

```
Auto <- read_csv("Auto.csv")
```

```
Rows: 397 Columns: 9
-- Column specification --------------------------------------------------------
Delimiter: ","
chr (2): horsepower, name
dbl (7): mpg, cylinders, displacement, weight, acceleration, year, origin

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

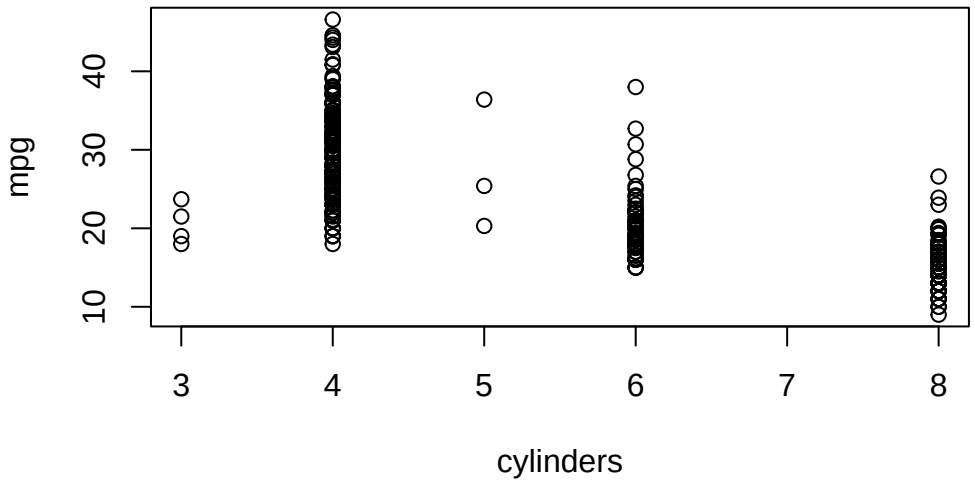
```
plot(Auto$cylinders,Auto$mpg)
```



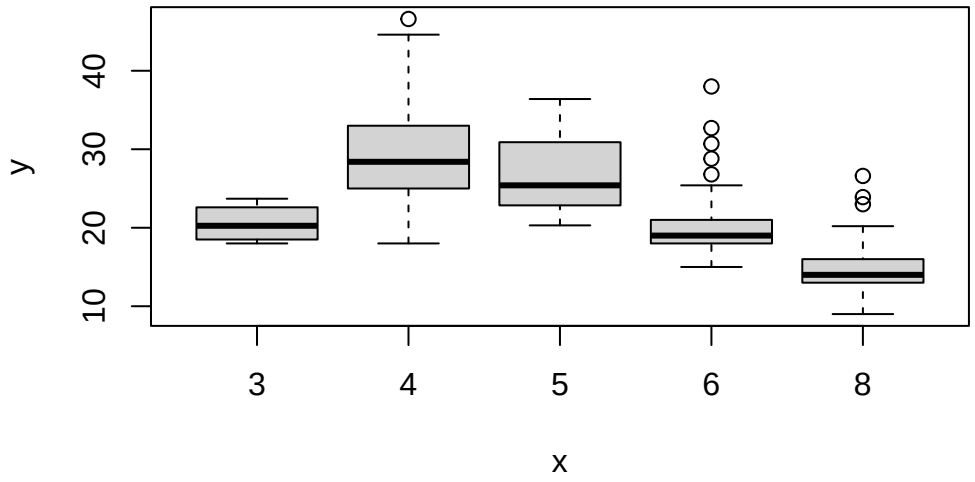
```
attach(Auto)
```

```
The following object is masked from package:lubridate:

    origin

The following object is masked from package:ggplot2:

    mpg
```

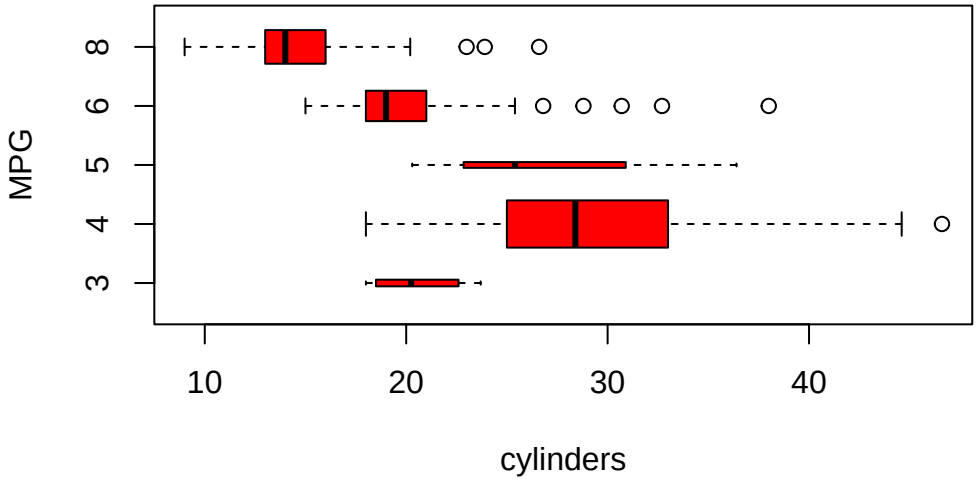
```
plot(cylinders, mpg)
```



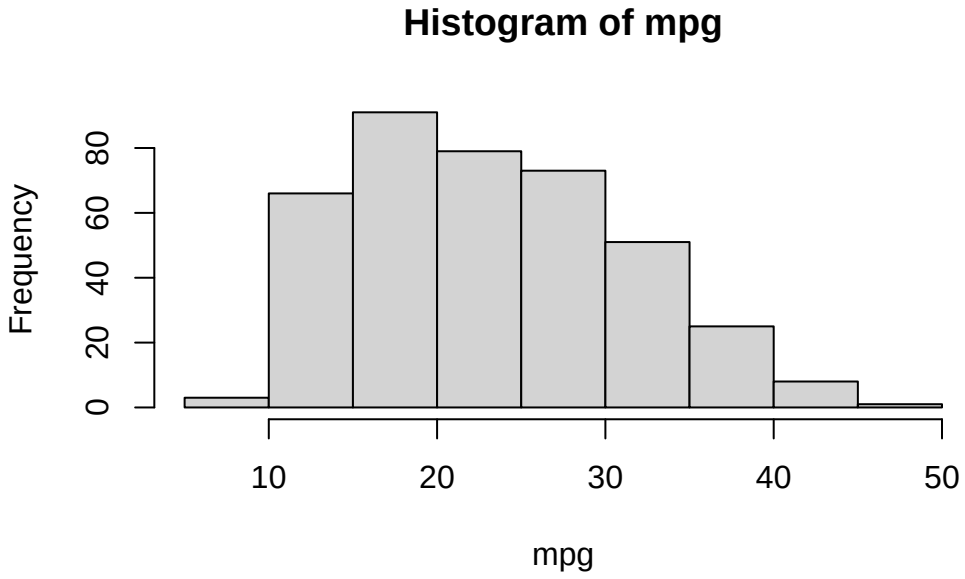

```
cylinders <- as.factor(cylinders)
plot(cylinders, mpg)
```



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




```
hist(mpg)
```




```
rm(cylinders)
Auto <- read_csv("Auto.csv")
```

```
Rows: 397 Columns: 9
-- Column specification --------------------------------------------------------
```

```
Delimiter: ","
chr (2): horsepower, name
dbl (7): mpg, cylinders, displacement, weight, acceleration, year, origin

i Use `spec()` to retrieve the full column specification for this data.
i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

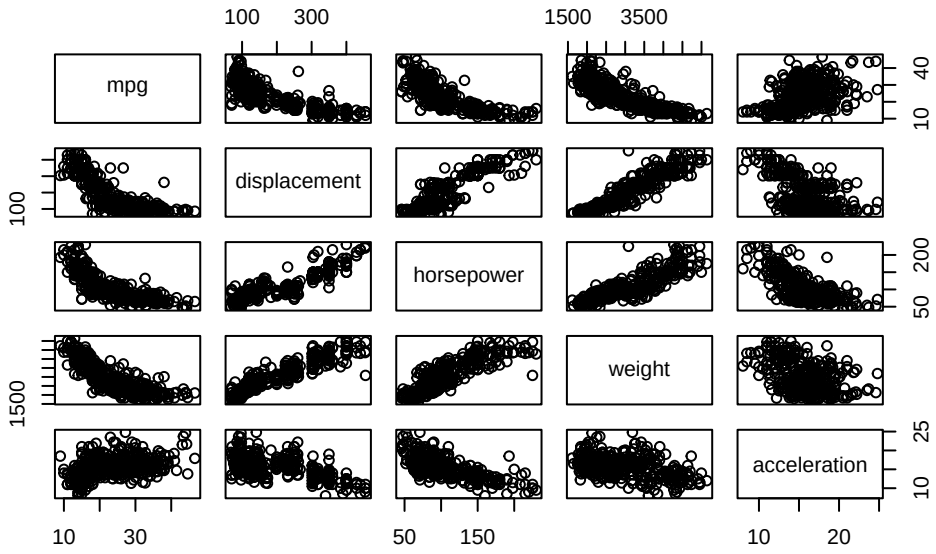
```
attach(Auto)
```

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

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

The following object is masked from package:lubridate:

    origin

The following object is masked from package:ggplot2:

    mpg
```

```
Auto$horsepower <- as.numeric(horsepower)
```

```
Warning: NAs introduced by coercion
```

```
pairs( ~ mpg + displacement + horsepower + weight + acceleration, data = Auto)
```

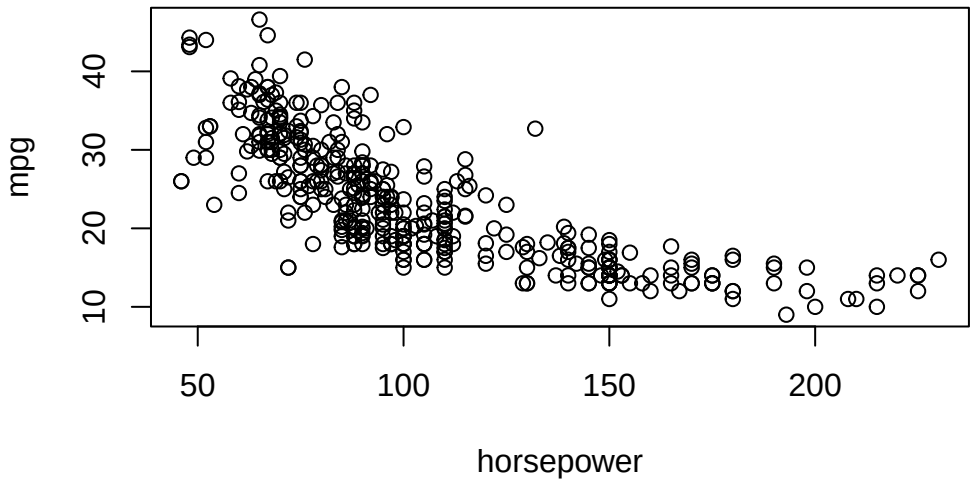


#run the two line of code together

```
attach(Auto)
```

```
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 object is masked from package:lubridate:

    origin

The following object is masked from package:ggplot2:

    mpg
```

```
plot(horsepower, mpg)
identify(horsepower, mpg, labels=name)
```




```
integer(0)
```

```
summary(Auto)
```

```
      mpg          cylinders      displacement     horsepower        weight
 Min.   : 9.00   Min.   :3.000   Min.   : 68.0   Min.   : 46.0   Min.   :1613
 1st Qu.:17.50   1st Qu.:4.000   1st Qu.:104.0   1st Qu.: 75.0   1st Qu.:2223
 Median :23.00   Median :4.000   Median :146.0   Median : 93.5   Median :2800
 Mean   :23.52   Mean   :5.458   Mean   :193.5   Mean   :104.5   Mean   :2970
 3rd Qu.:29.00   3rd Qu.:8.000   3rd Qu.:262.0   3rd Qu.:126.0   3rd Qu.:3609
 Max.   :46.60   Max.   :8.000   Max.   :455.0   Max.   :230.0   Max.   :5140
                                                 NA's   :5
  acceleration        year           origin          name
 Min.   : 8.00   Min.   :70.00   Min.   :1.000   Length:397
 1st Qu.:13.80   1st Qu.:73.00   1st Qu.:1.000   Class :character
 Median :15.50   Median :76.00   Median :1.000   Mode  :character
 Mean   :15.56   Mean   :75.99   Mean   :1.574
 3rd Qu.:17.10   3rd Qu.:79.00   3rd Qu.:2.000
 Max.   :24.80   Max.   :82.00   Max.   :3.000
```