ISLR_exercises——Cpt2

Chapter 2 1.(a)

```
#inflexible, a flexible model might try too hard to fit all data points
```

1.(b)

```
#inflexible, a flexible model might try too hard to utilize all predictors
```

1.(c)

```
#flexible, an inflexible model might neglect the underlying non-linear
relationship
```

1.(d)

```
#inflexible, a flexible model might try too hard to fit the noise
```

2.(a)

```
#regression, inference, n=500, p=3
```

2.(b)

```
#classification, prediction, n=20, p=13
```

2.(c)

```
#regression, prediction, n=52, p=3
```

3.(a)

```
#N/A
```

3.(b)

```
#Bayes error is constant
#training error drops monotonically as model get more complex and so is bias,
while variance moves towards the opposite direction
#test error = bias^2 + variance + Bayes error
```

4.(a)

```
#N/A
```

4.(b)

```
#N/A
```

4.(c)

```
#N/A
```

5

```
#pros:Less bias, cons:More variance, more likely to overfit
#true relationship is complex
#true relationship is simple
```

6

```
#whether we need to estimate the parameters in the final form of the model
#easier to interpret, easier to do inference
#oversimplify the true relationship
```

7.(a)

```
#Obs, Dis
#1,   3
#2,   2
#3,   3.16
#4    2.24
#5,   1.41
#6,   1.73
```

7.(b)

```
#green, one green
```

7.(c)

```
#red, two red, one green
```

7.(d)

```
#small, we need a flexible model
```

8.(a)

```
library(ISLR)
attach(College)
college <- College
```

8.(b)

```
head(rownames(college))
```

```
## [1] "Abilene Christian University" "Adelphi University"
## [3] "Adrian College"               "Agnes Scott College"
## [5] "Alaska Pacific University"    "Albertson College"
```

```
head(college)
```

```
##                              Private Apps Accept Enroll Top10perc
Top25perc
```

```
## Abilene Christian University     Yes 1660   1232    721        23
52
## Adelphi University               Yes 2186   1924    512        16
29
## Adrian College                   Yes 1428   1097    336        22
50
## Agnes Scott College              Yes  417    349    137        60
89
## Alaska Pacific University        Yes  193    146     55        16
44
## Albertson College                Yes  587    479    158        38
62
##                              F.Undergrad P.Undergrad Outstate Room.Board
Books
## Abilene Christian University        2885         537     7440       3300
450
## Adelphi University                  2683        1227    12280       6450
750
## Adrian College                      1036          99    11250       3750
400
## Agnes Scott College                  510          63    12960       5450
450
## Alaska Pacific University            249         869     7560       4120
800
## Albertson College                    678          41    13500       3335
500
##                              Personal PhD Terminal S.F.Ratio perc.alumni
Expend
## Abilene Christian University     2200  70       78      18.1          12
7041
## Adelphi University               1500  29       30      12.2          16
10527
## Adrian College                   1165  53       66      12.9          30
8735
## Agnes Scott College               875  92       97       7.7          37
19016
## Alaska Pacific University        1500  76       72      11.9           2
10922
## Albertson College                 675  67       73       9.4          11
9727
##                              Grad.Rate
## Abilene Christian University        60
## Adelphi University                  56
## Adrian College                      54
## Agnes Scott College                 59
## Alaska Pacific University           15
## Albertson College                   55
```

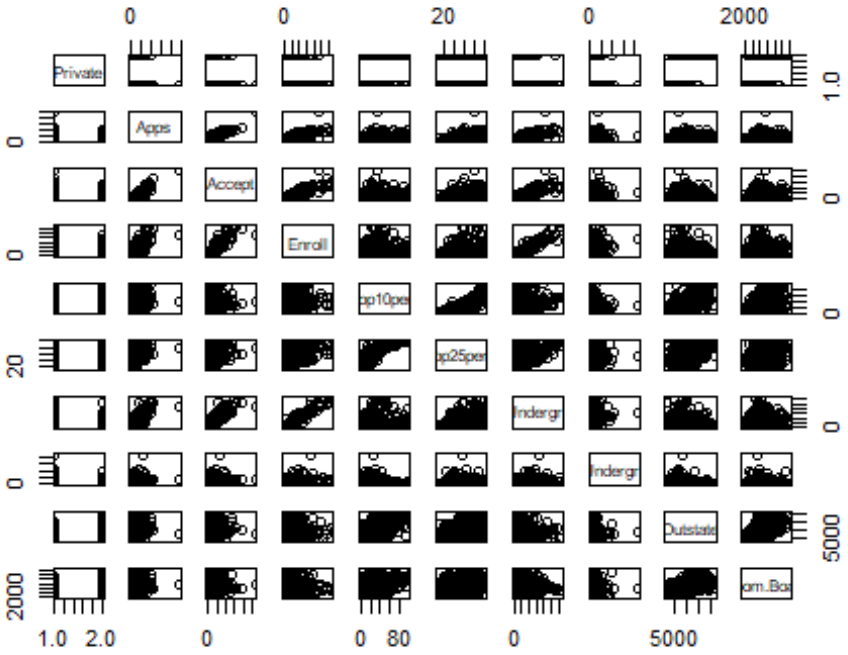
8.(c).i

```
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
```

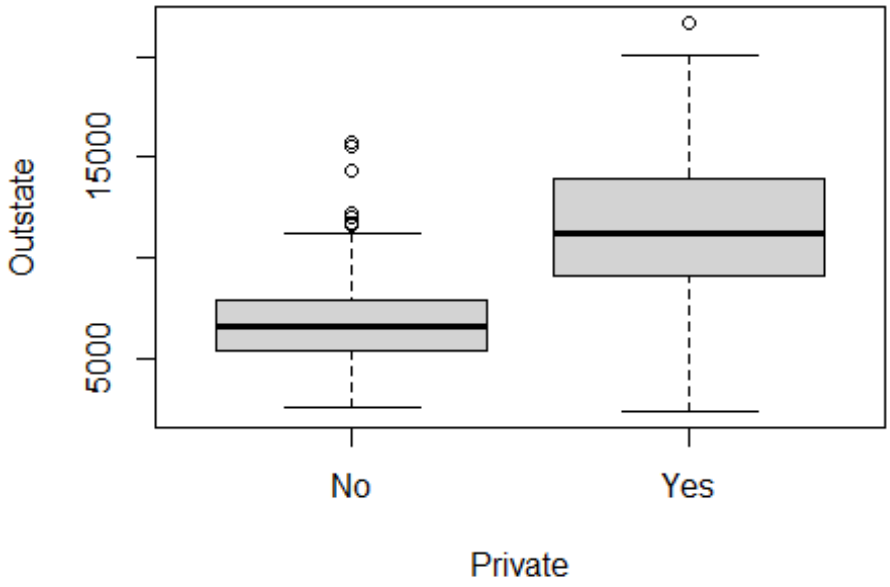
8.(c).ii

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



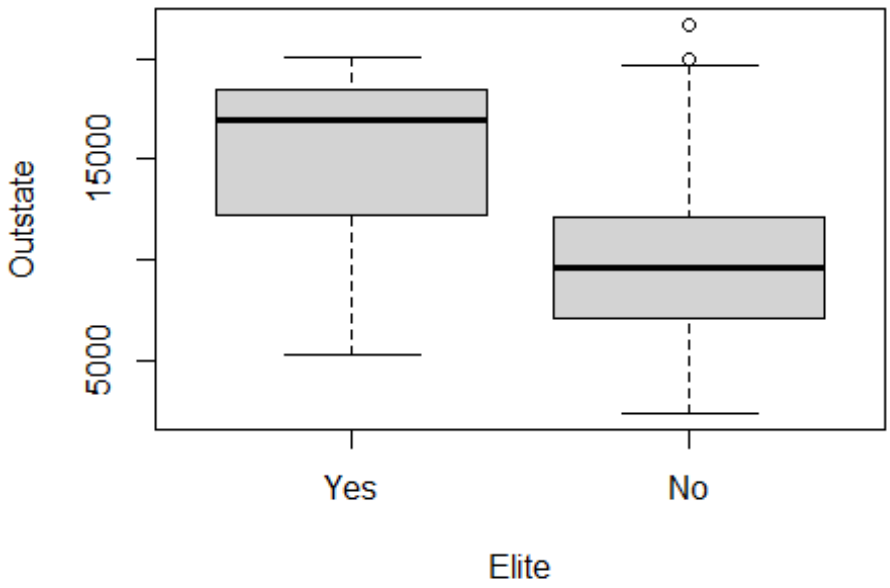
8.(c).iii

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



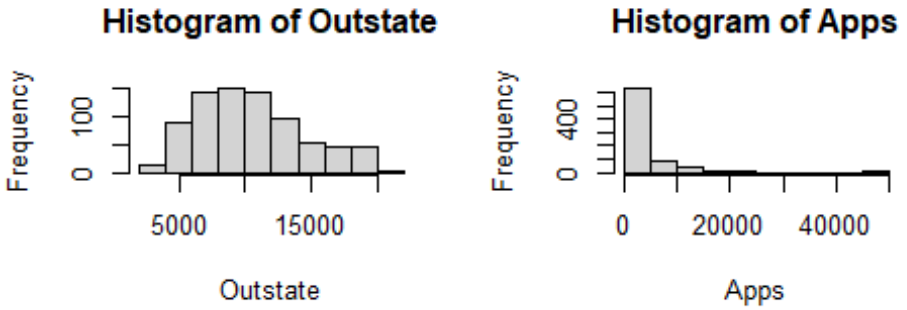
8.(c).iv

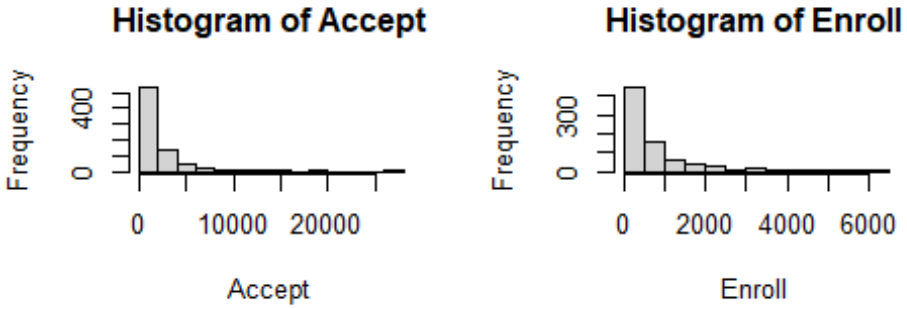
```
Elite <- rep("No", nrow (college))
Elite[college$Top10perc > 50] <- " Yes "
Elite <- as.factor(Elite)
college <- data.frame(college, Elite)
summary(Elite)

##  Yes     No
##    78   699

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



8.(c).v

```
par(mfrow = c(2, 2))
hist(Outstate)
hist(Apps)
hist(Accept)
hist(Enroll)
```





8.(c).vi

```
#garbage system
```

9.(a)

```
auto <- na.omit(Auto)
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   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)           :365

#name is qualitative and the rest is quantitative
```

9.(b)

```
for(c in (1:(ncol(auto)-1))){
  print(c(colnames(auto)[c], range(auto[, c])))
}
```

```
## [1] "mpg"  "9"    "46.6"
## [1] "cylinders" "3"         "8"
## [1] "displacement" "68"           "455"
## [1] "horsepower" "46"         "230"
## [1] "weight" "1613"   "5140"
## [1] "acceleration" "8"            "24.8"
## [1] "year" "70"   "82"
## [1] "origin" "1"      "3"
```

9.(c)

```
for(c in (1:(ncol(auto)-1))){
  print(c(colnames(auto)[c], mean(auto[, c]), sd(auto[, c])))
}
```

```
## [1] "mpg"              "23.4459183673469" "7.8050074865718"
## [1] "cylinders"        "5.4719387755102"  "1.70578324745278"
## [1] "displacement"     "194.411989795918" "104.644003908905"
## [1] "horsepower"       "104.469387755102" "38.4911599328285"
## [1] "weight"           "2977.58418367347" "849.402560042949"
## [1] "acceleration"     "15.5413265306122" "2.75886411918808"
## [1] "year"             "75.9795918367347" "3.68373654357783"
## [1] "origin"            "1.5765306122449"   "0.805518183418306"
```

9.(d)

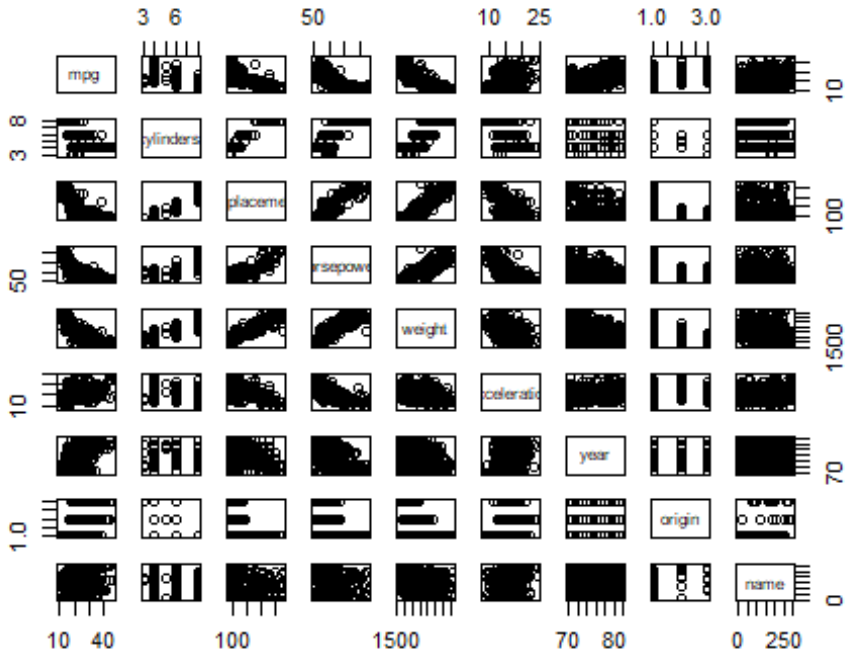
```
auto_adj <- auto[-c(10:85), ]
for(c in (1:(ncol(auto_adj)-1))){
  print(c(colnames(auto)[c], range(auto[, c]), mean(auto[, c]), sd(auto[,
c])))
}
```

```
## [1] "mpg"              "9"                "46.6"
"23.4459183673469"
## [5] "7.8050074865718"
## [1] "cylinders"        "3"                "8"
"5.4719387755102"
## [5] "1.70578324745278"
## [1] "displacement"     "68"               "455"
"194.411989795918"
```

```
## [5] "104.644003908905"
## [1] "horsepower"       "46"                "230"
"104.469387755102"
## [5] "38.4911599328285"
## [1] "weight"           "1613"              "5140"
"2977.58418367347"
## [5] "849.402560042949"
## [1] "acceleration"     "8"                 "24.8"
"15.5413265306122"
## [5] "2.75886411918808"
## [1] "year"             "70"                "82"
"75.9795918367347"
## [5] "3.68373654357783"
## [1] "origin"            "1"                  "3"
## [4] "1.5765306122449"   "0.805518183418306"
```

9.(e)

```
pairs(auto)
```




```
#garbage system
```

9.(f)

```
#except name, obvious pattern
```

10.(a)

```
library(ISLR2)
```

```
##
## Attaching package: 'ISLR2'
```

```
## The following objects are masked from 'package:ISLR':
##
##     Auto, Credit
```

```
head(Boston)
```

```
##      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
```

```
?Boston
```

```
## starting httpd help server ...
```

```
##  done
```

```
boston <- Boston
dim(boston)
```

```
## [1] 506  13
```

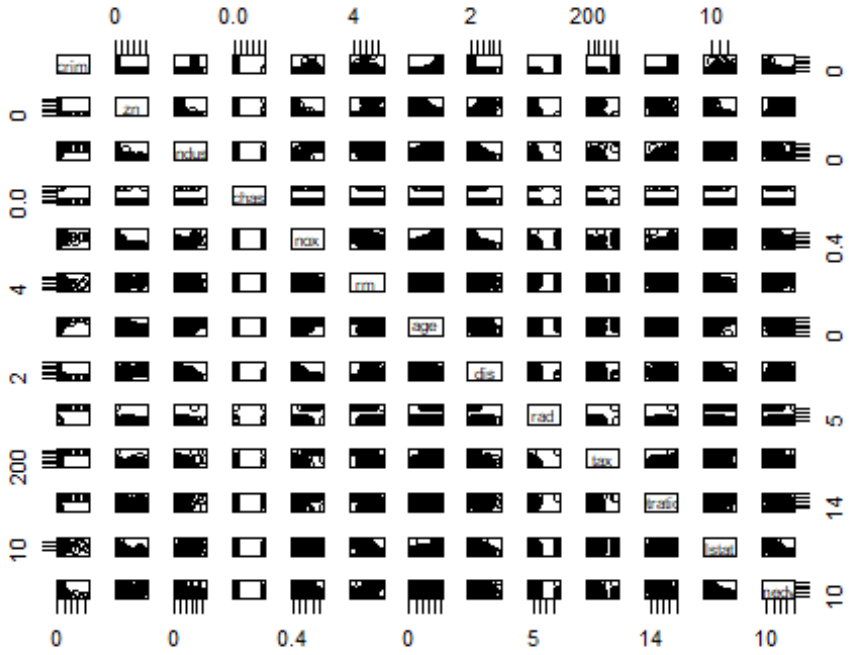
```
names(boston)
```

```
##  [1] "crim"    "zn"      "indus"   "chas"    "nox"     "rm"      "age"
##  [8] "dis"     "rad"     "tax"     "ptratio" "lstat"   "medv"
```

```
#each row is a suburb in Boston
```

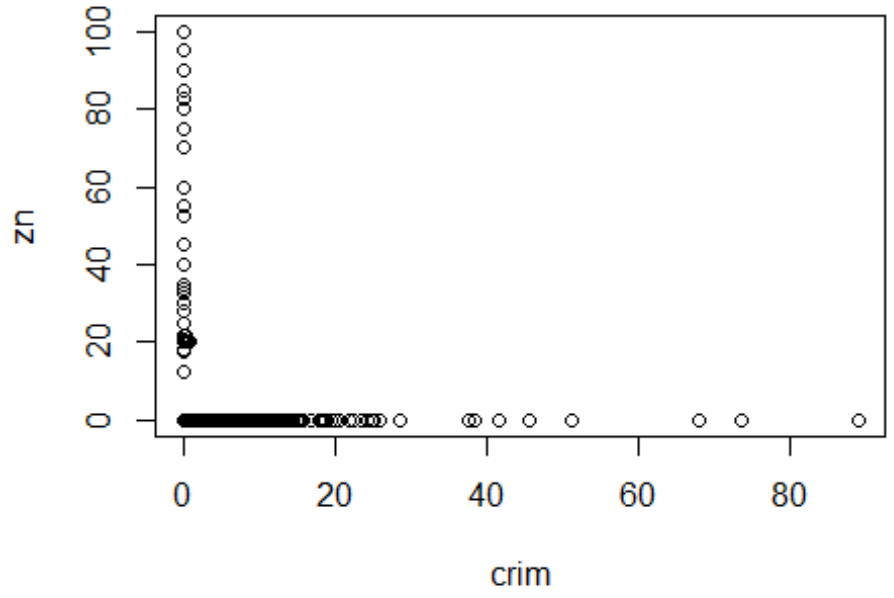
10.(b)

```
pairs(boston)
```



10.(c)

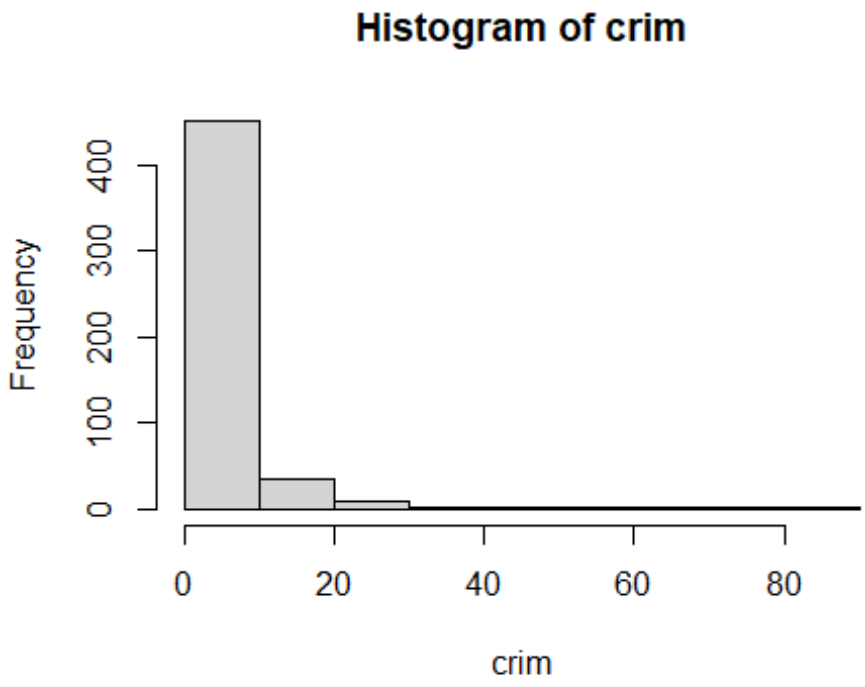
```
#all other predictors, e.g.
attach(boston)
plot(crim, zn)
```



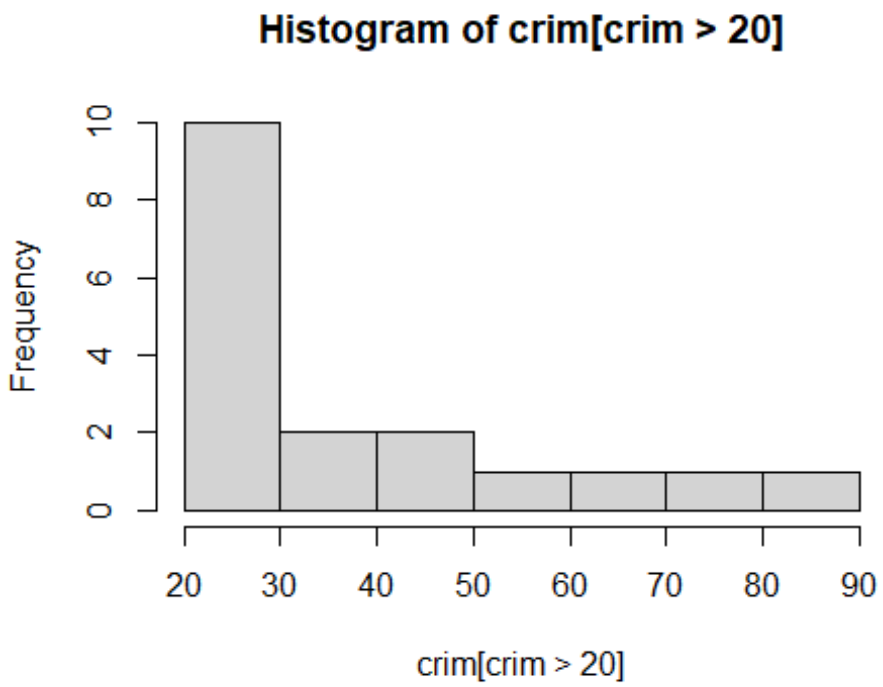
```
#only those without residential land zoned for lots over 25,000 sq.ft. has
crime
```

10.(d)

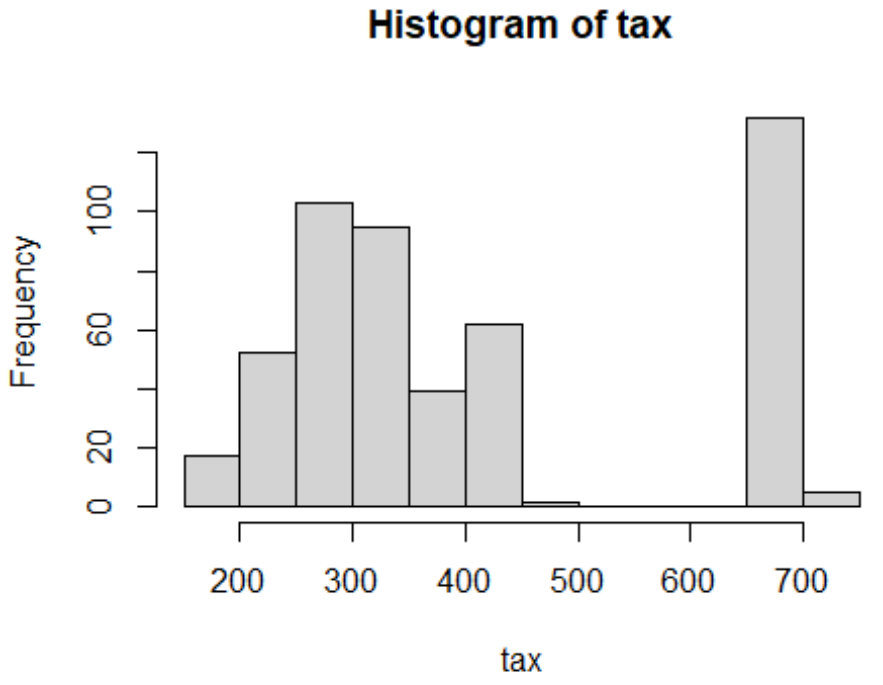
```
#yes, yes, no
hist(crim)
```



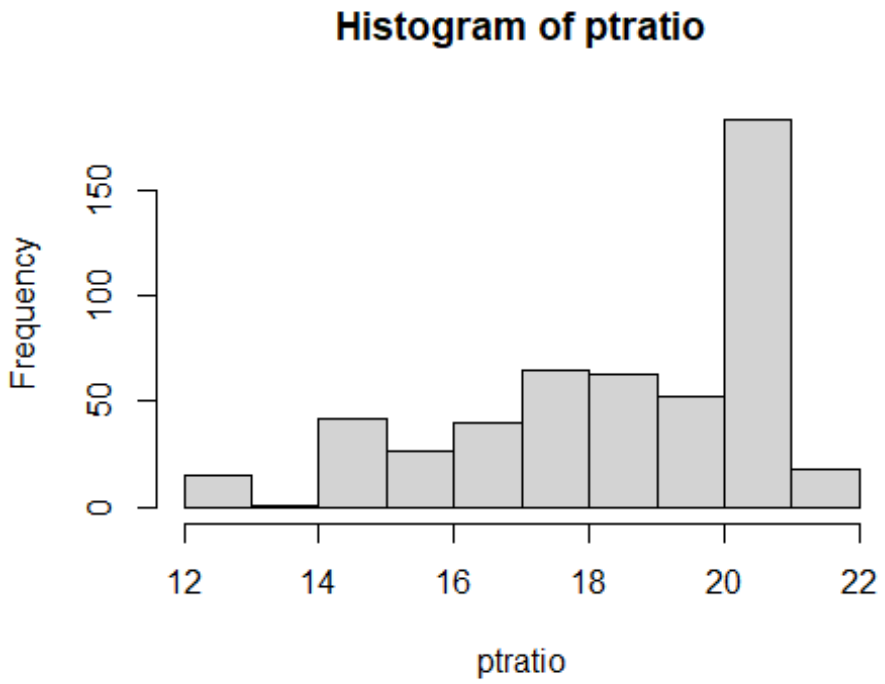

```
hist(crim[crim>20])
```




```
hist(tax)
```




```
hist(ptratio)
```



10.(e)

```
sum(chas)

## [1] 35
```

10.(f)

```
median(ptratio)

## [1] 19.05
```

10.(g)

```
boston[medv == min(medv), ]

##        crim zn indus chas   nox    rm age    dis rad tax ptratio lstat
medv
## 399 38.3518  0  18.1    0 0.693 5.453 100 1.4896  24 666    20.2 30.59
5
## 406 67.9208  0  18.1    0 0.693 5.683 100 1.4254  24 666    20.2 22.98
5
```

10.(h)

```
sum(rm > 7)

## [1] 64

sum(rm > 8)
```

```
## [1] 13
```

```
boston[rm > 8, ]
```

```
##        crim zn indus chas    nox    rm  age    dis rad tax ptratio lstat
medv
## 98  0.12083  0  2.89    0 0.4450 8.069 76.0 3.4952   2 276    18.0  4.21
38.7
## 164 1.51902  0 19.58    1 0.6050 8.375 93.9 2.1620   5 403    14.7  3.32
50.0
## 205 0.02009 95  2.68    0 0.4161 8.034 31.9 5.1180   4 224    14.7  2.88
50.0
## 225 0.31533  0  6.20    0 0.5040 8.266 78.3 2.8944   8 307    17.4  4.14
44.8
## 226 0.52693  0  6.20    0 0.5040 8.725 83.0 2.8944   8 307    17.4  4.63
50.0
## 227 0.38214  0  6.20    0 0.5040 8.040 86.5 3.2157   8 307    17.4  3.13
37.6
## 233 0.57529  0  6.20    0 0.5070 8.337 73.3 3.8384   8 307    17.4  2.47
41.7
## 234 0.33147  0  6.20    0 0.5070 8.247 70.4 3.6519   8 307    17.4  3.95
48.3
## 254 0.36894 22  5.86    0 0.4310 8.259  8.4 8.9067   7 330    19.1  3.54
42.8
## 258 0.61154 20  3.97    0 0.6470 8.704 86.9 1.8010   5 264    13.0  5.12
50.0
## 263 0.52014 20  3.97    0 0.6470 8.398 91.5 2.2885   5 264    13.0  5.91
48.8
## 268 0.57834 20  3.97    0 0.5750 8.297 67.0 2.4216   5 264    13.0  7.44
50.0
## 365 3.47428  0 18.10    1 0.7180 8.780 82.9 1.9047  24 666    20.2  5.29
21.9
```

```
#all between 8 and 9
```