HW2

Sandra Villamar and Shobhit Dronamraju

2023-01-26

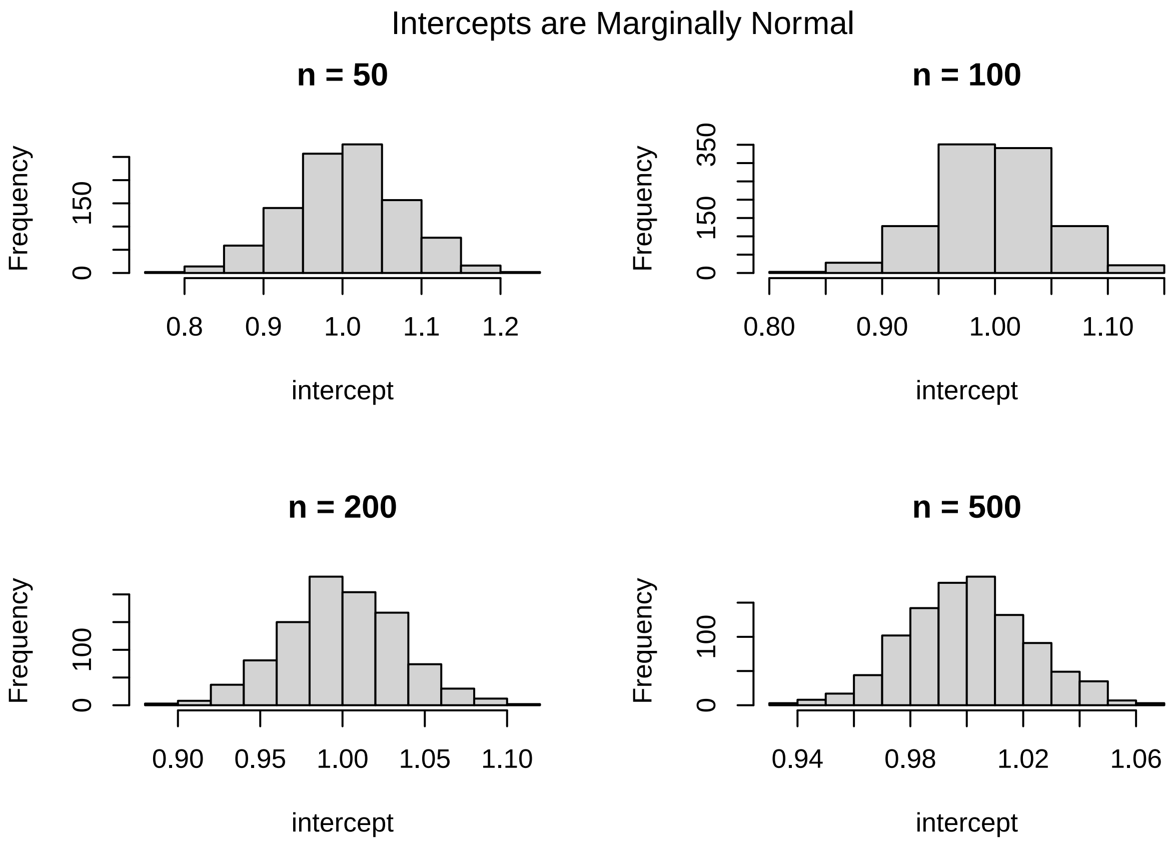
Problem 1A

```
library(hexbin)
library(gridExtra)
library(grid)
library(MASS)
library(car)
```

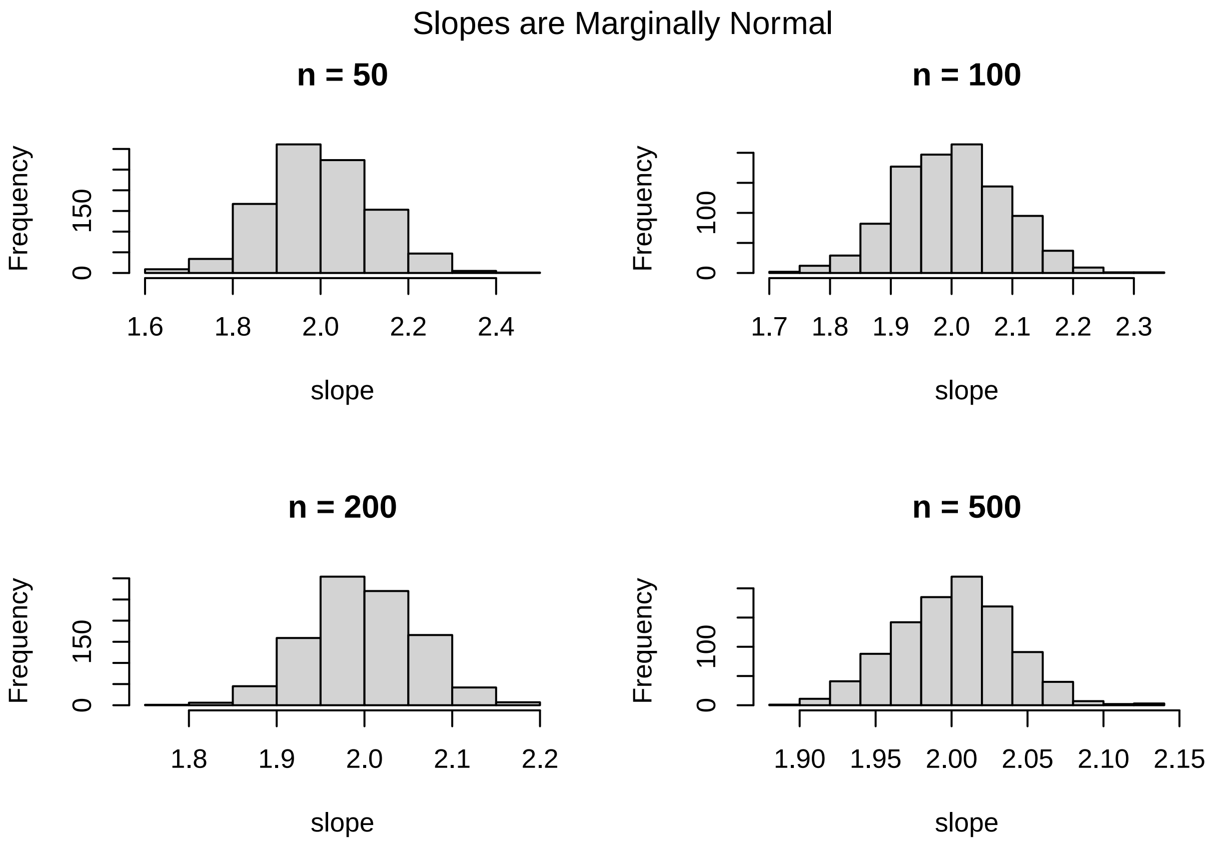
```
## Loading required package: carData
```

```
ns = c(50, 100, 200, 500)  # sample sizes
N = 1000  # trials
coefs = array(data = NA, dim = c(length(ns), N, 2))  # store intercept, slope

for(i in 1:length(ns)){
  n = ns[i]
  for(j in 1:N){
    x = runif(n, -1, 1)
    y = 1 + 2*x + rnorm(n, 0, 0.5)
    fit = lm(y ~ x)
    coefs[i, j,] = fit$coefficients
  }
}
```

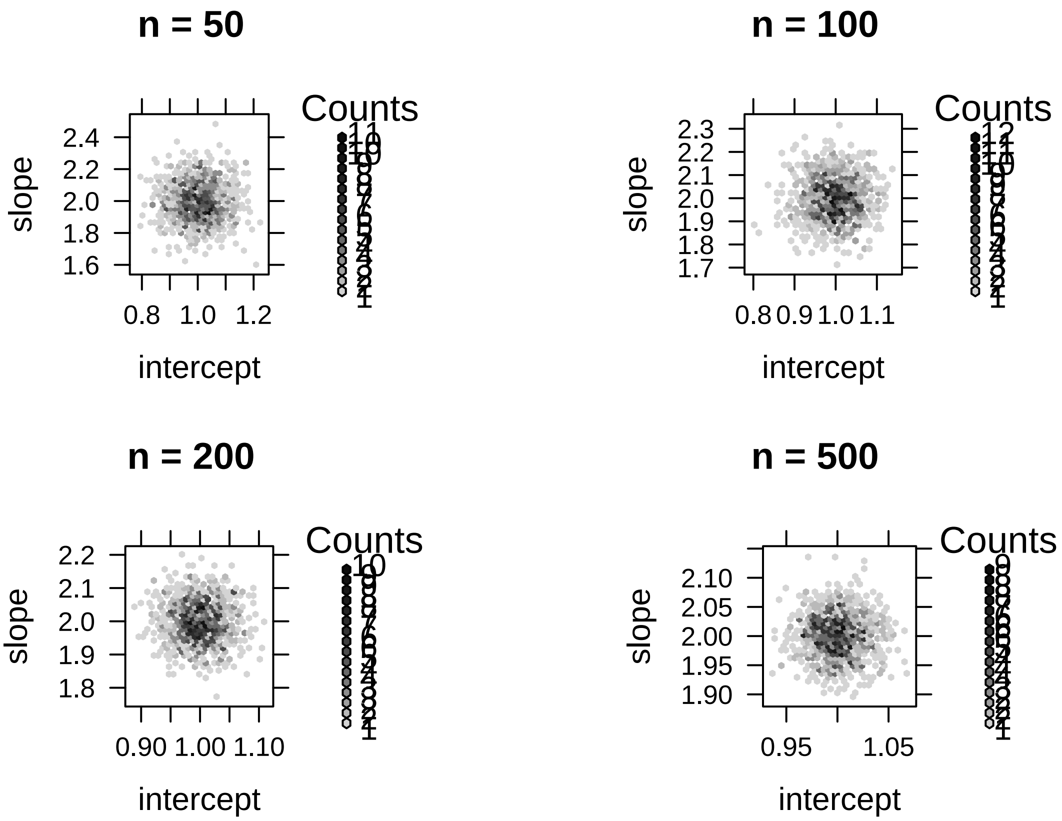
```
# intercepts marginally normal
par(mfrow=c(2,2))
for(i in 1:length(ns)){
  hist(coefs[i,,1], main=sprintf("n = %i", ns[i]), xlab="intercept")
}
mtext("Intercepts are Marginally Normal",
      side = 3,
      line = -1,
      outer = TRUE)
```



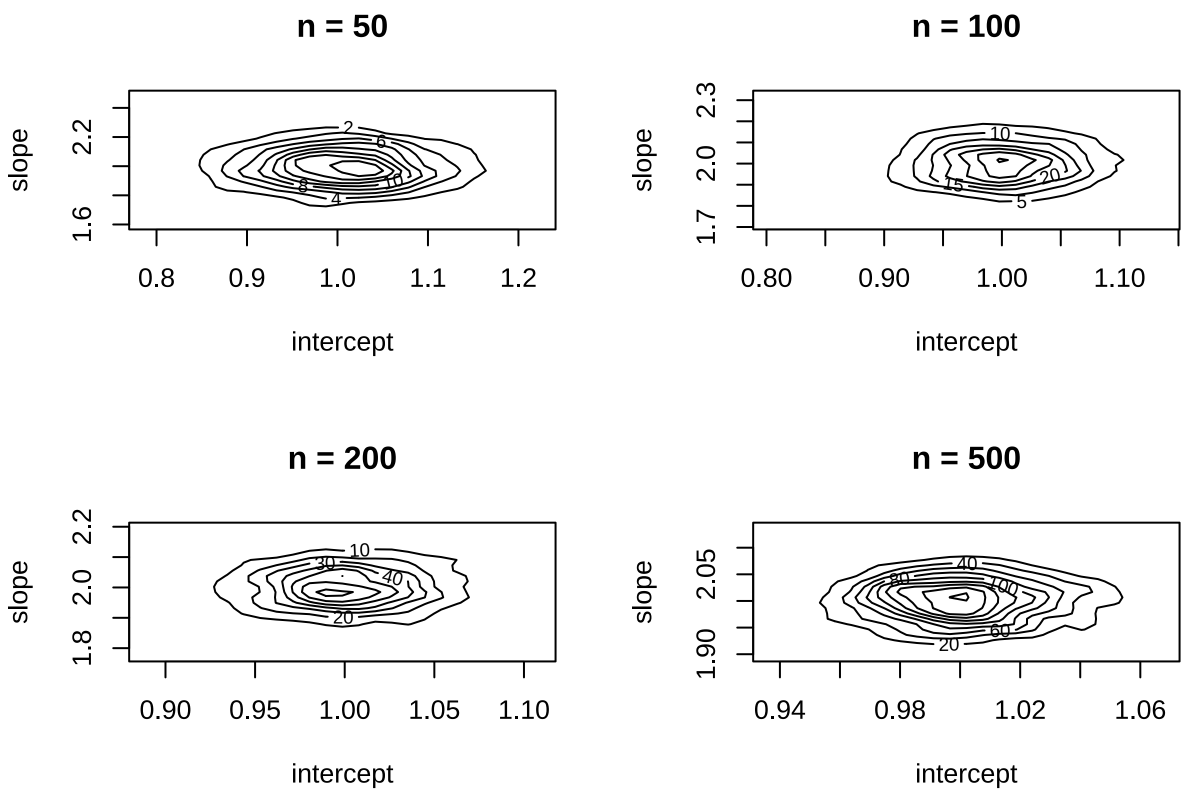
```
# slopes marginally normal
par(mfrow=c(2,2))
for(i in 1:length(ns)){
  hist(coefs[i,,2], main=sprintf("n = %i", ns[i]), xlab="slope")
}
mtext("Slopes are Marginally Normal",
      side = 3,
      line = -1,
      outer = TRUE)
```



```
# jointly normal

# hexbin
plotList <- lapply(1:length(ns), function(i) {
  intercept = coefs[i,,1]
  slope = coefs[i,,2]
  hexbinplot(slope ~ intercept, main=sprintf("n = %i", ns[i]), xlab="intercept", ylab="slope")
})

do.call(grid.arrange, c(plotList, ncol=2))
```



```
# plot level lines
par(mfrow=c(2,2))
for(i in 1:length(ns)){
  kde = kde2d(coefs[i, , 1], coefs[i, , 2])
  contour(kde, main = sprintf("n = %i", ns[i]), xlab = "intercept", ylab = "slope")
}
```

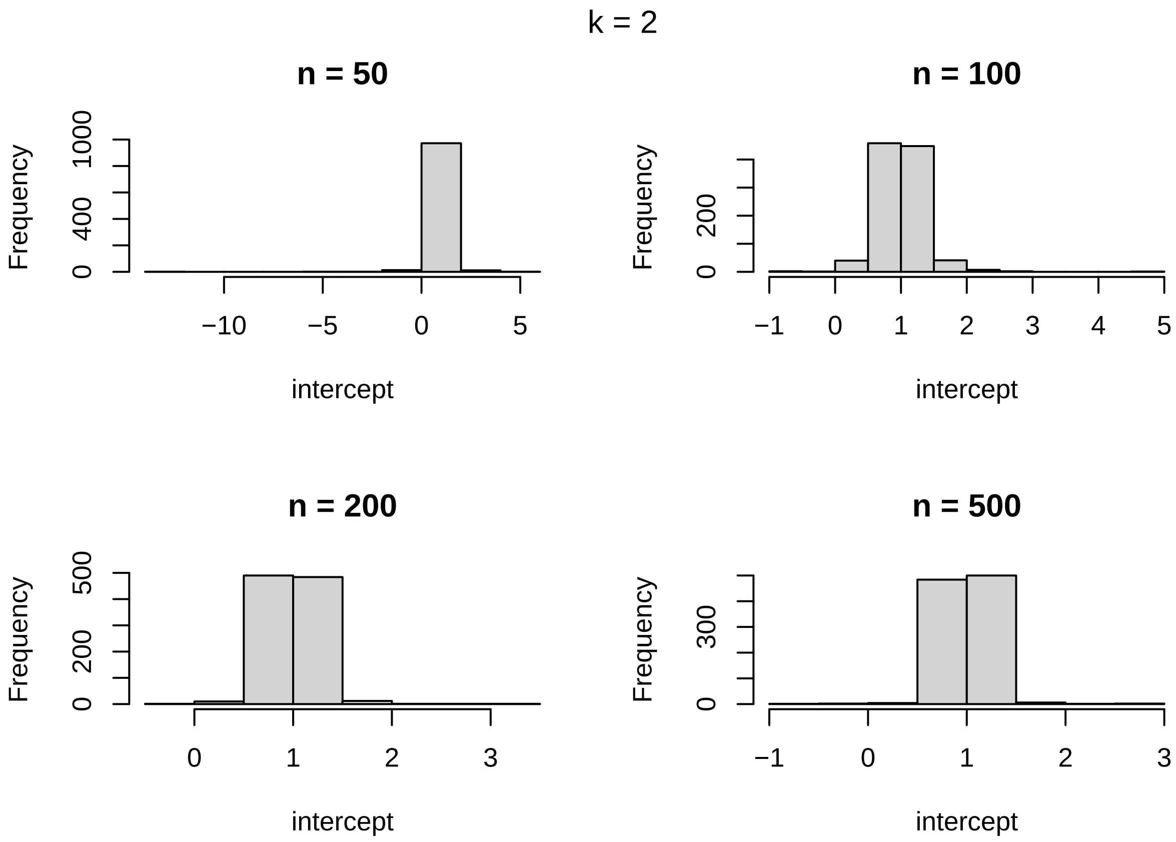


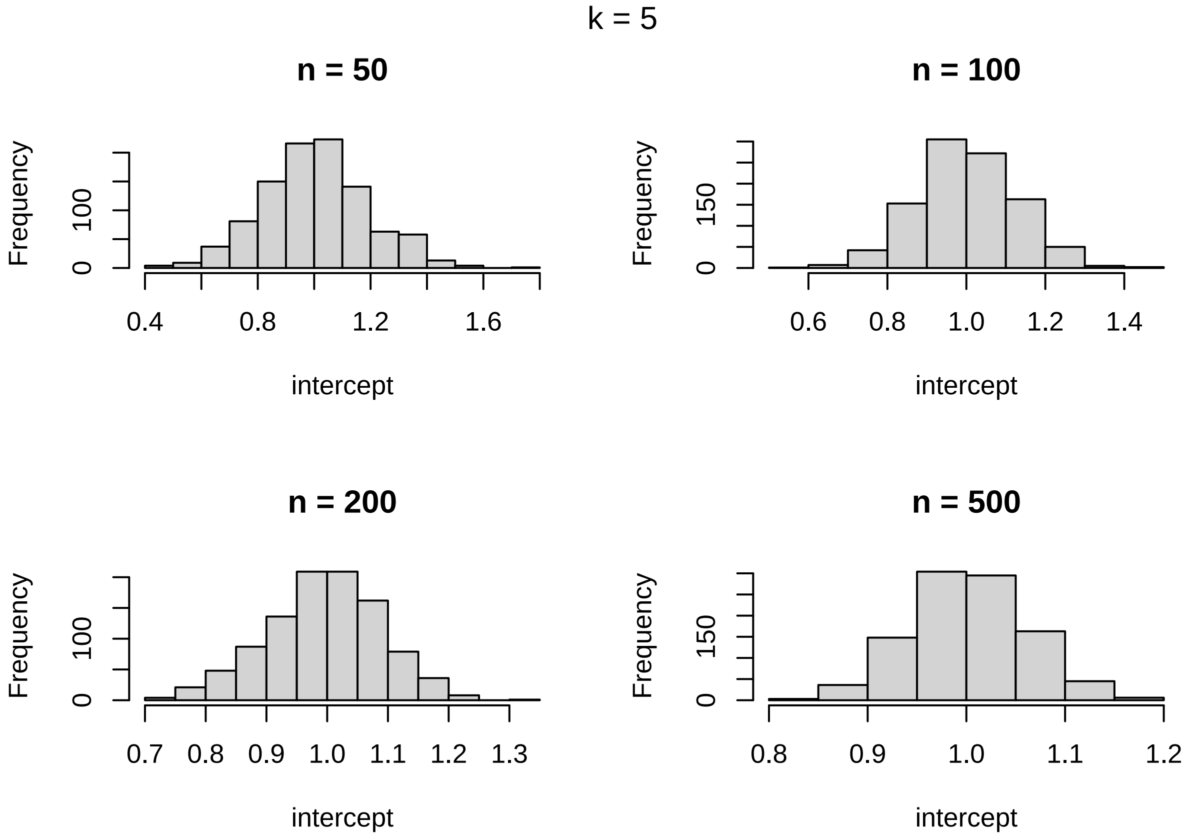
From the plots above, we can clearly see that the slope and intercept histograms display a normal distribution, and thus are both marginally normal. From the level lines plot, we see that the slope and intercept are jointly normal because each plot has an ellipse shape with the highest level line value at the center and decreasing level lines as points are farther away from the center.

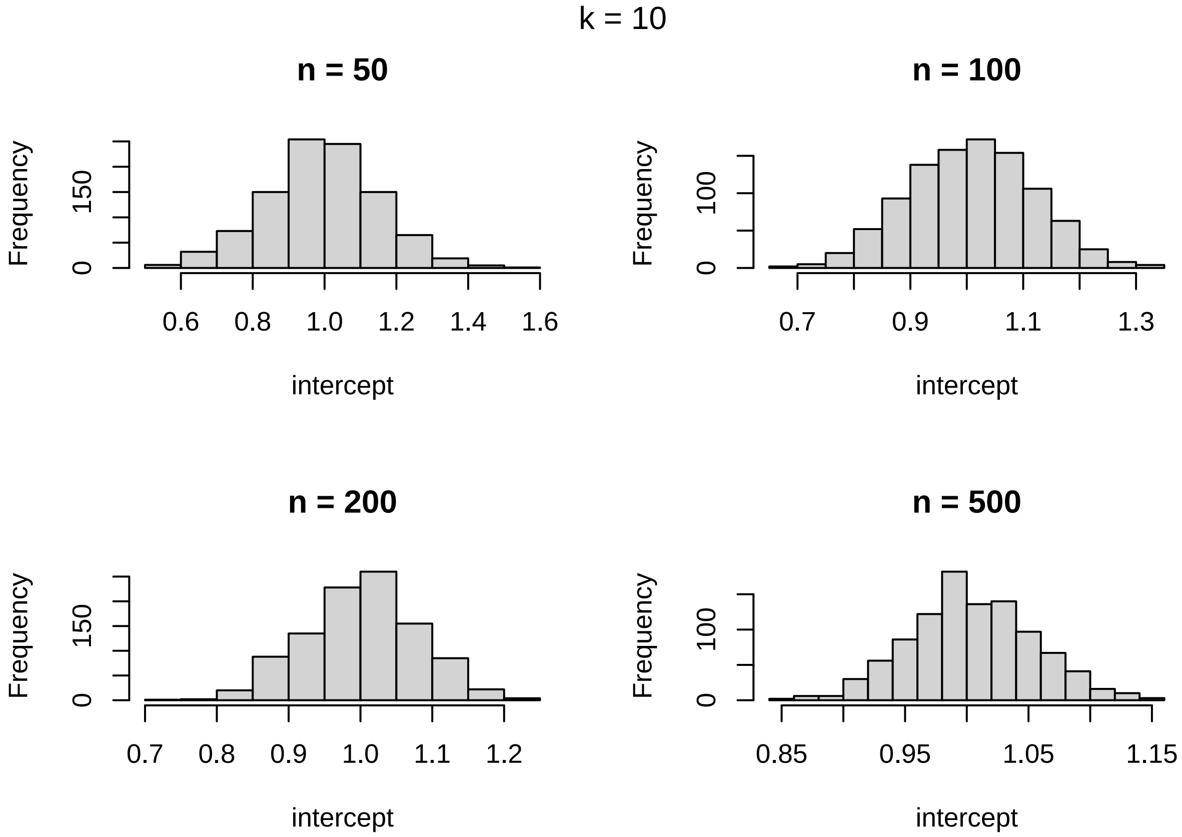
Problem 1B

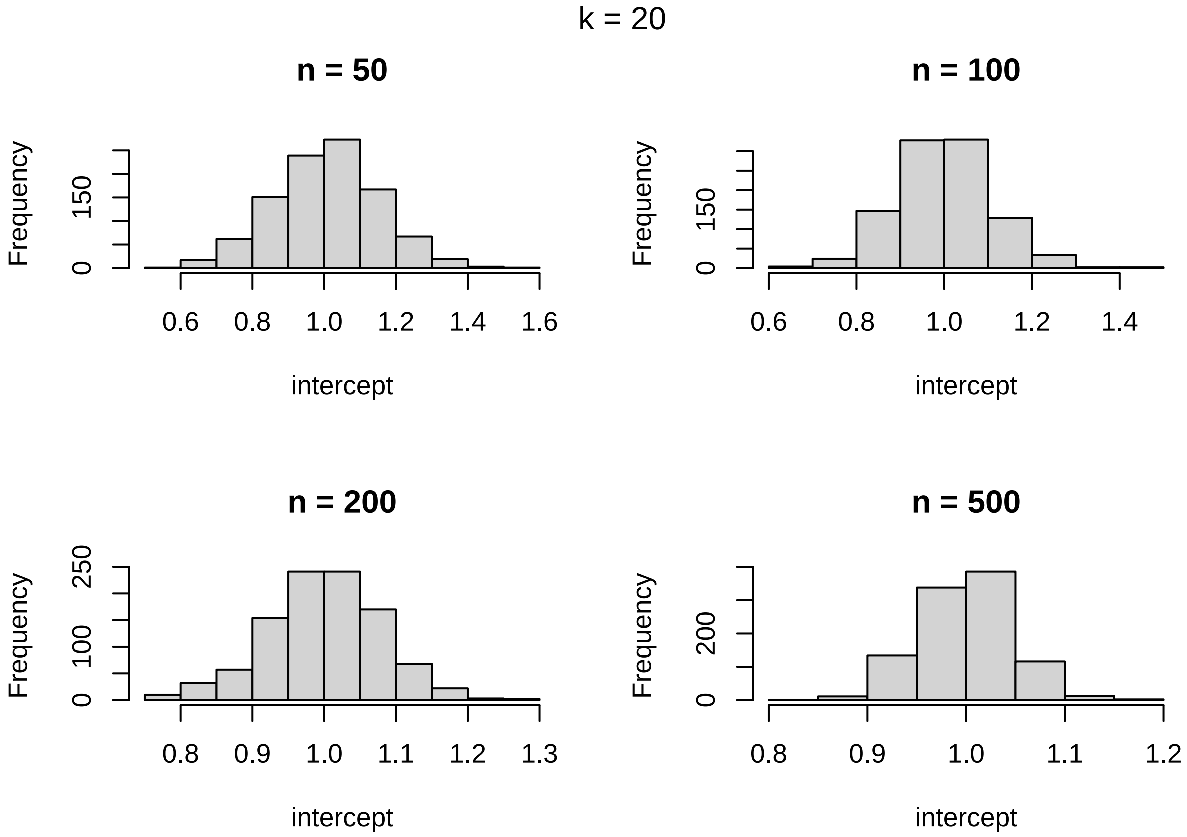
```r
ks = c(2, 5, 10, 20, 50)
coefs2 = array(data = NA, dim = c(length(ks), length(ns), N, 2))

for(h in 1:length(ks)){
  k = ks[h]

  for(i in 1:length(ns)){
    n = ns[i]

    for(j in 1:N){
      x = runif(n, -1, 1)
      y = 1 + 2*x + rt(n, df=k)
      fit = lm(y ~ x)
      coefs2[h, i, j,] = fit$coefficients
    }
  }
}
```

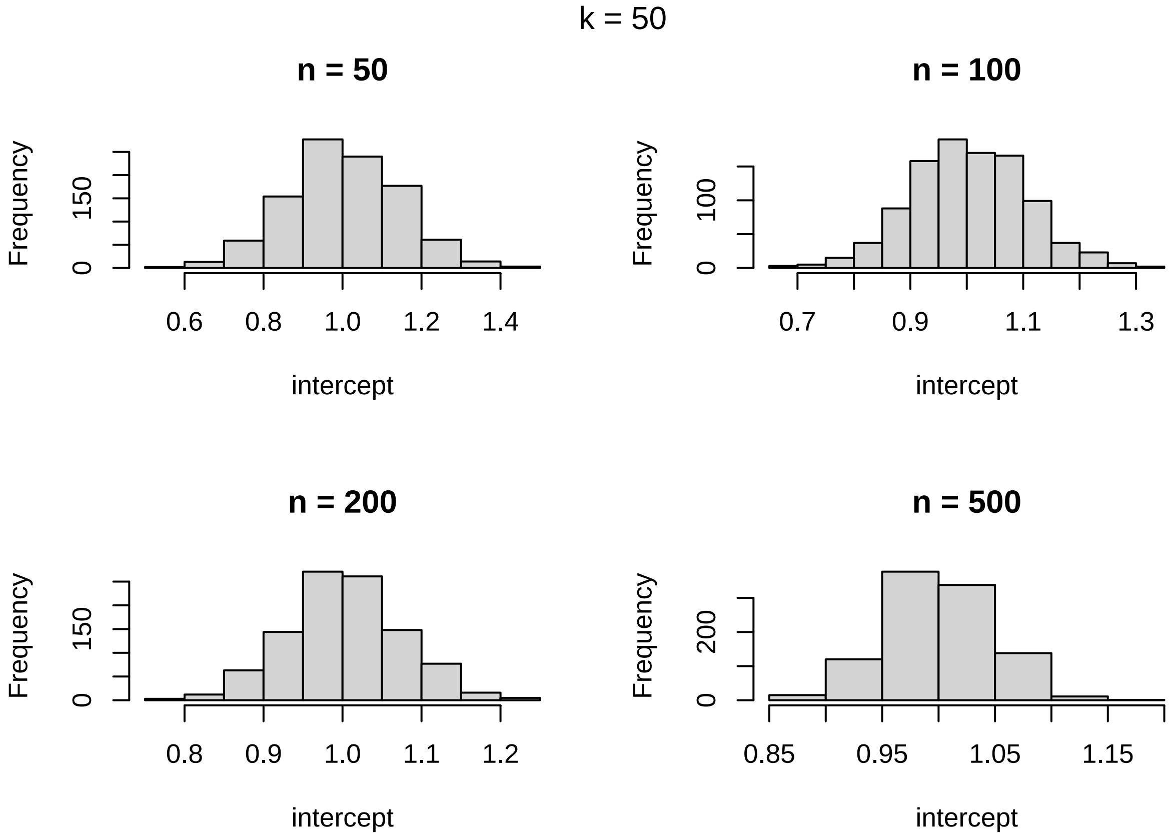
```r
# intercepts
for(h in 1:length(ks)){
  par(mfrow=c(2,2))
  for(i in 1:length(ns)){
    hist(coefs2[h, i, ,1], main=sprintf("n = %i", ns[i]), xlab="intercept")
  }
  mtext(sprintf("k = %i", ks[h]),
        side = 3,
        line = -1,
        outer = TRUE)
}
```



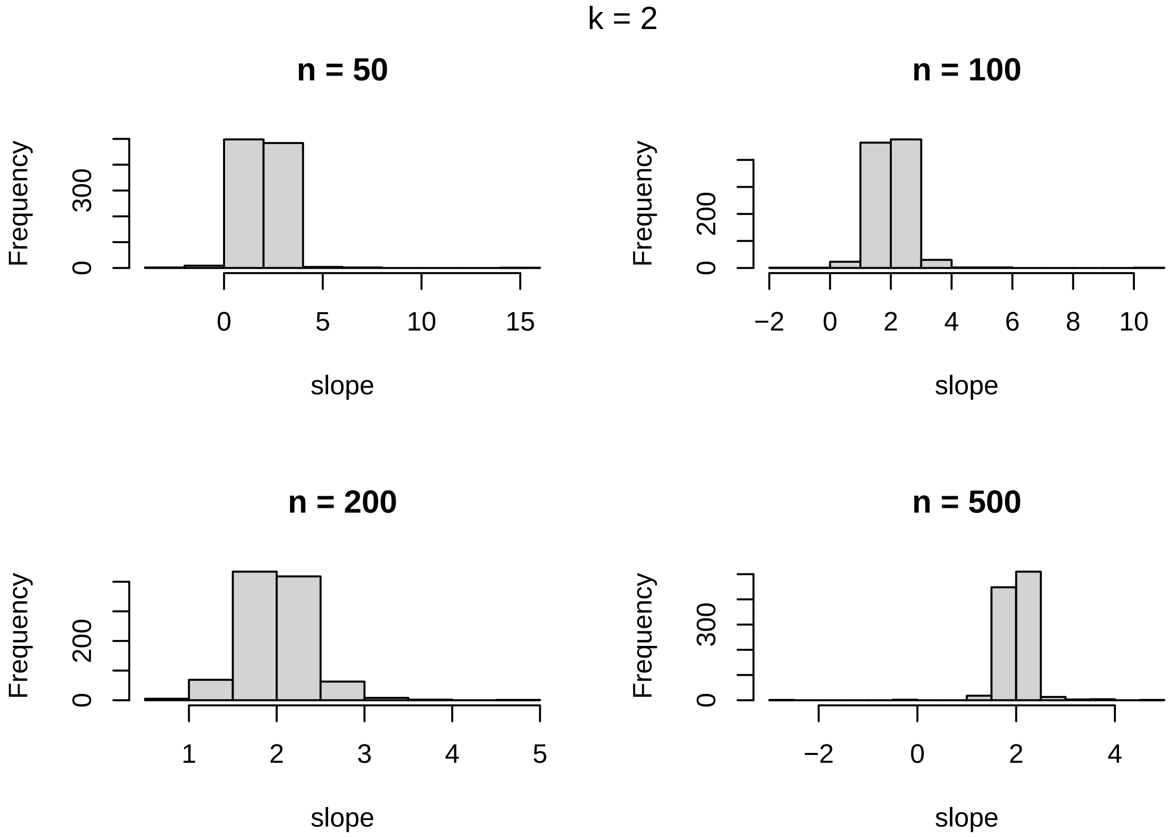

k = 5
n = 50
Frequency
0
100
0.4
0.8
1.2
1.6
intercept
n = 100
Frequency
0
150
0.6
0.8
1.0
1.2
1.4
intercept
n = 200
Frequency
0
100
0.7
0.8
0.9
1.0
1.1
1.2
1.3
intercept
n = 500
Frequency
0
150
0.8
0.9
1.0
1.1
1.2
intercept

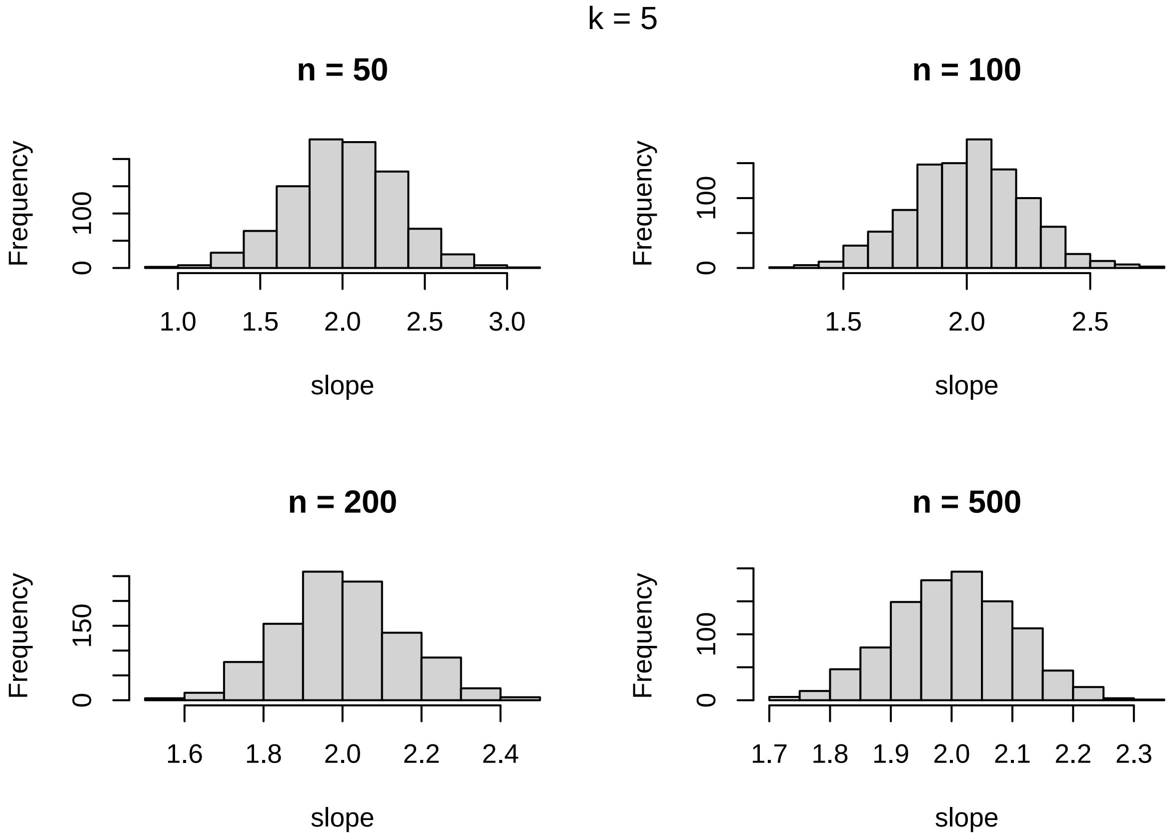

k = 10
n = 50
Frequency
0
150
0.6
0.8
1.0
1.2
1.4
1.6
intercept
n = 100
Frequency
0
100
0.7
0.9
1.1
1.3
intercept
n = 200
Frequency
0
150
0.7
0.8
0.9
1.0
1.1
1.2
intercept
n = 500
Frequency
0
100
0.85
0.95
1.05
1.15
intercept

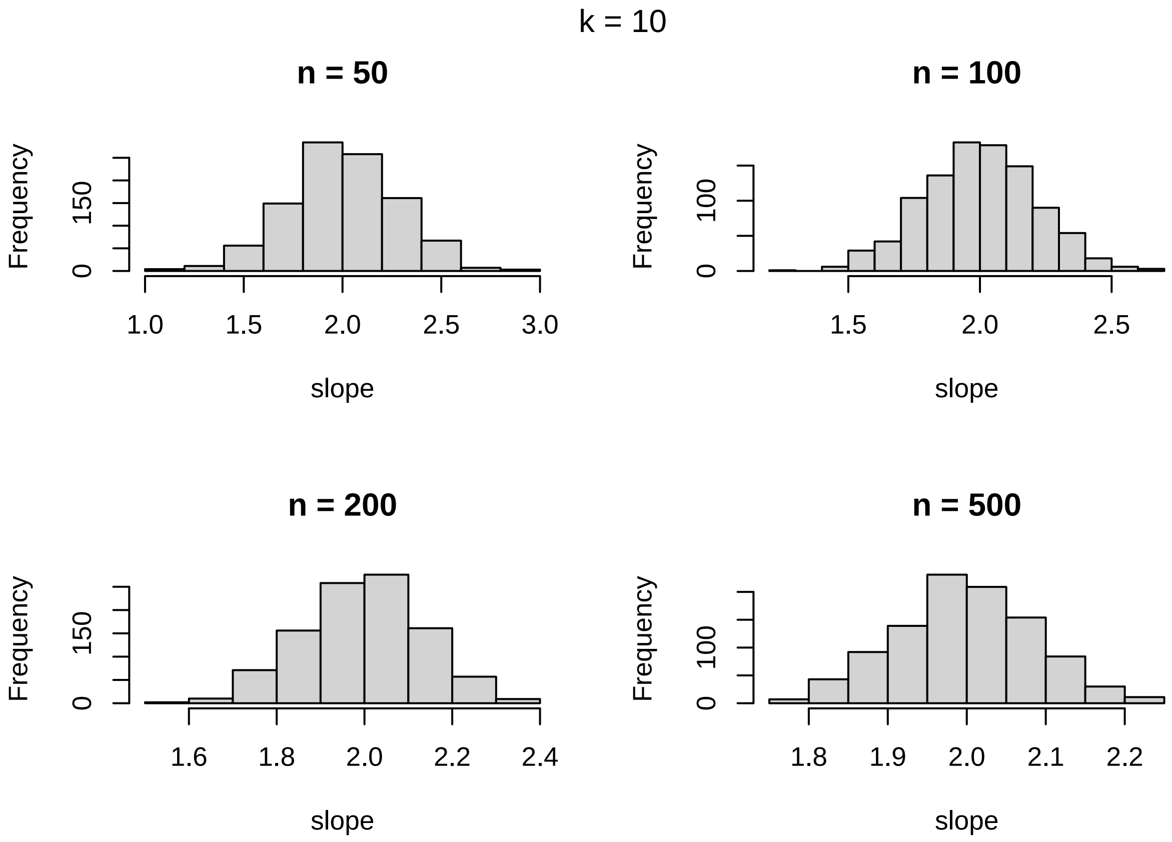

k = 20
n = 50
Frequency
0
150
0.6
0.8
1.0
1.2
1.4
1.6
intercept
n = 100
Frequency
0
150
0.6
0.8
1.0
1.2
1.4
intercept
n = 200
Frequency
0
100
250
0.8
0.9
1.0
1.1
1.2
1.3
intercept
n = 500
Frequency
0
200
0.8
0.9
1.0
1.1
1.2
intercept

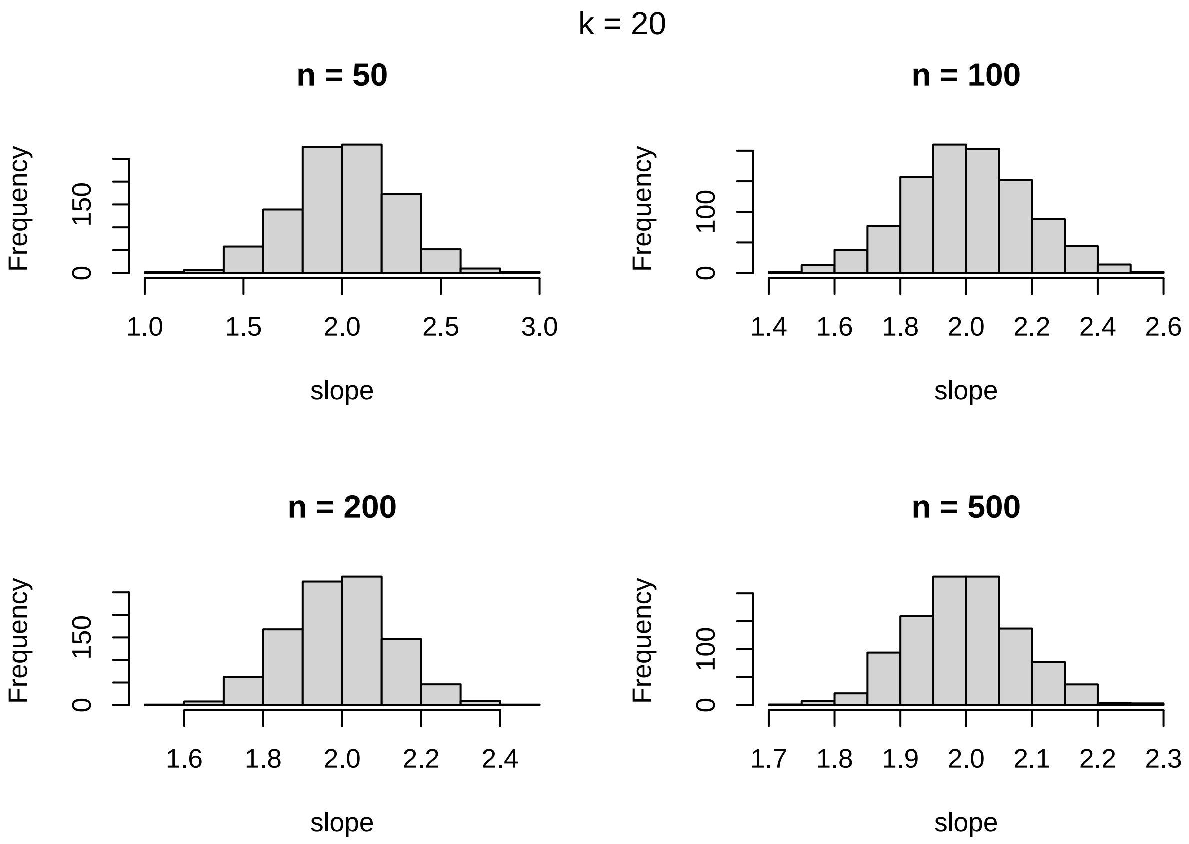


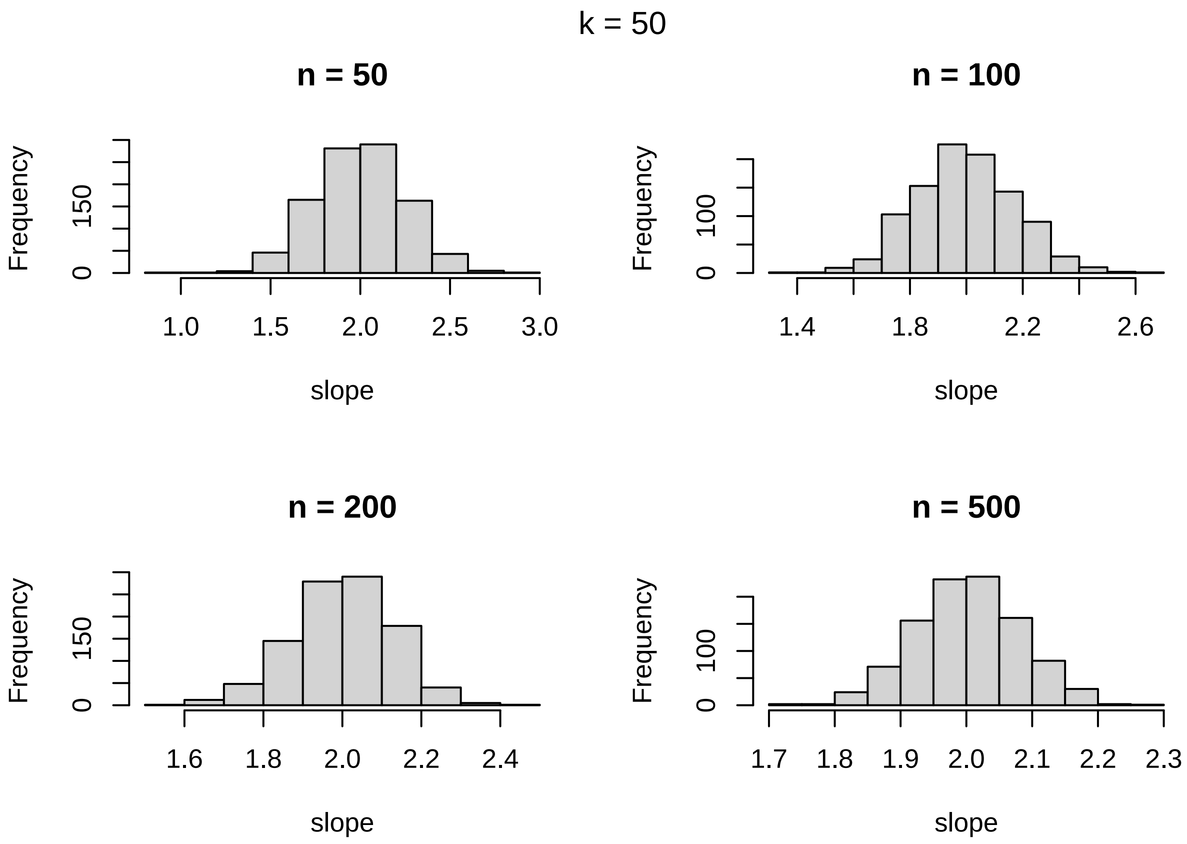
```
# slopes
for(h in 1:length(ks)){
  par(mfrow=c(2,2))
  for(i in 1:length(ns)){
    hist(coefs2[h, i, ,2], main=sprintf("n = %i", ns[i]), xlab="slope")
  }
  mtext(sprintf("k = %i", ks[h]),
        side = 3,
        line = -1,
        outer = TRUE)
}
```


k = 2
n = 50
Frequency
0
300
0
5
10
15
slope
n = 100
Frequency
0
200
−2
0
2
4
6
8
10
slope
n = 200
Frequency
0
200
1
2
3
4
5
slope
n = 500
Frequency
0
300
−2
0
2
4
slope

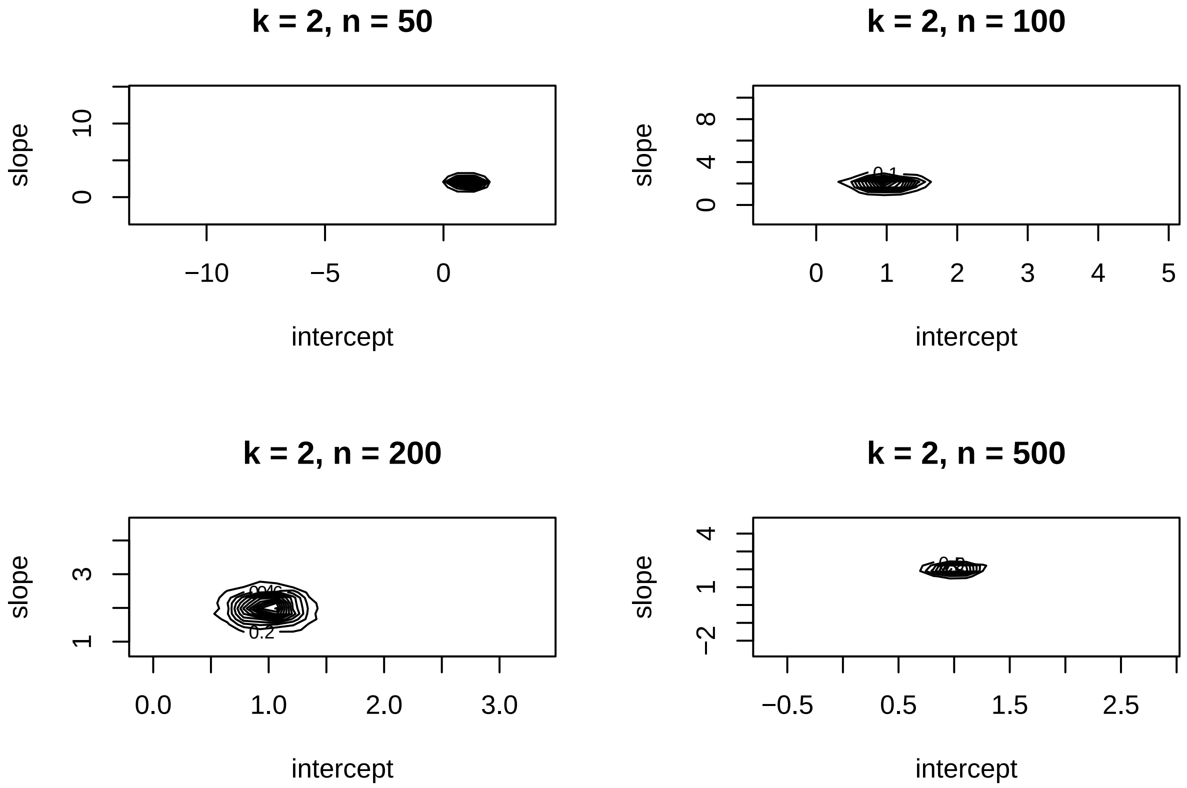

k = 5
n = 50
Frequency
0
100
1.0
1.5
2.0
2.5
3.0
slope
n = 100
Frequency
0
100
1.5
2.0
2.5
slope
n = 200
Frequency
0
150
1.6
1.8
2.0
2.2
2.4
slope
n = 500
Frequency
0
100
1.7
1.8
1.9
2.0
2.1
2.2
2.3
slope

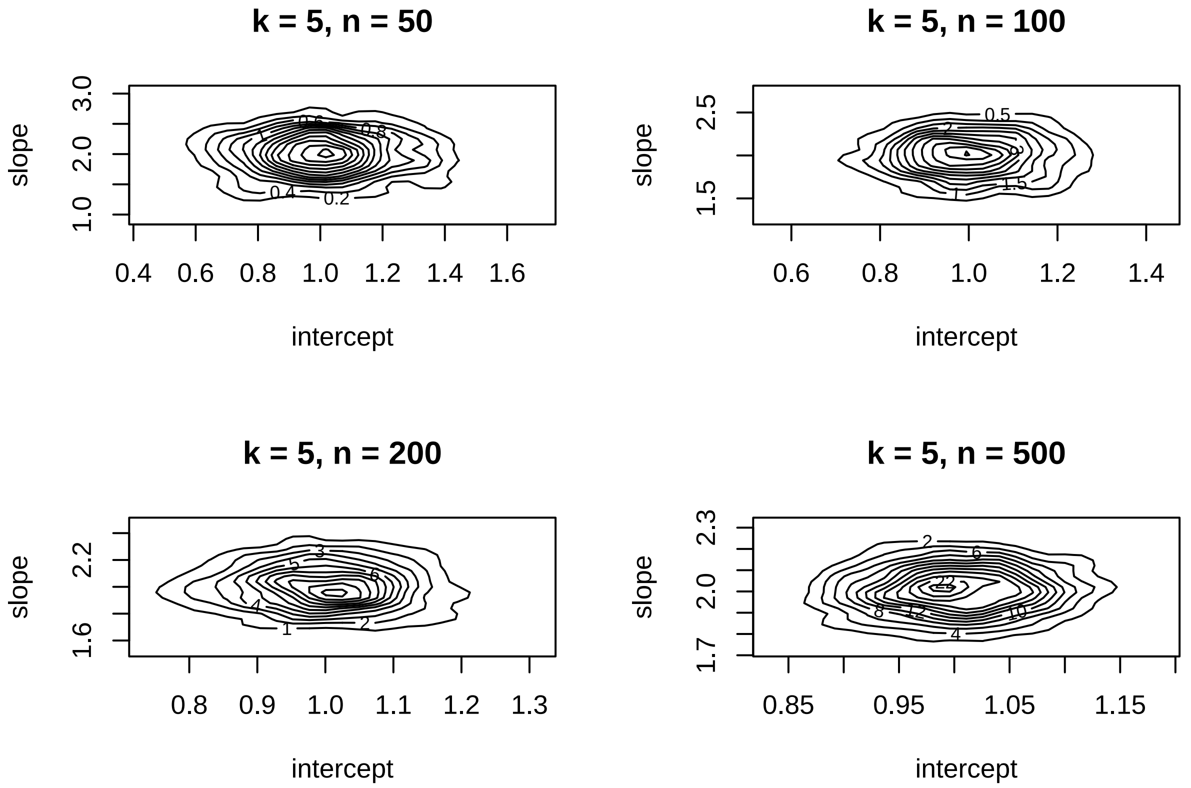

k = 10
n = 50
Frequency
0
150
1.0
1.5
2.0
2.5
3.0
slope
n = 100
Frequency
0
100
1.5
2.0
2.5
slope
n = 200
Frequency
0
150
1.6
1.8
2.0
2.2
2.4
slope
n = 500
Frequency
0
100
1.8
1.9
2.0
2.1
2.2
slope

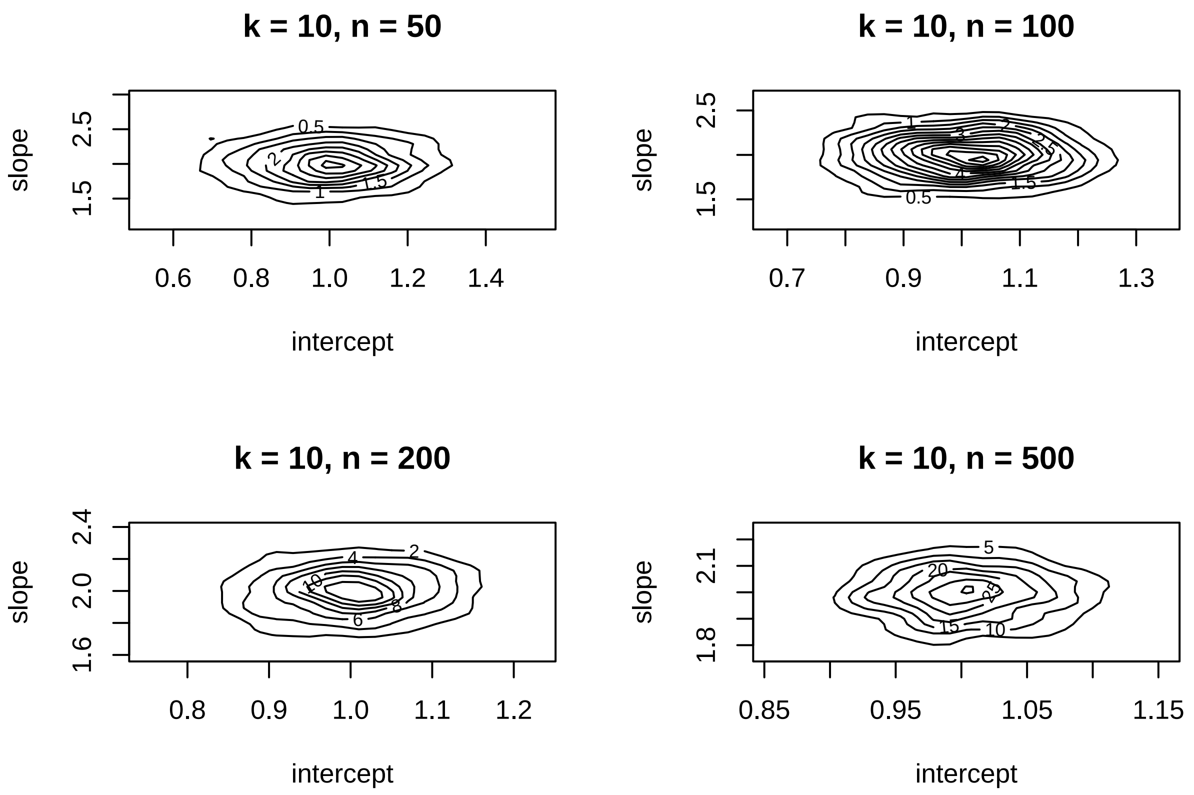

k = 20
n = 50
Frequency
0
150
1.0
1.5
2.0
2.5
3.0
slope
n = 100
Frequency
0
100
1.4
1.6
1.8
2.0
2.2
2.4
2.6
slope
n = 200
Frequency
0
150
1.6
1.8
2.0
2.2
2.4
slope
n = 500
Frequency
0
100
1.7
1.8
1.9
2.0
2.1
2.2
2.3
slope

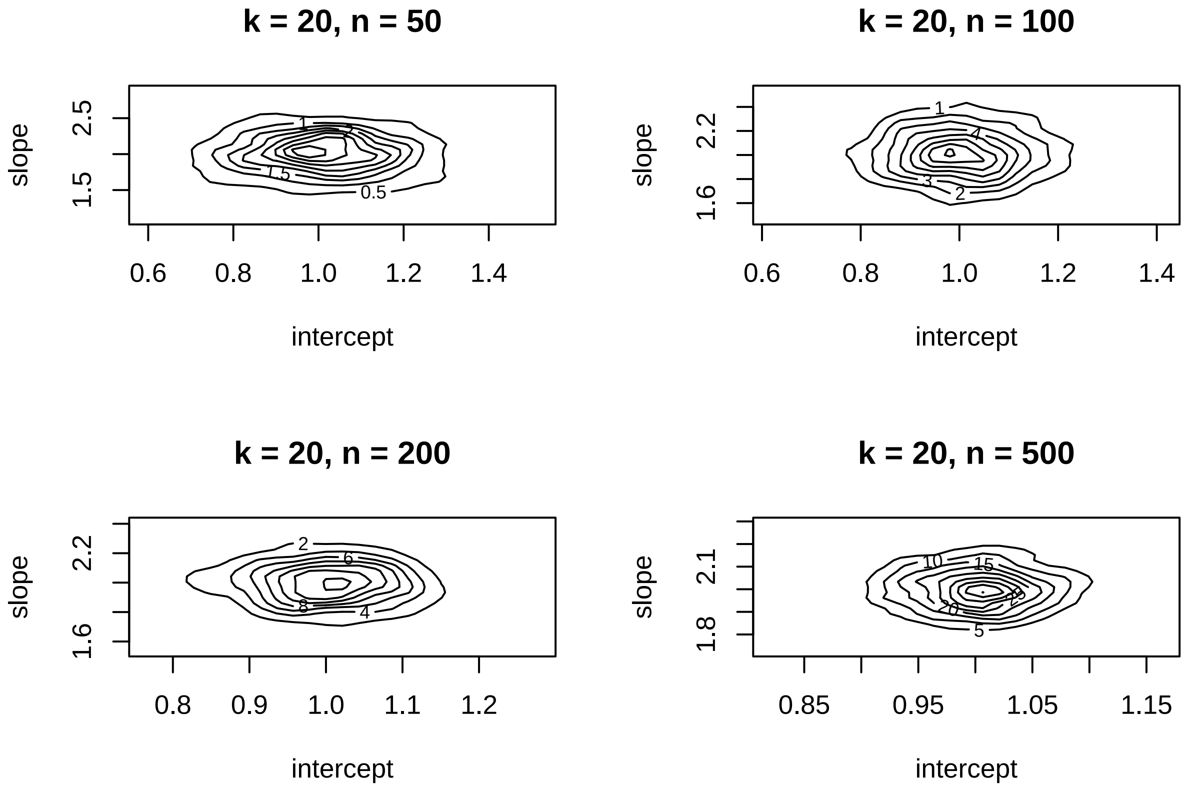


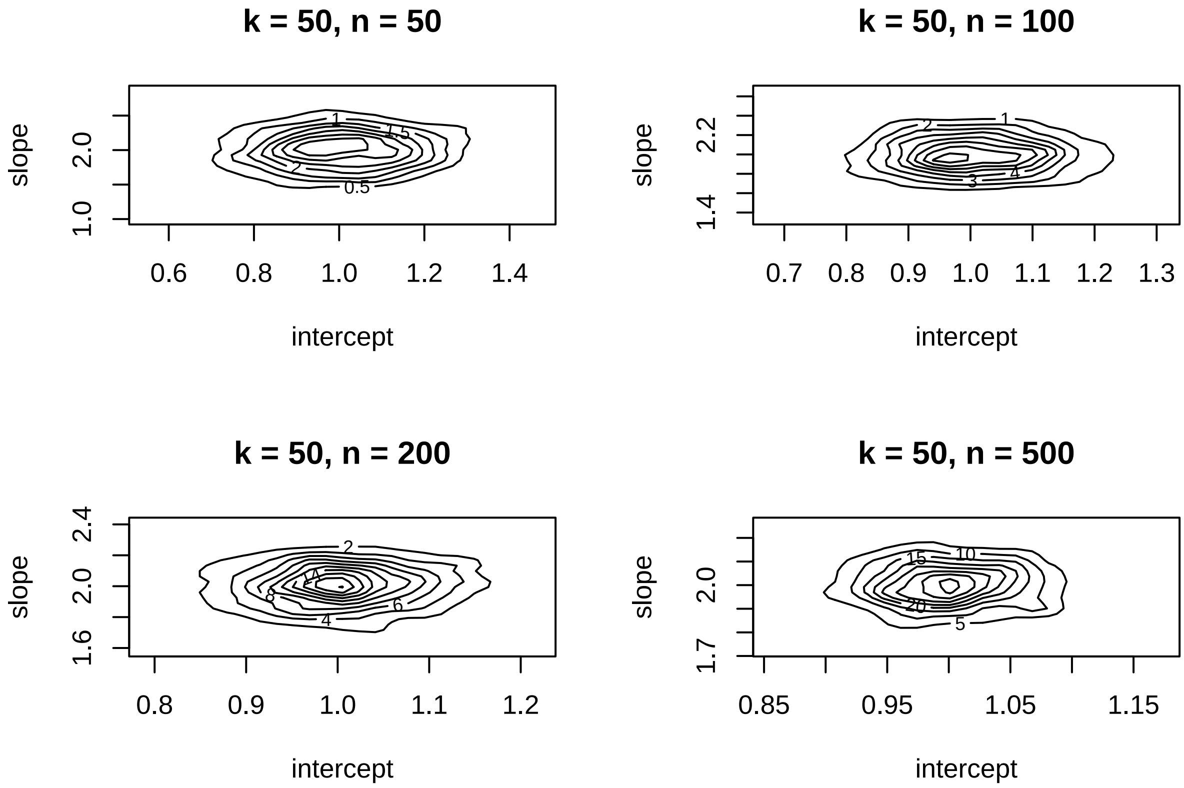
```
# jointly normal: plot level lines

par(mfrow=c(2,2))
for(h in 1:length(ks)){
  for(i in 1:length(ns)){
    kde = kde2d(coefs2[h, i, , 1], coefs2[h, i, ,2])
    contour(kde, main = sprintf("k = %i, n = %i", ks[h], ns[i]),
            xlab = "intercept", ylab = "slope")
  }
}
```


k = 2, n = 50
slope
0
10
intercept
−10
−5
0
k = 2, n = 100
slope
0
4
8
0.1
intercept
0
1
2
3
4
5
k = 2, n = 200
slope
1
3
0.2
intercept
0.0
1.0
2.0
3.0
k = 2, n = 500
slope
−2
1
4
intercept
−0.5
0.5
1.5
2.5


k = 5, n = 50
slope
intercept
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.0
2.0
3.0
0.6
0.8
1
0.4
0.2
k = 5, n = 100
slope
intercept
0.6
0.8
1.0
1.2
1.4
1.5
2.5
0.5
2
1.5
1
k = 5, n = 200
slope
intercept
0.8
0.9
1.0
1.1
1.2
1.3
1.6
2.2
3
5
6
4
1
2
k = 5, n = 500
slope
intercept
0.85
0.95
1.05
1.15
1.7
2.0
2.3
2
6
22
8
12
10
4


k = 10, n = 50
slope
1.5
2.5
0.6
0.8
1.0
1.2
1.4
intercept
0.5
2
1
1.5
k = 10, n = 100
slope
1.5
2.5
0.7
0.9
1.1
1.3
intercept
1
3
2
2.5
4
1.5
0.5
k = 10, n = 200
slope
1.6
2.0
2.4
0.8
0.9
1.0
1.1
1.2
intercept
4
2
10
6
8
k = 10, n = 500
slope
1.8
2.1
0.85
0.95
1.05
1.15
intercept
5
20
25
15
10


k = 20, n = 50
slope
2.5
1.5
intercept
0.6
0.8
1.0
1.2
1.4
1
2
1.5
0.5
k = 20, n = 100
slope
2.2
1.6
intercept
0.6
0.8
1.0
1.2
1.4
1
4
3
2
k = 20, n = 200
slope
2.2
1.6
intercept
0.8
0.9
1.0
1.1
1.2
2
6
8
4
k = 20, n = 500
slope
2.1
1.8
intercept
0.85
0.95
1.05
1.15
10
15
20
25
5



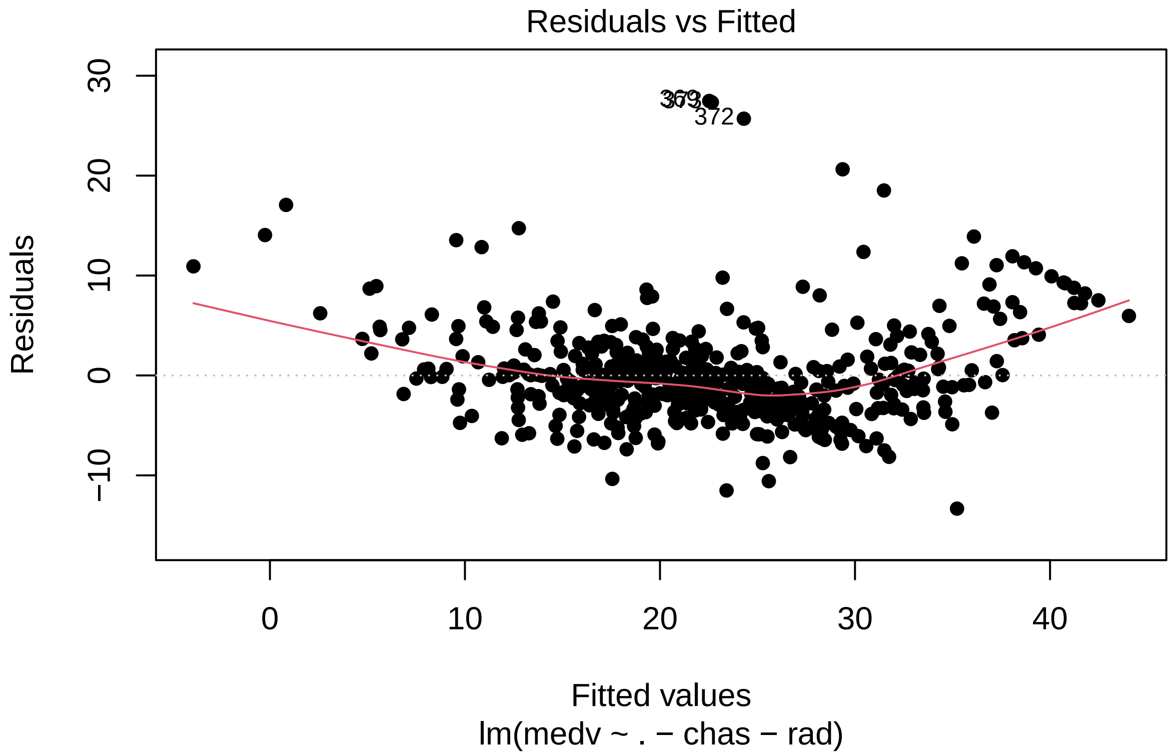
Comments: We notice that for $k = 2$ degrees of freedom, the marginal distributions for intercept and slope are densely clustered around one value (creating one sort of spike). For higher values of k, the distribution starts to spread out and looks like a normal distribution as we saw in Problem 1A.

From the plotted level lines of the joint distribution between slope and intercept, we see that the joint distribution is normal. We make this conclusion because for each of the plots, there is an ellipse shape with a higher level line in the center and declining values as points are farther away from the center.

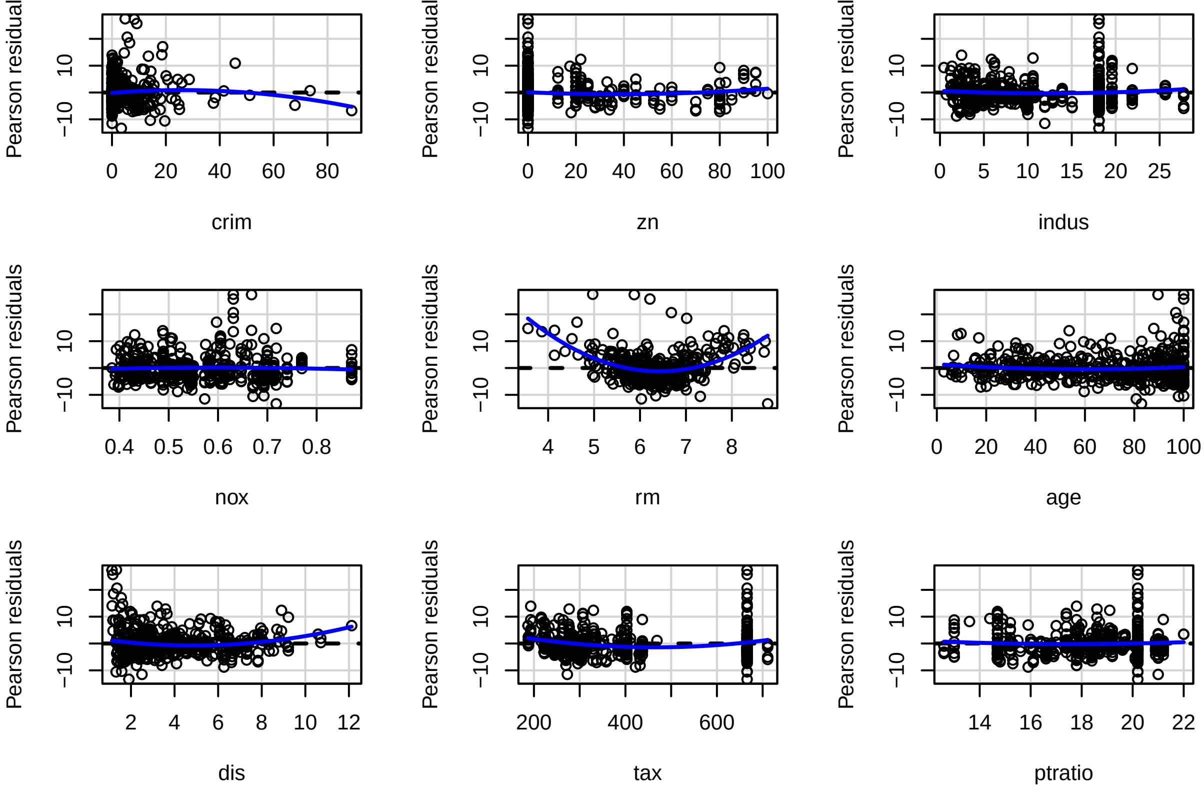
Problem 2A

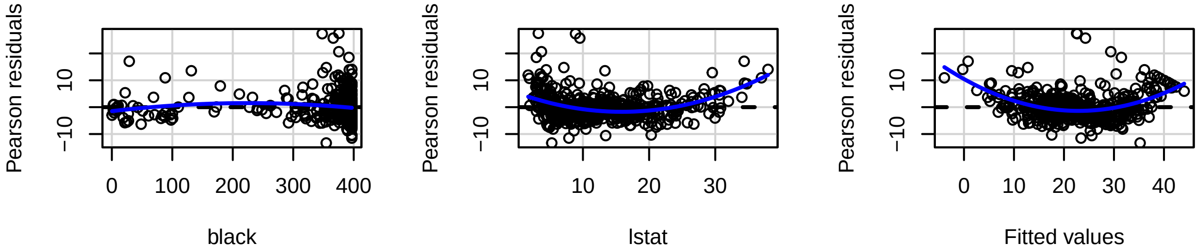
```r
# predict medv without discrete variables
fit = lm(medv ~ . - chas - rad, data = Boston)
```

```r
# assumption: mean zero, model accuracy
plot(fit, which=1, pch=16)# residuals vs. fitted values
```



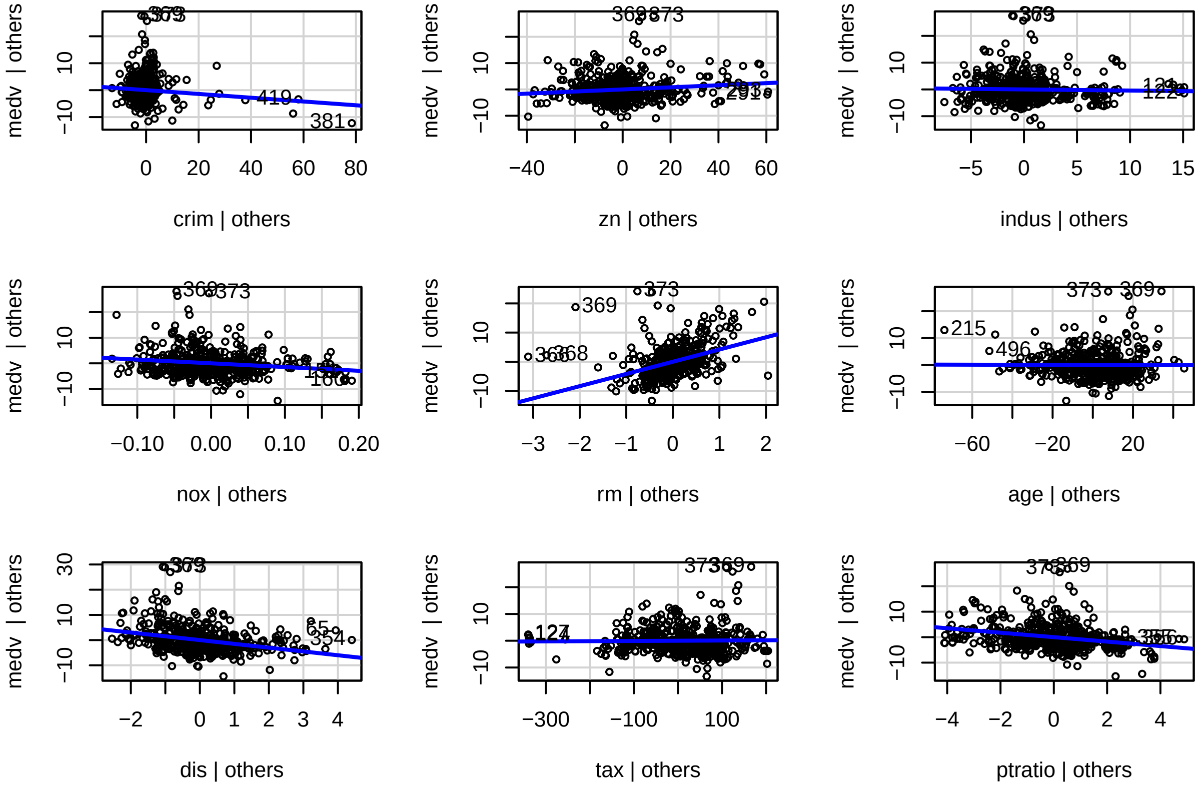
```r
residualPlots(fit)  # partial residual: looking at one variable
```


Pearson residuals
10
−10
0 20 40 60 80
crim
0 20 40 60 80 100
zn
0 5 10 15 20 25
indus
0.4 0.5 0.6 0.7 0.8
nox
4 5 6 7 8
rm
0 20 40 60 80 100
age
2 4 6 8 10 12
dis
200 400 600
tax
14 16 18 20 22
ptratio

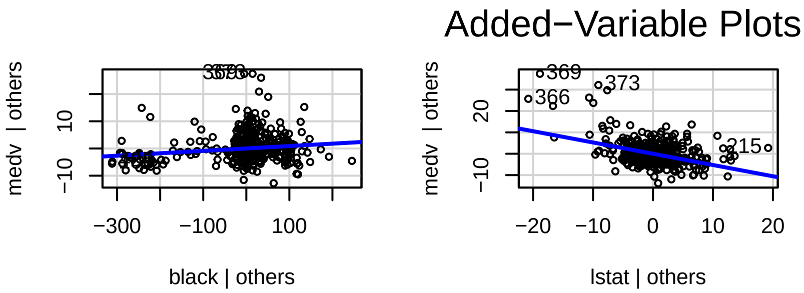


```
##            Test stat Pr(>|Test stat|)
## crim         -2.4854          0.01327 *
## zn            1.4242          0.15503
## indus         1.2836          0.19989
## nox          -1.0419          0.29799
## rm           12.6039        < 2.2e-16 ***
## age           1.7050          0.08883 .
## dis           4.3938        1.365e-05 ***
## tax           4.0370        6.275e-05 ***
## ptratio       0.9109          0.36282
## black        -2.1834          0.02948 *
## lstat        10.5122        < 2.2e-16 ***
## Tukey test   14.0829        < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
avPlots(fit)  # added variable: regressing out other variables
```



medv | others
crim | others
zn | others
indus | others
nox | others
rm | others
age | others
dis | others
tax | others
ptratio | others
369
373
419
381
291
121
122
215
496
368
365
65
354
127
160



We can clearly see that the residuals are not centered around zero. There is a curvature and more residual values lie above zero than below. When looking at the partial residual plots, we can make observations for one variable at a time:

- **crim:** centered around zero for low values, drops below zero as crime increases
- zn: centered around zero
- indus: centered around zero
- nox: centered around zero
- **rm:** clear curvature
- age: mostly centered around zero, a few higher points as age increases
- **dis:** at low values of dis, there are a few quite high points above zero but on the other hand, a dense collection of points below zero. as dis increases, we see more points above zero than below.
- tax: mostly centered around zero
- ptratio: centered around zero
- black: mostly centered around zero, slightly more points above zero
- **lstat:** slight curvature especially for higher values of lstat

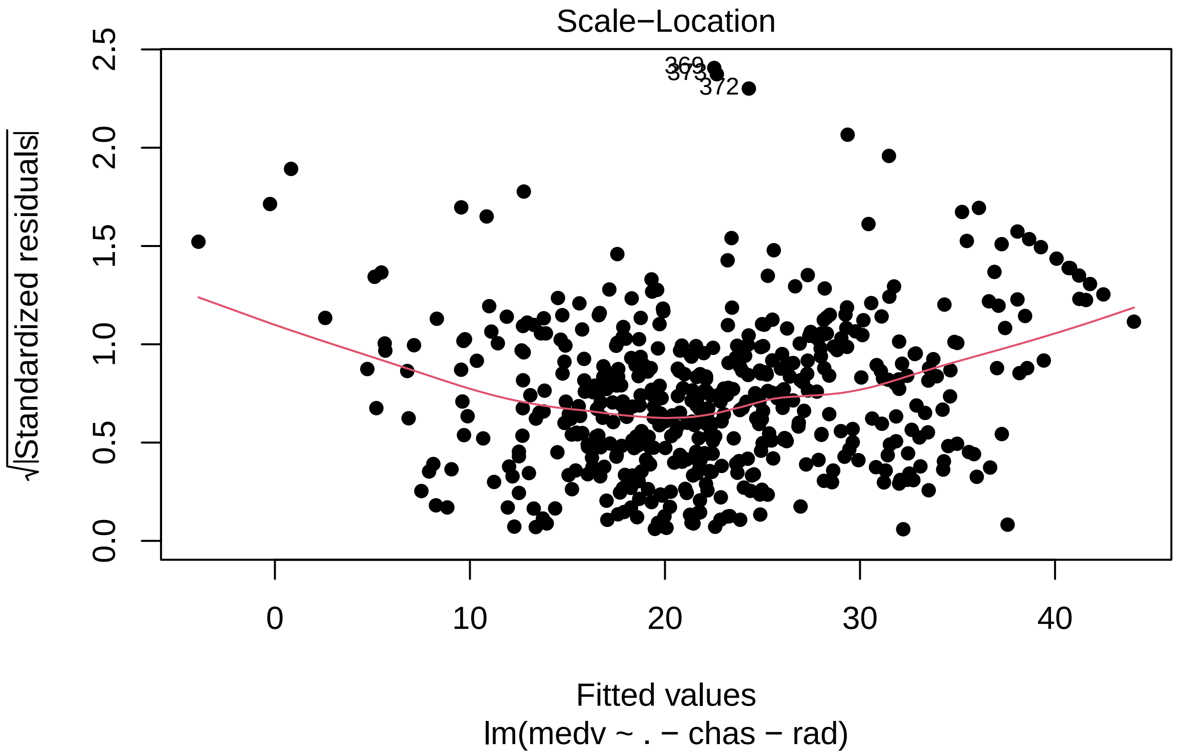
Looking at the added-variable plots, we make the following observations:

- **crime, lstat, and black** all have dense clusters of points that do not fit a linear model very well. One solution may be to transform the variable so that the points are more spread out.

- **zn, indus, nox, age, and tax** have barely, if any, linear relationship to medv. Almost all produce a horizontal line which signifies that these variables do not affect the response medv.

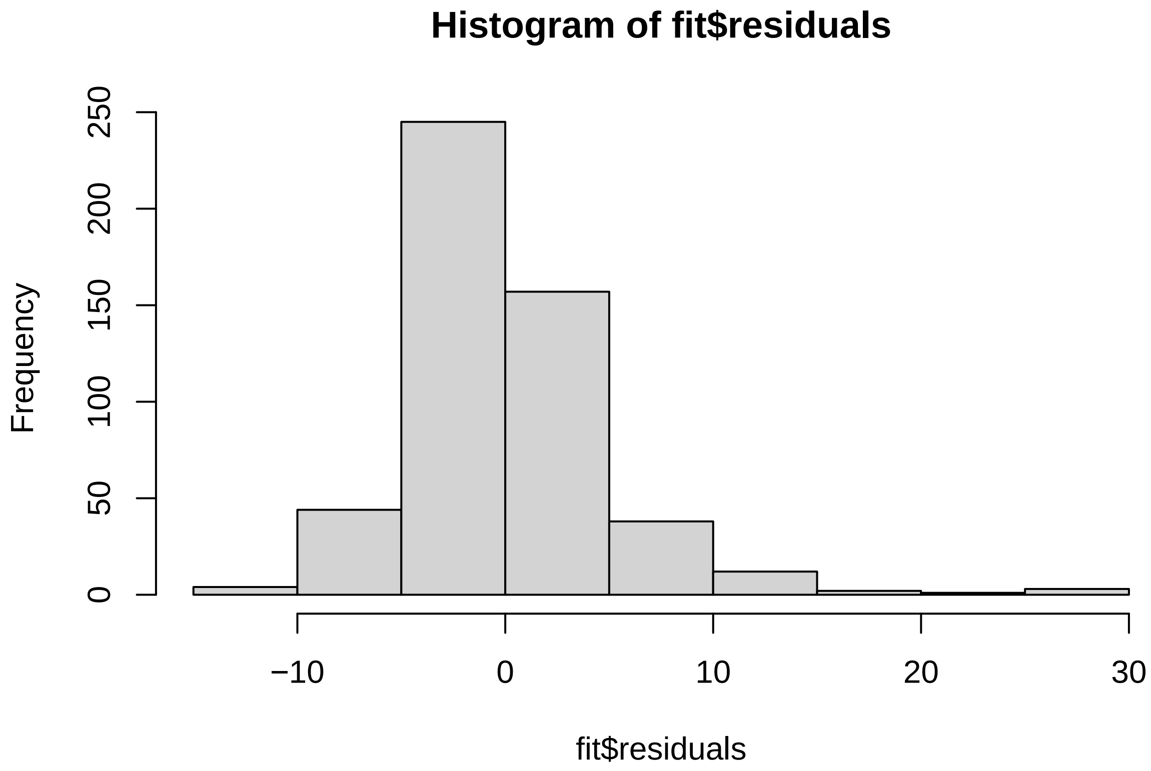
- **rm and dis** probably have the most apparent linear relationship to medv. rm is positively correlated and dis is negatively correlated to medv.

```
# assumption: homoscedasticity
plot(fit, which=3, cex=1, pch=16)  # scale location
```

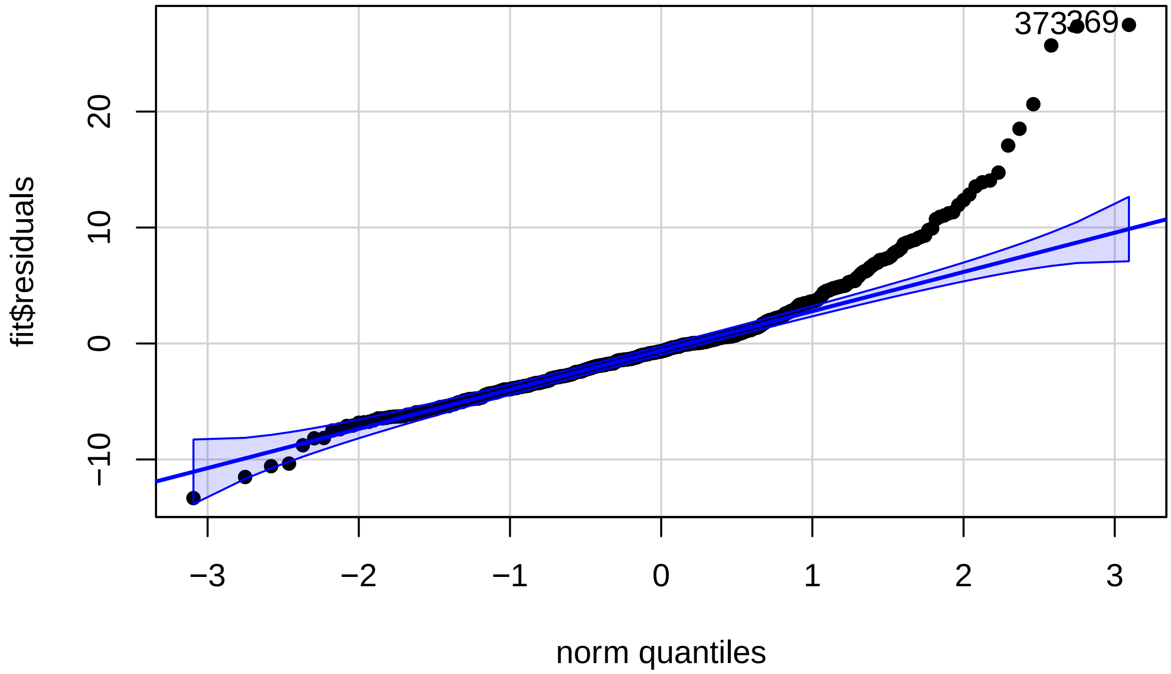


From the residuals vs. fitted plot (plotted in part A) and the scale-location plot above, we can see that the residuals do **not** form a more or less horizontal line. Instead, there is a curvature. It is hard to tell if this is due to heteroscedasticity (since we do not see a fan shape) or if this is simply due to outliers.

```
# assumption: normality
hist(fit$residuals)
```



```
qqPlot(fit$residuals, cex=1, pch=16)   # q-q plot
```

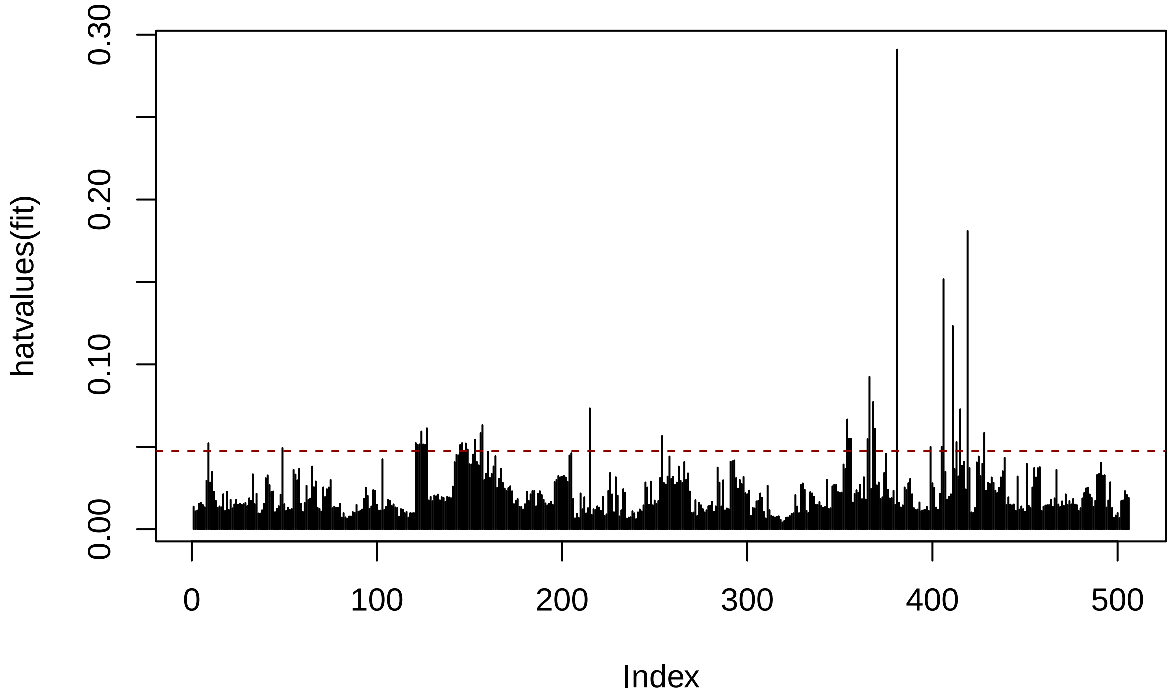


```
## [1] 369 373
```

When looking at the distribution of the residuals, we can see that the histogram produces a mostly normal shape except a bit heavier tailed on the right side. The q-q plot validates this as well, as we see that most of the points lie within the confidence band until the rightmost points. It seems that the normality assumption may not hold.

Problem 2B

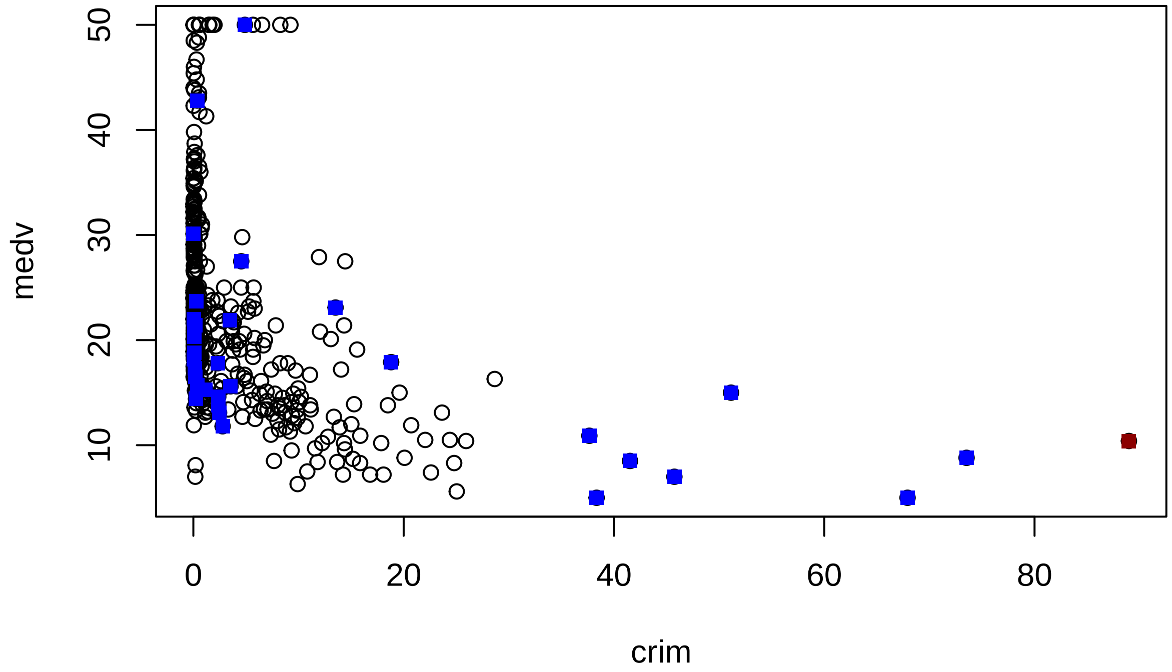
```
# outlier in predictor
plot(hatvalues(fit), type = "h")
p = length(fit$coefficients) - 1  # num of variables in model
n = dim(Boston)[1]
abline(h = 2 * (p + 1) / n, lty = 2,col = 'darkred')
```



```
# outlier in predictor
fit$model[hatvalues(fit) > 0.25, ]  # most significant
```

```
##     medv    crim zn indus chas   nox    rm  age    dis rad tax ptratio black
## 381 10.4 88.9762  0  18.1    0 0.671 6.968 91.9 1.4165  24 666    20.2 396.9
##     lstat
## 381 17.21
```

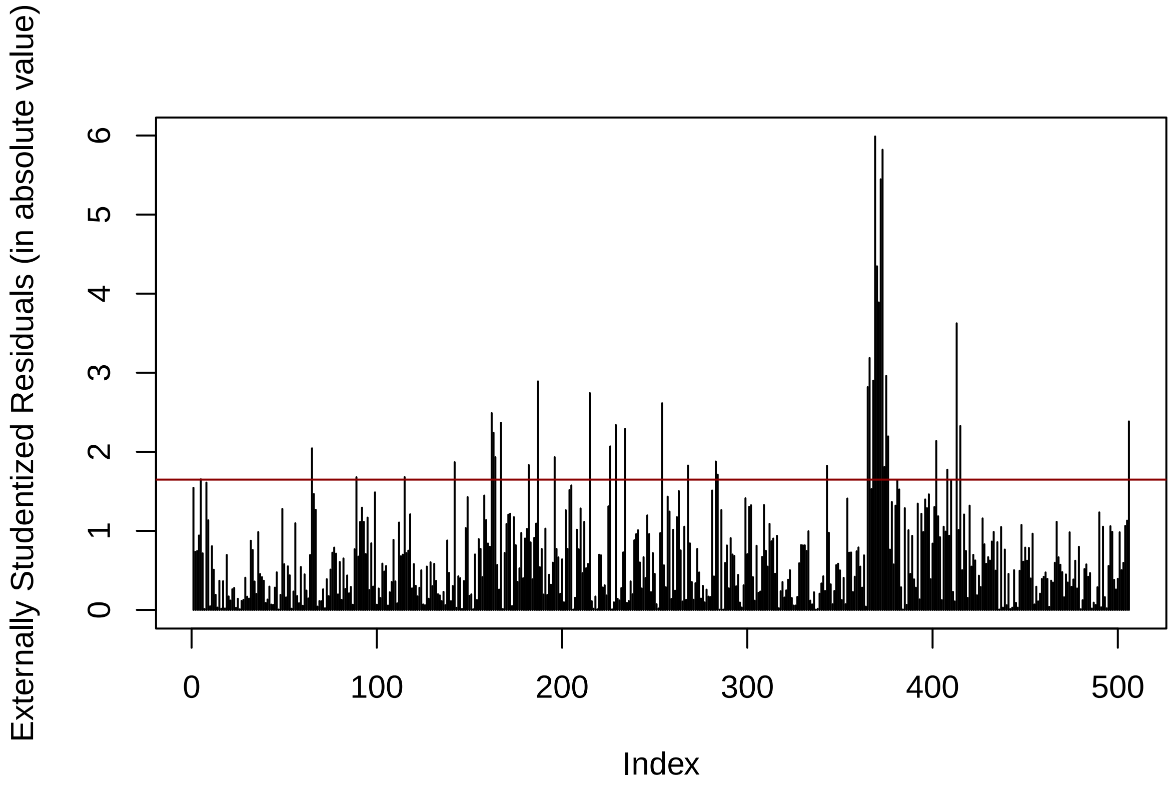
```
outliers_pred = hatvalues(fit) > 2 * (p + 1) / n
predictors = colnames(fit$model)
predictors = predictors [! predictors %in% c("medv", "chas", "rad")]
# for(predictor in predictors){
#   plot(fit$model[, predictor], fit$model[, "medv"], xlab = predictor, ylab = "medv")
#   points(fit$model[outliers_pred, predictor], fit$model[outliers_pred, "medv"],
#          col = 'blue', pch = 15)
#   points(fit$model[381, predictor], fit$model[381, "medv"], col = 'darkred', pch = 15)
# }

plot(fit$model[, "crim"], fit$model[, "medv"], xlab = "crim", ylab = "medv")
points(fit$model[outliers_pred, "crim"], fit$model[outliers_pred, "medv"],
       col = 'blue', pch = 15)
points(fit$model[381, "crim"], fit$model[381, "medv"], col = 'darkred', pch = 15)
```



Here, the blue points indicate the outliers in predictor (according to the threshold) and the red point indicates the most significant outlier in predictor, namely index 381 as printed above. By looking at each of the predictors, we see that index 381's crime value is significantly higher than the rest of the dataset.

Problem 2C

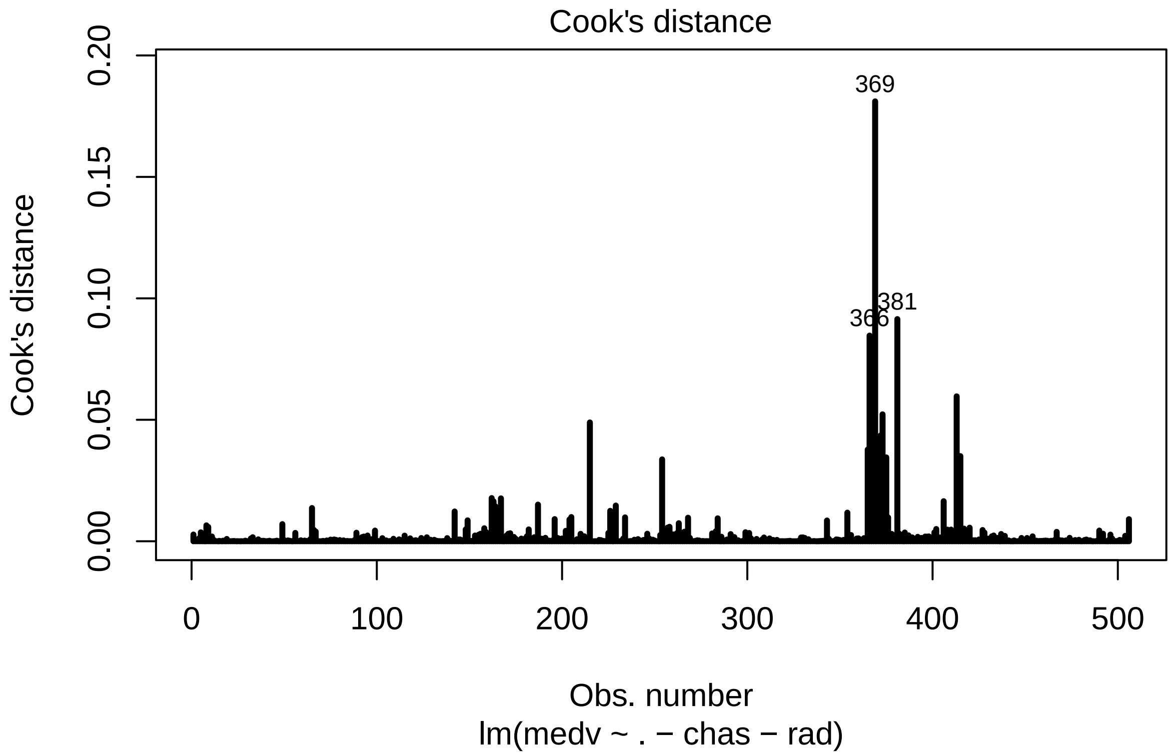
```
# outliers in response
plot(abs(rstudent(fit)), type = "h",
     ylab = "Externally Studentized Residuals (in absolute value)")
abline(h = qt(.95, n - p - 2),col = 'darkred')
```



```
fit$model[abs(rstudent(fit)) > 4, ]
```

```
##     medv     crim zn indus chas   nox    rm   age    dis rad tax ptratio  black
## 369   50 4.89822  0  18.1    0 0.631 4.970 100.0 1.3325  24 666    20.2 375.52
## 370   50 5.66998  0  18.1    1 0.631 6.683  96.8 1.3567  24 666    20.2 375.33
## 372   50 9.23230  0  18.1    0 0.631 6.216 100.0 1.1691  24 666    20.2 366.15
## 373   50 8.26725  0  18.1    1 0.668 5.875  89.6 1.1296  24 666    20.2 347.88
##     lstat
## 369  3.26
## 370  3.73
## 372  9.53
## 373  8.88
```

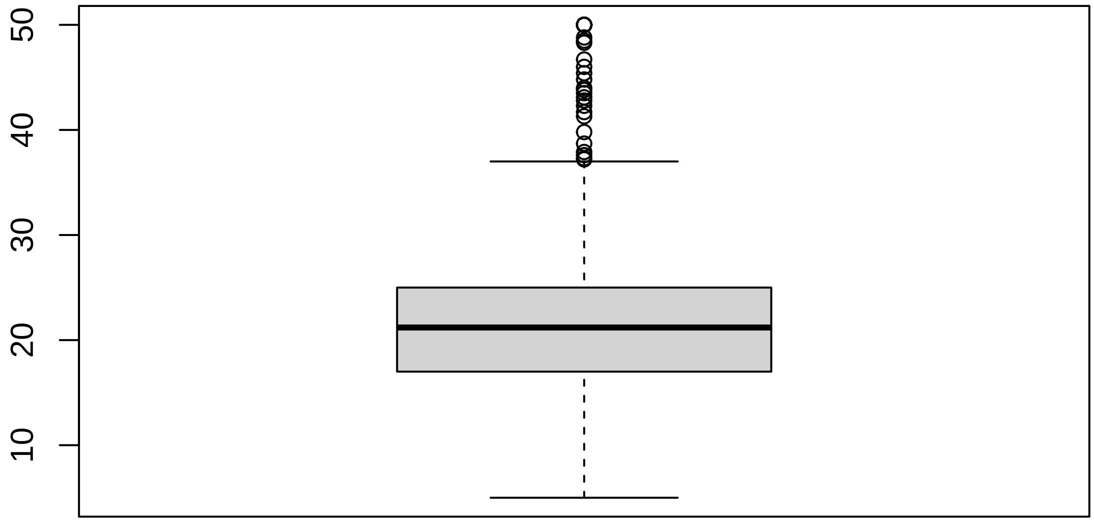
```
plot(fit, which=4, lwd=3) # Cook's distances
abline(h = 1, lty=2)
```



```
fit$model[c(366, 369, 381), ]
```

```
##     medv     crim zn indus chas   nox    rm   age    dis rad tax ptratio  black
## 366 27.5  4.55587  0  18.1    0 0.718 3.561  87.9 1.6132  24 666    20.2 354.70
## 369 50.0  4.89822  0  18.1    0 0.631 4.970 100.0 1.3325  24 666    20.2 375.52
## 381 10.4 88.97620  0  18.1    0 0.671 6.968  91.9 1.4165  24 666    20.2 396.90
##     lstat
## 366  7.12
## 369  3.26
## 381 17.21
```

```
outliers_resp = boxplot(Boston$medv)$out
```



```
outliers_resp_ind = which(Boston$medv %in% outliers_resp)

outliers_resp
```

```
##  [1] 38.7 43.8 41.3 50.0 50.0 50.0 50.0 37.2 39.8 37.9 50.0 50.0 42.3 48.5 50.0
## [16] 44.8 50.0 37.6 46.7 41.7 48.3 42.8 44.0 50.0 43.1 48.8 50.0 43.5 45.4 46.0
## [31] 50.0 37.3 50.0 50.0 50.0 50.0 50.0
```

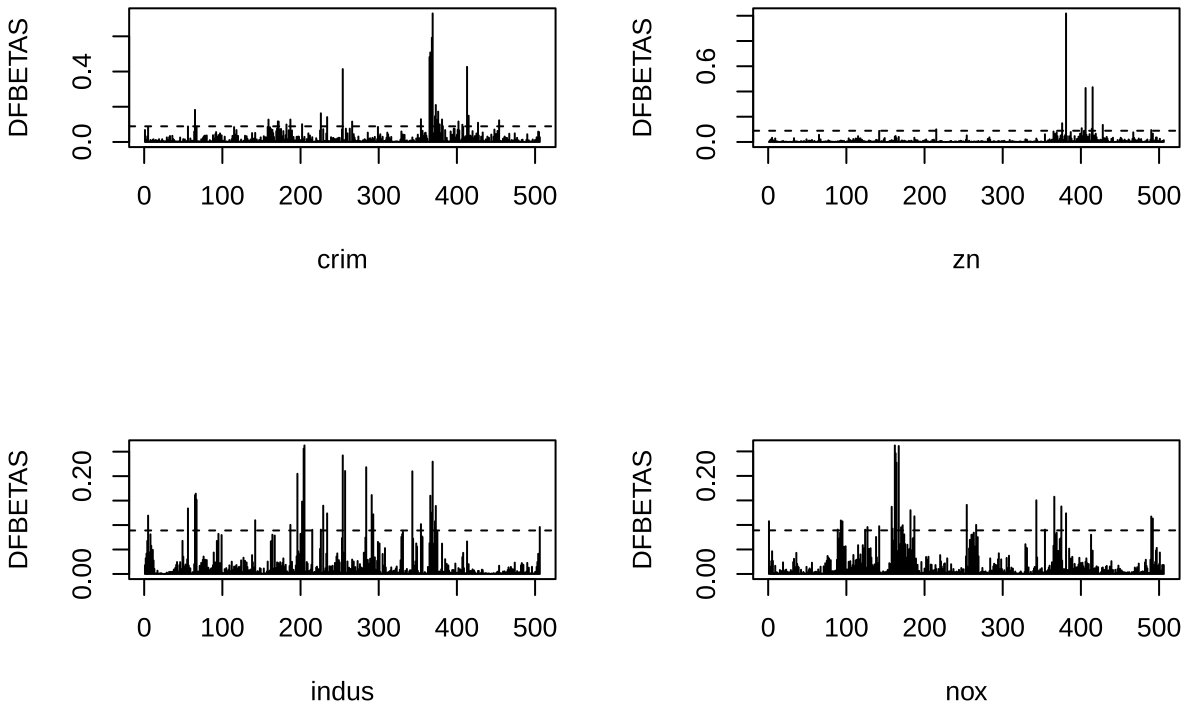
```
outliers_resp_ind
```

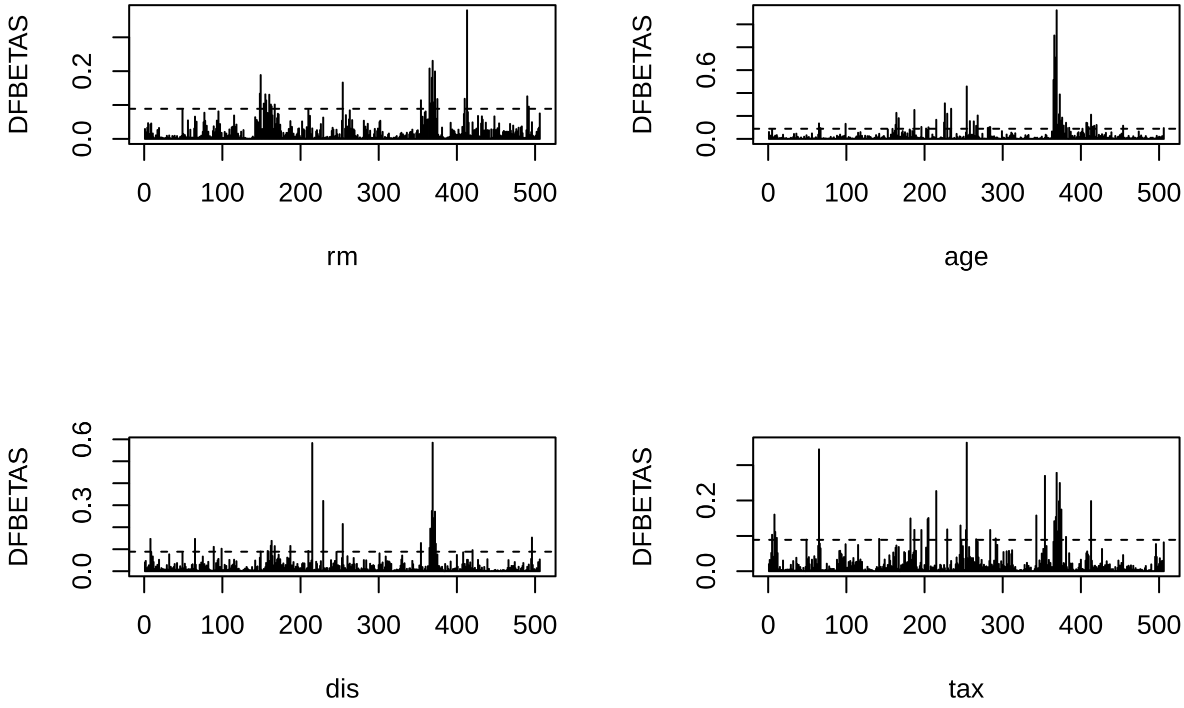
```
##  [1]  98  99 158 162 163 164 167 180 181 183 187 196 203 204 205 225 226 227 229
## [20] 233 234 254 257 258 262 263 268 269 281 283 284 292 369 370 371 372 373
```

It is interesting to note that all the outliers in response happen to be around the same index. From the Externally Studentized Residuals plot, we see that the indices 369, 370, 372, and 373 are the most significant outliers. From the Cook's distance plot, we that the indices 366, 369, and 381 stand out - however, none of the values are above 1 which is the rule of thumb threshold for Cook's distance. After looking at the boxplot of the response variable *medv*, we see that the indices 369, 370, 372, and 373 all have value 50 for *medv*, which is the highest value that occurs and is far above the interquartile range.

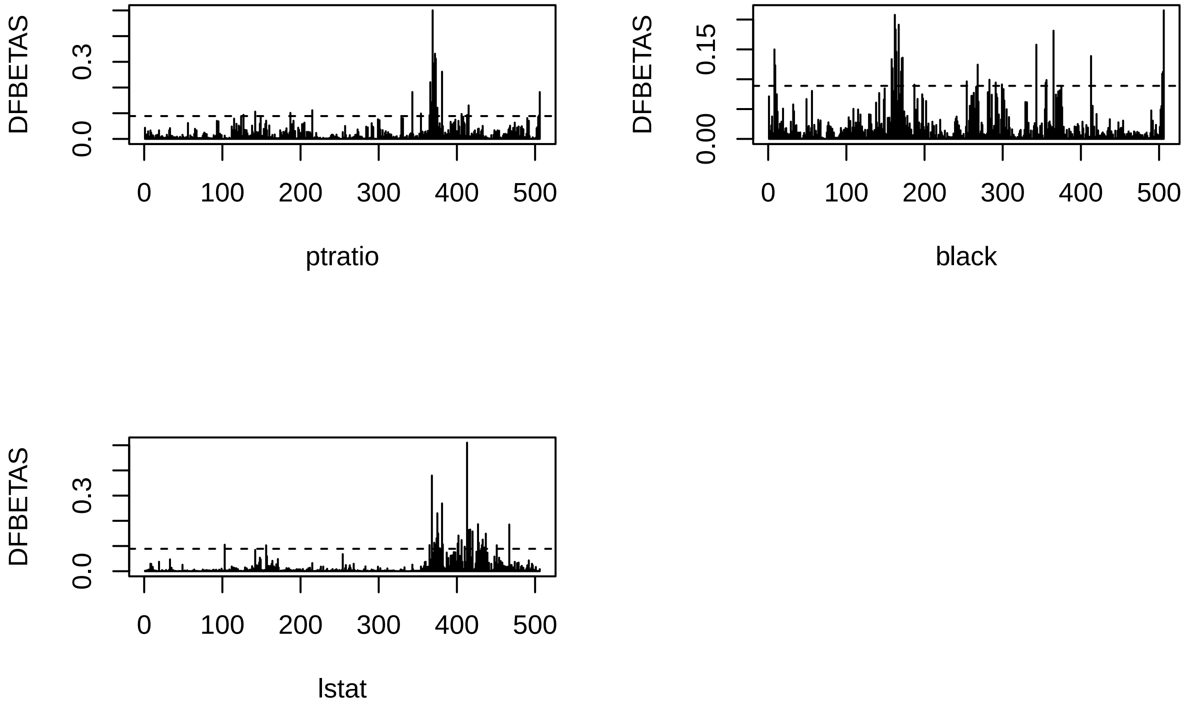
Problem 2D

```r
# influential observations: outlier in predictor and response

# DFBETAS
par(mfrow=c(2,2))
for (j in 1:p){
    plot(abs(dfbetas(fit)[,j]), type='h', xlab=predictors[j], ylab='DFBETAS')
    abline(h = 2/sqrt(n), lty=2) # threshold for suspects
    }
```

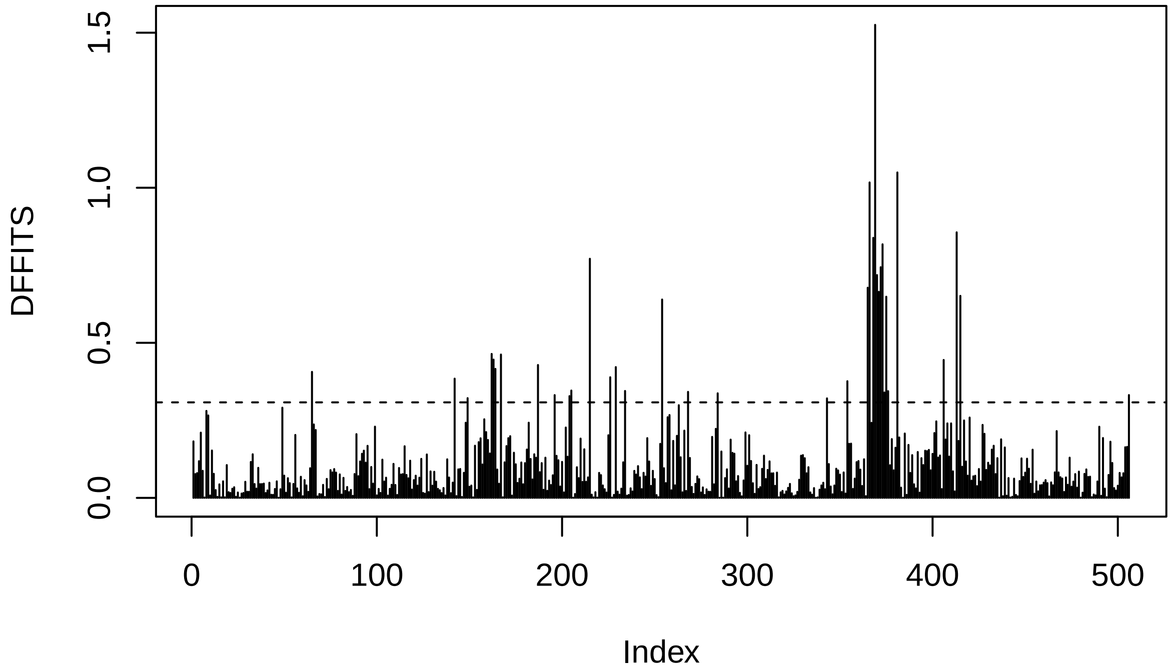




```
# DFFITS
par(mfrow=c(1,1))
```



```
plot(abs(dffits(fit)), typ='h', ylab='DFFITS')
abline(h = 2*sqrt((p+1)/n), lty=2) # threshold for suspects
```



```
fit$model[abs(rstudent(fit)) > qt(.95, n - p - 2)
          & hatvalues(fit) > 2 * (p + 1) / n, ]
```

```
##     medv     crim zn indus chas   nox    rm   age    dis rad tax ptratio  black
## 215 23.7  0.28955  0 10.59    0 0.489 5.412   9.8 3.5875   4 277    18.6 348.93
## 254 42.8  0.36894 22  5.86    0 0.431 8.259   8.4 8.9067   7 330    19.1 396.90
## 365 21.9  3.47428  0 18.10    1 0.718 8.780  82.9 1.9047  24 666    20.2 354.55
## 366 27.5  4.55587  0 18.10    0 0.718 3.561  87.9 1.6132  24 666    20.2 354.70
## 368 23.1 13.52220  0 18.10    0 0.631 3.863 100.0 1.5106  24 666    20.2 131.42
## 369 50.0  4.89822  0 18.10    0 0.631 4.970 100.0 1.3325  24 666    20.2 375.52
## 413 17.9 18.81100  0 18.10    0 0.597 4.628 100.0 1.5539  24 666    20.2  28.79
## 415  7.0 45.74610  0 18.10    0 0.693 4.519 100.0 1.6582  24 666    20.2  88.27
##     lstat
## 215 29.55
## 254  3.54
## 365  5.29
## 366  7.12
## 368 13.33
## 369  3.26
## 413 34.37
## 415 36.98
```

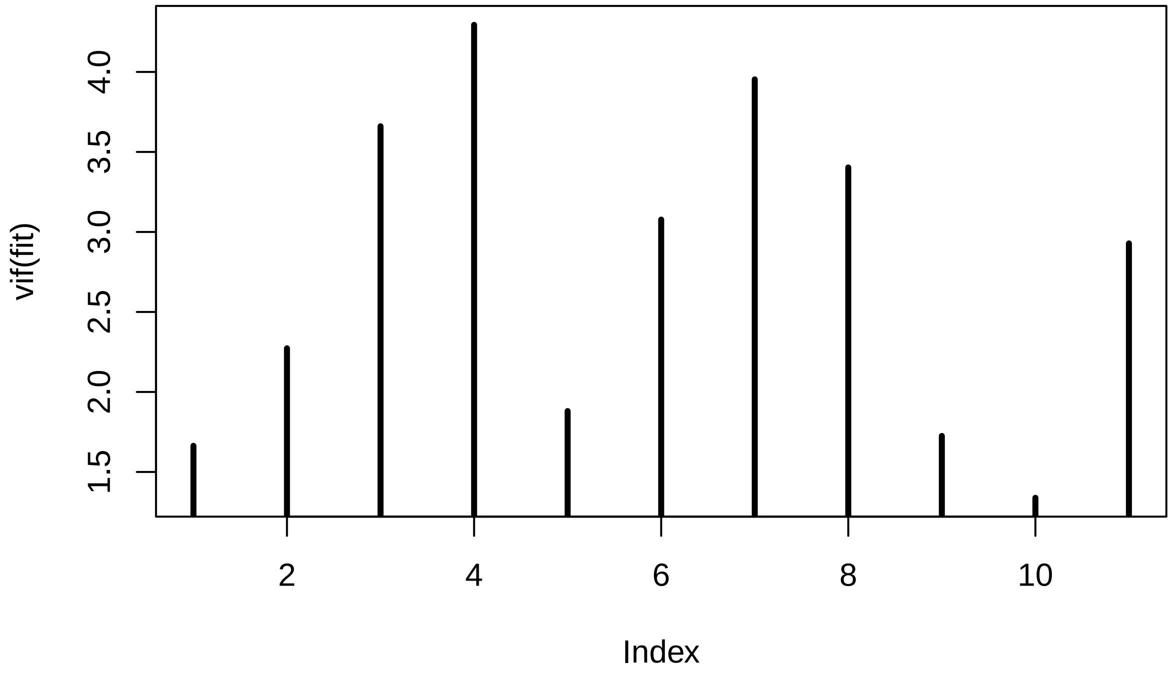
The data observations printed above all violate both the hat value threshold and the externally studentized residual threshold, meaning that they are outliers in both predictor and response i.e. influential observations.

Problem 2E

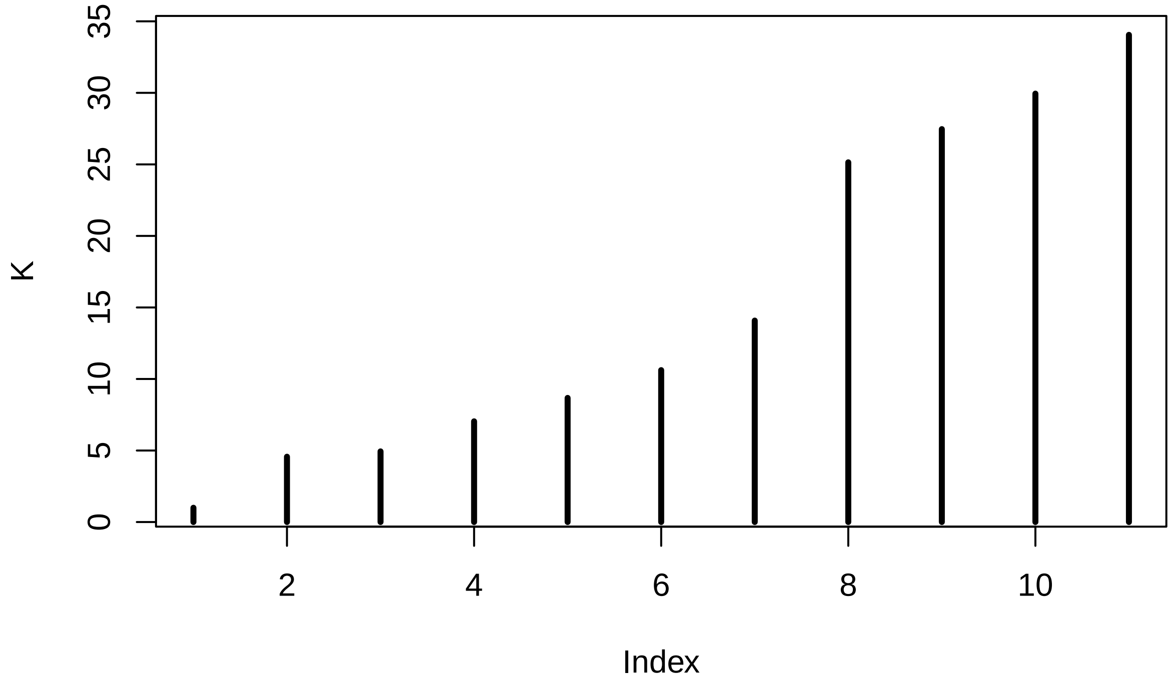
```
# multicollinearity

# checking via pairwise correlations b/w predictors
dat = subset(Boston, select = -c(chas, rad))
round( cor(dat) , 2) # rounded to 2 digits
```

```
##          crim    zn indus   nox    rm   age   dis   tax ptratio black lstat
## crim     1.00 -0.20  0.41  0.42 -0.22  0.35 -0.38  0.58    0.29 -0.39  0.46
## zn      -0.20  1.00 -0.53 -0.52  0.31 -0.57  0.66 -0.31   -0.39  0.18 -0.41
## indus    0.41 -0.53  1.00  0.76 -0.39  0.64 -0.71  0.72    0.38 -0.36  0.60
## nox      0.42 -0.52  0.76  1.00 -0.30  0.73 -0.77  0.67    0.19 -0.38  0.59
## rm      -0.22  0.31 -0.39 -0.30  1.00 -0.24  0.21 -0.29   -0.36  0.13 -0.61
## age      0.35 -0.57  0.64  0.73 -0.24  1.00 -0.75  0.51    0.26 -0.27  0.60
## dis     -0.38  0.66 -0.71 -0.77  0.21 -0.75  1.00 -0.53   -0.23  0.29 -0.50
## tax      0.58 -0.31  0.72  0.67 -0.29  0.51 -0.53  1.00    0.46 -0.44  0.54
## ptratio  0.29 -0.39  0.38  0.19 -0.36  0.26 -0.23  0.46    1.00 -0.18  0.37
## black   -0.39  0.18 -0.36 -0.38  0.13 -0.27  0.29 -0.44   -0.18  1.00 -0.37
## lstat    0.46 -0.41  0.60  0.59 -0.61  0.60 -0.50  0.54    0.37 -0.37  1.00
## medv    -0.39  0.36 -0.48 -0.43  0.70 -0.38  0.25 -0.47   -0.51  0.33 -0.74
##          medv
## crim    -0.39
## zn       0.36
## indus   -0.48
## nox     -0.43
## rm       0.70
## age     -0.38
## dis      0.25
## tax     -0.47
## ptratio -0.51
## black    0.33
## lstat   -0.74
## medv     1.00
```

```
# checking via variance inflation factors (VIF)
plot(vif(fit), type='h', lwd=3)
abline(h = 10, lty=2) # threshold for suspects
```



```
# checking via condition indices
C = cor(dat[, predictors]) # correlation matrix for the predictors
L = eigen(C) # eigenvalues
K = max(L$val)/L$val # condition indices
plot(K, type='h', lwd=3)
abline(h = 1000, lty=2) # threshold for suspects
```



Looks like multicollinearity is not an issue. None of the correlations in the map have absolute value above 0.8. In addition, for each variable the VIF stays below 10 and the condition number below 1000.

Contributions:

We worked on all parts of this assignment together.