PQHS 471 Homework 1; Gregory Powers

8a

loads data into r from a csv

```
college <- read.csv('c:/sas/r/college.csv')
```

8b

```
fix(college)
rownames(college) = college[,1]
college = college[,-1]
fix(college)
```

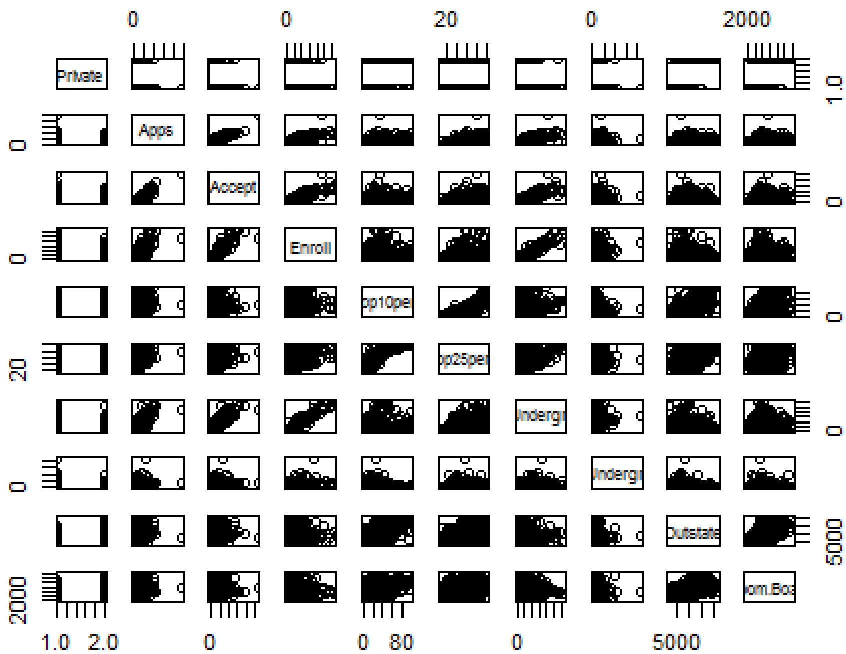
8c

```
summary(college)
```

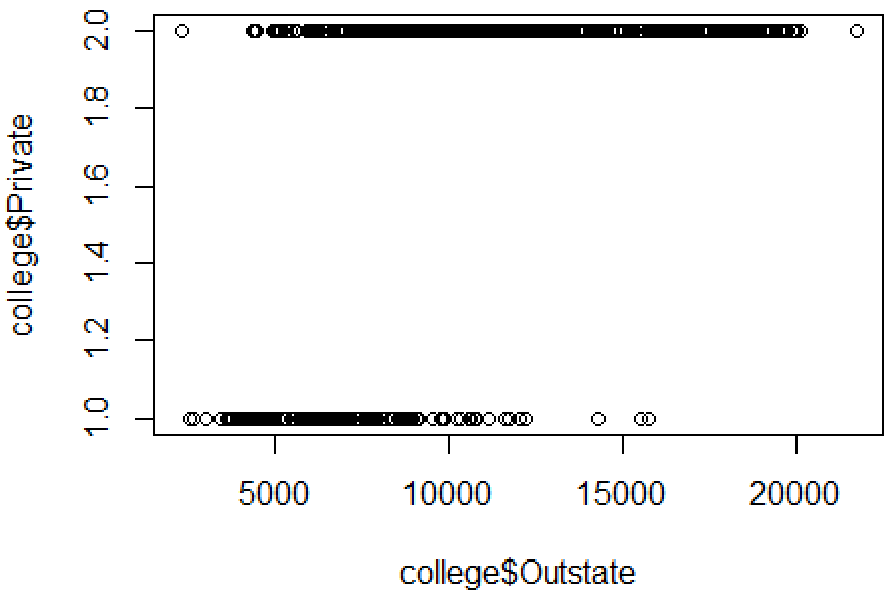
```
##  Private        Apps           Accept          Enroll       Top10perc    
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00  
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00  
##            Median : 1558   Median : 1110   Median : 434   Median :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
```

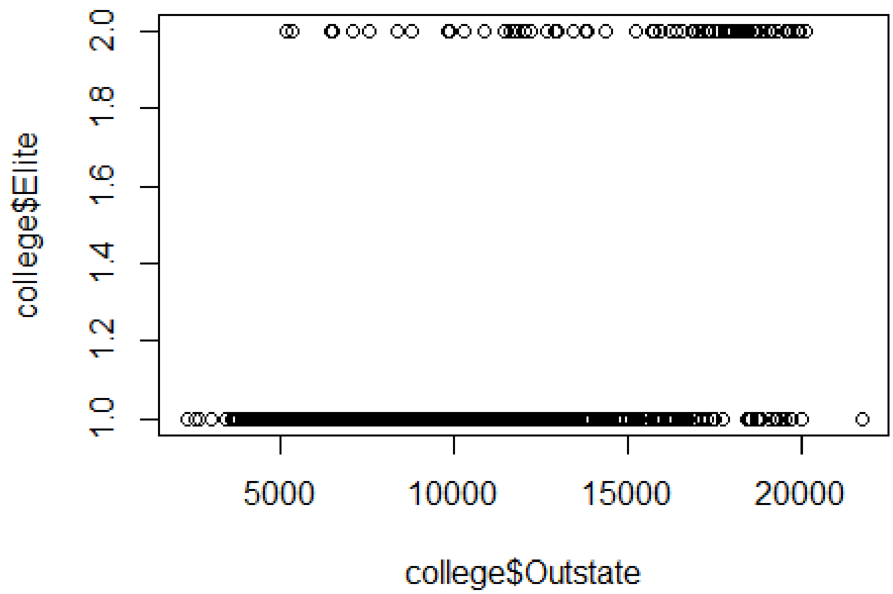
```
pairs(college [,1:10])
```




```
plot(college$Outstate, college$Private)
```

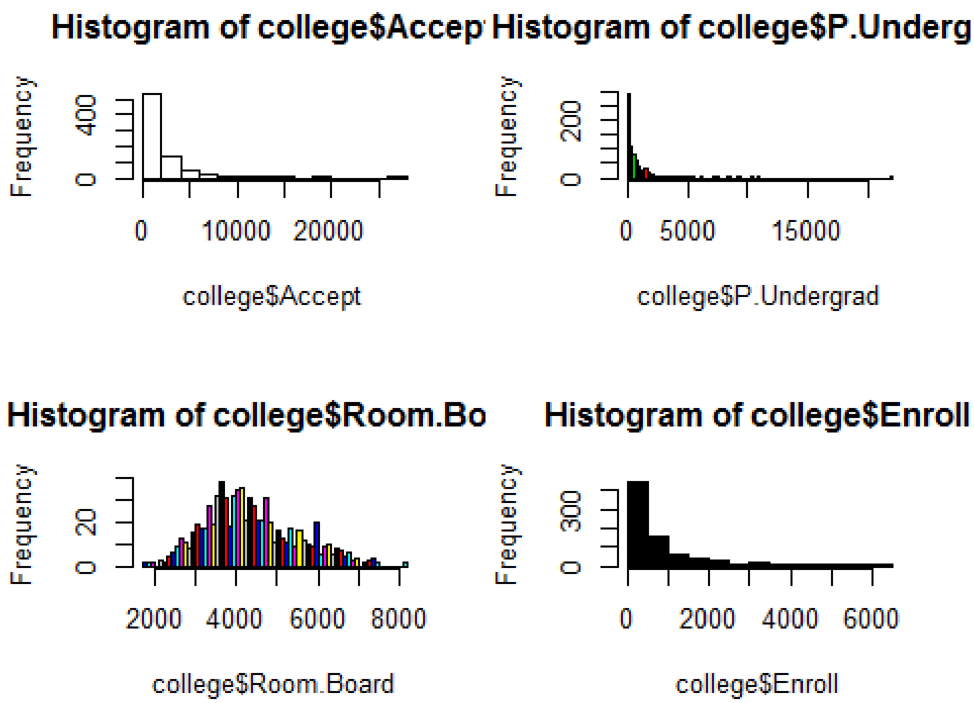



```
Elite = rep("No", nrow(college))
Elite[college$Top10perc>50] = "Yes"
Elite = as.factor(Elite)
college = data.frame(college, Elite)
summary(Elite)

##  No Yes
## 699  78

plot(college$Outstate, college$Elite)
```



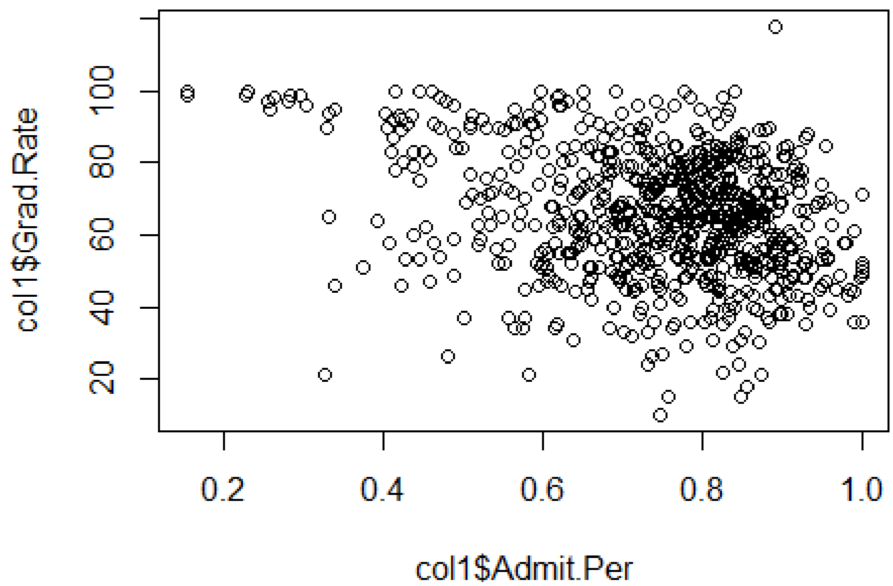
#8d

```
par(mfrow=c(2,2))
hist(college$Accept)
hist(college$P.Undergrad,breaks = 100,col = 1:3)
hist(college$Room.Board, breaks = 50, col = 4:10)
hist(college$Enroll, breaks = 10, col = 1)
```

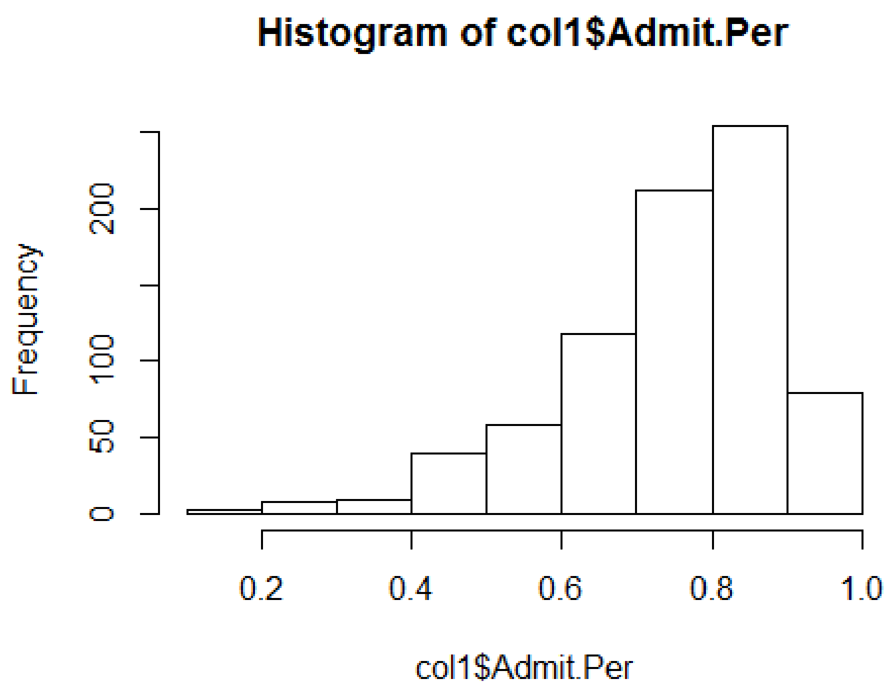


#8f

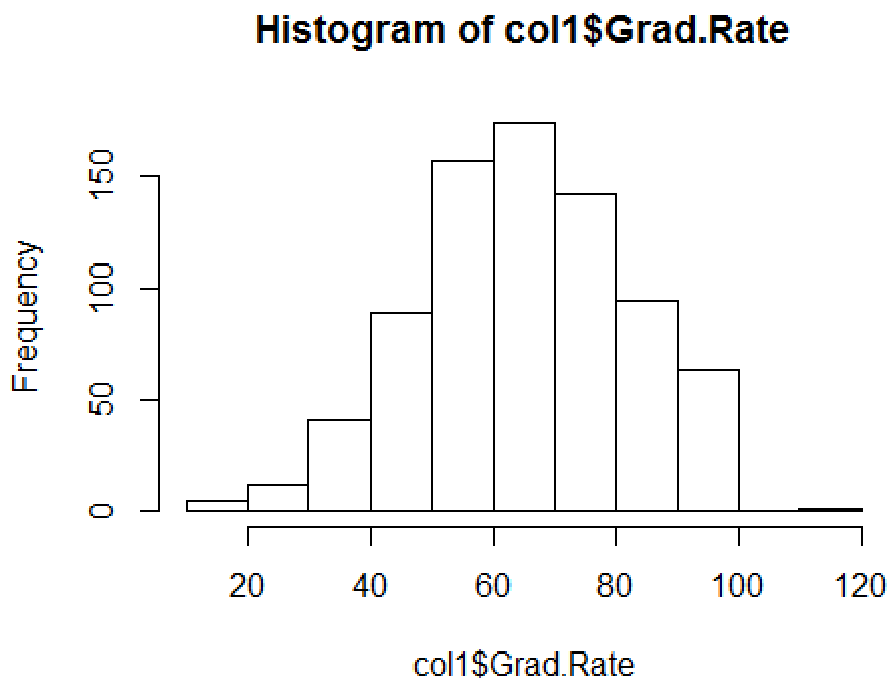
```
par(mfrow=c(1,1))
Admit.Per <- college$Accept/college$Apps
col1 <- data.frame(college, Admit.Per)
plot(col1$Admit.Per, col1$Grad.Rate)
```




```
hist(col1$Admit.Per)
```



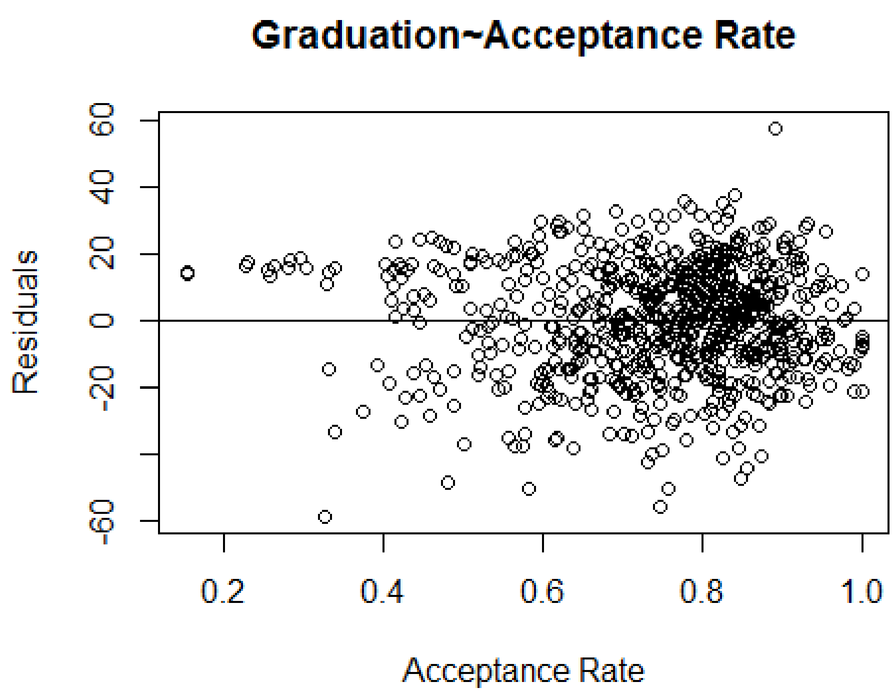

```
hist(col1$Grad.Rate)
```




```
cor.test(col1$Admit.Per, col1$Grad.Rate,  method = c("pearson"))

##
##  Pearson's product-moment correlation
##
## data:  col1$Admit.Per and col1$Grad.Rate
## t = -8.3397, df = 775, p-value = 3.39e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.3502356 -0.2211010
## sample estimates:
##        cor
## -0.2869715

colgrad <- lm(Grad.Rate ~ Admit.Per, data = col1)
summary(colgrad)

##
## Call:
## lm(formula = Grad.Rate ~ Admit.Per, data = col1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -58.491 -10.806   0.968  12.496  57.411
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
```

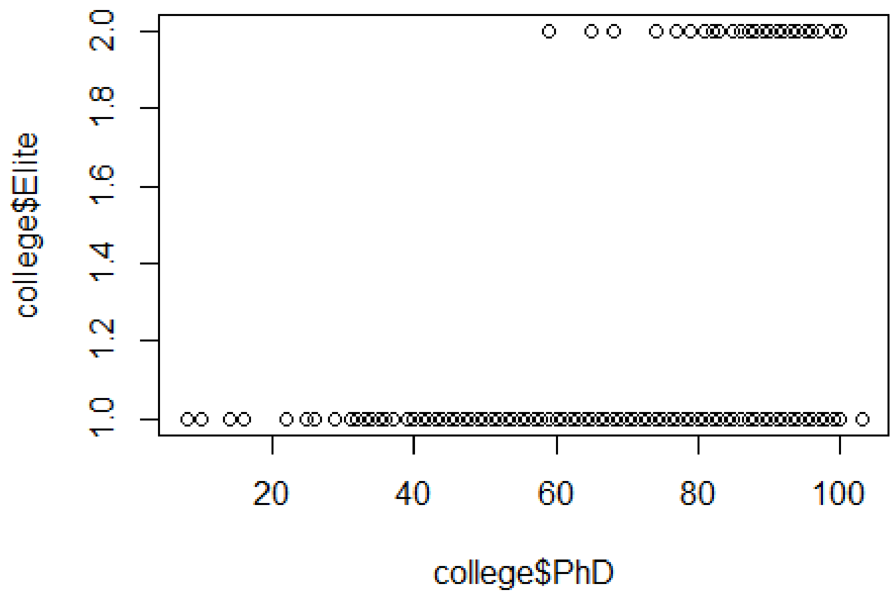
```
## (Intercept)   90.493      3.059   29.58  < 2e-16 ***
## Admit.Per    -33.510      4.018   -8.34 3.39e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 16.47 on 775 degrees of freedom
## Multiple R-squared:  0.08235,    Adjusted R-squared:  0.08117
## F-statistic: 69.55 on 1 and 775 DF,  p-value: 3.39e-16
```

```
colgrad.res <- resid(colgrad)
plot(col1$Admit.Per, colgrad.res,
     ylab="Residuals", xlab="Acceptance Rate",
     main="Graduation~Acceptance Rate")
abline(0, 0)
```

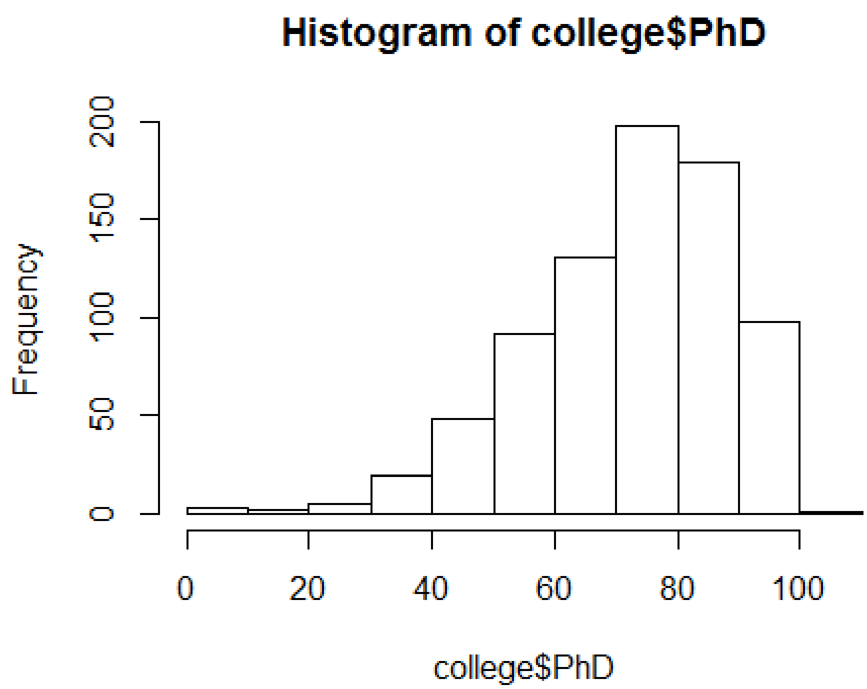


There is a statistically significant negative association between a college's acceptance rate (admissions/applications) and its rate of graduation; acceptance rate accounts for 8.1% of the variation in graduation rate. As acceptance rates increase, graduation rates decrease. More selective schools choose only the most prepared applicants (see below).

```
plot(college$PhD, college$Elite)
```




```r
hist(college$PhD)
```

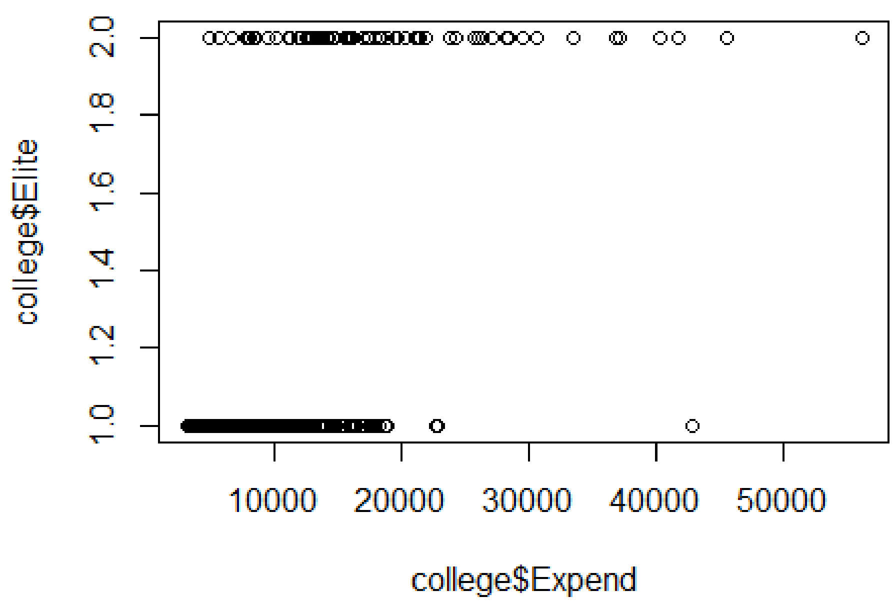


There seems to be a relationship between elite status and having a PhD program. This is borne out by the below t-test, which gives sufficient evidence to reject the null hypothesis that the group means are equal: on average, 89% of elite schools have PhD programs, vs. ~71% non-elite.

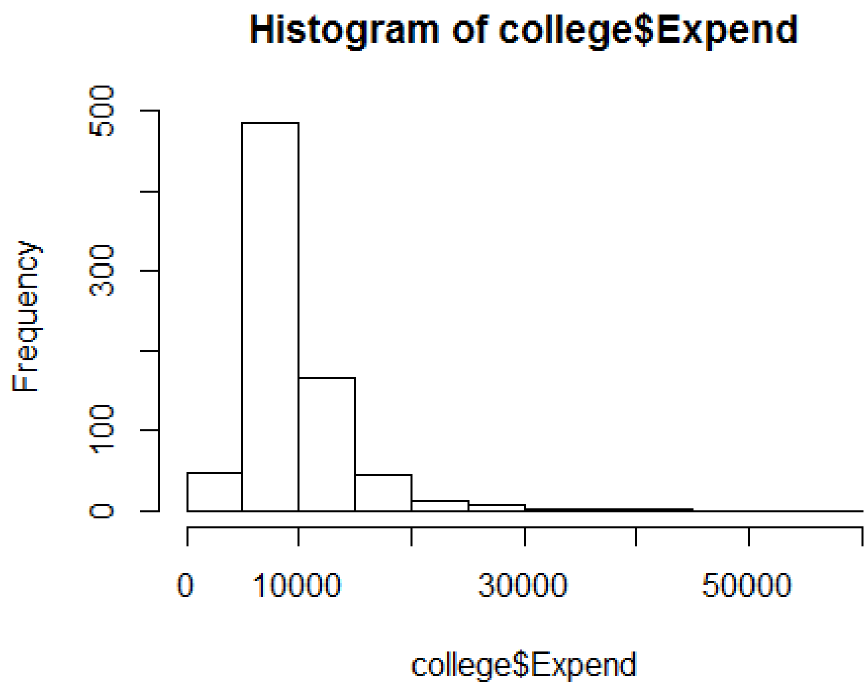
```
t.test(college$PhD~college$Elite)

##
##  Welch Two Sample t-test
##
## data:  college$PhD by college$Elite
## t = -17.029, df = 157.5, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -20.66733 -16.37140
## sample estimates:
##  mean in group No mean in group Yes
##          70.80114          89.32051
```

Are elite colleges more expensive on average than non-elite schools?

```
plot(college$Expend, college$Elite)
```

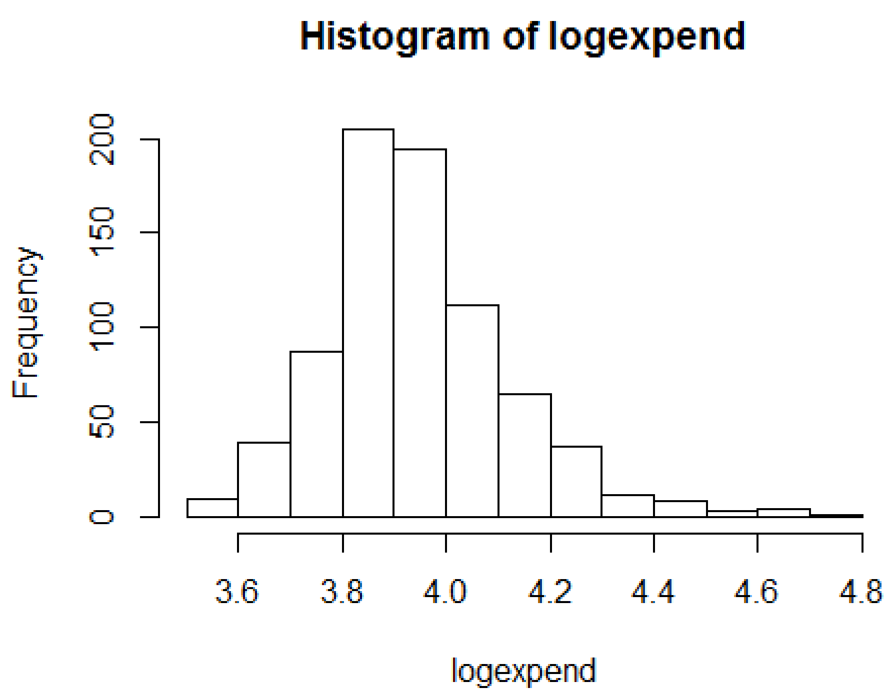



```
hist(college$Expend)
```



because the distribution appears skewed, the Expend variable will be log transformed.

```
logexpend <- log10(college$Expend)
hist(logexpend)
```



```r
t.test(logexpend~college$Elite)

##
##  Welch Two Sample t-test
##
## data:  logexpend by college$Elite
## t = -12.061, df = 85.084, p-value < 2.2e-16
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -0.3483542 -0.2497599
## sample estimates:
##  mean in group No mean in group Yes
##          3.913651          4.212708
```

Elite colleges are, on average, more expensive than non-elite schools.

```r
cor.test(college$Expend, college$S.F.Ratio, method = c("pearson"))

##
##  Pearson's product-moment correlation
##
## data:  college$Expend and college$S.F.Ratio
## t = -20.019, df = 775, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.6283629 -0.5354875
## sample estimates:
##       cor
## -0.583832

cor.test(college$Expend, college$Top10perc, method = c("pearson"))

##
##  Pearson's product-moment correlation
##
## data:  college$Expend and college$Top10perc
## t = 24.517, df = 775, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  0.6193712 0.6987651
## sample estimates:
##       cor
## 0.6609134

cor.test(college$Expend, college$Grad.Rate, method = c("pearson"))

##
##  Pearson's product-moment correlation
##
## data:  college$Expend and college$Grad.Rate
## t = 11.803, df = 775, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
```

```
## 95 percent confidence interval:
##  0.3290431 0.4483664
## sample estimates:
##       cor
## 0.3903427
```

```
cor.test(col1$Expend, col1$Admit.Per, method = c("pearson"))
```

```
##
##  Pearson's product-moment correlation
##
## data:  col1$Expend and col1$Admit.Per
## t = -12.464, df = 775, p-value < 2.2e-16
## alternative hypothesis: true correlation is not equal to 0
## 95 percent confidence interval:
##  -0.4655750 -0.3482992
## sample estimates:
##        cor
## -0.4086223
```

```
summary(lm(Grad.Rate ~ Expend + S.F.Ratio + Admit.Per + PhD, data = col1))
```

```
##
## Call:
## lm(formula = Grad.Rate ~ Expend + S.F.Ratio + Admit.Per + PhD,
##     data = col1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -53.221  -9.657   0.398   9.746  65.215
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  7.217e+01  5.811e+00  12.418  < 2e-16 ***
## Expend       4.935e-04  1.539e-04   3.208  0.00139 **
## S.F.Ratio   -7.830e-01  1.755e-01  -4.461 9.35e-06 ***
## Admit.Per   -1.779e+01  4.205e+00  -4.230 2.61e-05 ***
## PhD          1.768e-01  3.831e-02   4.616 4.58e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 15.32 on 772 degrees of freedom
## Multiple R-squared:  0.2089, Adjusted R-squared:  0.2048
## F-statistic: 50.96 on 4 and 772 DF,  p-value: < 2.2e-16
```

Expend, S.F. Ratio, PhD and Admit.Per are significantly associated with graduation rate. Holding other coefficients constant, Expend and PhD are positively associated; S.F. Ratio and Admit.Per negatively. This model accounts for about 21% of the variation in graduation rate.

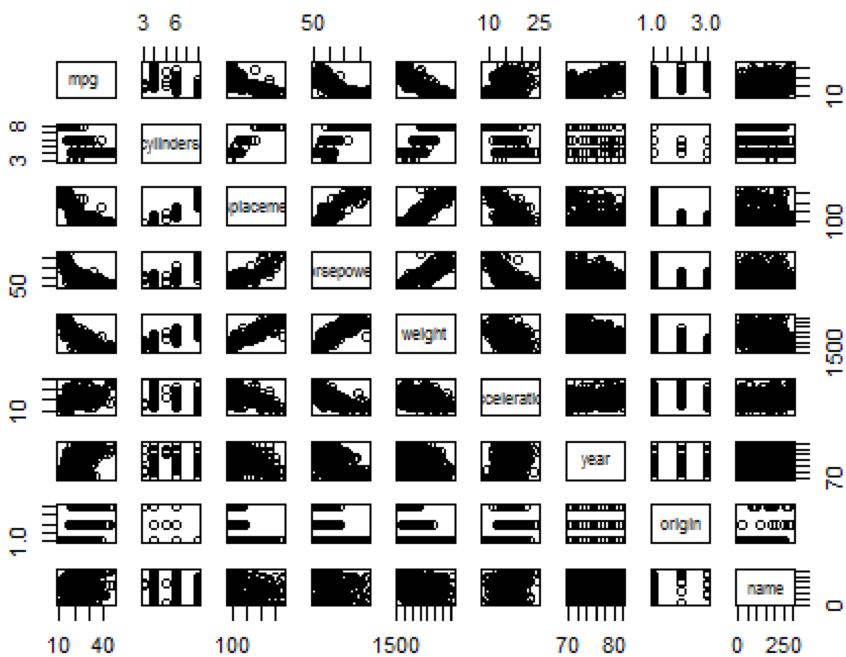
Schools classified as the elite are the most selective and expensive, and the elite get the most able students and have the best student to faculty ratios. They also have helpful PhD students which attract the best faculty and can also teach and TA.

Going from the above summary statement, the college data set seems to have values that are out of range: "PhD" has a max value of 103%, Grad.Rate which is presumably also a proportion, has a max value of 118%.

Chapter 3

9a

```
library(ISLR)

## Warning: package 'ISLR' was built under R version 3.4.3

pairs(Auto)
```



#9b

```
cor(subset(Auto, select=-name))

##                     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
```

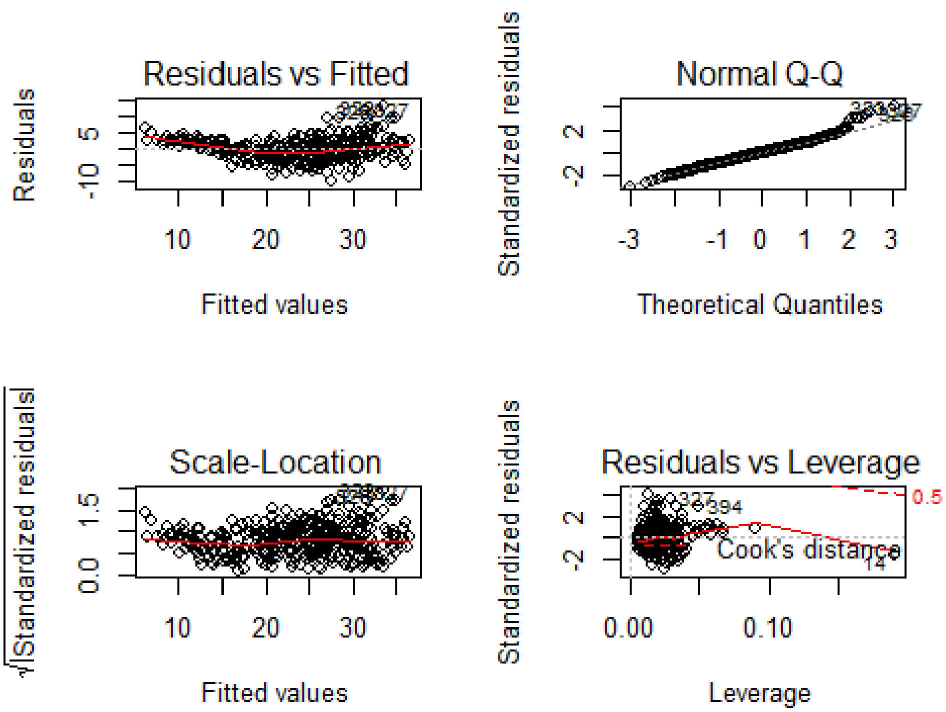
```
## weight       -0.8322442  0.8975273    0.9329944  0.8645377  1.0000000
## acceleration  0.4233285 -0.5046834   -0.5438005 -0.6891955 -0.4168392
## year          0.5805410 -0.3456474   -0.3698552 -0.4163615 -0.3091199
## origin        0.5652088 -0.5689316   -0.6145351 -0.4551715 -0.5850054
##              acceleration       year     origin
## mpg             0.4233285  0.5805410  0.5652088
## cylinders      -0.5046834 -0.3456474 -0.5689316
## displacement   -0.5438005 -0.3698552 -0.6145351
## horsepower     -0.6891955 -0.4163615 -0.4551715
## weight         -0.4168392 -0.3091199 -0.5850054
## acceleration    1.0000000  0.2903161  0.2127458
## year            0.2903161  1.0000000  0.1815277
## origin          0.2127458  0.1815277  1.0000000
```

9c

```
auto.fit <- lm(mpg~.-name, data = Auto)
summary(auto.fit)

##
## Call:
## lm(formula = mpg ~ . - name, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -9.5903 -2.1565 -0.1169  1.8690 13.0604
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -17.218435   4.644294  -3.707  0.00024 ***
## cylinders     -0.493376   0.323282  -1.526  0.12780
## displacement   0.019896   0.007515   2.647  0.00844 **
## horsepower    -0.016951   0.013787  -1.230  0.21963
## weight        -0.006474   0.000652  -9.929  < 2e-16 ***
## acceleration   0.080576   0.098845   0.815  0.41548
## year           0.750773   0.050973  14.729  < 2e-16 ***
## origin         1.426141   0.278136   5.127 4.67e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.328 on 384 degrees of freedom
## Multiple R-squared:  0.8215, Adjusted R-squared:  0.8182
## F-statistic: 252.4 on 7 and 384 DF,  p-value: < 2.2e-16
```

1. There is a statistically significant association between MPG and displacement, weight, year, and origin.
2. Displacement, weight, year, and origin.
3. Newer cars have higher MPG: average MPG improves by 0.7508 per year. #9d

```r
par(mfrow=c(2,2))
plot(auto.fit)
```



The residuals plot suggests several outliers, as does the QQ plot. The leverage plot identifies obs. 327, 394, and 14 as having high leverage. The residuals vs. fitted hints at a missing higher-order (quadratic) term.

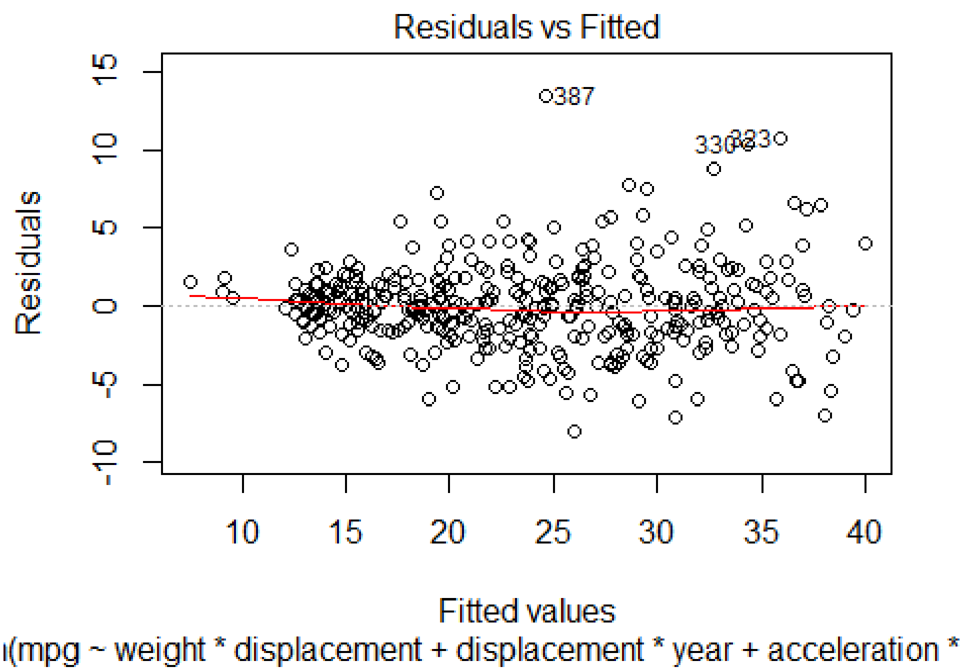
9e

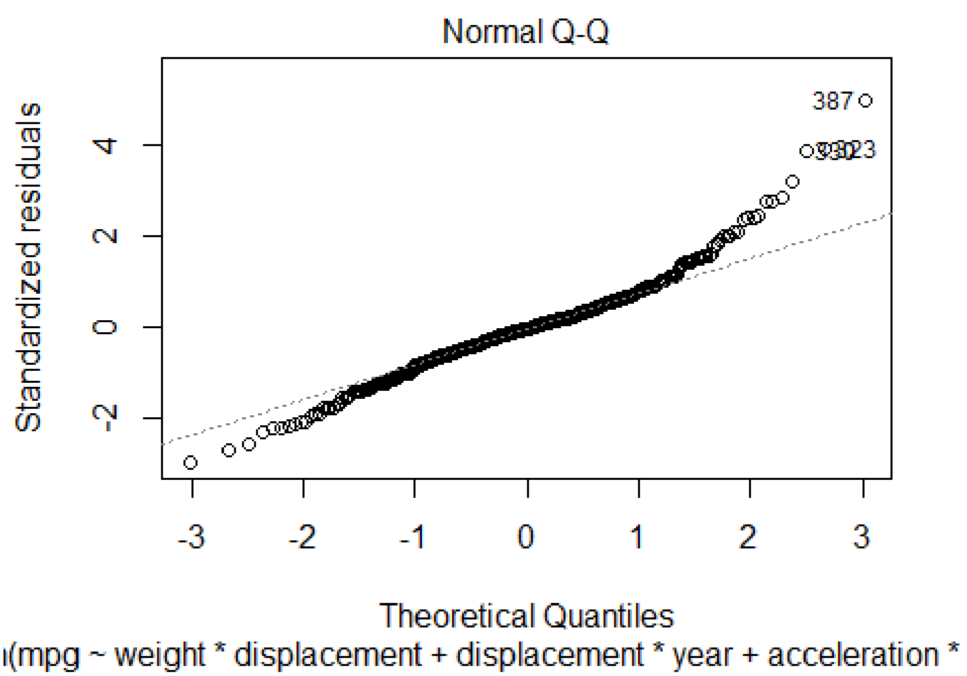
```r
auto.fit2 <- lm(mpg~weight*displacement + displacement*year +
acceleration*horsepower + acceleration*horsepower*origin, data = Auto)
summary(auto.fit2)
```

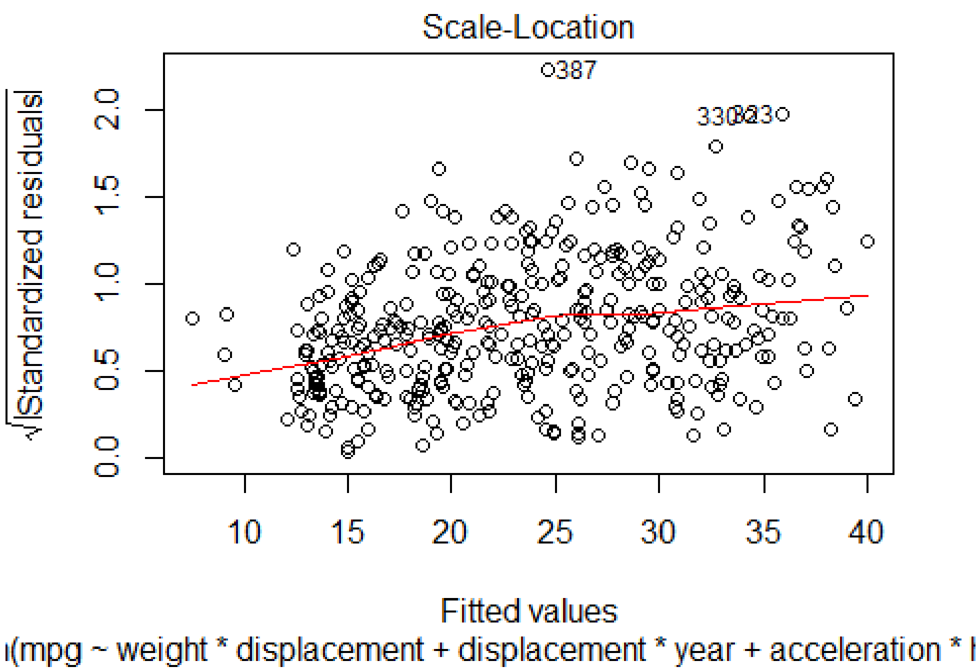
```
##
## Call:
## lm(formula = mpg ~ weight * displacement + displacement * year +
##     acceleration * horsepower + acceleration * horsepower * origin,
##     data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -7.9877 -1.5233 -0.0496  1.3267 13.4200
##
## Coefficients:
##                                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     -9.979e+00  9.738e+00  -1.025 0.306147
## weight                          -8.223e-03  9.638e-04  -8.531 3.52e-16 ***
```

```
## displacement                      1.043e-01  3.838e-02   2.716 0.006902 **
## year                              1.134e+00  9.182e-02  12.353  < 2e-16 ***
## acceleration                     -1.562e+00  4.604e-01  -3.393 0.000765 ***
## horsepower                       -2.407e-01  7.416e-02  -3.245 0.001279 **
## origin                           -2.020e+01  5.239e+00  -3.855 0.000136 ***
## weight:displacement               1.699e-05  2.617e-06   6.493 2.64e-10 ***
## displacement:year                -2.158e-03  4.835e-04  -4.464 1.06e-05 ***
## acceleration:horsepower           1.286e-02  4.831e-03   2.661 0.008119 **
## acceleration:origin               1.328e+00  3.154e-01   4.212 3.17e-05 ***
## horsepower:origin                 1.933e-01  5.787e-02   3.341 0.000918 ***
## acceleration:horsepower:origin -1.274e-02  3.755e-03  -3.392 0.000767 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.768 on 379 degrees of freedom
## Multiple R-squared:  0.8781, Adjusted R-squared:  0.8742
## F-statistic: 227.4 on 12 and 379 DF,  p-value: < 2.2e-16
```

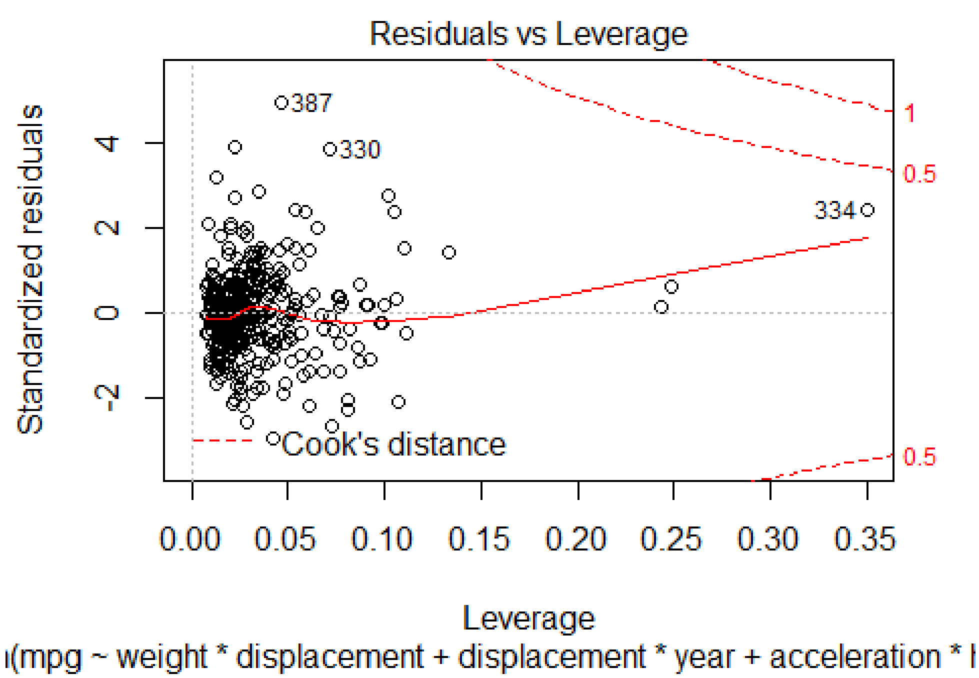
```
plot(auto.fit2)
```


Residuals vs Fitted
Residuals
Fitted values
ı(mpg ~ weight * displacement + displacement * year + acceleration * I
387
330
323
15
10
5
0
-5
-10
10
15
20
25
30
35
40

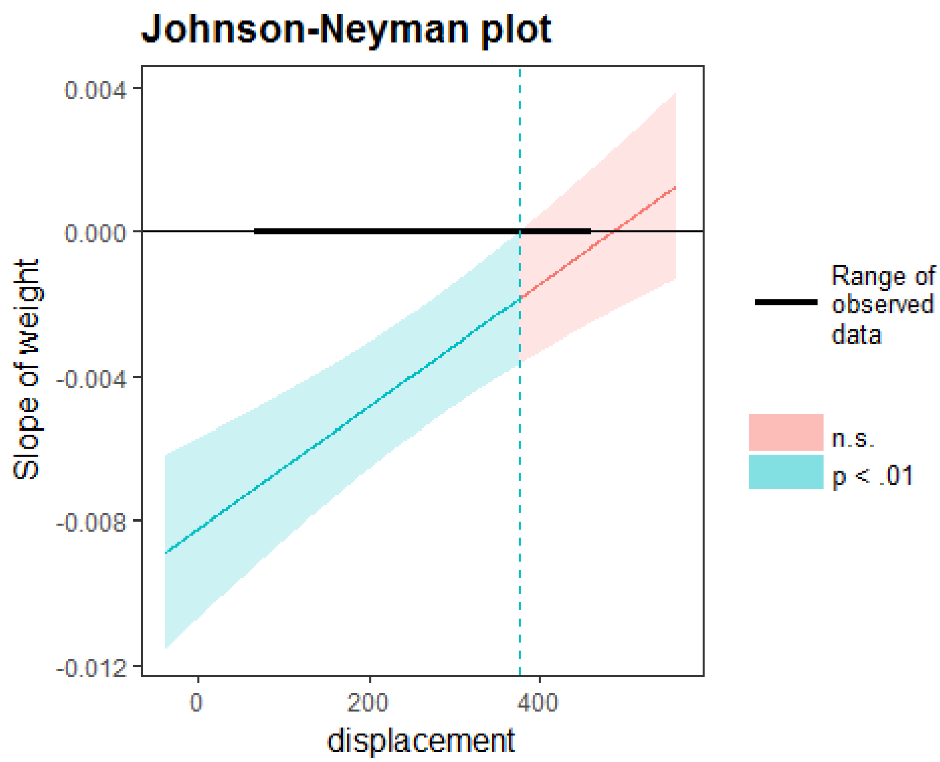

Normal Q-Q
Standardized residuals
Theoretical Quantiles
ı(mpg ~ weight * displacement + displacement * year + acceleration * I
387
330
323
4
2
0
-2
-3
-2
-1
0
1
2
3



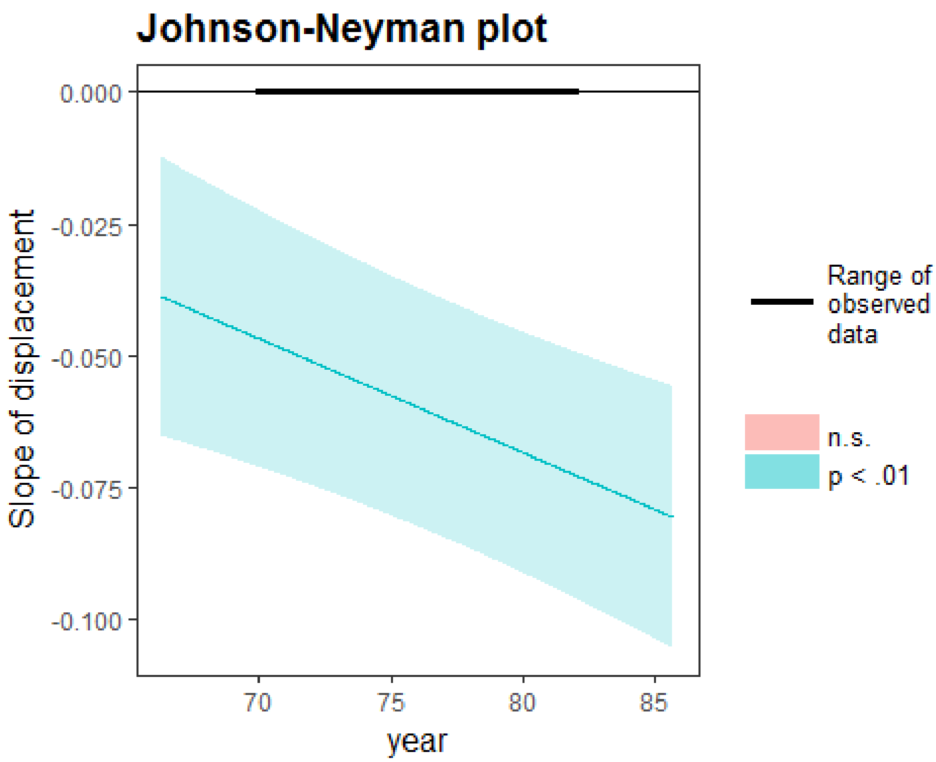

```
library(jtools)

## Warning: package 'jtools' was built under R version 3.4.3
```



```
johnson_neyman(auto.fit2, pred = weight, modx = displacement, alpha = 0.01)

## JOHNSON-NEYMAN INTERVAL
##
## The slope of weight is p < .01 when displacement is OUTSIDE this interval:
## [376.1, 672.31]
## Note: The range of observed values of displacement is [68, 455]
```

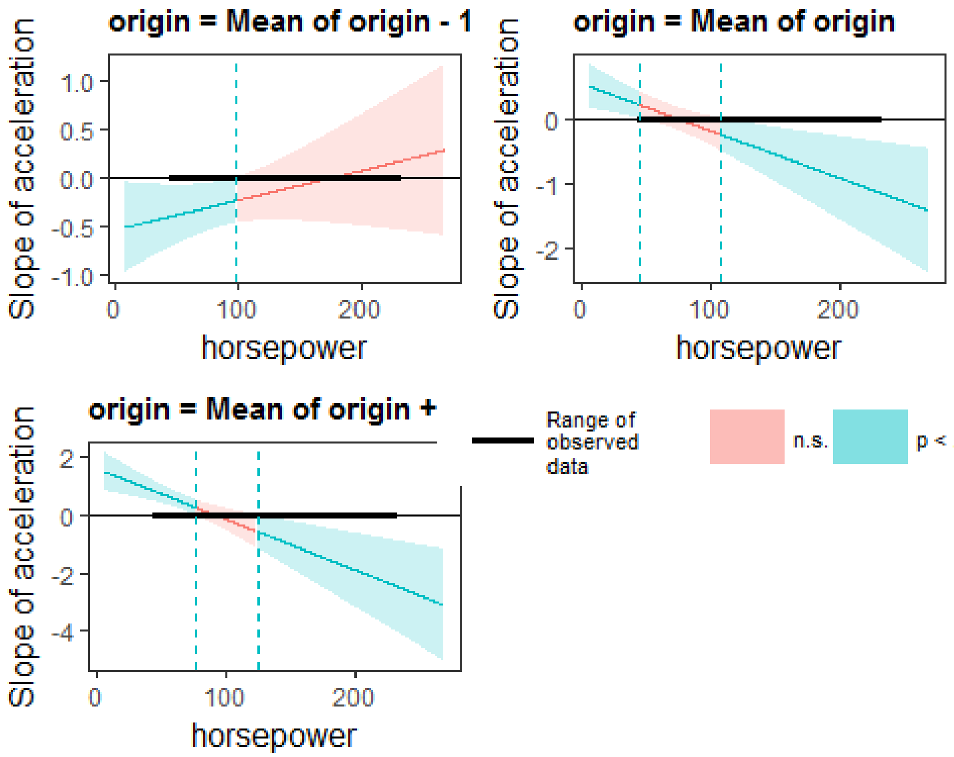



```
johnson_neyman(auto.fit2, pred = displacement, modx = year, alpha = 0.01)

## JOHNSON-NEYMAN INTERVAL
##
## The slope of displacement is p < .01 when year is OUTSIDE this interval:
## [5.2, 62.02]
## Note: The range of observed values of year is [70, 82]
```



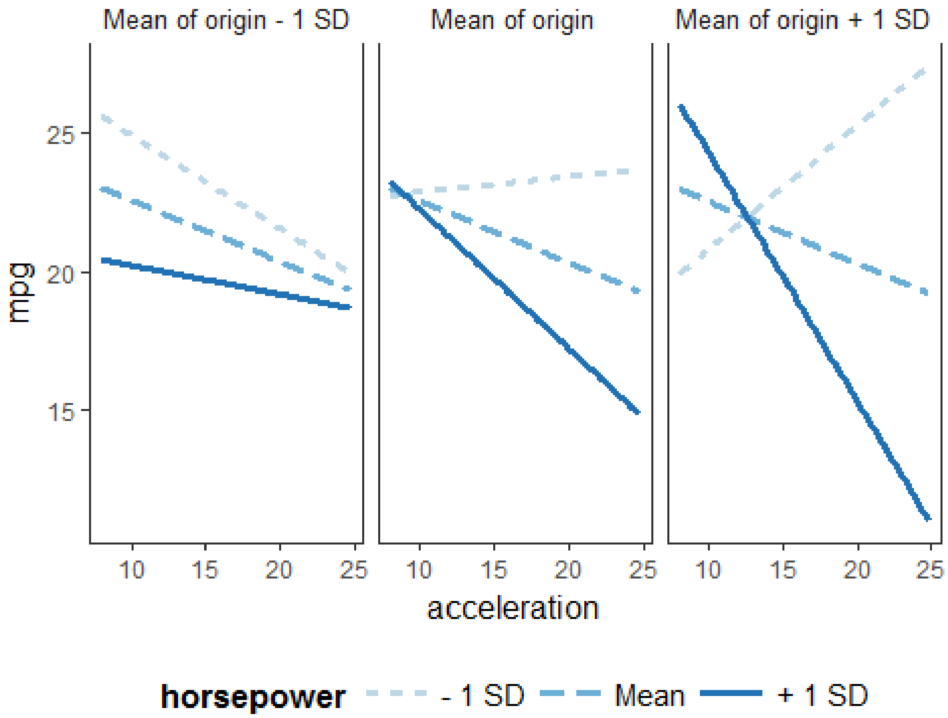

```
sim_slopes(auto.fit2, pred = acceleration, modx = horsepower, mod2 = origin,
jnplot = TRUE)

## ###############################################################
## While origin (2nd moderator) = 0.77 (Mean of origin - 1 SD)
## ###############################################################
##
## JOHNSON-NEYMAN INTERVAL
##
## The slope of acceleration is p < .05 when horsepower is INSIDE this
interval:
## [-32.86, 98.95]
## Note: The range of observed values of horsepower is [46, 230]
##
## SIMPLE SLOPES ANALYSIS
##
## Slope of acceleration when horsepower = 65.98 (- 1 SD):
##  Est.  S.E.     p
## -0.34  0.13  0.01
##
## Slope of acceleration when horsepower = 104.47 (Mean):
##  Est.  S.E.     p
## -0.22  0.12  0.07
##
## Slope of acceleration when horsepower = 142.96 (+ 1 SD):
##  Est.  S.E.     p
## -0.10  0.17  0.55
```

```
##
## ############################################################
## While origin (2nd moderator) = 1.58 (Mean of origin)
## ############################################################
##
## JOHNSON-NEYMAN INTERVAL
##
## The slope of acceleration is p < .05 when horsepower is OUTSIDE this
interval:
## [45.43, 108.31]
## Note: The range of observed values of horsepower is [46, 230]
##
## SIMPLE SLOPES ANALYSIS
##
## Slope of acceleration when horsepower = 65.98 (- 1 SD):
## Est. S.E.    p
## 0.06 0.09 0.53
##
## Slope of acceleration when horsepower = 104.47 (Mean):
##  Est.  S.E.     p
## -0.22  0.12  0.07
##
## Slope of acceleration when horsepower = 142.96 (+ 1 SD):
##  Est.  S.E.     p
## -0.50  0.20  0.01
##
## ############################################################
## While origin (2nd moderator) = 2.38 (Mean of origin + 1 SD)
## ############################################################
##
## JOHNSON-NEYMAN INTERVAL
##
## The slope of acceleration is p < .05 when horsepower is OUTSIDE this
interval:
## [76.49, 124.73]
## Note: The range of observed values of horsepower is [46, 230]
##
## SIMPLE SLOPES ANALYSIS
##
## Slope of acceleration when horsepower = 65.98 (- 1 SD):
## Est. S.E.    p
## 0.45 0.13 0.00
##
## Slope of acceleration when horsepower = 104.47 (Mean):
##  Est.  S.E.     p
## -0.22  0.21  0.29
##
## Slope of acceleration when horsepower = 142.96 (+ 1 SD):
##  Est.  S.E.     p
## -0.90  0.38  0.02
```

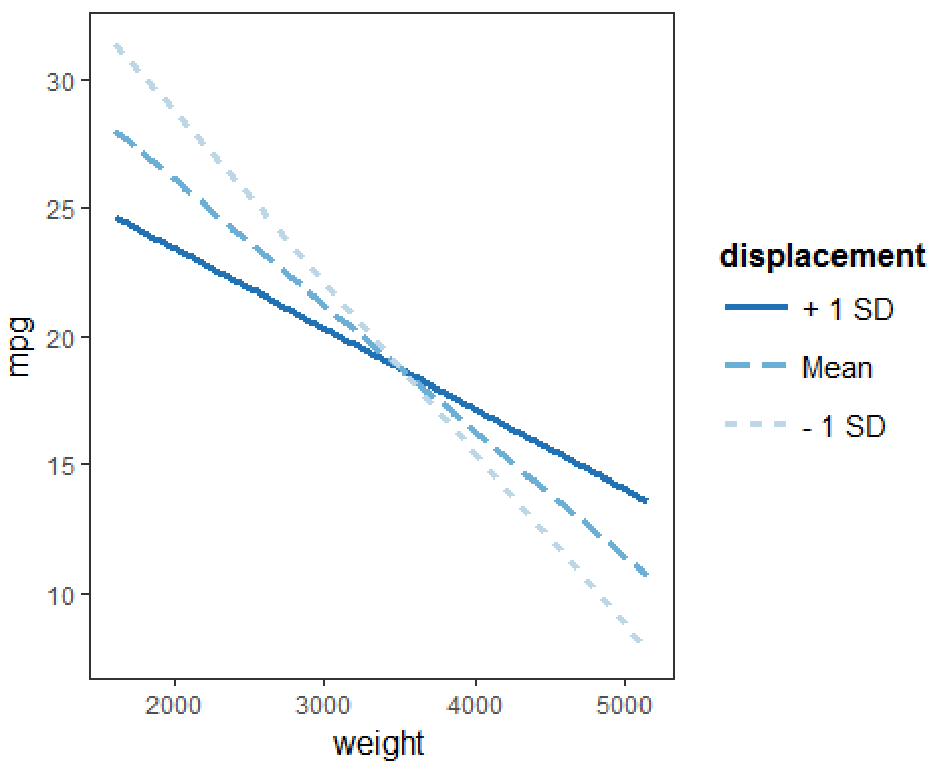



```
probe_interaction(auto.fit2, pred = acceleration, modx = horsepower, mod2 =
origin)

## ##############################################################
## While origin (2nd moderator) = 0.77 (Mean of origin - 1 SD)
## ##############################################################
##
## JOHNSON-NEYMAN INTERVAL
##
## The slope of acceleration is p < .05 when horsepower is INSIDE this
interval:
## [-32.86, 98.95]
## Note: The range of observed values of horsepower is [46, 230]
##
## SIMPLE SLOPES ANALYSIS
##
## Slope of acceleration when horsepower = 65.98 (- 1 SD):
##  Est.  S.E.     p
## -0.34  0.13  0.01
##
## Slope of acceleration when horsepower = 104.47 (Mean):
##  Est.  S.E.     p
## -0.22  0.12  0.07
##
## Slope of acceleration when horsepower = 142.96 (+ 1 SD):
##  Est.  S.E.     p
## -0.10  0.17  0.55
```

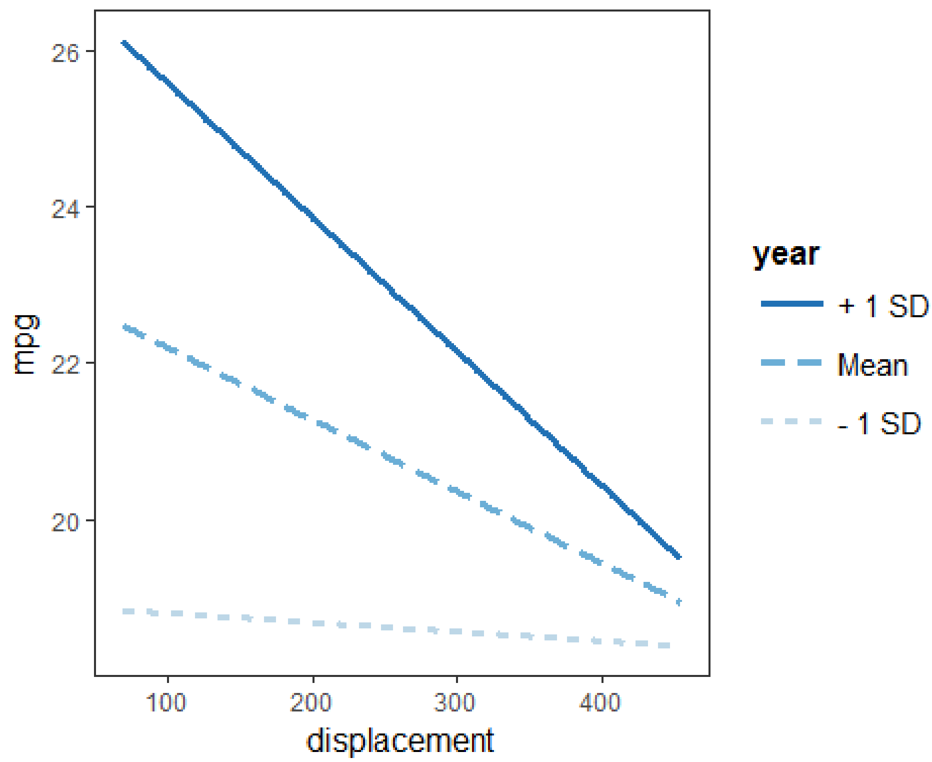
```
##
## ###############################################################
## While origin (2nd moderator) = 1.58 (Mean of origin)
## ###############################################################
##
## JOHNSON-NEYMAN INTERVAL
##
## The slope of acceleration is p < .05 when horsepower is OUTSIDE this
interval:
## [45.43, 108.31]
## Note: The range of observed values of horsepower is [46, 230]
##
## SIMPLE SLOPES ANALYSIS
##
## Slope of acceleration when horsepower = 65.98 (- 1 SD):
## Est. S.E.    p
## 0.06 0.09 0.53
##
## Slope of acceleration when horsepower = 104.47 (Mean):
##  Est.  S.E.     p
## -0.22  0.12  0.07
##
## Slope of acceleration when horsepower = 142.96 (+ 1 SD):
##  Est.  S.E.     p
## -0.50  0.20  0.01
##
## ###############################################################
## While origin (2nd moderator) = 2.38 (Mean of origin + 1 SD)
## ###############################################################
##
## JOHNSON-NEYMAN INTERVAL
##
## The slope of acceleration is p < .05 when horsepower is OUTSIDE this
interval:
## [76.49, 124.73]
## Note: The range of observed values of horsepower is [46, 230]
##
## SIMPLE SLOPES ANALYSIS
##
## Slope of acceleration when horsepower = 65.98 (- 1 SD):
## Est. S.E.    p
## 0.45 0.13 0.00
##
## Slope of acceleration when horsepower = 104.47 (Mean):
##  Est.  S.E.     p
## -0.22  0.21  0.29
##
## Slope of acceleration when horsepower = 142.96 (+ 1 SD):
##  Est.  S.E.     p
## -0.90  0.38  0.02
```



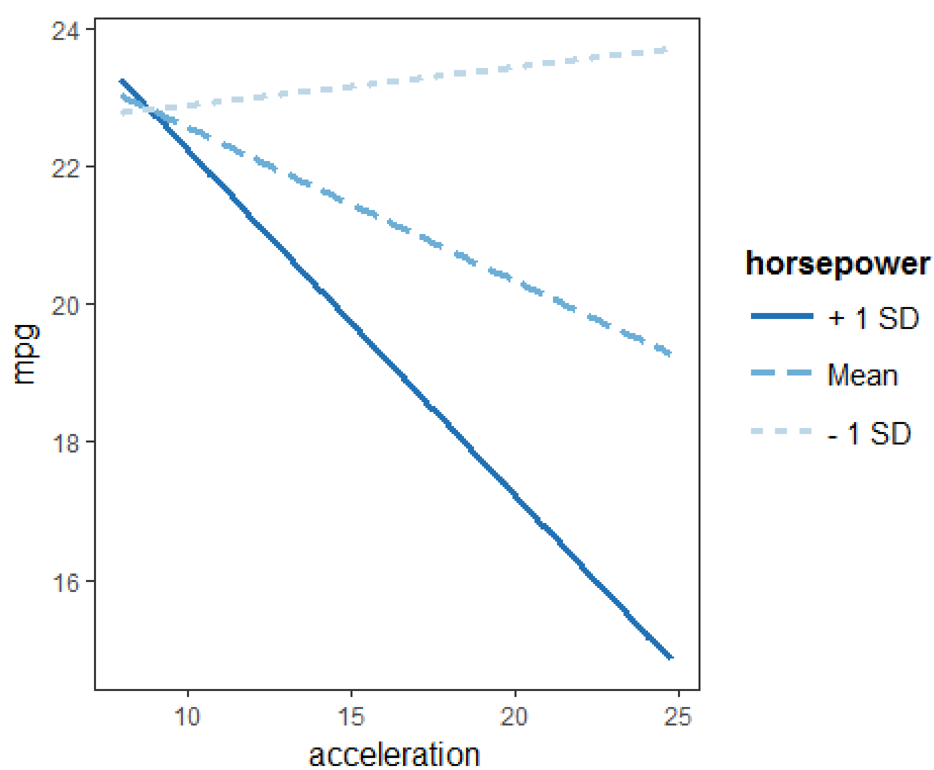

```r
interact_plot(auto.fit2, pred = "weight", modx = "displacement")
```




```r
interact_plot(auto.fit2, pred = "displacement", modx = "year")
```




```r
interact_plot(auto.fit2, pred = "acceleration", modx = "horsepower")
```



Weight by displacement, displacement by year, acceleration by horsepower, horsepower by origin and acceleration by horsepower by origin are statistically significant interactions. Normally

I would not try to fit so many interaction terms or even three way interactions as they are very hard to interpret; however, I wanted to take this opportunity to learn more about R's interaction plots and Johnson-Neyman plots which may help to interpret said interactions. The latter are very hard to make in SAS.

9f

```
auto.fit3 <- lm(mpg~log(weight)+sqrt(displacement)+I(horsepower^2), data =
Auto)
summary(auto.fit3)
```

```
##
## Call:
## lm(formula = mpg ~ log(weight) + sqrt(displacement) + I(horsepower^2),
##     data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.769  -2.764  -0.448   2.095  16.184
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        1.639e+02  1.474e+01  11.124  < 2e-16 ***
## log(weight)       -1.688e+01  2.104e+00  -8.025 1.22e-14 ***
## sqrt(displacement) -3.959e-01  1.849e-01  -2.142   0.0329 *
## I(horsepower^2)   -6.387e-05  3.904e-05  -1.636   0.1026
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.127 on 388 degrees of freedom
## Multiple R-squared:  0.7225, Adjusted R-squared:  0.7204
## F-statistic: 336.8 on 3 and 388 DF,  p-value: < 2.2e-16
```

```
anova(auto.fit3, auto.fit)
```

```
## Analysis of Variance Table
##
## Model 1: mpg ~ log(weight) + sqrt(displacement) + I(horsepower^2)
## Model 2: mpg ~ (cylinders + displacement + horsepower + weight +
acceleration +
##     year + origin + name) - name
##   Res.Df    RSS Df Sum of Sq      F    Pr(>F)
## 1    388 6609.5
## 2    384 4252.2  4    2357.3 53.219 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The log weight of a car and the square root of its displacement have a statistically significant relationship with MPG. Horsepower^2 is not statistically associated with MPG.

This model, however, does not seem to out perform the orginal in terms of variance explained (as determined by the smaller RSS)

15

Please forgive the clumsy approach to question 15. I am new to r.

```
library(MASS)

## Warning: package 'MASS' was built under R version 3.4.3

library(dplyr)

## Warning: package 'dplyr' was built under R version 3.4.3

##
## Attaching package: 'dplyr'

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

names(Boston)

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

summary(Boston)

##       crim                zn             indus            chas
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       nox               rm             age              dis
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
```

```
##       rad              tax           ptratio          black       
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32  
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38  
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44  
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67  
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23  
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90  
##      lstat            medv      
##  Min.   : 1.73   Min.   : 5.00  
##  1st Qu.: 6.95   1st Qu.:17.02  
##  Median :11.36   Median :21.20  
##  Mean   :12.65   Mean   :22.53  
##  3rd Qu.:16.95   3rd Qu.:25.00  
##  Max.   :37.97   Max.   :50.00
```

```r
glimpse(Boston)
```

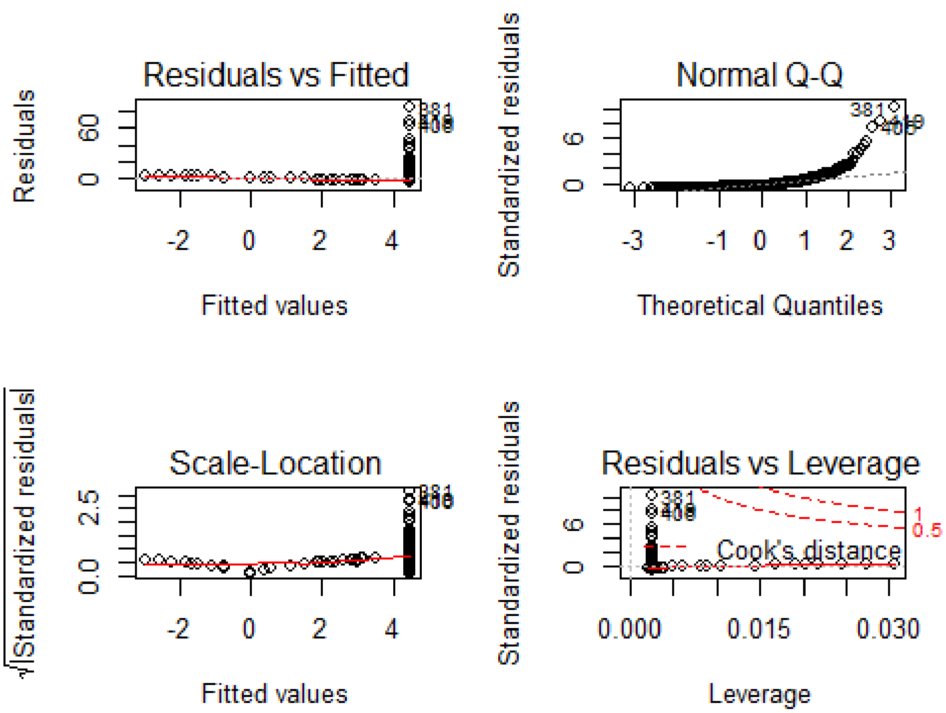
```
## Observations: 506
## Variables: 14
## $ crim    <dbl> 0.00632, 0.02731, 0.02729, 0.03237, 0.06905, 0.02985, ...
## $ zn      <dbl> 18.0, 0.0, 0.0, 0.0, 0.0, 0.0, 12.5, 12.5, 12.5, 12.5,...
## $ indus   <dbl> 2.31, 7.07, 7.07, 2.18, 2.18, 2.18, 7.87, 7.87, 7.87, ...
## $ chas    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ nox     <dbl> 0.538, 0.469, 0.469, 0.458, 0.458, 0.458, 0.524, 0.524...
## $ rm      <dbl> 6.575, 6.421, 7.185, 6.998, 7.147, 6.430, 6.012, 6.172...
## $ age     <dbl> 65.2, 78.9, 61.1, 45.8, 54.2, 58.7, 66.6, 96.1, 100.0,...
## $ dis     <dbl> 4.0900, 4.9671, 4.9671, 6.0622, 6.0622, 6.0622, 5.5605...
## $ rad     <int> 1, 2, 2, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4, 4, ...
## $ tax     <dbl> 296, 242, 242, 222, 222, 222, 311, 311, 311, 311, 311,...
## $ ptratio <dbl> 15.3, 17.8, 17.8, 18.7, 18.7, 18.7, 15.2, 15.2, 15.2, ...
## $ black   <dbl> 396.90, 396.90, 392.83, 394.63, 396.90, 394.12, 395.60...
## $ lstat   <dbl> 4.98, 9.14, 4.03, 2.94, 5.33, 5.21, 12.43, 19.15, 29.9...
## $ medv    <dbl> 24.0, 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, ...
```

```r
lm.zn <- lm(crim~zn, data = Boston)
summary(lm.zn)
```

```
## 
## Call:
## lm(formula = crim ~ zn, data = Boston)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -4.429 -4.222 -2.620  1.250 84.523 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  4.45369    0.41722  10.675  < 2e-16 ***
## zn          -0.07393    0.01609  -4.594 5.51e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 8.435 on 504 degrees of freedom
## Multiple R-squared:  0.04019,    Adjusted R-squared:  0.03828
## F-statistic:  21.1 on 1 and 504 DF,  p-value: 5.506e-06
```

```
par(mfrow=c(2,2))
plot(lm.zn)
```

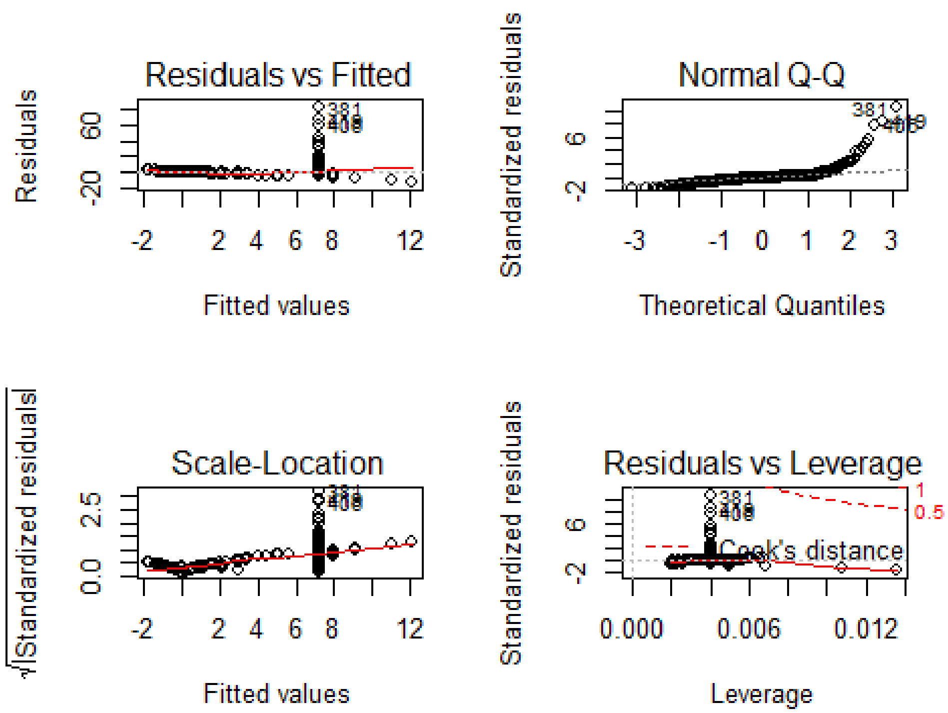


zn is significantly associated with crime; however, zn accounts for only 4% of the variance in crim. The plots indicate the presence of outliers.

```
lm.indus <- lm(crim~indus, data = Boston)
summary(lm.indus)
```

```
##
## Call:
## lm(formula = crim ~ indus, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.972  -2.698  -0.736   0.712  81.813
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -2.06374    0.66723  -3.093  0.00209 **
## indus        0.50978    0.05102   9.991  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 7.866 on 504 degrees of freedom
## Multiple R-squared:  0.1653, Adjusted R-squared:  0.1637
## F-statistic: 99.82 on 1 and 504 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(lm.indus)
```



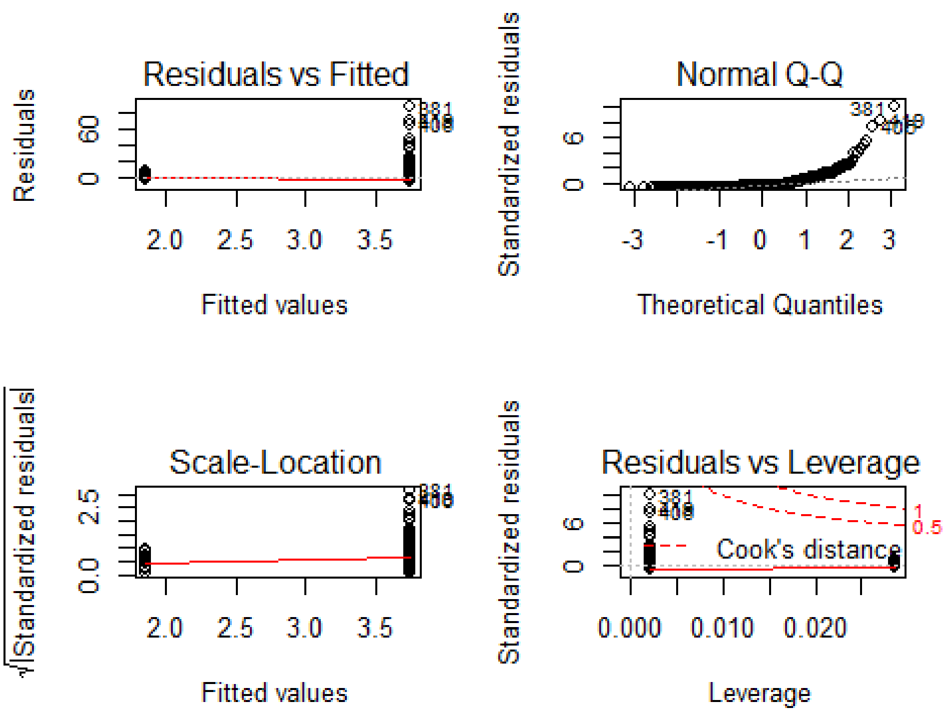
Indus is significant.

Though it results in a better fit, there are still a number of outliers.

```
lm.chas <- lm(crim~chas, data = Boston)
summary(lm.chas)

##
## Call:
## lm(formula = crim ~ chas, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -3.738 -3.661 -3.435  0.018 85.232
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   3.7444     0.3961   9.453   <2e-16 ***
## chas         -1.8928     1.5061  -1.257    0.209
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 8.597 on 504 degrees of freedom
## Multiple R-squared:  0.003124,	Adjusted R-squared:  0.001146
## F-statistic: 1.579 on 1 and 504 DF,  p-value: 0.2094
```

```
par(mfrow=c(2,2))
plot(lm.chas)
```

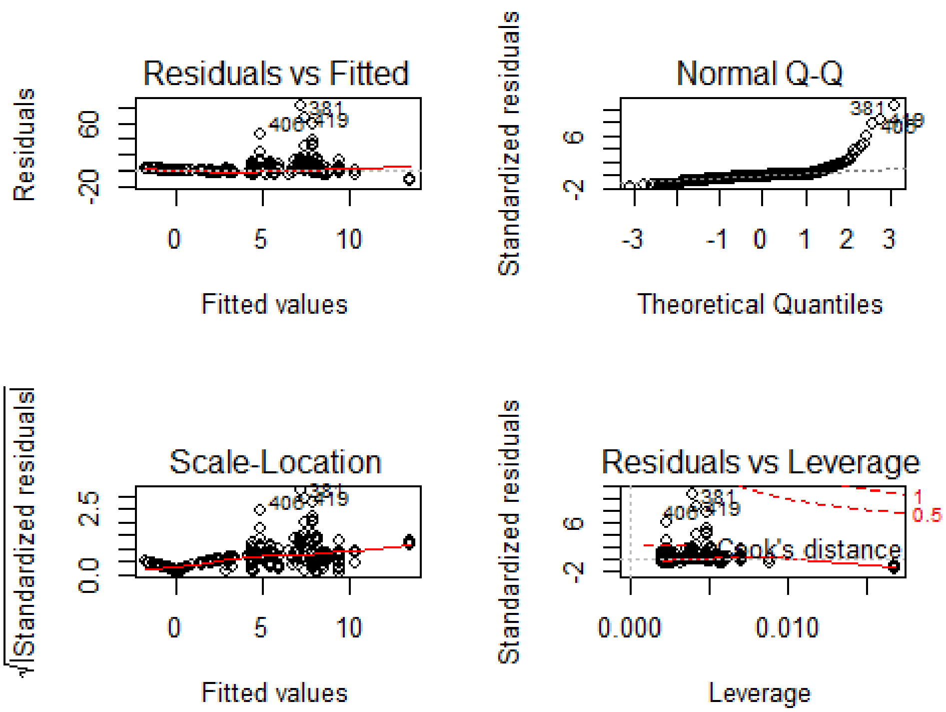


There is no evidence to support an statistically significant association between chas and crime.

```
lm.nox <- lm(crim~nox, data = Boston)
summary(lm.nox)
```

```
##
## Call:
## lm(formula = crim ~ nox, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.371  -2.738  -0.974   0.559  81.728
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -13.720      1.699  -8.073 5.08e-15 ***
## nox           31.249      2.999  10.419  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.81 on 504 degrees of freedom
```

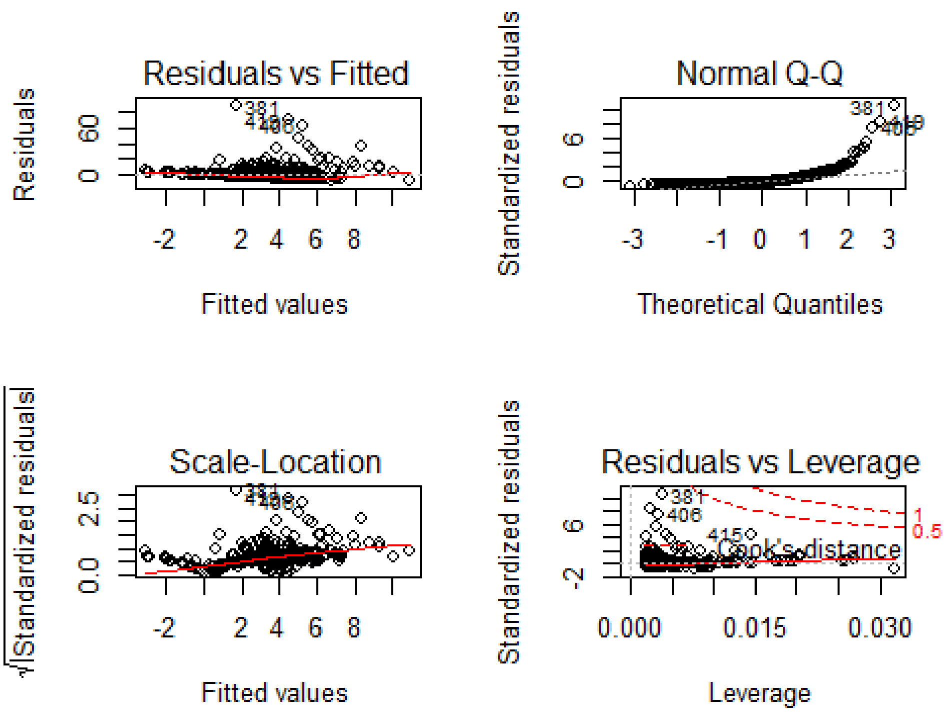
```
## Multiple R-squared:  0.1772, Adjusted R-squared:  0.1756 
## F-statistic: 108.6 on 1 and 504 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(lm.nox)
```



nox is significantly associated with crime. As with the above, though this is the best fit yet (r2=.177), there still are a number of outliers.

```
lm.rm  <- lm(crim~rm, data = Boston)
summary(lm.rm )

## 
## Call:
## lm(formula = crim ~ rm, data = Boston)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -6.604 -3.952 -2.654  0.989 87.197 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept)   20.482      3.365   6.088 2.27e-09 ***
## rm            -2.684      0.532  -5.045 6.35e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 8.401 on 504 degrees of freedom
```

```
## Multiple R-squared:  0.04807,	Adjusted R-squared:  0.04618 
## F-statistic: 25.45 on 1 and 504 DF,  p-value: 6.347e-07

par(mfrow=c(2,2))
plot(lm.rm )
```

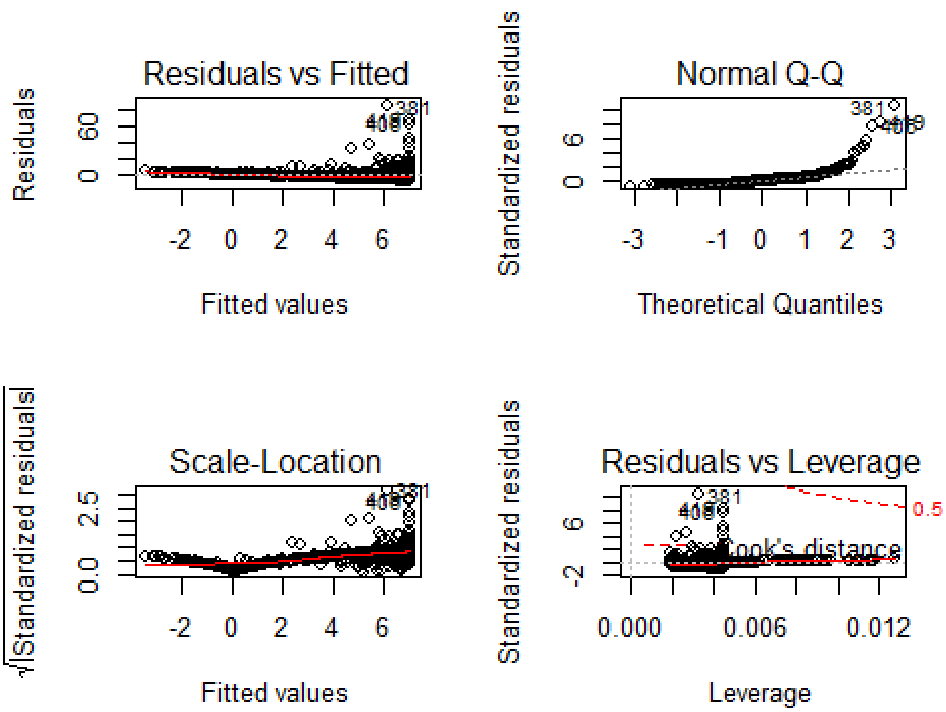


rm is significantly associated with crime, accounting for (only) 4.8% of the variance in crime.

```
lm.age  <- lm(crim~age, data = Boston)
summary(lm.age)

## 
## Call:
## lm(formula = crim ~ age, data = Boston)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -6.789 -4.257 -1.230  1.527 82.849 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -3.77791    0.94398  -4.002 7.22e-05 ***
## age          0.10779    0.01274   8.463 2.85e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 8.057 on 504 degrees of freedom
```

```
## Multiple R-squared:  0.1244, Adjusted R-squared:  0.1227
## F-statistic: 71.62 on 1 and 504 DF,  p-value: 2.855e-16
```

```
par(mfrow=c(2,2))
plot(lm.age)
```

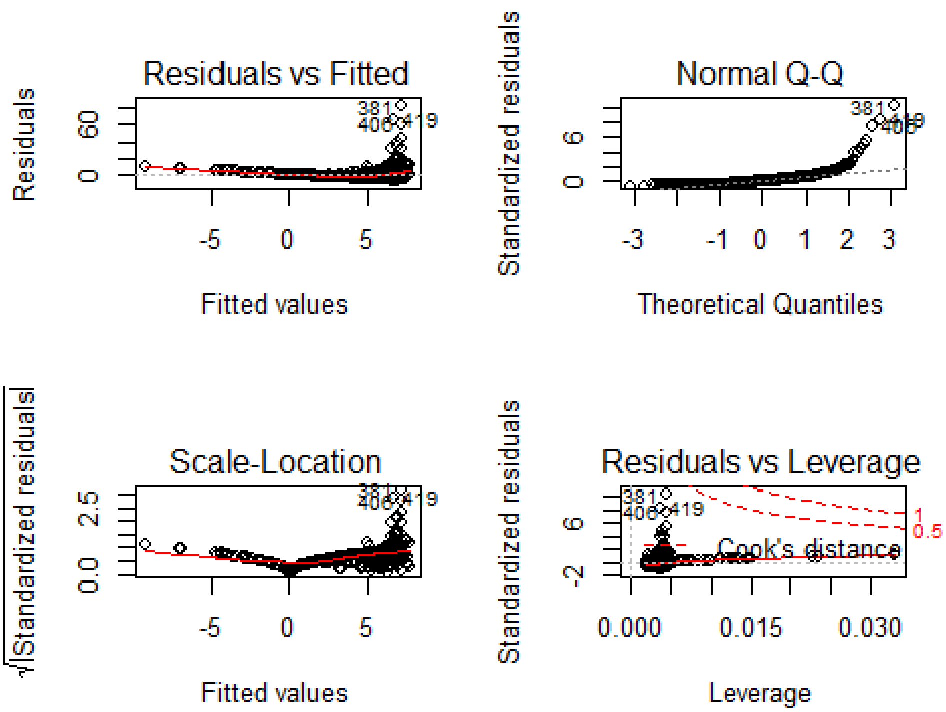


Age is significantly associated with crime.

```
lm.dis  <- lm(crim~dis, data = Boston)
summary(lm.dis)
```

```
##
## Call:
## lm(formula = crim ~ dis, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.708 -4.134 -1.527  1.516 81.674
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   9.4993     0.7304  13.006   <2e-16 ***
## dis          -1.5509     0.1683  -9.213   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.965 on 504 degrees of freedom
```

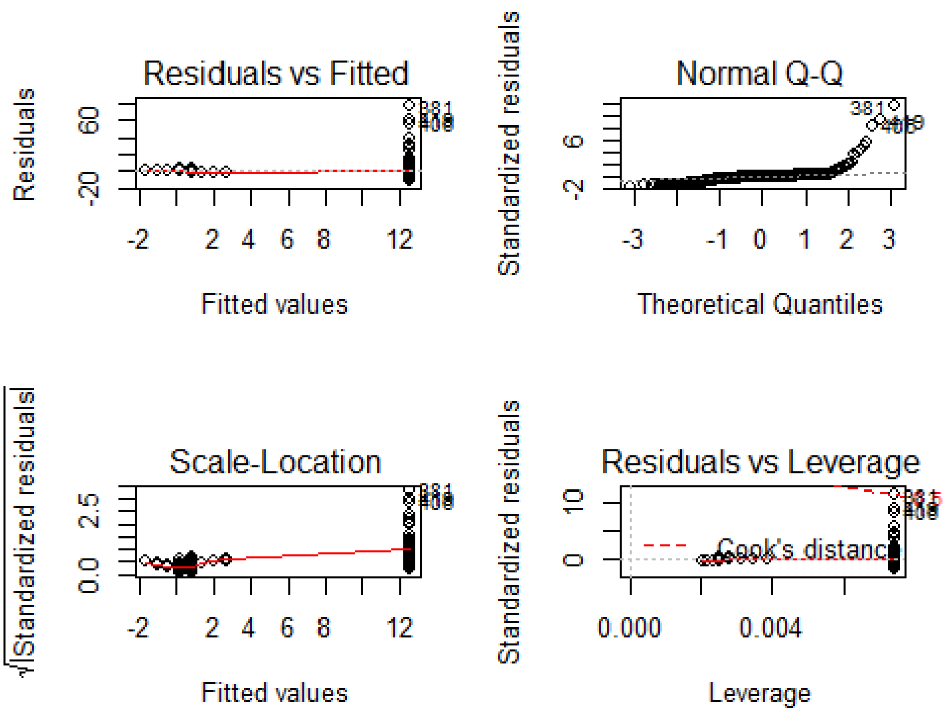
```
## Multiple R-squared:  0.1441, Adjusted R-squared:  0.1425
## F-statistic: 84.89 on 1 and 504 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(lm.dis)
```



Dis is significantly associated with crime.

```
lm.rad  <- lm(crim~rad, data = Boston)
summary(lm.rad)

## 
## Call:
## lm(formula = crim ~ rad, data = Boston)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -10.164  -1.381  -0.141   0.660  76.433 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -2.28716    0.44348  -5.157 3.61e-07 ***
## rad          0.61791    0.03433  17.998  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 6.718 on 504 degrees of freedom
```

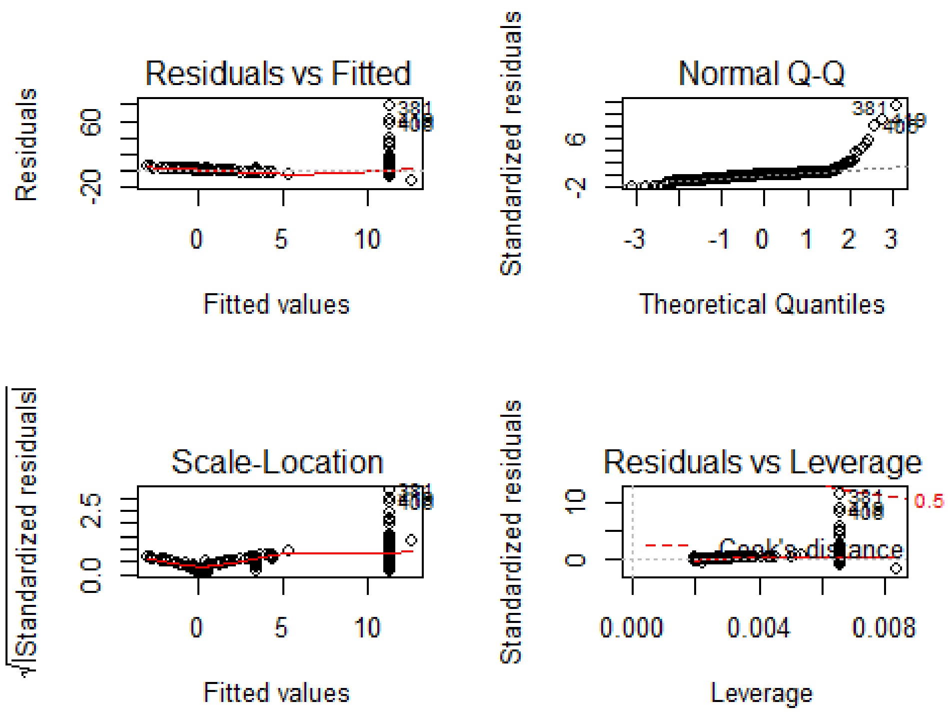
```
## Multiple R-squared:  0.3913, Adjusted R-squared:   0.39
## F-statistic: 323.9 on 1 and 504 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(lm.rad)
```



Rad is significantly associated with crime.

```
lm.tax  <- lm(crim~tax, data = Boston)
summary(lm.tax)

##
## Call:
## lm(formula = crim ~ tax, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -12.513  -2.738  -0.194   1.065  77.696
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -8.528369   0.815809  -10.45   <2e-16 ***
## tax          0.029742   0.001847   16.10   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.997 on 504 degrees of freedom
```

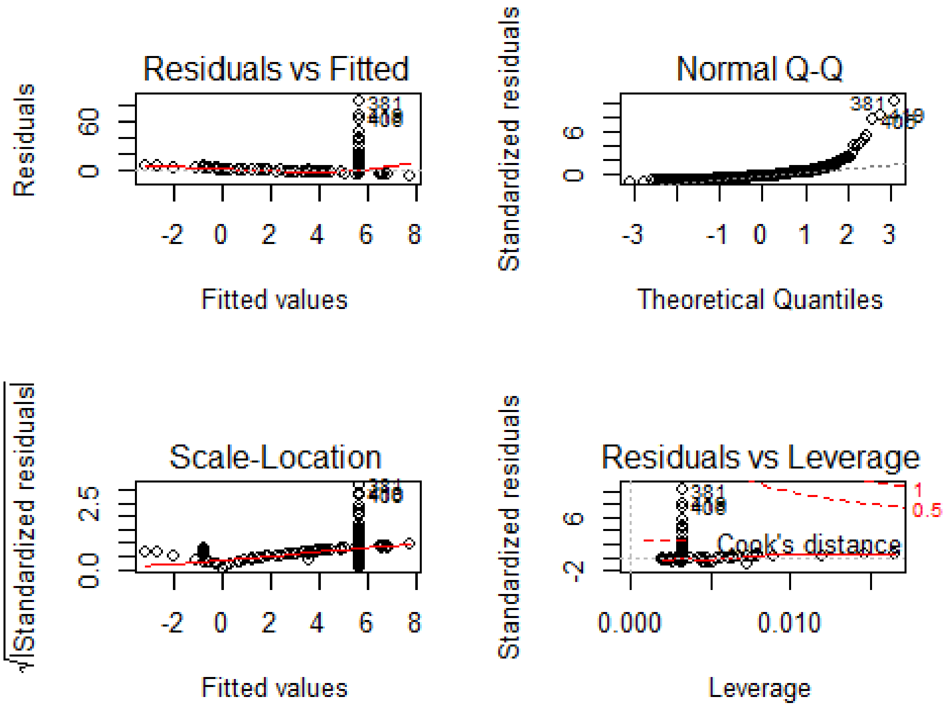
```
## Multiple R-squared:  0.3396, Adjusted R-squared:  0.3383
## F-statistic: 259.2 on 1 and 504 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(lm.tax)
```



Rad is significantly associated with crime.

```
lm.ptratio  <- lm(crim~ptratio, data = Boston)
summary(lm.ptratio)

##
## Call:
## lm(formula = crim ~ ptratio, data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -7.654 -3.985 -1.912  1.825 83.353
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -17.6469     3.1473  -5.607 3.40e-08 ***
## ptratio       1.1520     0.1694   6.801 2.94e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.24 on 504 degrees of freedom
```

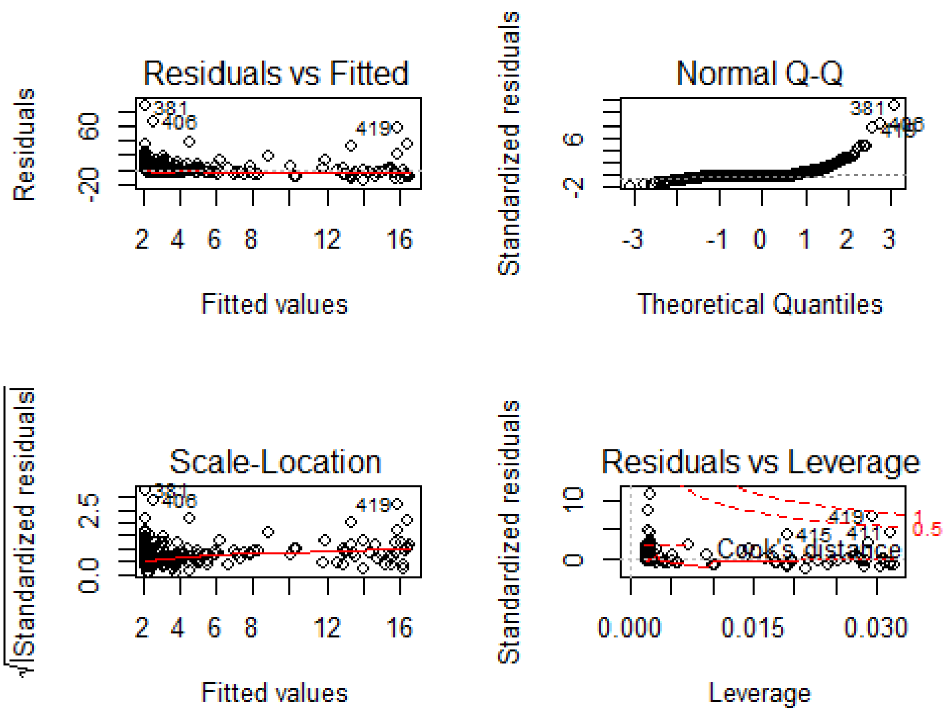
```
## Multiple R-squared:  0.08407,    Adjusted R-squared:  0.08225 
## F-statistic: 46.26 on 1 and 504 DF,  p-value: 2.943e-11

par(mfrow=c(2,2))
plot(lm.ptratio)
```



ptratio is significantly associated with crime.

```
lm.black  <- lm(crim~black, data = Boston)
summary(lm.black)

## 
## Call:
## lm(formula = crim ~ black, data = Boston)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -13.756  -2.299  -2.095  -1.296  86.822 
## 
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 16.553529   1.425903  11.609   <2e-16 ***
## black       -0.036280   0.003873  -9.367   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 7.946 on 504 degrees of freedom
```

```
## Multiple R-squared:  0.1483, Adjusted R-squared:  0.1466
## F-statistic: 87.74 on 1 and 504 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(lm.black)
```

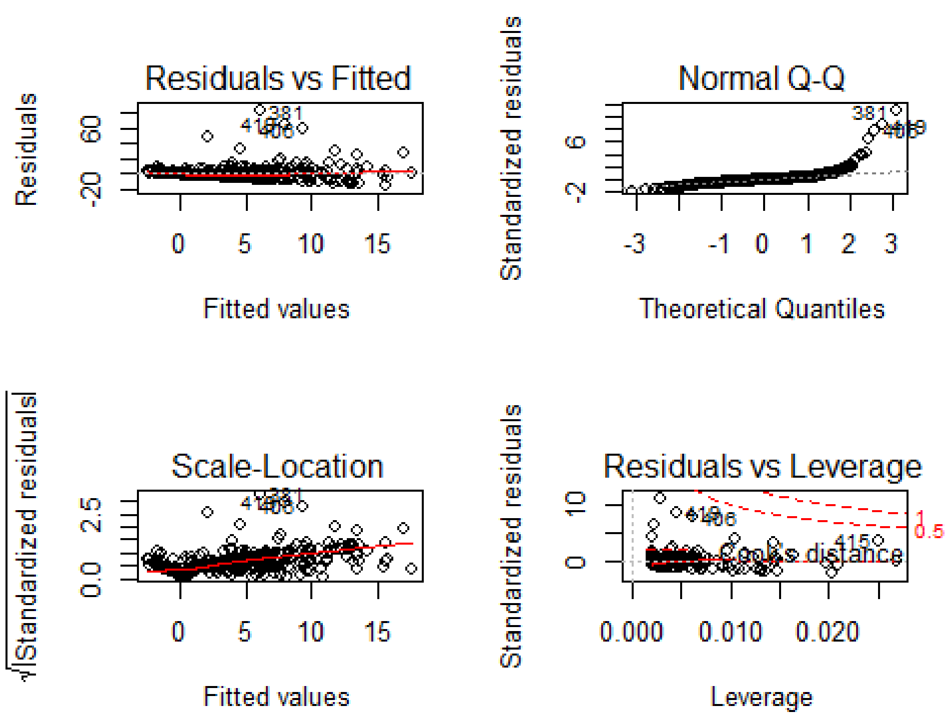


Black is significantly associated with crime.

```
lm.lstat  <- lm(crim~lstat, data = Boston)
summary(lm.lstat)

##
## Call:
## lm(formula = crim ~ lstat, data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.925  -2.822  -0.664   1.079  82.862
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -3.33054    0.69376  -4.801 2.09e-06 ***
## lstat        0.54880    0.04776  11.491  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.664 on 504 degrees of freedom
```

```
## Multiple R-squared:  0.2076, Adjusted R-squared:  0.206
## F-statistic:   132 on 1 and 504 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm.lstat)
```

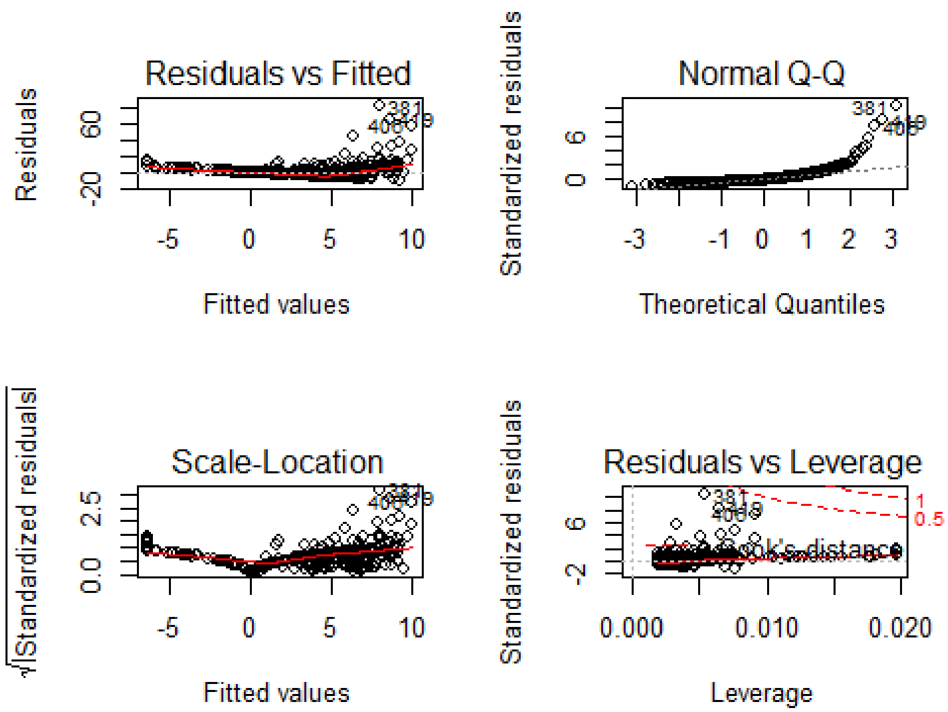


lstat is significantly associated with crime.

```
lm.medv <- lm(crim~medv, data = Boston)
summary(lm.medv)
```

```
##
## Call:
## lm(formula = crim ~ medv, data = Boston)
##
## Residuals:
##    Min      1Q Median      3Q     Max
## -9.071 -4.022 -2.343  1.298 80.957
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 11.79654    0.93419   12.63   <2e-16 ***
## medv        -0.36316    0.03839   -9.46   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.934 on 504 degrees of freedom
```

```
## Multiple R-squared:  0.1508, Adjusted R-squared:  0.1491
## F-statistic: 89.49 on 1 and 504 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(lm.medv)
```



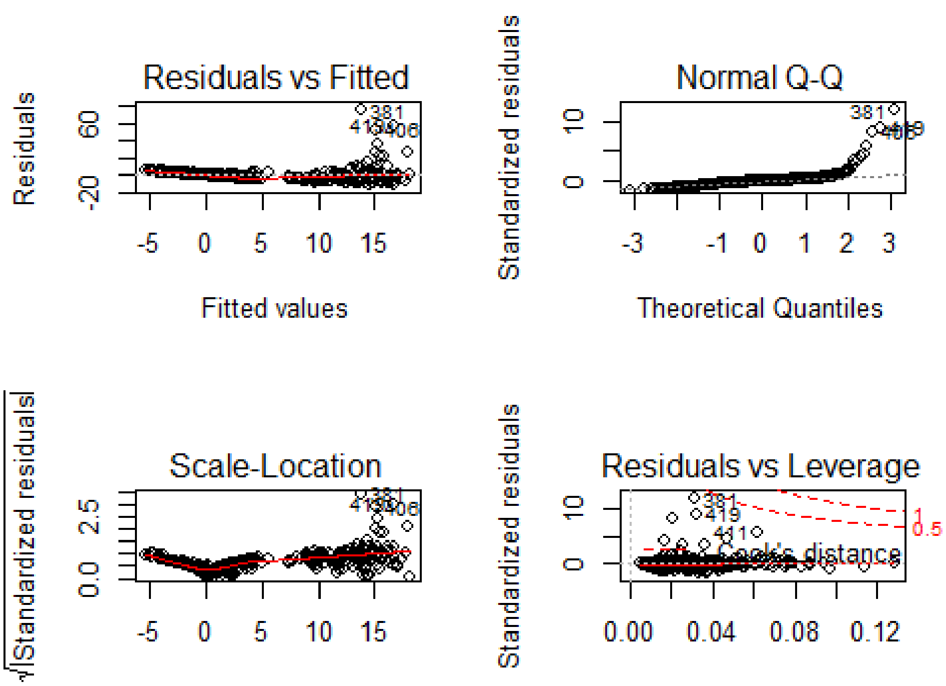
medv is significantly associated with crime.

In summation, using simple OLS, all IVs are significantly associated with crime excepting chas.

15b

```
lm.full <- lm(crim~., data = Boston)
summary(lm.full)

##
## Call:
## lm(formula = crim ~ ., data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -9.924 -2.120 -0.353  1.019 75.051
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  17.033228   7.234903   2.354 0.018949 *
```

```
## zn            0.044855   0.018734   2.394 0.017025 *
## indus        -0.063855   0.083407  -0.766 0.444294
## chas         -0.749134   1.180147  -0.635 0.525867
## nox         -10.313535   5.275536  -1.955 0.051152 .
## rm            0.430131   0.612830   0.702 0.483089
## age           0.001452   0.017925   0.081 0.935488
## dis          -0.987176   0.281817  -3.503 0.000502 ***
## rad           0.588209   0.088049   6.680 6.46e-11 ***
## tax          -0.003780   0.005156  -0.733 0.463793
## ptratio      -0.271081   0.186450  -1.454 0.146611
## black        -0.007538   0.003673  -2.052 0.040702 *
## lstat         0.126211   0.075725   1.667 0.096208 .
## medv         -0.198887   0.060516  -3.287 0.001087 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.439 on 492 degrees of freedom
## Multiple R-squared:  0.454,  Adjusted R-squared:  0.4396
## F-statistic: 31.47 on 13 and 492 DF,  p-value: < 2.2e-16

par(mfrow=c(2,2))
plot(lm.full)
```



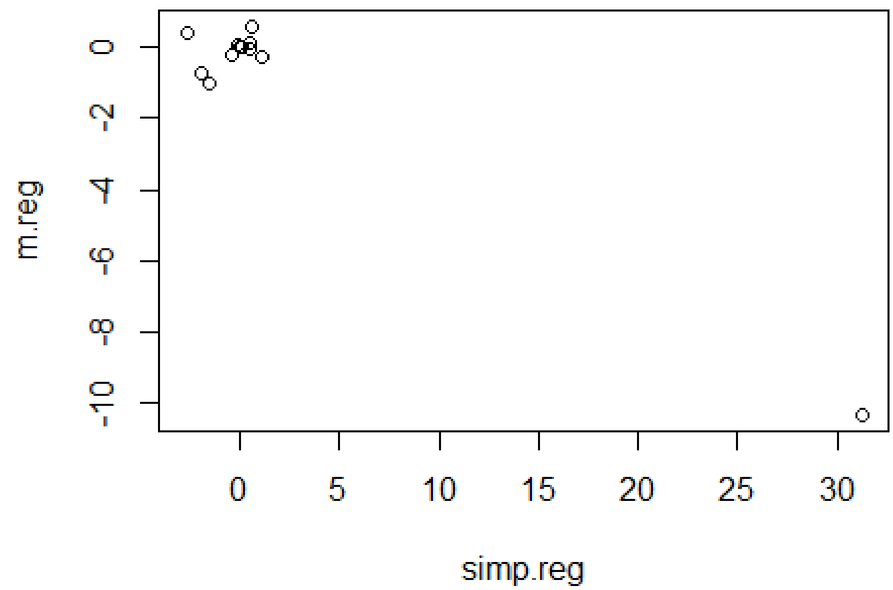
The overall model is significant. We have sufficient evidence to reject the null hypothesis that the following coefficients are zero: zn, dis, rad, black and medv.

15c

```r
names(lm.zn)

##  [1] "coefficients"  "residuals"     "effects"       "rank"
##  [5] "fitted.values" "assign"        "qr"            "df.residual"
##  [9] "xlevels"       "call"          "terms"         "model"

summary(lm.zn$coefficients)

##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
## -0.07394  1.05797  2.18988  2.18988  3.32179  4.45369

lm.zn$coefficients[2]

##          zn
## -0.07393498
```

Checking to see how r stores the data necessary for this question.

```r
simp.reg <- c(coef(lm.zn)[2],
              coef(lm.indus)[2],
              coef(lm.chas)[2],
              coef(lm.nox)[2],
              coef(lm.rm)[2],
              coef(lm.age)[2],
              coef(lm.dis)[2],
              coef(lm.rad)[2],
              coef(lm.tax)[2],
              coef(lm.ptratio)[2],
              coef(lm.black)[2],
              coef(lm.lstat)[2],
              coef(lm.medv)[2])
m.reg = coef(lm.full)[2:14]
plot(simp.reg, m.reg)
```




```
c(simp.reg, m.reg)

##            zn         indus          chas           nox            rm
##  -0.073934977   0.509776331  -1.892776551  31.248531201  -2.684051224
##           age           dis           rad           tax       ptratio
##   0.107786227  -1.550901682   0.617910927   0.029742253   1.151982787
##         black         lstat          medv            zn         indus
##  -0.036279641   0.548804782  -0.363159922   0.044855215  -0.063854824
##          chas           nox            rm           age           dis
##  -0.749133611 -10.313534912   0.430130506   0.001451643  -0.987175726
##           rad           tax       ptratio         black         lstat
##   0.588208591  -0.003780016  -0.271080558  -0.007537505   0.126211376
##          medv
##  -0.198886821
```

Many of these values vary, which is to be expected: in the case of multiple regression, the coefficients are conditional. The largest deviation is the variable nox, which changes drastically from the simple to multiple model.

15d

```
summary(lm(crim~poly(zn, 3), data = Boston))

##
## Call:
## lm(formula = crim ~ poly(zn, 3), data = Boston)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -4.821  -4.614  -1.294   0.473  84.130 
## 
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)    
## (Intercept)      3.6135     0.3722   9.709  < 2e-16 ***
## poly(zn, 3)1 -38.7498     8.3722  -4.628  4.7e-06 ***
## poly(zn, 3)2  23.9398     8.3722   2.859  0.00442 ** 
## poly(zn, 3)3 -10.0719     8.3722  -1.203  0.22954    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 8.372 on 502 degrees of freedom
## Multiple R-squared:  0.05824,    Adjusted R-squared:  0.05261 
## F-statistic: 10.35 on 3 and 502 DF,  p-value: 1.281e-06
```

```r
summary(lm(crim~poly(indus, 3), data = Boston))
```

```
## 
## Call:
## lm(formula = crim ~ poly(indus, 3), data = Boston)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -8.278 -2.514  0.054  0.764 79.713 
## 
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)    
## (Intercept)        3.614      0.330  10.950  < 2e-16 ***
## poly(indus, 3)1   78.591      7.423  10.587  < 2e-16 ***
## poly(indus, 3)2  -24.395      7.423  -3.286  0.00109 ** 
## poly(indus, 3)3  -54.130      7.423  -7.292  1.2e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 7.423 on 502 degrees of freedom
## Multiple R-squared:  0.2597, Adjusted R-squared:  0.2552 
## F-statistic: 58.69 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
summary(lm(crim~poly(nox, 3), data = Boston))
```

```
## 
## Call:
## lm(formula = crim ~ poly(nox, 3), data = Boston)
## 
## Residuals:
##    Min     1Q Median     3Q    Max 
## -9.110 -2.068 -0.255  0.739 78.302 
## 
## Coefficients:
```

```
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3216  11.237  < 2e-16 ***
## poly(nox, 3)1   81.3720     7.2336  11.249  < 2e-16 ***
## poly(nox, 3)2  -28.8286     7.2336  -3.985 7.74e-05 ***
## poly(nox, 3)3  -60.3619     7.2336  -8.345 6.96e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.234 on 502 degrees of freedom
## Multiple R-squared:  0.297,  Adjusted R-squared:  0.2928
## F-statistic: 70.69 on 3 and 502 DF,  p-value: < 2.2e-16
```

```
summary(lm(crim~poly(rm, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(rm, 3), data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -18.485  -3.468  -2.221  -0.015  87.219
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)     3.6135     0.3703   9.758  < 2e-16 ***
## poly(rm, 3)1  -42.3794     8.3297  -5.088 5.13e-07 ***
## poly(rm, 3)2   26.5768     8.3297   3.191  0.00151 **
## poly(rm, 3)3   -5.5103     8.3297  -0.662  0.50858
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.33 on 502 degrees of freedom
## Multiple R-squared:  0.06779,   Adjusted R-squared:  0.06222
## F-statistic: 12.17 on 3 and 502 DF,  p-value: 1.067e-07
```

```
summary(lm(crim~poly(dis, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(dis, 3), data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.757  -2.588   0.031   1.267  76.378
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3259  11.087  < 2e-16 ***
## poly(dis, 3)1  -73.3886     7.3315 -10.010  < 2e-16 ***
## poly(dis, 3)2   56.3730     7.3315   7.689 7.87e-14 ***
## poly(dis, 3)3  -42.6219     7.3315  -5.814 1.09e-08 ***
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.331 on 502 degrees of freedom
## Multiple R-squared:  0.2778, Adjusted R-squared:  0.2735
## F-statistic: 64.37 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
summary(lm(crim~poly(rad, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(rad, 3), data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -10.381  -0.412  -0.269   0.179  76.217
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.2971  12.164  < 2e-16 ***
## poly(rad, 3)1 120.9074     6.6824  18.093  < 2e-16 ***
## poly(rad, 3)2  17.4923     6.6824   2.618  0.00912 **
## poly(rad, 3)3   4.6985     6.6824   0.703  0.48231
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.682 on 502 degrees of freedom
## Multiple R-squared:    0.4,  Adjusted R-squared:  0.3965
## F-statistic: 111.6 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
summary(lm(crim~poly(tax, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(tax, 3), data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.273  -1.389   0.046   0.536  76.950
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3047  11.860  < 2e-16 ***
## poly(tax, 3)1 112.6458     6.8537  16.436  < 2e-16 ***
## poly(tax, 3)2  32.0873     6.8537   4.682 3.67e-06 ***
## poly(tax, 3)3  -7.9968     6.8537  -1.167    0.244
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.854 on 502 degrees of freedom
```

```
## Multiple R-squared:  0.3689, Adjusted R-squared:  0.3651
## F-statistic:  97.8 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
summary(lm(crim~poly(ptratio, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(ptratio, 3), data = Boston)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.833 -4.146 -1.655  1.408 82.697
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)          3.614      0.361  10.008  < 2e-16 ***
## poly(ptratio, 3)1   56.045      8.122   6.901 1.57e-11 ***
## poly(ptratio, 3)2   24.775      8.122   3.050  0.00241 **
## poly(ptratio, 3)3  -22.280      8.122  -2.743  0.00630 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.122 on 502 degrees of freedom
## Multiple R-squared:  0.1138, Adjusted R-squared:  0.1085
## F-statistic: 21.48 on 3 and 502 DF,  p-value: 4.171e-13
```

```r
summary(lm(crim~poly(black, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(black, 3), data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.096  -2.343  -2.128  -1.439  86.790
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3536  10.218   <2e-16 ***
## poly(black, 3)1 -74.4312     7.9546  -9.357   <2e-16 ***
## poly(black, 3)2   5.9264     7.9546   0.745    0.457
## poly(black, 3)3  -4.8346     7.9546  -0.608    0.544
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.955 on 502 degrees of freedom
## Multiple R-squared:  0.1498, Adjusted R-squared:  0.1448
## F-statistic: 29.49 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
summary(lm(crim~poly(lstat, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(lstat, 3), data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -15.234  -2.151  -0.486   0.066  83.353
##
## Coefficients:
##                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.6135     0.3392  10.654   <2e-16 ***
## poly(lstat, 3)1  88.0697     7.6294  11.543   <2e-16 ***
## poly(lstat, 3)2  15.8882     7.6294   2.082   0.0378 *
## poly(lstat, 3)3 -11.5740     7.6294  -1.517   0.1299
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 7.629 on 502 degrees of freedom
## Multiple R-squared:  0.2179, Adjusted R-squared:  0.2133
## F-statistic: 46.63 on 3 and 502 DF,  p-value: < 2.2e-16
```

```r
summary(lm(crim~poly(medv, 3), data = Boston))
```

```
##
## Call:
## lm(formula = crim ~ poly(medv, 3), data = Boston)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -24.427  -1.976  -0.437   0.439  73.655
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.614      0.292  12.374  < 2e-16 ***
## poly(medv, 3)1  -75.058      6.569 -11.426  < 2e-16 ***
## poly(medv, 3)2   88.086      6.569  13.409  < 2e-16 ***
## poly(medv, 3)3  -48.033      6.569  -7.312 1.05e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 6.569 on 502 degrees of freedom
## Multiple R-squared:  0.4202, Adjusted R-squared:  0.4167
## F-statistic: 121.3 on 3 and 502 DF,  p-value: < 2.2e-16
```

Results:

1. ZN: significant linear and quadratic association.
2. Indus: linear, quadratic and cubic.
3. Nox: linear, quadratic and cubic.
4. Rm: linear and quadratic.

5. Dis: linear, quadratic and cubic.
6. Rad: linear and quadratic.
7. Tax: linear and quadratic.
8. Ptratio: linear, quadratic and cubic.
9. Black: linear only.
10. Lstat: linear and quadratic.
11. Medv: linear, quadratic and cubic.

Chapter 4

13

```
library(MASS)
library(dplyr)
library(Hmisc)

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     combine, src, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, round.POSIXt, trunc.POSIXt, units

library(corrplot)

## Warning: package 'corrplot' was built under R version 3.4.3

## corrplot 0.84 loaded

library(caret)

## Warning: package 'caret' was built under R version 3.4.3

##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:survival':
##
##     cluster
```

```r
summary(Boston)
```

```
##       crim                zn             indus            chas
##  Min.   : 0.00632   Min.   :  0.00   Min.   : 0.46   Min.   :0.00000
##  1st Qu.: 0.08204   1st Qu.:  0.00   1st Qu.: 5.19   1st Qu.:0.00000
##  Median : 0.25651   Median :  0.00   Median : 9.69   Median :0.00000
##  Mean   : 3.61352   Mean   : 11.36   Mean   :11.14   Mean   :0.06917
##  3rd Qu.: 3.67708   3rd Qu.: 12.50   3rd Qu.:18.10   3rd Qu.:0.00000
##  Max.   :88.97620   Max.   :100.00   Max.   :27.74   Max.   :1.00000
##       nox               rm             age              dis
##  Min.   :0.3850   Min.   :3.561   Min.   :  2.90   Min.   : 1.130
##  1st Qu.:0.4490   1st Qu.:5.886   1st Qu.: 45.02   1st Qu.: 2.100
##  Median :0.5380   Median :6.208   Median : 77.50   Median : 3.207
##  Mean   :0.5547   Mean   :6.285   Mean   : 68.57   Mean   : 3.795
##  3rd Qu.:0.6240   3rd Qu.:6.623   3rd Qu.: 94.08   3rd Qu.: 5.188
##  Max.   :0.8710   Max.   :8.780   Max.   :100.00   Max.   :12.127
##       rad              tax           ptratio          black
##  Min.   : 1.000   Min.   :187.0   Min.   :12.60   Min.   :  0.32
##  1st Qu.: 4.000   1st Qu.:279.0   1st Qu.:17.40   1st Qu.:375.38
##  Median : 5.000   Median :330.0   Median :19.05   Median :391.44
##  Mean   : 9.549   Mean   :408.2   Mean   :18.46   Mean   :356.67
##  3rd Qu.:24.000   3rd Qu.:666.0   3rd Qu.:20.20   3rd Qu.:396.23
##  Max.   :24.000   Max.   :711.0   Max.   :22.00   Max.   :396.90
##      lstat            medv
##  Min.   : 1.73   Min.   : 5.00
##  1st Qu.: 6.95   1st Qu.:17.02
##  Median :11.36   Median :21.20
##  Mean   :12.65   Mean   :22.53
##  3rd Qu.:16.95   3rd Qu.:25.00
##  Max.   :37.97   Max.   :50.00
```

```r
glimpse(Boston)
```

```
## Observations: 506
## Variables: 14
## $ crim    <dbl> 0.00632, 0.02731, 0.02729, 0.03237, 0.06905, 0.02985, ...
## $ zn      <dbl> 18.0, 0.0, 0.0, 0.0, 0.0, 0.0, 12.5, 12.5, 12.5, 12.5,...
## $ indus   <dbl> 2.31, 7.07, 7.07, 2.18, 2.18, 2.18, 7.87, 7.87, 7.87, ...
## $ chas    <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, ...
## $ nox     <dbl> 0.538, 0.469, 0.469, 0.458, 0.458, 0.458, 0.524, 0.524...
## $ rm      <dbl> 6.575, 6.421, 7.185, 6.998, 7.147, 6.430, 6.012, 6.172...
## $ age     <dbl> 65.2, 78.9, 61.1, 45.8, 54.2, 58.7, 66.6, 96.1, 100.0,...
## $ dis     <dbl> 4.0900, 4.9671, 4.9671, 6.0622, 6.0622, 6.0622, 5.5605...
## $ rad     <int> 1, 2, 2, 3, 3, 3, 5, 5, 5, 5, 5, 5, 5, 4, 4, 4, 4, 4, ...
## $ tax     <dbl> 296, 242, 242, 222, 222, 222, 311, 311, 311, 311, 311,...
## $ ptratio <dbl> 15.3, 17.8, 17.8, 18.7, 18.7, 18.7, 15.2, 15.2, 15.2, ...
## $ black   <dbl> 396.90, 396.90, 392.83, 394.63, 396.90, 394.12, 395.60...
```

```
## $ lstat   <dbl> 4.98, 9.14, 4.03, 2.94, 5.33, 5.21, 12.43, 19.15, 29.9...
## $ medv    <dbl> 24.0, 21.6, 34.7, 33.4, 36.2, 28.7, 22.9, 27.1, 16.5, ...
```

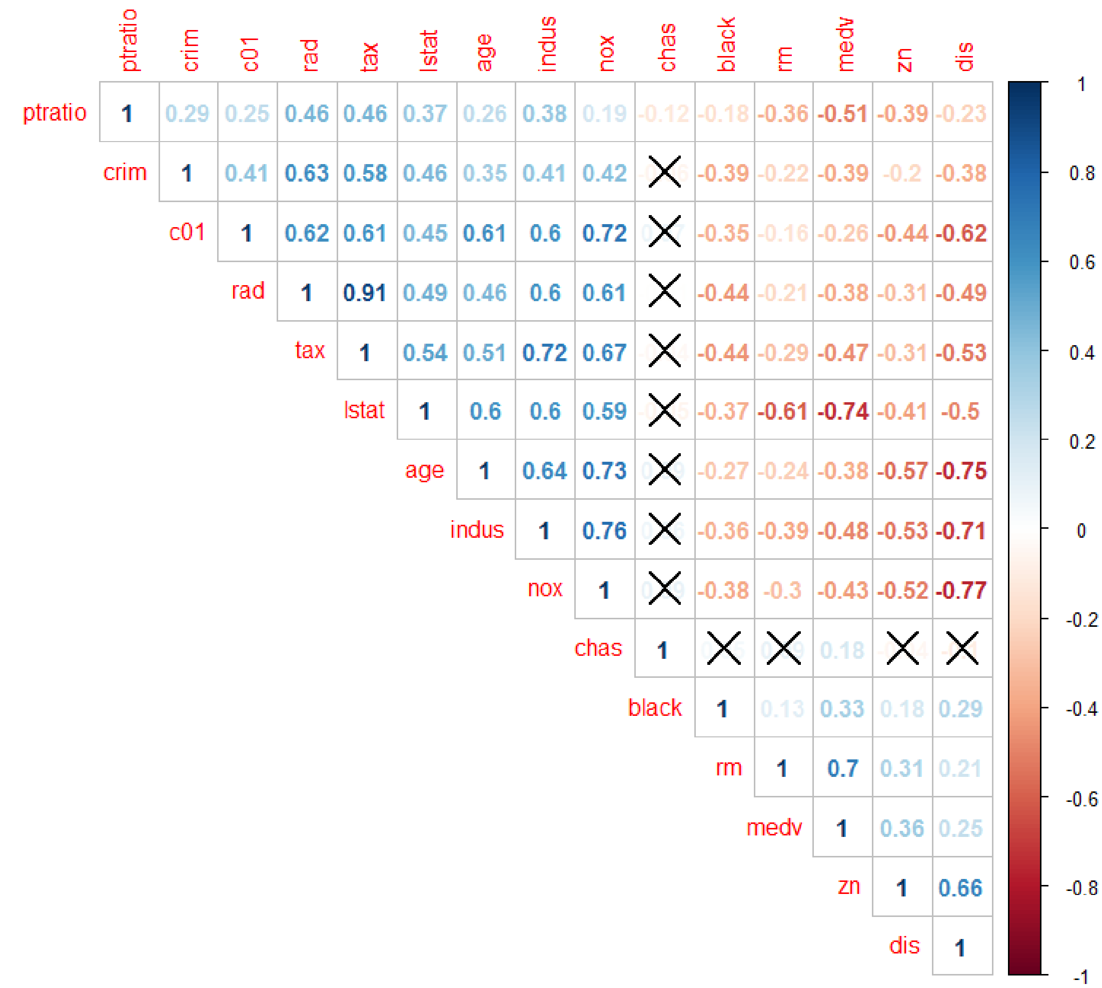
```
c01 <- with(Boston, ifelse(crim > median(crim), 1, 0))
crmdf <- data.frame(Boston, c01)
summary(crmdf$c01)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.5     0.5     1.0     1.0
```

```
cor.1 <- rcorr(as.matrix(crmdf))
cor.2 <- cor(crmdf)
head(round(cor.2,2))
```

```
##        crim    zn indus  chas   nox    rm   age   dis   rad   tax ptratio
## crim   1.00 -0.20  0.41 -0.06  0.42 -0.22  0.35 -0.38  0.63  0.58    0.29
## zn    -0.20  1.00 -0.53 -0.04 -0.52  0.31 -0.57  0.66 -0.31 -0.31   -0.39
## indus  0.41 -0.53  1.00  0.06  0.76 -0.39  0.64 -0.71  0.60  0.72    0.38
## chas  -0.06 -0.04  0.06  1.00  0.09  0.09  0.09 -0.10 -0.01 -0.04   -0.12
## nox    0.42 -0.52  0.76  0.09  1.00 -0.30  0.73 -0.77  0.61  0.67    0.19
## rm    -0.22  0.31 -0.39  0.09 -0.30  1.00 -0.24  0.21 -0.21 -0.29   -0.36
##       black lstat  medv   c01
## crim  -0.39  0.46 -0.39  0.41
## zn     0.18 -0.41  0.36 -0.44
## indus -0.36  0.60 -0.48  0.60
## chas   0.05 -0.05  0.18  0.07
## nox   -0.38  0.59 -0.43  0.72
## rm     0.13 -0.61  0.70 -0.16
```

```
corrplot(cor.2, type = "upper", order = "hclust", method = "number", p.mat =
cor.1$P, sig.level = .01)
```




```r
#Using dplyr to partition into 80/20
set.seed(2468)
ctrain <- sample_frac(crmdf, 0.8)
dataid <-as.numeric(rownames(ctrain))
ctest <- crmdf[-dataid,]
```

Logistic Regression Models:

```r
logfit.test <- glm(c01 ~ . - c01 - crim, data = ctrain, family = binomial)
summary(logfit.test)

##
## Call:
## glm(formula = c01 ~ . - c01 - crim, family = binomial, data = ctrain)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
```

```
## -2.1701  -0.0977   0.0000   0.0007   3.4799
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -40.329939   7.523574  -5.360 8.30e-08 ***
## zn           -0.083487   0.039469  -2.115  0.03441 *
## indus        -0.085984   0.051310  -1.676  0.09378 .
## chas          0.290080   0.774209   0.375  0.70790
## nox          55.676894   9.402389   5.922 3.19e-09 ***
## rm           -0.927167   0.832999  -1.113  0.26569
## age           0.021071   0.013659   1.543  0.12291
## dis           0.875621   0.267030   3.279  0.00104 **
## rad           0.812787   0.188790   4.305 1.67e-05 ***
## tax          -0.006669   0.002897  -2.302  0.02133 *
## ptratio       0.393480   0.144660   2.720  0.00653 **
## black        -0.008293   0.005390  -1.538  0.12393
## lstat         0.109932   0.055502   1.981  0.04763 *
## medv          0.232217   0.084987   2.732  0.00629 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 560.56  on 404  degrees of freedom
## Residual deviance: 156.60  on 391  degrees of freedom
## AIC: 184.6
##
## Number of Fisher Scoring iterations: 9
```

```
logfit.prob <- predict(logfit.test, ctest, type="response")
logfit.pred <- rep(0, length(logfit.prob))
logfit.pred[logfit.prob > .5] = 1
table(logfit.pred, ctest$c01)
```

```
##
## logfit.pred  0  1
##           0 51  4
##           1  9 37
```

```
mean(logfit.pred != ctest$c01)
```

```
## [1] 0.1287129
```

This model has a 12.87% test error rate. Lets try it with just the most correlated variables

```
#Logistic model 2
logfit.test2 <- glm(c01 ~ rad + tax +  nox + indus + lstat + dis + zn, data =
ctrain, family = binomial)
summary(logfit.test2)$coef
```

```
##                  Estimate  Std. Error    z value     Pr(>|z|)
## (Intercept) -27.380609758 4.424856144 -6.1879096 6.096728e-10
## rad           0.772547452 0.157422213  4.9074869 9.225080e-07
## tax          -0.007327355 0.002614036 -2.8030812 5.061693e-03
## nox          46.445656188 7.709294799  6.0246310 1.694959e-09
## indus        -0.070199129 0.047236330 -1.4861258 1.372458e-01
## lstat         0.035252969 0.035886177  0.9823551 3.259249e-01
## dis           0.442166516 0.208096168  2.1248182 3.360179e-02
## zn           -0.075936943 0.033445679 -2.2704560 2.317993e-02
```

```
logfit.prob2 <- predict(logfit.test2, ctest, type="response")
logfit.pred2 <- rep(0, length(logfit.prob))
logfit.pred2[logfit.prob2 > .5] = 1
names(logfit.pred2)
```

```
## NULL
```

```
table(logfit.pred2, ctest$c01)
```

```
##
## logfit.pred2  0  1
##            0 47  4
##            1 13 37
```

```
mean(logfit.pred2 != ctest$c01)
```

```
## [1] 0.1683168
```

The test error of this model is 16.63%. Not an improvement.

LDA Models:

```
lda.fit <- with(ctrain, lda(c01 ~ . - c01 - crim, data = ctrain))
names(lda.fit)
```

```
##  [1] "prior"   "counts"  "means"   "scaling" "lev"     "svd"     "N"
##  [8] "call"    "terms"   "xlevels"
```

```
lda.fit$prior
```

```
##         0         1
## 0.4765432 0.5234568
```

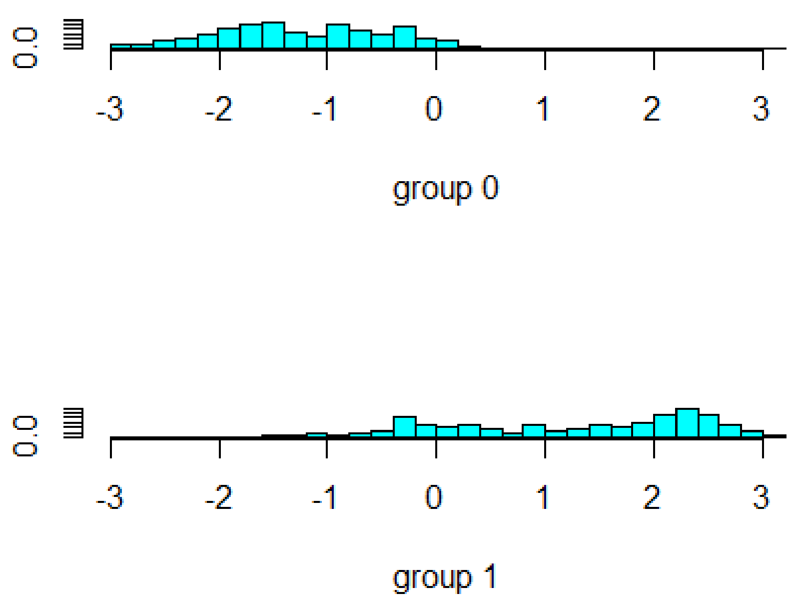
```
lda.pred = predict(lda.fit, ctest)
table(lda.pred$class, ctest$c01)
```

```
##
##      0  1
##   0 58 11
##   1  2 30
```

```
mean(lda.pred$class != ctest$c01)
```

```
## [1] 0.1287129
```

```r
plot(lda.fit, panel = lda.fit, cex = 0.7, dimen = 2,
     abbrev = FALSE)
```



This model gives us a test error rate of 12.87%

```r
lda.fit2 <- with(ctrain, lda(c01 ~ . - c01 - crim - tax - indus - zn - chas,
data = ctrain))
lda.fit2$prior
```

```
##         0         1 
## 0.4765432 0.5234568
```

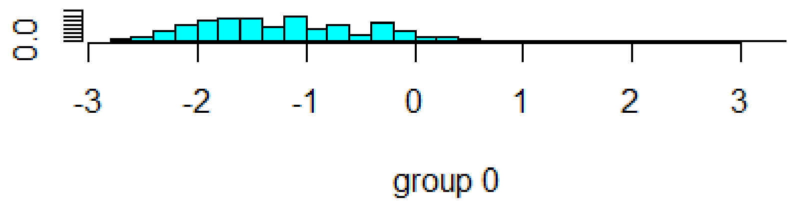
```r
lda.pred2 = predict(lda.fit2, ctest)
table(lda.pred2$class, ctest$c01)
```

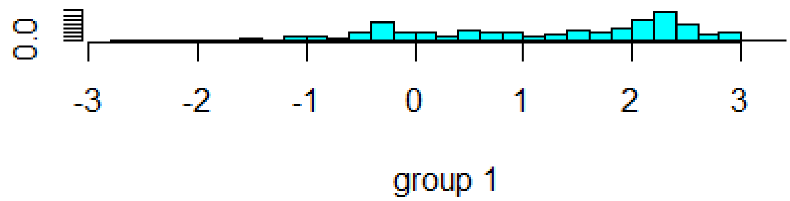
```
##    
##      0  1
##   0 60 11
##   1  0 30
```

```r
mean(lda.pred2$class != ctest$c01)
```

```
## [1] 0.1089109
```

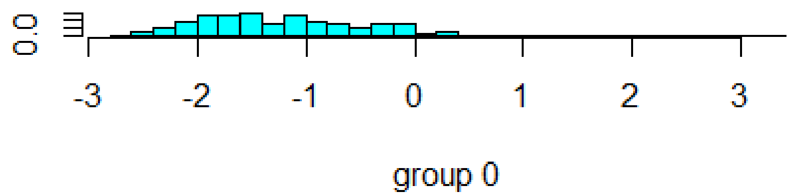
```r
plot(lda.fit2, panel = lda.fit, cex = 0.7, dimen = 2,
     abbrev = FALSE)
```

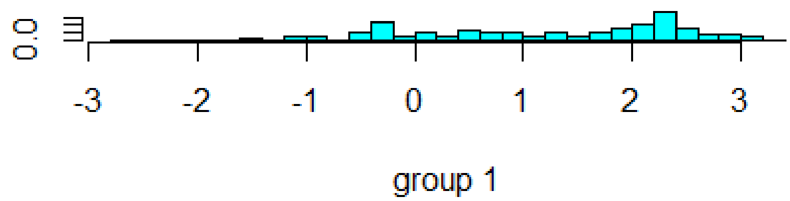




Removing tax, indus and zn improves the test error rate to 10.89%

```
lda.fit3 <- with(ctrain, lda(c01 ~ . - c01 - crim  - tax - indus - zn  - dis
- chas - rm - black , data = ctrain))
lda.fit3$prior

##         0         1
## 0.4765432 0.5234568

lda.pred3 = predict(lda.fit3, ctest)
table(lda.pred3$class, ctest$c01)

##
##      0  1
##   0 58 11
##   1  2 30

mean(lda.pred3$class != ctest$c01)

## [1] 0.1287129

plot(lda.fit3, panel = lda.fit, cex = 0.7, dimen = 2,
     abbrev = FALSE)
```





Further removing dis, chas, rm & black increases the test error rate to 12.87%

For fun, a QDA model

```
library(klaR)

## Warning: package 'klaR' was built under R version 3.4.3

qda.fit <- with(ctrain, qda(c01 ~ . - c01 - crim, data = ctrain))
qda.fit$prior

##         0         1
## 0.4765432 0.5234568

qda.fit <- predict(qda.fit, ctest)
table(qda.fit$class, ctest$c01)

##
##      0  1
##   0 59 11
##   1  1 30

mean(qda.fit$class != ctest$c01)

## [1] 0.1188119
```

The same model fit as linear had an error rate of 12.87%; here the error rate is 11.88%.

KNN Models:

```r
library(class)

## Warning: package 'class' was built under R version 3.4.3

set.seed(5654)
train.x <- with(ctrain, cbind(zn, indus, chas, nox, rm, age, dis, rad, tax,
ptratio, black, lstat, medv))
test.x <- with(ctest, cbind(zn, indus, chas, nox, rm, age, dis, rad, tax,
ptratio, black, lstat, medv))
knn1 <- knn(train.x, test.x, ctrain$c01, k=1)
mean(knn1 != ctest$c01)

## [1] 0.06930693

knn2 <- knn(train.x, test.x, ctrain$c01, k=5)
mean(knn2 != ctest$c01)

## [1] 0.06930693

knn3 <- knn(train.x, test.x, ctrain$c01, k=10)
mean(knn3 != ctest$c01)

## [1] 0.1089109

knn4 <- knn(train.x, test.x, ctrain$c01, k=20)
mean(knn4 != ctest$c01)

## [1] 0.1485149

knn5 <- knn(train.x, test.x, ctrain$c01, k=50)
mean(knn5 != ctest$c01)

## [1] 0.1782178

knn6 <- knn(train.x, test.x, ctrain$c01, k=75)
mean(knn6 != ctest$c01)

## [1] 0.1881188

table(knn1, ctest$c01)

##
## knn1  0  1
##    0 55  2
##    1  5 39

table(knn2, ctest$c01)

##
## knn2  0  1
##    0 55  2
##    1  5 39

table(knn3, ctest$c01)
```

```
##
## knn3  0  1
##    0 51  2
##    1  9 39
```

```
table(knn4, ctest$c01)
```

```
##
## knn4  0  1
##    0 48  3
##    1 12 38
```

```
table(knn5, ctest$c01)
```

```
##
## knn5  0  1
##    0 47  5
##    1 13 36
```

```
table(knn6, ctest$c01)
```

```
##
## knn6  0  1
##    0 54 13
##    1  6 28
```

Using all variables in the data set:

1. k=1 produces a model with a 6.9% test error rate
2. k=5 produces a model with a 6.9% test error rate
3. k=10 produces a model with a 10.89% test error rate
4. k=20 produces a model with a 14.85% test error rate
5. k=50 produces a model with a 17.82% test error rate
6. k=150 produces a model with a 18.81.86% test error rate

```
#Here only the most strongly correlated variables with c01 are kept
train.x2 <- with(ctrain, cbind(rad, tax, dis, nox, indus))
test.x2 <- with(ctest, cbind(rad, tax, dis, nox, indus))

knn21 <- knn(train.x2, test.x2, ctrain$c01, k=1)
mean(knn21 != ctest$c01)
```

```
## [1] 0.05940594
```

```
knn22 <- knn(train.x2, test.x2, ctrain$c01, k=5)
mean(knn22 != ctest$c01)
```

```
## [1] 0.05940594
```

```
knn23 <- knn(train.x2, test.x2, ctrain$c01, k=10)
mean(knn23 != ctest$c01)
```

```
## [1] 0.05940594

knn24 <- knn(train.x2, test.x2, ctrain$c01, k=20)
mean(knn24 != ctest$c01)

## [1] 0.07920792

knn25 <- knn(train.x2, test.x2, ctrain$c01, k=50)
mean(knn25 != ctest$c01)

## [1] 0.2673267

knn26 <- knn(train.x2, test.x2, ctrain$c01, k=75)
mean(knn26 != ctest$c01)

## [1] 0.2574257

table(knn21, ctest$c01)

##
## knn21  0  1
##     0 54  0
##     1  6 41

table(knn22, ctest$c01)

##
## knn22  0  1
##     0 54  0
##     1  6 41

table(knn23, ctest$c01)

##
## knn23  0  1
##     0 55  1
##     1  5 40

table(knn24, ctest$c01)

##
## knn24  0  1
##     0 53  1
##     1  7 40

table(knn25, ctest$c01)

##
## knn25  0  1
##     0 36  3
##     1 24 38

table(knn26, ctest$c01)
```

```
##
## knn26  0  1
##     0 37  3
##     1 23 38
```

1. k=1 through k = 10 have the same test error rate: 5.94%.
2. k=20 has a test error rate of 9.9%
3. k=50 has a 27.72% test error rate
4. k=75 has a 25.74% test error rate

In this case, selecting only the most correlated variables with out outcome produces models with less error when k<20 vis-a-vis clustering using all provided variables. In general, the trend is that as the value of K increases, so does test error.

Overall, given the variables selected, KNN had the lowest error rate, followed by QDA, LDA and logistic regression. If the goal classification, KNN is the best performing option given these data, methods and train/test specification. Logistic regression-while not as accurate as the more flexible methods-does have the advantage of having the most interpretable parameter estimates, which may be of more importance than predictive performance depending on the goals of a given analysis.