ISLR Notes and Exercises

Nathaniel Lai

December, 7, 2017

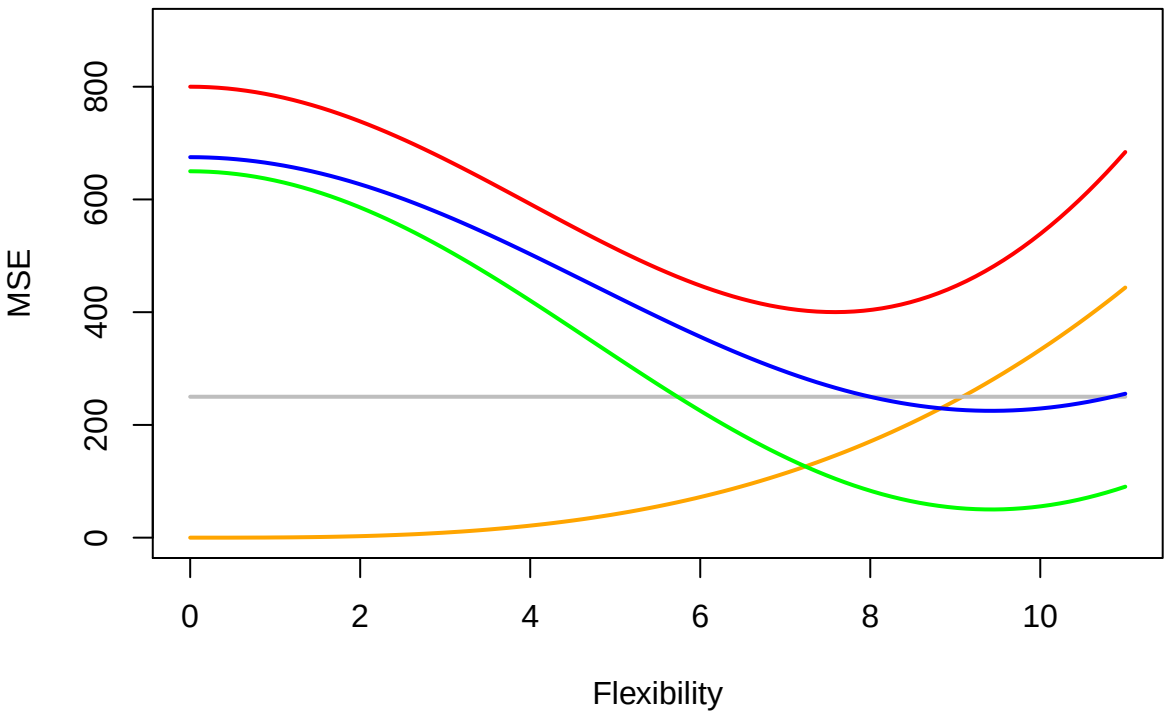
Contents

Chapter 2: Statistical Learning

Bias-Variance Tradeoff

The key concept for statistical learning is bias-variance tradeoff summarized in the diagram (Weatherwax).

- `red` = test error
- `orange` = estimator variance
- `green` = model bias
- `gray` = irreducible error
- `blue` = training error



where

$$MSE = \frac{1}{n}\sum (y_i - \hat{f}(x_i)^2$$

and the training error rate is :

$$\frac{1}{n}\sum_{i=1}^{n} I(y_i \neq \hat{y}_i)$$

The bias-variance tradeoff is:

$$E(y_0 - \hat{f}(x_0))^2 = Var(\hat{f}(x_0) + [Bias(\hat{f}(x_0))]^2 + Var(\varepsilon)$$

$$Expected\ MSE = \underbrace{variance + bias}_{reducible\ error} + irreducible\ error$$

"When a given method yields a small training MSE but a large test MSE, we are said to be **overfitting** the data. This happens because our statistical learning procedure is working too hard to find patterns in the training data, and may be picking up some patterns that are just caused by random chance rather than by true properties of the unknown function f. When we overfit the training data, the test MSE will be very large because the supposed patterns that the method found in the training data simply don't exist in the test data. Note that regardless of whether or not overfitting has occurred, we almost always expect the training MSE to be smaller than the test MSE because most statistical learning methods either directly or indirectly seek to minimize the training MSE. Overfitting refers specifically to the case in which a less flexible model would have yielded a smaller test MSE." (ISLR P.32)

It is possible to show that the test error rate is minimized, on average, by a very simple classifier that assigns each observation to the most likely class, given its predictor values. In other words, we should simply assign a test observation with predictor vector x_0 to the class j for which the conditional probability (ISLR P.38)

$$Pr(Y = j|X = x_0)$$

is the largest with the error rate:

$$1 - E\Big(\max_j Pr(Y = j|X\Big)$$

Advantages and Disadvantages of flexible model

###Question 5

Advantages of a very flexible model include better fit to data and fewer prior assumptions. Disadvantages are hard to interpret and prone to overfitting.

A more flexible approach might be preferred is the underlying data is very complex (simple linear fit doesn't suffice) or if we mainly care about the result and not inference, (provided that sample size is large enough). A less flexible model is preferred is the underlying data has a simple shape or if inference and interpretability are important.

Parametric and Non-parametric Methods

###Question 6

For parametric methods, we make an assumption about the shape of the underlying data, select a model form, and fit the data to our selected form. The advantage here is that we can incorporate any prior/expert knowledge and don't tend to have too many parameters that need to be fit. To the extent that our prior/expert assumptions are wrong, then that would be a disadvantage.

Non-parametric methods don't make explicit assumptions on the shape of the data. This can have the advantage of not needing to make an assumption on the form of the function and can more accurately fit a wider range of shapes for the underlying data. The key disadvantage is that they need a large number of observations to fit an accurate estimate.

Applied Questions

###Question 8

This question is on standard regrassion prcedures and the use of `ggplot`.

Part a)

```
require(ISLR)
```

```
## Loading required package: ISLR
```

```
data(College)
str(College)
```

```
## 'data.frame':    777 obs. of  18 variables:
##  $ Private    : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Apps       : num  1660 2186 1428 417 193 ...
##  $ Accept     : num  1232 1924 1097 349 146 ...
##  $ Enroll     : num  721 512 336 137 55 158 103 489 227 172 ...
##  $ Top10perc  : num  23 16 22 60 16 38 17 37 30 21 ...
##  $ Top25perc  : num  52 29 50 89 44 62 45 68 63 44 ...
##  $ F.Undergrad: num  2885 2683 1036 510 249 ...
##  $ P.Undergrad: num  537 1227 99 63 869 ...
##  $ Outstate   : num  7440 12280 11250 12960 7560 ...
##  $ Room.Board : num  3300 6450 3750 5450 4120 ...
##  $ Books      : num  450 750 400 450 800 500 500 450 300 660 ...
##  $ Personal   : num  2200 1500 1165 875 1500 ...
##  $ PhD        : num  70 29 53 92 76 67 90 89 79 40 ...
##  $ Terminal   : num  78 30 66 97 72 73 93 100 84 41 ...
##  $ S.F.Ratio  : num  18.1 12.2 12.9 7.7 11.9 9.4 11.5 13.7 11.3 11.5 ...
##  $ perc.alumni: num  12 16 30 37 2 11 26 37 23 15 ...
##  $ Expend     : num  7041 10527 8735 19016 10922 ...
##  $ Grad.Rate  : num  60 56 54 59 15 55 63 73 80 52 ...
```

```
class(College)
```

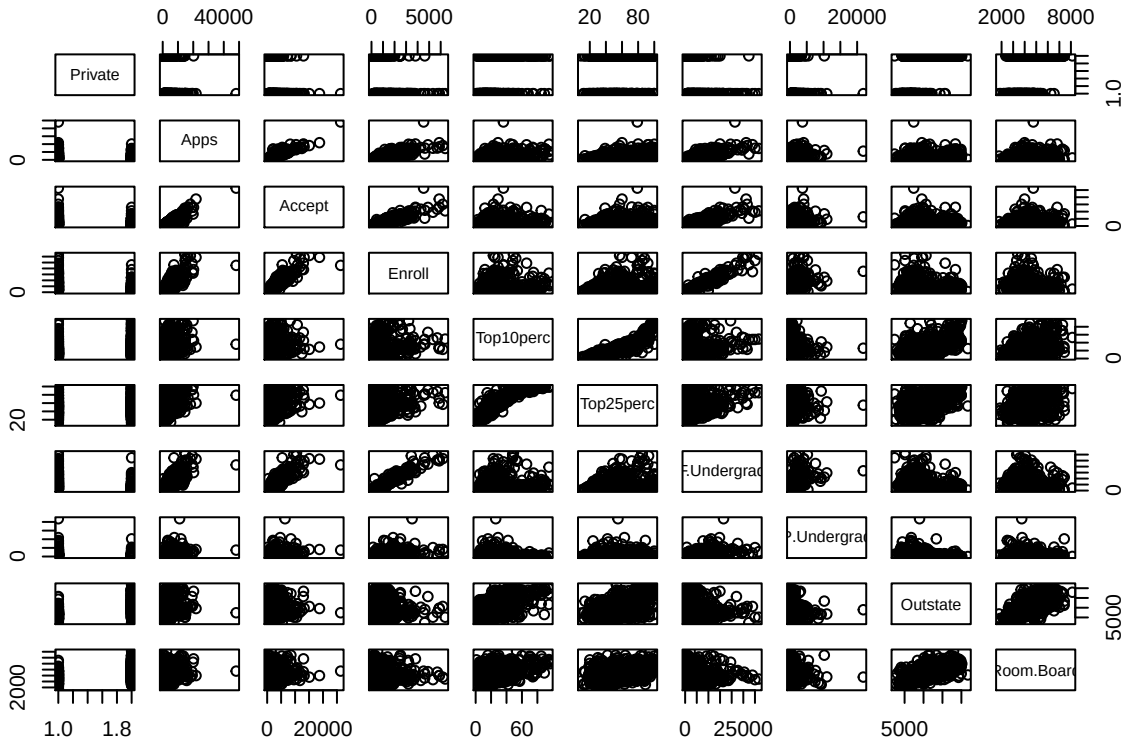
```
## [1] "data.frame"
```

Part b)

```
# these steps were already taken on College data in the ISLR package
rownames(College) <- College[,1]  # set row names
College <- College[,-1]  # drop first col
# i.
summary(College)
```

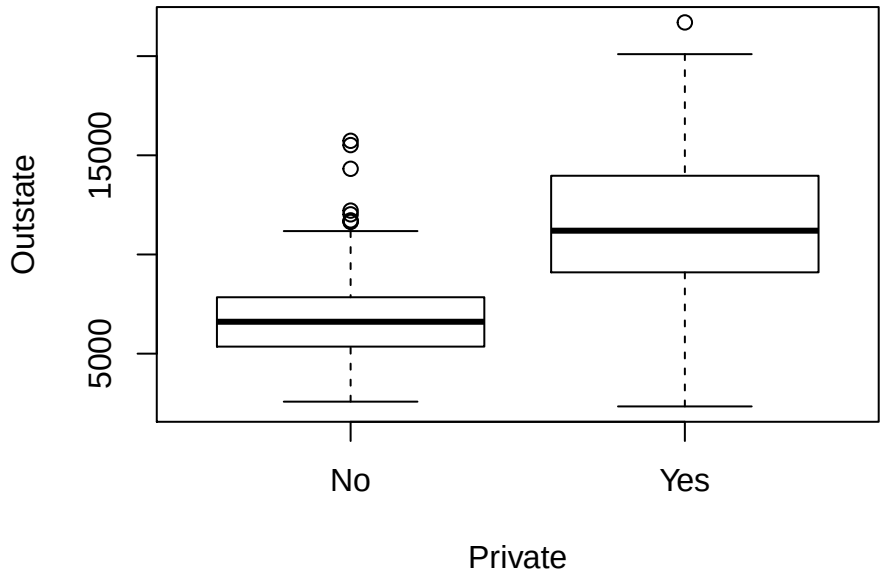
Part c)

```
# ii.
pairs(College[,1:10])
```




```
# iii.
```

```
boxplot(College$Outstate~College$Private, data=College, xlab="Private", ylab="Outstate")
```

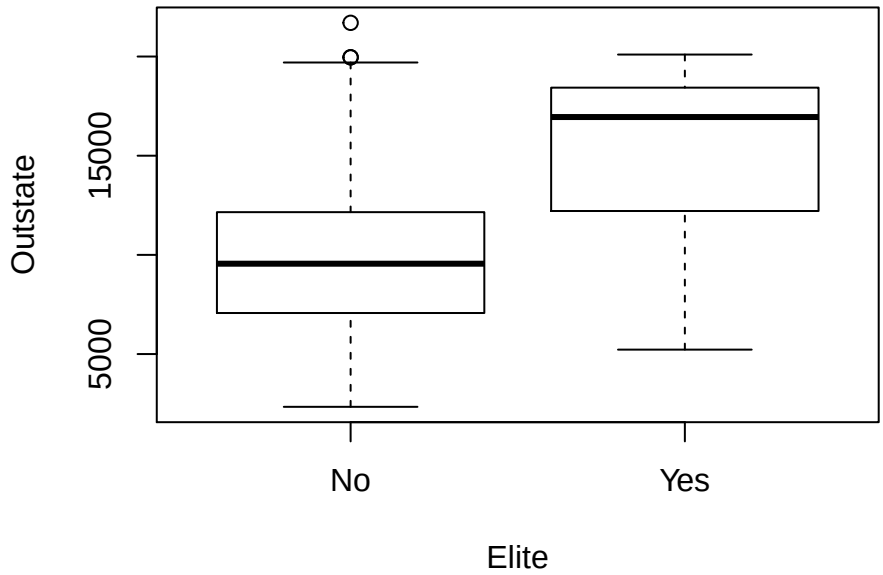



```
# iv.
Elite <- rep("No", nrow(College))
Elite[College$Top10perc>50] <- "Yes"
College <- data.frame(College, Elite)
summary(College)  # 78 Elite
```

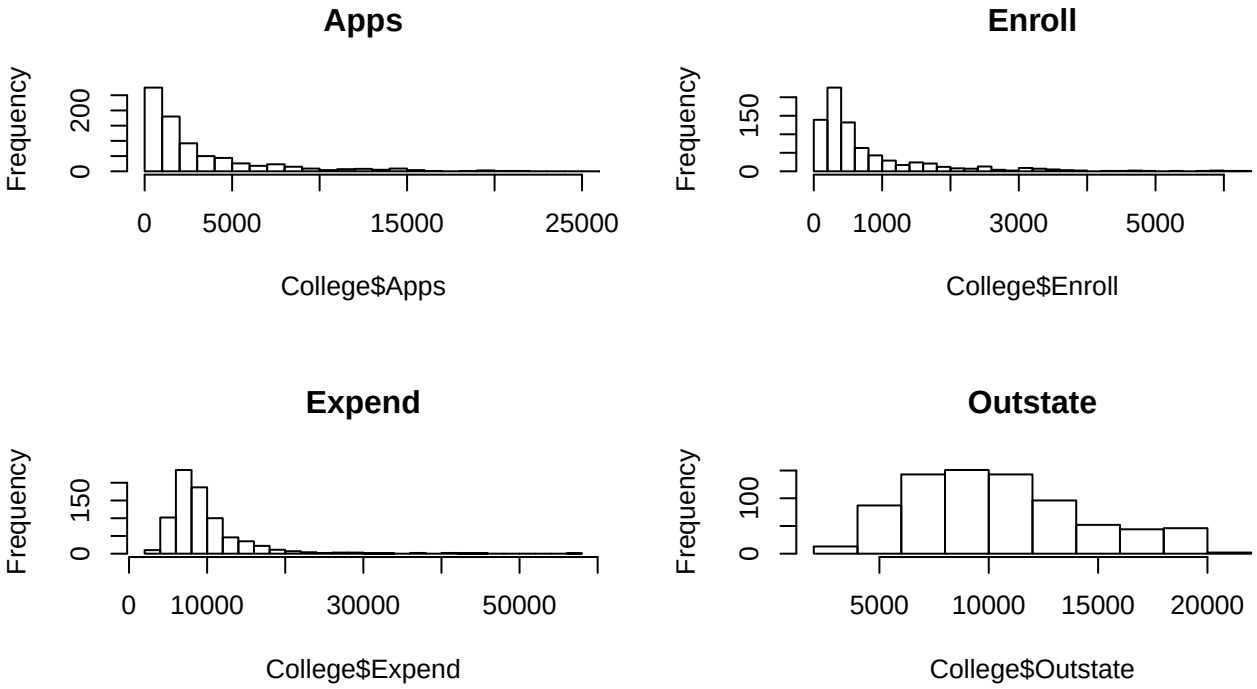
```
##  Private        Apps           Accept          Enroll       Top10perc
##  No :212   Min.   :   81   Min.   :   72   Min.   :  35   Min.   : 1.00
##  Yes:565   1st Qu.:  776   1st Qu.:  604   1st Qu.: 242   1st Qu.:15.00
```

```
##            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      Elite
##  Min.   : 10.00   No :699
##  1st Qu.: 53.00   Yes: 78
##  Median : 65.00
##  Mean   : 65.46
##  3rd Qu.: 78.00
##  Max.   :118.00
```

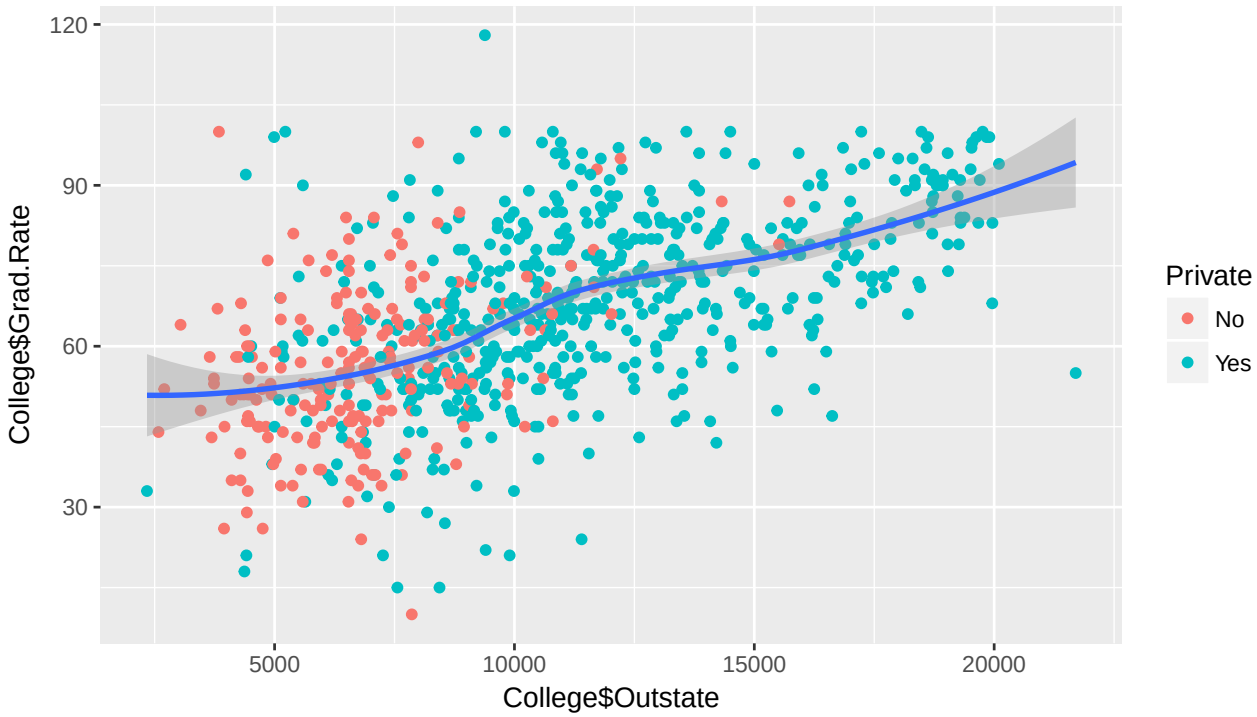
```
boxplot(College$Outstate~College$Elite, data=College, xlab="Elite", ylab="Outstate")
```

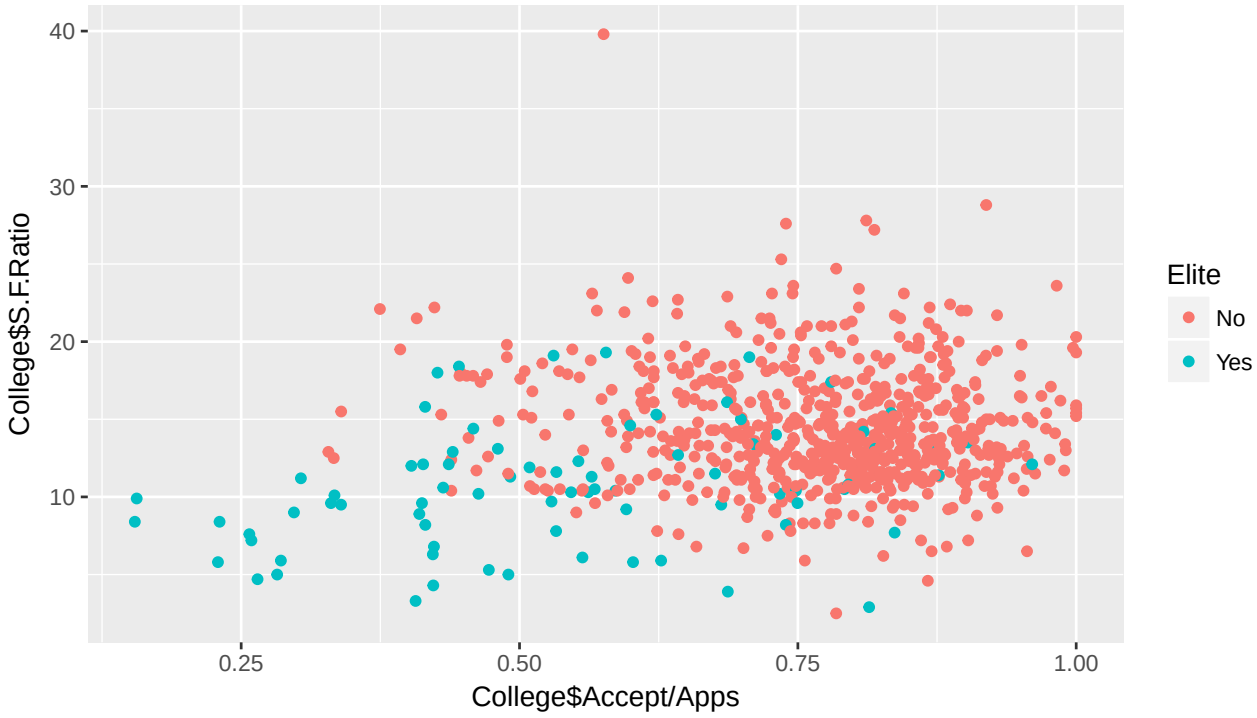


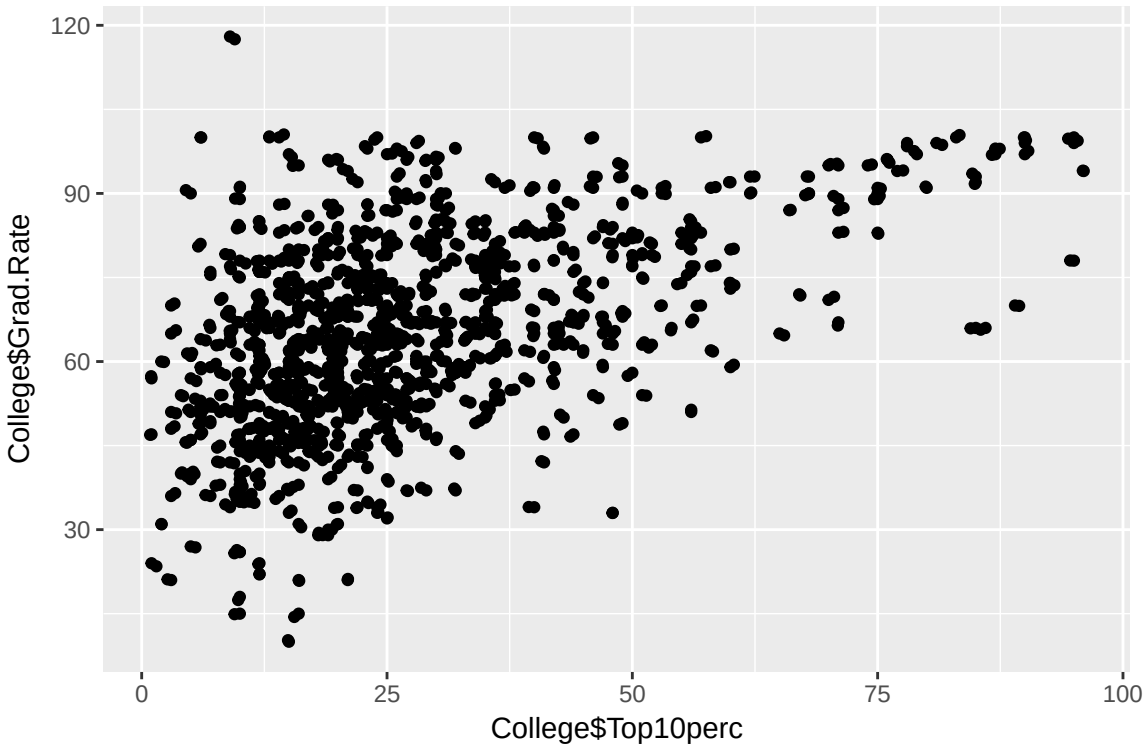
```r
# v.
par(mfrow=c(2,2))
hist(College$Apps, breaks=50, xlim=c(0,25000), main="Apps")
hist(College$Enroll, breaks=25, main="Enroll")
hist(College$Expend, breaks=25, main="Expend")
hist(College$Outstate, main="Outstate")
```




```
## `geom_smooth()` using method = 'loess'
```


College$Grad.Rate
120
90
60
30
5000
10000
15000
20000
College$Outstate
Private
No
Yes


College$S.F.Ratio
40
30
20
10
0.25
0.50
0.75
1.00
College$Accept/Apps
Elite
No
Yes



Colleges with the most students from top 10% perc do not necessarily have the highest graduation rate. Also, rate over 100 is erroneous.

Chapter 3: Linear Regression

Prediction and Inference

From discussion about the two concepts on Cross Valided,

"Inference: Given a set of data you want to infer how the output is generated as a function of the data.

Prediction: Given a new measurement, you want to use an existing data set to build a model that reliably chooses the correct identifier from a set of outcomes.""

The definitions of inference, in statistics and econmetrics, confuse me.

ETM (2004): "If we are to interpret any given set of OLS parameter estimates, we need to know, at least approximately, how $\hat{\ }$ $\bar{\ }$ is actually distributed. For purposes of **inference**, the most important feature of the distribution of any vector of parameter estimates is the matrix of its central second moments."

It seems that inference, in ETM (2004), concerns about the distribution (1st and 2nd monments) of the estimator while, in ISLR (2013), inference, is about the sesearh of a corrct model for the dependent and independent variables.

Potenital Problems of Regression:

- Non-linearity of the response-predictor relationships.
- Correlation of error terms.
- Non-constant variance of error terms.
- Outliers.
- High-leverage points.
- Collinearity.

###Question 2

KNN regression averages the closest observations to estimate prediction, KNN classifier assigns classification group based on majority of closest observations.

"Given a positive K-nearest integer K and a test observation x_0, the KNN classifier first identifies the neighbors K points in the training data that are closest to x_0, represented by N_0. It then estimates the conditional probability for class j as the fraction of points in N_0 whose response values equal j:

$$Pr(Y = j|X = x_0) = \frac{1}{K} \sum_{i \in N_0} I(y_i = j)$$

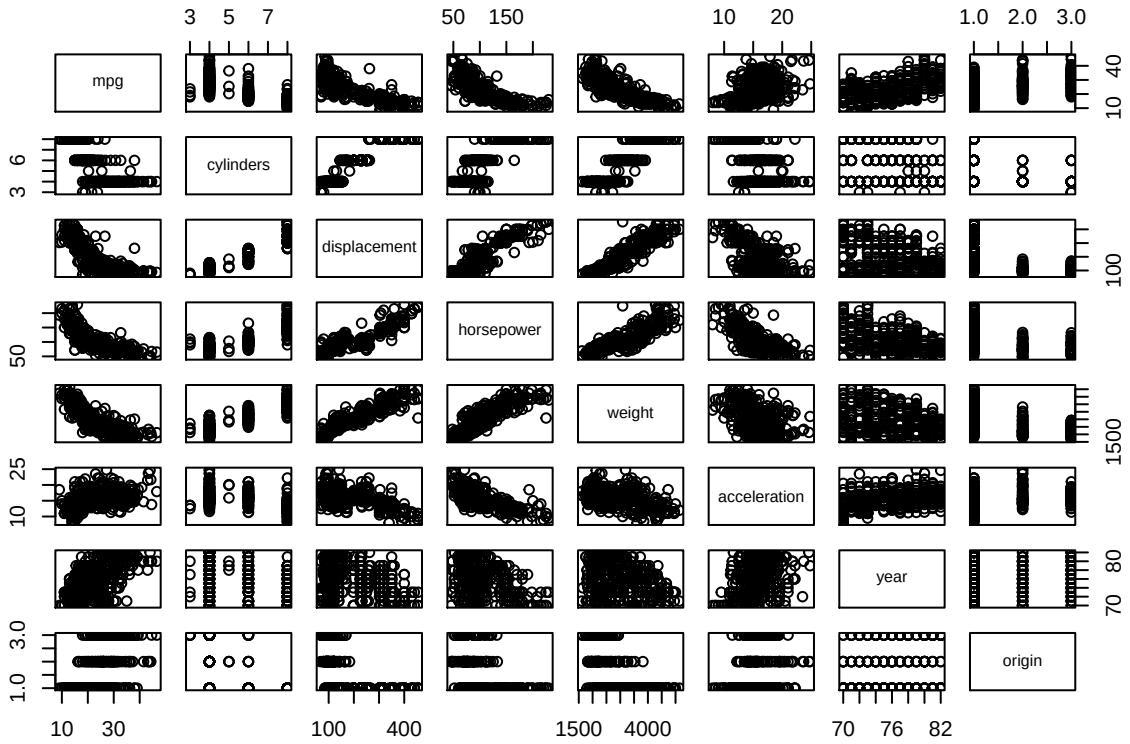
Finally, KNN applies Bayes rule and classifies the test observation x_0 to the class with the largest probability.

Applied Questions

###Question 9

This question aims to teach standard regrassion prcedures.

```
Auto = read.csv("/Users/nathaniellai/Desktop/R/ISLR/data/Auto.csv", header=T, na.strings="?")
Auto = na.omit(Auto)
pairs(Auto[,-9])
```




```
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
```

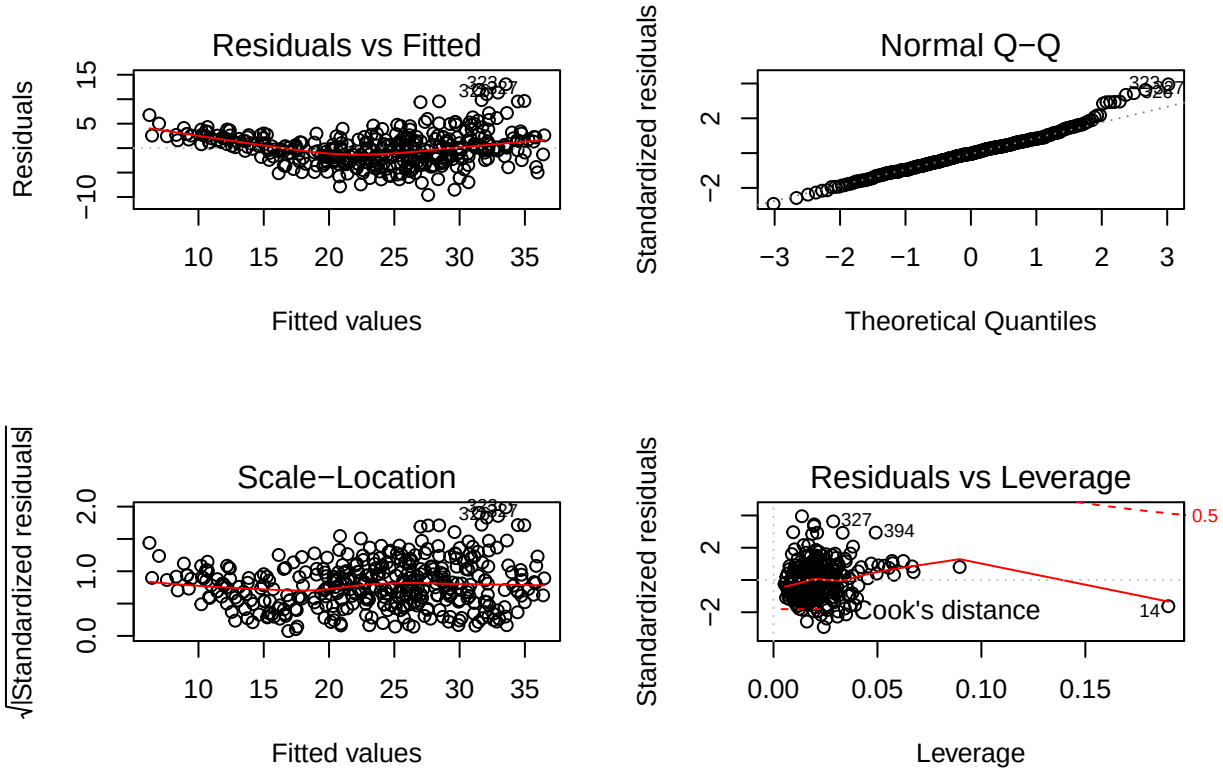
```
lm.fit0 <- lm(mpg ~ . -name, data=Auto)
summary(lm.fit0)
```

```
##
## 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
```

Overall, the model supports a relationship bwteen predictors and the response, as suggested by the low p-value of the F test. Of the seven variables (excluding the incept), displacement, wight, year and oruginhave statistically significant effects onmpgwhilecylinders,horsepower, andaccelerationdo not. The variable,year, indicates that, for every one year,mpgincreases by the0.7507727. In other words, cars become more fuel efficient every year.

```
par(mfrow=c(2,2))
plot(lm.fit0)
```



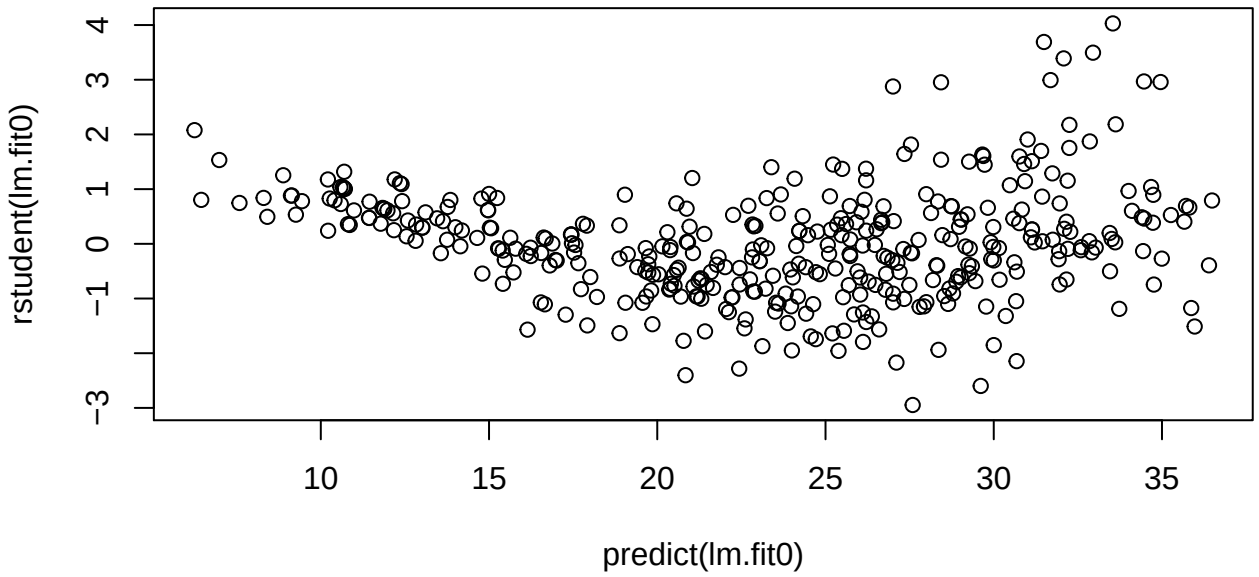
From the Residual vs Fitted plot, there seems to be a quadratic relationship between the residuals and the fitted values, suggesting that non-linearity between the predictors and response. Polynomial regression or non-linear transformation such as interaction of the variables may be needed.

The Scale-Location plot, also known as the Spread-Location plot, shows if residuals are spread equally along the ranges of predictors. Homoscedasticity seems not to hold in the model as indicated by the non-horizontal line with equally (randomly) spread points.

The QQ plot displays a steeper slope on the right tail, implying a positive skewness of the residuals.

The Residual vs Leverage plot suggests that `buick estate wagon (sw)` (obversation 14) has high leverage, despite not a high magnitude residual.

```
plot(predict(lm.fit0), rstudent(lm.fit0))
```



There are possible outliers as seen in the plot of studentized residuals because there are data with a value greater than 3.

```
# Interaction Terms
lm.fit0 <- lm(mpg ~ . -name, data=Auto)
lm.fit1 <- lm(mpg~cylinders+weight*cylinders+year+origin, data=Auto)
lm.fit2 <- lm(mpg~acceleration+weight*acceleration+year+origin, data=Auto)
lm.fit3 <- lm(mpg~horsepower+weight*horsepower+year+origin, data=Auto)
summary(lm.fit0)
```

```
##
## 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
```

```
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders + weight * cylinders + year + origin,
##     data = Auto)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -11.5277  -1.7587  -0.2015   1.5147  12.7885
##
## Coefficients:
##                    Estimate Std. Error t value Pr(>|t|)
## (Intercept)       3.8845201  4.5378720   0.856   0.3925
## cylinders        -4.4875789  0.5639369  -7.958 1.97e-14 ***
## weight           -0.0144268  0.0010804 -13.353  < 2e-16 ***
## year              0.8115738  0.0455113  17.832  < 2e-16 ***
## origin            0.5345940  0.2506425   2.133   0.0336 *
## cylinders:weight  0.0013967  0.0001637   8.533 3.30e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.078 on 386 degrees of freedom
## Multiple R-squared:  0.8464, Adjusted R-squared:  0.8444
## F-statistic: 425.5 on 5 and 386 DF,  p-value: < 2.2e-16
```

```
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = mpg ~ acceleration + weight * acceleration + year +
##     origin, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1780 -2.0869  0.1691  1.7532 12.2679
##
## Coefficients:
##                        Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -4.639e+01  5.612e+00  -8.266 2.26e-15 ***
## acceleration         1.602e+00  2.352e-01   6.813 3.67e-11 ***
## weight               1.481e-03  1.124e-03   1.318 0.188315
## year                 8.134e-01  4.762e-02  17.081  < 2e-16 ***
## origin               9.390e-01  2.478e-01   3.789 0.000175 ***
## acceleration:weight -5.024e-04  7.457e-05  -6.738 5.85e-11 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.169 on 386 degrees of freedom
## Multiple R-squared:  0.8372, Adjusted R-squared:  0.8351
## F-statistic: 397.1 on 5 and 386 DF,  p-value: < 2.2e-16
```

```
summary(lm.fit3)
```

```
##
## Call:
## lm(formula = mpg ~ horsepower + weight * horsepower + year +
##     origin, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.6051 -1.7722 -0.1304  1.5205 12.0369
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)        8.145e-01  3.969e+00   0.205  0.83753
## horsepower        -2.160e-01  2.055e-02 -10.514  < 2e-16 ***
## weight            -1.106e-02  6.343e-04 -17.435  < 2e-16 ***
## year               7.677e-01  4.464e-02  17.195  < 2e-16 ***
## origin             7.224e-01  2.328e-01   3.103  0.00206 **
## horsepower:weight  5.501e-05  5.051e-06  10.891  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 2.931 on 386 degrees of freedom
## Multiple R-squared:  0.8608, Adjusted R-squared:  0.859
## F-statistic: 477.5 on 5 and 386 DF,  p-value: < 2.2e-16
```

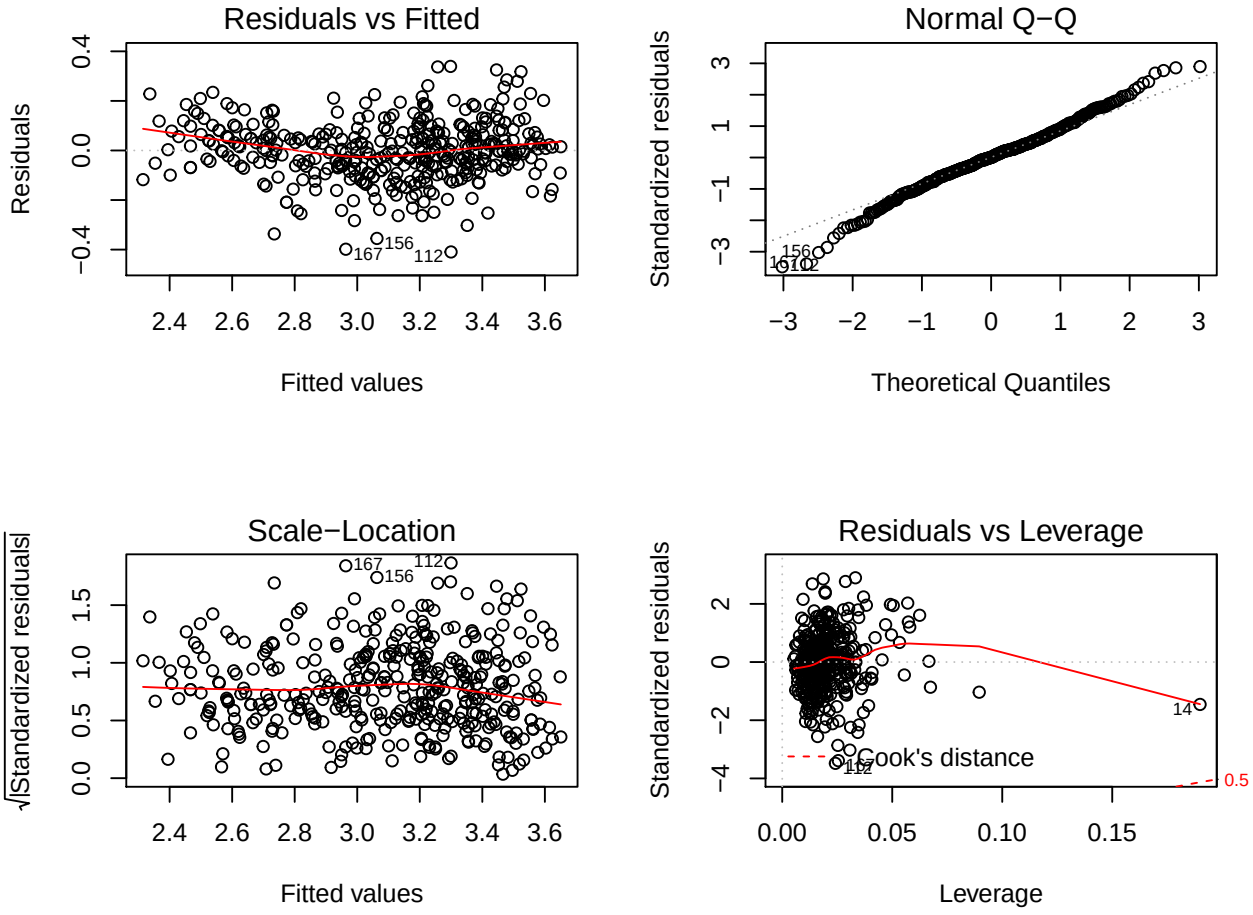
Insignificant variables's effect to `mpg` maybe captured by *synergy* or interaction terms. Demonstarated aboved, interaction bwteen `weight` and `cylinders` (`lm.fit1`), bwteen `weight` and `acceleration` (`lm.fit2`), bwteen `weight` and and `horsepower` (`lm.fit3`) are all statistically significant.

```
# Non-linear Transformations of the Predictors
lm.fit4<-lm(log(mpg)~cylinders+displacement+horsepower+weight+acceleration+year+origin,data=Auto)
summary(lm.fit2)
```

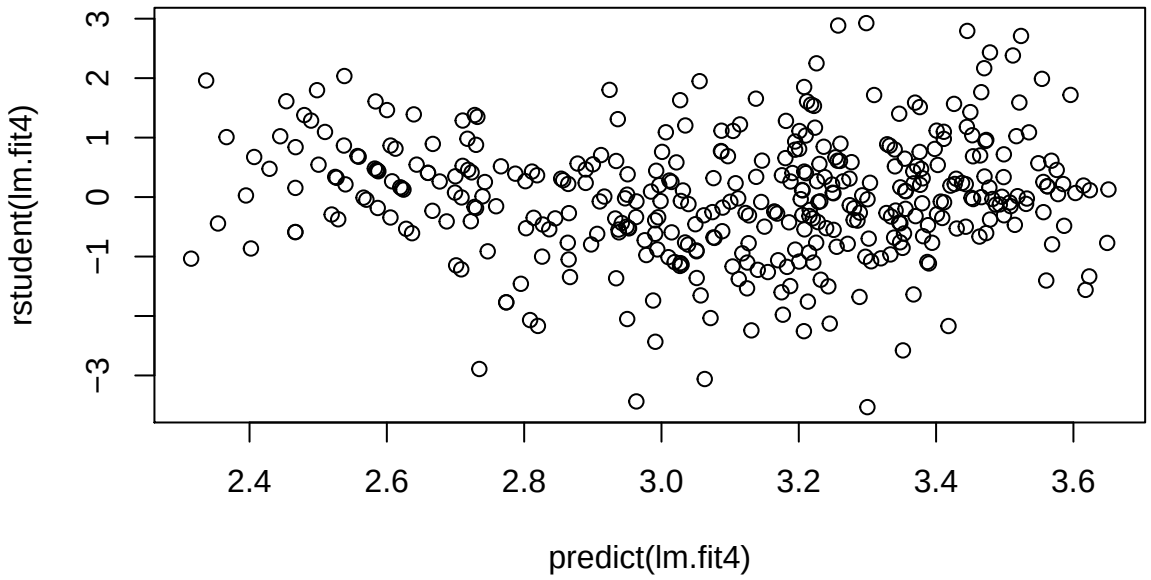
```
##
## Call:
## lm(formula = mpg ~ acceleration + weight * acceleration + year +
##     origin, data = Auto)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -8.1780 -2.0869  0.1691  1.7532 12.2679
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -4.639e+01  5.612e+00  -8.266 2.26e-15 ***
## acceleration         1.602e+00  2.352e-01   6.813 3.67e-11 ***
## weight               1.481e-03  1.124e-03   1.318 0.188315
## year                 8.134e-01  4.762e-02  17.081  < 2e-16 ***
## origin               9.390e-01  2.478e-01   3.789 0.000175 ***
```

```
## acceleration:weight -5.024e-04  7.457e-05  -6.738 5.85e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3.169 on 386 degrees of freedom
## Multiple R-squared:  0.8372, Adjusted R-squared:  0.8351
## F-statistic: 397.1 on 5 and 386 DF,  p-value: < 2.2e-16
```

```
par(mfrow=c(2,2))
plot(lm.fit4)
```




```
par(mfrow=c(1,1))
plot(predict(lm.fit4),rstudent(lm.fit4))
```

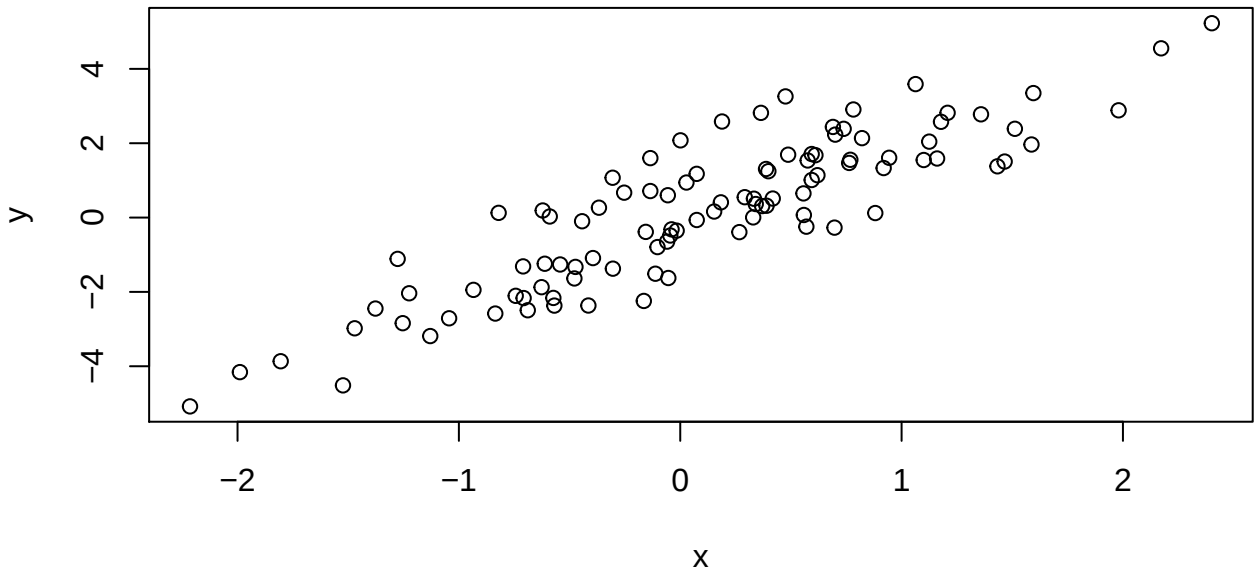


As indicated by the Residual vs Fitted plot, QQ plot and the Scale-Location plot, heteroskedasticity appears to be a feature in the previous model with linear predictors. Also in the scatter matrix, `displacement`, `horsepower` and `weight` show a similar nonlinear pattern against response `mpg`. Non-linear transformations of the predictors may be appropriate. Using `log(mpg)` as the response variable, the outputs show that log transform of `mpg` yield a higher R^2 and residuals more normally distributed.

###Question 11

This and the next questions (as well as question 5) ask about simple linear regression without an intercept.

```
set.seed (1)
x=rnorm (100)
y=2*x+rnorm (100)
plot(x,y)
```




```
# Regress y on x. Result is highly significant
lm.fit0 <- lm(y~x+0)
```

```r
summary(lm.fit0)
```

```
##
## Call:
## lm(formula = y ~ x + 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9154 -0.6472 -0.1771  0.5056  2.3109
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x   1.9939     0.1065   18.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9586 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

```r
# Regress x on y. Result is highly significant
lm.fit1 <- lm(x~y+0)
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = x ~ y + 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.8699 -0.2368  0.1030  0.2858  0.8938
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## y  0.39111    0.02089   18.73   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4246 on 99 degrees of freedom
## Multiple R-squared:  0.7798, Adjusted R-squared:  0.7776
## F-statistic: 350.7 on 1 and 99 DF,  p-value: < 2.2e-16
```

First, the multiple R^2, adjusted R^2, t-statistics, and F-statistics are the same in the two models. Second, since $\hat{x} = \hat{\beta}_x y$ versus $\hat{y} = \hat{\beta}_y x$, so the betas should be inverse of each other ($\hat{\beta}_x = \frac{1}{\hat{\beta}_y}$) but they are somewhat off here ($\frac{1}{0.39111} = 2.557 \neq 1.994$).

```r
lm.fit = lm(y~x)
lm.fit2 = lm(x~y)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = y ~ x)
##
## Residuals:
```

```
##     Min      1Q  Median      3Q     Max
## -1.8768 -0.6138 -0.1395  0.5394  2.3462
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.03769    0.09699  -0.389    0.698
## x            1.99894    0.10773  18.556   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9628 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

```
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = x ~ y)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.90848 -0.28101  0.06274  0.24570  0.85736
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03880    0.04266    0.91    0.365
## y            0.38942    0.02099   18.56   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.4249 on 98 degrees of freedom
## Multiple R-squared:  0.7784, Adjusted R-squared:  0.7762
## F-statistic: 344.3 on 1 and 98 DF,  p-value: < 2.2e-16
```

The t-statistics are the same.

###Question 12

Generate an example in R with n = 100 observations in which the coefficient estimate for the regression of X onto Y is different from the coefficient estimate for the regression of Y onto X.

```
# Question 11a is the example in point.
```

Generate an example in R with n = 100 observations in which the coefficient estimate for the regression of X onto Y is the same as the coefficient estimate for the regression of Y onto X.

Focus on the denominator in equation 3.38. If $\sum(x_{i'}^2) = \sum(y_{i'}^2)$, $\hat{\beta}$ of regressing y on x will be equal to that of regressing x on y. To illusrate, see

```
set.seed(1)
x <- rnorm(100)
# Generate random sample (i.e. y ) from x without replacement
y <- -sample(x, 100)
# suh that:
sum(x^2)==sum(y^2)
```

```
## [1] TRUE
```

```
lm.fit_x <- lm(y~x+0)
lm.fit_y <- lm(x~y+0)
summary(lm.fit_x)
```

```
##
## Call:
## lm(formula = y ~ x + 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.3926 -0.6877 -0.1027  0.5124  2.2315
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## x -0.02148    0.10048  -0.214    0.831
##
## Residual standard error: 0.9046 on 99 degrees of freedom
## Multiple R-squared:  0.0004614,  Adjusted R-squared:  -0.009635
## F-statistic: 0.0457 on 1 and 99 DF,  p-value: 0.8312
```

```
summary(lm.fit_y)
```

```
##
## Call:
## lm(formula = x ~ y + 0)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.2400 -0.5154  0.1213  0.6788  2.3959
##
## Coefficients:
##   Estimate Std. Error t value Pr(>|t|)
## y -0.02148    0.10048  -0.214    0.831
##
## Residual standard error: 0.9046 on 99 degrees of freedom
## Multiple R-squared:  0.0004614,  Adjusted R-squared:  -0.009635
## F-statistic: 0.0457 on 1 and 99 DF,  p-value: 0.8312
```

###Question 14

This problem focuses on the collinearity problem.

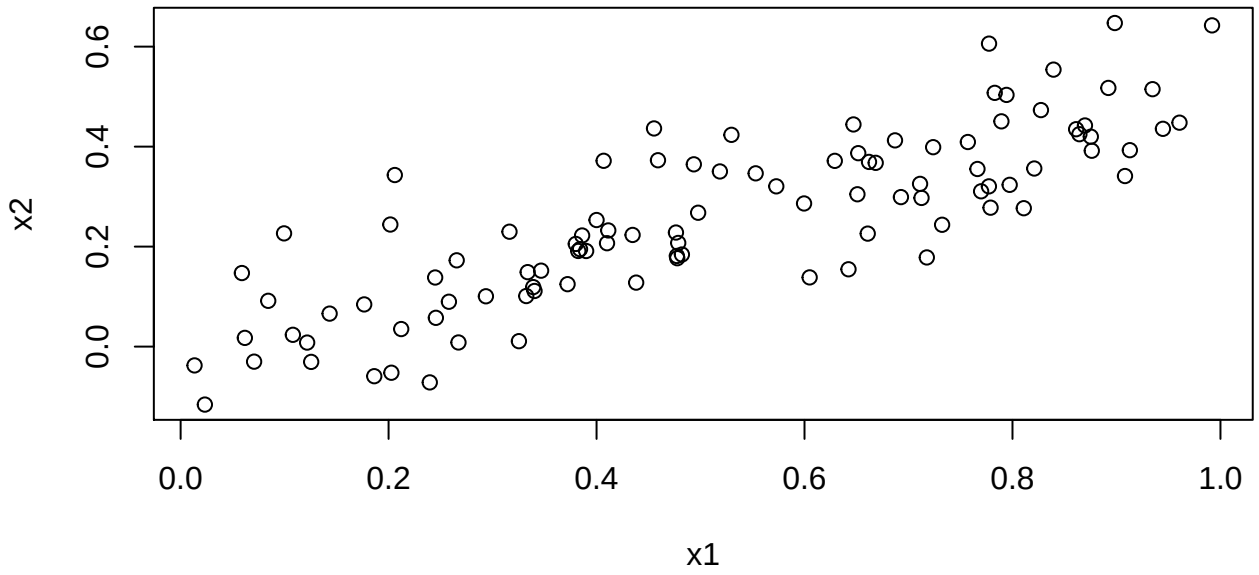
```
set.seed (1)
x1=runif (100)
x2 =0.5*x1+rnorm (100) /10
y=2+2*x1 +0.3*x2+rnorm (100)
```

The form of the linear model is $y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \varepsilon$ where $\beta_0 = 2$, $\beta_1 = 2$ and $\beta_2 = 0.3$.

```
cor(x1,x2)
```

```
## [1] 0.8351212
```

```
plot(x1,x2)
```




```
lm.fit <- lm(y~x1+x2)
summary(lm.fit)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -2.8311 -0.7273 -0.0537  0.6338  2.3359
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1305     0.2319   9.188 7.61e-15 ***
## x1            1.4396     0.7212   1.996   0.0487 *
## x2            1.0097     1.1337   0.891   0.3754
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.056 on 97 degrees of freedom
## Multiple R-squared:  0.2088, Adjusted R-squared:  0.1925
## F-statistic:  12.8 on 2 and 97 DF,  p-value: 1.164e-05
```

Estimated beta coefficients are $\hat{\beta}_0 = 2.13$, $\hat{\beta}_1 = 1.44$ and $\hat{\beta}_2 = 1.01$. Coefficient for x1 is statistically significant but the coefficient for x2 is not. Null hypothesis for x_1, $H_0 : \beta_1 = 0$, is rejected at 0.01 significant level while that of x_2, $H_0 : \beta_2 = 0$, is retained.

```
par(mfrow=c(2,1), mar=c(2, 3, 2, 1), mgp=c(2, 0.8, 0))
lm.fit1 <- lm(y~x1)
summary(lm.fit1)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
```

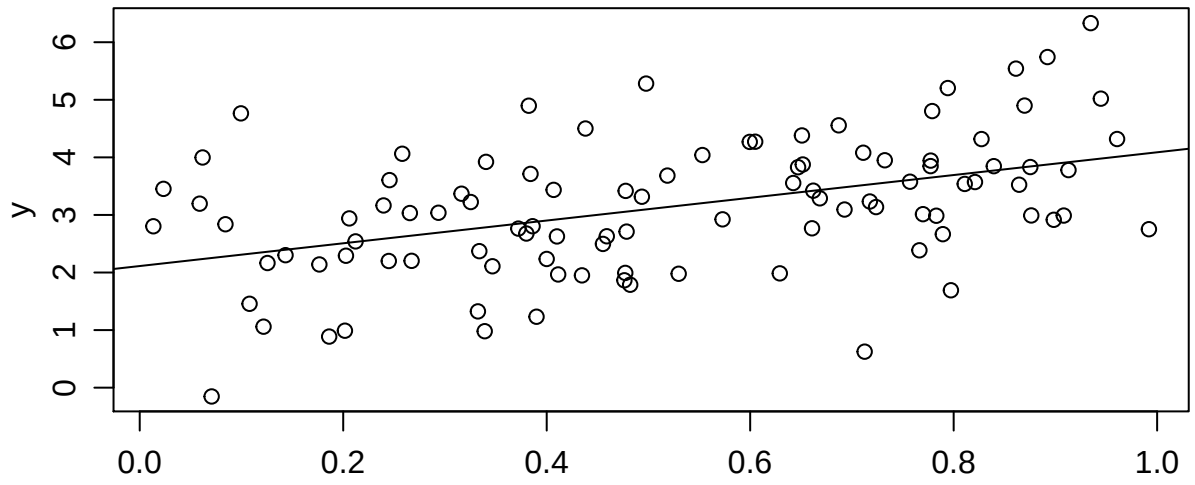
```
## -2.89495 -0.66874 -0.07785  0.59221  2.45560
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.1124     0.2307   9.155 8.27e-15 ***
## x1            1.9759     0.3963   4.986 2.66e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.055 on 98 degrees of freedom
## Multiple R-squared:  0.2024, Adjusted R-squared:  0.1942
## F-statistic: 24.86 on 1 and 98 DF,  p-value: 2.661e-06
```

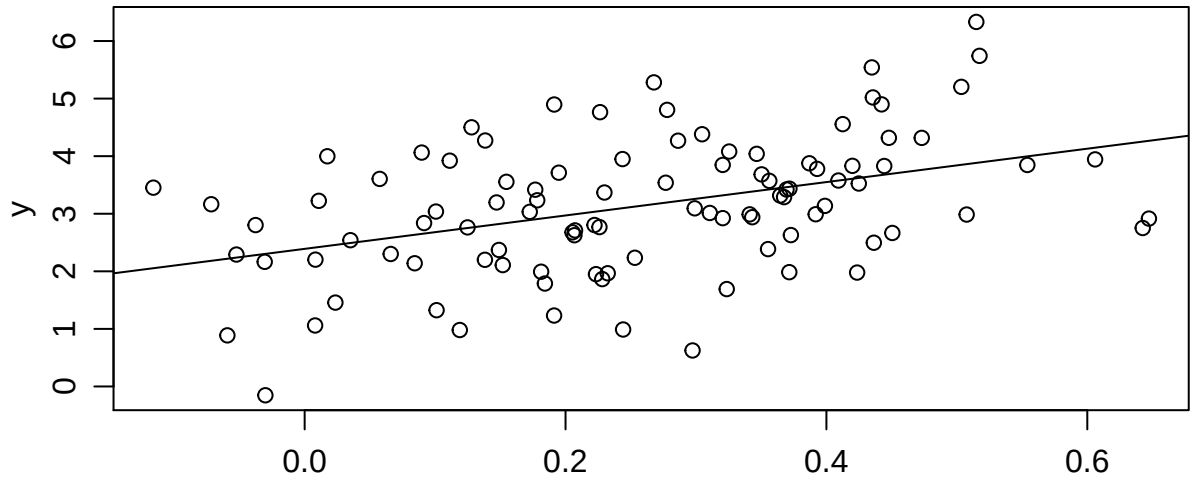
```
plot(x1,y)
abline(lm.fit1)

lm.fit2 <- lm(y~x2)
summary(lm.fit2)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.62687 -0.75156 -0.03598  0.72383  2.44890
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3899     0.1949   12.26  < 2e-16 ***
## x2            2.8996     0.6330    4.58 1.37e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.072 on 98 degrees of freedom
## Multiple R-squared:  0.1763, Adjusted R-squared:  0.1679
## F-statistic: 20.98 on 1 and 98 DF,  p-value: 1.366e-05
```

```
plot(x2,y)
abline(lm.fit2)
```





Individually, both x_1 and x_2 enter the simple regression model with highly significant statistical levels.

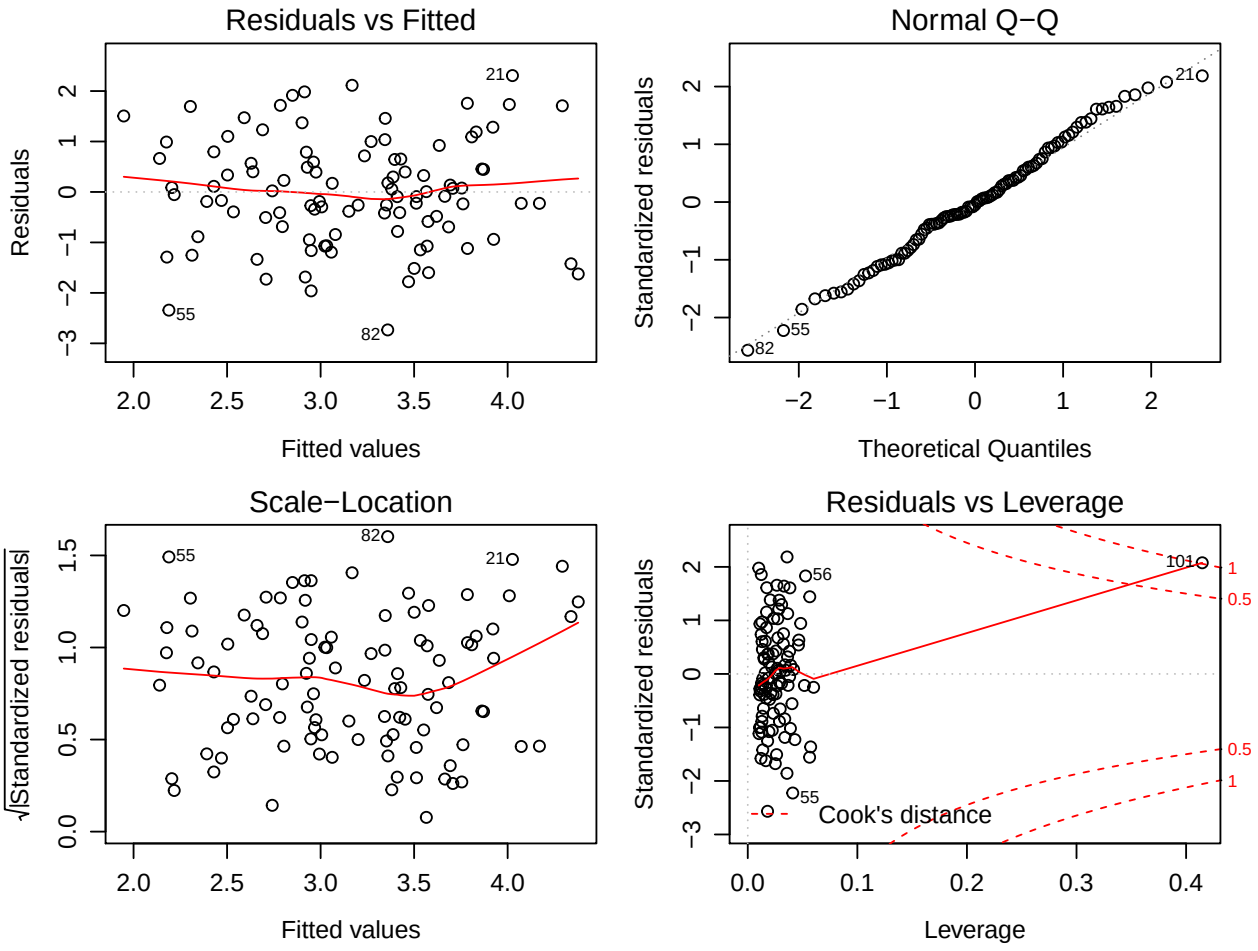
There is no contradiction. The problem lies in collinearity. It is hard to distinguish their individual effects from the combined effects when regressed upon together.

```
x1=c(x1, 0.1)
x2=c(x2, 0.8)
y=c(y,6)
par(mfrow=c(2,2), mar=c(3.5, 3.5, 2, 1), mgp=c(2.4, 0.8, 0))
# regression with both x1 and x2
fit.lm <- lm(y~x1+x2)
summary(fit.lm)
```

```
##
## Call:
## lm(formula = y ~ x1 + x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.73348 -0.69318 -0.05263  0.66385  2.30619
##
## Coefficients:
```

```
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2267     0.2314   9.624 7.91e-16 ***
## x1            0.5394     0.5922   0.911  0.36458
## x2            2.5146     0.8977   2.801  0.00614 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.075 on 98 degrees of freedom
## Multiple R-squared:  0.2188, Adjusted R-squared:  0.2029
## F-statistic: 13.72 on 2 and 98 DF,  p-value: 5.564e-06
```

```
plot(fit.lm)
```

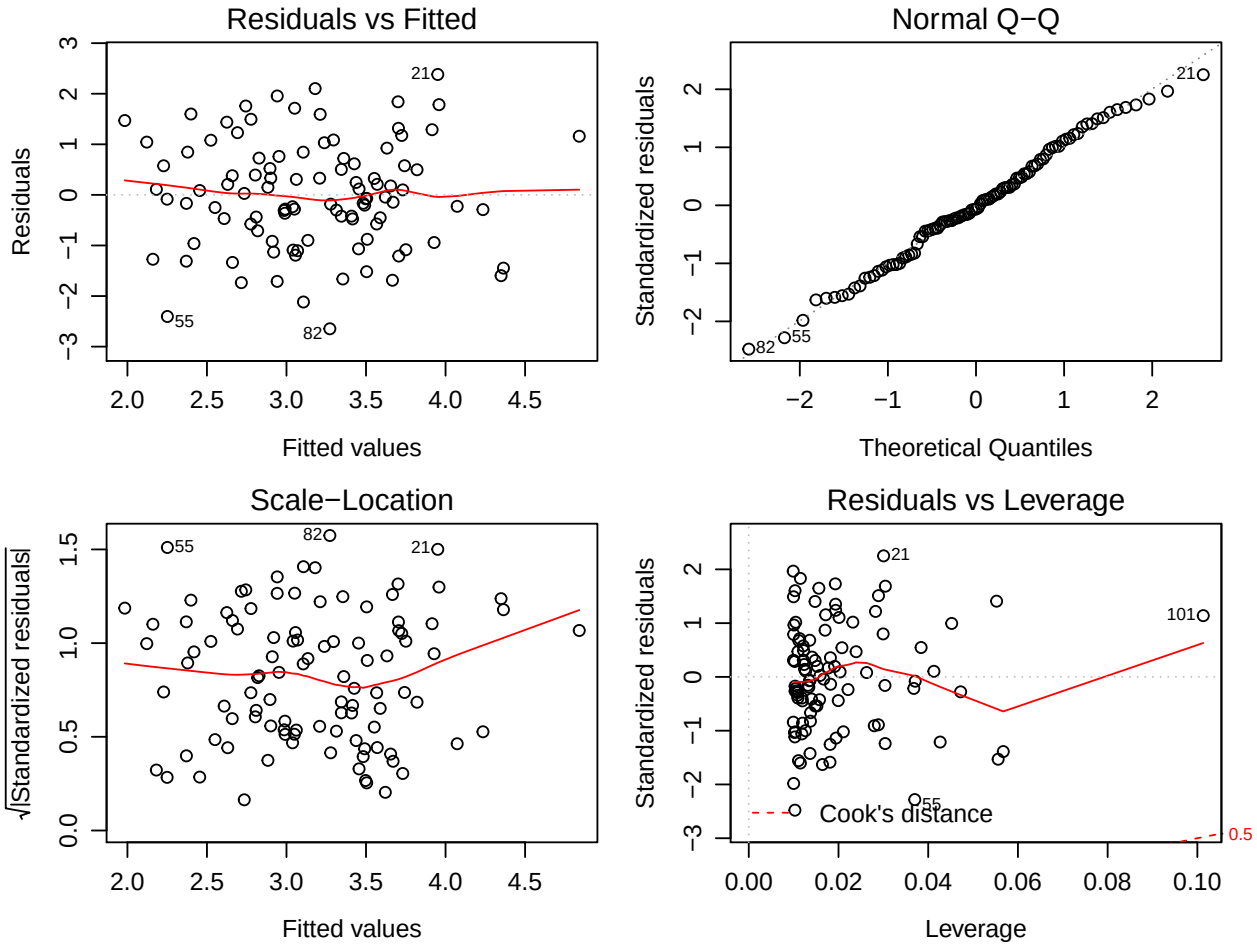



```
# regression with x1 only
fit.lm1 <- lm(y~x2)
summary(fit.lm1)
```

```
##
## Call:
## lm(formula = y ~ x2)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -2.64729 -0.71021 -0.06899  0.72699  2.38074
```

```
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.3451     0.1912  12.264  < 2e-16 ***
## x2            3.1190     0.6040   5.164 1.25e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.074 on 99 degrees of freedom
## Multiple R-squared:  0.2122, Adjusted R-squared:  0.2042
## F-statistic: 26.66 on 1 and 99 DF,  p-value: 1.253e-06
```

```
plot(fit.lm1)
```

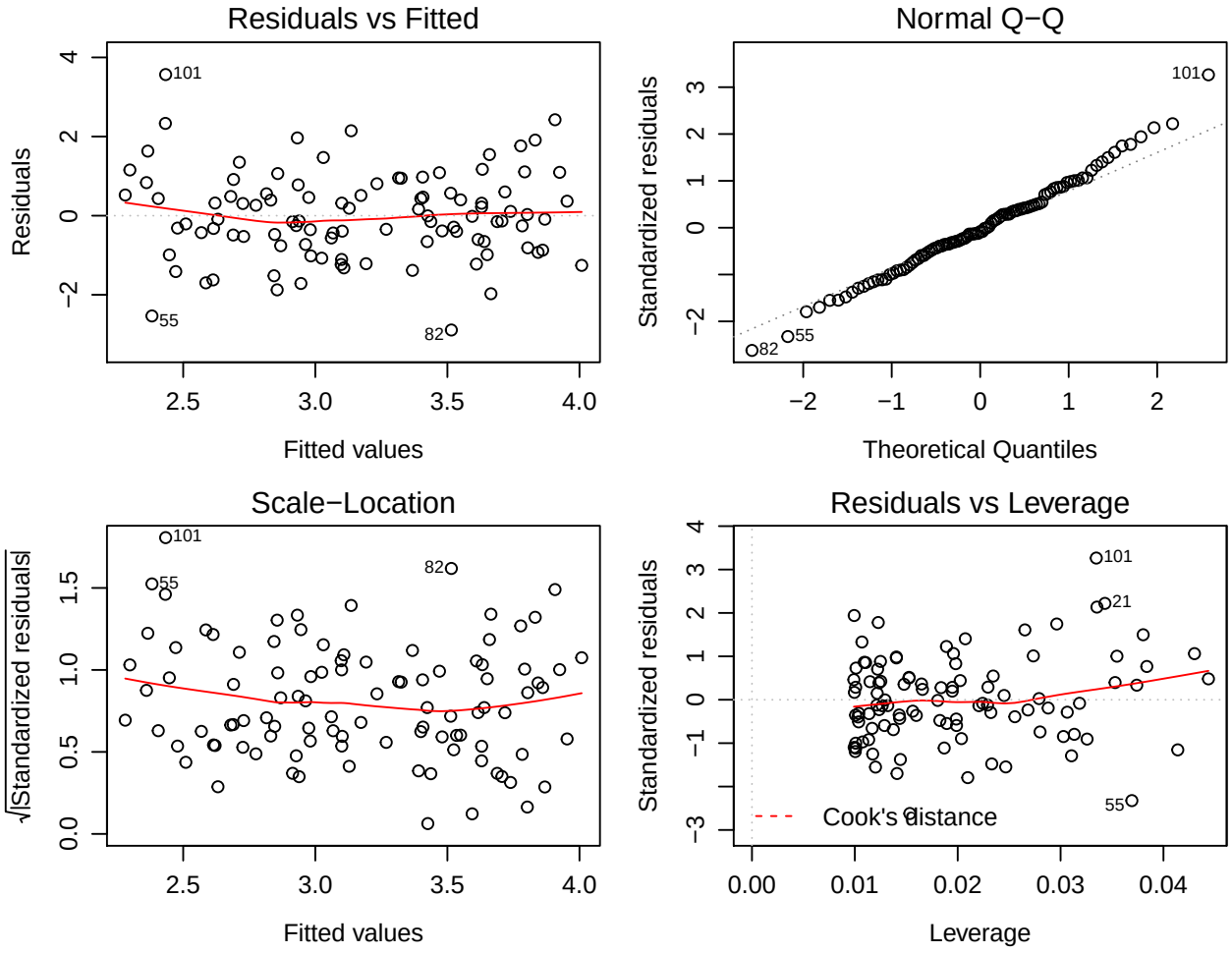



```
# regression with x2 only
fit.lm2 <- lm(y~x1)
summary(fit.lm2)
```

```
##
## Call:
## lm(formula = y ~ x1)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
```

```
## -2.8897 -0.6556 -0.0909  0.5682  3.5665
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   2.2569     0.2390   9.445 1.78e-15 ***
## x1            1.7657     0.4124   4.282 4.29e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.111 on 99 degrees of freedom
## Multiple R-squared:  0.1562, Adjusted R-squared:  0.1477
## F-statistic: 18.33 on 1 and 99 DF,  p-value: 4.295e-05
```

```
plot(fit.lm2)
```



The new observation ($[y, x_1, x_2] = [6, 0.1, 0.8]$) is an outlier for x_2 and has high leverage for both x_1 and x_2. From the residuals vs leverage plot, observation 101 falls on the right hand side in all three models. In particular, it stands out as the red line is extensivelt tilted relative to the dotted black line indicating high leverage (Cook's Distance) for the model in which x_1 and x_2 are the predictors of y.