Statistical Learning - Homework 3 (Applied)

James Hahn

Chapter 4 - Exercise 11

a)

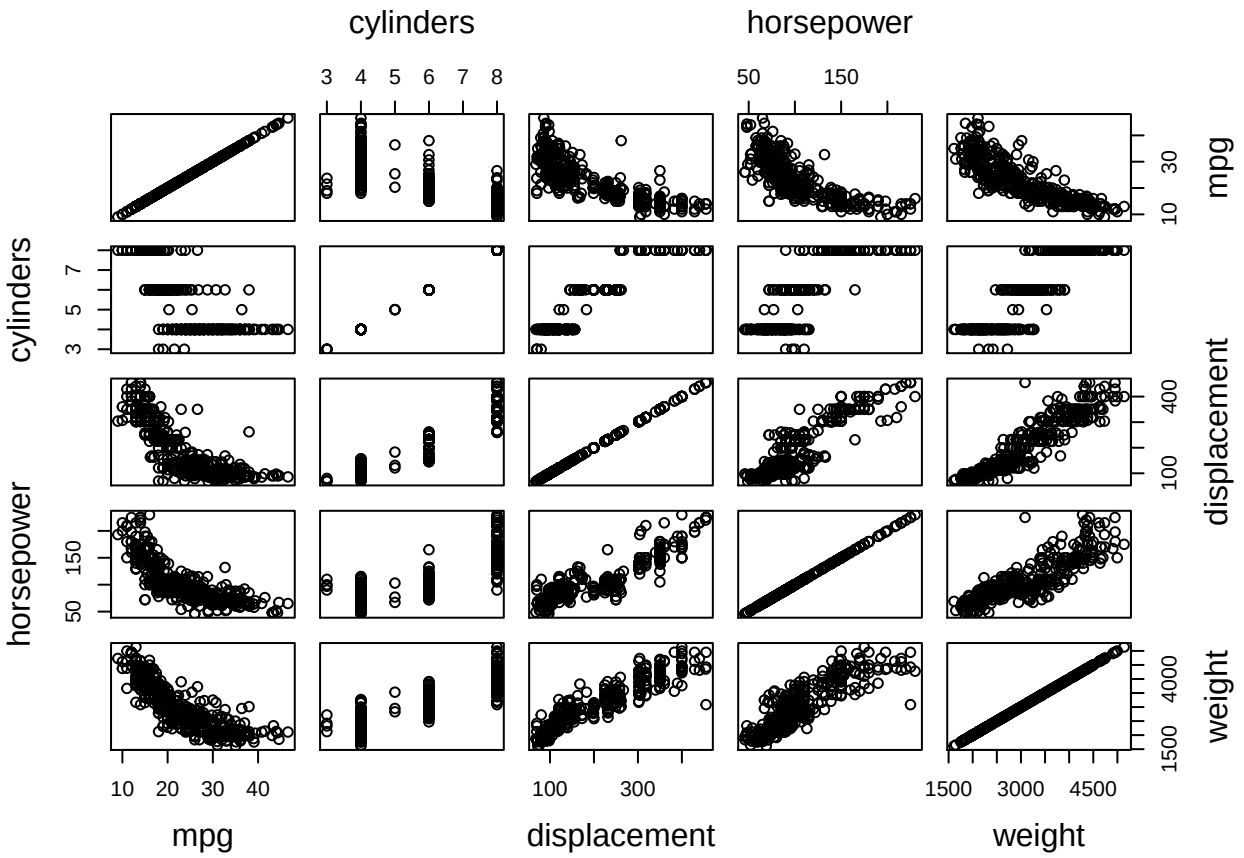
```
library(ISLR)
library(MASS)
library(class)

attach(Auto)
mpg01 <- rep(0, length(mpg))
mpg01[mpg > median(mpg)] <- 1
Auto <- data.frame(Auto, mpg01)
```

b)

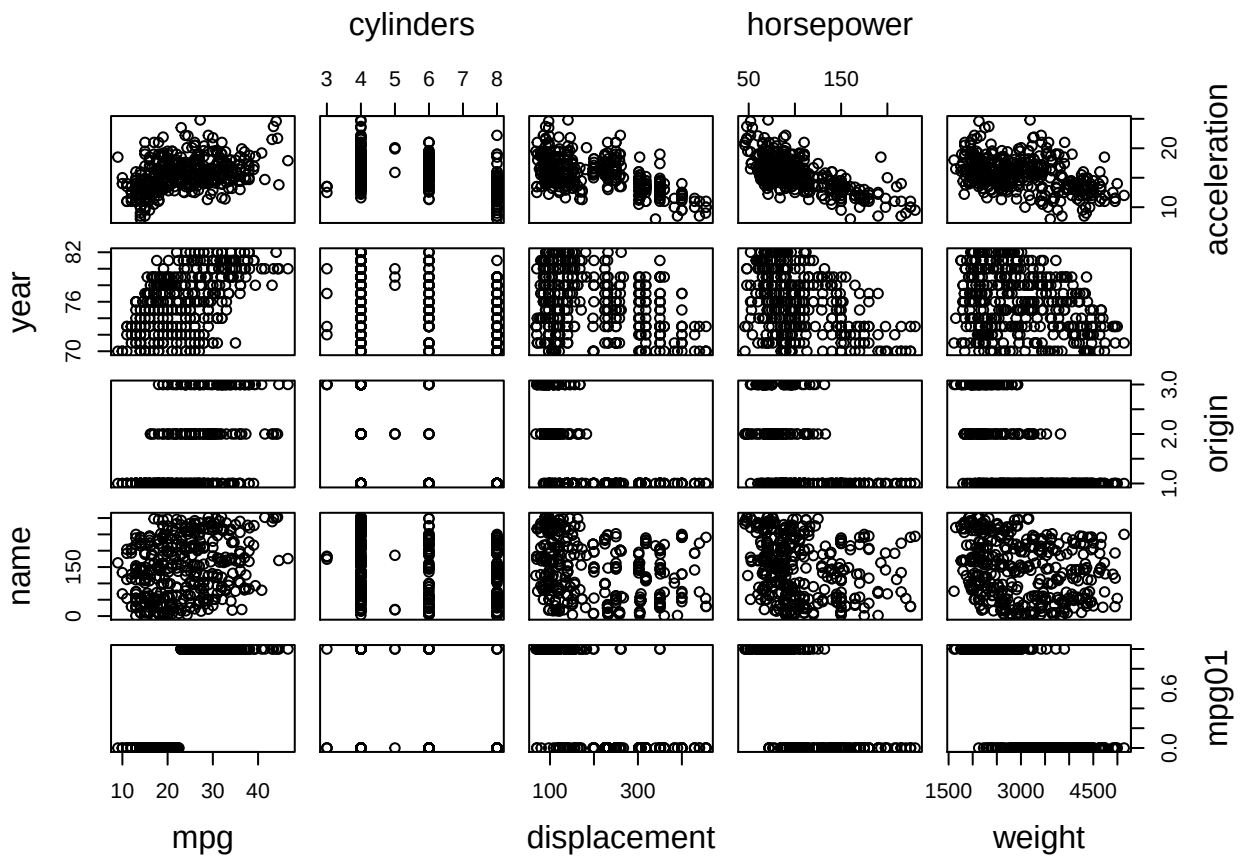
```
library(TeachingDemos)

cor(Auto[, -9])
```

```
##                     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
## mpg01         0.8369392 -0.7591939   -0.7534766 -0.6670526 -0.7577566
##              acceleration       year     origin      mpg01
## mpg             0.4233285  0.5805410  0.5652088  0.8369392
## cylinders      -0.5046834 -0.3456474 -0.5689316 -0.7591939
## displacement   -0.5438005 -0.3698552 -0.6145351 -0.7534766
## horsepower     -0.6891955 -0.4163615 -0.4551715 -0.6670526
## weight         -0.4168392 -0.3091199 -0.5850054 -0.7577566
## acceleration    1.0000000  0.2903161  0.2127458  0.3468215
## year            0.2903161  1.0000000  0.1815277  0.4299042
## origin          0.2127458  0.1815277  1.0000000  0.5136984
## mpg01           0.3468215  0.4299042  0.5136984  1.0000000
```

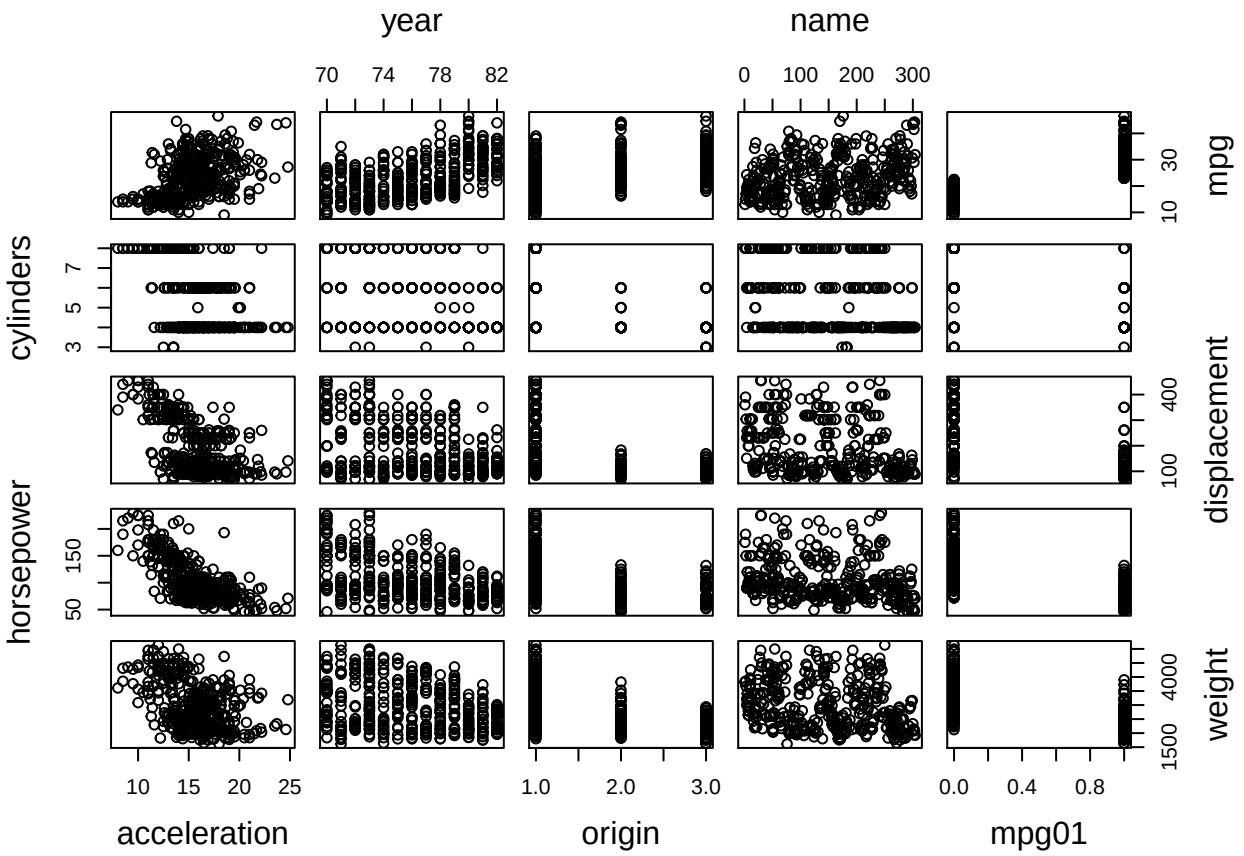
```
pairs2(Auto[,1:5], Auto[,1:5])
```



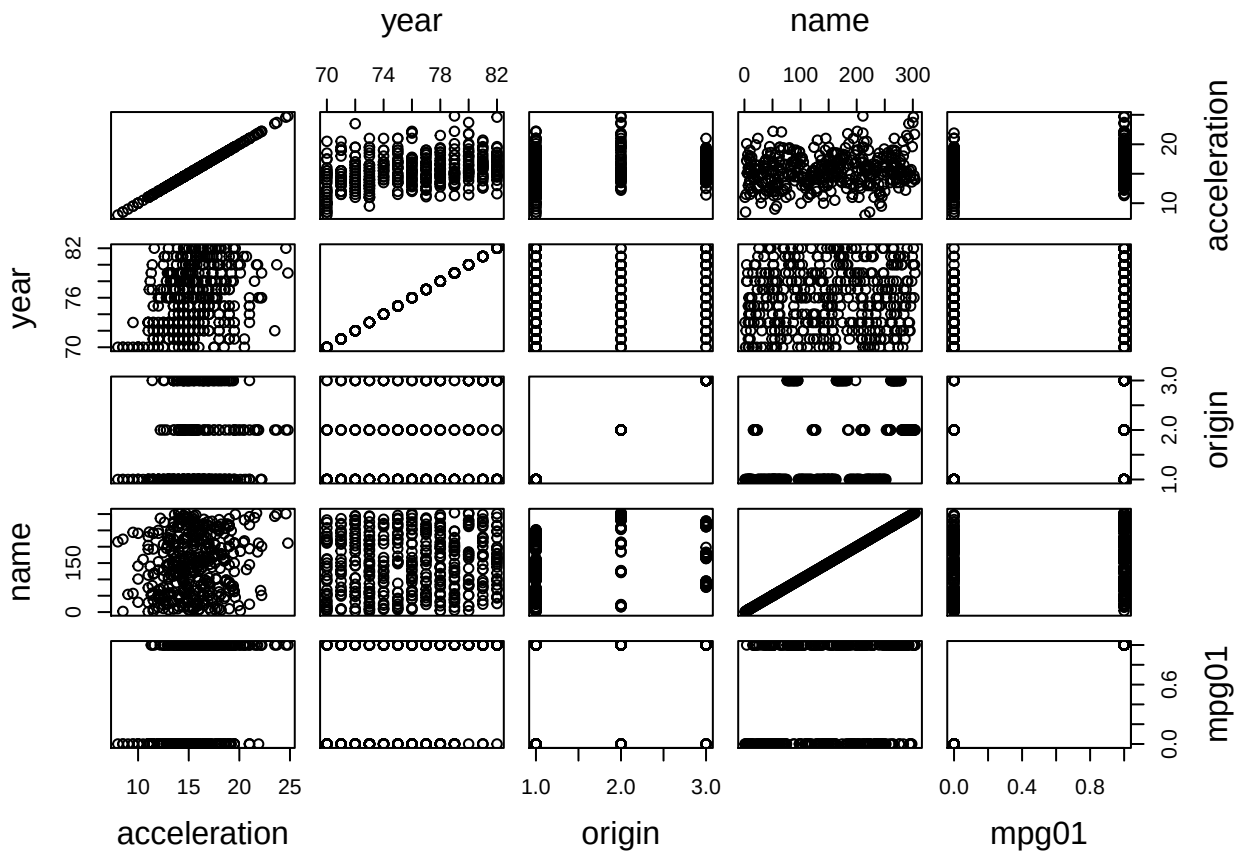

```
pairs2(Auto[,1:5], Auto[,6:10])
```



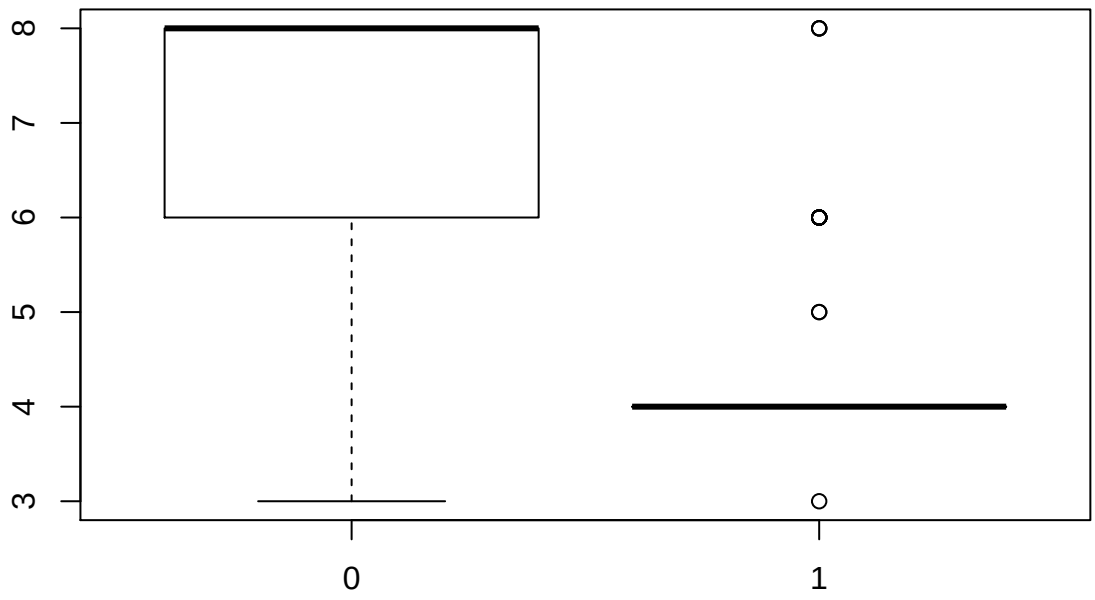

```
pairs2(Auto[,6:10], Auto[,1:5])
```



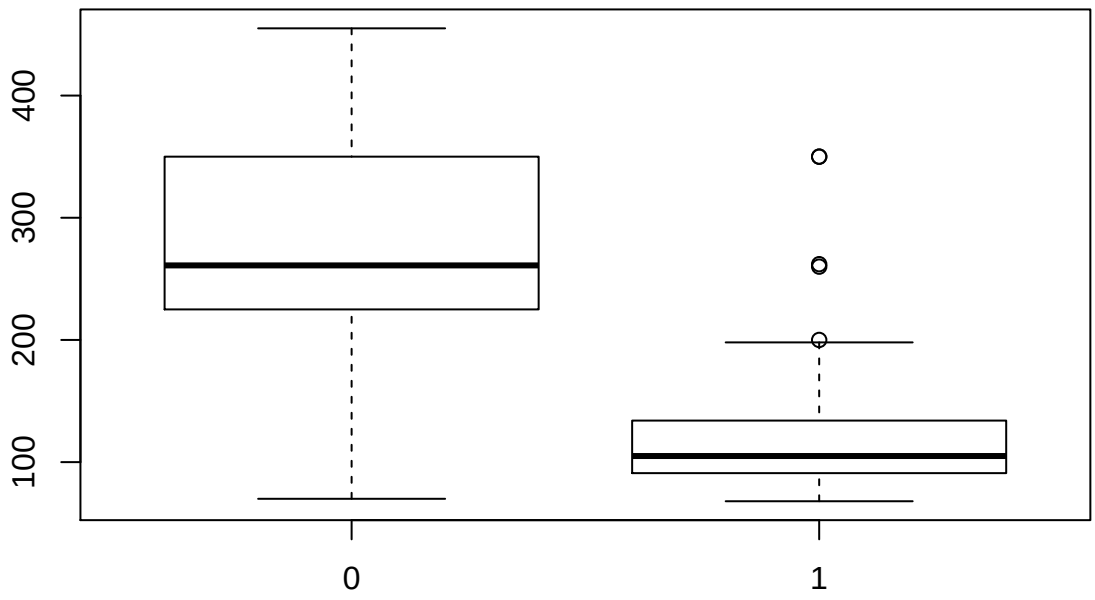

```
pairs2(Auto[,6:10], Auto[,6:10])
```



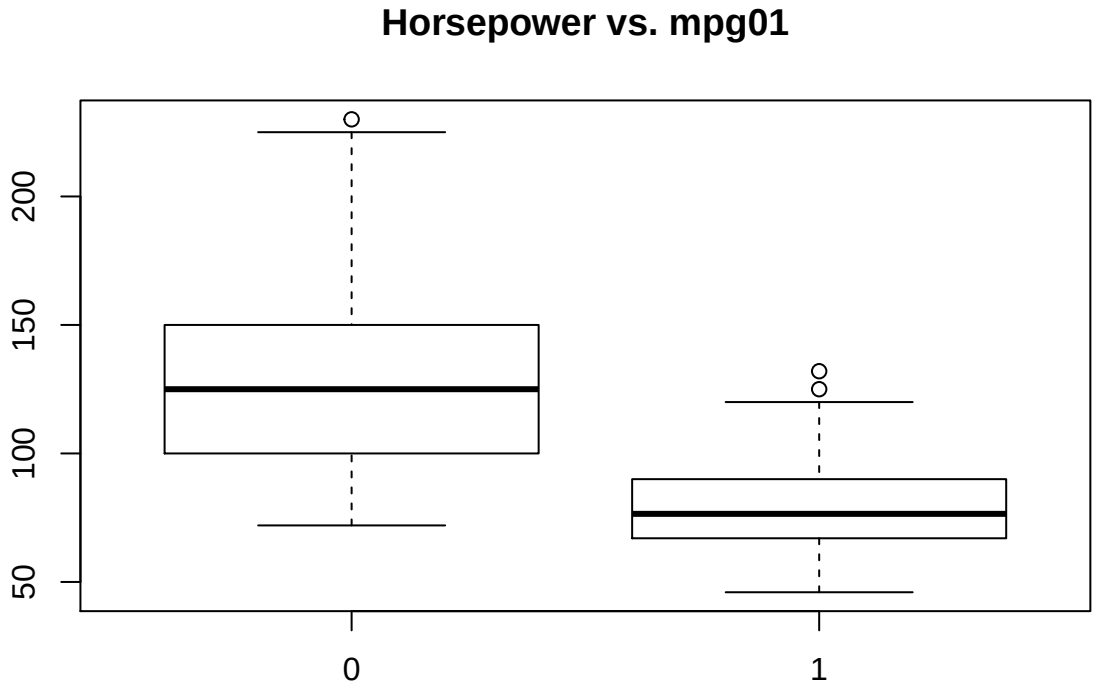

```
boxplot(cylinders ~ mpg01, data = Auto, main = "Cylinders vs. mpg01")
```



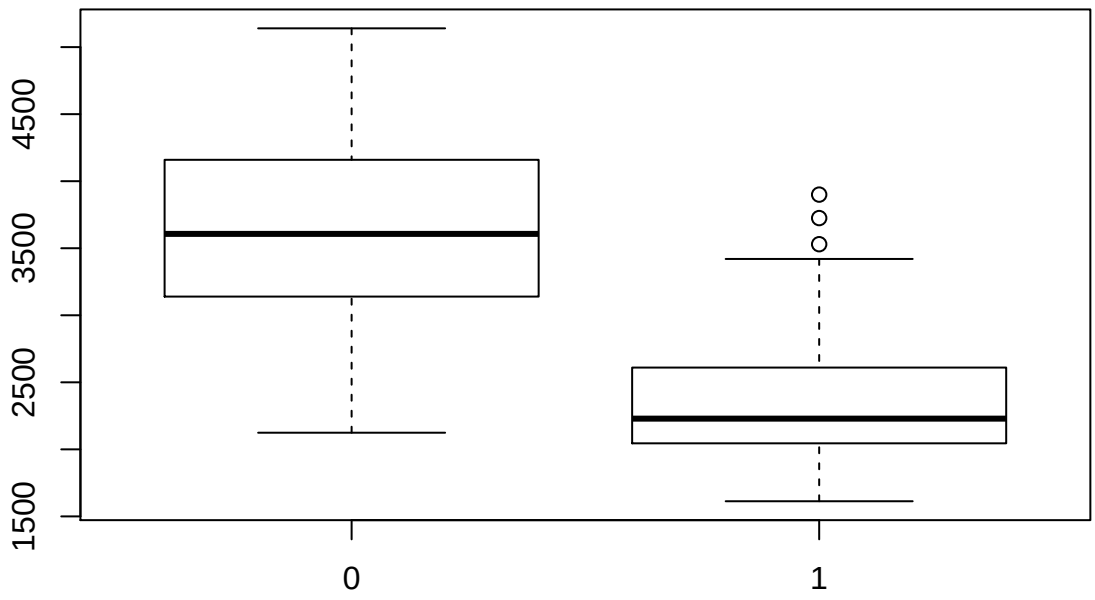

```
boxplot(displacement ~ mpg01, data = Auto, main = "Displacement vs. mpg01")
```



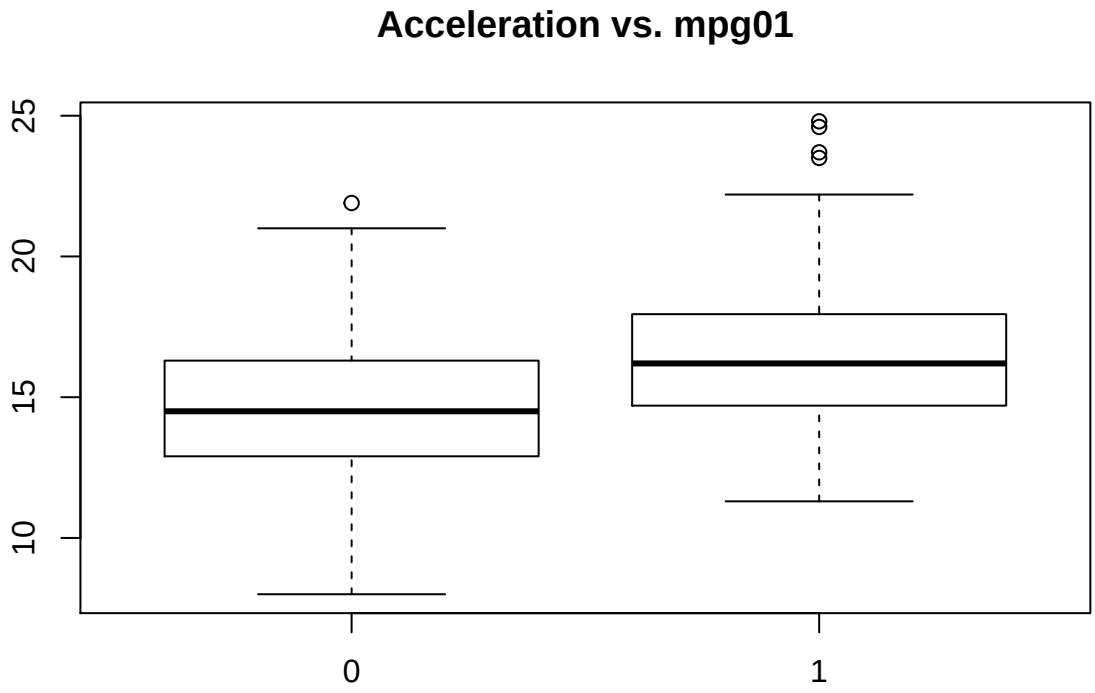

```
boxplot(horsepower ~ mpg01, data = Auto, main = "Horsepower vs. mpg01")
```



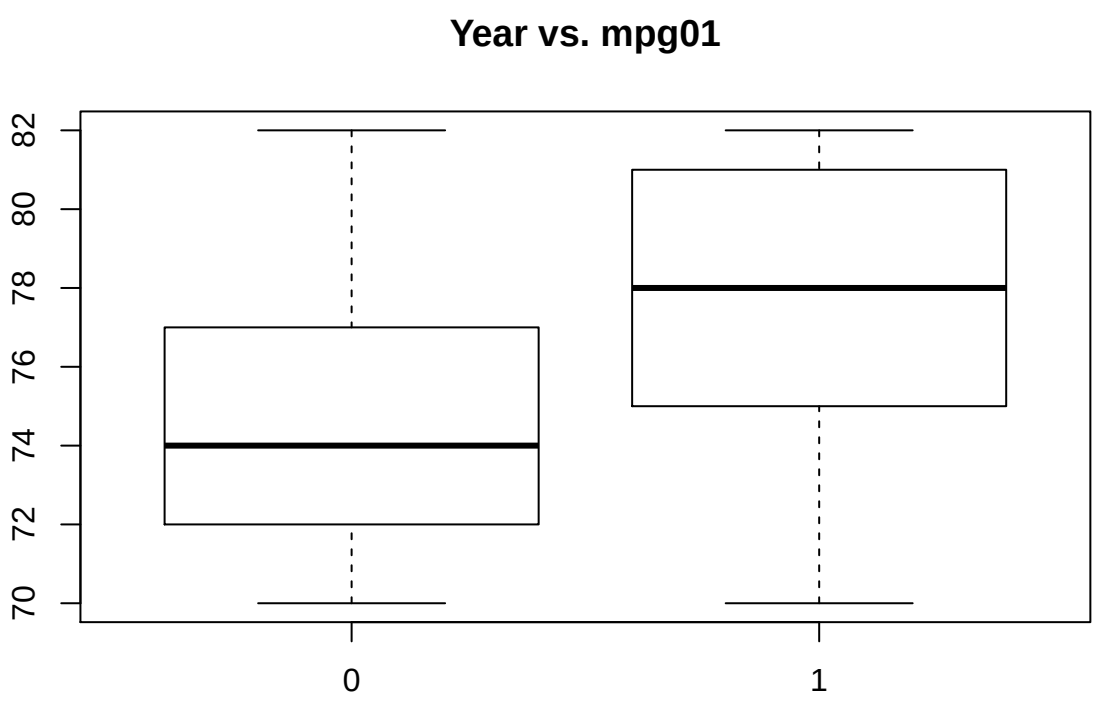

```
boxplot(weight ~ mpg01, data = Auto, main = "Weight vs. mpg01")
```




```r
boxplot(acceleration ~ mpg01, data = Auto, main = "Acceleration vs. mpg01")
```




```
boxplot(year ~ mpg01, data = Auto, main = "Year vs. mpg01")
```



From the above graphs, we can see there is some relation between mpg01 and the following features: cylinders, weight, displacement, and horsepower.

c)

```
train <- (year %% 2 == 0)
Auto.train <- Auto[train, ]
Auto.test <- Auto[!train, ]
mpg01.test <- mpg01[!train]
```

d)

```
fit.lda <- lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, subset = train)
fit.lda
```

```
## Call:
## lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto,
##     subset = train)
##
## Prior probabilities of groups:
##         0         1
## 0.4571429 0.5428571
##
## Group means:
##   cylinders   weight displacement horsepower
## 0  6.812500 3604.823     271.7396  133.14583
## 1  4.070175 2314.763     111.6623   77.92105
##
```

```
## Coefficients of linear discriminants:
##                         LD1
## cylinders    -0.6741402638
## weight       -0.0011465750
## displacement  0.0004481325
## horsepower    0.0059035377
```

```
pred.lda <- predict(fit.lda, Auto.test)
table(pred.lda$class, mpg01.test)
```

```
##    mpg01.test
##      0  1
##   0 86  9
##   1 14 73
```

```
mean(pred.lda$class != mpg01.test)
```

```
## [1] 0.1263736
```

The test error is 12.63736%.

e)

```
fit.qda <- qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, subset = train)
fit.qda
```

```
## Call:
## qda(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto,
##     subset = train)
##
## Prior probabilities of groups:
##         0         1
## 0.4571429 0.5428571
##
## Group means:
##   cylinders   weight displacement horsepower
## 0  6.812500 3604.823     271.7396  133.14583
## 1  4.070175 2314.763     111.6623   77.92105
```

```
pred.qda <- predict(fit.qda, Auto.test)
table(pred.qda$class, mpg01.test)
```

```
##    mpg01.test
##      0  1
##   0 89 13
##   1 11 69
```

```
mean(pred.qda$class != mpg01.test)
```

```
## [1] 0.1318681
```

The test error is 13.18681%.

f)

```
fit.glm <- glm(mpg01 ~ cylinders + weight + displacement + horsepower, data = Auto, family = binomial,
summary(fit.glm)
```

```
##
## Call:
```

```
## glm(formula = mpg01 ~ cylinders + weight + displacement + horsepower,
##     family = binomial, data = Auto, subset = train)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -2.48027  -0.03413   0.10583   0.29634   2.57584
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  17.658730   3.409012   5.180 2.22e-07 ***
## cylinders    -1.028032   0.653607  -1.573   0.1158
## weight       -0.002922   0.001137  -2.569   0.0102 *
## displacement  0.002462   0.015030   0.164   0.8699
## horsepower   -0.050611   0.025209  -2.008   0.0447 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 289.58  on 209  degrees of freedom
## Residual deviance:  83.24  on 205  degrees of freedom
## AIC: 93.24
##
## Number of Fisher Scoring iterations: 7
```

```
probs <- predict(fit.glm, Auto.test, type = "response")
pred.glm <- rep(0, length(probs))
pred.glm[probs > 0.5] <- 1
table(pred.glm, mpg01.test)
```

```
##         mpg01.test
## pred.glm  0  1
##        0 89 11
##        1 11 71
```

```
mean(pred.glm != mpg01.test)
```

```
## [1] 0.1208791
```

The test error is 12.08791%.

g)

```
train.X <- cbind(cylinders, weight, displacement, horsepower)[train, ]
test.X <- cbind(cylinders, weight, displacement, horsepower)[!train, ]
train.mpg01 <- mpg01[train]
set.seed(1)
pred.knn <- knn(train.X, test.X, train.mpg01, k = 1)
table(pred.knn, mpg01.test)
```

```
##         mpg01.test
## pred.knn  0  1
##        0 83 11
##        1 17 71
```

```
mean(pred.knn != mpg01.test)
```

```
## [1] 0.1538462
```

```
pred.knn <- knn(train.X, test.X, train.mpg01, k = 10)
table(pred.knn, mpg01.test)
```

```
##         mpg01.test
## pred.knn  0  1
##        0 77  7
##        1 23 75
```

```
mean(pred.knn != mpg01.test)
```

```
## [1] 0.1648352
```

```
pred.knn <- knn(train.X, test.X, train.mpg01, k = 100)
table(pred.knn, mpg01.test)
```

```
##         mpg01.test
## pred.knn  0  1
##        0 81  7
##        1 19 75
```

```
mean(pred.knn != mpg01.test)
```

```
## [1] 0.1428571
```

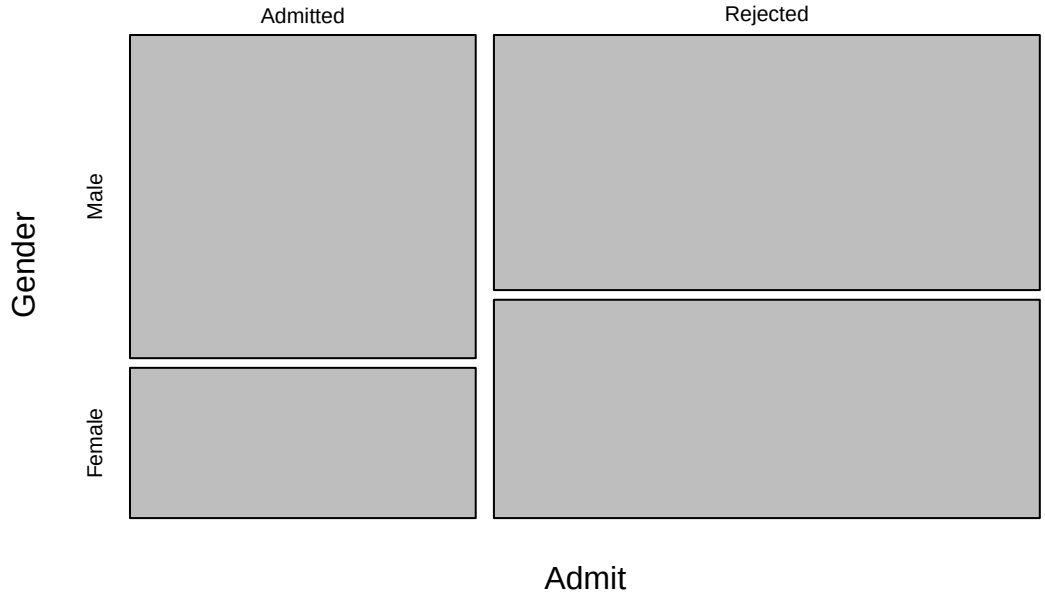
K=1: The test error is 15.38462%. K=10: The test error is 16.48352%. K=100: The test error is 14.28571%. Therefore, K=100 performs the best on this dataset.

Homework 3 - Question 6

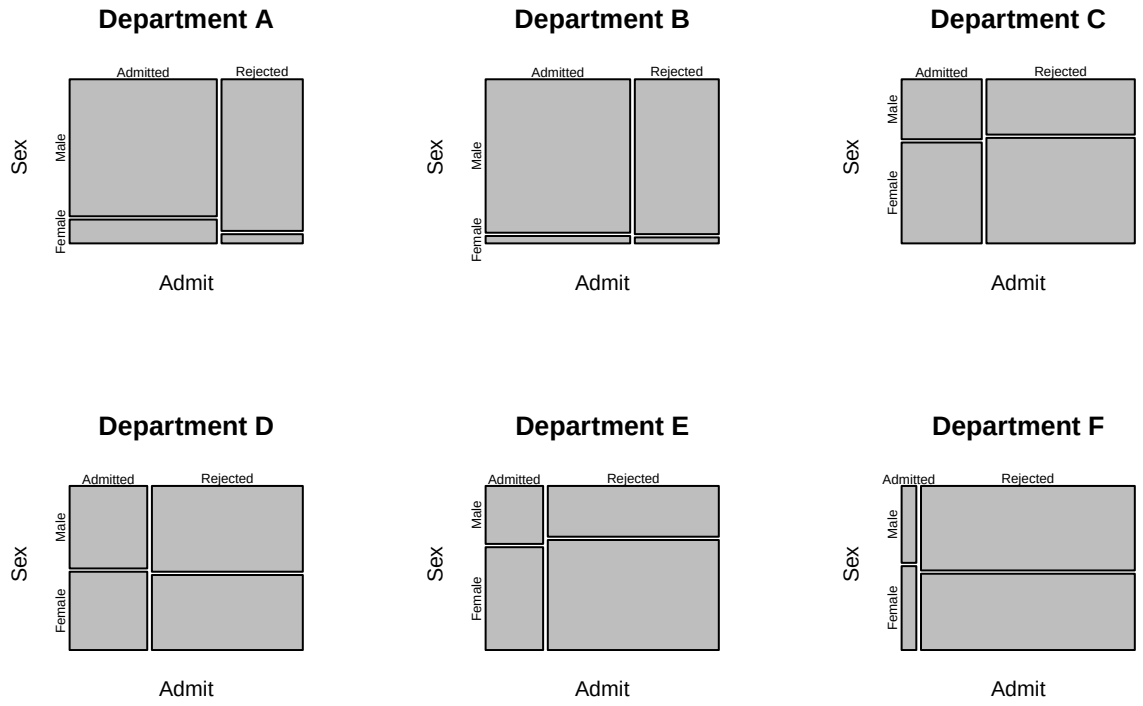
a)

```
example(UCBAdmissions)
```

```
##
## UCBAdm> require(graphics)
##
## UCBAdm> ## Data aggregated over departments
## UCBAdm> apply(UCBAdmissions, c(1, 2), sum)
##           Gender
## Admit      Male Female
##   Admitted 1198    557
##   Rejected 1493   1278
##
## UCBAdm> mosaicplot(apply(UCBAdmissions, c(1, 2), sum),
## UCBAdm+            main = "Student admissions at UC Berkeley")
```




```
##
## UCBAdm> ## Data for individual departments
## UCBAdm> opar <- par(mfrow = c(2, 3), oma = c(0, 0, 2, 0))
##
## UCBAdm> for(i in 1:6)
## UCBAdm+   mosaicplot(UCBAdmissions[,,i],
## UCBAdm+     xlab = "Admit", ylab = "Sex",
## UCBAdm+     main = paste("Department", LETTERS[i]))
```




```
##
## UCBAdm> mtext(expression(bold("Student admissions at UC Berkeley")),
## UCBAdm+       outer = TRUE, cex = 1.5)
##
## UCBAdm> par(opar)
```

The first plot implies males make up a significantly higher proportion of the admitted students compared to females. Meanwhile, males and females make up about the same proportion of rejected students (50/50 split). Therefore, yes, there seems to be a bias toward males being admitted more than females. The overall percentage of men that were accepted is 44.5188%, while the percentage for women is 30.3542%.

b) These 6 plots show that none of the departments are inherently biased. We can see this by comparing the admitted box splits to the rejected box splits. If there is a serious imbalance when comparing admitted to rejected, then there is some bias, as seen in part a). This is not the case for these 6 plots.

c) The general idea of the paradox is that a specific trend appears when a bunch of data is aggregated (such as in the initial plot), but then completely disappears when the data is split into their respective subgroups (such as in the subsequent 6 plots).

d) We can probably explain this away by claiming females applied to extremely competitive departments that already have low acceptance rates, while men typically applied to less competitive departments with higher acceptance rates. So, more men ended up getting into their desired departments anyway, thus increasing the number of overall men accepted compared to females.

e)

```
data(UCBAdmissions)
Adm <- as.integer(UCBAdmissions)[(1:(6*2))*2-1]
Rej <- as.integer(UCBAdmissions)[(1:(6*2))*2]
```

```
Dept <- gl(6,2,6*2,labels=c("A","B","C","D","E","F"))
Sex <- gl(2,1,6*2,labels=c("Male","Female"))
Ratio <- Adm/(Rej+Adm)

berk <- data.frame(Adm,Rej,Sex,Dept,Ratio)

head(berk)
```

```
##   Adm Rej    Sex Dept     Ratio
## 1 512 313   Male    A 0.6206061
## 2  89  19 Female    A 0.8240741
## 3 353 207   Male    B 0.6303571
## 4  17   8 Female    B 0.6800000
## 5 120 205   Male    C 0.3692308
## 6 202 391 Female    C 0.3406408
```

```
LogReg.gender <- glm(cbind(Adm,Rej)~Sex,data=berk,family=binomial("logit"))
summary(LogReg.gender)
```

```
##
## Call:
## glm(formula = cbind(Adm, Rej) ~ Sex, family = binomial("logit"),
##     data = berk)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -16.7915   -4.7613   -0.4365    5.1025   11.2022
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.22013    0.03879  -5.675 1.38e-08 ***
## SexFemale   -0.61035    0.06389  -9.553  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 877.06  on 11  degrees of freedom
## Residual deviance: 783.61  on 10  degrees of freedom
## AIC: 856.55
##
## Number of Fisher Scoring iterations: 4
```

We can claim the admission rate of females is statistically significant with p-value $<$ 2e-16. So, there is a bias against females.

f)

```
LogReg.gender <- glm(cbind(Adm,Rej)~Sex+Dept,data=berk,family=binomial("logit"))
summary(LogReg.gender)
```

```
##
## Call:
## glm(formula = cbind(Adm, Rej) ~ Sex + Dept, family = binomial("logit"),
##     data = berk)
##
```

```
## Deviance Residuals:
##       1        2        3        4        5        6        7        8
## -1.2487   3.7189  -0.0560   0.2706   1.2533  -0.9243   0.0826  -0.0858
##       9       10       11       12
##  1.2205  -0.8509  -0.2076   0.2052
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.58205    0.06899   8.436   <2e-16 ***
## SexFemale    0.09987    0.08085   1.235    0.217
## DeptB       -0.04340    0.10984  -0.395    0.693
## DeptC       -1.26260    0.10663 -11.841   <2e-16 ***
## DeptD       -1.29461    0.10582 -12.234   <2e-16 ***
## DeptE       -1.73931    0.12611 -13.792   <2e-16 ***
## DeptF       -3.30648    0.16998 -19.452   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 877.056  on 11  degrees of freedom
## Residual deviance:  20.204  on  5  degrees of freedom
## AIC: 103.14
##
## Number of Fisher Scoring iterations: 4
```

After taking departments into consideration, we can see the female acceptance rate is no longer statistically significant (p-value of 0.217), so there is no bias against females. The coefficient for females drops from -0.61035 to 0.09987, so it is closer to 0 compared to part e), making it less of a factor.

In this problem, we have shown the Simpson paradox, which is the idea of aggregated data showing specific trends, but subgroups of the data not showing those trends at all. Thus, the data may seem biased, but in fact is actually balanced and further inference of the data can lead to a higher-level reasoning about this paradox. One can uncover these biases by either plotting the subgroups, as we did in parts a) and b), or by fitting a model to the data, such as logistic regression, and doing hypothesis testing on the predictors to see if they are statistically significant, indicating a significant trend in the data.

Bonus)

```r
LogReg.gender <- glm(cbind(Adm,Rej) ~ Sex + Dept + Sex*Dept,data=berk,family=binomial("logit"))
summary(LogReg.gender)
```

```
##
## Call:
## glm(formula = cbind(Adm, Rej) ~ Sex + Dept + Sex * Dept, family = binomial("logit"),
##     data = berk)
##
## Deviance Residuals:
##  [1]  0  0  0  0  0  0  0  0  0  0  0  0
##
## Coefficients:
##                 Estimate Std. Error z value Pr(>|z|)
## (Intercept)      0.49212    0.07175   6.859 6.94e-12 ***
## SexFemale        1.05208    0.26271   4.005 6.21e-05 ***
## DeptB            0.04163    0.11319   0.368  0.71304
## DeptC           -1.02764    0.13550  -7.584 3.34e-14 ***
```

```
## DeptD           -1.19608    0.12641  -9.462  < 2e-16 ***
## DeptE           -1.44908    0.17681  -8.196 2.49e-16 ***
## DeptF           -3.26187    0.23120 -14.109  < 2e-16 ***
## SexFemale:DeptB -0.83205    0.51039  -1.630  0.10306
## SexFemale:DeptC -1.17700    0.29956  -3.929 8.53e-05 ***
## SexFemale:DeptD -0.97009    0.30262  -3.206  0.00135 **
## SexFemale:DeptE -1.25226    0.33032  -3.791  0.00015 ***
## SexFemale:DeptF -0.86318    0.40267  -2.144  0.03206 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8.7706e+02  on 11  degrees of freedom
## Residual deviance: 1.1591e-13  on  0  degrees of freedom
## AIC: 92.94
##
## Number of Fisher Scoring iterations: 3
```

This model is definitely different compared to the previous two models because Females in Department C are suddently statistically significant. Similar to the previous model, departments C-F are statistically significant, while Department B is not significant. In this model, we have gone back to female admission rates once again being statistically significant. As such, including the gender/department interaction term makes some of the previously insignificant interactions being statistically significant. For females in department C, there seems to be a slight bias against females since the p-value is statistically significant (8.53e-05) and the coefficient is negative. This is hardly observed in the previous plots, but the p-value isn't as low as the other p-values (8.53e-05 compared to the results above with p-values of 3.34e-14, 2e-16, etc.), so the bias isn't as obvious. I would say this agrees with our data in the plots from part (b) since none of the departments' p-values are statistically significant when interacted with gender, except for departments C, D, and E, which agrees with the slight skew of the charts in (b).

Homework 3 - Question 7

```
library(DMwR)
```

```
## Loading required package: lattice
```

```
## Loading required package: grid
```

```
library(class)
library(MASS)
library(stats)

newsDataOriginal <- read.table("OnlineNewsPopularity.csv", header=TRUE, sep=",")
newsDataOriginal$shares = as.numeric(newsDataOriginal$shares)
names(newsDataOriginal)
```

```
##  [1] "url"                          "timedelta"
##  [3] "n_tokens_title"               "n_tokens_content"
##  [5] "n_unique_tokens"              "n_non_stop_words"
##  [7] "n_non_stop_unique_tokens"     "num_hrefs"
##  [9] "num_self_hrefs"               "num_imgs"
## [11] "num_videos"                   "average_token_length"
## [13] "num_keywords"                 "data_channel_is_lifestyle"
## [15] "data_channel_is_entertainment" "data_channel_is_bus"
```

```
## [17] "data_channel_is_socmed"         "data_channel_is_tech"
## [19] "data_channel_is_world"          "kw_min_min"
## [21] "kw_max_min"                     "kw_avg_min"
## [23] "kw_min_max"                     "kw_max_max"
## [25] "kw_avg_max"                     "kw_min_avg"
## [27] "kw_max_avg"                     "kw_avg_avg"
## [29] "self_reference_min_shares"      "self_reference_max_shares"
## [31] "self_reference_avg_sharess"     "weekday_is_monday"
## [33] "weekday_is_tuesday"             "weekday_is_wednesday"
## [35] "weekday_is_thursday"            "weekday_is_friday"
## [37] "weekday_is_saturday"            "weekday_is_sunday"
## [39] "is_weekend"                     "LDA_00"
## [41] "LDA_01"                         "LDA_02"
## [43] "LDA_03"                         "LDA_04"
## [45] "global_subjectivity"            "global_sentiment_polarity"
## [47] "global_rate_positive_words"     "global_rate_negative_words"
## [49] "rate_positive_words"            "rate_negative_words"
## [51] "avg_positive_polarity"          "min_positive_polarity"
## [53] "max_positive_polarity"          "avg_negative_polarity"
## [55] "min_negative_polarity"          "max_negative_polarity"
## [57] "title_subjectivity"             "title_sentiment_polarity"
## [59] "abs_title_subjectivity"         "abs_title_sentiment_polarity"
## [61] "shares"
```

```
summary(newsDataOriginal)
```

```
##                                                      url
##  http://mashable.com/2013/01/07/amazon-instant-video-browser/  :    1
##  http://mashable.com/2013/01/07/ap-samsung-sponsored-tweets/   :    1
##  http://mashable.com/2013/01/07/apple-40-billion-app-downloads/:    1
##  http://mashable.com/2013/01/07/astronaut-notre-dame-bcs/      :    1
##  http://mashable.com/2013/01/07/att-u-verse-apps/              :    1
##  http://mashable.com/2013/01/07/beewi-smart-toys/              :    1
##  (Other)                                                       :39638
##    timedelta     n_tokens_title n_tokens_content n_unique_tokens
##  Min.   :  8.0   Min.   : 2.0   Min.   :   0.0   Min.   :  0.0000
##  1st Qu.:164.0   1st Qu.: 9.0   1st Qu.: 246.0   1st Qu.:  0.4709
##  Median :339.0   Median :10.0   Median : 409.0   Median :  0.5392
##  Mean   :354.5   Mean   :10.4   Mean   : 546.5   Mean   :  0.5482
##  3rd Qu.:542.0   3rd Qu.:12.0   3rd Qu.: 716.0   3rd Qu.:  0.6087
##  Max.   :731.0   Max.   :23.0   Max.   :8474.0   Max.   :701.0000
##
##  n_non_stop_words    n_non_stop_unique_tokens   num_hrefs
##  Min.   :   0.0000   Min.   :  0.0000         Min.   :  0.00
##  1st Qu.:   1.0000   1st Qu.:  0.6257         1st Qu.:  4.00
##  Median :   1.0000   Median :  0.6905         Median :  8.00
##  Mean   :   0.9965   Mean   :  0.6892         Mean   : 10.88
##  3rd Qu.:   1.0000   3rd Qu.:  0.7546         3rd Qu.: 14.00
##  Max.   :1042.0000   Max.   :650.0000         Max.   :304.00
##
##  num_self_hrefs       num_imgs         num_videos    average_token_length
##  Min.   :  0.000   Min.   :  0.000   Min.   : 0.00   Min.   :0.000
##  1st Qu.:  1.000   1st Qu.:  1.000   1st Qu.: 0.00   1st Qu.:4.478
##  Median :  3.000   Median :  1.000   Median : 0.00   Median :4.664
##  Mean   :  3.294   Mean   :  4.544   Mean   : 1.25   Mean   :4.548
```

```
##  3rd Qu.:  4.000   3rd Qu.:  4.000   3rd Qu.: 1.00   3rd Qu.:4.855
##  Max.   :116.000   Max.   :128.000   Max.   :91.00   Max.   :8.042
##
##   num_keywords    data_channel_is_lifestyle data_channel_is_entertainment
##  Min.   : 1.000   Min.   :0.00000           Min.   :0.000
##  1st Qu.: 6.000   1st Qu.:0.00000           1st Qu.:0.000
##  Median : 7.000   Median :0.00000           Median :0.000
##  Mean   : 7.224   Mean   :0.05295           Mean   :0.178
##  3rd Qu.: 9.000   3rd Qu.:0.00000           3rd Qu.:0.000
##  Max.   :10.000   Max.   :1.00000           Max.   :1.000
##
##  data_channel_is_bus data_channel_is_socmed data_channel_is_tech
##  Min.   :0.0000      Min.   :0.0000         Min.   :0.0000
##  1st Qu.:0.0000      1st Qu.:0.0000         1st Qu.:0.0000
##  Median :0.0000      Median :0.0000         Median :0.0000
##  Mean   :0.1579      Mean   :0.0586         Mean   :0.1853
##  3rd Qu.:0.0000      3rd Qu.:0.0000         3rd Qu.:0.0000
##  Max.   :1.0000      Max.   :1.0000         Max.   :1.0000
##
##  data_channel_is_world   kw_min_min       kw_max_min       kw_avg_min
##  Min.   :0.0000        Min.   : -1.00   Min.   :     0   Min.   :   -1.0
##  1st Qu.:0.0000        1st Qu.: -1.00   1st Qu.:   445   1st Qu.:  141.8
##  Median :0.0000        Median : -1.00   Median :   660   Median :  235.5
##  Mean   :0.2126        Mean   : 26.11   Mean   :  1154   Mean   :  312.4
##  3rd Qu.:0.0000        3rd Qu.:  4.00   3rd Qu.:  1000   3rd Qu.:  357.0
##  Max.   :1.0000        Max.   :377.00   Max.   :298400   Max.   :42827.9
##
##    kw_min_max       kw_max_max       kw_avg_max       kw_min_avg
##  Min.   :     0   Min.   :     0   Min.   :     0   Min.   : -1
##  1st Qu.:     0   1st Qu.:843300   1st Qu.:172847   1st Qu.:   0
##  Median :  1400   Median :843300   Median :244572   Median :1024
##  Mean   : 13612   Mean   :752324   Mean   :259282   Mean   :1117
##  3rd Qu.:  7900   3rd Qu.:843300   3rd Qu.:330980   3rd Qu.:2057
##  Max.   :843300   Max.   :843300   Max.   :843300   Max.   :3613
##
##    kw_max_avg       kw_avg_avg    self_reference_min_shares
##  Min.   :     0   Min.   :    0   Min.   :     0
##  1st Qu.:  3562   1st Qu.: 2382   1st Qu.:   639
##  Median :  4356   Median : 2870   Median :  1200
##  Mean   :  5657   Mean   : 3136   Mean   :  3999
##  3rd Qu.:  6020   3rd Qu.: 3600   3rd Qu.:  2600
##  Max.   :298400   Max.   :43568   Max.   :843300
##
##  self_reference_max_shares self_reference_avg_sharess weekday_is_monday
##  Min.   :     0            Min.   :     0.0           Min.   :0.000
##  1st Qu.:  1100            1st Qu.:   981.2           1st Qu.:0.000
##  Median :  2800            Median :  2200.0           Median :0.000
##  Mean   : 10329            Mean   :  6401.7           Mean   :0.168
##  3rd Qu.:  8000            3rd Qu.:  5200.0           3rd Qu.:0.000
##  Max.   :843300            Max.   :843300.0           Max.   :1.000
##
##  weekday_is_tuesday weekday_is_wednesday weekday_is_thursday
##  Min.   :0.0000     Min.   :0.0000       Min.   :0.0000
##  1st Qu.:0.0000     1st Qu.:0.0000       1st Qu.:0.0000
```

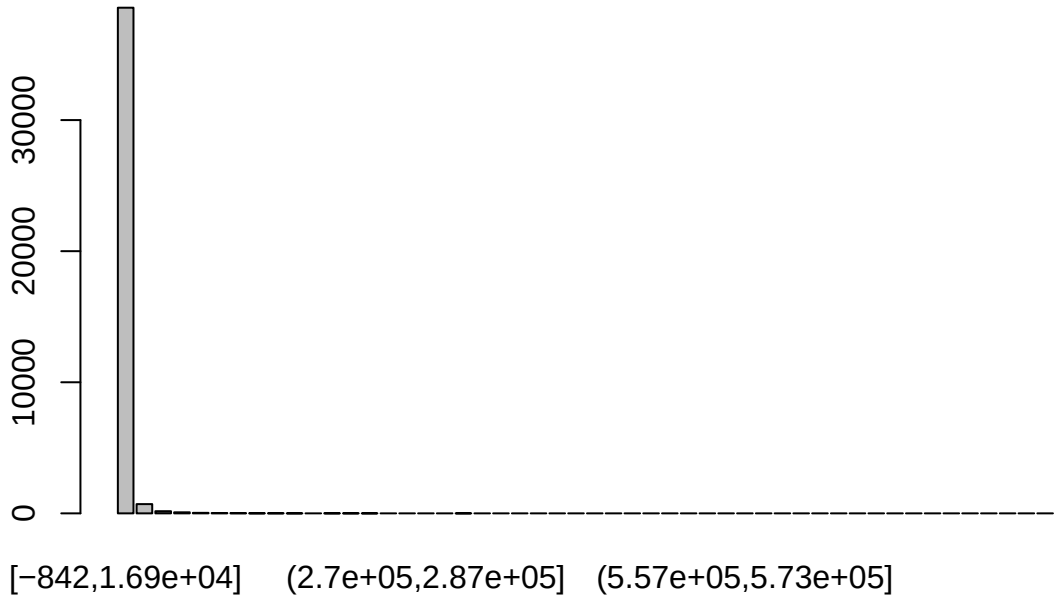
```
##  Median :0.0000     Median :0.0000       Median :0.0000
##  Mean   :0.1864     Mean   :0.1875       Mean   :0.1833
##  3rd Qu.:0.0000     3rd Qu.:0.0000       3rd Qu.:0.0000
##  Max.   :1.0000     Max.   :1.0000       Max.   :1.0000
##
##  weekday_is_friday weekday_is_saturday weekday_is_sunday   is_weekend
##  Min.   :0.0000    Min.   :0.00000     Min.   :0.00000   Min.   :0.0000
##  1st Qu.:0.0000    1st Qu.:0.00000     1st Qu.:0.00000   1st Qu.:0.0000
##  Median :0.0000    Median :0.00000     Median :0.00000   Median :0.0000
##  Mean   :0.1438    Mean   :0.06188     Mean   :0.06904   Mean   :0.1309
##  3rd Qu.:0.0000    3rd Qu.:0.00000     3rd Qu.:0.00000   3rd Qu.:0.0000
##  Max.   :1.0000    Max.   :1.00000     Max.   :1.00000   Max.   :1.0000
##
##      LDA_00            LDA_01            LDA_02            LDA_03
##  Min.   :0.00000   Min.   :0.00000   Min.   :0.00000   Min.   :0.00000
##  1st Qu.:0.02505   1st Qu.:0.02501   1st Qu.:0.02857   1st Qu.:0.02857
##  Median :0.03339   Median :0.03334   Median :0.04000   Median :0.04000
##  Mean   :0.18460   Mean   :0.14126   Mean   :0.21632   Mean   :0.22377
##  3rd Qu.:0.24096   3rd Qu.:0.15083   3rd Qu.:0.33422   3rd Qu.:0.37576
##  Max.   :0.92699   Max.   :0.92595   Max.   :0.92000   Max.   :0.92653
##
##      LDA_04        global_subjectivity global_sentiment_polarity
##  Min.   :0.00000   Min.   :0.0000      Min.   :-0.39375
##  1st Qu.:0.02857   1st Qu.:0.3962      1st Qu.: 0.05776
##  Median :0.04073   Median :0.4535      Median : 0.11912
##  Mean   :0.23403   Mean   :0.4434      Mean   : 0.11931
##  3rd Qu.:0.39999   3rd Qu.:0.5083      3rd Qu.: 0.17783
##  Max.   :0.92719   Max.   :1.0000      Max.   : 0.72784
##
##  global_rate_positive_words global_rate_negative_words rate_positive_words
##  Min.   :0.00000            Min.   :0.000000           Min.   :0.0000
##  1st Qu.:0.02838            1st Qu.:0.009615           1st Qu.:0.6000
##  Median :0.03902            Median :0.015337           Median :0.7105
##  Mean   :0.03962            Mean   :0.016612           Mean   :0.6822
##  3rd Qu.:0.05028            3rd Qu.:0.021739           3rd Qu.:0.8000
##  Max.   :0.15549            Max.   :0.184932           Max.   :1.0000
##
##  rate_negative_words avg_positive_polarity min_positive_polarity
##  Min.   :0.0000      Min.   :0.0000        Min.   :0.00000
##  1st Qu.:0.1852      1st Qu.:0.3062        1st Qu.:0.05000
##  Median :0.2800      Median :0.3588        Median :0.10000
##  Mean   :0.2879      Mean   :0.3538        Mean   :0.09545
##  3rd Qu.:0.3846      3rd Qu.:0.4114        3rd Qu.:0.10000
##  Max.   :1.0000      Max.   :1.0000        Max.   :1.00000
##
##  max_positive_polarity avg_negative_polarity min_negative_polarity
##  Min.   :0.0000        Min.   :-1.0000       Min.   :-1.0000
##  1st Qu.:0.6000        1st Qu.:-0.3284       1st Qu.:-0.7000
##  Median :0.8000        Median :-0.2533       Median :-0.5000
##  Mean   :0.7567        Mean   :-0.2595       Mean   :-0.5219
##  3rd Qu.:1.0000        3rd Qu.:-0.1869       3rd Qu.:-0.3000
##  Max.   :1.0000        Max.   : 0.0000       Max.   : 0.0000
##
##  max_negative_polarity title_subjectivity title_sentiment_polarity
```

```
##  Min.   :-1.0000       Min.   :0.0000     Min.   :-1.00000
##  1st Qu.:-0.1250       1st Qu.:0.0000     1st Qu.: 0.00000
##  Median :-0.1000       Median :0.1500     Median : 0.00000
##  Mean   :-0.1075       Mean   :0.2824     Mean   : 0.07143
##  3rd Qu.:-0.0500       3rd Qu.:0.5000     3rd Qu.: 0.15000
##  Max.   : 0.0000       Max.   :1.0000     Max.   : 1.00000
##
##  abs_title_subjectivity abs_title_sentiment_polarity     shares
##  Min.   :0.0000         Min.   :0.0000               Min.   :     1
##  1st Qu.:0.1667         1st Qu.:0.0000               1st Qu.:   946
##  Median :0.5000         Median :0.0000               Median :  1400
##  Mean   :0.3418         Mean   :0.1561               Mean   :  3395
##  3rd Qu.:0.5000         3rd Qu.:0.2500               3rd Qu.:  2800
##  Max.   :0.5000         Max.   :1.0000               Max.   :843300
##
```

```
summary(newsDataOriginal$shares)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1     946    1400    3395    2800  843300
```

```
newsDataLen <- nrow(newsDataOriginal)
shares_bins <- cut(newsDataOriginal$shares, 50, include.lowest=TRUE)
plot(shares_bins)
```

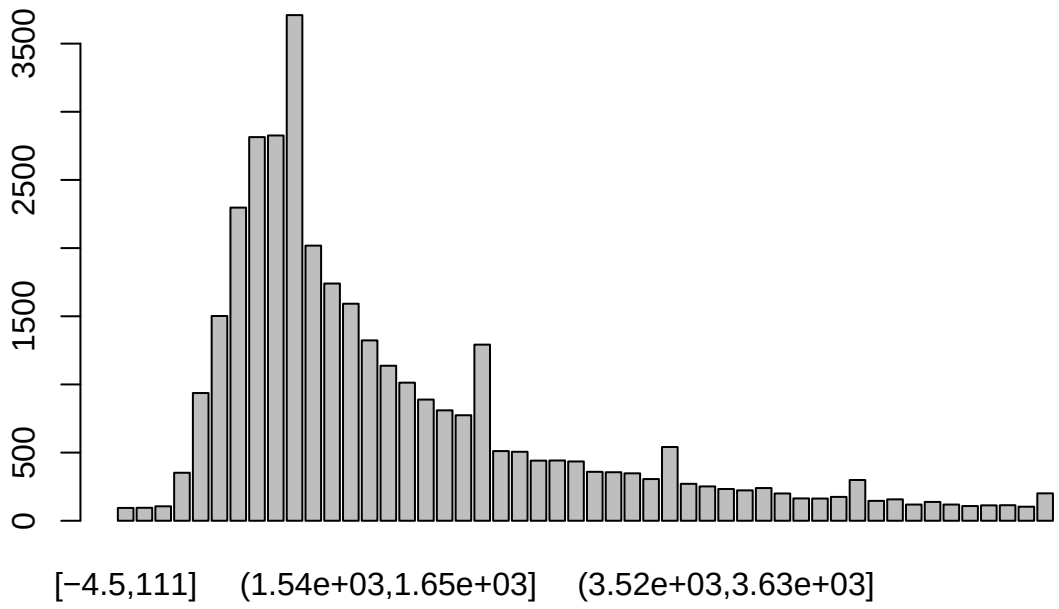



```
sharesIqr <- IQR(newsDataOriginal$shares)
shares75Quant <- quantile(newsDataOriginal$shares, 0.75)
shares25Quant <- quantile(newsDataOriginal$shares, 0.25)
```

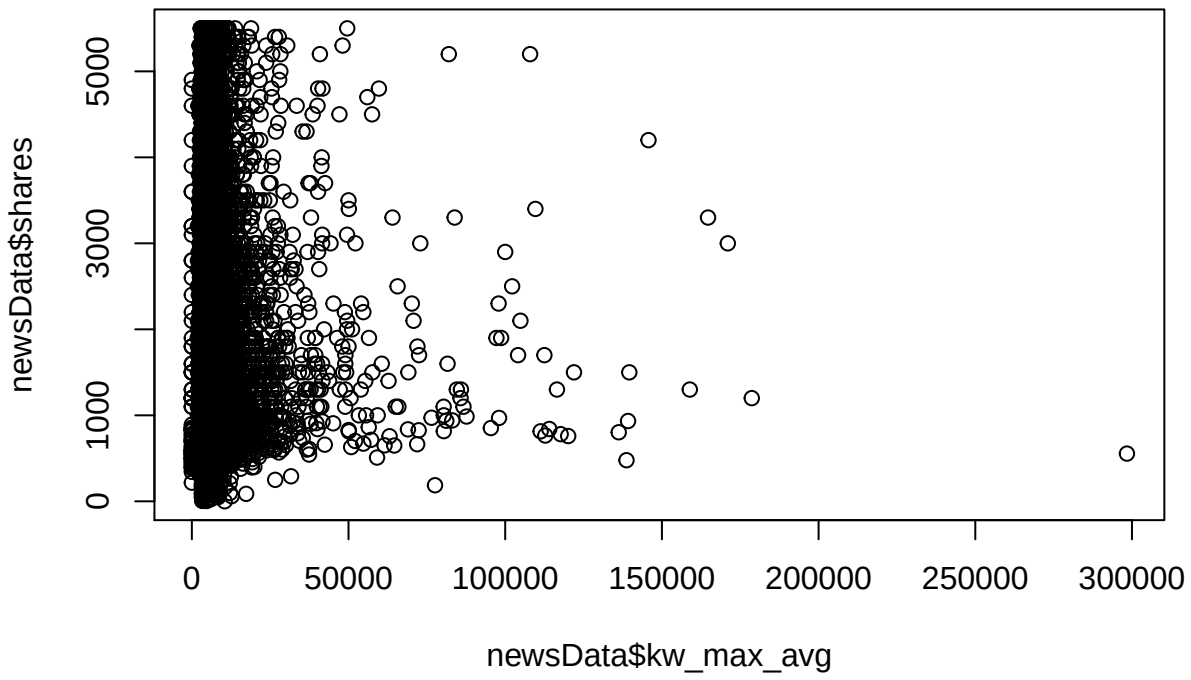
```r
newsData <- newsDataOriginal[newsDataOriginal$shares < (1.5*sharesIqr + shares75Quant) & newsDataOrigina
summary(newsData$shares)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       1     903    1300    1672    2100    5500
```

```r
numOutliers <- (newsDataLen - nrow(newsData))

shares_bins <- cut(newsData$shares, 50, include.lowest=TRUE)
plot(shares_bins)
```




```r
plot(newsData$kw_max_avg, newsData$shares)
```




```
newsDataQuant <- newsData[, sapply(newsData, class) == "numeric"]
names(newsDataQuant)
```

```
##  [1] "timedelta"                    "n_tokens_title"
##  [3] "n_tokens_content"             "n_unique_tokens"
##  [5] "n_non_stop_words"             "n_non_stop_unique_tokens"
##  [7] "num_hrefs"                    "num_self_hrefs"
##  [9] "num_imgs"                     "num_videos"
## [11] "average_token_length"         "num_keywords"
## [13] "data_channel_is_lifestyle"    "data_channel_is_entertainment"
## [15] "data_channel_is_bus"          "data_channel_is_socmed"
## [17] "data_channel_is_tech"         "data_channel_is_world"
## [19] "kw_min_min"                   "kw_max_min"
## [21] "kw_avg_min"                   "kw_min_max"
## [23] "kw_max_max"                   "kw_avg_max"
## [25] "kw_min_avg"                   "kw_max_avg"
## [27] "kw_avg_avg"                   "self_reference_min_shares"
## [29] "self_reference_max_shares"    "self_reference_avg_sharess"
## [31] "weekday_is_monday"            "weekday_is_tuesday"
## [33] "weekday_is_wednesday"         "weekday_is_thursday"
## [35] "weekday_is_friday"            "weekday_is_saturday"
## [37] "weekday_is_sunday"            "is_weekend"
## [39] "LDA_00"                       "LDA_01"
## [41] "LDA_02"                       "LDA_03"
## [43] "LDA_04"                       "global_subjectivity"
## [45] "global_sentiment_polarity"    "global_rate_positive_words"
```

```
## [47] "global_rate_negative_words"    "rate_positive_words"
## [49] "rate_negative_words"           "avg_positive_polarity"
## [51] "min_positive_polarity"         "max_positive_polarity"
## [53] "avg_negative_polarity"         "min_negative_polarity"
## [55] "max_negative_polarity"         "title_subjectivity"
## [57] "title_sentiment_polarity"      "abs_title_subjectivity"
## [59] "abs_title_sentiment_polarity"  "shares"
```

```
cor(as.matrix(newsData[, 61]), as.matrix(newsData[,-1])) # correlations with 'shares' and every other v
```

```
##       timedelta n_tokens_title n_tokens_content n_unique_tokens
## [1,] 0.03657173    -0.04204983       0.04782074     -0.04909971
##      n_non_stop_words n_non_stop_unique_tokens num_hrefs num_self_hrefs
## [1,]      -0.01318755              -0.05080723 0.0776524     0.04316208
##        num_imgs   num_videos average_token_length num_keywords
## [1,] 0.05592683 -0.002898373          -0.02555183   0.06553517
##      data_channel_is_lifestyle data_channel_is_entertainment
## [1,]                0.03143692                    -0.1054218
##      data_channel_is_bus data_channel_is_socmed data_channel_is_tech
## [1,]         0.001639743              0.1149444           0.09737915
##      data_channel_is_world kw_min_min kw_max_min kw_avg_min  kw_min_max
## [1,]             -0.137431 0.03989283 0.02247175 0.03162921 0.007840949
##       kw_max_max kw_avg_max kw_min_avg kw_max_avg kw_avg_avg
## [1,] -0.02491639 0.01602475 0.08951021 0.06315745  0.1476776
##      self_reference_min_shares self_reference_max_shares
## [1,]                0.04458771                0.05480541
##      self_reference_avg_sharess weekday_is_monday weekday_is_tuesday
## [1,]                 0.05719484       -0.02269312        -0.03918078
##      weekday_is_wednesday weekday_is_thursday weekday_is_friday
## [1,]          -0.04175593         -0.02504239       0.009257667
##      weekday_is_saturday weekday_is_sunday is_weekend     LDA_00
## [1,]            0.101764        0.08975654  0.1399974 0.07562637
##           LDA_01     LDA_02     LDA_03     LDA_04 global_subjectivity
## [1,] -0.07674991 -0.1366928 0.03927561 0.08673353          0.05829045
##      global_sentiment_polarity global_rate_positive_words
## [1,]                0.06326113                 0.06326329
##      global_rate_negative_words rate_positive_words rate_negative_words
## [1,]                -0.02546071          0.04474235         -0.06757323
##      avg_positive_polarity min_positive_polarity max_positive_polarity
## [1,]             0.0190065           -0.03203813           0.03322045
##      avg_negative_polarity min_negative_polarity max_negative_polarity
## [1,]          -0.003511512          -0.004899245           0.003100281
##      title_subjectivity title_sentiment_polarity abs_title_subjectivity
## [1,]         0.02585881                0.0452892            0.004831034
##      abs_title_sentiment_polarity shares
## [1,]                   0.02951515      1
```

Refer to above code. The above code does a lot of work. I have done some preprocessing on the data. For example, I plotted the original news data with 20 histogram bins and immediately realized the distribution was significantly skewed to the right. I concluded there were definitely outliers in the data, so I went into further analysis. I did a summary of the shares data, which is the target/predicted label, and saw the first quartile was at 946 shares, third quartile was at 2800 shares, and then the min and max were 1 and 843,300 respectively. Therefore, with an IQR of 1854, I calculated outliers as being outside the range (946 - IQR*1.5, 2800 + IQR*1.5). There were 4541 outliers in the data, taking the dataset from 39644 samples to 35103 samples. This had an immediate impact on the calculation of correlations. Although not depicted in the

code above, I did analysis before removing the outliers and the correlations between shares and all other features were in the range [-0.07, +0.08]. As such, there were no strong correlations. After removing the outliers, the range increased to [-0.137, +0.148] with the strongest positive and negative relationships being with data_channel_is_entertainment (-0.105), data_channel_is_socmed (0.115), data_channel_is_world (-0.137), kw_avg_avg (0.148), weekday_is_saturday (0.102), is_weekend (0.140), and LDA_02 (-0.137).

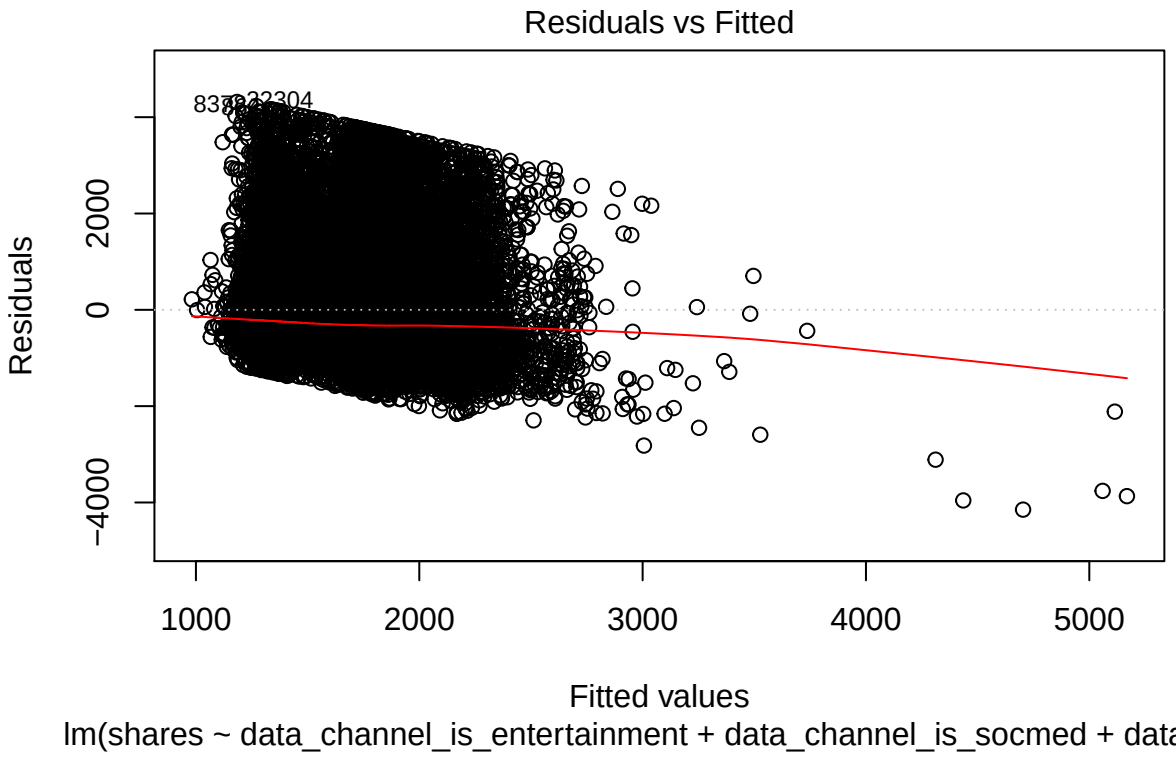
```
newsLm <- lm(shares ~ data_channel_is_entertainment + data_channel_is_socmed + data_channel_is_world + k
summary(newsLm)
```

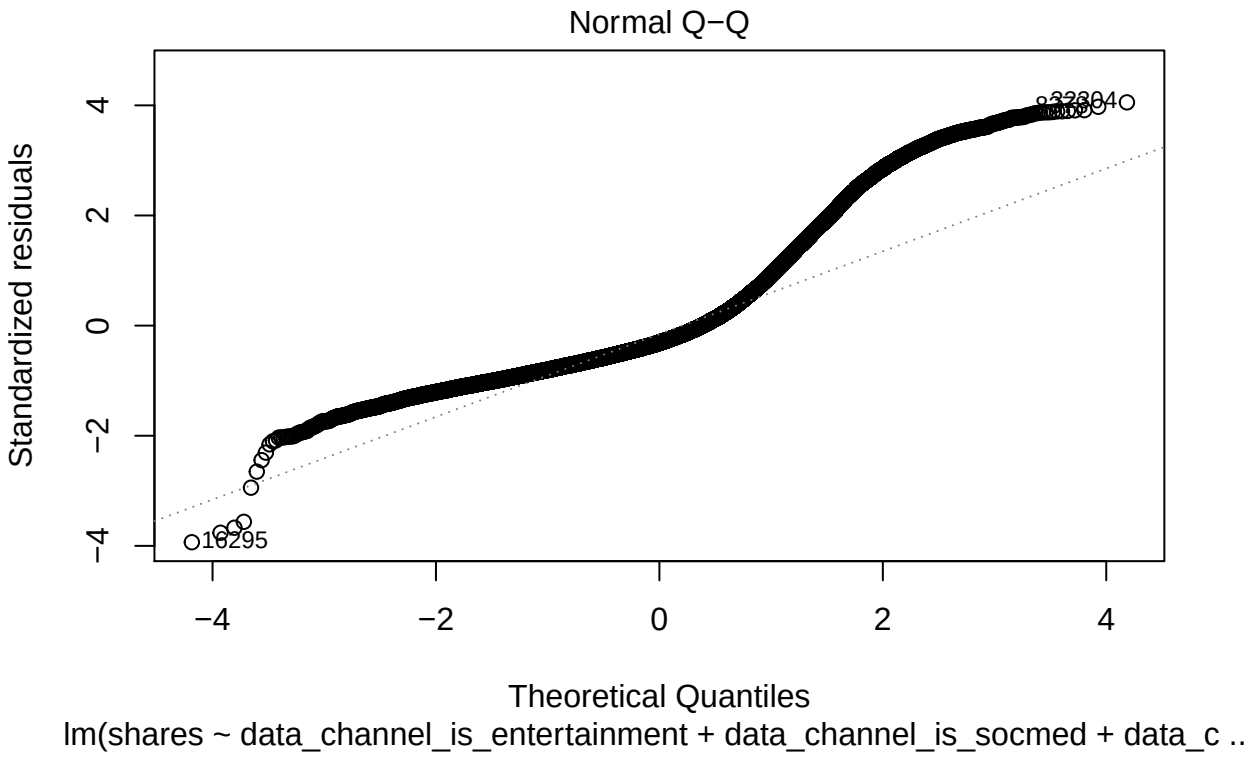
```
##
## Call:
## lm(formula = shares ~ data_channel_is_entertainment + data_channel_is_socmed +
##     data_channel_is_world + kw_avg_avg + LDA_02 + weekday_is_saturday:is_weekend,
##     data = newsData)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -4148.4  -702.7  -330.6   376.7  4315.9
##
## Coefficients:
##                                  Estimate Std. Error t value Pr(>|t|)
## (Intercept)                     1.503e+03  1.814e+01  82.869  < 2e-16 ***
## data_channel_is_entertainment  -3.735e+02  1.546e+01 -24.149  < 2e-16 ***
## data_channel_is_socmed          4.064e+02  2.547e+01  15.958  < 2e-16 ***
## data_channel_is_world          -1.774e+02  2.614e+01  -6.786 1.17e-11 ***
## kw_avg_avg                      9.634e-02  4.814e-03  20.012  < 2e-16 ***
## LDA_02                         -2.996e+02  3.726e+01  -8.043 9.06e-16 ***
## weekday_is_saturday:is_weekend  4.382e+02  2.392e+01  18.321  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1065 on 35096 degrees of freedom
## Multiple R-squared:  0.06892,    Adjusted R-squared:  0.06876
## F-statistic:   433 on 6 and 35096 DF,  p-value: < 2.2e-16
```

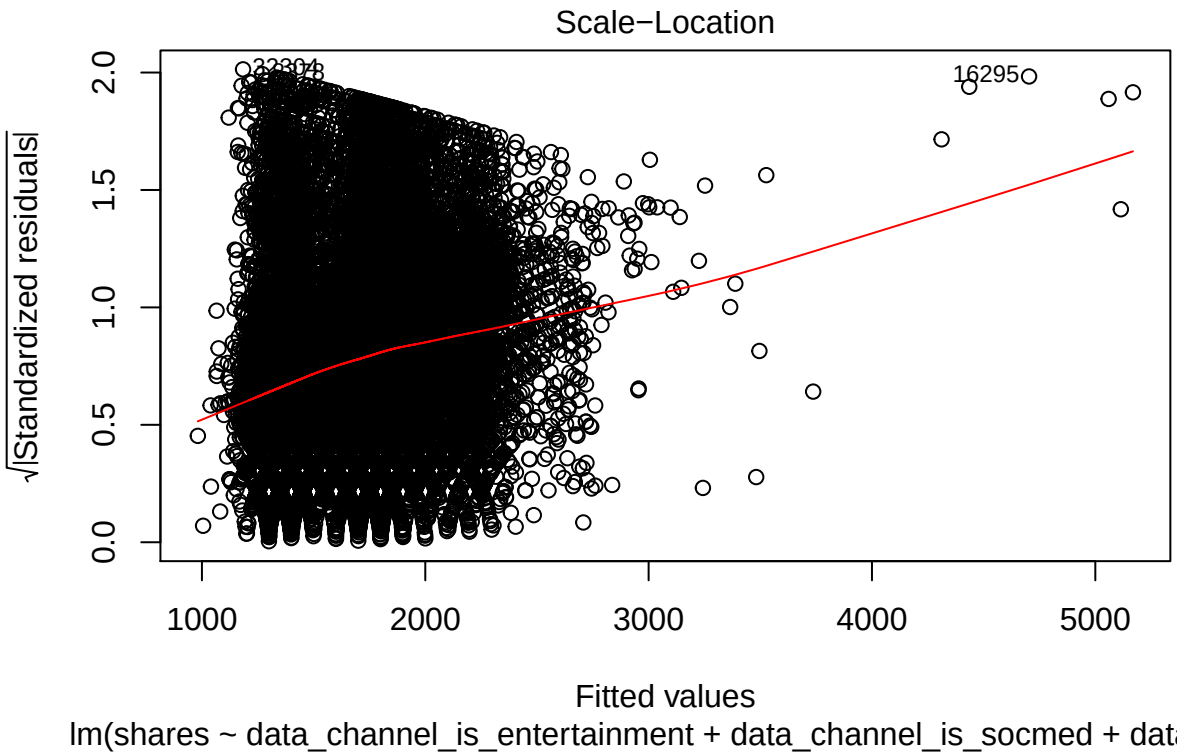
```
confint(newsLm)
```

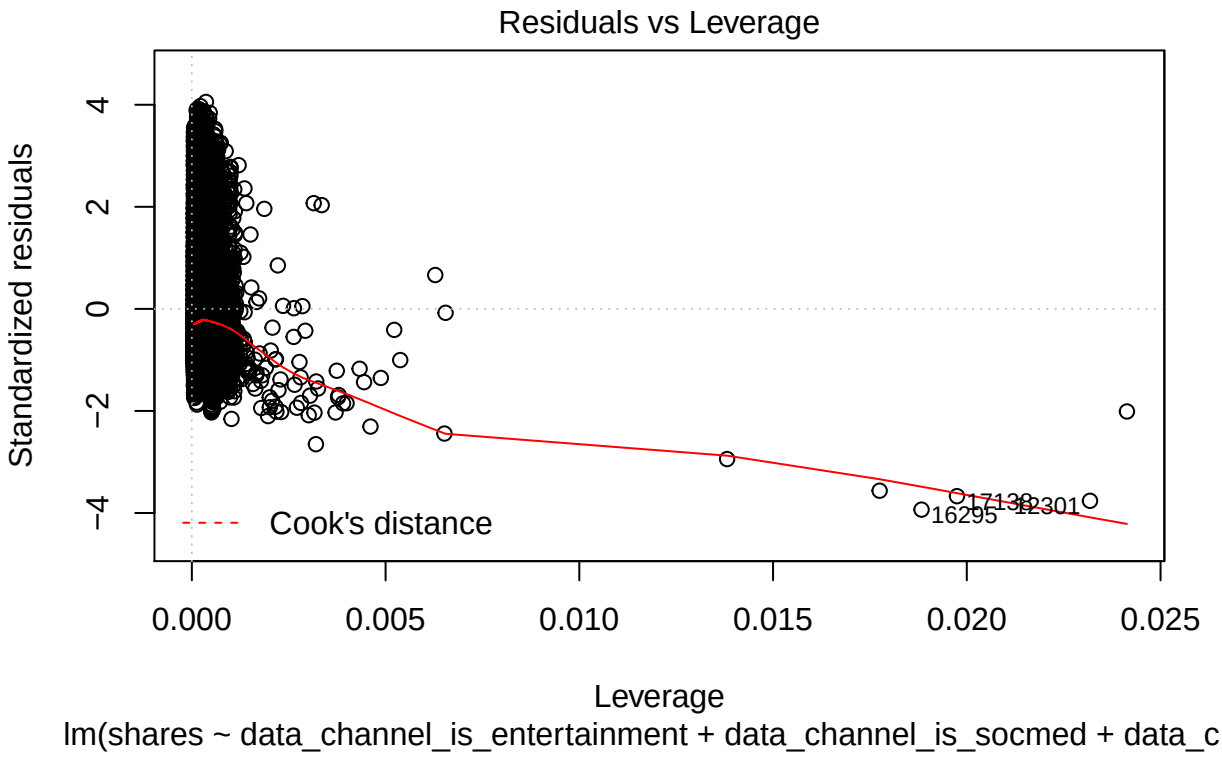
```
##                                     2.5 %       97.5 %
## (Intercept)                   1467.75629317 1538.8698611
## data_channel_is_entertainment  -403.76991542 -343.1466631
## data_channel_is_socmed          356.45013776  456.2751912
## data_channel_is_world          -228.59146119 -126.1292503
## kw_avg_avg                        0.08690009    0.1057705
## LDA_02                         -372.66818744 -226.6203518
## weekday_is_saturday:is_weekend  391.29454466  485.0492264
```

```
plot(newsLm)
```


Residuals vs Fitted
Residuals
2000
0
−4000
1000
2000
3000
4000
5000
Fitted values
lm(shares ~ data_channel_is_entertainment + data_channel_is_socmed + data_c ..


Normal Q−Q
Standardized residuals
Theoretical Quantiles
lm(shares ~ data_channel_is_entertainment + data_channel_is_socmed + data_c ..
10295
−4
−2
0
2
4


Scale−Location
√|Standardized residuals|
0.0
0.5
1.0
1.5
2.0
1000
2000
3000
4000
5000
16295
Fitted values
lm(shares ~ data_channel_is_entertainment + data_channel_is_socmed + data_c ..



Refer to above code. The above most highly correlated variables were plugged into a multiple linear regression model to predict number of shares for an article. The features weekday_is_saturday and is_weekend were used as an interaction term since they seem to be strongly related from a higher-level non-scientific perspective and for obvious reasons (Saturday is part of the weekend). The resulting model's terms were all statistically significant at a 5% level with p-values < 0.025. All of the coefficients were estimated to be in the positive of negative hundreds or thousands (i.e. the intercept is 1503, data_channel_is_entertainment is -373.5, and data_channel_is_socmed is 4.06.4). The only exception to this is kw_avg_avg, which has a coefficient of 0.0963, which is a decent amount closer to having a coefficient of 0 than the other terms. The RSS was 1065, which is significantly smaller than what was experienced in pre-liminary testing without removing outliers, when the RSS was around 11k. The multiple r-squred is 0.069, which is not too bad considering we are working with 60 predictors. The resulting 95% confidence intervals for the coefficients are relatively small, which is a positive because we can tell the general trend of the data and be confident since the range of possible values doesn't vary too much for each coefficient. Overall, the residuals are high, ranging from -4149 to 4316. This is not good news since the shares column ranges from values 1 to 5500. The better news is the first quartile of the residuals is -702.7 and the third quartile is 376.7, so the middle 50% of data doesn't differ by a ton.

```
newsDataBinary <- data.frame(newsDataQuant) # make a copy
newsDataBinary$shares <- ifelse(newsDataBinary$shares > quantile(newsDataBinary$shares, 0.5), 1, 0)
head(newsDataBinary)
```

```
##   timedelta n_tokens_title n_tokens_content n_unique_tokens
## 1       731             12              219       0.6635945
## 2       731              9              255       0.6047431
## 3       731              9              211       0.5751295
## 4       731              9              531       0.5037879
## 5       731             13             1072       0.4156456
```

```
## 6       731             10              370       0.5598886
##   n_non_stop_words n_non_stop_unique_tokens num_hrefs num_self_hrefs
## 1                1                0.8153846         4              2
## 2                1                0.7919463         3              1
## 3                1                0.6638655         3              1
## 4                1                0.6656347         9              0
## 5                1                0.5408895        19             19
## 6                1                0.6981982         2              2
##   num_imgs num_videos average_token_length num_keywords
## 1        1          0              4.680365            5
## 2        1          0              4.913725            4
## 3        1          0              4.393365            6
## 4        1          0              4.404896            7
## 5       20          0              4.682836            7
## 6        0          0              4.359459            9
##   data_channel_is_lifestyle data_channel_is_entertainment
## 1                         0                             1
## 2                         0                             0
## 3                         0                             0
## 4                         0                             1
## 5                         0                             0
## 6                         0                             0
##   data_channel_is_bus data_channel_is_socmed data_channel_is_tech
## 1                   0                      0                   0
## 2                   1                      0                   0
## 3                   1                      0                   0
## 4                   0                      0                   0
## 5                   0                      0                   1
## 6                   0                      0                   1
##   data_channel_is_world kw_min_min kw_max_min kw_avg_min kw_min_max
## 1                     0          0          0          0          0
## 2                     0          0          0          0          0
## 3                     0          0          0          0          0
## 4                     0          0          0          0          0
## 5                     0          0          0          0          0
## 6                     0          0          0          0          0
##   kw_max_max kw_avg_max kw_min_avg kw_max_avg kw_avg_avg
## 1          0          0          0          0          0
## 2          0          0          0          0          0
## 3          0          0          0          0          0
## 4          0          0          0          0          0
## 5          0          0          0          0          0
## 6          0          0          0          0          0
##   self_reference_min_shares self_reference_max_shares
## 1                       496                       496
## 2                         0                         0
## 3                       918                       918
## 4                         0                         0
## 5                       545                     16000
## 6                      8500                      8500
##   self_reference_avg_sharess weekday_is_monday weekday_is_tuesday
## 1                    496.000                 1                  0
## 2                      0.000                 1                  0
## 3                    918.000                 1                  0
```

```
## 4                     0.000                1                 0
## 5                  3151.158                1                 0
## 6                  8500.000                1                 0
##   weekday_is_wednesday weekday_is_thursday weekday_is_friday
## 1                    0                   0                 0
## 2                    0                   0                 0
## 3                    0                   0                 0
## 4                    0                   0                 0
## 5                    0                   0                 0
## 6                    0                   0                 0
##   weekday_is_saturday weekday_is_sunday is_weekend     LDA_00     LDA_01
## 1                   0                 0          0 0.50033120 0.37827893
## 2                   0                 0          0 0.79975569 0.05004668
## 3                   0                 0          0 0.21779229 0.03333446
## 4                   0                 0          0 0.02857322 0.41929964
## 5                   0                 0          0 0.02863281 0.02879355
## 6                   0                 0          0 0.02224528 0.30671758
##       LDA_02     LDA_03     LDA_04 global_subjectivity
## 1 0.04000468 0.04126265 0.04012254           0.5216171
## 2 0.05009625 0.05010067 0.05000071           0.3412458
## 3 0.03335142 0.03333354 0.68218829           0.7022222
## 4 0.49465083 0.02890472 0.02857160           0.4298497
## 5 0.02857518 0.02857168 0.88542678           0.5135021
## 6 0.02223128 0.02222429 0.62658158           0.4374086
##   global_sentiment_polarity global_rate_positive_words
## 1                0.09256198                0.04566210
## 2                0.14894781                0.04313725
## 3                0.32333333                0.05687204
## 4                0.10070467                0.04143126
## 5                0.28100348                0.07462687
## 6                0.07118419                0.02972973
##   global_rate_negative_words rate_positive_words rate_negative_words
## 1                0.013698630           0.7692308           0.2307692
## 2                0.015686275           0.7333333           0.2666667
## 3                0.009478673           0.8571429           0.1428571
## 4                0.020715631           0.6666667           0.3333333
## 5                0.012126866           0.8602151           0.1397849
## 6                0.027027027           0.5238095           0.4761905
##   avg_positive_polarity min_positive_polarity max_positive_polarity
## 1             0.3786364            0.10000000                   0.7
## 2             0.2869146            0.03333333                   0.7
## 3             0.4958333            0.10000000                   1.0
## 4             0.3859652            0.13636364                   0.8
## 5             0.4111274            0.03333333                   1.0
## 6             0.3506100            0.13636364                   0.6
##   avg_negative_polarity min_negative_polarity max_negative_polarity
## 1            -0.3500000               -0.600           -0.2000000
## 2            -0.1187500               -0.125           -0.1000000
## 3            -0.4666667               -0.800           -0.1333333
## 4            -0.3696970               -0.600           -0.1666667
## 5            -0.2201923               -0.500           -0.0500000
## 6            -0.1950000               -0.400           -0.1000000
##   title_subjectivity title_sentiment_polarity abs_title_subjectivity
## 1          0.5000000              -0.1875000             0.00000000
```

```
## 2          0.0000000                0.0000000             0.50000000
## 3          0.0000000                0.0000000             0.50000000
## 4          0.0000000                0.0000000             0.50000000
## 5          0.4545455                0.1363636             0.04545455
## 6          0.6428571                0.2142857             0.14285714
##   abs_title_sentiment_polarity shares
## 1                    0.1875000      0
## 2                    0.0000000      0
## 3                    0.0000000      1
## 4                    0.0000000      0
## 5                    0.1363636      0
## 6                    0.2142857      0
```

```
newsClassif <- newsDataBinary # current classification set we're using
trainIndex <- sample(1:nrow(newsClassif), 0.8*nrow(newsClassif)) # train indices
testIndex <- setdiff(1:nrow(newsClassif), trainIndex) # test indices
train <- newsClassif[trainIndex,]
test <- newsClassif[testIndex,]
trainX <- newsClassif[trainIndex, -61]
trainY <- newsClassif[trainIndex, "shares"]
testX <- as.data.frame(newsClassif[testIndex, -61])
testY <- as.data.frame(newsClassif[testIndex, "shares"])

newsLogit <- glm(shares ~ data_channel_is_entertainment + data_channel_is_socmed + data_channel_is_world
summary(newsLogit)
```

```
##
## Call:
## glm(formula = shares ~ data_channel_is_entertainment + data_channel_is_socmed +
##     data_channel_is_world + kw_avg_avg + LDA_02 + weekday_is_saturday:is_weekend,
##     family = binomial, data = newsDataBinary)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.3302  -1.1270  -0.8155   1.1339   1.8008
##
## Coefficients:
##                                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)                    -2.956e-01  3.817e-02  -7.744 9.66e-15 ***
## data_channel_is_entertainment  -8.040e-01  3.052e-02 -26.343  < 2e-16 ***
## data_channel_is_socmed          8.407e-01  5.330e-02  15.774  < 2e-16 ***
## data_channel_is_world          -3.374e-01  5.095e-02  -6.623 3.51e-11 ***
## kw_avg_avg                      1.479e-04  1.053e-05  14.055  < 2e-16 ***
## LDA_02                         -6.817e-01  7.354e-02  -9.270  < 2e-16 ***
## weekday_is_saturday:is_weekend  1.044e+00  5.047e-02  20.688  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 48563  on 35102  degrees of freedom
## Residual deviance: 45884  on 35096  degrees of freedom
## AIC: 45898
##
## Number of Fisher Scoring iterations: 4
```

```
confint(newsLogit)
```

```
## Waiting for profiling to be done...
```

```
##                                       2.5 %        97.5 %
## (Intercept)                     -0.3706079938 -0.2209943394
## data_channel_is_entertainment   -0.8639628703 -0.7443142378
## data_channel_is_socmed           0.7368689370  0.9458269258
## data_channel_is_world           -0.4373372228 -0.2376227841
## kw_avg_avg                       0.0001274285  0.0001686885
## LDA_02                          -0.8259220178 -0.5376280440
## weekday_is_saturday:is_weekend   0.9457033653  1.1435645793
```

```
logitPredict <- predict(newsLogit, testX, type="response")
table(logitPredict > 0.5, t(testY))
```

```
##
##            0    1
##   FALSE 2246 1237
##   TRUE  1450 2088
```

Refer to above code. Next, the number of shares was binarized, converting the problem from regression to classification. If the number of shares was less than the median, it was replaced with a 0, and it was replaced with a 1 otherwise, turning it into a binary classification problem with balanced classes. The range of residuals is now [-3.330, 1.801], whereas it was [-4148.4, 4315.9] for multiple linear regression. This reduction is to be expected since residuals are just the predicted value minus the expected value and the scale has changed since we're working with binarized data for the shares. The threshold for binarization of the logistic regression predictions is a probability of 50%. A confusion matrix is provided. The accuracy of the model is 61.84%, with true positive rates of both classes hovering around 61%. It is difficult to compare this to the linear regression model above since this task was a binary classification. Also, the response feature has been scaled from the original data, so we cannot compare their coefficient values. All we can say is all coefficients are still statistically significant, agreeing with the linear regression model.

```
newsDataTrinary <- data.frame(newsDataQuant) # make a copy
newsDataTrinary$shares <- ifelse(newsDataTrinary$shares > quantile(newsDataTrinary$shares, 0.333), ifels

newsClassif <- newsDataBinary # current classification set we're using for all classification problems

trainIndex <- sample(1:nrow(newsClassif), 0.8*nrow(newsClassif)) # train indices
testIndex <- setdiff(1:nrow(newsClassif), trainIndex) # test indices
train <- newsClassif[trainIndex,]
test <- newsClassif[testIndex,]
trainX <- newsClassif[trainIndex, -61]
trainY <- newsClassif[trainIndex, "shares"]
testX <- as.data.frame(newsClassif[testIndex, -61])
testY <- as.data.frame(newsClassif[testIndex, "shares"])

newsLda <- lda(shares ~ data_channel_is_entertainment + data_channel_is_socmed + data_channel_is_world +
newsLda
```

```
## Call:
## lda(shares ~ data_channel_is_entertainment + data_channel_is_socmed +
##     data_channel_is_world + kw_avg_avg + LDA_02 + weekday_is_saturday:is_weekend,
##     data = newsClassif)
##
## Prior probabilities of groups:
```

```
##         0         1
## 0.5267356 0.4732644
##
## Group means:
##   data_channel_is_entertainment data_channel_is_socmed
## 0                     0.2247161             0.03044889
## 1                     0.1319449             0.08583639
##   data_channel_is_world kw_avg_avg    LDA_02
## 0             0.2800433   2913.395 0.2645749
## 1             0.1625233   3213.397 0.1832966
##   weekday_is_saturday:is_weekend
## 0                     0.03380206
## 1                     0.08956841
##
## Coefficients of linear discriminants:
##                                          LD1
## data_channel_is_entertainment  -1.4743744124
## data_channel_is_socmed          1.4471979727
## data_channel_is_world          -0.6561855569
## kw_avg_avg                      0.0002395235
## LDA_02                         -1.2081622799
## weekday_is_saturday:is_weekend  1.7842973409
```

```
ldaPredict <- predict(newsLda, testX)$class
table(t(ldaPredict), t(testY))
```

```
##
##        0    1
##   0 2221 1205
##   1 1517 2078
```

```
newsQda <- qda(shares ~ data_channel_is_entertainment + data_channel_is_socmed + data_channel_is_world +
newsQda
```

```
## Call:
## qda(shares ~ data_channel_is_entertainment + data_channel_is_socmed +
##     data_channel_is_world + kw_avg_avg + LDA_02 + weekday_is_saturday:is_weekend,
##     data = newsClassif)
##
## Prior probabilities of groups:
##         0         1
## 0.5267356 0.4732644
##
## Group means:
##   data_channel_is_entertainment data_channel_is_socmed
## 0                     0.2247161             0.03044889
## 1                     0.1319449             0.08583639
##   data_channel_is_world kw_avg_avg    LDA_02
## 0             0.2800433   2913.395 0.2645749
## 1             0.1625233   3213.397 0.1832966
##   weekday_is_saturday:is_weekend
## 0                     0.03380206
## 1                     0.08956841
```

```
qdaPredict <- predict(newsQda, testX)$class
table(t(qdaPredict), t(testY))
```

```
##
##         0    1
##    0 3439 2577
##    1  299  706
```

Refer to above code. The final step to convert this problem from regression to classification was to convert labels to -1, 0, and 1 for the shares (response) feature. In this case, -1 represents a non-popular article, 1 is a popular article, and 0 means the article wasn't popular or non-popular. The splits were made at the 1/3 and 2/3 quantiles.

In between training the kNN and converting the classifcation into a three-way task, both LDA and QDA were tested. LDA on a 3-way task resulted in a 43.56% accuracy and the 2-way task had a 61.57% accuracy, both a decent amount above their baselines of 33.33% and 50.00%. QDA resulted in a 3-way accuracy of 40.12% and a 2-way accuracy of 58.60%. It is important to note the true negative rate on the binary task is 91.24%, but the true positive rate is only 21.21%. This does not occur in LDA, as its TPR and TNR are relatively equal. In both QDA and LDA, the "neutral" class in between the popular and non-popular classes in three-way classification resulted in the highest false negative rate. This is not surprising since we expect the most ambiguous class to perform the worst in terms of Type I and Type II errors.

```
newsKnn <- kNN(shares ~ data_channel_is_entertainment + data_channel_is_socmed + data_channel_is_world +
table(newsKnn, t(testY)) # confusion matrix
```

```
##
## newsKnn    0    1
##       0 2773 1597
##       1  965 1686
```

Refer to above code. Then, the kNN function (found in the "DMwR" package), an abstracted layer on top of the vanilla R knn() function, was used as a classification. Instead of using all 59 predictors and 1 response for knn(), kNN provides the ability to input a formula for training, requiring less preprocessing of data. An 80%/20% train/test split was performed for classification. Those percentages are justified since we already have an abundance of data, and the response is only either two-way or three-way classification, so a test size of 7021 samples should be more than enough to accurately assess the model. A confusion matrix was created on the binary task for kNN (k = 3). Out of the 7021 samples, 1683 (23.97%) were true positives, 2433 (34.65%) were true negatives, 1578 (22.48%) were false negatives, and 1327 (18.90%) were false positives. As such, this results in a 59% binary classification accuracy, only slightly above the baseline of 50%. When k was increased to 11, the true positive and false negative rates stayed consistent, but the true negative rate increased and false positive rate decreased, resulting in an accuracy of 61%. As a simplified summary, when k = 101, accuracy was 64%. Further values of k were not tested at risk of introducing high bias into the model. Due to the high number of samples, k = 101 seems to be a reasonable parameter value, while also producing the highest accuracy of the three tested models, whereas k = 3 might indicate a risk of overfitting. It is important to note this kNN model was only trained on the highly correlated predictors as mentioned above; using all predictors with k = 101 results in an accuracy of 63% with similar TPR/TNR/FPR/FNR. Finally, on the three-way classification task, on the newsDataTrinary data, with k = 101, an accuracy of 46.46% is achieved, which is significantly higher than the baseline os 33.33%. We can conclude kNN with k = 101 is an effective classifier for both classification tasks discussed. In fact, kNN is the preferred classifier over the other two classifiers, LDA and QDA.

Please be aware, I did not explicitly include code for both two-way and three-way classification in the snippets above. Instead, I created one variable called "newsDataClassif" that can be assigned either the binary or trinary data, swapping out the data used for classification tasks for a bunch of commands. As such, if you do not immediately see results for the three-way classification in the graphs/charts/output above, don't be startled, it's just a matter of assigning "newsDataClassif" the value of newsDataTrinary.

To wrap up, we displayed a bunch of information about the dataset. For example, we calculated correlations, removed outliers, performed multiple linear regression, logistic regression, LDA, QDA, and knn with several values of k. We thoroughly investigated the dataset and explored both regression and classification of

the problem, where the regression task involves predicting the number of shares for an example, and the classification task is predicting “popular vs. non-popular” or “popular vs. non-popular vs. in-between”.