hw2

Collin

4/20/2021

1a)

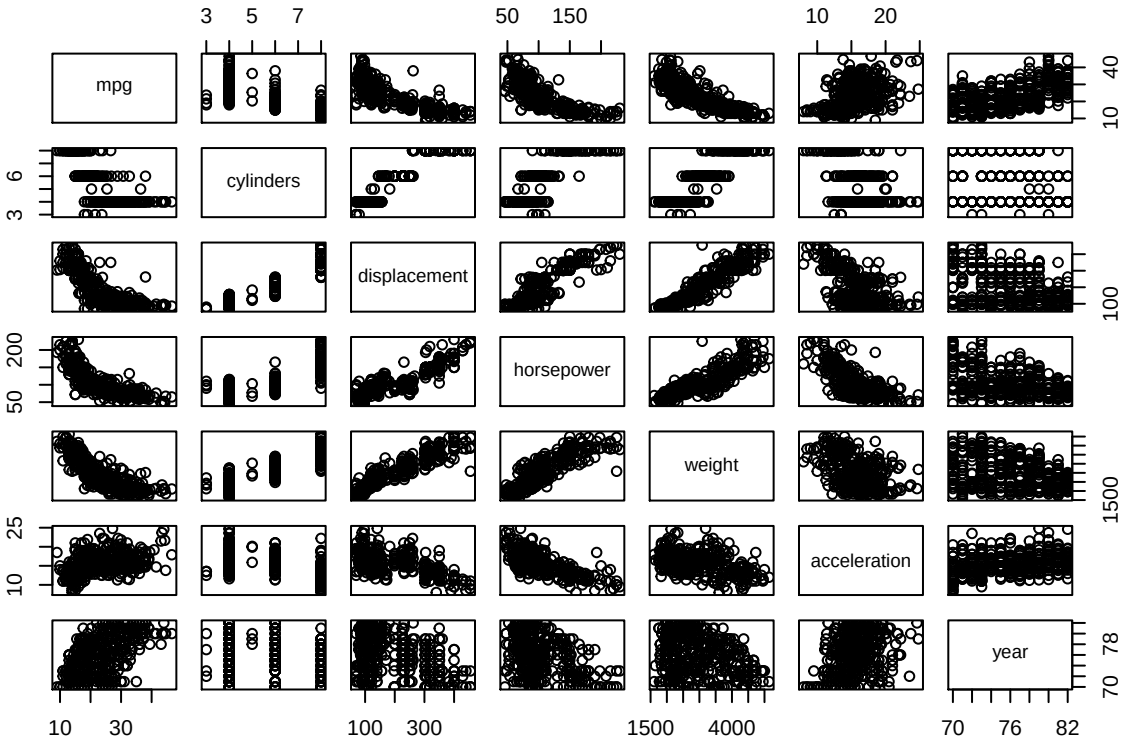
```
library(ggplot2)
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --
```

```
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter()     masks stats::filter()
## x dplyr::group_rows() masks kableExtra::group_rows()
## x dplyr::lag()        masks stats::lag()
```

```
library(ISLR)
options(digits = 3)
data = na.omit(Auto)
pairs(data[,1:7])
```



```
?R2
```

```
## No documentation for 'R2' in specified packages and libraries:
## you could try '??R2'
```

Mpg appears to be negatively correlated with power-related variables, and also has a positive association with year (indicating fuel efficiency has improved over time)

1b)

```
cor(data[,1:7])
```

```
##                 mpg cylinders displacement horsepower weight acceleration
## mpg           1.000    -0.778       -0.805     -0.778 -0.832        0.423
## cylinders    -0.778     1.000        0.951      0.843  0.898       -0.505
## displacement -0.805     0.951        1.000      0.897  0.933       -0.544
## horsepower   -0.778     0.843        0.897      1.000  0.865       -0.689
## weight       -0.832     0.898        0.933      0.865  1.000       -0.417
## acceleration  0.423    -0.505       -0.544     -0.689 -0.417        1.000
## year          0.581    -0.346       -0.370     -0.416 -0.309        0.290
##                year
## mpg           0.581
## cylinders    -0.346
## displacement -0.370
## horsepower   -0.416
## weight       -0.309
## acceleration  0.290
## year          1.000
```

Mpg is negatively correlated with `weight`, `cylinders`, and `displacement`, and it is positively correlated with `acceleration` and `year`.

1c)

```
model = lm(mpg~cylinders+displacement+horsepower+weight+acceleration+year, data = data)
summary(model)
```

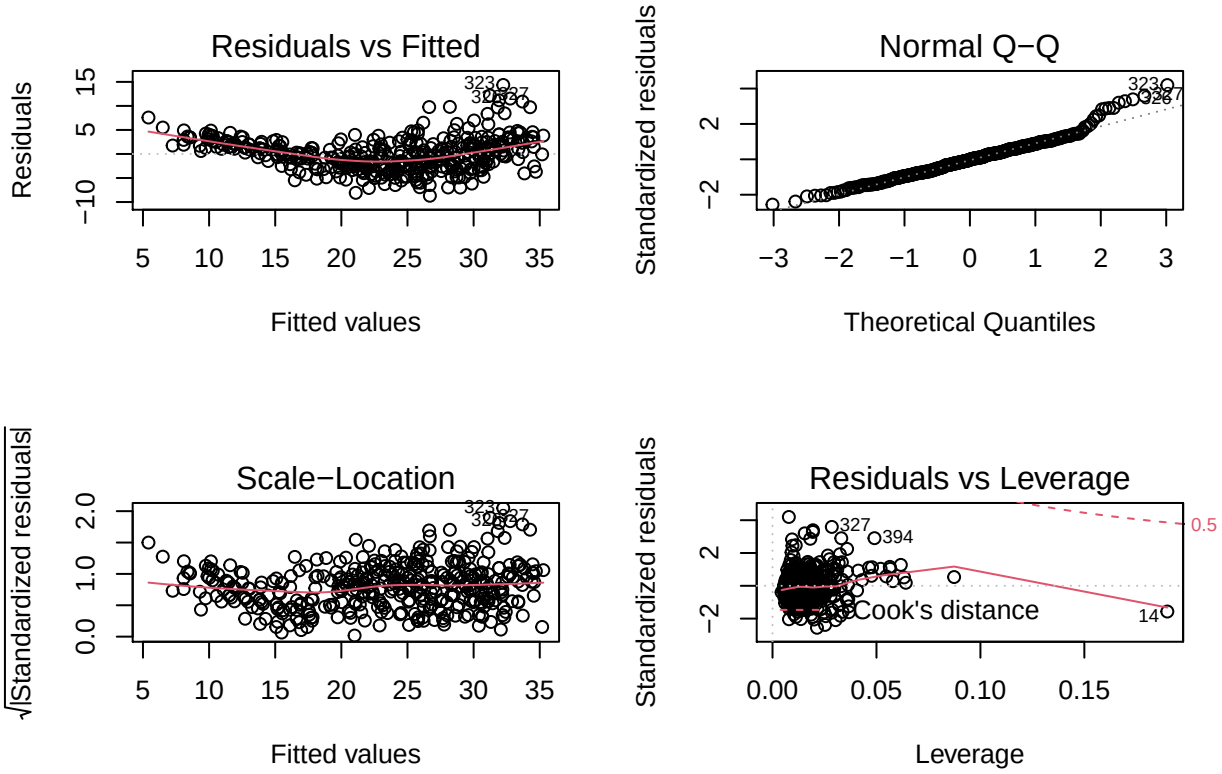
```
##
## Call:
## lm(formula = mpg ~ cylinders + displacement + horsepower + weight +
##     acceleration + year, data = data)
##
## Residuals:
##    Min     1Q Median     3Q    Max
##  -8.69  -2.39  -0.08   2.03  14.36
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -1.45e+01   4.76e+00   -3.05   0.0024 **
## cylinders    -3.30e-01   3.32e-01   -0.99   0.3212
## displacement  7.68e-03   7.36e-03    1.04   0.2973
## horsepower   -3.91e-04   1.38e-02   -0.03   0.9775
## weight       -6.79e-03   6.70e-04  -10.14   <2e-16 ***
## acceleration  8.53e-02   1.02e-01    0.84   0.4038
## year          7.53e-01   5.26e-02   14.32   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 3.44 on 385 degrees of freedom
## Multiple R-squared:  0.809,  Adjusted R-squared:  0.806
## F-statistic:  272 on 6 and 385 DF,  p-value: <2e-16
```

Given an a p-value <2e-16, this regression is overall highly significant. In other words, there definitely does appear to be a relationship between the predictor variables and the response.

1d)

```
par(mfrow=c(2,2))
plot(model)
```



Observation 14 appears to have significantly greater leverage than the rest of the observations. Based on the `Residuals vs Fitted Values` plot, it also appears that there is nonconstant variance. 1e)

```
model_int = lm(mpg~cylinders*year, data = data)
summary(model_int)
```

```
##
## Call:
## lm(formula = mpg ~ cylinders * year, data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -11.216  -2.579  -0.156   2.257  15.253
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)    -61.6178    15.1028   -4.08  5.5e-05 ***
## cylinders        5.5104     2.7370    2.01    0.045 *
## year             1.3405     0.1991    6.73  6.0e-11 ***
```

```
## cylinders:year  -0.1135     0.0365   -3.11    0.002 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.13 on 388 degrees of freedom
## Multiple R-squared:  0.722,  Adjusted R-squared:  0.72
## F-statistic:  336 on 3 and 388 DF,  p-value: <2e-16
```

The interaction between `cylinders` and `year` is statistically significant. This suggests that the effect of `cylinders` on `mpg` may also depend on the `year`. It is also worth pointing out that the effect of the interaction term is negative, and the `cylinders` term is now positive. Perhaps the `cylinders` coefficient in the previous model was picking up some serious bias, and in actuality `cylinders` have become more fuel efficient.

1f)

```
model_nonlinear = lm(mpg~log(cylinders) + I(weight^2) + I(sqrt(displacement)), data = data)
summary(model_nonlinear)
```

```
##
## Call:
## lm(formula = mpg ~ log(cylinders) + I(weight^2) + I(sqrt(displacement)),
##     data = data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -13.295  -2.841  -0.483   2.339  17.460
##
## Coefficients:
##                          Estimate Std. Error t value Pr(>|t|)
## (Intercept)              4.22e+01   1.90e+00   22.23  < 2e-16 ***
## log(cylinders)           1.19e+00   2.37e+00    0.50     0.62
## I(weight^2)             -4.62e-07   1.06e-07   -4.34  1.8e-05 ***
## I(sqrt(displacement)) -1.21e+00   2.47e-01   -4.90  1.4e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4.36 on 388 degrees of freedom
## Multiple R-squared:  0.69,   Adjusted R-squared:  0.687
## F-statistic:  288 on 3 and 388 DF,  p-value: <2e-16
```

Here I take a look at the log, square root and squared transformations of the predictors. `weight` squared and the square root of `displacement` are highly significant, but the log of `cylinders` is not.

2a)

```
library(ISLR)
library(glmnet)
```

```
## Loading required package: Matrix
```

```
##
## Attaching package: 'Matrix'
```

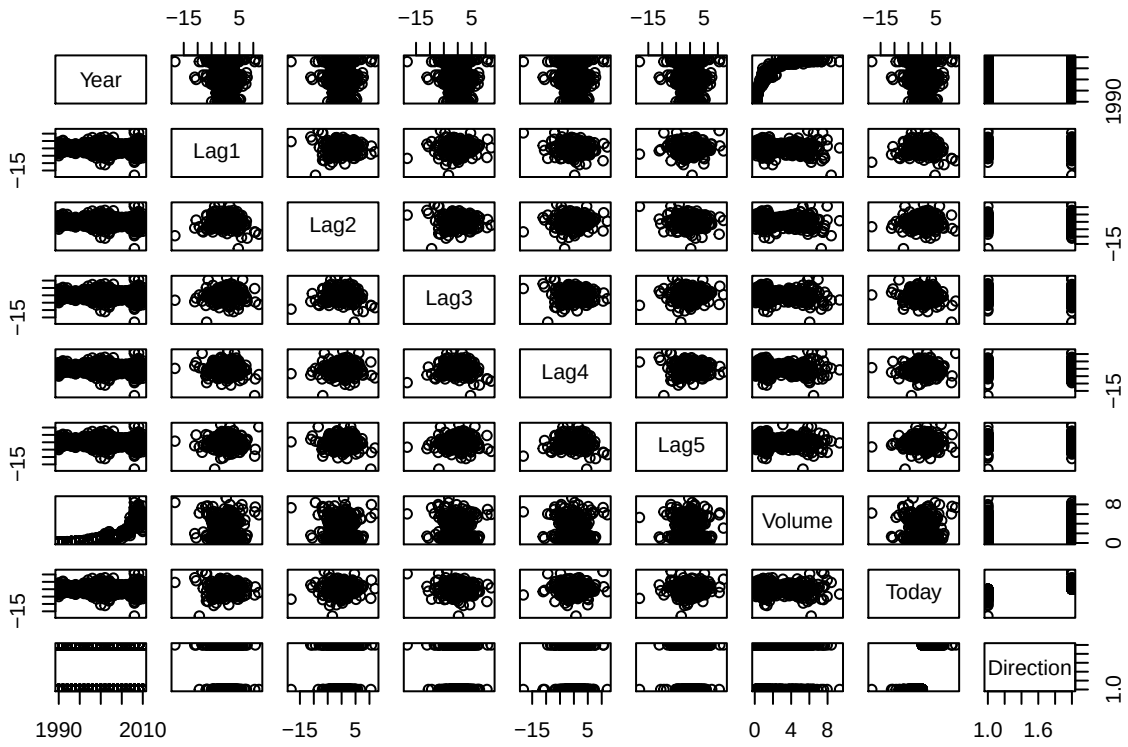
```
## The following objects are masked from 'package:tidyr':
##
##     expand, pack, unpack
```

```
## Loaded glmnet 4.1-1
```

```
summary(Weekly)
```

```
##       Year           Lag1              Lag2              Lag3         
##  Min.   :1990   Min.   :-18.20   Min.   :-18.20   Min.   :-18.20  
##  1st Qu.:1995   1st Qu.: -1.15   1st Qu.: -1.15   1st Qu.: -1.16  
##  Median :2000   Median :  0.24   Median :  0.24   Median :  0.24  
##  Mean   :2000   Mean   :  0.15   Mean   :  0.15   Mean   :  0.15  
##  3rd Qu.:2005   3rd Qu.:  1.41   3rd Qu.:  1.41   3rd Qu.:  1.41  
##  Max.   :2010   Max.   : 12.03   Max.   : 12.03   Max.   : 12.03  
##       Lag4              Lag5              Volume          Today        Direction 
##  Min.   :-18.20   Min.   :-18.20   Min.   :0.09   Min.   :-18.20   Down:484  
##  1st Qu.: -1.16   1st Qu.: -1.17   1st Qu.:0.33   1st Qu.: -1.15   Up  :605  
##  Median :  0.24   Median :  0.23   Median :1.00   Median :  0.24             
##  Mean   :  0.15   Mean   :  0.14   Mean   :1.57   Mean   :  0.15             
##  3rd Qu.:  1.41   3rd Qu.:  1.41   3rd Qu.:2.05   3rd Qu.:  1.41             
##  Max.   : 12.03   Max.   : 12.03   Max.   :9.33   Max.   : 12.03
```

```
pairs(Weekly)
```



Most of the relationships appear very noisy and uninteresting, but there does appear to be some positive relationship between `year` and `volume`.

2b)

```
logistic_model = glm(Direction~., data = Weekly[,c(2:7,9)], family = "binomial")
summary(logistic_model)
```

```
##
## Call:
## glm(formula = Direction ~ ., family = "binomial", data = Weekly[,
##     c(2:7, 9)])
##
## Deviance Residuals:
```

```
##    Min      1Q  Median      3Q     Max
## -1.695  -1.256   0.991   1.085   1.458
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.2669     0.0859    3.11   0.0019 **
## Lag1         -0.0413     0.0264   -1.56   0.1181
## Lag2          0.0584     0.0269    2.18   0.0296 *
## Lag3         -0.0161     0.0267   -0.60   0.5469
## Lag4         -0.0278     0.0265   -1.05   0.2937
## Lag5         -0.0145     0.0264   -0.55   0.5833
## Volume       -0.0227     0.0369   -0.62   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500
##
## Number of Fisher Scoring iterations: 4
```

Lag 2 appears to be the only statistically significant predictor of `Direction`.

2c)

```
log_predict = predict(logistic_model, Weekly, type = "response")
logistic_model_classification = ifelse(log_predict > .5, "Up", "Down")
prediction_table = table(logistic_model_classification, Weekly$Direction)

# counter = 0
# for(direction in Weekly$Direction){
#   if(direction == "Down")
#     counter = counter + 1
# }
prediction_table
```

```
##
## logistic_model_classification Down  Up
##                          Down   54  48
##                          Up    430 557
```

```
#accuracy
sum(diag(prediction_table))/sum(prediction_table)
```

```
## [1] 0.561
```

The prediction accuracy is about 56% overall, and it shows when the logistic regression is right and wrong. For `Down` predictions, it is approximately $ 53%$ accurate, and for `Up` predictions it is approximately $\approx 56\%$ accurate. Overall, this is not a very accurate model, as it hardly beats a prediction under pure chance (50/50).

2d)

```
set.seed(1)
train = Weekly %>% filter(Year >= 1990 & Year <= 2008) #training data
test = Weekly %>% filter(Year > 2008) #test data
logistic_model2 = glm(Direction~Lag2, data = train, family = "binomial")
```

```
summary(logistic_model2)
```

```
##
## Call:
## glm(formula = Direction ~ Lag2, family = "binomial", data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -1.54   -1.26    1.02    1.09    1.37
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)   0.2033     0.0643    3.16   0.0016 **
## Lag2          0.0581     0.0287    2.02   0.0430 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1350.5  on 983  degrees of freedom
## AIC: 1355
##
## Number of Fisher Scoring iterations: 4
```

```
log2_pred = predict(logistic_model2, test, type = "response")
log2.pred = ifelse(log2_pred > .5, "Up", "Down")
prediction_table2 = table(log2.pred, test$Direction)
prediction_table2
```

```
##
## log2.pred Down Up
##      Down    9  5
##      Up     34 56
```

```
sum(diag(prediction_table2))/sum(prediction_table2)
```

```
## [1] 0.625
```

We now have an accuracy rate of about 62.5%. The model got 9/14 `Down` predictions and 59/90 `Up` predictions correct, respectively.

2e)

```
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```
lda_model = lda(Direction~Lag2, data = train)
lda_pred = predict(lda_model, test)$class
table(lda_pred, test$Direction)
```

```
##
```

```
## lda_pred Down Up
##     Down    9  5
##     Up     34 56
```

```
sum(diag(table(lda_pred, test$Direction)))/sum(table(lda_pred, test$Direction))
```

```
## [1] 0.625
```

Interestingly enough, this results in the same accuracy as the previous logistic regression model with one predictor (Lag2).

2f)

```
qda_model = qda(Direction~Lag2, data = train)
qda_pred = predict(qda_model, test)$class
table(qda_pred, test$Direction)
```

```
##
## qda_pred Down Up
##     Down    0  0
##     Up     43 61
```

```
sum(diag(table(qda_pred, test$Direction)))/sum(table(qda_pred, test$Direction))
```

```
## [1] 0.587
```

This quadratic linear discriminant model results in a lower prediction accuracy (58.7%) compared to the second logistic regression model and LDA model.

2g)

```
library(class)
train_knn = as.matrix(train$Lag2)
test_knn = as.matrix(test$Lag2)
knn_pred = knn(train_knn, test_knn, train$Direction, k=1)
table(knn_pred, test$Direction)
```

```
##
## knn_pred Down Up
##     Down   21 30
##     Up     22 31
```

```
sum(diag(table(knn_pred, test$Direction)))/sum(table(knn_pred, test$Direction))
```

```
## [1] 0.5
```

2h) The logistic regression and LDA models appear to have produced the best results, with QDA following suit. This K-nearest neighbors model, with k = 1, appears to be basically worthless, with an accuracy rate of .51.

2i)

```
#k = 2
knn_pred2 = knn(train_knn, test_knn, train$Direction, k=2)
table(knn_pred2, test$Direction)
```

```
##
## knn_pred2 Down Up
##      Down   18 25
##      Up     25 36
```

```r
knn_pred2_accuracy = sum(diag(table(knn_pred2, test$Direction)))/sum(table(knn_pred2, test$Direction))
```

```r
knn_pred5 = knn(train_knn, test_knn, train$Direction, k=5)
table(knn_pred5, test$Direction)
```

```
##
## knn_pred5 Down Up
##      Down   16 22
##      Up     27 39
```

```r
knn_pred5_accuracy = sum(diag(table(knn_pred5, test$Direction)))/sum(table(knn_pred5, test$Direction))
```

```r
knn_pred10 = knn(train_knn, test_knn, train$Direction, k=10)
table(knn_pred10, test$Direction)
```

```
##
## knn_pred10 Down Up
##       Down   18 21
##       Up     25 40
```

```r
knn_pred10_accuracy = sum(diag(table(knn_pred10, test$Direction)))/sum(table(knn_pred10, test$Direction
```

```r
logistic_model3 = glm(Direction~I(Lag1^2)+I(Lag2^2), data = train, family = "binomial")
summary(logistic_model3)
```

```
##
## Call:
## glm(formula = Direction ~ I(Lag1^2) + I(Lag2^2), family = "binomial",
##     data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
##  -1.64   -1.26    1.08    1.09    1.10
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 0.192999   0.069800    2.77   0.0057 **
## I(Lag1^2)   0.000816   0.004574    0.18   0.8584
## I(Lag2^2)   0.002491   0.004742    0.53   0.5994
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1354.3  on 982  degrees of freedom
## AIC: 1360
##
## Number of Fisher Scoring iterations: 3
```

```
log3_pred = predict(logistic_model3, test, type = "response")
log3.pred = ifelse(log3_pred > .5, "Up", "Down")
prediction_table3 = table(log3.pred, test$Direction)
prediction_table3
```

```
##
## log3.pred Down Up
##        Up   43 61
```

```
sum(diag(prediction_table3))/sum(prediction_table3)
```

```
## [1] 0.413
```

```
#dataframe with results

results = tibble(pred2 = knn_pred2_accuracy, pred5 = knn_pred5_accuracy, pred10 = knn_pred10_accuracy)
results = results %>% rename("k_equals_2" = "pred2") %>% rename("k_equals_5" = "pred5") %>% rename("k_e
results %>%
  kbl() %>%
  kable_styling()
```

k_equals_2	k_equals_5	k_equals_10
0.519	0.529	0.558

Funny enough, with this third logistic regression model (with squared transformed variables `Lag1` and `Lag2`), the accuracy his *horrible*: about .41.

Looking at the various KNN models, it appears that the model with $K = 10$ is the most accurate, with an accuracy rate of about 56%. Granted, its accuracy rate is marginally better than the models where $K = 2$ (52%) and $K = 10$ (53%)