Project Part 3
Andrew Schwartz (abschwa2)

Goals of the analysis:

The objective of this analysis is to see if a logistic regression model can be made using the stock data gathered to predict the classification of whether or not a stock's price will be within 15% of its 52-week high price. The analysis will be conducted using R studio.

R Studio Analysis:

Building the Models:

After reading in our merged and wrangled data set, the first thing we can do is make a subset of the data containing only variables of interest. Attributes that serve the purpose of identifying a specific stock such as a stocks company name, ticker symbol, and its SEC filings link will not give any insight to the goal of our analysis, so they are not initially included.

```
> stockSub <- subset(stocks, select=c(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 17, 18))
>
```

Next we ensure that our categorical variables are being treated as factors:

```
> stockSub$Sector <- as.factor(stockSub$Sector)
> stockSub$SubIndustry <- as.factor(stockSub$SubIndustry)
> stockSub$Headquarters <- as.factor(stockSub$Headquarters)
>
```

Now, we split the data into a training data set that will be used to build the logistic regression model, and a test data set that will be used to measure the accuracy of the model. The data is split randomly into 70% training data and 30% testing data.

```
> ind <- sample(2, nrow(stockSub),replace=T,prob=c(0.70,0.30))
> train_data = stockSub[ind==1,]
> test_data = stockSub[ind==2,]
>
```

Now, using the training data and the glm() function, the model can be built:

```
> model <- glm(Within15 ~.,family=binomial(link='logit'),data=train_data)
Warning messages:
1: glm.fit: algorithm did not converge
```

When initially trying to build the model, we get an interesting error. The “algorithm did not converge” error can occur for a couple of reasons, but it is most likely due to something called “perfect separation” where a predictor variable perfectly separates the response variable. This makes sense given that if the “percent52” variable is less than or equal to 15%, then the corresponding boolean response value will always be set as true. This error can also be caused by something called “multicollinearity” where predictor variables are linearly dependent. The “percent52” variable will be removed from the analysis for the reason mentioned. Furthermore, the “Price” and “52WeekHigh" predictors were used to calculate the “percent52” attribute (Percent52 = 1 - (price / 52WeekHigh)) * 100) and thus gives these variables linear dependency for calculating the response variables. Only one of the two attributes can be included at a time, so 3 models will be initially built: 1 using neither the “Price” or “52WeekHigh” attribute and then 2 models that include only one of the attributes respectively. Additionally, the inclusion of the “SubIndustry” and “Headquarters” attributes also lead to the error, so they were not included in the analysis. The most likely reason that these attributes can not be included is due to the fact that each value is very specific, and although they are categorical variables, there is very little grouping of stocks with similar values. With multiple categories spreading the data very thin, this would make it very difficult for the model to converge the data on a solution for the model.

By making 3 specific subsets of the original data along with their corresponding test and training data sets as done previously, we can then build the models by retrying the glm() function.

```
> model_Neither <- glm(Within15 ~.,family=binomial(link='logit'),data=train_Neither)
> model_Price <- glm(Within15 ~.,family=binomial(link='logit'),data=train_Price)
> model_52High <- glm(Within15 ~.,family=binomial(link='logit'),data=train_52High)
>
```

Assessing the Models:

The summary() function is used to view each of the models. To pick a model to move forward with for the analysis, the model with the smallest Akaike’s Information Criterion (AIC) will be chosen. The model with neither the price or the 52WeekHigh attribute had an AIC of 346.72, the model with the only the price attribute had an AIC of 338.86, and the model with only the 52WeekHigh attribute had an AIC of 343.11. Therefore, the model that uses the price attribute and not the 52WeekHigh attribute will be chosen as

the model to use moving forward.The coefficients for the regression model are shown in the summary() function:

```
> summary(model_Price)

Call:
glm(formula = Within15 ~ ., family = binomial(link = "logit"),
    data = train_Price)

Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                    8.076e+00  1.774e+01   0.455  0.64890
SectorConsumer Staples        -6.994e-01  6.276e-01  -1.114  0.26507
SectorEnergy                  -1.698e+00  7.802e-01  -2.177  0.02949 *
SectorFinancials               1.802e-01  6.717e-01   0.268  0.78850
SectorHealth Care             -3.618e-01  6.116e-01  -0.592  0.55410
SectorIndustrials              1.887e-03  6.095e-01   0.003  0.99753
SectorInformation Technology  -1.692e+00  6.025e-01  -2.809  0.00497 **
SectorMaterials                4.486e-01  7.985e-01   0.562  0.57421
SectorReal Estate             -8.384e-01  9.921e-01  -0.845  0.39806
SectorTelecommunication Services -9.662e-01  2.125e+00  -0.455  0.64933
SectorUtilities               -1.479e+00  7.833e-01  -1.889  0.05893 .
DateAdded                     -8.032e-03  9.155e-03  -0.877  0.38034
Founded                        4.694e-03  3.799e-03   1.236  0.21652
Price                          3.373e-02  1.201e-02   2.808  0.00498 **
PriceToEarnings                5.898e-03  4.920e-03   1.199  0.23066
DividendYield                 -1.566e-01  1.565e-01  -1.001  0.31685
EarningsPerShare               5.875e-02  5.051e-02   1.163  0.24484
X52WeekLow                    -4.669e-02  1.456e-02  -3.207  0.00134 **
MarketCap                      6.695e-12  4.758e-12   1.407  0.15941
EBITDA                        -2.744e-11  5.244e-11  -0.523  0.60072
PriceToSales                   3.282e-02  5.964e-02   0.550  0.58214
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 363.73  on 274  degrees of freedom
Residual deviance: 296.85  on 254  degrees of freedom
AIC: 338.85

Number of Fisher Scoring iterations: 5

>
```

From this information, we can see the variables that had the most influence over our model. The Sector, Price, and 52WeekLow attributes are the variables that had the most influence on the outcome of our model. We can use the anova() function to measure the deviance of our model to see how much variability in the response variable was explained by the explanatory variables.

```
> anova(model_Price, test="Chisq")
Analysis of Deviance Table

Model: binomial, link: logit

Response: Within15

Terms added sequentially (first to last)


                Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
NULL                             274     363.73
Sector          10   37.188       264     326.54  5.25e-05 ***
DateAdded        1    0.142       263     326.40 0.7061295
Founded          1    3.702       262     322.70 0.0543575 .
Price            1    0.549       261     322.15 0.4587313
PriceToEarnings  1    4.191       260     317.96 0.0406429 *
DividendYield    1    1.391       259     316.57 0.2382326
EarningsPerShare 1    0.845       258     315.72 0.3578940
X52WeekLow       1   13.902       257     301.82 0.0001925 ***
MarketCap        1    4.330       256     297.49 0.0374376 *
EBITDA           1    0.340       255     297.15 0.5598329
PriceToSales     1    0.305       254     296.85 0.5808195
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
>
```

This table shows that the Sector, Founded, PriceToEarnings, 52WeekLow, and MarketCap were significant attributes in explaining the variability of the Within15 response variable. The price attribute is interesting in that it is significant in terms of explaining the output of the response variable but not significant in explaining the variability of the response variable. There is a very high deviance for the null model, which indicates a poor performance for when we only include the intercept

Testing the Models:

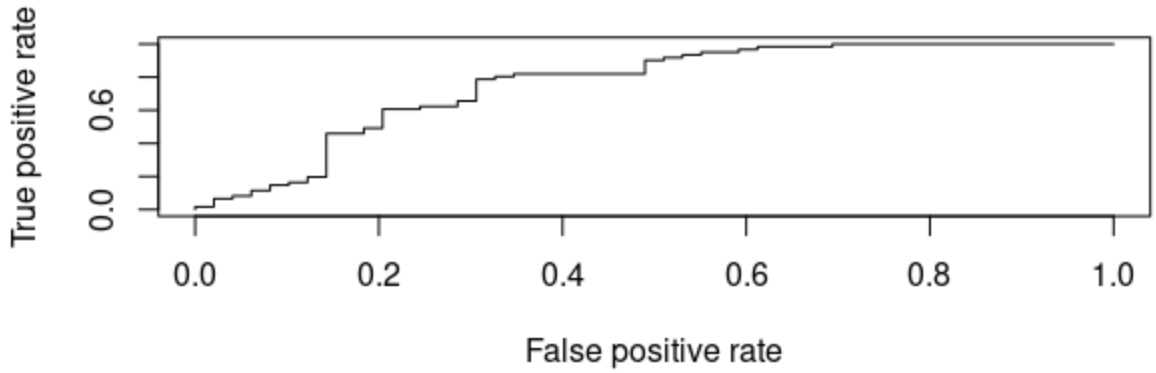
Now the model is ready to be tested with the test data. Because the response variable is binary, values predicted over 0.5 will be assigned the value of 1 (meaning the stock's price is within 15% of its 52 week high), otherwise the value will be 0.

```
> results <- predict(model_Price, test_Price, type='response')
> results <- ifelse(results > 0.5,1,0)
> error <- mean(results != test_data$Within15)
> print(paste('Accuracy',1-error))
[1] "Accuracy 0.718181818181818"
>
```

The model predicted the “Within15” attribute with 72% accuracy (and a corresponding 28% error)

We can also use the non-rounded results to also view the rate of false positives to the rate of true positives, and then view the corresponding area under the curve

```
> p <- predict(model_Price, test_Price, type='response')
> pr <- prediction(p, test_Price$Within15)
> prf <- performance(pr, measure = "tpr", x.measure = "fpr")
> plot(prf)
> auc <- performance(pr, measure = "auc")
> auc <- auc@y.values[[1]]
> auc
[1] 0.7591168
>
```



We see some interesting results in the plot and corresponding area under the curve (AUC). If the model gave an AUC equal to 1, the model would be able to correctly distinguish between all the positive and the negative “Within15” values of stocks. If, however, the AUC had been 0, then the model would have predicted all negative “Within15” values as positives and all positive values as negatives. An AUC of 0.5 would simply suggest that “we are right just about as much as we are wrong” and is not desirable viewing the accuracy of the model. Because the model has an AUC of 0.76, there is a high chance that the classifier will be able to distinguish the positive class values from the negative ones, and this is a relatively good value for the model.

Conclusion and Reflection of The Results

The results of the model show that its ability to predict whether a stock’s price was within 15% of its 52 week high price were relatively successful. There is no clear cut formula for why stock’s price is performing at a certain price range from its 52 week high price, otherwise everyone who tried investing in the stock market would be able to use such a formula to succeed. Given this notion, a model with a 72% prediction accuracy and 0.76 AUC for the rate of false positives vs true positives is relatively good. For future enhancement, updated data for stocks listing within the S&P 500 could be pulled in for further analysis or more attributes about the current stocks could be added to enhance the current regression model. The model has a number of applications. Say that the constructed model falsely predicts that a stock is performing within 15% of its 52 week high price, when in reality it is not. This stock could be an outlier for the model, but it also could be an indication that the stock is undervalued, and that it might enter within 15% of its 52 week high price in the near future. This information would be valuable in determining if an investor should buy a given stock.