# project_0414

*Ting-Wei Lin*

*4/14/2020*

## Logistic Regression

### Data

Joint two datasets, trump tweets and s&p.

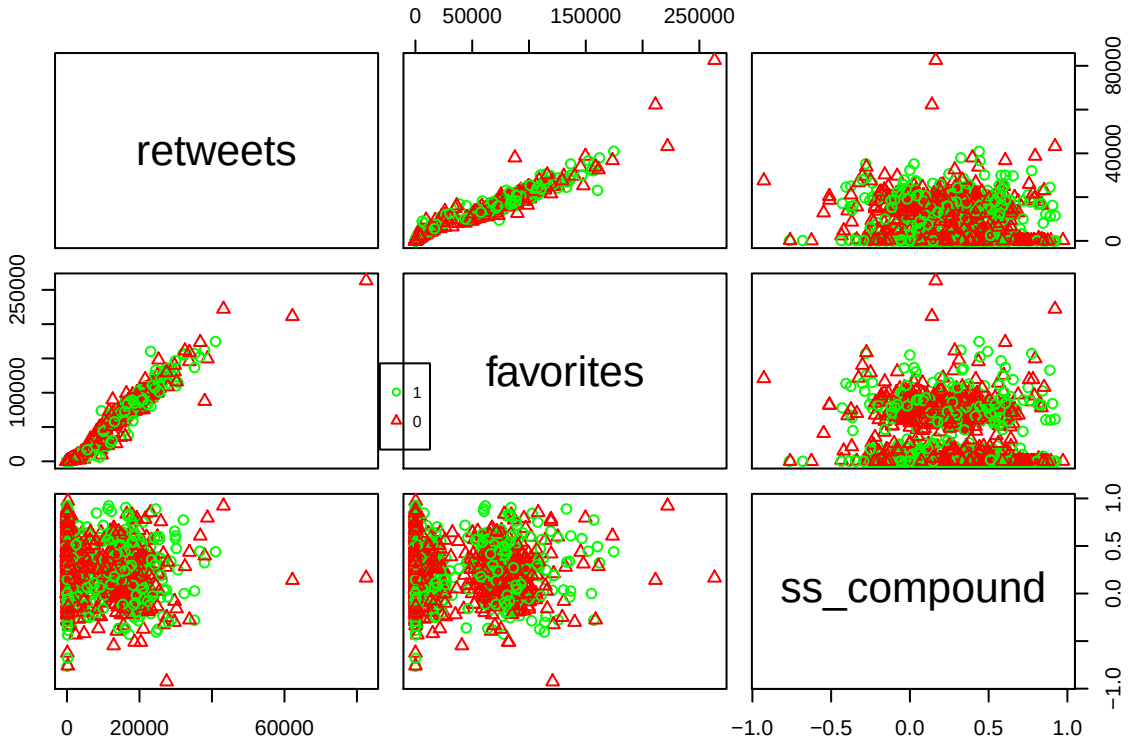
s&p: "delta1" equals to 0 means compared to the previous day the Adj.close is lower, and 1 otherwise.

"mc" equals to 0 means compared to ten days before Adj.close fluctuated less than 5%, and 1 otherwise.

### Fit logistic model

Stock price rise or fall in the next day

```
##### lag = 1
pairs(train[, select_variables], col=c("green","red")[train$delta_day1],
      pch=c(1,2)[train$delta_day1])

par(xpd=TRUE)
legend(0.34, 0.51, as.vector(unique(train$delta_day1)),
       col=c("green", "red"), pch=1:3, cex = 0.5)
```



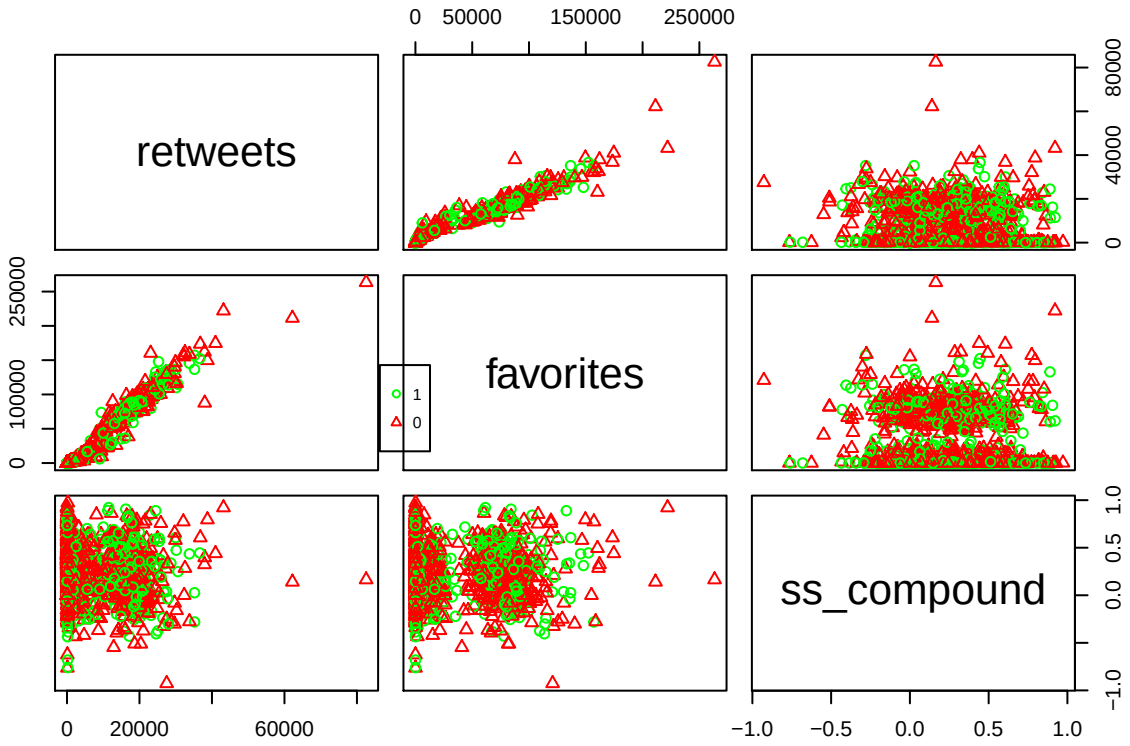

```
#0 stock price lower than previous day


mod_log1 = glm(delta_day1 ~ retweets + favorites + ss_compound,
               data = train, family = binomial)
summary(mod_log1)
```

```
##
## Call:
## glm(formula = delta_day1 ~ retweets + favorites + ss_compound,
##     family = binomial, data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.363  -1.241   1.072   1.113   1.334
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.226e-01  7.909e-02   2.814  0.00489 **
## retweets    -2.356e-05  3.192e-05  -0.738  0.46036
## favorites    5.333e-06  7.109e-06   0.750  0.45313
## ss_compound -2.687e-01  1.972e-01  -1.363  0.17295
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 2192.1  on 1587  degrees of freedom
## Residual deviance: 2189.9  on 1584  degrees of freedom
## AIC: 2197.9
##
## Number of Fisher Scoring iterations: 3
```

Stock price rise or fall in the next two days

```
##### lag = 2
pairs(train[, select_variables], col=c("green","red")[train$delta_day2],
      pch=c(1,2)[train$delta_day2])

par(xpd=TRUE)
legend(0.34, 0.51, as.vector(unique(train$delta_day2)),
       col=c("green", "red"), pch=1:3, cex = 0.5)
```



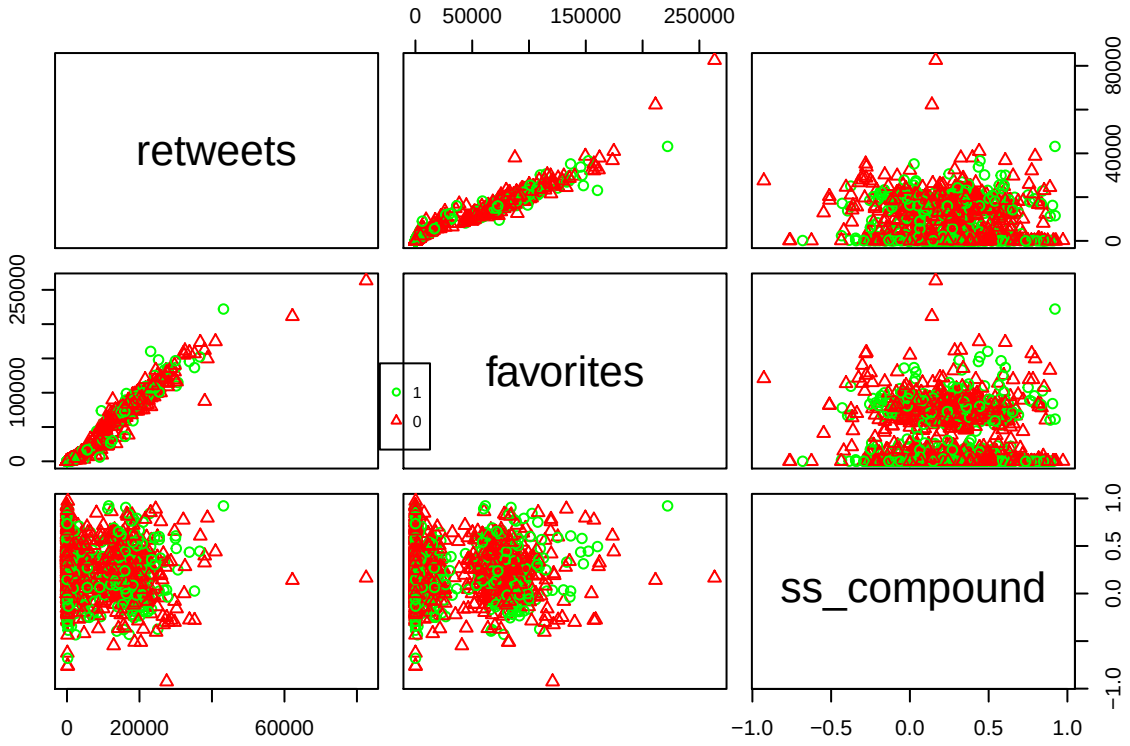

```
#0 stock price lower than previous day


mod_log2 = glm(delta_day2 ~ retweets + favorites + ss_compound,
               data = train, family = binomial)
summary(mod_log2)
```

```
##
```

```
## Call:
## glm(formula = delta_day2 ~ retweets + favorites + ss_compound,
##     family = binomial, data = train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.403  -1.312   1.012   1.048   1.064
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.976e-01  7.979e-02   3.730 0.000191 ***
## retweets     7.619e-06  3.272e-05   0.233 0.815846
## favorites   -4.906e-07  7.274e-06  -0.067 0.946225
## ss_compound  3.472e-02  1.993e-01   0.174 0.861672
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2156.0  on 1587  degrees of freedom
## Residual deviance: 2155.1  on 1584  degrees of freedom
## AIC: 2163.1
##
## Number of Fisher Scoring iterations: 4
```

Stock price rise or fall in the next five days

```
##### lag = 5
pairs(train[, select_variables], col=c("green","red")[train$delta_day5],
      pch=c(1,2)[train$delta_day5])

par(xpd=TRUE)
legend(0.34, 0.51, as.vector(unique(train$delta_day5)),
       col=c("green", "red"), pch=1:3, cex = 0.5)
```



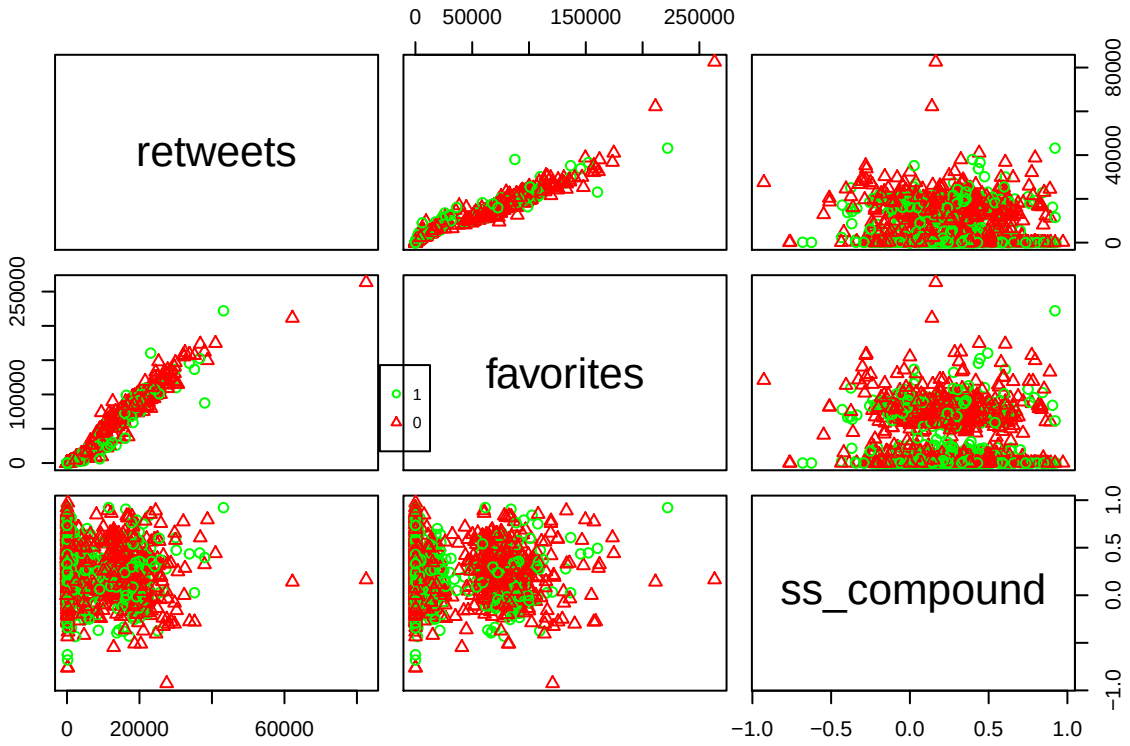

```
#0 stock price lower than previous day


mod_log5 = glm(delta_day5 ~ retweets + favorites + ss_compound,
               data = train, family = binomial)
summary(mod_log5)
```

```
##
## Call:
## glm(formula = delta_day5 ~ retweets + favorites + ss_compound,
##     family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6463  -1.2868   0.9507   1.0690   1.1275
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  2.305e-01  8.112e-02   2.841   0.0045 **
## retweets     6.329e-05  3.680e-05   1.720   0.0855 .
## favorites   -1.025e-05  8.060e-06  -1.272   0.2033
## ss_compound  4.063e-02  2.011e-01   0.202   0.8399
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 2150.4  on 1587  degrees of freedom
## Residual deviance: 2139.9  on 1584  degrees of freedom
## AIC: 2147.9
##
## Number of Fisher Scoring iterations: 4
```

Stock price rise or fall in the next ten days

```
##### lag = 10
pairs(train[, select_variables], col=c("green","red")[train$delta_day10],
      pch=c(1,2)[train$delta_day10])

par(xpd=TRUE)
legend(0.34, 0.51, as.vector(unique(train$delta_day10)),
       col=c("green", "red"), pch=1:3, cex = 0.5)
```



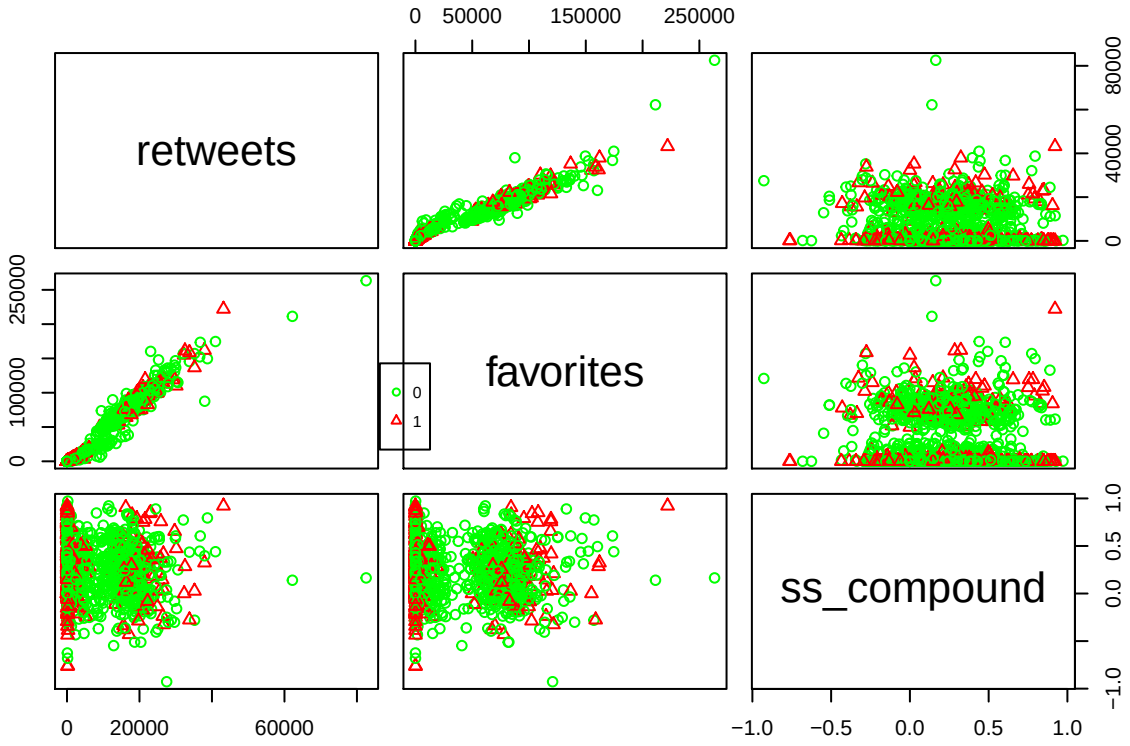

```
#0 stock price lower than previous day


mod_log10 = glm(delta_day10 ~ retweets + favorites + ss_compound,
                data = train, family = binomial)
summary(mod_log10)
```

```
##
```

```
## Call:
## glm(formula = delta_day10 ~ retweets + favorites + ss_compound,
##     family = binomial, data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.0959  -1.3296   0.8773   1.0185   1.2025
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept)  3.276e-01  8.048e-02   4.071 4.69e-05 ***
## retweets    -6.731e-05  3.356e-05  -2.006  0.04486 *
## favorites    1.988e-05  7.563e-06   2.629  0.00857 **
## ss_compound  2.389e-01  2.044e-01   1.169  0.24248
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2110.6  on 1587  degrees of freedom
## Residual deviance: 2090.1  on 1584  degrees of freedom
## AIC: 2098.1
##
## Number of Fisher Scoring iterations: 4
```

Stock price has a massive change (>3%) in ten days

```
##### lag = 10
pairs(train[, select_variables], col=c("green","red")[train$mc3],
      pch=c(1,2)[train$mc3])

par(xpd=TRUE)
legend(0.34, 0.51, as.vector(unique(train$mc3)),
       col=c("green", "red"), pch=1:3, cex = 0.5)
```



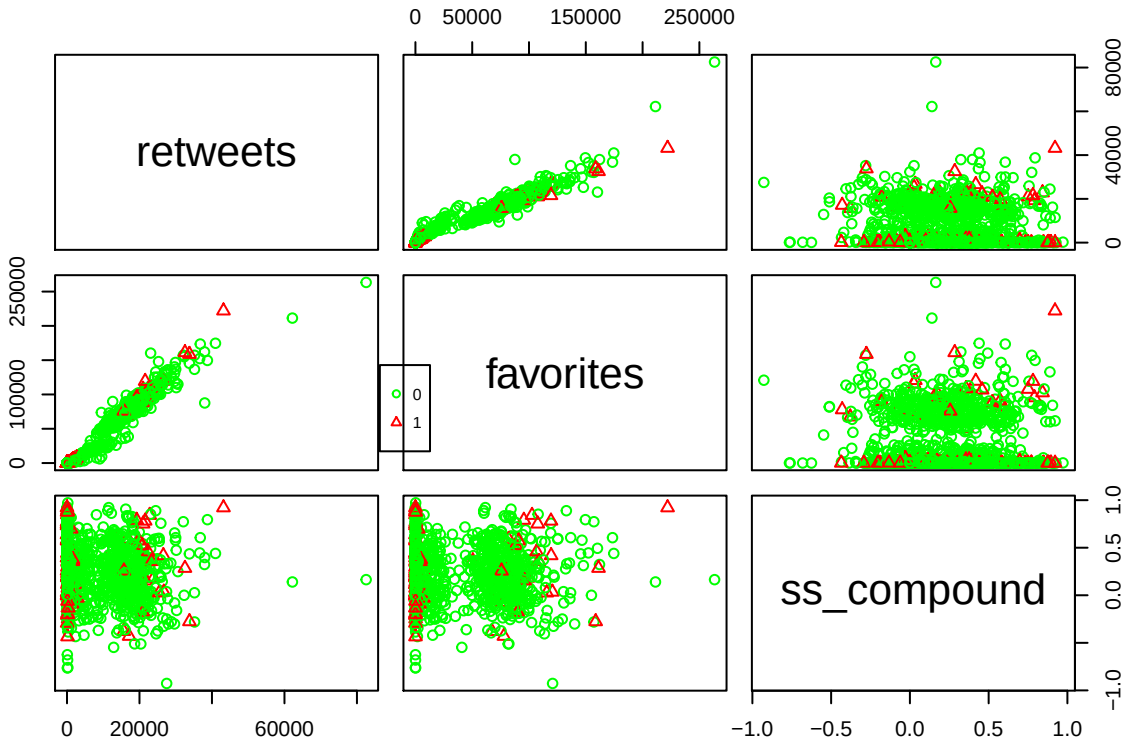

```
#0 stock price lower than previous day


mod_logmc3 = glm(mc3 ~ retweets + favorites + ss_compound,
                 data = train, family = binomial)
summary(mod_logmc3)
```

```
##
## Call:
## glm(formula = mc3 ~ retweets + favorites + ss_compound, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.8908  -0.7221  -0.7051  -0.6380   1.9493
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -1.201e+00  9.607e-02 -12.502   <2e-16 ***
## retweets    -6.048e-05  4.422e-05  -1.368    0.171
## favorites    1.175e-05  9.646e-06   1.218    0.223
## ss_compound -1.057e-02  2.384e-01  -0.044    0.965
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
##
##     Null deviance: 1672.5  on 1587  degrees of freedom
## Residual deviance: 1669.7  on 1584  degrees of freedom
## AIC: 1677.7
##
## Number of Fisher Scoring iterations: 4
```

Stock price has a massive change (>5%) in ten days

```
##### lag = 10
pairs(train[, select_variables], col=c("green","red")[train$mc5],
      pch=c(1,2)[train$mc5])

par(xpd=TRUE)
legend(0.34, 0.51, as.vector(unique(train$mc5)),
       col=c("green", "red"), pch=1:3, cex = 0.5)
```




```
#0 stock price lower than previous day


mod_logmc5 = glm(mc5 ~ retweets + favorites + ss_compound,
                 data = train, family = binomial)
summary(mod_logmc5)
```

```
##
```

```
## Call:
## glm(formula = mc5 ~ retweets + favorites + ss_compound, family = binomial,
##     data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9532  -0.3733  -0.3619  -0.3394   2.4816
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.690e+00  1.680e-01 -16.014   <2e-16 ***
## retweets    -1.703e-04  7.916e-05  -2.152   0.0314 *
## favorites    3.726e-05  1.669e-05   2.233   0.0256 *
## ss_compound  1.938e-01  4.046e-01   0.479   0.6319
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 751.97  on 1587  degrees of freedom
## Residual deviance: 745.84  on 1584  degrees of freedom
## AIC: 753.84
##
## Number of Fisher Scoring iterations: 5
```

### Prediction

I chose mc $> 5\%$ as a predictor to fit our test data.

```
##Prediction using logistic regression
pred = predict(mod_logmc5, test[, select_variables])
head(pred)
```

```
##         3         5         7        17        25        26
## -2.628060 -2.545862 -2.546160 -2.697839 -2.691414 -2.693613
```

```
predProbs = binomial()$linkinv(pred)
pred_log = rep("Decrease", nrow(test))


pred_log[predProbs > .5] = "Increase"
table(pred_log, test$mc5)
```

```
##
## pred_log     0   1
##   Decrease 639  42
```

```
err_log = sum(pred_log != test$mc5) / nrow(test)
err_log
```

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

Although some models may have significant variables which seem to fit well, the prediction is really bad.

## Kmeans Clustering

Since we can see there is a obvious boundary in the pairwise plots, and it is difficult to find a label to classify if the stock price will rise or fall so I tried to use kmeans clustering to classify data into two groups.

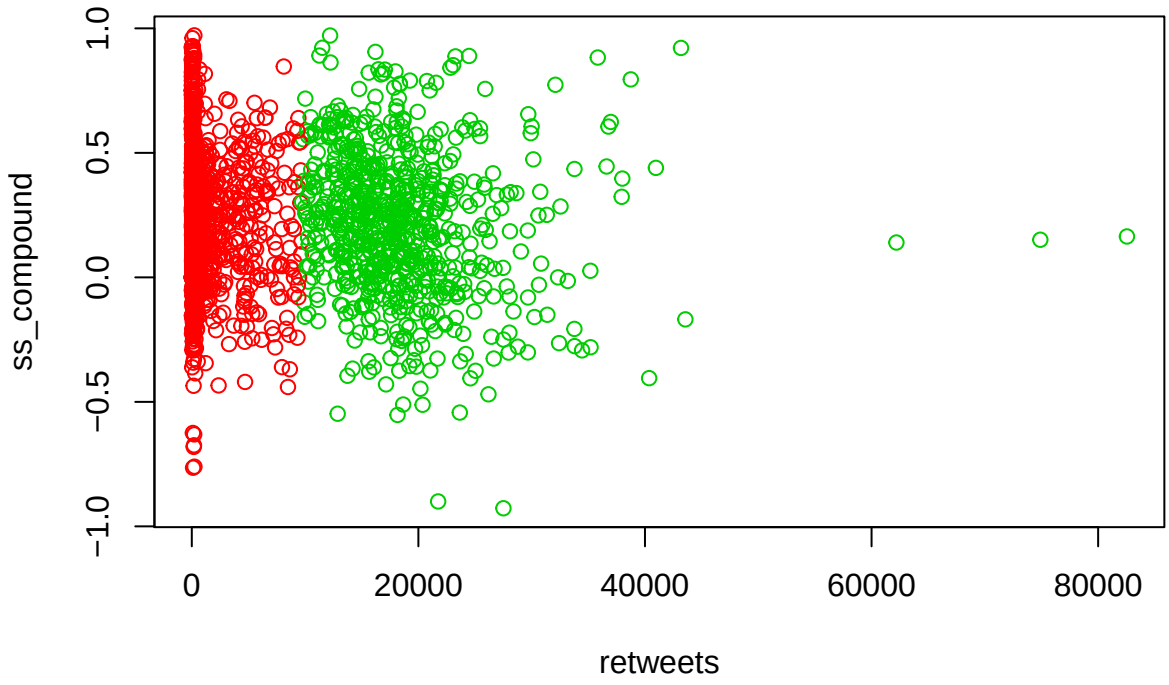
```
library(dplyr)

tweet = read.csv("/Users/Sabrina/Documents/2019UMICH/STATS503/project/data/grouped_date_new.csv")
sp = read.csv("/Users/Sabrina/Documents/2019UMICH/STATS503/project/data/sp_indicator.csv")

tweet =
  tweet %>%
  rename(Date = date)

join =
  tweet %>%
  left_join(sp, by = "Date")
```

```
## Warning: Column `Date` joining factors with different levels, coercing to
## character vector
```

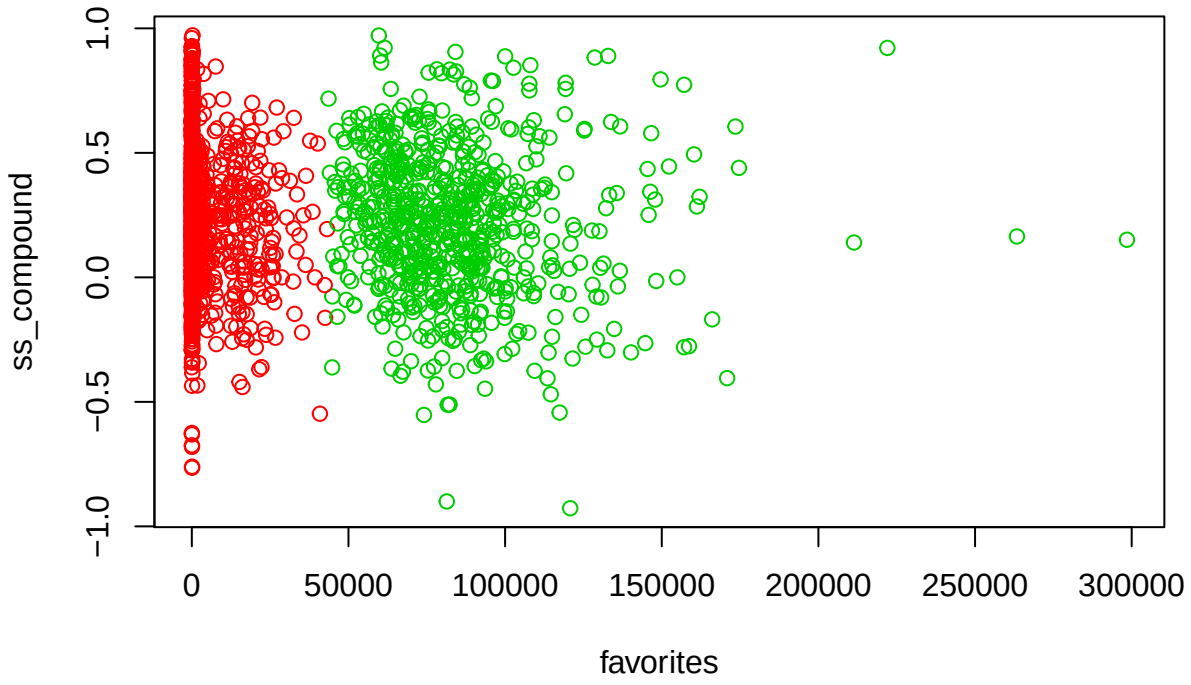
```
join_rmna = join[complete.cases(join), ]

fav = join_rmna$favorites
retweets = join_rmna$retweets
ss = join_rmna$ss_compound

nrow(join_rmna)
```

```
## [1] 2269
```

```
x = matrix(c(retweets, ss),nrow = nrow(join_rmna))

y = matrix(c(fav, ss),nrow = nrow(join_rmna))

km.outx = kmeans(x, 2)
km.outy = kmeans(y, 2)

plot(x, col = (km.outx$cluster+1), xlab = "retweets", ylab = "ss_compound")
```




```
plot(y, col = (km.outy$cluster+1), xlab = "favorites", ylab = "ss_compound")
```




Although the data can be clearly seperated into two groups, the ss_compound seems to be a really bad predictors to predict fluctuation of stock price.