NTFX

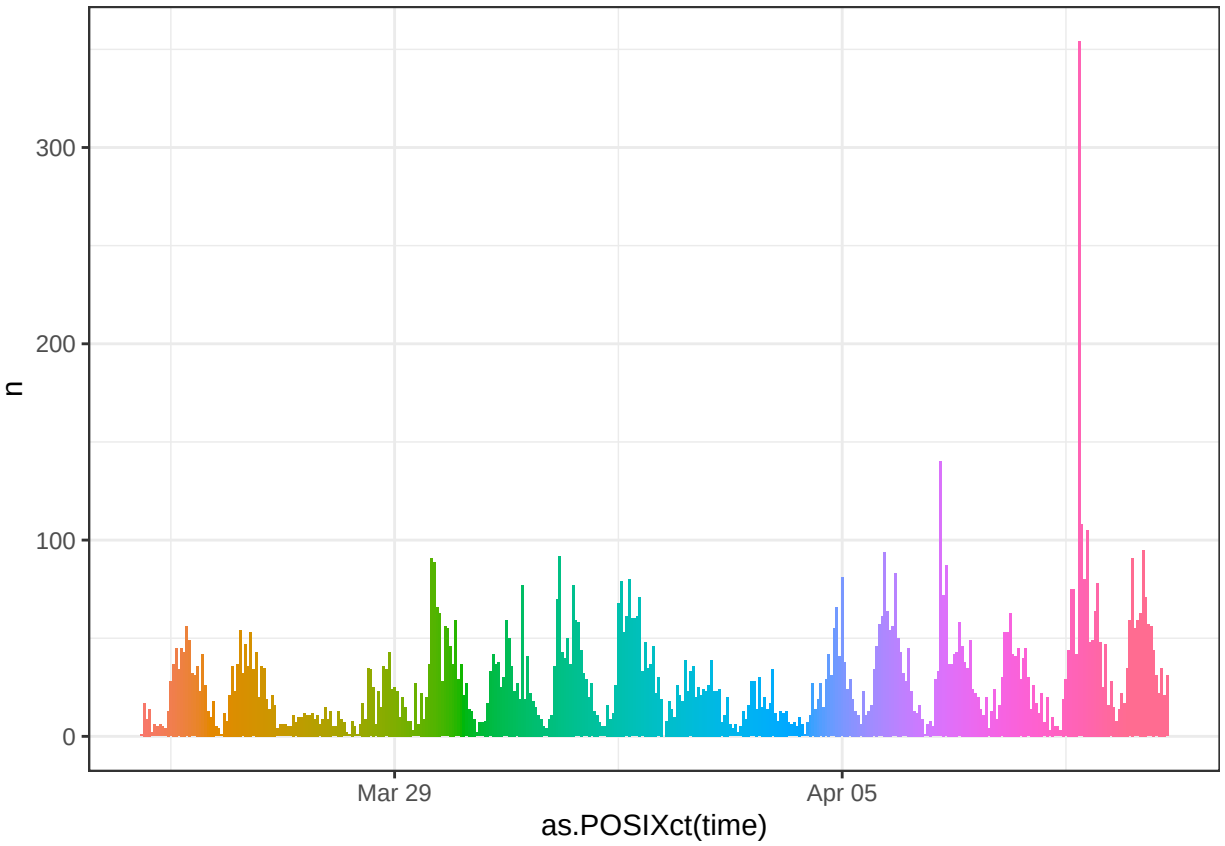
Evan Day

2023-05-08

NFLX

Read Text file and Text Cleanning

The following table shows the tweet number per hour with a barplot.



paste all the text together group by hour, the following table shows an example of the text dataframe.

```
## # A tibble: 6 x 3
## # Groups:   date [1]
```

```
##   date       time                text
##   <date>     <chr>               <chr>
## 1 2021-03-25 2021-03-25 01:00:00 " Top tweeted stocks TSLA BA GME AMZN INTC SE ~
## 2 2021-03-25 2021-03-25 02:00:00 " Current bearish engulfing monthly candles as~
## 3 2021-03-25 2021-03-25 03:00:00 " Favorite sportscard from Griffey just beauti~
## 4 2021-03-25 2021-03-25 04:00:00 " Is FUBO really the future ROKU Market believ~
## 5 2021-03-25 2021-03-25 05:00:00 " Current bearish engulfing monthly candles as~
## 6 2021-03-25 2021-03-25 06:00:00 " NFLX Let s see if we can hold going forward ~

## [1] "there are total 385 observation"
```

Sentiment Data frame with bing, afinn, and nrc

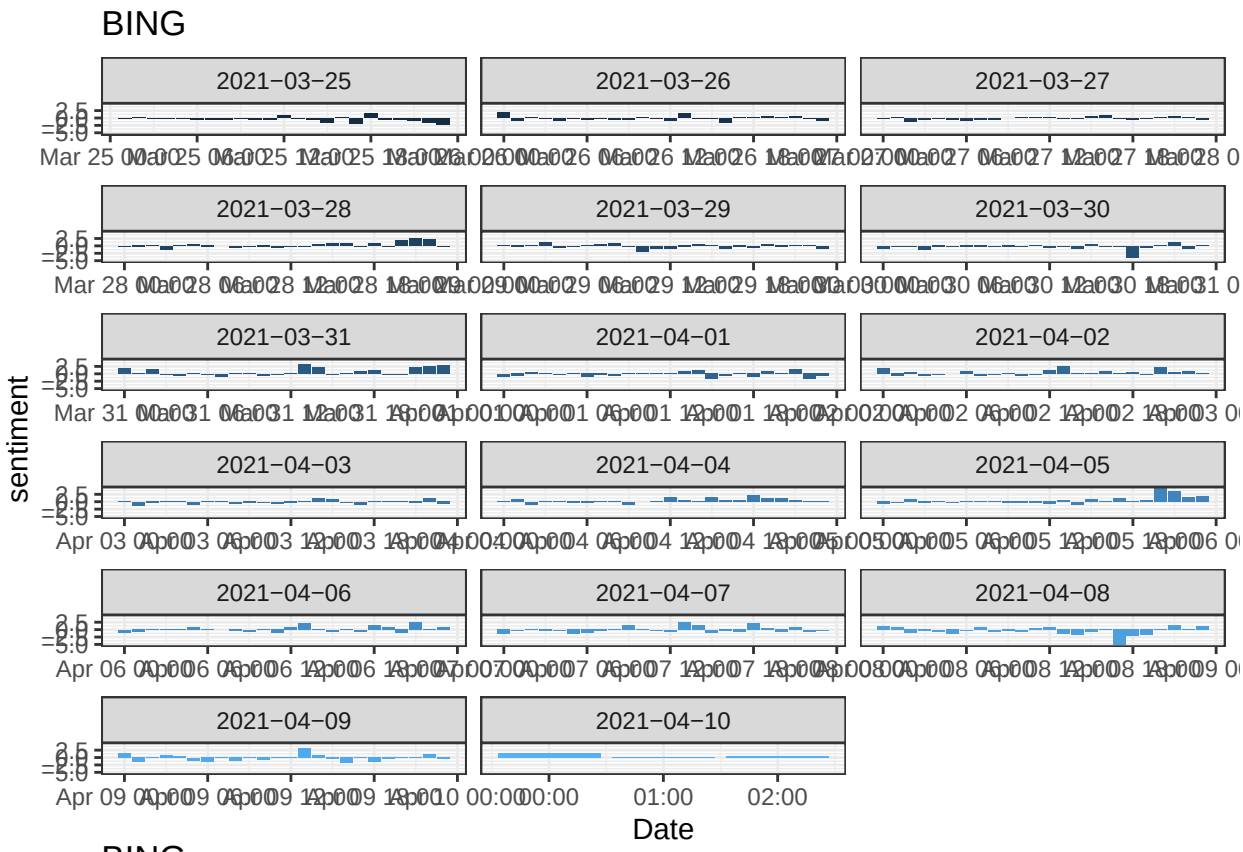
We start with the bing data frame

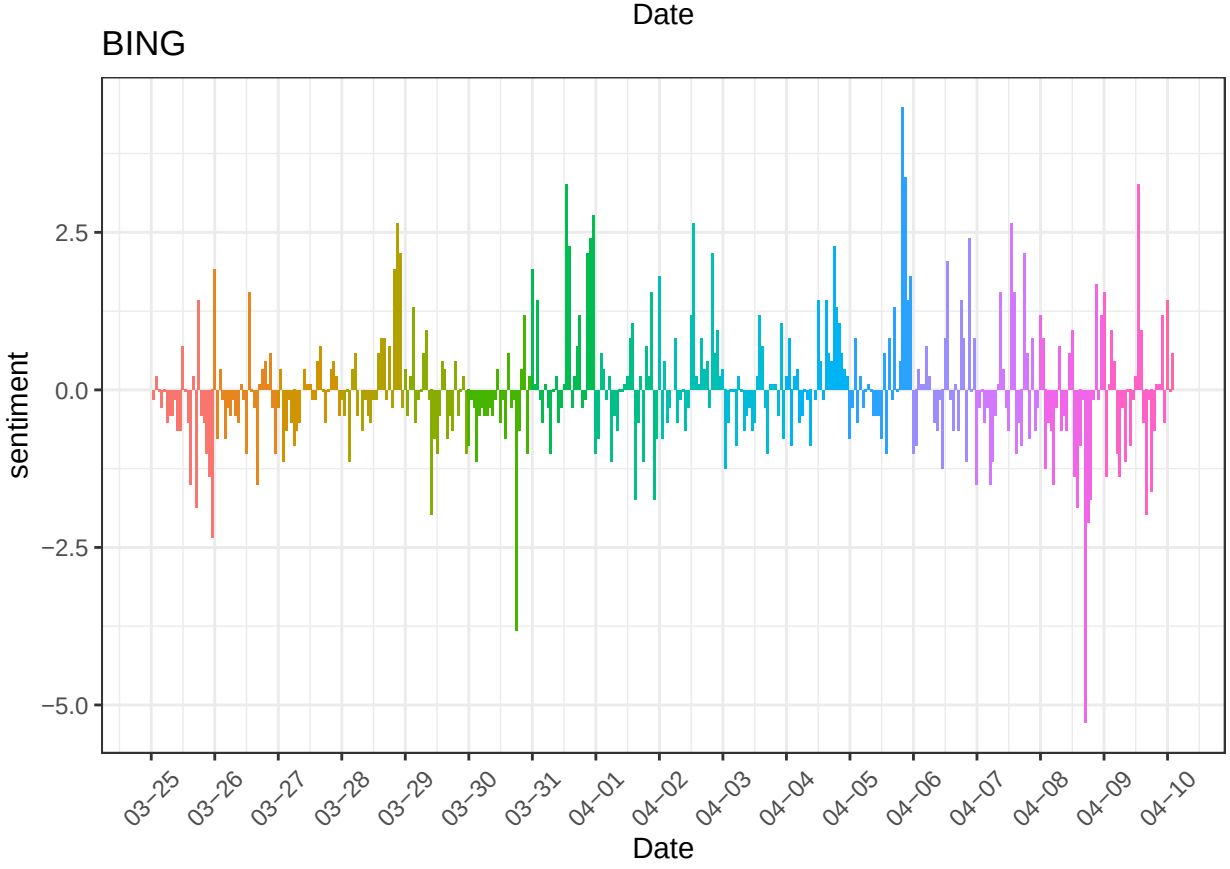
```
## # A tibble: 6 x 3
## # Groups:   date [1]
##   date       time                sentiment
##   <date>     <chr>                   <dbl>
## 1 2021-03-25 2021-03-25 01:00:00         1
## 2 2021-03-25 2021-03-25 02:00:00         4
## 3 2021-03-25 2021-03-25 03:00:00         2
## 4 2021-03-25 2021-03-25 04:00:00         0
## 5 2021-03-25 2021-03-25 05:00:00         2
## 6 2021-03-25 2021-03-25 06:00:00        -2
```

then, we normalize the sentiment, normalized data has mean = 0 // aother way is rescale to c(-3,3)

```
## # A tibble: 6 x 3
## # Groups:   date [1]
##   date       time                sentiment
##   <date>     <chr>                   <dbl>
## 1 2021-03-25 2021-03-25 01:00:00   -0.157
## 2 2021-03-25 2021-03-25 02:00:00    0.209
## 3 2021-03-25 2021-03-25 03:00:00   -0.0352
## 4 2021-03-25 2021-03-25 04:00:00   -0.279
## 5 2021-03-25 2021-03-25 05:00:00   -0.0352
## 6 2021-03-25 2021-03-25 06:00:00   -0.523
```

and then, we plot the normalized sentiment against the time.

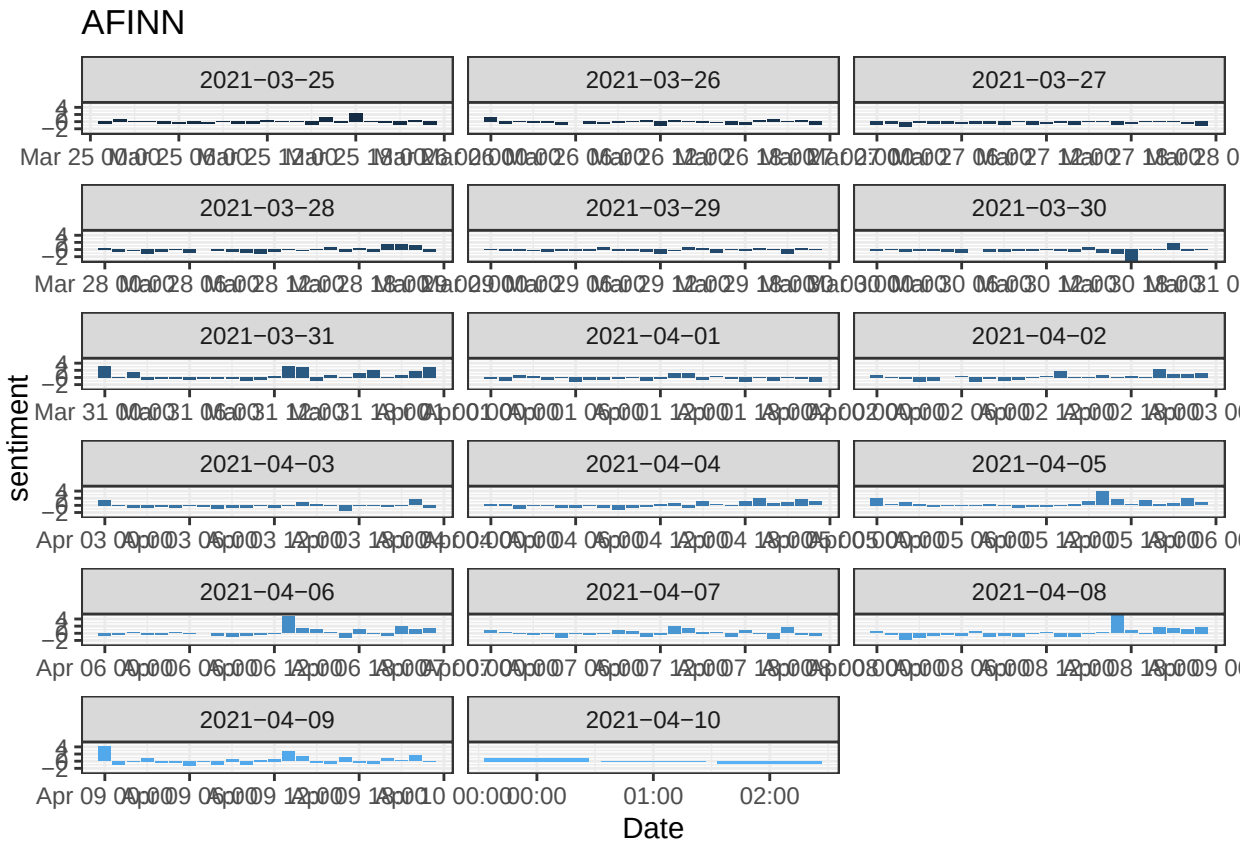


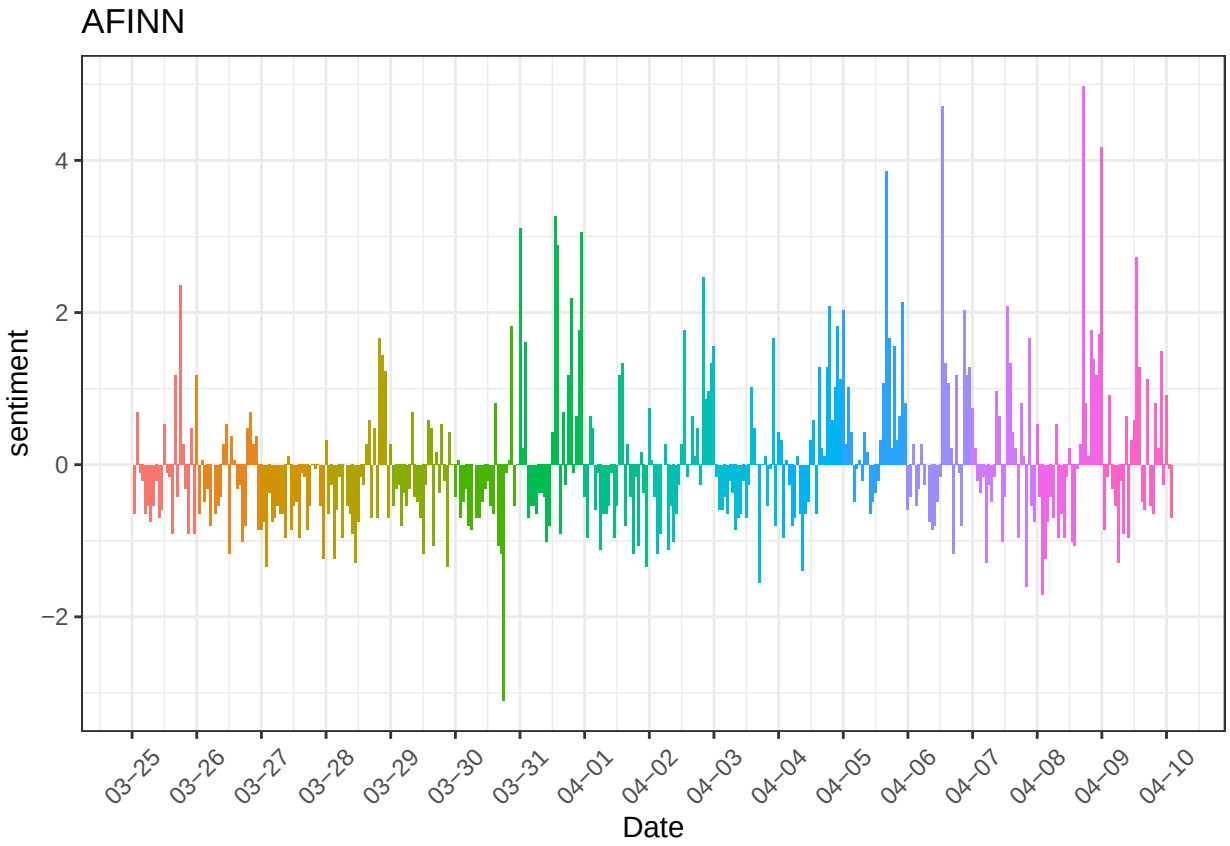


And then, we deal with the afinn sentiment dataframe

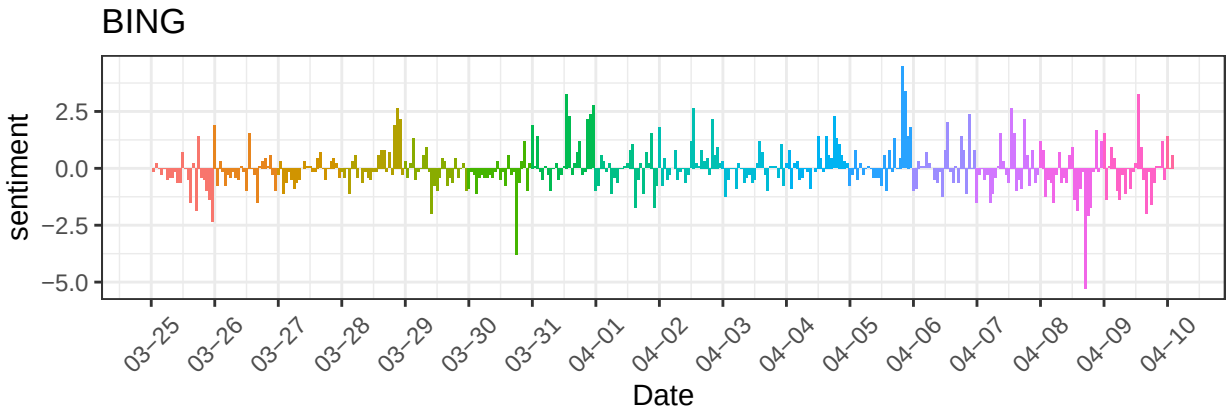
```
## # A tibble: 6 x 3
## # Groups:   date [1]
##   date       time                sentiment
##   <date>     <chr>                   <dbl>
## 1 2021-03-25 2021-03-25 01:00:00    -0.642
## 2 2021-03-25 2021-03-25 02:00:00     0.695
## 3 2021-03-25 2021-03-25 03:00:00    -0.107
## 4 2021-03-25 2021-03-25 04:00:00    -0.214
## 5 2021-03-25 2021-03-25 05:00:00    -0.642
## 6 2021-03-25 2021-03-25 06:00:00    -0.535
```

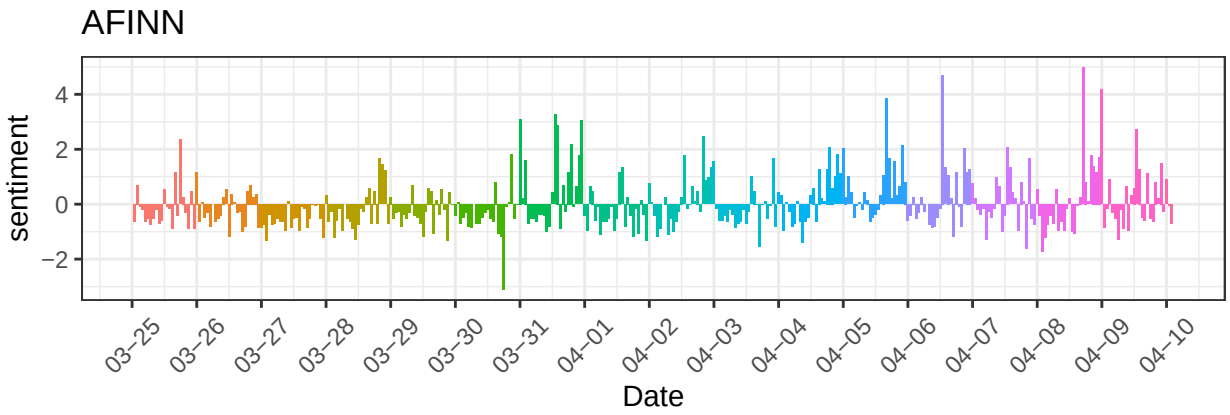
and then, we plot the normalized sentiment against the time. // Aother method is rescale to c(-3,3)





we compare the two sentiment plot together

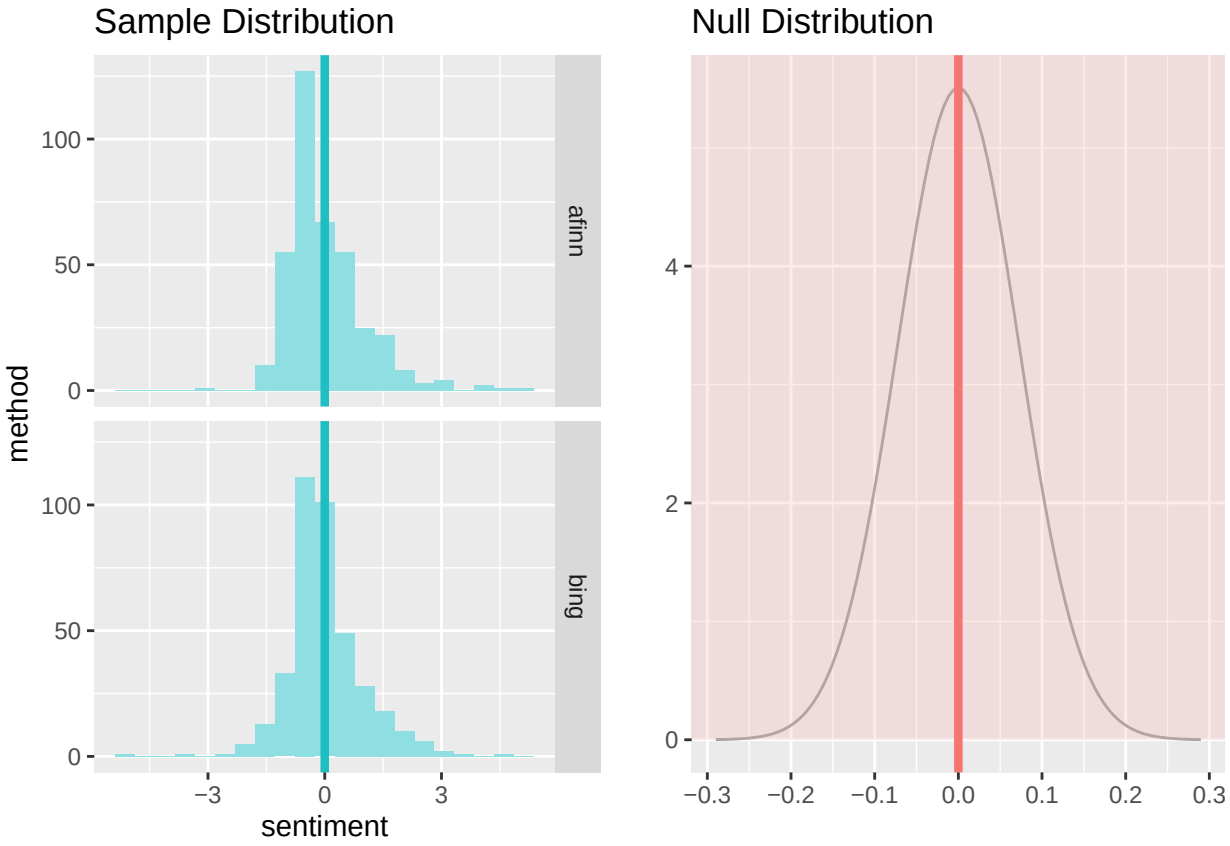




using t-test to check the whether there is a difference between bing lexicon and afinn lexicon, however the distriibution must be similar. (this is meaningless, because we have already normalize the data, the distributio will be almost the same

```
## Loading required package: BayesFactor

## Loading required package: coda

## Loading required package: Matrix

##
## Attaching package: 'Matrix'

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

## ************
## Welcome to BayesFactor 0.9.12-4.3. If you have questions, please contact Richard Morey (richarddmorey
##
## Type BFManual() to open the manual.
## ************

## Warning: Missing null value, set to 0
```

```
## Response variable: numerical
## Explanatory variable: categorical (2 levels)
## n_afinn = 381, y_bar_afinn = 0, s_afinn = 1
## n_bing = 381, y_bar_bing = 0, s_bing = 1
## H0: mu_afinn =  mu_bing
## HA: mu_afinn != mu_bing
## t = 0, df = 380
## p_value = 1
```



we should use the KS-test to check the distribution: as a result, reject the null h0, the distribution are different.

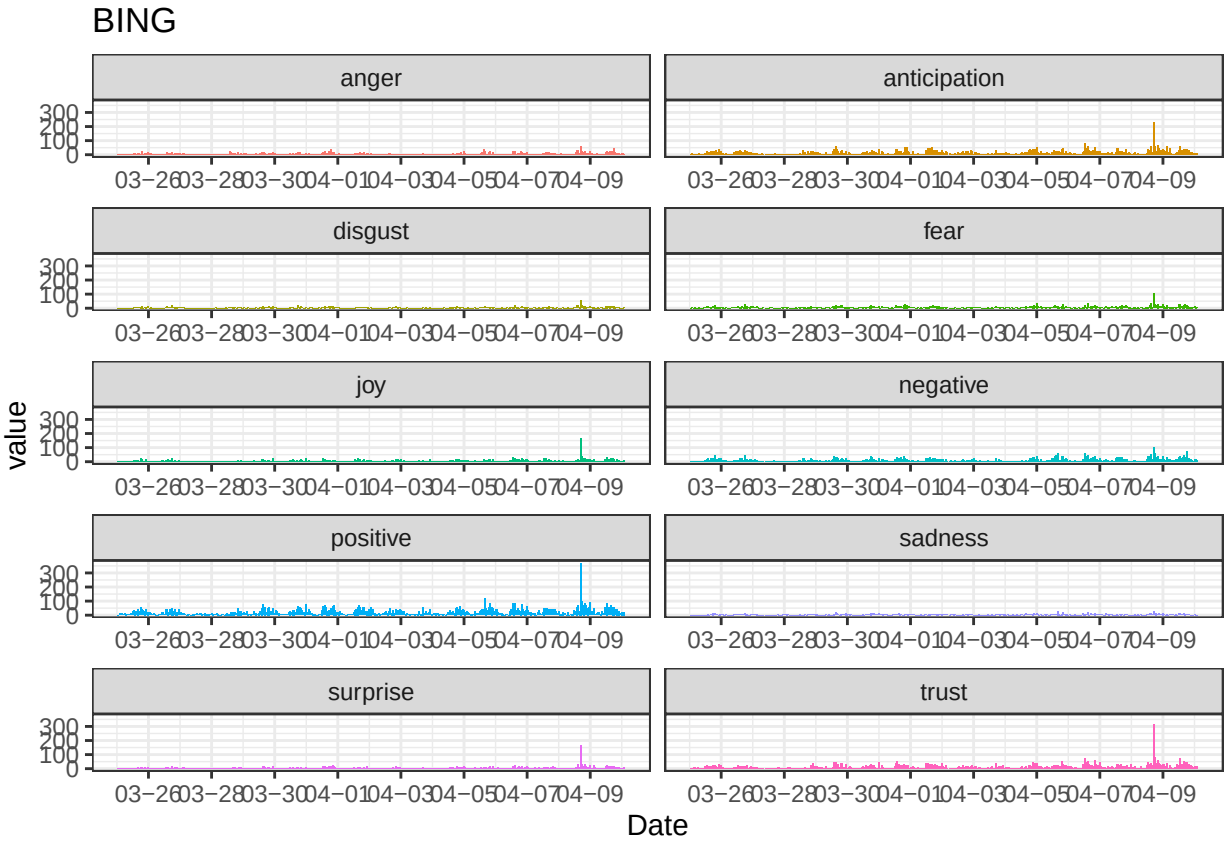
```
## Warning in ks.test(bing_afinn$bing, bing_afinn$afinn, alternative =
## "two.sided"): p-value will be approximate in the presence of ties
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  bing_afinn$bing and bing_afinn$afinn
## D = 0.13312, p-value = 0.00238
## alternative hypothesis: two-sided
```

Then, here is the method with nrc lexicon

```
## # A tibble: 6 x 12
## # Groups:   date [1]
```

```
##   date       time       anger anticipation disgust  fear   joy negative positive
##   <date>     <chr>      <dbl>        <dbl>   <dbl> <dbl> <dbl>    <dbl>    <dbl>
## 1 2021-03-25 2021-03-2~     0            1       0     0     0        1        1
## 2 2021-03-25 2021-03-2~     2            8       0     4     3        4       12
## 3 2021-03-25 2021-03-2~     1            4       0     1     4        2       12
## 4 2021-03-25 2021-03-2~     5            5       1     4     5        6       14
## 5 2021-03-25 2021-03-2~     1            2       0     1     1        1        4
## 6 2021-03-25 2021-03-2~     2            4       1     2     4        2       13
## # ... with 3 more variables: sadness <dbl>, surprise <dbl>, trust <dbl>

##
## Attaching package: 'reshape2'

## The following object is masked from 'package:tidyr':
##
##     smiths

## No id variables; using all as measure variables
```



NFLX

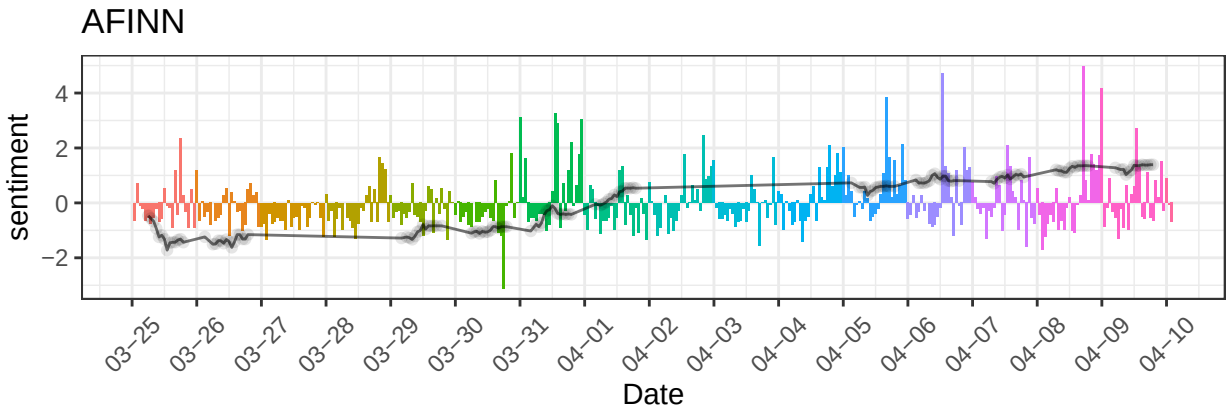
Stock Information

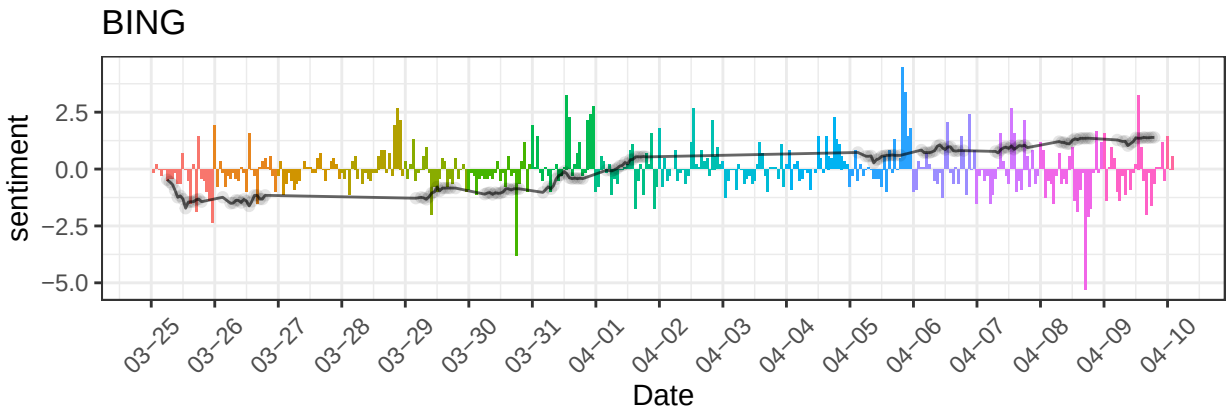
```
## # A tibble: 6 x 2
```

```
##   time                price
##   <chr>               <dbl>
## 1 2021-03-25 06:00:00  521.
## 2 2021-03-25 07:00:00  519
## 3 2021-03-25 08:00:00  517
## 4 2021-03-25 09:00:00  512.
## 5 2021-03-25 10:00:00  506.
## 6 2021-03-25 11:00:00  508.
```

normalize the price data:

```
## # A tibble: 6 x 2
##   time                price
##   <chr>               <dbl>
## 1 2021-03-25 06:00:00 -0.468
## 2 2021-03-25 07:00:00 -0.565
## 3 2021-03-25 08:00:00 -0.673
## 4 2021-03-25 09:00:00 -0.933
## 5 2021-03-25 10:00:00 -1.24
## 6 2021-03-25 11:00:00 -1.18
```





2. Build the model dataframe:

```
## Joining, by = c("datetime", "date")
```

Here we need to deal with several questions: 1. Stock maket open at 9 am and close at 4 pm 2. At the open time, stock market record the XX:30, which is not consistent with sentimennt XX::00 3. At close time, stock market also record some stock price

Separate the dataframe into close data_frame and open data_frame

```
## # A tibble: 6 x 15
##   datetime             price date       time_stock anger anticipation disgust
##   <dttm>               <dbl> <date>     <chr>      <dbl>        <dbl>   <dbl>
## 1 2021-03-25 06:00:00 -0.468 2021-03-25 06:00          2            4       1
## 2 2021-03-25 07:00:00 -0.565 2021-03-25 07:00          2            8       2
## 3 2021-03-25 08:00:00 -0.673 2021-03-25 08:00          1            4       0
## 4 2021-03-25 17:00:00 -1.35  2021-03-25 17:00          9           13       3
## 5 2021-03-25 18:00:00 -1.32  2021-03-25 18:00          7           27       8
## 6 2021-03-25 19:00:00 -1.43  2021-03-25 19:00         25           24      15
## # ... with 8 more variables: fear <dbl>, joy <dbl>, negative <dbl>,
## #   positive <dbl>, sadness <dbl>, surprise <dbl>, trust <dbl>, state <chr>
```

```
## # A tibble: 6 x 15
##   datetime             price date       time_stock anger anticipation disgust
##   <dttm>               <dbl> <date>     <chr>      <dbl>        <dbl>   <dbl>
## 1 2021-03-25 09:00:00 -0.933 2021-03-25 09:00          1            3       0
## 2 2021-03-25 10:00:00 -1.24  2021-03-25 10:00          0            2       0
## 3 2021-03-25 11:00:00 -1.18  2021-03-25 11:00          0            3       0
## 4 2021-03-25 12:00:00 -1.32  2021-03-25 12:00          4           13       2
## 5 2021-03-25 13:00:00 -1.72  2021-03-25 13:00         11           26       3
## 6 2021-03-25 14:00:00 -1.45  2021-03-25 14:00         10           21       3
## # ... with 8 more variables: fear <dbl>, joy <dbl>, negative <dbl>,
## #   positive <dbl>, sadness <dbl>, surprise <dbl>, trust <dbl>, state <chr>
```

NFLX NRC Regression Model result

1. this is the model for total recording

```
##
## Call:
## lm(formula = price ~ anger + anticipation + disgust + fear +
##     joy + negative + positive + sadness + surprise + trust, data = full_nrc)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7427 -0.8951  0.1454  0.8990  1.6875
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -8.934e-16  7.655e-02   0.000   1.0000
## anger        -1.442e-01  2.536e-01  -0.569   0.5705
## anticipation  1.116e-01  2.807e-01   0.398   0.6915
## disgust       1.052e-01  2.289e-01   0.460   0.6465
## fear         -8.316e-02  2.588e-01  -0.321   0.7485
## joy          -6.757e-01  3.393e-01  -1.992   0.0483 *
```

```
## negative      4.413e-01  3.051e-01   1.446   0.1502
## positive      5.690e-02  3.482e-01   0.163   0.8704
## sadness      -2.256e-01  1.587e-01  -1.421   0.1574
## surprise      3.187e-01  3.508e-01   0.908   0.3652
## trust         3.490e-01  3.785e-01   0.922   0.3580
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9591 on 146 degrees of freedom
## Multiple R-squared:  0.1391, Adjusted R-squared:  0.08009
## F-statistic: 2.358 on 10 and 146 DF,  p-value: 0.01298
```

```
## randomForest 4.7-1
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:gridExtra':
##
##     combine
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

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

```
##
## Attaching package: 'xgboost'
```

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

```
## [1]  train-rmse:0.884653
## [2]  train-rmse:0.724726
## [3]  train-rmse:0.620232
## [4]  train-rmse:0.531058
## [5]  train-rmse:0.456655
## [6]  train-rmse:0.400430
## [7]  train-rmse:0.358717
## [8]  train-rmse:0.310605
## [9]  train-rmse:0.272931
## [10] train-rmse:0.250551
## [11] train-rmse:0.228353
## [12] train-rmse:0.210125
## [13] train-rmse:0.197036
## [14] train-rmse:0.181076
```

```
## [15] train-rmse:0.160363
## [16] train-rmse:0.151668
## [17] train-rmse:0.146564
## [18] train-rmse:0.134749
## [19] train-rmse:0.123570
## [20] train-rmse:0.120217
## [21] train-rmse:0.105224
## [22] train-rmse:0.093517
## [23] train-rmse:0.089135
## [24] train-rmse:0.080663
## [25] train-rmse:0.077468
## [26] train-rmse:0.069614
## [27] train-rmse:0.065601
## [28] train-rmse:0.056948
## [29] train-rmse:0.054400
## [30] train-rmse:0.048094
## [31] train-rmse:0.044792
## [32] train-rmse:0.043175
## [33] train-rmse:0.040942
## [34] train-rmse:0.036322
## [35] train-rmse:0.033292
## [36] train-rmse:0.032390
## [37] train-rmse:0.029714
## [38] train-rmse:0.027554
## [39] train-rmse:0.026756
## [40] train-rmse:0.024802
## [41] train-rmse:0.021684
## [42] train-rmse:0.020324
## [43] train-rmse:0.017862
## [44] train-rmse:0.016321
## [45] train-rmse:0.014757
## [46] train-rmse:0.013953
## [47] train-rmse:0.013374
## [48] train-rmse:0.011679
## [49] train-rmse:0.011022
## [50] train-rmse:0.010039
## [51] train-rmse:0.009026
## [52] train-rmse:0.008288
## [53] train-rmse:0.007815
## [54] train-rmse:0.007179
## [55] train-rmse:0.006859
## [56] train-rmse:0.006548
## [57] train-rmse:0.006133
## [58] train-rmse:0.005890
## [59] train-rmse:0.005428
## [60] train-rmse:0.005043
## [61] train-rmse:0.004594
## [62] train-rmse:0.004312
## [63] train-rmse:0.003915
## [64] train-rmse:0.003620
## [65] train-rmse:0.003327
## [66] train-rmse:0.003235
## [67] train-rmse:0.002966
## [68] train-rmse:0.002771
```

```
## [69] train-rmse:0.002497
## [70] train-rmse:0.002317
## [71] train-rmse:0.002206
## [72] train-rmse:0.001993
## [73] train-rmse:0.001805
## [74] train-rmse:0.001654
## [75] train-rmse:0.001478
## [76] train-rmse:0.001432
## [77] train-rmse:0.001299
## [78] train-rmse:0.001240
## [79] train-rmse:0.001240
## [80] train-rmse:0.001240
## [81] train-rmse:0.001240
## [82] train-rmse:0.001240
## [83] train-rmse:0.001240
## [84] train-rmse:0.001240
## [85] train-rmse:0.001240
## [86] train-rmse:0.001240
## [87] train-rmse:0.001240
## [88] train-rmse:0.001240
## [89] train-rmse:0.001240
## [90] train-rmse:0.001240
## [91] train-rmse:0.001240
## [92] train-rmse:0.001240
## [93] train-rmse:0.001240
## [94] train-rmse:0.001240
## [95] train-rmse:0.001240
## [96] train-rmse:0.001240
## [97] train-rmse:0.001240
## [98] train-rmse:0.001240
## [99] train-rmse:0.001240
## [100]    train-rmse:0.001240
```

2. this is the model for close recording

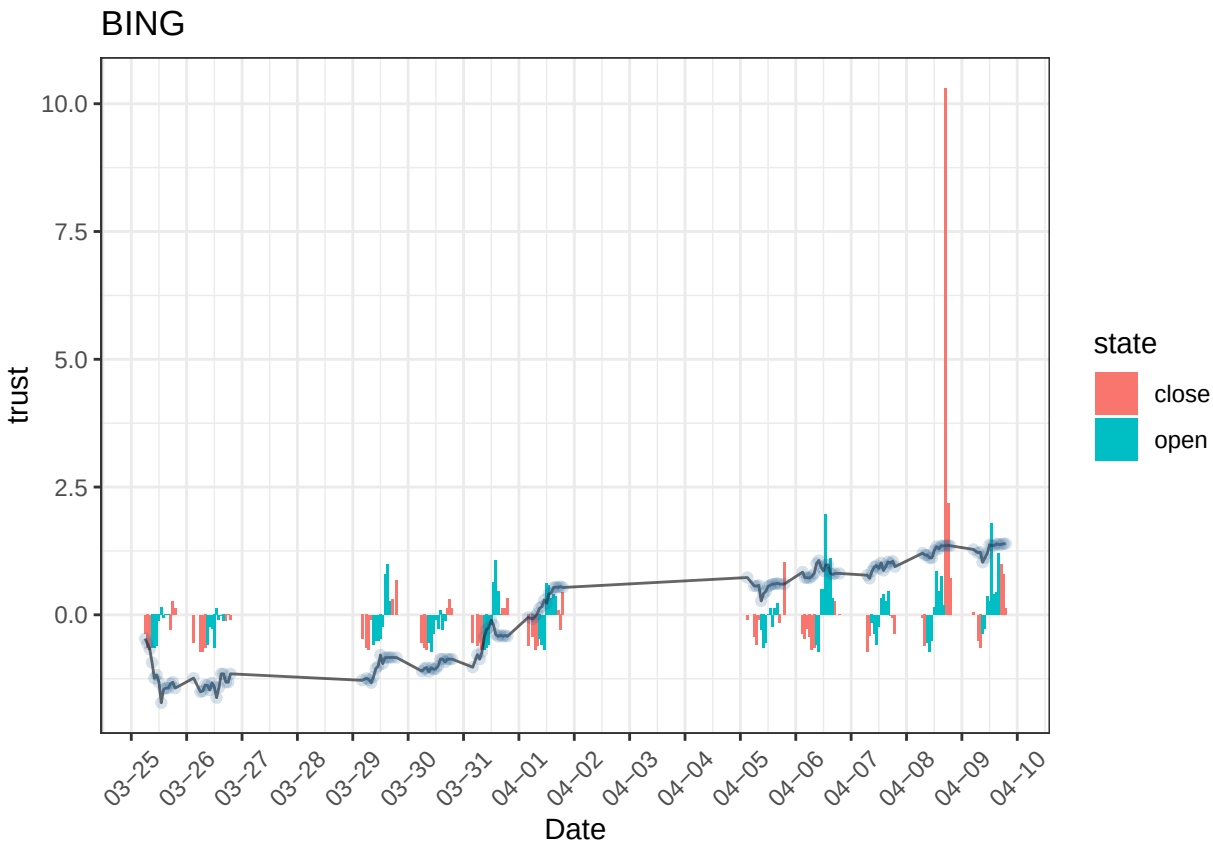
```
##
## Call:
## lm(formula = price ~ anger + anticipation + disgust + fear +
##     joy + negative + positive + sadness + surprise + trust, data = full_nrc[which(full_nrc$state ==
##     "close"), ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.46702 -0.81164  0.07964  0.87684  1.79687
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.05499    0.13701   0.401    0.690
## anger         0.39703    0.43809   0.906    0.369
## anticipation  0.96220    0.61767   1.558    0.125
## disgust       0.10540    0.41147   0.256    0.799
## fear         -0.22000    0.45331  -0.485    0.629
## joy          -0.25874    0.70392  -0.368    0.715
## negative     -0.43589    0.58276  -0.748    0.458
```

```
## positive     -0.13111    0.63739  -0.206    0.838
## sadness      -0.01230    0.30521  -0.040    0.968
## surprise     -0.28545    0.73494  -0.388    0.699
## trust         0.17265    0.73138   0.236    0.814
##
## Residual standard error: 0.9989 on 58 degrees of freedom
## Multiple R-squared:  0.1369, Adjusted R-squared:  -0.01187
## F-statistic: 0.9202 on 10 and 58 DF,  p-value: 0.5214
```

3. this is the model for open recording

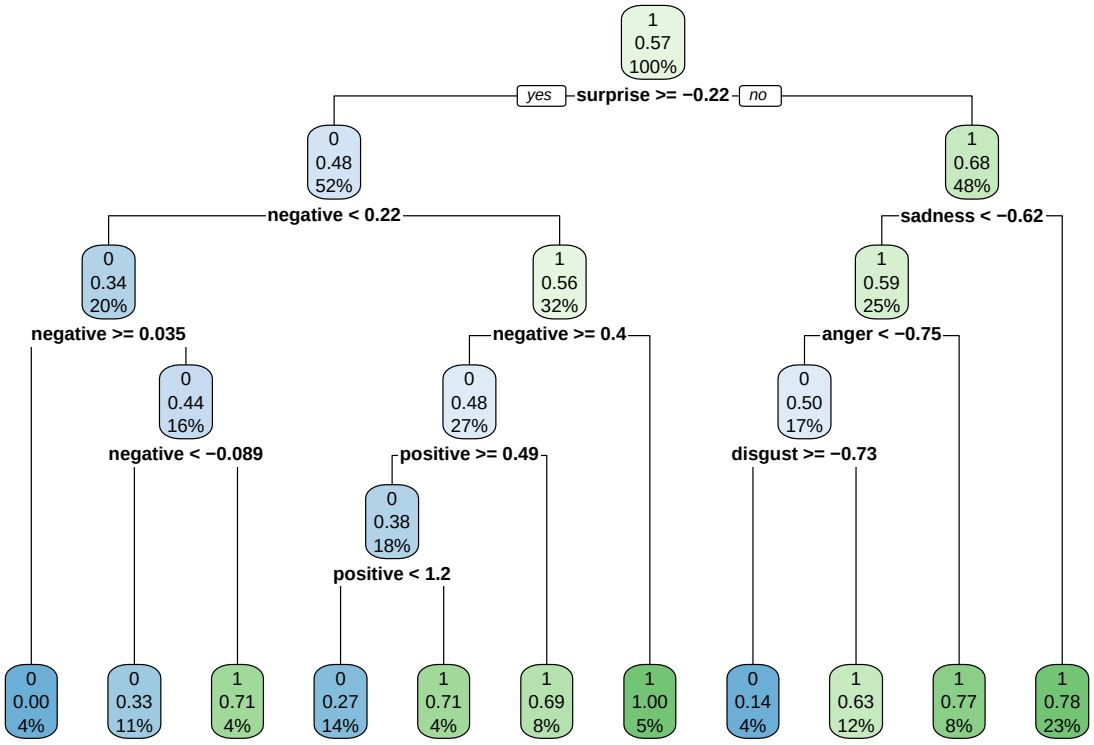
```
##
## Call:
## lm(formula = price ~ anger + anticipation + disgust + fear +
##     joy + negative + positive + sadness + surprise + trust, data = full_nrc[which(full_nrc$state ==
##     "open"), ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.77849 -0.77778  0.06436  0.79973  1.58906
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    0.0645     0.1043   0.618   0.5381
## anger         -0.5804     0.3495  -1.661   0.1009
## anticipation  -0.7183     0.3869  -1.857   0.0672 .
## disgust        0.1723     0.3133   0.550   0.5840
## fear           0.1037     0.3325   0.312   0.7559
## joy           -0.3713     0.4646  -0.799   0.4267
## negative       0.9108     0.3621   2.515   0.0140 *
## positive       0.2092     0.4354   0.480   0.6323
## sadness       -0.4368     0.1957  -2.232   0.0285 *
## surprise       0.9492     0.5449   1.742   0.0855 .
## trust          0.6139     0.4624   1.328   0.1882
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9194 on 77 degrees of freedom
## Multiple R-squared:  0.2671, Adjusted R-squared:  0.1719
## F-statistic: 2.806 on 10 and 77 DF,  p-value: 0.005126
```

the most relative variable is the trust sentiment, plotting its plot and stock price



NRC Decision Tree

maximum Tree




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  0  1
##          0 41 13
##          1 26 77
##
##                Accuracy : 0.7516
##                  95% CI : (0.6764, 0.817)
##     No Information Rate : 0.5732
##     P-Value [Acc > NIR] : 2.478e-06
##
##                   Kappa : 0.4794
##
##  Mcnemar's Test P-Value : 0.05466
##
##             Sensitivity : 0.6119
##             Specificity : 0.8556
##          Pos Pred Value : 0.7593
##          Neg Pred Value : 0.7476
##              Prevalence : 0.4268
##          Detection Rate : 0.2611
##    Detection Prevalence : 0.3439
##       Balanced Accuracy : 0.7337
##
##        'Positive' Class : 0
##

## [1]  train-logloss:0.647067
```

```
## [2]  train-logloss:0.615288
## [3]  train-logloss:0.593004
## [4]  train-logloss:0.573366
## [5]  train-logloss:0.558364
## [6]  train-logloss:0.545012
## [7]  train-logloss:0.536964
## [8]  train-logloss:0.526609
## [9]  train-logloss:0.519682
## [10] train-logloss:0.515167
## [11] train-logloss:0.509134
## [12] train-logloss:0.505694
## [13] train-logloss:0.500561
## [14] train-logloss:0.497981
## [15] train-logloss:0.492491
## [16] train-logloss:0.490379
## [17] train-logloss:0.488409
## [18] train-logloss:0.486949
## [19] train-logloss:0.484728
## [20] train-logloss:0.482801
## [21] train-logloss:0.480642
## [22] train-logloss:0.479183
## [23] train-logloss:0.478482
## [24] train-logloss:0.476626
## [25] train-logloss:0.474824
## [26] train-logloss:0.473557
## [27] train-logloss:0.472313
## [28] train-logloss:0.471126
## [29] train-logloss:0.470298
## [30] train-logloss:0.469275
## [31] train-logloss:0.468247
## [32] train-logloss:0.467641
## [33] train-logloss:0.467013
## [34] train-logloss:0.466105
## [35] train-logloss:0.465591
## [36] train-logloss:0.465255
## [37] train-logloss:0.464801
## [38] train-logloss:0.463907
## [39] train-logloss:0.463185
## [40] train-logloss:0.462722
## [41] train-logloss:0.462276
## [42] train-logloss:0.461742
## [43] train-logloss:0.461310
## [44] train-logloss:0.460795
## [45] train-logloss:0.460529
## [46] train-logloss:0.460103
## [47] train-logloss:0.459698
## [48] train-logloss:0.459228
## [49] train-logloss:0.458858
## [50] train-logloss:0.458581
## [51] train-logloss:0.458295
## [52] train-logloss:0.458055
## [53] train-logloss:0.457816
## [54] train-logloss:0.457528
## [55] train-logloss:0.457230
```

```
## [56] train-logloss:0.456946
## [57] train-logloss:0.456709
## [58] train-logloss:0.456494
## [59] train-logloss:0.456253
## [60] train-logloss:0.456057
## [61] train-logloss:0.455779
## [62] train-logloss:0.455578
## [63] train-logloss:0.455428
## [64] train-logloss:0.455252
## [65] train-logloss:0.455041
## [66] train-logloss:0.454895
## [67] train-logloss:0.454723
## [68] train-logloss:0.454590
## [69] train-logloss:0.454483
## [70] train-logloss:0.454344
## [71] train-logloss:0.454164
## [72] train-logloss:0.454009
## [73] train-logloss:0.453918
## [74] train-logloss:0.453795
## [75] train-logloss:0.453597
## [76] train-logloss:0.453448
## [77] train-logloss:0.453386
## [78] train-logloss:0.453207
## [79] train-logloss:0.453124
## [80] train-logloss:0.453038
## [81] train-logloss:0.452913
## [82] train-logloss:0.452836
## [83] train-logloss:0.452690
## [84] train-logloss:0.452578
## [85] train-logloss:0.452510
## [86] train-logloss:0.452383
## [87] train-logloss:0.452278
## [88] train-logloss:0.452187
## [89] train-logloss:0.452126
## [90] train-logloss:0.452036
## [91] train-logloss:0.451957
## [92] train-logloss:0.451896
## [93] train-logloss:0.451827
## [94] train-logloss:0.451775
## [95] train-logloss:0.451695
## [96] train-logloss:0.451648
## [97] train-logloss:0.451546
## [98] train-logloss:0.451503
## [99] train-logloss:0.451446
## [100]    train-logloss:0.451370
```

bing and Afinn regression

```
## Joining, by = "word"
## Joining, by = c("datetime", "date")
```

```
## Warning in log(price): NaNs produced
```

```
##
```

```
## Call:
## lm(formula = log(price) ~ negative + positive, data = full_bing)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.2983 -0.2667  0.1457  0.4146  0.6374
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.430335   0.110507  -3.894 0.000207 ***
## negative     0.008062   0.007335   1.099 0.275065
## positive     0.001750   0.008279   0.211 0.833160
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.6154 on 78 degrees of freedom
##   (75 observations deleted due to missingness)
## Multiple R-squared:  0.06192,    Adjusted R-squared:  0.03786
## F-statistic: 2.574 on 2 and 78 DF,  p-value: 0.08268

##
## Call:
## lm(formula = price ~ negative + positive, data = full_bing_close)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6205 -0.8936  0.1495  0.9099  1.4643
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.284881   0.172527  -1.651    0.104
## negative     0.011265   0.013707   0.822    0.414
## positive     0.006424   0.017685   0.363    0.718
##
## Residual standard error: 0.9756 on 65 degrees of freedom
## Multiple R-squared:  0.06851,    Adjusted R-squared:  0.03985
## F-statistic:  2.39 on 2 and 65 DF,  p-value: 0.09959

##
## Call:
## lm(formula = price ~ negative + positive, data = full_bing_open)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7334 -0.9110  0.2088  0.8445  1.4978
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.361961   0.177146  -2.043   0.0441 *
## negative     0.009157   0.014982   0.611   0.5427
## positive     0.014101   0.012778   1.103   0.2729
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
```

```
## Residual standard error: 0.9792 on 85 degrees of freedom
## Multiple R-squared:  0.08223,    Adjusted R-squared:  0.06063
## F-statistic: 3.808 on 2 and 85 DF,  p-value: 0.02607
```

```
## Joining, by = c("datetime", "date")
```

```
## Warning in log(price): NaNs produced
```

```
##
## Call:
## lm(formula = log(price) ~ sentiment, data = full_afinn)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.3577 -0.2985  0.1297  0.4339  0.6369
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.26737    0.07047  -3.794 0.000289 ***
## sentiment    0.06243    0.05918   1.055 0.294669
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.627 on 79 degrees of freedom
##   (73 observations deleted due to missingness)
## Multiple R-squared:  0.01389,    Adjusted R-squared:  0.001409
## F-statistic: 1.113 on 1 and 79 DF,  p-value: 0.2947
```

```
##
## Call:
## lm(formula = price ~ sentiment, data = full_afinn_close)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7373 -0.9152  0.1103  0.8364  1.4868
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.01359    0.12103   0.112    0.911
## sentiment    0.17169    0.11370   1.510    0.136
##
## Residual standard error: 0.9763 on 64 degrees of freedom
## Multiple R-squared:  0.0344, Adjusted R-squared:  0.01931
## F-statistic:  2.28 on 1 and 64 DF,  p-value: 0.136
```

```
##
## Call:
## lm(formula = price ~ sentiment, data = full_afinn_open)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.7113 -0.9529  0.1885  0.8690  1.5031
##
```

```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.008886   0.106711   0.083   0.9338
## sentiment   0.176051   0.096040   1.833   0.0702 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9969 on 86 degrees of freedom
## Multiple R-squared:  0.0376, Adjusted R-squared:  0.02641
## F-statistic:  3.36 on 1 and 86 DF,  p-value: 0.07025
```

Predict the following days

```
## # A tibble: 6 x 3
## # Groups:   date [1]
##   date       time                text
##   <date>     <chr>               <chr>
## 1 2021-04-09 2021-04-09 03:00:00 " JPM JPM Getting ready to POP here MA holding~
## 2 2021-04-09 2021-04-09 04:00:00 " JPM JPM Getting ready to POP here MA holding~
## 3 2021-04-09 2021-04-09 05:00:00 " NFLX made an ascending triangle scout wick t~
## 4 2021-04-09 2021-04-09 06:00:00 " There are a lot of big names that have yet t~
## 5 2021-04-09 2021-04-09 07:00:00 " CLOV UAL STPK NFLX BP Dark pool large option~
## 6 2021-04-09 2021-04-09 08:00:00 " Mark Your Calendars for an Upcoming Explosio~
```

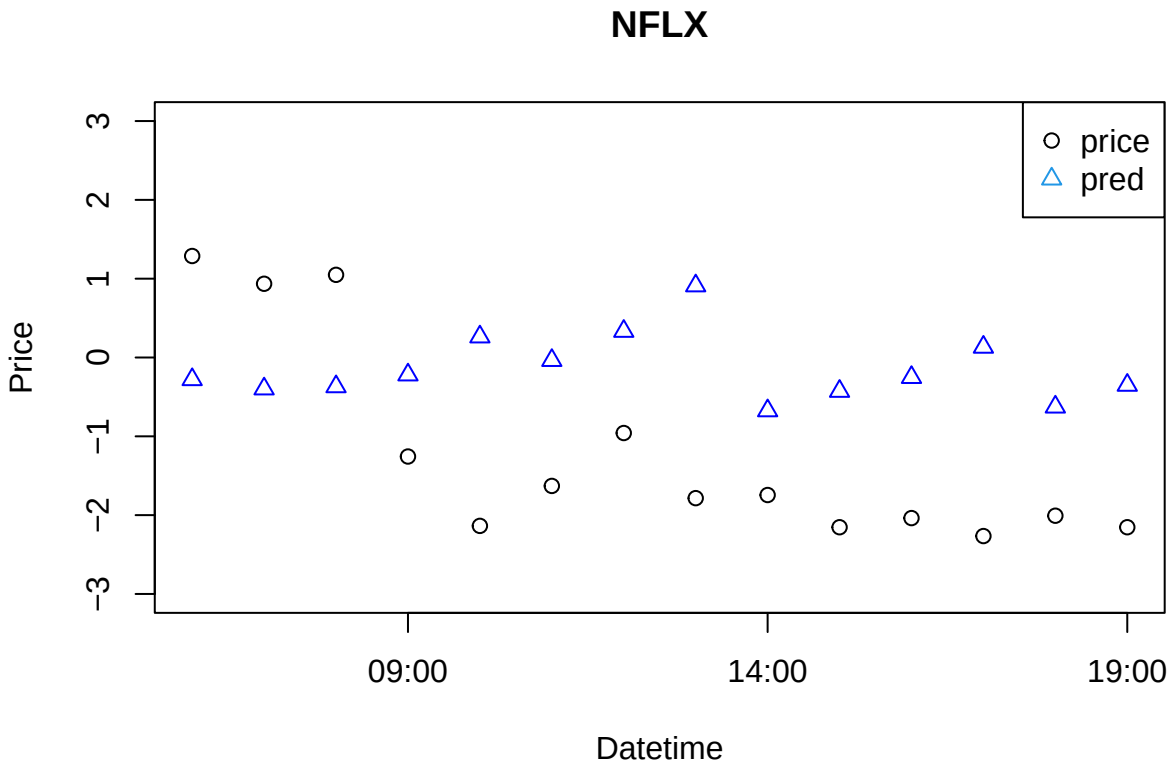
```
## [1] "there are total 195 observation"
```

```
## Joining, by = "word"
## Joining, by = "word"
## `summarise()` has grouped output by 'date'. You can override using the
## `.groups` argument.
## Joining, by = "word"
## `summarise()` has grouped output by 'date'. You can override using the
## `.groups` argument.
## Joining, by = "word"
## Joining, by = c("datetime", "date")
```

```
## # A tibble: 6 x 15
##   datetime            price date       time_stock anger anticipation disgust
##   <dttm>              <dbl> <date>     <chr>      <dbl>        <dbl>   <dbl>
## 1 2021-04-09 05:00:00 0.694 2021-04-09 05:00         11           43       1
## 2 2021-04-09 07:00:00 0.425 2021-04-09 07:00          0            5       1
## 3 2021-04-09 08:00:00 0.481 2021-04-09 08:00          4            5       3
## 4 2021-04-09 17:00:00 1.08  2021-04-09 17:00         26           38      12
## 5 2021-04-09 18:00:00 1.14  2021-04-09 18:00         48           36      14
## 6 2021-04-09 19:00:00 1.14  2021-04-09 19:00         17           20       9
## # ... with 8 more variables: fear <dbl>, joy <dbl>, negative <dbl>,
## #   positive <dbl>, sadness <dbl>, surprise <dbl>, trust <dbl>, state <chr>
```

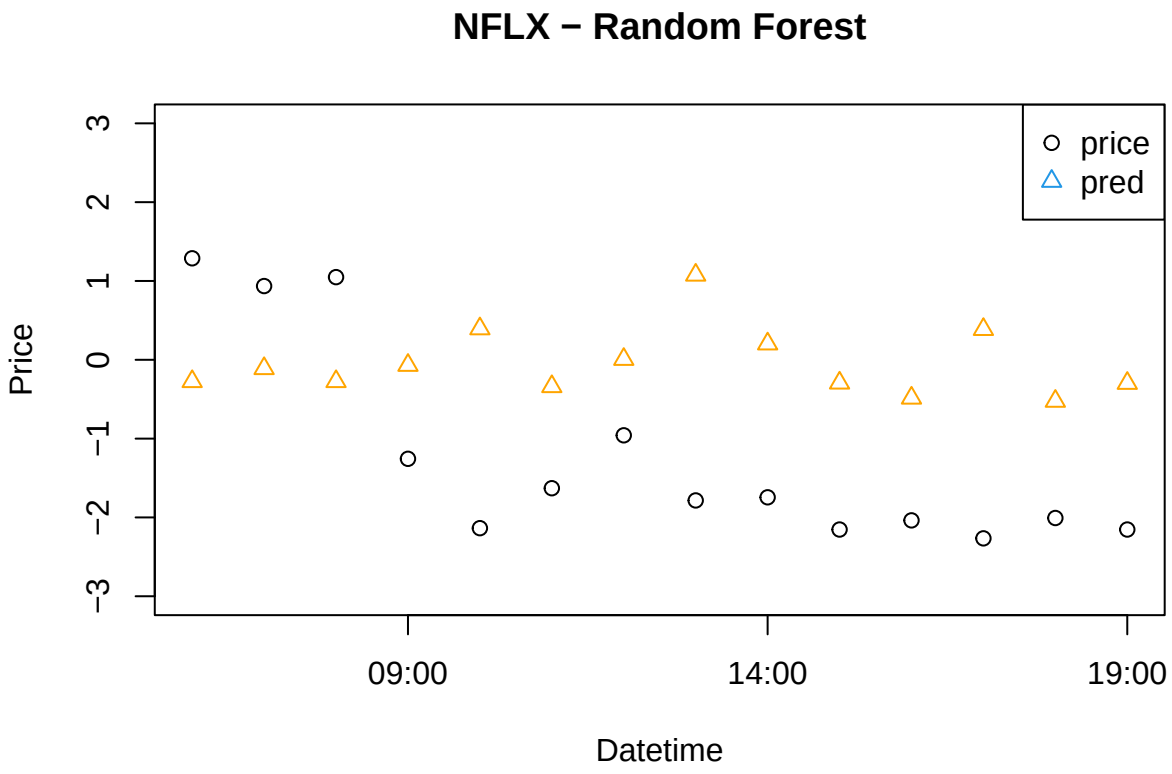
```
## # A tibble: 6 x 15
##   datetime              price date       time_stock anger anticipation disgust
##   <dttm>                <dbl> <date>     <chr>      <dbl>        <dbl>   <dbl>
## 1 2021-04-09 09:00:00 -0.309  2021-04-09 09:00          5            5       4
```

```
## 2 2021-04-09 10:00:00  0.0481 2021-04-09 10:00          5           12       1
## 3 2021-04-09 11:00:00  0.384  2021-04-09 11:00          5           19       2
## 4 2021-04-09 12:00:00  1.06   2021-04-09 12:00         19           32      12
## 5 2021-04-09 13:00:00  0.995  2021-04-09 13:00         25           63      13
## 6 2021-04-09 14:00:00  1.00   2021-04-09 14:00         16           35      13
## # ... with 8 more variables: fear <dbl>, joy <dbl>, negative <dbl>,
## #   positive <dbl>, sadness <dbl>, surprise <dbl>, trust <dbl>, state <chr>

## Joining, by = "word"
## Joining, by = c("datetime", "date")
## Joining, by = c("datetime", "date")
```

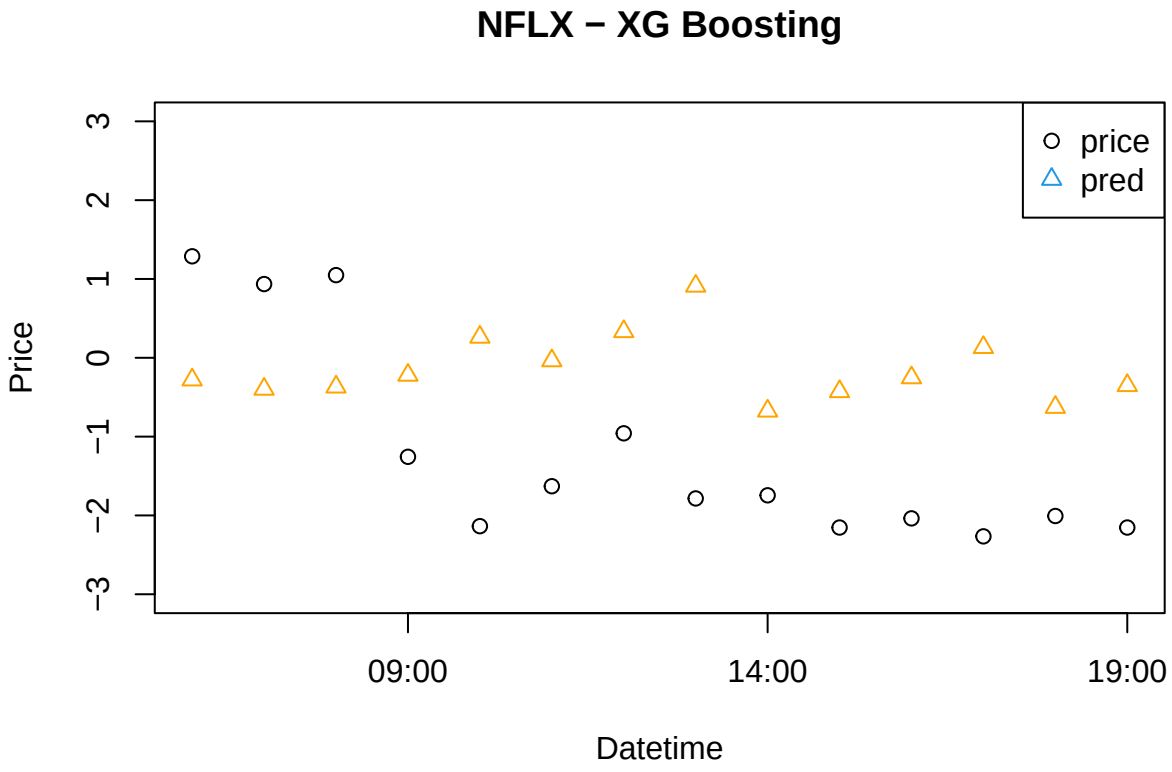



```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 0 1
##          1 8 5
##
##                Accuracy : 0.3571
##                  95% CI : (0.1276, 0.6486)
##     No Information Rate : 0.5714
##     P-Value [Acc > NIR] : 0.9703
##
##                   Kappa : -0.1455
##
##  Mcnemar's Test P-Value : 0.0455
##
##             Sensitivity : 0.00000
```

```
##             Specificity : 0.83333
##          Pos Pred Value : 0.00000
##          Neg Pred Value : 0.38462
##              Prevalence : 0.57143
##          Detection Rate : 0.00000
##    Detection Prevalence : 0.07143
##       Balanced Accuracy : 0.41667
##
##        'Positive' Class : 0
##


## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 0 1
##          1 8 5
##
##                Accuracy : 0.3571
##                  95% CI : (0.1276, 0.6486)
##     No Information Rate : 0.5714
##     P-Value [Acc > NIR] : 0.9703
##
##                   Kappa : -0.1455
##
##  Mcnemar's Test P-Value : 0.0455
##
##             Sensitivity : 0.00000
##             Specificity : 0.83333
##          Pos Pred Value : 0.00000
##          Neg Pred Value : 0.38462
##              Prevalence : 0.57143
##          Detection Rate : 0.00000
##    Detection Prevalence : 0.07143
##       Balanced Accuracy : 0.41667
##
##        'Positive' Class : 0
##
```




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 7 5
##          1 1 1
##
##                Accuracy : 0.5714
##                  95% CI : (0.2886, 0.8234)
##     No Information Rate : 0.5714
##     P-Value [Acc > NIR] : 0.6105
##
##                   Kappa : 0.0455
##
##  Mcnemar's Test P-Value : 0.2207
##
##             Sensitivity : 0.8750
##             Specificity : 0.1667
##          Pos Pred Value : 0.5833
##          Neg Pred Value : 0.5000
##              Prevalence : 0.5714
##          Detection Rate : 0.5000
##    Detection Prevalence : 0.8571
##       Balanced Accuracy : 0.5208
##
##        'Positive' Class : 0
##
```




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 3 1
##          1 5 5
##
##                Accuracy : 0.5714
##                  95% CI : (0.2886, 0.8234)
##     No Information Rate : 0.5714
##     P-Value [Acc > NIR] : 0.6105
##
##                   Kappa : 0.1923
##
##  Mcnemar's Test P-Value : 0.2207
##
##             Sensitivity : 0.3750
##             Specificity : 0.8333
##          Pos Pred Value : 0.7500
##          Neg Pred Value : 0.5000
##              Prevalence : 0.5714
##          Detection Rate : 0.2143
##    Detection Prevalence : 0.2857
##       Balanced Accuracy : 0.6042
##
##        'Positive' Class : 0
##
```