MSFT

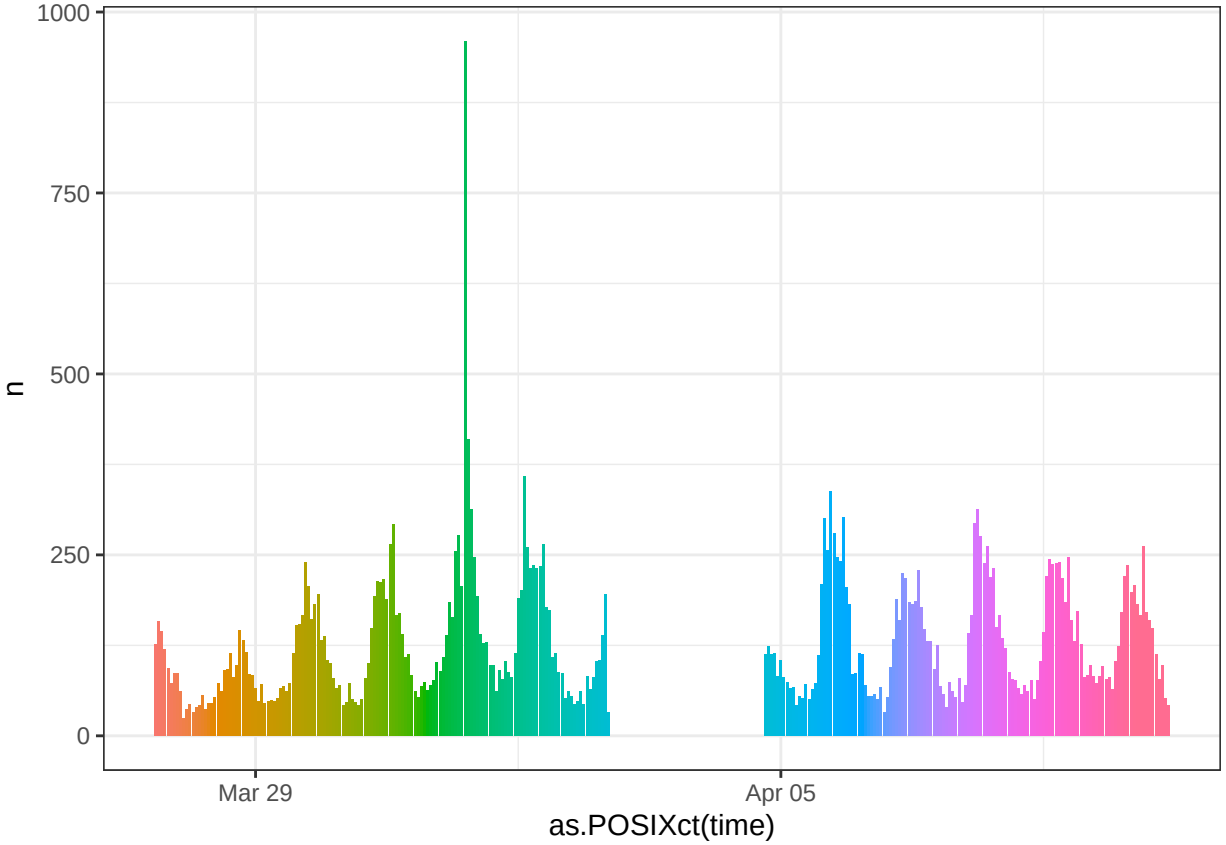
Evan Day

2023-05-08

MSFT

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 2021-03-27 2021-03-27 16:00:00 " Grab a comfy seat a favorite bev and tune in~
## 2 2021-03-27 2021-03-27 17:00:00 " Kindly enter your newly created Microsoft em~
## 3 2021-03-27 2021-03-27 18:00:00 " This spring clean out your MicrosoftTeams ch~
## 4 2021-03-27 2021-03-27 19:00:00 " Make MicrosoftSearch uniquely yours with new~
## 5 2021-03-27 2021-03-27 20:00:00 " Those ears   I have a small dividend portfol~
## 6 2021-03-27 2021-03-27 21:00:00 " We understand how important your account and~

## [1] "there are total 276 observation"
```

Sentiment Data frame with bing, afinn, and nrc

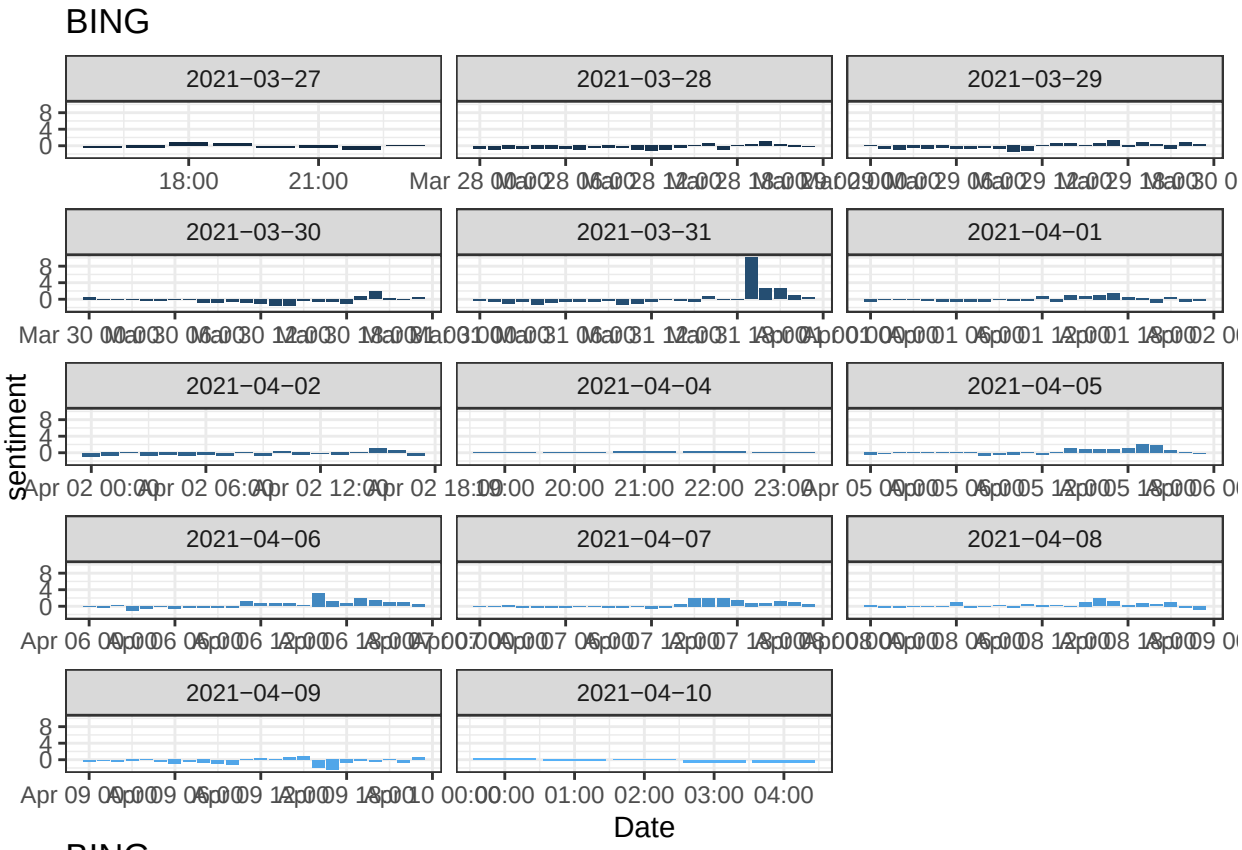
We start with the bing data frame

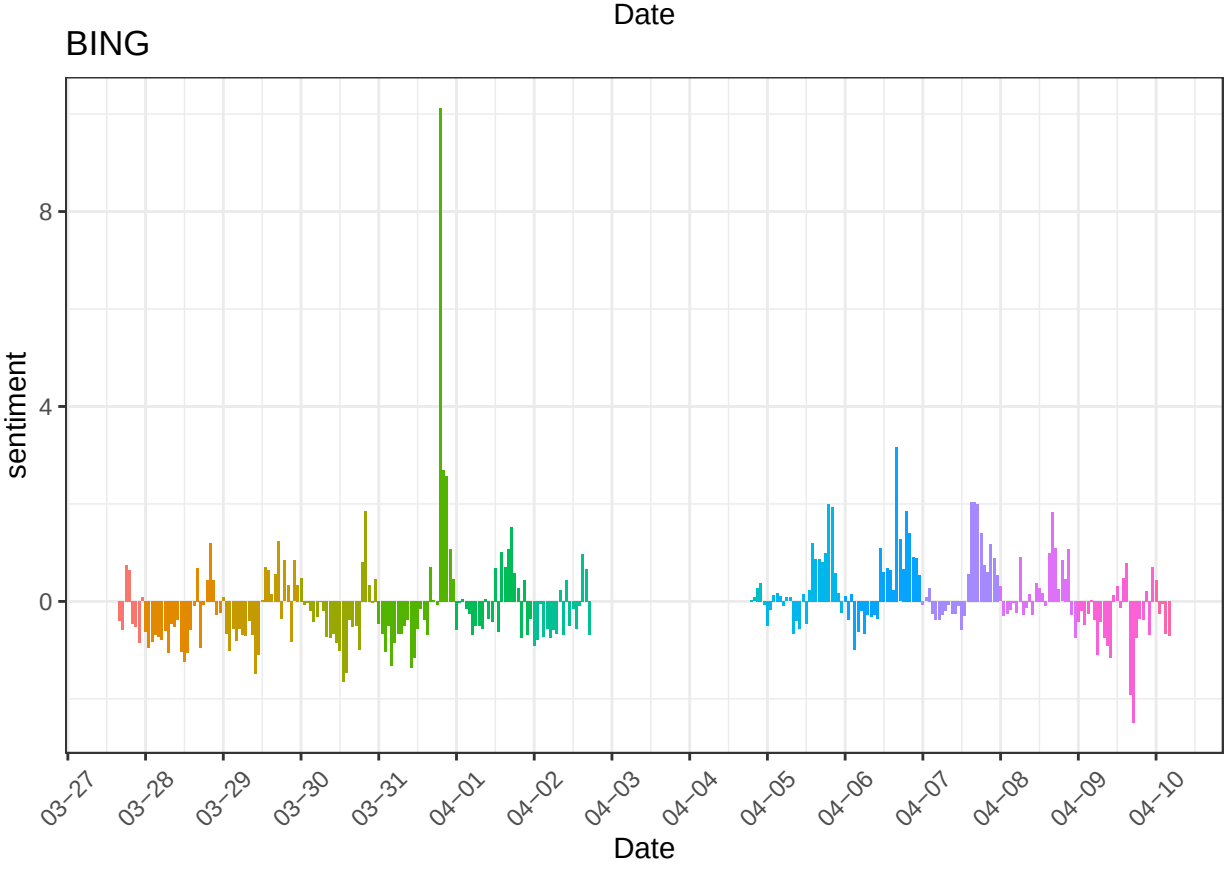
```
## # A tibble: 6 x 3
## # Groups:   date [1]
##   date       time                sentiment
##   <date>     <chr>                   <dbl>
## 1 2021-03-27 2021-03-27 16:00:00        21
## 2 2021-03-27 2021-03-27 17:00:00        14
## 3 2021-03-27 2021-03-27 18:00:00        66
## 4 2021-03-27 2021-03-27 19:00:00        62
## 5 2021-03-27 2021-03-27 20:00:00        19
## 6 2021-03-27 2021-03-27 21:00:00        16
```

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 2021-03-27 2021-03-27 16:00:00    -0.403
## 2 2021-03-27 2021-03-27 17:00:00    -0.581
## 3 2021-03-27 2021-03-27 18:00:00     0.739
## 4 2021-03-27 2021-03-27 19:00:00     0.637
## 5 2021-03-27 2021-03-27 20:00:00    -0.454
## 6 2021-03-27 2021-03-27 21:00:00    -0.530
```

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

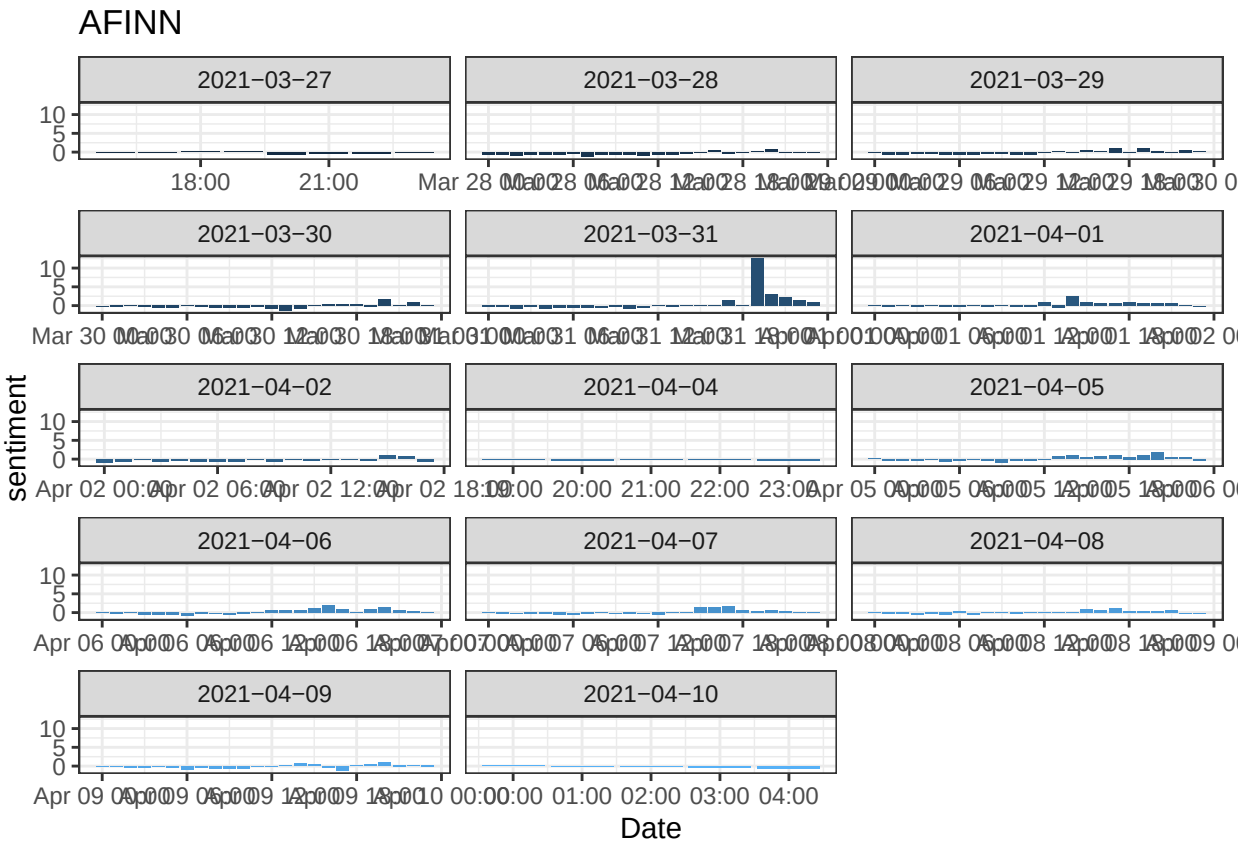


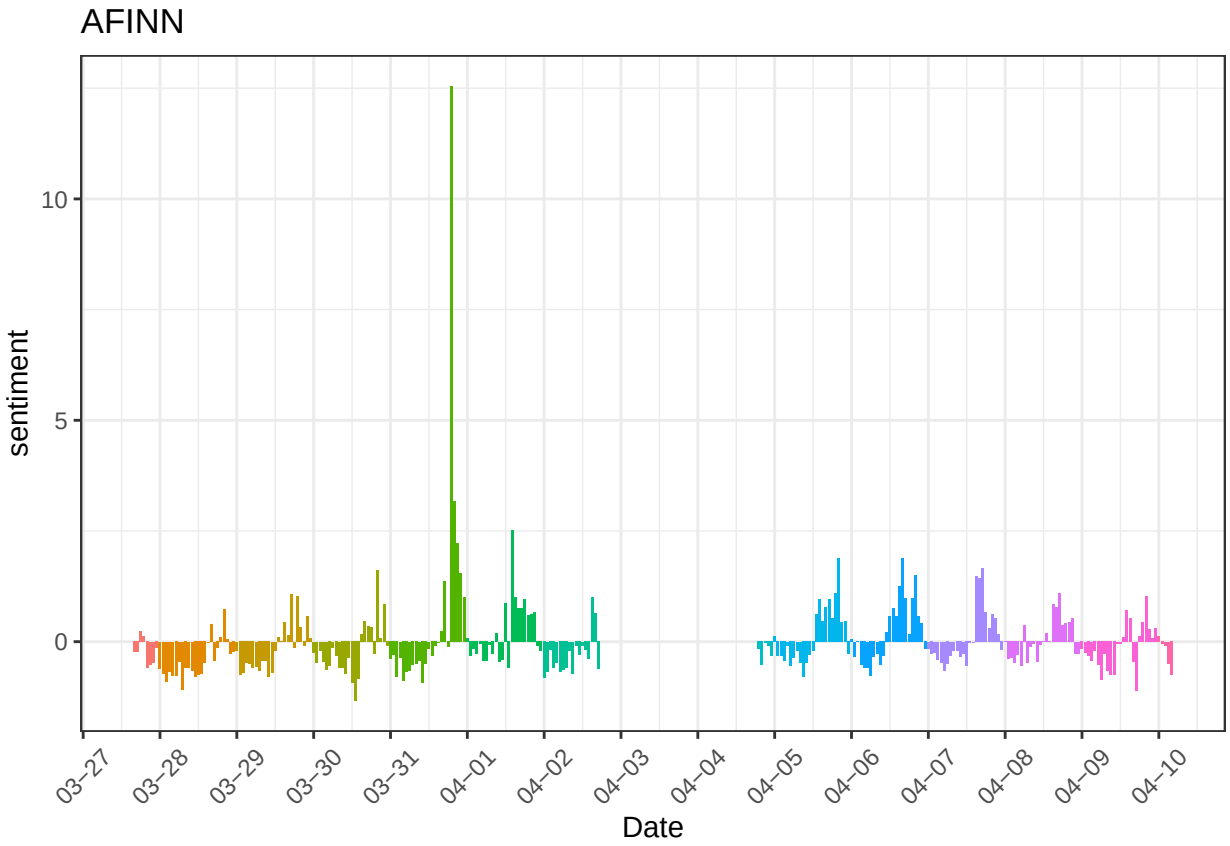


And then, we deal with the afinn sentiment dataframe

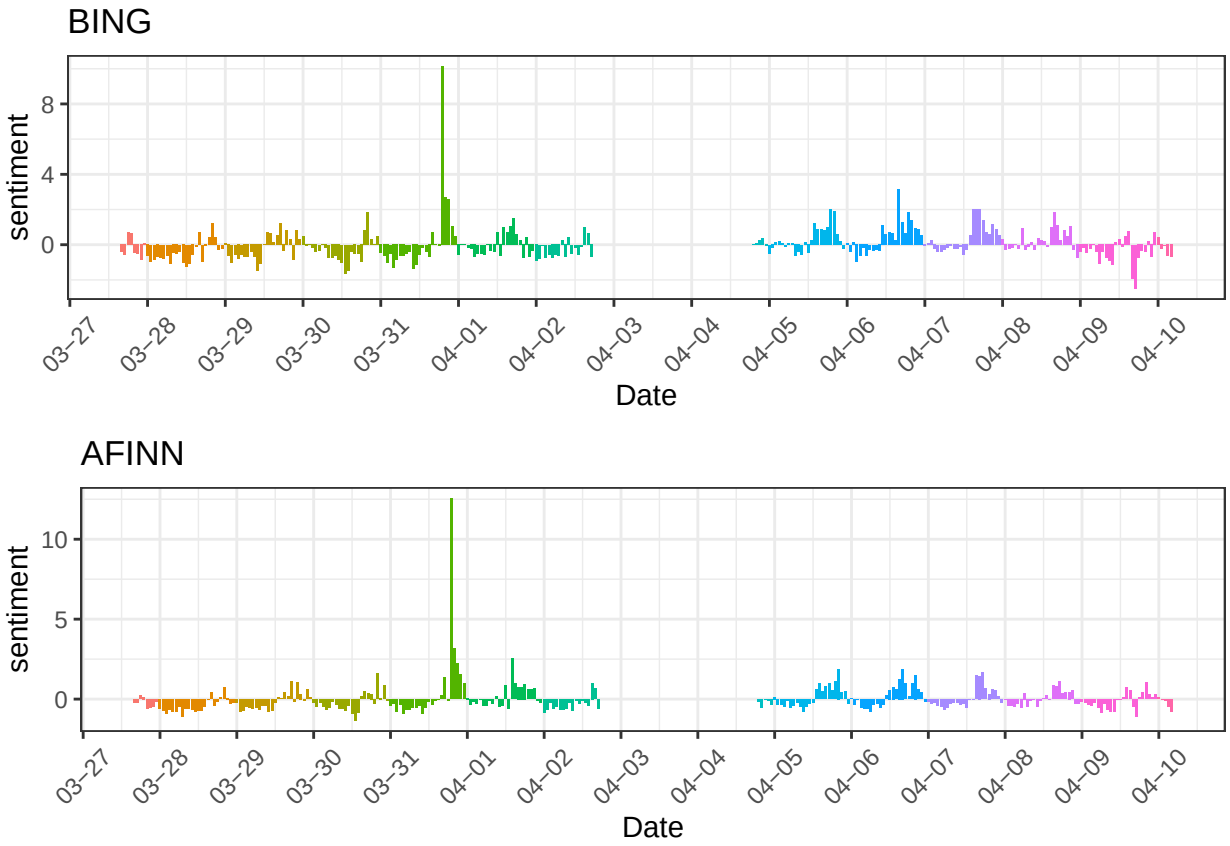
```
## # A tibble: 6 x 3
## # Groups:   date [1]
##   date       time                sentiment
##   <date>     <chr>                   <dbl>
## 1 2021-03-27 2021-03-27 16:00:00    -0.235
## 2 2021-03-27 2021-03-27 17:00:00    -0.235
## 3 2021-03-27 2021-03-27 18:00:00     0.240
## 4 2021-03-27 2021-03-27 19:00:00     0.128
## 5 2021-03-27 2021-03-27 20:00:00    -0.590
## 6 2021-03-27 2021-03-27 21:00:00    -0.515
```

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





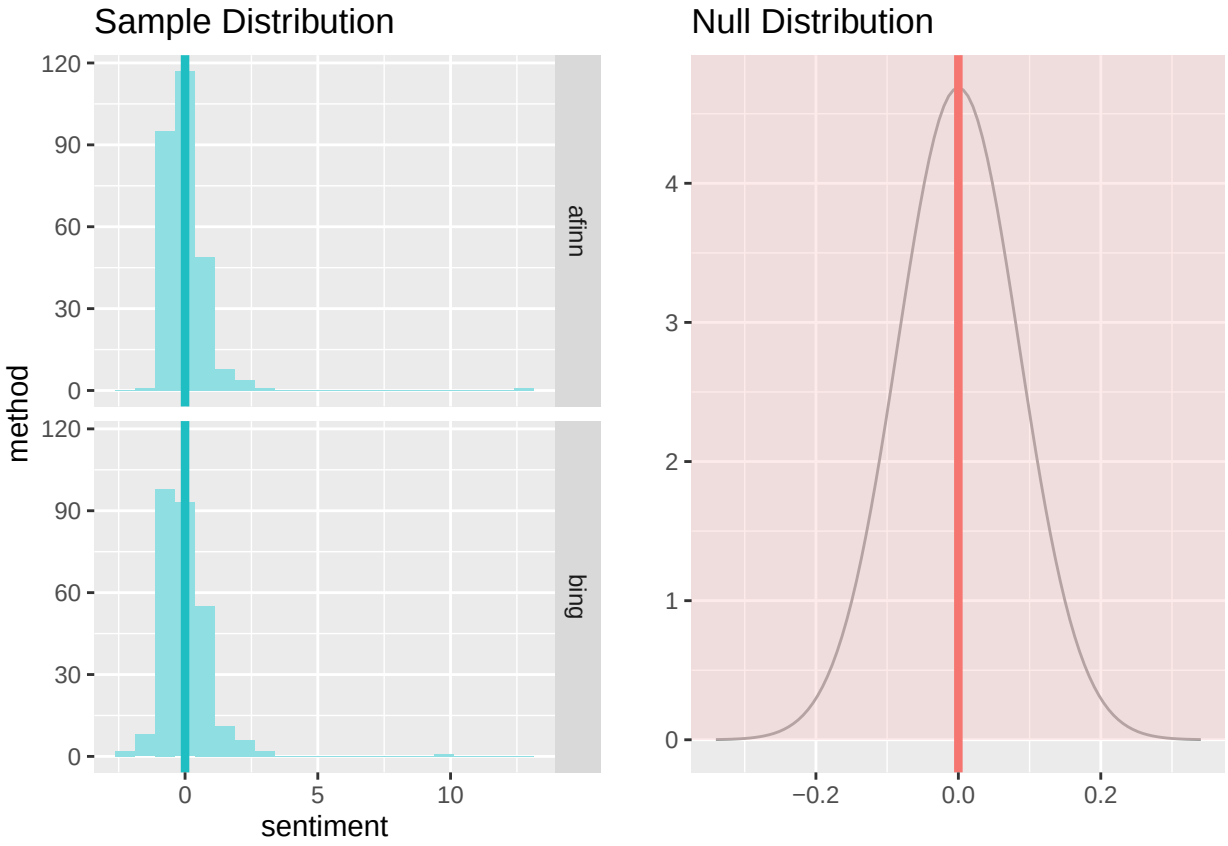
we compare the two sentiment plot together



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 = 276, y_bar_afinn = 0, s_afinn = 1
## n_bing = 276, y_bar_bing = 0, s_bing = 1
## H0: mu_afinn =  mu_bing
## HA: mu_afinn != mu_bing
## t = 0, df = 275
## p_value = 1
```



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

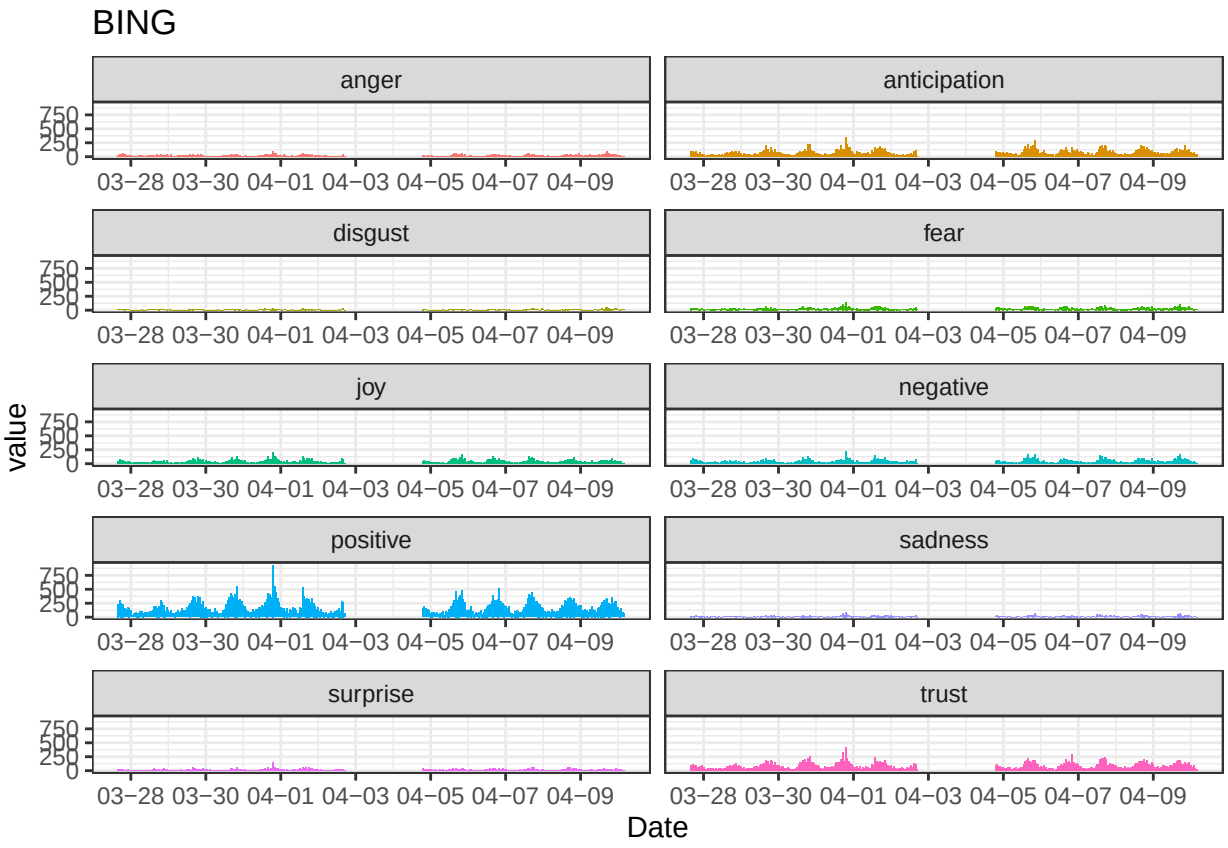
```
## Warning in ks.test(bing_afinn$bing, bing_afinn$afinn, alternative =
## "two.sided"): p-value will be approximate in the presence of ties
```

```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  bing_afinn$bing and bing_afinn$afinn
## D = 0.097826, p-value = 0.1425
## 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 2021-03-27 2021-03-2~    32           96      15    40    53       62      239
## 2 2021-03-27 2021-03-2~    45           99      10    36    77      106      294
## 3 2021-03-27 2021-03-2~    37           85      14    27    55       73      252
## 4 2021-03-27 2021-03-2~    54           94      16    35    71       82      235
## 5 2021-03-27 2021-03-2~    36           56      19    36    42       59      156
## 6 2021-03-27 2021-03-2~    23           47      12    21    35       47      136
## # ... 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
```



MSFT

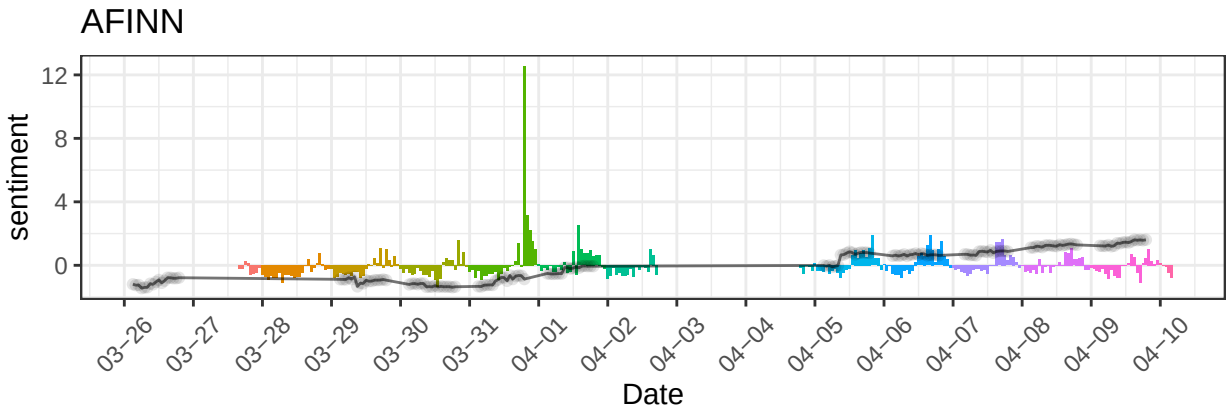
Stock Information

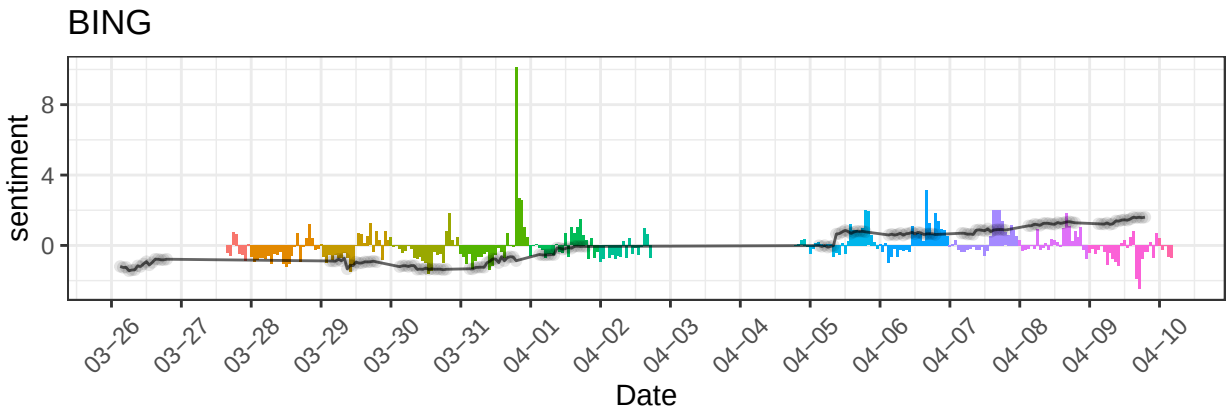
```
## # A tibble: 6 x 2
```

```
##   time                price
##   <chr>               <dbl>
## 1 2021-03-26 03:00:00  233.
## 2 2021-03-26 04:00:00  233.
## 3 2021-03-26 05:00:00  233.
## 4 2021-03-26 06:00:00  231.
## 5 2021-03-26 07:00:00  232.
## 6 2021-03-26 08:00:00  232.
```

normalize the price data:

```
## # A tibble: 6 x 2
##   time                price
##   <chr>               <dbl>
## 1 2021-03-26 03:00:00 -1.19
## 2 2021-03-26 04:00:00 -1.25
## 3 2021-03-26 05:00:00 -1.24
## 4 2021-03-26 06:00:00 -1.44
## 5 2021-03-26 07:00:00 -1.40
## 6 2021-03-26 08:00:00 -1.39
```





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 2021-03-29 03:00:00 -0.884 2021-03-29 03:00          9           38       1
## 2 2021-03-29 04:00:00 -0.892 2021-03-29 04:00         10           52       3
## 3 2021-03-29 05:00:00 -0.892 2021-03-29 05:00         10           48       3
## 4 2021-03-29 06:00:00 -0.781 2021-03-29 06:00         18           39       7
## 5 2021-03-29 07:00:00 -0.873 2021-03-29 07:00         18           32       4
## 6 2021-03-29 08:00:00 -0.724 2021-03-29 08:00         22           51       7
## # ... 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 2021-03-29 09:00:00 -1.33  2021-03-29 09:00         18           54       7
## 2 2021-03-29 10:00:00 -1.12  2021-03-29 10:00         27           48       7
## 3 2021-03-29 11:00:00 -1.15  2021-03-29 11:00         23           51       7
## 4 2021-03-29 12:00:00 -0.945 2021-03-29 12:00         22           83       5
## 5 2021-03-29 13:00:00 -1.00  2021-03-29 13:00         35           89      18
## 6 2021-03-29 14:00:00 -0.997 2021-03-29 14:00         49           97      22
## # ... with 8 more variables: fear <dbl>, joy <dbl>, negative <dbl>,
## #   positive <dbl>, sadness <dbl>, surprise <dbl>, trust <dbl>, state <chr>
```

MSFT 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
## -1.75620 -0.78035 -0.02109  0.76402  1.80988
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.120402   0.073823   1.631   0.1051
## anger       -0.141165   0.222939  -0.633   0.5276
## anticipation 0.130523   0.240377   0.543   0.5880
## disgust     -0.023810   0.143664  -0.166   0.8686
## fear         0.037835   0.207755   0.182   0.8558
## joy          0.302331   0.249167   1.213   0.2270
```

```
## negative      0.610149   0.262925   2.321   0.0217 *
## positive     -1.297312   0.306793  -4.229 4.21e-05 ***
## sadness       0.228880   0.165507   1.383   0.1689
## surprise      0.006483   0.178847   0.036   0.9711
## trust         0.311527   0.301079   1.035   0.3026
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9102 on 141 degrees of freedom
## Multiple R-squared:  0.1958, Adjusted R-squared:  0.1388
## F-statistic: 3.433 on 10 and 141 DF,  p-value: 0.0004712
```

```
## 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:0.892824
## [2]  train-rmse:0.735617
## [3]  train-rmse:0.627907
## [4]  train-rmse:0.556777
## [5]  train-rmse:0.507627
## [6]  train-rmse:0.422059
## [7]  train-rmse:0.363159
## [8]  train-rmse:0.314978
## [9]  train-rmse:0.289112
## [10] train-rmse:0.266354
## [11] train-rmse:0.253880
## [12] train-rmse:0.230102
## [13] train-rmse:0.208537
## [14] train-rmse:0.185056
```

```
## [15] train-rmse:0.169309
## [16] train-rmse:0.159939
## [17] train-rmse:0.145049
## [18] train-rmse:0.126138
## [19] train-rmse:0.121070
## [20] train-rmse:0.103143
## [21] train-rmse:0.097552
## [22] train-rmse:0.086827
## [23] train-rmse:0.074241
## [24] train-rmse:0.065481
## [25] train-rmse:0.057157
## [26] train-rmse:0.051839
## [27] train-rmse:0.047281
## [28] train-rmse:0.044471
## [29] train-rmse:0.040884
## [30] train-rmse:0.036178
## [31] train-rmse:0.030739
## [32] train-rmse:0.026856
## [33] train-rmse:0.023000
## [34] train-rmse:0.019931
## [35] train-rmse:0.017271
## [36] train-rmse:0.015440
## [37] train-rmse:0.014278
## [38] train-rmse:0.013525
## [39] train-rmse:0.011691
## [40] train-rmse:0.010412
## [41] train-rmse:0.009492
## [42] train-rmse:0.008052
## [43] train-rmse:0.007238
## [44] train-rmse:0.006594
## [45] train-rmse:0.005940
## [46] train-rmse:0.005122
## [47] train-rmse:0.004736
## [48] train-rmse:0.004022
## [49] train-rmse:0.003486
## [50] train-rmse:0.003056
## [51] train-rmse:0.002720
## [52] train-rmse:0.002506
## [53] train-rmse:0.002249
## [54] train-rmse:0.002008
## [55] train-rmse:0.001870
## [56] train-rmse:0.001711
## [57] train-rmse:0.001635
## [58] train-rmse:0.001479
## [59] train-rmse:0.001325
## [60] train-rmse:0.001270
## [61] train-rmse:0.001152
## [62] train-rmse:0.001084
## [63] train-rmse:0.001031
## [64] train-rmse:0.001026
## [65] train-rmse:0.001026
## [66] train-rmse:0.001026
## [67] train-rmse:0.001026
## [68] train-rmse:0.001026
```

```
## [69] train-rmse:0.001026
## [70] train-rmse:0.001026
## [71] train-rmse:0.001026
## [72] train-rmse:0.001026
## [73] train-rmse:0.001026
## [74] train-rmse:0.001026
## [75] train-rmse:0.001026
## [76] train-rmse:0.001026
## [77] train-rmse:0.001026
## [78] train-rmse:0.001026
## [79] train-rmse:0.001026
## [80] train-rmse:0.001026
## [81] train-rmse:0.001026
## [82] train-rmse:0.001026
## [83] train-rmse:0.001026
## [84] train-rmse:0.001026
## [85] train-rmse:0.001026
## [86] train-rmse:0.001026
## [87] train-rmse:0.001026
## [88] train-rmse:0.001026
## [89] train-rmse:0.001026
## [90] train-rmse:0.001026
## [91] train-rmse:0.001026
## [92] train-rmse:0.001026
## [93] train-rmse:0.001026
## [94] train-rmse:0.001026
## [95] train-rmse:0.001026
## [96] train-rmse:0.001026
## [97] train-rmse:0.001026
## [98] train-rmse:0.001026
## [99] train-rmse:0.001026
## [100]    train-rmse:0.001026
```

2. this is the model for close recording

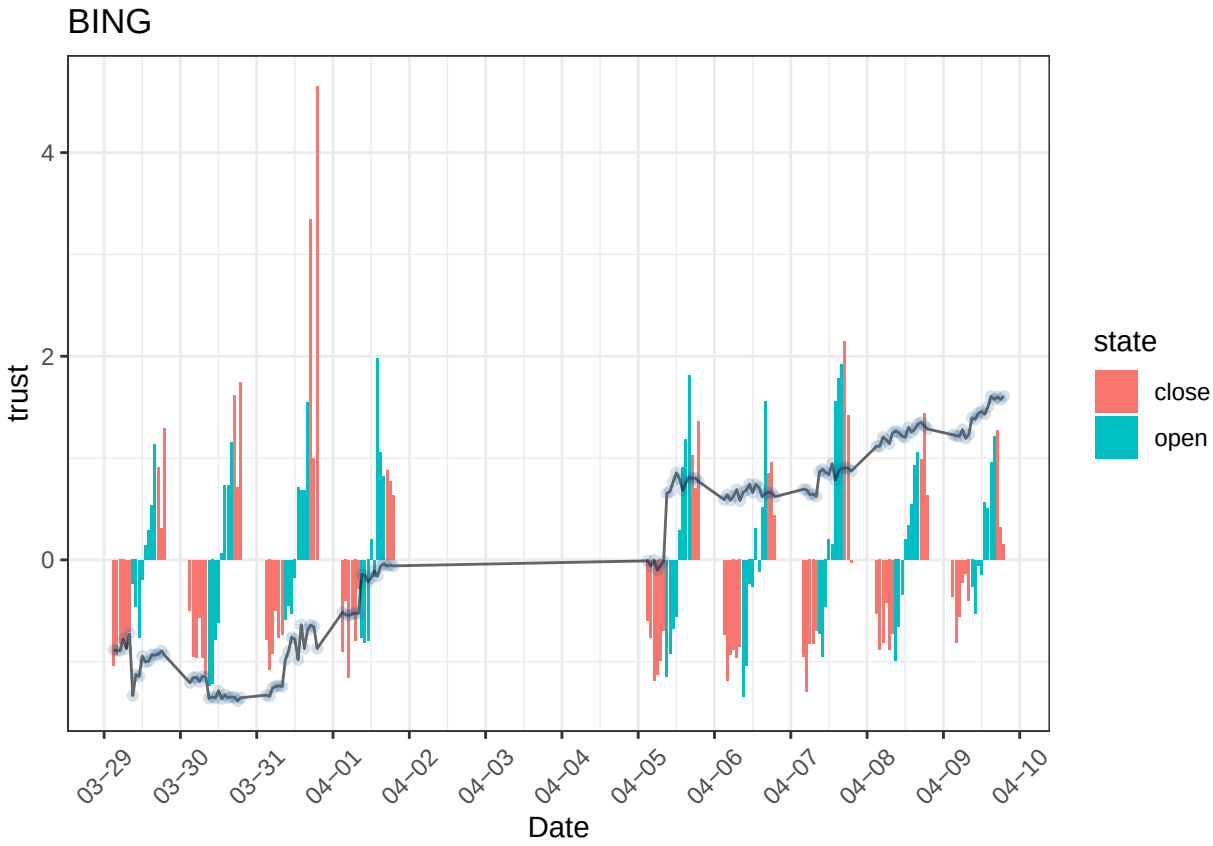
```
##
## Call:
## lm(formula = price ~ anger + anticipation + disgust + fear +
##     joy + negative + positive + sadness + surprise + trust, data = full_nrc[which(full_nrc$state ==
##     "close"), ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.49019 -0.74842 -0.04595  0.74792  1.58978
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.11662    0.10519   1.109   0.2715
## anger         0.10097    0.37572   0.269   0.7889
## anticipation -0.28169    0.42069  -0.670   0.5054
## disgust      -0.46481    0.25100  -1.852   0.0683 .
## fear          0.12785    0.32057   0.399   0.6913
## joy           0.51476    0.43917   1.172   0.2452
## negative      1.17796    0.55340   2.129   0.0369 *
```

```
## positive     -1.35224    0.52122  -2.594   0.0116 *
## sadness       0.40891    0.26486   1.544   0.1272
## surprise      0.02495    0.21870   0.114   0.9095
## trust        -0.08397    0.43294  -0.194   0.8468
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.907 on 69 degrees of freedom
## Multiple R-squared:  0.2273, Adjusted R-squared:  0.1153
## F-statistic:  2.03 on 10 and 69 DF,  p-value: 0.04304
```

3. this is the model for open recording

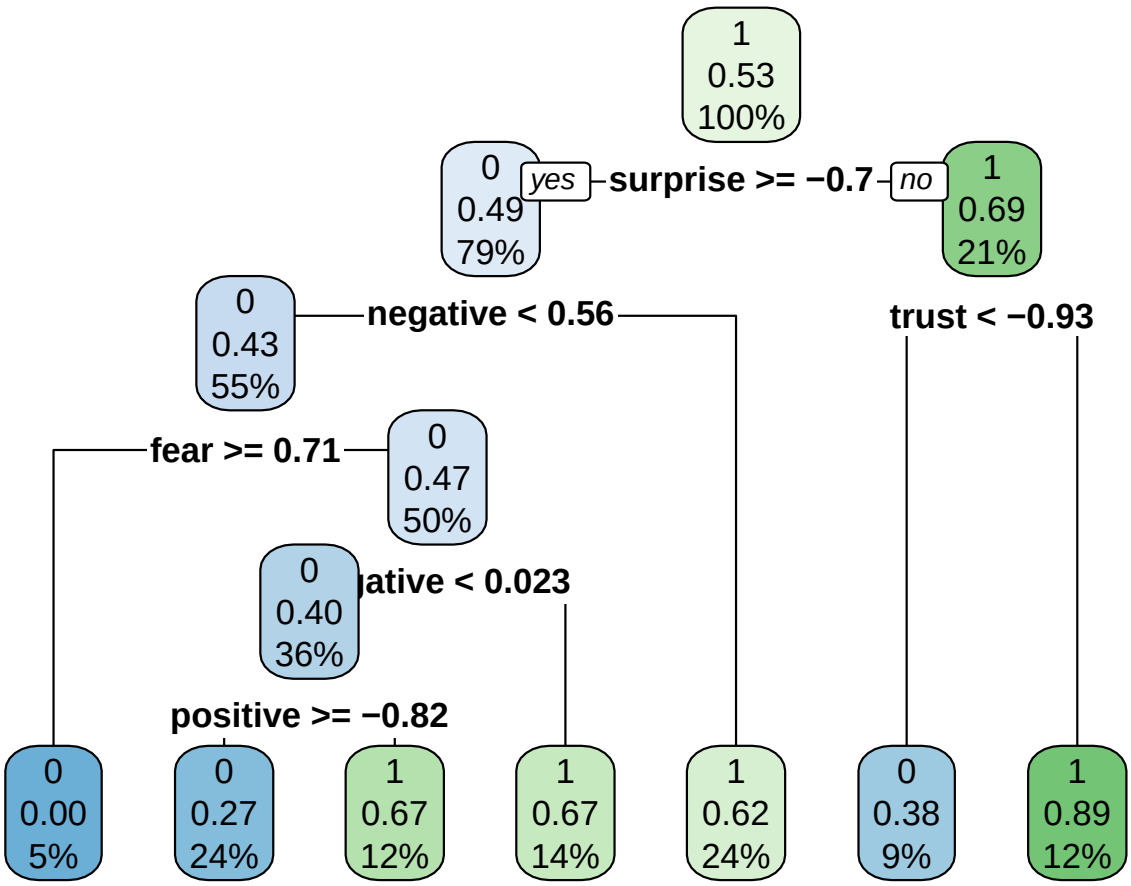
```
##
## Call:
## lm(formula = price ~ anger + anticipation + disgust + fear +
##     joy + negative + positive + sadness + surprise + trust, data = full_nrc[which(full_nrc$state ==
##     "open"), ])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.9377 -0.6536  0.1289  0.6586  1.7734
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.14433    0.11237   1.284  0.20387
## anger        -0.52111    0.30809  -1.691  0.09586 .
## anticipation -0.01701    0.40638  -0.042  0.96675
## disgust       0.12961    0.19529   0.664  0.50941
## fear          0.08222    0.31145   0.264  0.79267
## joy           0.33457    0.33460   1.000  0.32131
## negative      0.53520    0.37315   1.434  0.15660
## positive     -1.64718    0.44945  -3.665  0.00052 ***
## sadness       0.11202    0.25242   0.444  0.65876
## surprise      0.10674    0.36051   0.296  0.76817
## trust         1.02171    0.56379   1.812  0.07487 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.92 on 61 degrees of freedom
## Multiple R-squared:  0.2745, Adjusted R-squared:  0.1555
## F-statistic: 2.308 on 10 and 61 DF,  p-value: 0.02252
```

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 42 15
##          1 29 66
##
##                Accuracy : 0.7105
##                  95% CI : (0.6315, 0.7811)
##     No Information Rate : 0.5329
##     P-Value [Acc > NIR] : 5.776e-06
##
##                   Kappa : 0.4114
##
##  Mcnemar's Test P-Value : 0.05002
##
##             Sensitivity : 0.5915
##             Specificity : 0.8148
##          Pos Pred Value : 0.7368
##          Neg Pred Value : 0.6947
##              Prevalence : 0.4671
##          Detection Rate : 0.2763
##    Detection Prevalence : 0.3750
##       Balanced Accuracy : 0.7032
##
```

```
##        'Positive' Class : 0
##

## [1]  train-logloss:0.657048
## [2]  train-logloss:0.622814
## [3]  train-logloss:0.596351
## [4]  train-logloss:0.570412
## [5]  train-logloss:0.556099
## [6]  train-logloss:0.547227
## [7]  train-logloss:0.534710
## [8]  train-logloss:0.527844
## [9]  train-logloss:0.517701
## [10] train-logloss:0.509808
## [11] train-logloss:0.501983
## [12] train-logloss:0.494455
## [13] train-logloss:0.489532
## [14] train-logloss:0.485628
## [15] train-logloss:0.482740
## [16] train-logloss:0.480852
## [17] train-logloss:0.478051
## [18] train-logloss:0.476617
## [19] train-logloss:0.475370
## [20] train-logloss:0.472805
## [21] train-logloss:0.470710
## [22] train-logloss:0.469828
## [23] train-logloss:0.467960
## [24] train-logloss:0.465971
## [25] train-logloss:0.464123
## [26] train-logloss:0.462438
## [27] train-logloss:0.461512
## [28] train-logloss:0.460131
## [29] train-logloss:0.458885
## [30] train-logloss:0.458073
## [31] train-logloss:0.457518
## [32] train-logloss:0.456721
## [33] train-logloss:0.456067
## [34] train-logloss:0.455298
## [35] train-logloss:0.454496
## [36] train-logloss:0.453956
## [37] train-logloss:0.453466
## [38] train-logloss:0.452781
## [39] train-logloss:0.452387
## [40] train-logloss:0.451886
## [41] train-logloss:0.451267
## [42] train-logloss:0.450680
## [43] train-logloss:0.450268
## [44] train-logloss:0.450043
## [45] train-logloss:0.449804
## [46] train-logloss:0.449519
## [47] train-logloss:0.449350
## [48] train-logloss:0.449087
## [49] train-logloss:0.448723
## [50] train-logloss:0.448432
## [51] train-logloss:0.448224
```

```
## [52] train-logloss:0.447956
## [53] train-logloss:0.447608
## [54] train-logloss:0.447299
## [55] train-logloss:0.447151
## [56] train-logloss:0.447044
## [57] train-logloss:0.446865
## [58] train-logloss:0.446690
## [59] train-logloss:0.446490
## [60] train-logloss:0.446387
## [61] train-logloss:0.446257
## [62] train-logloss:0.446058
## [63] train-logloss:0.445964
## [64] train-logloss:0.445836
## [65] train-logloss:0.445626
## [66] train-logloss:0.445544
## [67] train-logloss:0.445418
## [68] train-logloss:0.445352
## [69] train-logloss:0.445274
## [70] train-logloss:0.445214
## [71] train-logloss:0.445125
## [72] train-logloss:0.445037
## [73] train-logloss:0.444921
## [74] train-logloss:0.444810
## [75] train-logloss:0.444744
## [76] train-logloss:0.444651
## [77] train-logloss:0.444577
## [78] train-logloss:0.444500
## [79] train-logloss:0.444407
## [80] train-logloss:0.444361
## [81] train-logloss:0.444296
## [82] train-logloss:0.444208
## [83] train-logloss:0.444131
## [84] train-logloss:0.444073
## [85] train-logloss:0.444037
## [86] train-logloss:0.443993
## [87] train-logloss:0.443903
## [88] train-logloss:0.443841
## [89] train-logloss:0.443791
## [90] train-logloss:0.443718
## [91] train-logloss:0.443672
## [92] train-logloss:0.443628
## [93] train-logloss:0.443591
## [94] train-logloss:0.443547
## [95] train-logloss:0.443489
## [96] train-logloss:0.443445
## [97] train-logloss:0.443414
## [98] train-logloss:0.443376
## [99] train-logloss:0.443328
## [100]    train-logloss:0.443296
```

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
## -0.51978 -0.23892 -0.04478  0.28628  0.59593
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.1365357  0.0712993  -1.915  0.05931 .
## negative     0.0056398  0.0014364   3.926  0.00019 ***
## positive    -0.0021790  0.0009213  -2.365  0.02061 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2946 on 75 degrees of freedom
##   (74 observations deleted due to missingness)
## Multiple R-squared:  0.1736, Adjusted R-squared:  0.1515
## F-statistic: 7.877 on 2 and 75 DF,  p-value: 0.0007852
```

```
##
## Call:
## lm(formula = price ~ negative + positive, data = full_bing_close)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.58386 -0.89985 -0.03086  0.78201  1.45848
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.0306825  0.1754229  -0.175    0.862
## negative     0.0021425  0.0045473   0.471    0.639
## positive    -0.0001274  0.0022167  -0.057    0.954
##
## Residual standard error: 0.9738 on 77 degrees of freedom
## Multiple R-squared:  0.005971,   Adjusted R-squared:  -0.01985
## F-statistic: 0.2313 on 2 and 77 DF,  p-value: 0.7941
```

```
##
## Call:
## lm(formula = price ~ negative + positive, data = full_bing_open)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.58279 -0.71705  0.05908  0.65101  2.57366
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
```

```
## (Intercept)  0.041514   0.249994   0.166  0.86859
## negative    -0.014665   0.004728  -3.102  0.00279 **
## positive     0.010344   0.003088   3.350  0.00131 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9345 on 69 degrees of freedom
## Multiple R-squared:  0.1532, Adjusted R-squared:  0.1286
## F-statistic: 6.241 on 2 and 69 DF,  p-value: 0.003225
```

```
## 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
## -0.5069 -0.3059 -0.0430  0.2671  0.5517
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.05403    0.03657  -1.478    0.144
## sentiment   -0.04576    0.05490  -0.834    0.407
##
## Residual standard error: 0.3205 on 76 degrees of freedom
##   (74 observations deleted due to missingness)
## Multiple R-squared:  0.00906,    Adjusted R-squared:  -0.003979
## F-statistic: 0.6948 on 1 and 76 DF,  p-value: 0.4071
```

```
##
## Call:
## lm(formula = price ~ sentiment, data = full_afinn_close)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.43726 -0.95765 -0.08377  0.82857  1.55570
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.06110    0.10848   0.563    0.575
## sentiment   -0.02427    0.07100  -0.342    0.733
##
## Residual standard error: 0.9697 on 78 degrees of freedom
## Multiple R-squared:  0.001495,   Adjusted R-squared:  -0.01131
## F-statistic: 0.1168 on 1 and 78 DF,  p-value: 0.7334
```

```
##
## Call:
## lm(formula = price ~ sentiment, data = full_afinn_open)
##
```

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.6983 -0.9832  0.1375  0.7999  1.5900
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.1680     0.1142   1.471   0.1459
## sentiment     0.4033     0.1626   2.480   0.0156 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9667 on 70 degrees of freedom
## Multiple R-squared:  0.08076,    Adjusted R-squared:  0.06762
## F-statistic: 6.149 on 1 and 70 DF,  p-value: 0.01555
```

Predict the following days

```
## # A tibble: 6 x 3
## # Groups:   date [1]
##   date       time                text
##   <date>     <chr>               <chr>
## 1 2021-04-12 2021-04-12 13:00:00 " Thats the thing They are all unopened packag~
## 2 2021-04-12 2021-04-12 14:00:00 " GAFAM GOOG AMZN FB AAPL MSFT   testing   Mic~
## 3 2021-04-12 2021-04-12 15:00:00 " GAFAM GOOG AMZN FB AAPL MSFT   What s Moving~
## 4 2021-04-12 2021-04-12 16:00:00 " And the pricing was invented by take two and~
## 5 2021-04-12 2021-04-12 17:00:00 " GAFAM GOOG AMZN FB AAPL MSFT   I can only ho~
## 6 2021-04-12 2021-04-12 18:00:00 " Mid Day Market Update Crude Oil Rises iRhyth~
```

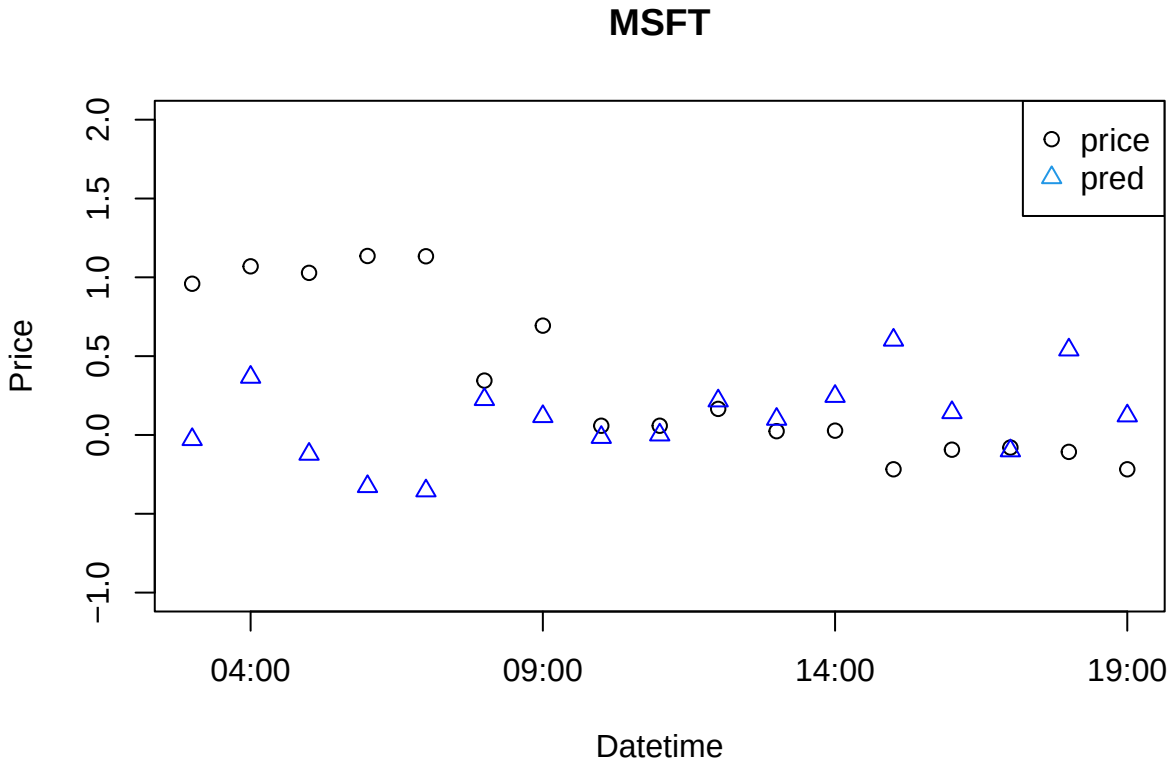
```
## [1] "there are total 111 observation"
```

```
## 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 2021-04-12 17:00:00 -0.0762 2021-04-12 17:00         56          161      21
## 2 2021-04-12 18:00:00 -0.0866 2021-04-12 18:00         40          136      13
## 3 2021-04-12 19:00:00 -0.107  2021-04-12 19:00         37          141      24
## 4 2021-04-13 03:00:00  0.155  2021-04-13 03:00         44           74      42
## 5 2021-04-13 04:00:00  0.0653 2021-04-13 04:00         25           60       9
## 6 2021-04-13 05:00:00 -0.0590 2021-04-13 05:00         59          101      40
## # ... 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
```

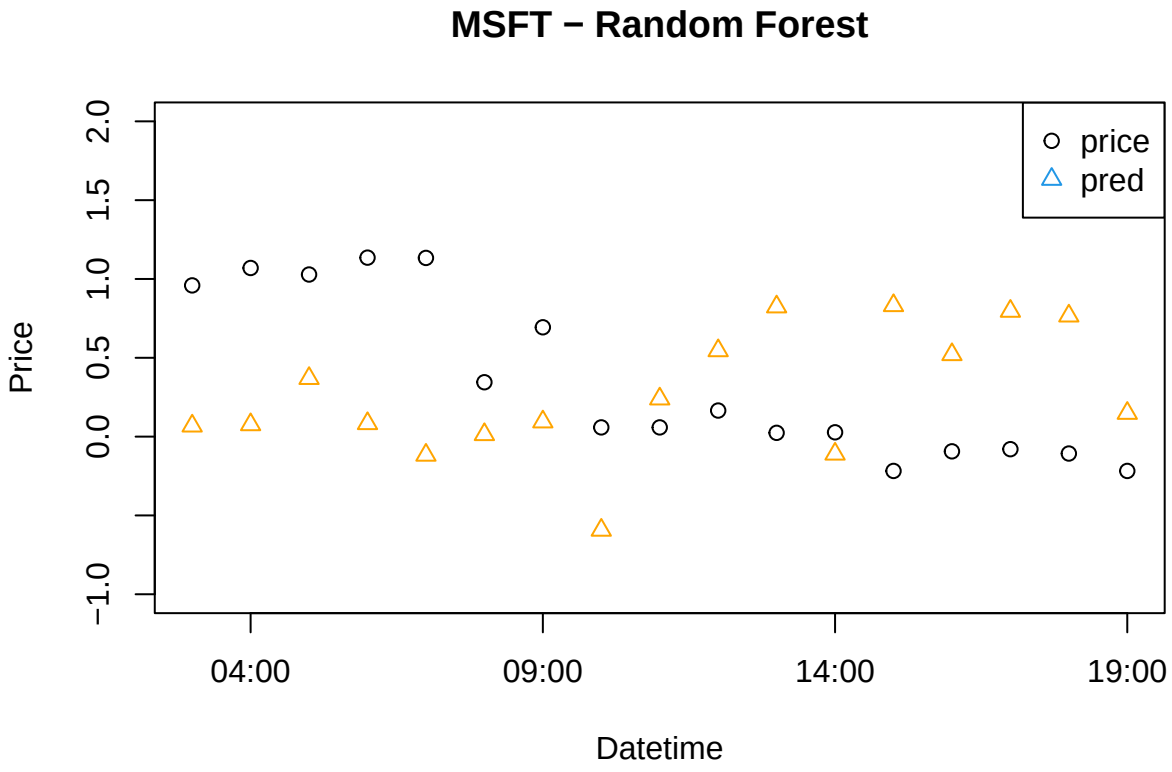
```
##   datetime              price date       time_stock anger anticipation disgust
##   <dttm>                <dbl> <date>     <chr>      <dbl>        <dbl>   <dbl>
## 1 2021-04-12 13:00:00  0.0636 2021-04-12 13:00         54          205      36
## 2 2021-04-12 14:00:00 -0.182  2021-04-12 14:00         43          205      25
## 3 2021-04-12 15:00:00 -0.107  2021-04-12 15:00         44          209      30
## 4 2021-04-12 16:00:00 -0.0935 2021-04-12 16:00         57          197      17
## 5 2021-04-13 09:00:00  0.455  2021-04-13 09:00         10           31       9
## 6 2021-04-13 10:00:00  0.455  2021-04-13 10:00         23           63       9
## # ... 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")
```



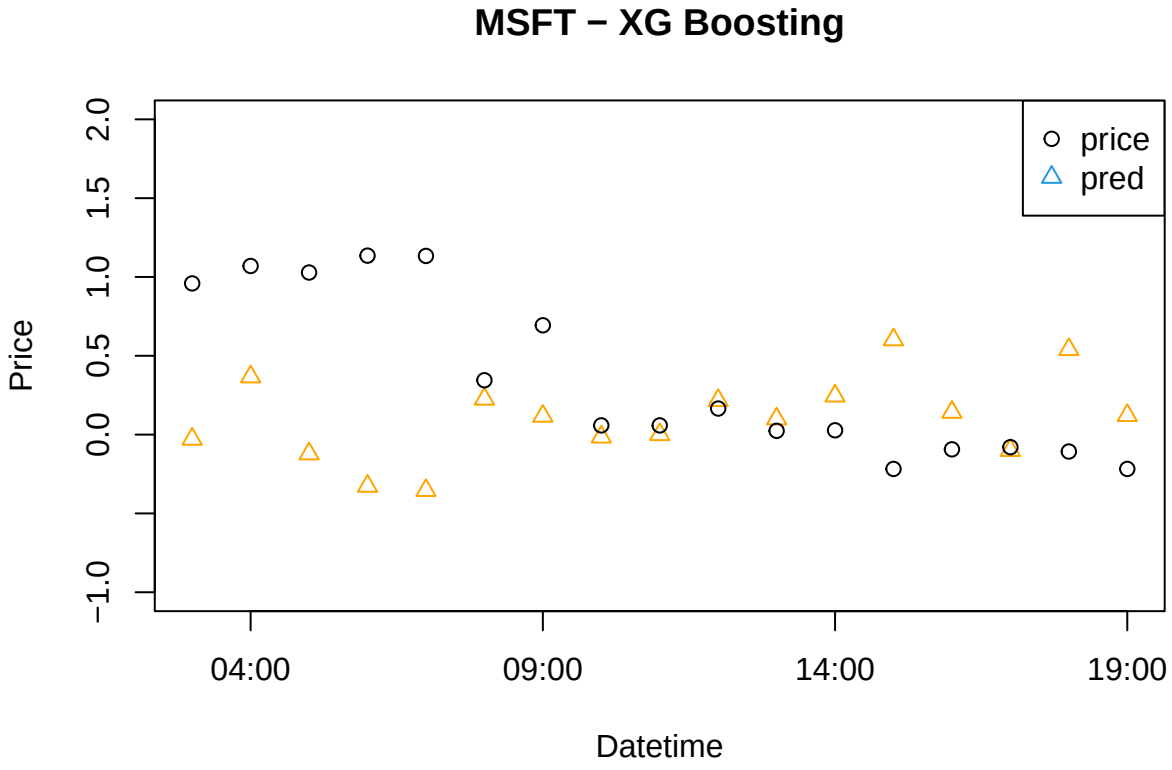

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 3 1
##          1 7 6
##
##                Accuracy : 0.5294
##                  95% CI : (0.2781, 0.7702)
##     No Information Rate : 0.5882
##     P-Value [Acc > NIR] : 0.7716
##
##                   Kappa : 0.1392
##
```

```
##  Mcnemar's Test P-Value : 0.0771
##
##             Sensitivity : 0.3000
##             Specificity : 0.8571
##          Pos Pred Value : 0.7500
##          Neg Pred Value : 0.4615
##              Prevalence : 0.5882
##          Detection Rate : 0.1765
##    Detection Prevalence : 0.2353
##       Balanced Accuracy : 0.5786
##
##        'Positive' Class : 0
##
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 3 1
##          1 7 6
##
##                Accuracy : 0.5294
##                  95% CI : (0.2781, 0.7702)
##     No Information Rate : 0.5882
##     P-Value [Acc > NIR] : 0.7716
##
##                   Kappa : 0.1392
##
##  Mcnemar's Test P-Value : 0.0771
##
##             Sensitivity : 0.3000
##             Specificity : 0.8571
##          Pos Pred Value : 0.7500
##          Neg Pred Value : 0.4615
##              Prevalence : 0.5882
##          Detection Rate : 0.1765
##    Detection Prevalence : 0.2353
##       Balanced Accuracy : 0.5786
##
##        'Positive' Class : 0
##
```




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 6 4
##          1 4 3
##
##                Accuracy : 0.5294
##                  95% CI : (0.2781, 0.7702)
##     No Information Rate : 0.5882
##     P-Value [Acc > NIR] : 0.7716
##
##                   Kappa : 0.0286
##
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.6000
##             Specificity : 0.4286
##          Pos Pred Value : 0.6000
##          Neg Pred Value : 0.4286
##              Prevalence : 0.5882
##          Detection Rate : 0.3529
##    Detection Prevalence : 0.5882
##       Balanced Accuracy : 0.5143
##
##        'Positive' Class : 0
##
```




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 4 2
##          1 6 5
##
##                Accuracy : 0.5294
##                  95% CI : (0.2781, 0.7702)
##     No Information Rate : 0.5882
##     P-Value [Acc > NIR] : 0.7716
##
##                   Kappa : 0.1053
##
##  Mcnemar's Test P-Value : 0.2888
##
##             Sensitivity : 0.4000
##             Specificity : 0.7143
##          Pos Pred Value : 0.6667
##          Neg Pred Value : 0.4545
##              Prevalence : 0.5882
##          Detection Rate : 0.2353
##    Detection Prevalence : 0.3529
##       Balanced Accuracy : 0.5571
##
##        'Positive' Class : 0
##
```