AAPL

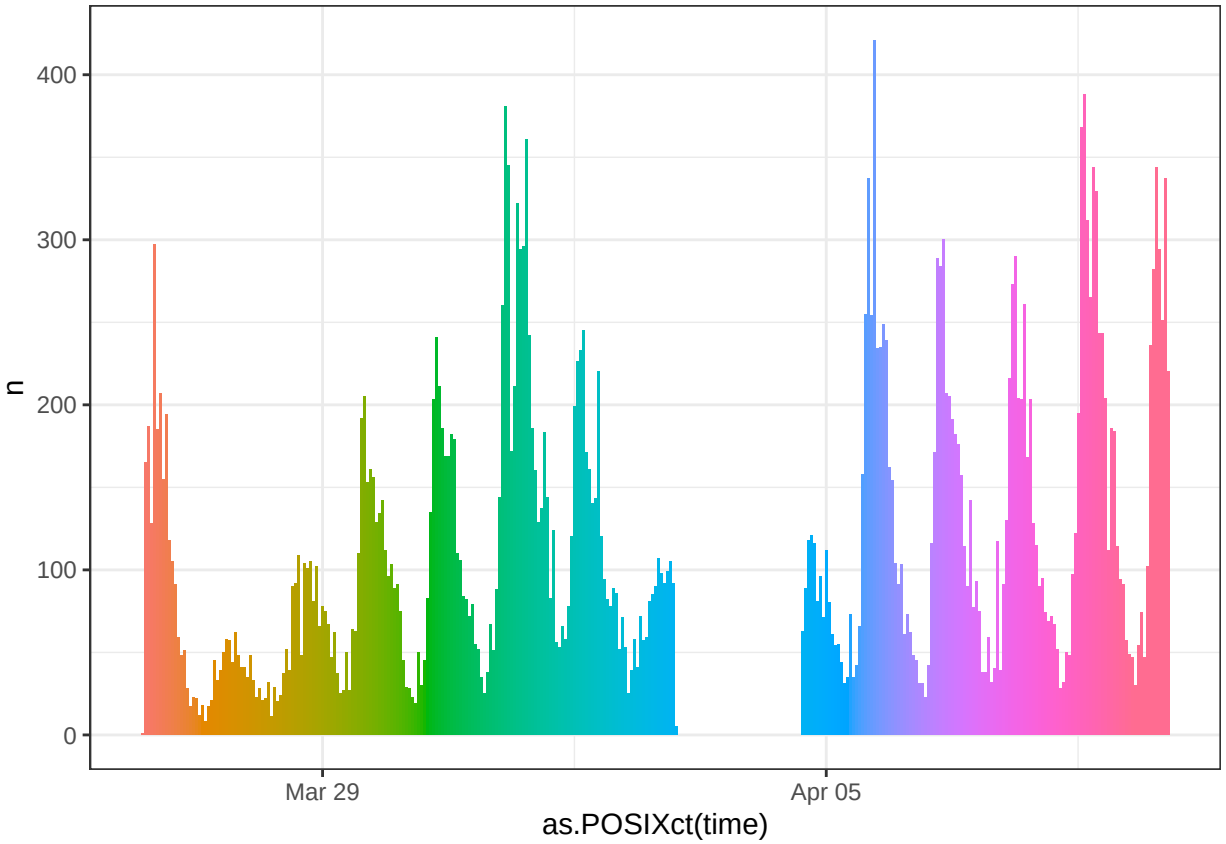
Evan Day

2023-05-08

AAPL

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 2021-03-26 2021-03-26 12:00:00 " Notable open interest changes for March th A~
## 2 2021-03-26 2021-03-26 13:00:00 " Get ahead of the trend here at Xtraders Cong~
## 3 2021-03-26 2021-03-26 14:00:00 " The Secrets They Don t Want You To Know   Yo~
## 4 2021-03-26 2021-03-26 15:00:00 " Collect Per Month In Passive Income With A F~
## 5 2021-03-26 2021-03-26 16:00:00 " BTC Bitcoin Right from Morgan Stanley Privat~
## 6 2021-03-26 2021-03-26 17:00:00 " Most Active Equity Options For Midday Friday~
```

```
## [1] "there are total 302 observation"
```

Sentiment Data frame with bing, afinn, and nrc

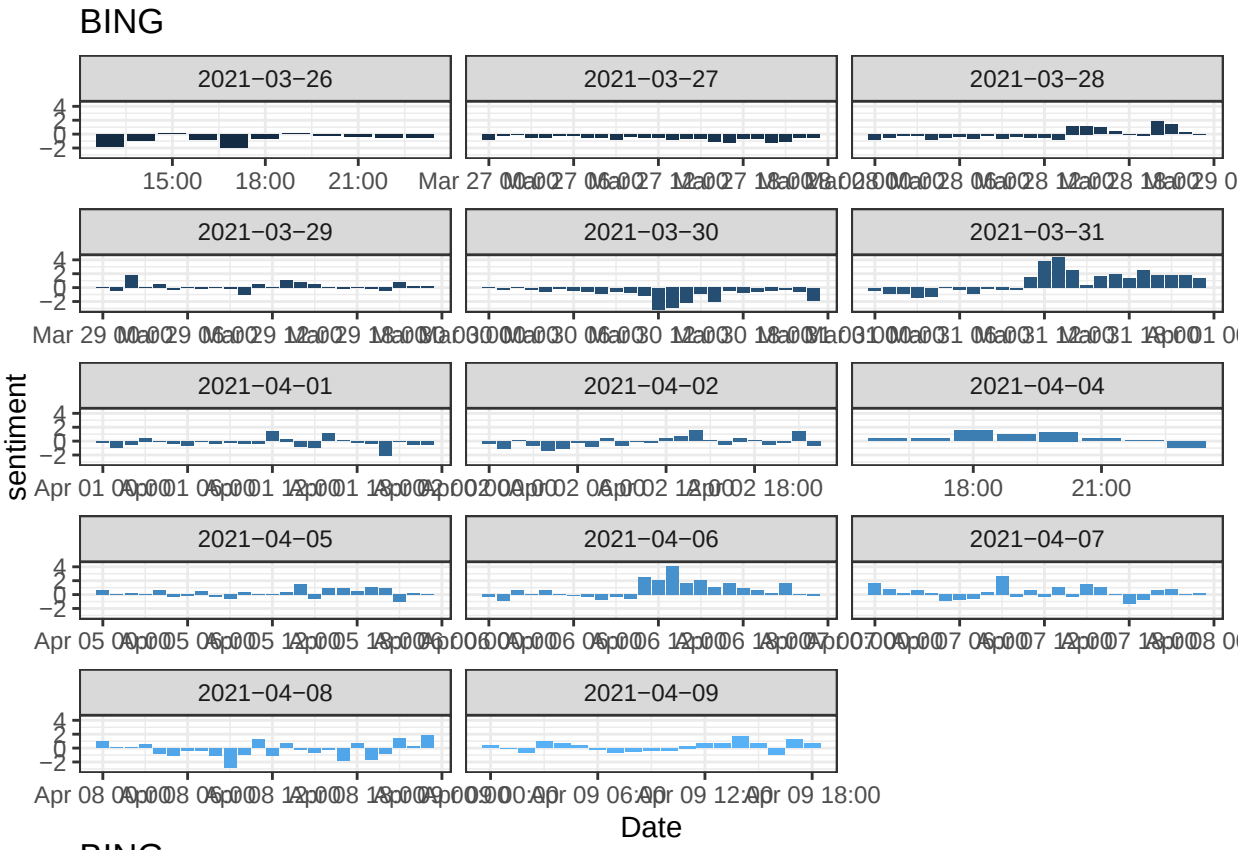
We start with the bing data frame

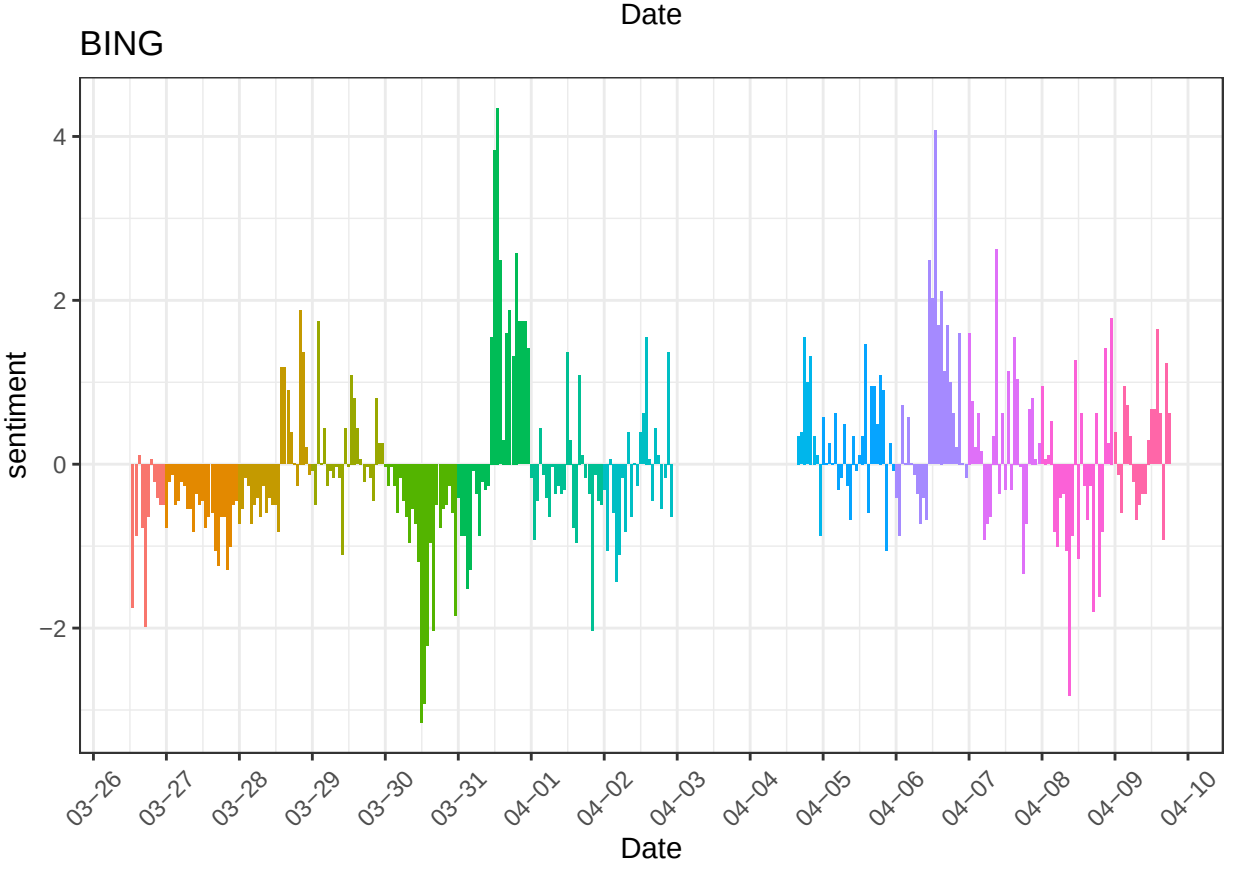
```
## # A tibble: 6 x 3
## # Groups:   date [1]
##   date       time                sentiment
##   <date>     <chr>                   <dbl>
## 1 2021-03-26 2021-03-26 13:00:00       -23
## 2 2021-03-26 2021-03-26 14:00:00        -4
## 3 2021-03-26 2021-03-26 15:00:00        17
## 4 2021-03-26 2021-03-26 16:00:00        -2
## 5 2021-03-26 2021-03-26 17:00:00       -28
## 6 2021-03-26 2021-03-26 18:00:00         1
```

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 2021-03-26 2021-03-26 13:00:00    -1.75
## 2 2021-03-26 2021-03-26 14:00:00    -0.869
## 3 2021-03-26 2021-03-26 15:00:00     0.109
## 4 2021-03-26 2021-03-26 16:00:00    -0.776
## 5 2021-03-26 2021-03-26 17:00:00    -1.99
## 6 2021-03-26 2021-03-26 18:00:00    -0.636
```

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

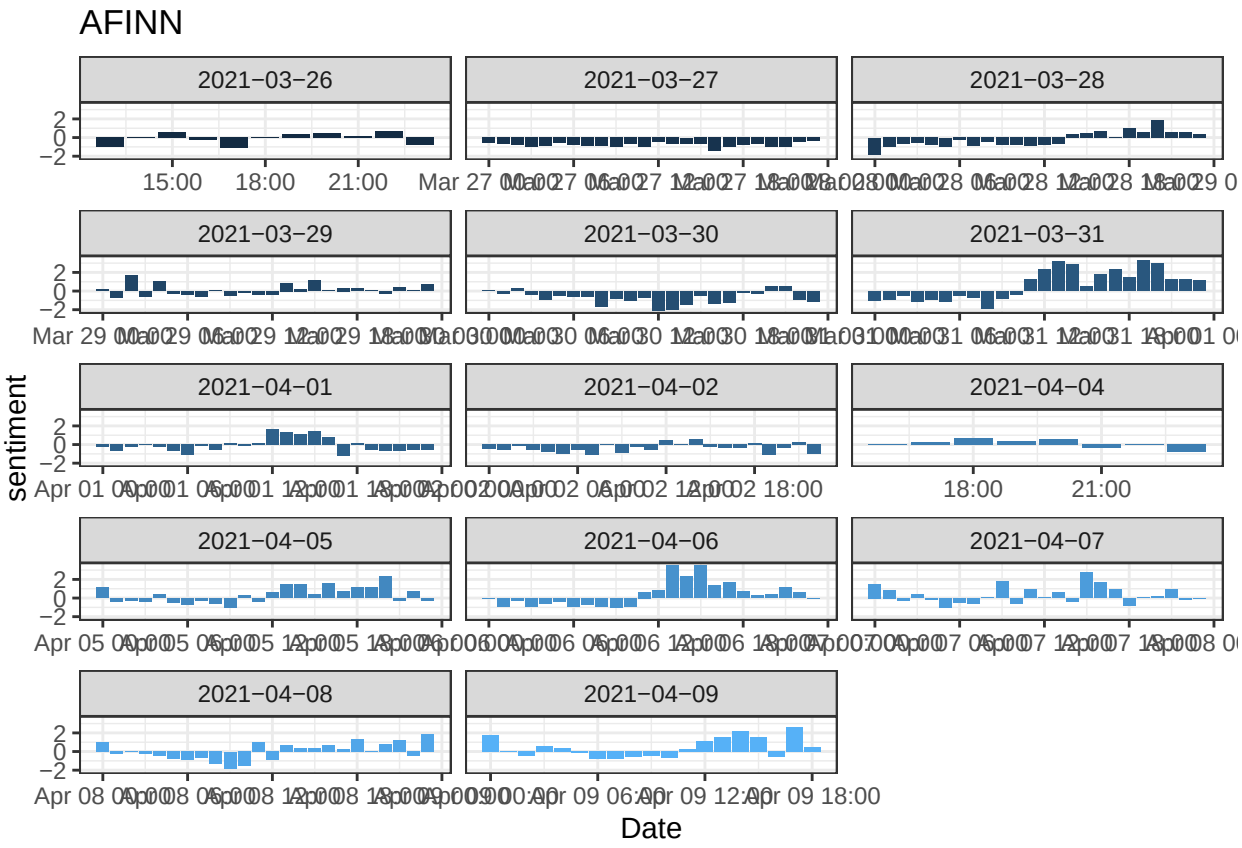


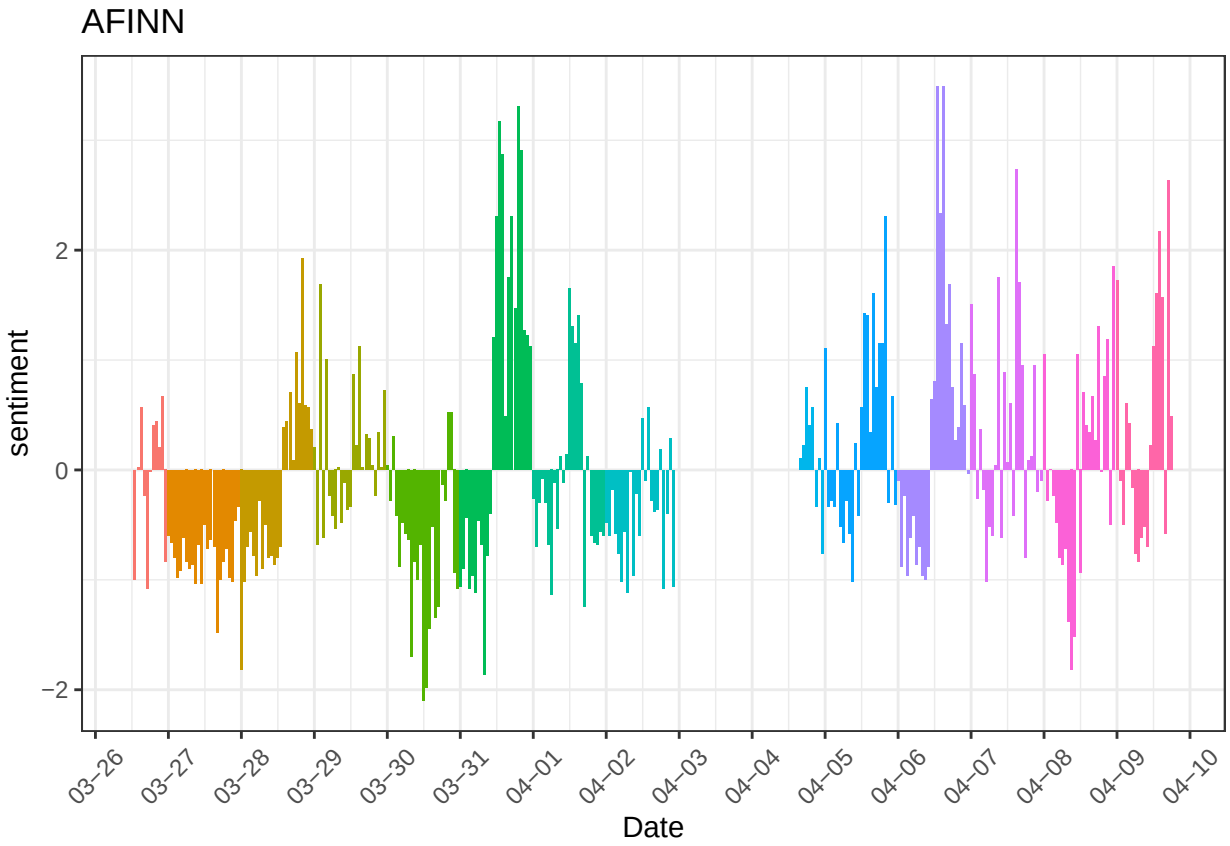


And then, we deal with the afinn sentiment dataframe

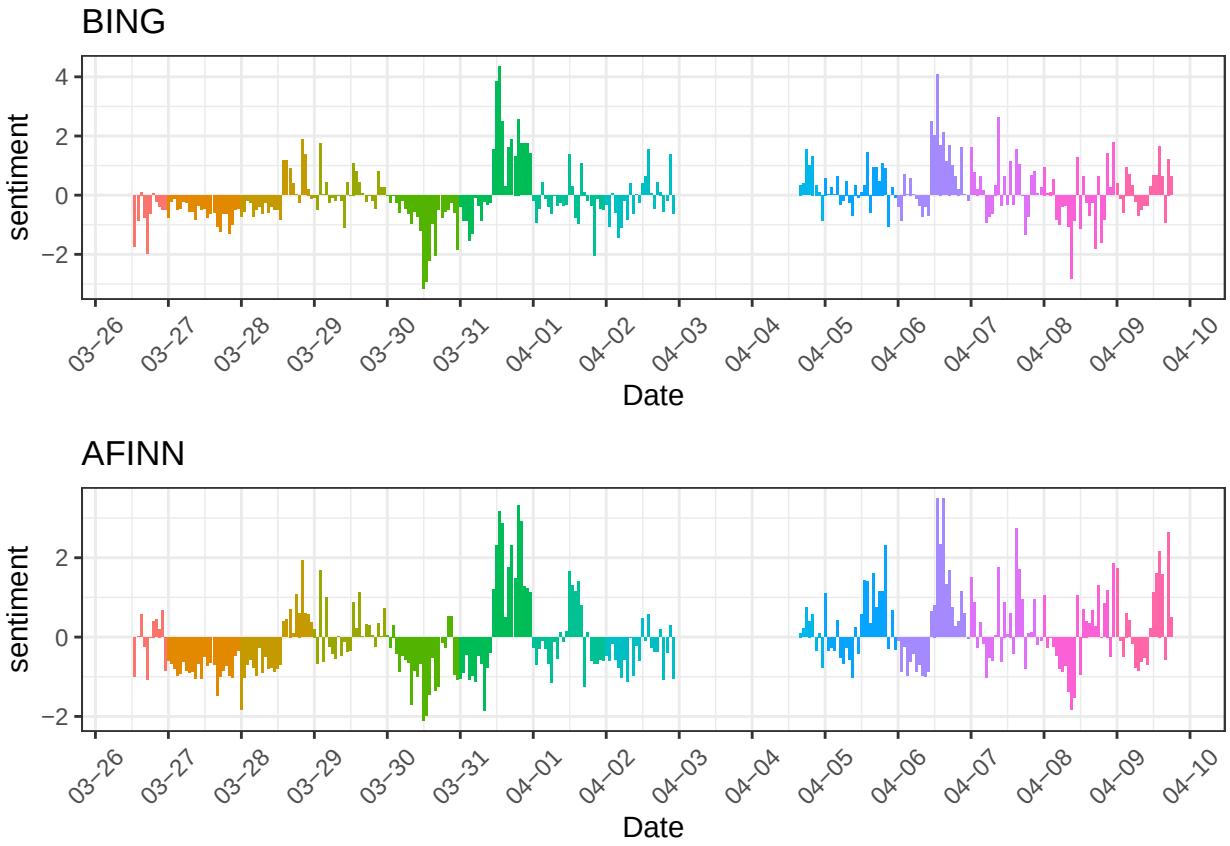
```
## # A tibble: 6 x 3
## # Groups:   date [1]
##   date       time                sentiment
##   <date>     <chr>                   <dbl>
## 1 2021-03-26 2021-03-26 13:00:00   -0.998
## 2 2021-03-26 2021-03-26 14:00:00    0.0244
## 3 2021-03-26 2021-03-26 15:00:00    0.566
## 4 2021-03-26 2021-03-26 16:00:00   -0.236
## 5 2021-03-26 2021-03-26 17:00:00   -1.08
## 6 2021-03-26 2021-03-26 18:00:00   -0.0157
```

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





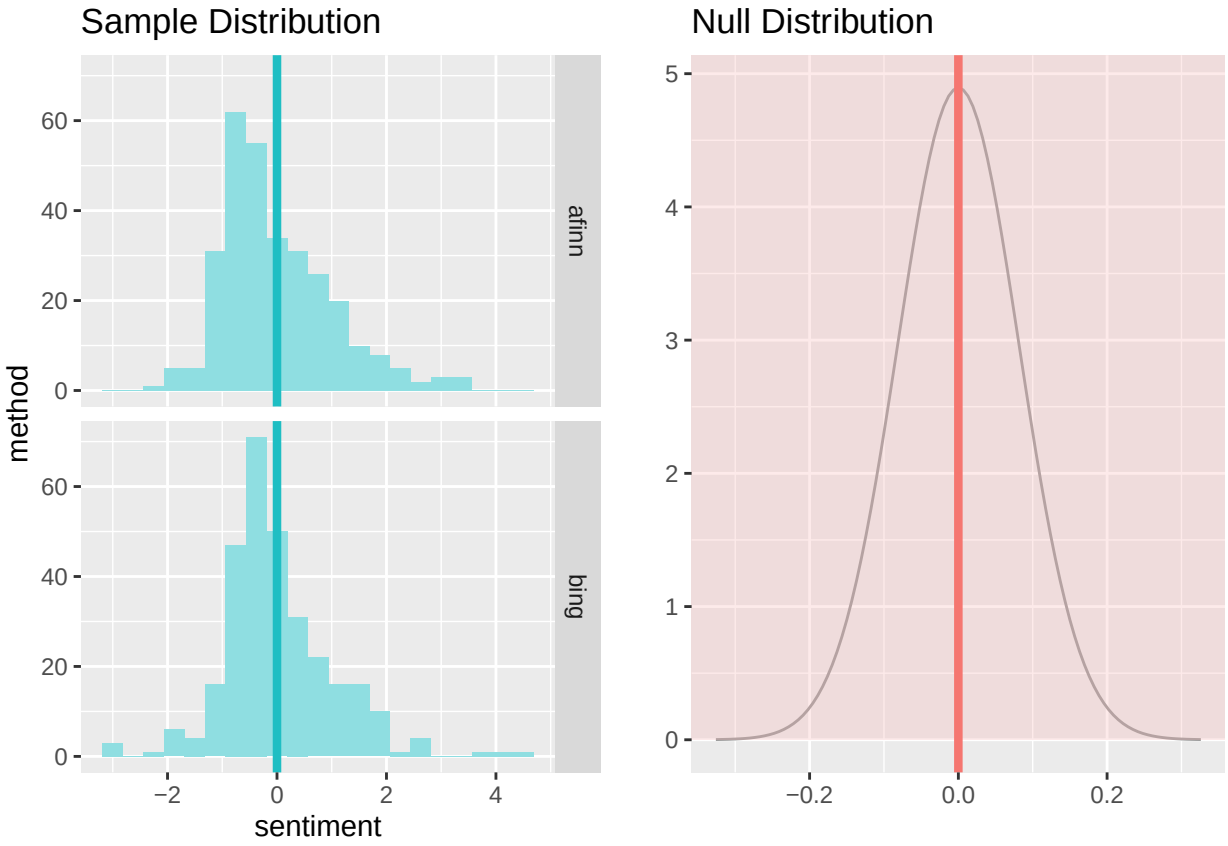
we compare the two sentiment plot together



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 = 301, y_bar_afinn = 0, s_afinn = 1
## n_bing = 301, y_bar_bing = 0, s_bing = 1
## H0: mu_afinn =  mu_bing
## HA: mu_afinn != mu_bing
## t = 0, df = 300
## 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
```

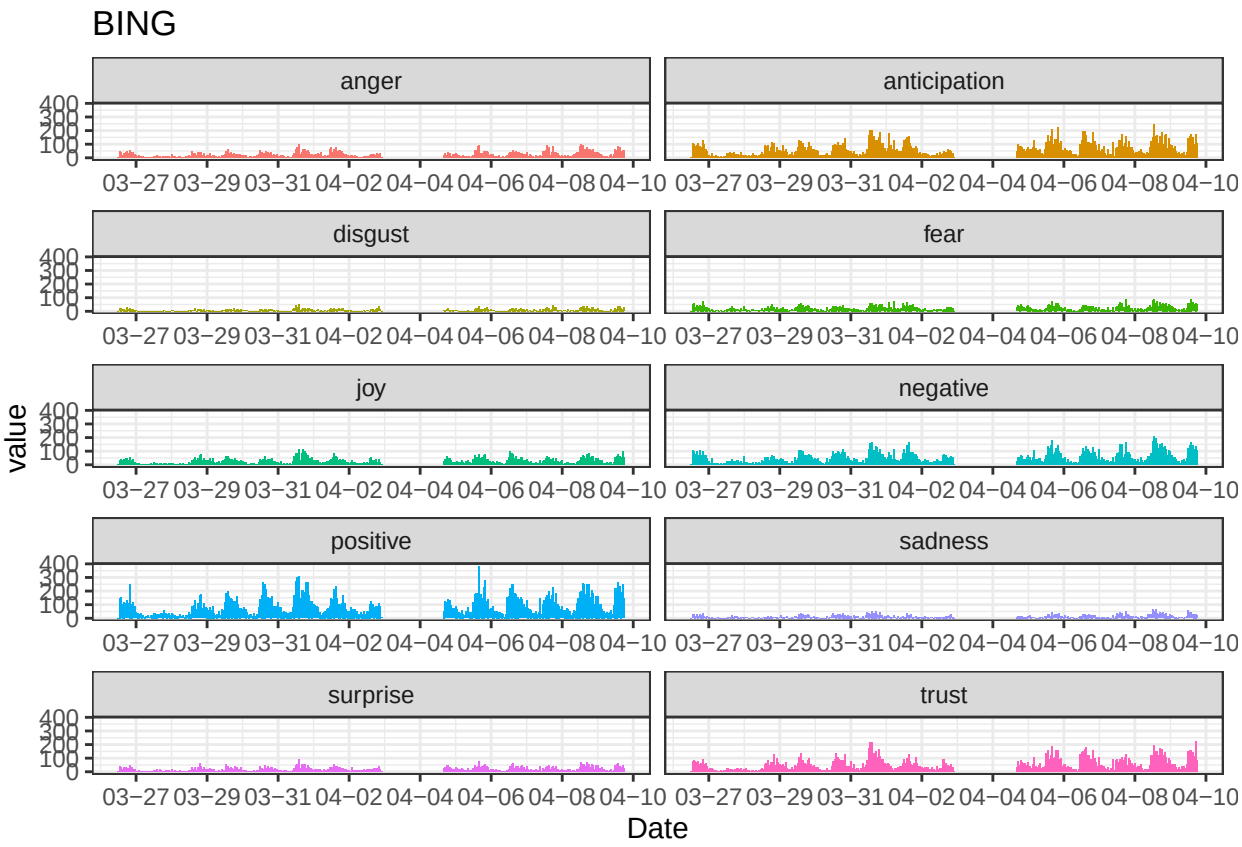
```
##
##  Two-sample Kolmogorov-Smirnov test
##
## data:  bing_afinn$bing and bing_afinn$afinn
## D = 0.11296, p-value = 0.04296
## alternative hypothesis: two-sided
```

Then, here is the method with nrc lexicon

```
##
## Attaching package: 'reshape2'
```

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

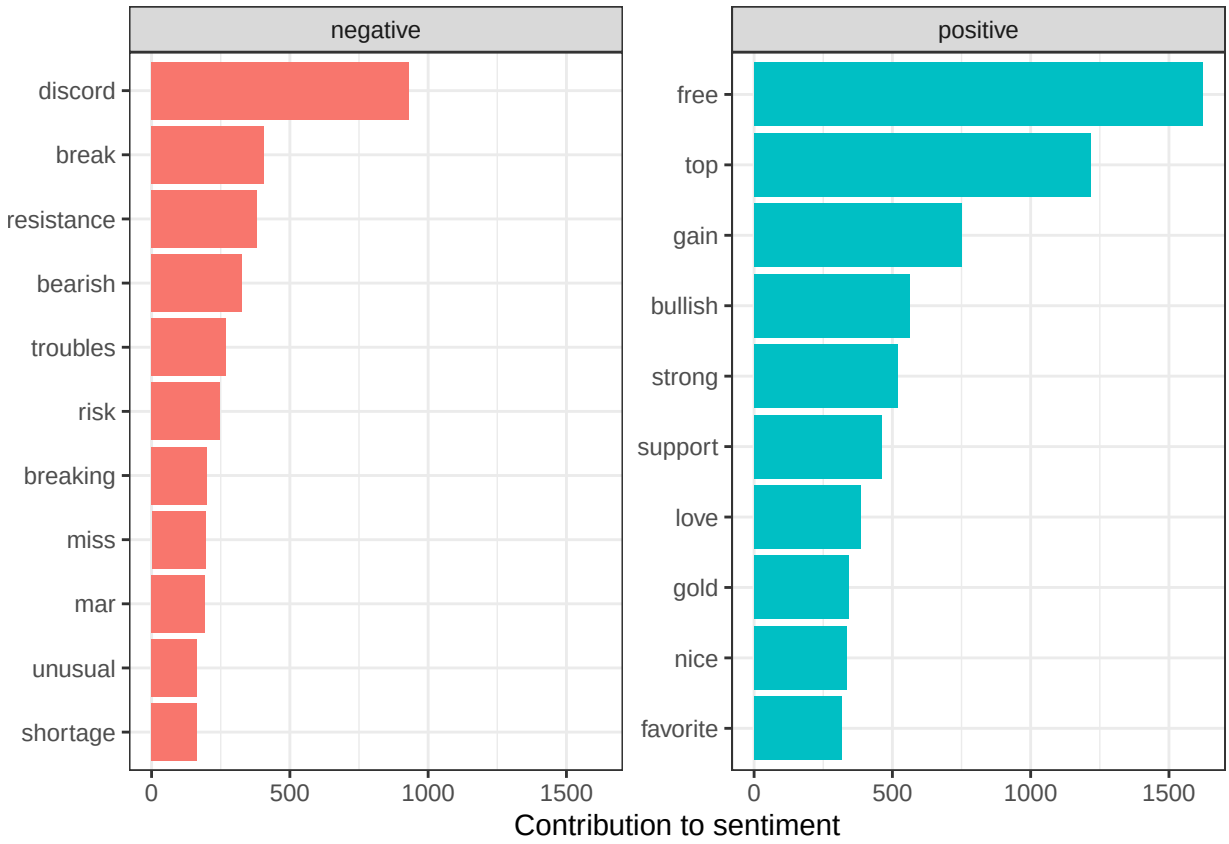
```
## No id variables; using all as measure variables
```



---

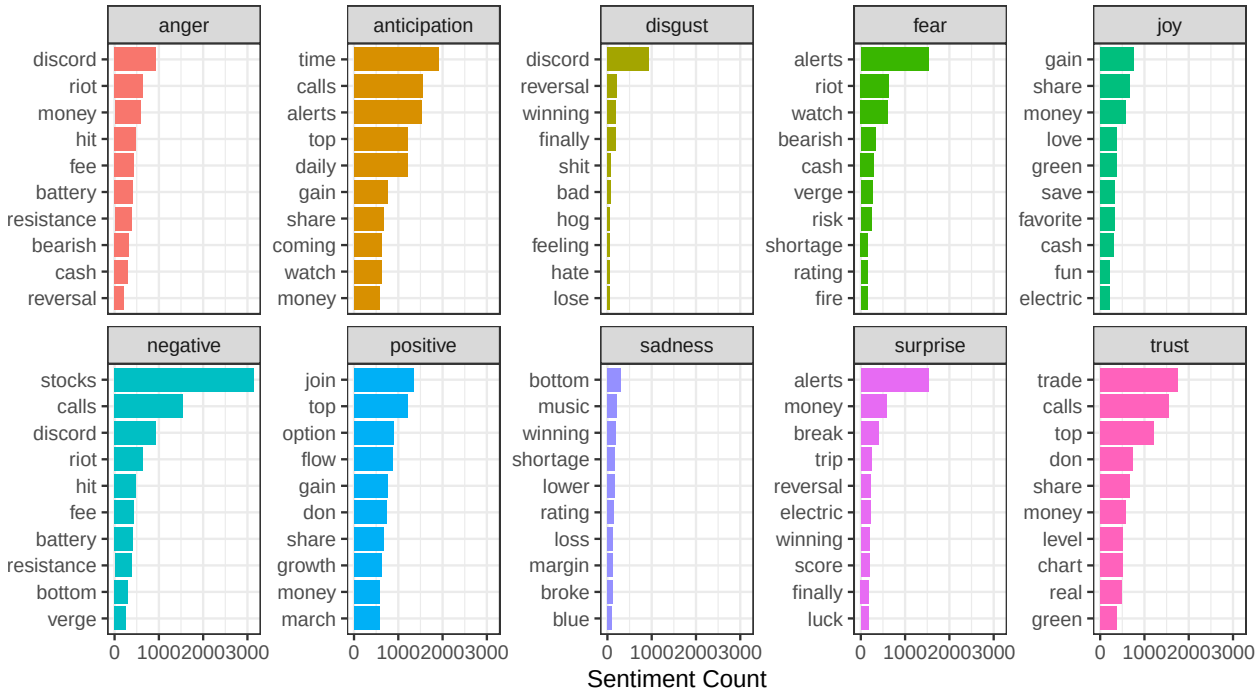
top 10 bing_words Count

```
## Joining, by = "word"
## Selecting by n
```



top 10 nrc_words Count

```
## Joining, by = "word"
## Selecting by n
```



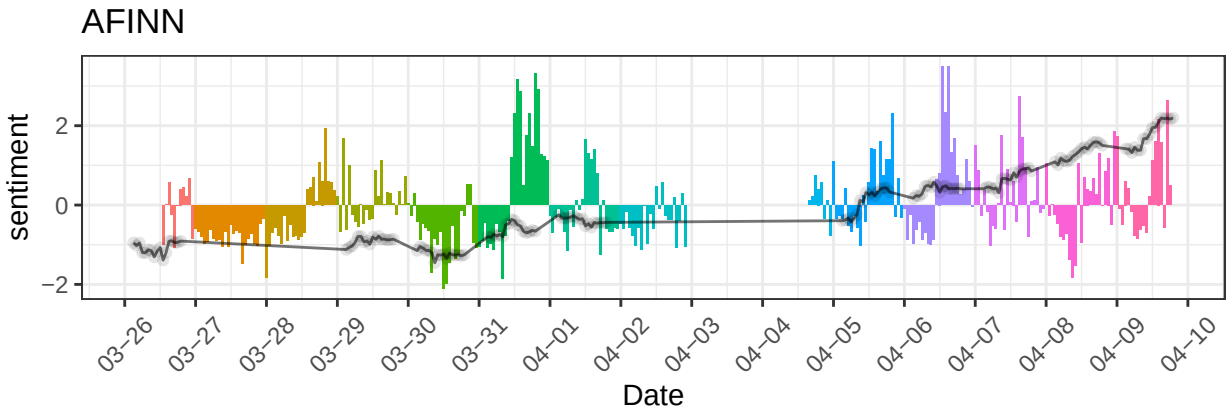
AAPL

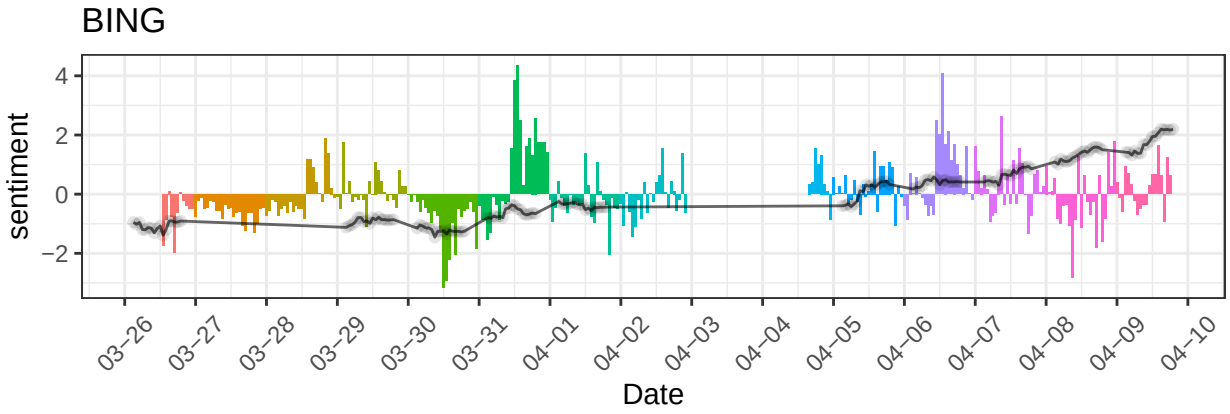
Stock Information

```
## # A tibble: 6 x 2
##   time                price
##   <chr>               <dbl>
## 1 2021-03-26 03:00:00  121.
## 2 2021-03-26 04:00:00  121.
## 3 2021-03-26 05:00:00  121.
## 4 2021-03-26 06:00:00  120.
## 5 2021-03-26 07:00:00  120.
## 6 2021-03-26 08:00:00  120.
```

normalize the price data:

```
## # A tibble: 6 x 2
##   time                 price
##   <chr>                <dbl>
## 1 2021-03-26 03:00:00 -0.950
## 2 2021-03-26 04:00:00 -1.02
## 3 2021-03-26 05:00:00 -0.956
## 4 2021-03-26 06:00:00 -1.20
## 5 2021-03-26 07:00:00 -1.20
## 6 2021-03-26 08:00:00 -1.13
```





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 2021-03-26 17:00:00 -0.948 2021-03-26 17:00         42          109       9
## 2 2021-03-26 18:00:00 -0.937 2021-03-26 18:00         46           81      33
## 3 2021-03-26 19:00:00 -0.908 2021-03-26 19:00         33           77      12
## 4 2021-03-29 03:00:00 -1.12  2021-03-29 03:00         12           50       1
## 5 2021-03-29 04:00:00 -1.07  2021-03-29 04:00         15           61       7
## 6 2021-03-29 05:00:00 -1.03  2021-03-29 05:00         18           43       4
## # ... 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 2021-03-26 12:00:00 -1.07  2021-03-26 12:00          0            0       0
## 2 2021-03-26 13:00:00 -1.39  2021-03-26 13:00         50          110      20
## 3 2021-03-26 14:00:00 -1.20  2021-03-26 14:00         44           99      26
## 4 2021-03-26 15:00:00 -0.908 2021-03-26 15:00         20           82      12
## 5 2021-03-26 16:00:00 -0.903 2021-03-26 16:00         36           89      19
## 6 2021-03-29 09:00:00 -0.932 2021-03-29 09:00          6           16       0
## # ... with 8 more variables: fear <dbl>, joy <dbl>, negative <dbl>,
## #   positive <dbl>, sadness <dbl>, surprise <dbl>, trust <dbl>, state <chr>
```

AAPL 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
## -1.7375 -0.7492 -0.1033  0.6058  1.8985
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.04924    0.07184   0.685 0.494193
```

```
## anger         0.26117    0.25585   1.021 0.309016
## anticipation  0.30958    0.28936   1.070 0.286403
## disgust       0.34785    0.17070   2.038 0.043339 *
## fear          0.10471    0.20165   0.519 0.604330
## joy          -0.49209    0.19673  -2.501 0.013453 *
## negative     -0.45541    0.33773  -1.348 0.179568
## positive     -0.66249    0.26733  -2.478 0.014321 *
## sadness       0.10412    0.16537   0.630 0.529914
## surprise     -0.14826    0.21509  -0.689 0.491705
## trust         0.92080    0.26867   3.427 0.000788 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9088 on 149 degrees of freedom
## Multiple R-squared:  0.1925, Adjusted R-squared:  0.1383
## F-statistic: 3.552 on 10 and 149 DF,  p-value: 0.0003068

## 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:0.837273
## [2]  train-rmse:0.663605
## [3]  train-rmse:0.545285
## [4]  train-rmse:0.451916
## [5]  train-rmse:0.383799
## [6]  train-rmse:0.341505
## [7]  train-rmse:0.297456
## [8]  train-rmse:0.265506
## [9]  train-rmse:0.244323
```

```
## [10] train-rmse:0.211398
## [11] train-rmse:0.179870
## [12] train-rmse:0.165826
## [13] train-rmse:0.157928
## [14] train-rmse:0.137630
## [15] train-rmse:0.121883
## [16] train-rmse:0.115046
## [17] train-rmse:0.105106
## [18] train-rmse:0.093283
## [19] train-rmse:0.089738
## [20] train-rmse:0.079369
## [21] train-rmse:0.068433
## [22] train-rmse:0.066362
## [23] train-rmse:0.056792
## [24] train-rmse:0.052996
## [25] train-rmse:0.051860
## [26] train-rmse:0.048802
## [27] train-rmse:0.041750
## [28] train-rmse:0.040405
## [29] train-rmse:0.036808
## [30] train-rmse:0.034454
## [31] train-rmse:0.031503
## [32] train-rmse:0.029369
## [33] train-rmse:0.027207
## [34] train-rmse:0.024150
## [35] train-rmse:0.021868
## [36] train-rmse:0.019786
## [37] train-rmse:0.017829
## [38] train-rmse:0.016334
## [39] train-rmse:0.015523
## [40] train-rmse:0.013968
## [41] train-rmse:0.012239
## [42] train-rmse:0.010660
## [43] train-rmse:0.010026
## [44] train-rmse:0.008818
## [45] train-rmse:0.008467
## [46] train-rmse:0.007309
## [47] train-rmse:0.006529
## [48] train-rmse:0.005851
## [49] train-rmse:0.005531
## [50] train-rmse:0.004842
## [51] train-rmse:0.004277
## [52] train-rmse:0.003918
## [53] train-rmse:0.003777
## [54] train-rmse:0.003595
## [55] train-rmse:0.003238
## [56] train-rmse:0.003082
## [57] train-rmse:0.002728
## [58] train-rmse:0.002604
## [59] train-rmse:0.002288
## [60] train-rmse:0.002160
## [61] train-rmse:0.002041
## [62] train-rmse:0.001838
## [63] train-rmse:0.001693
```

```
## [64] train-rmse:0.001572
## [65] train-rmse:0.001438
## [66] train-rmse:0.001321
## [67] train-rmse:0.001219
## [68] train-rmse:0.001120
## [69] train-rmse:0.001071
## [70] train-rmse:0.001032
## [71] train-rmse:0.000934
## [72] train-rmse:0.000880
## [73] train-rmse:0.000880
## [74] train-rmse:0.000880
## [75] train-rmse:0.000880
## [76] train-rmse:0.000880
## [77] train-rmse:0.000880
## [78] train-rmse:0.000880
## [79] train-rmse:0.000880
## [80] train-rmse:0.000880
## [81] train-rmse:0.000880
## [82] train-rmse:0.000880
## [83] train-rmse:0.000880
## [84] train-rmse:0.000880
## [85] train-rmse:0.000880
## [86] train-rmse:0.000880
## [87] train-rmse:0.000880
## [88] train-rmse:0.000880
## [89] train-rmse:0.000880
## [90] train-rmse:0.000880
## [91] train-rmse:0.000880
## [92] train-rmse:0.000880
## [93] train-rmse:0.000880
## [94] train-rmse:0.000880
## [95] train-rmse:0.000880
## [96] train-rmse:0.000880
## [97] train-rmse:0.000880
## [98] train-rmse:0.000880
## [99] train-rmse:0.000880
## [100]    train-rmse:0.000880
```

```
## # A tibble: 160 x 1
##     price
##     <dbl>
##  1 -0.948
##  2 -0.937
##  3 -0.908
##  4 -1.12
##  5 -1.07
##  6 -1.03
##  7 -0.945
##  8 -0.790
##  9 -0.787
## 10 -0.879
## # ... with 150 more rows
```

2. this is the model for close recording

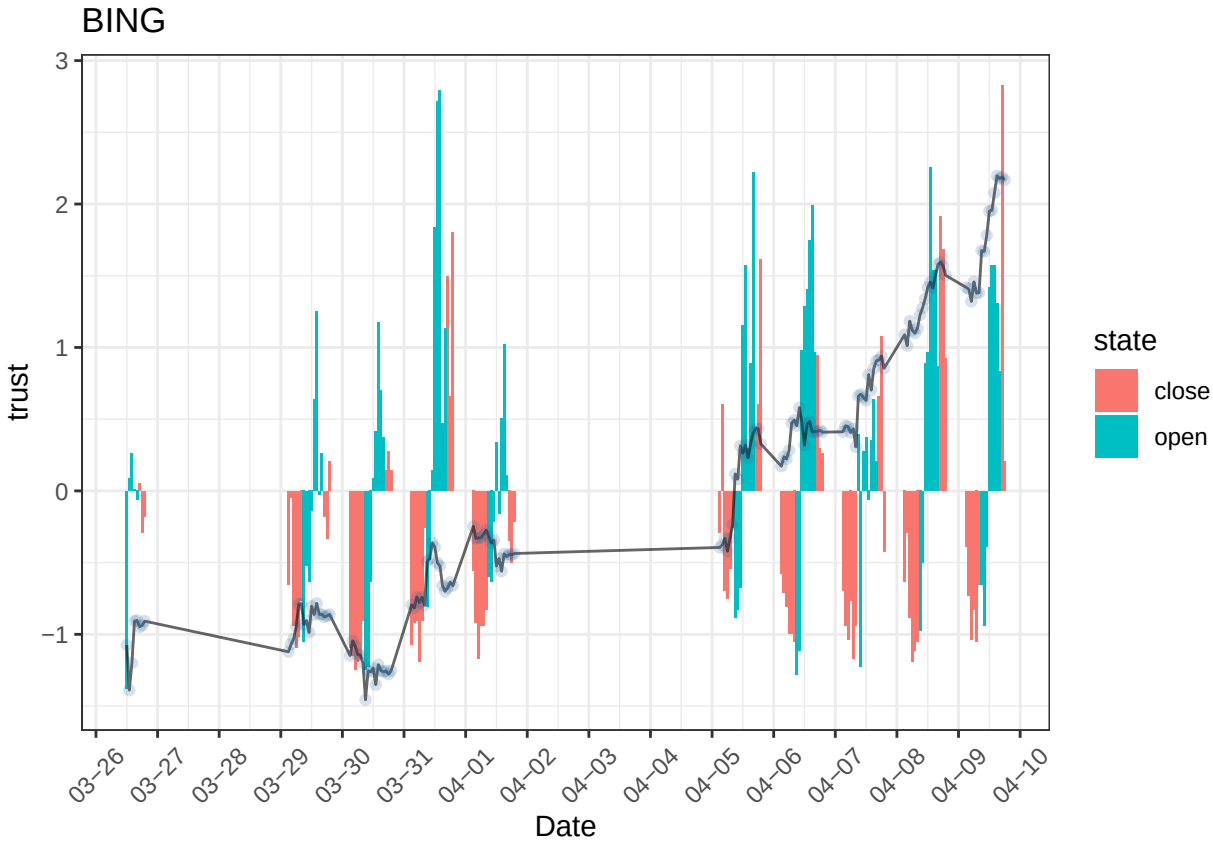
```
##
## Call:
## lm(formula = price ~ anger + anticipation + disgust + fear +
##     joy + negative + positive + sadness + surprise + trust, data = full_nrc[which(full_nrc$state ==
##     "close"), ])
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.40711 -0.65572 -0.06639  0.48256  1.84127
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   0.18272    0.11878   1.538   0.1284
## anger         0.73518    0.49632   1.481   0.1429
## anticipation  0.51785    0.45852   1.129   0.2625
## disgust       0.44564    0.26273   1.696   0.0942 .
## fear         -0.11797    0.34516  -0.342   0.7335
## joy          -0.76366    0.31511  -2.423   0.0179 *
## negative     -1.02955    0.55128  -1.868   0.0659 .
## positive     -0.78786    0.53133  -1.483   0.1425
## sadness      -0.02186    0.33375  -0.066   0.9480
## surprise      0.23530    0.34081   0.690   0.4922
## trust         1.24245    0.46984   2.644   0.0100 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.8777 on 72 degrees of freedom
## Multiple R-squared:  0.2148, Adjusted R-squared:  0.1058
## F-statistic:  1.97 on 10 and 72 DF,  p-value: 0.04938
```

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
## -1.89993 -0.64692  0.02912  0.59126  1.84910
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  -0.010616   0.119399  -0.089   0.9294
## anger        -0.125131   0.361404  -0.346   0.7303
## anticipation  0.460781   0.441379   1.044   0.3003
## disgust       0.493966   0.264976   1.864   0.0667 .
## fear          0.177234   0.277279   0.639   0.5249
## joy          -0.522642   0.311549  -1.678   0.0982 .
## negative     -0.003953   0.481702  -0.008   0.9935
## positive     -0.630636   0.344168  -1.832   0.0714 .
## sadness       0.191901   0.200071   0.959   0.3410
## surprise     -0.448921   0.302613  -1.483   0.1427
```

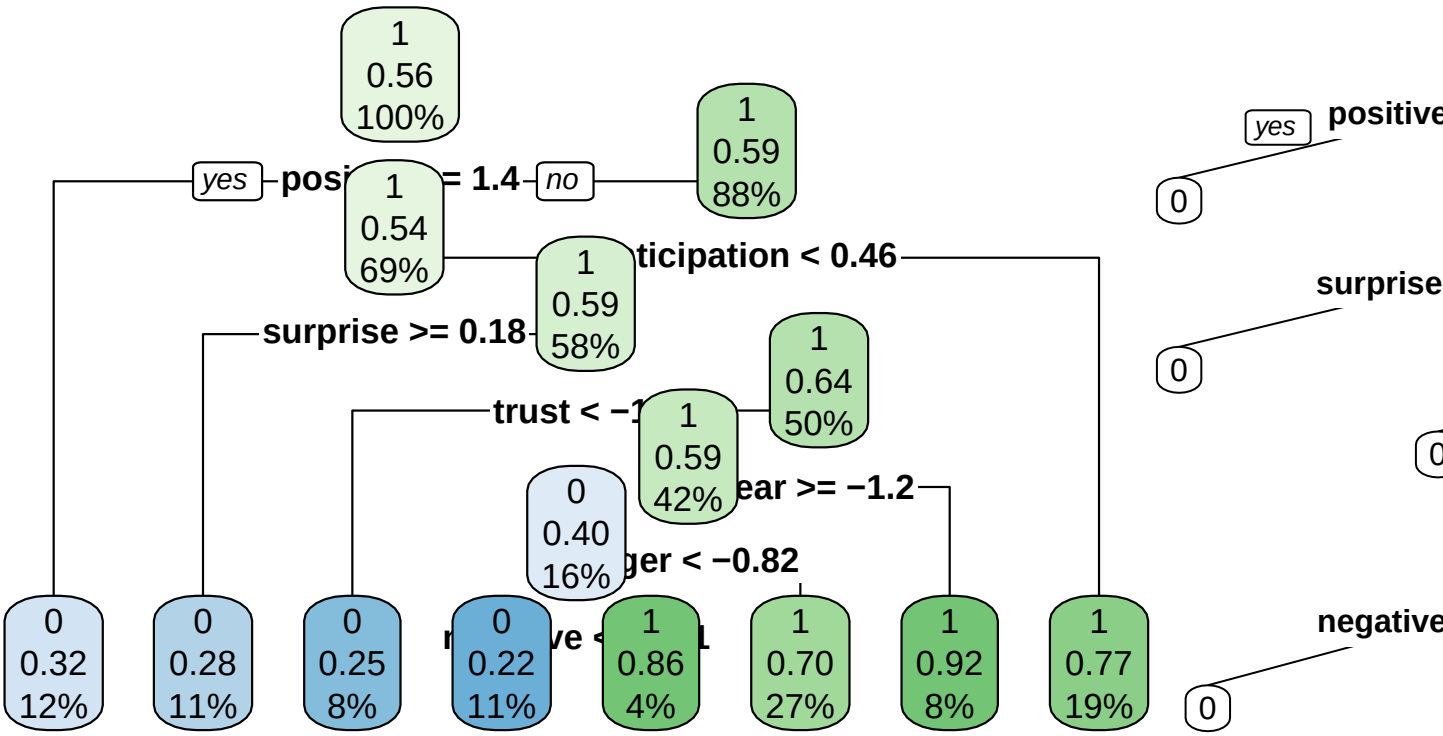
```
## trust          0.692049   0.367765   1.882   0.0643 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9468 on 66 degrees of freedom
## Multiple R-squared:  0.2724, Adjusted R-squared:  0.1622
## F-statistic: 2.471 on 10 and 66 DF,  p-value: 0.01406
```

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 49 18
##          1 22 71
##
##                Accuracy : 0.75
##                  95% CI : (0.6755, 0.815)
##     No Information Rate : 0.5562
##     P-Value [Acc > NIR] : 3.021e-07
##
##                   Kappa : 0.4907
##
##  Mcnemar's Test P-Value : 0.6353
##
##             Sensitivity : 0.6901
##             Specificity : 0.7978
##          Pos Pred Value : 0.7313
##          Neg Pred Value : 0.7634
##              Prevalence : 0.4437
##          Detection Rate : 0.3063
##    Detection Prevalence : 0.4188
##       Balanced Accuracy : 0.7439
##
##        'Positive' Class : 0
##

## [1]  train-logloss:0.651964
## [2]  train-logloss:0.624090
```

```
## [3]  train-logloss:0.604810
## [4]  train-logloss:0.590351
## [5]  train-logloss:0.581105
## [6]  train-logloss:0.574626
## [7]  train-logloss:0.566785
## [8]  train-logloss:0.560828
## [9]  train-logloss:0.555481
## [10] train-logloss:0.551756
## [11] train-logloss:0.548191
## [12] train-logloss:0.545540
## [13] train-logloss:0.542852
## [14] train-logloss:0.539975
## [15] train-logloss:0.538564
## [16] train-logloss:0.537530
## [17] train-logloss:0.535800
## [18] train-logloss:0.534164
## [19] train-logloss:0.532990
## [20] train-logloss:0.532306
## [21] train-logloss:0.530532
## [22] train-logloss:0.528881
## [23] train-logloss:0.527990
## [24] train-logloss:0.527507
## [25] train-logloss:0.526182
## [26] train-logloss:0.525658
## [27] train-logloss:0.524763
## [28] train-logloss:0.524068
## [29] train-logloss:0.523467
## [30] train-logloss:0.523006
## [31] train-logloss:0.522699
## [32] train-logloss:0.522356
## [33] train-logloss:0.521979
## [34] train-logloss:0.521758
## [35] train-logloss:0.521387
## [36] train-logloss:0.521024
## [37] train-logloss:0.520796
## [38] train-logloss:0.520572
## [39] train-logloss:0.520303
## [40] train-logloss:0.520124
## [41] train-logloss:0.519956
## [42] train-logloss:0.519579
## [43] train-logloss:0.519319
## [44] train-logloss:0.519216
## [45] train-logloss:0.519009
## [46] train-logloss:0.518663
## [47] train-logloss:0.518467
## [48] train-logloss:0.518204
## [49] train-logloss:0.517974
## [50] train-logloss:0.517873
## [51] train-logloss:0.517645
## [52] train-logloss:0.517531
## [53] train-logloss:0.517375
## [54] train-logloss:0.517291
## [55] train-logloss:0.517184
## [56] train-logloss:0.516999
```

```
## [57] train-logloss:0.516866
## [58] train-logloss:0.516770
## [59] train-logloss:0.516676
## [60] train-logloss:0.516597
## [61] train-logloss:0.516492
## [62] train-logloss:0.516383
## [63] train-logloss:0.516305
## [64] train-logloss:0.516203
## [65] train-logloss:0.516094
## [66] train-logloss:0.516035
## [67] train-logloss:0.515978
## [68] train-logloss:0.515907
## [69] train-logloss:0.515833
## [70] train-logloss:0.515698
## [71] train-logloss:0.515646
## [72] train-logloss:0.515595
## [73] train-logloss:0.515544
## [74] train-logloss:0.515494
## [75] train-logloss:0.515425
## [76] train-logloss:0.515294
## [77] train-logloss:0.515253
## [78] train-logloss:0.515190
## [79] train-logloss:0.515152
## [80] train-logloss:0.515112
## [81] train-logloss:0.515041
## [82] train-logloss:0.515011
## [83] train-logloss:0.514965
## [84] train-logloss:0.514914
## [85] train-logloss:0.514856
## [86] train-logloss:0.514820
## [87] train-logloss:0.514772
## [88] train-logloss:0.514719
## [89] train-logloss:0.514673
## [90] train-logloss:0.514594
## [91] train-logloss:0.514560
## [92] train-logloss:0.514514
## [93] train-logloss:0.514442
## [94] train-logloss:0.514412
## [95] train-logloss:0.514374
## [96] train-logloss:0.514341
## [97] train-logloss:0.514297
## [98] train-logloss:0.514262
## [99] train-logloss:0.514239
## [100]    train-logloss:0.514212
```

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
## -1.83285 -0.46605  0.01126  0.63378  1.04983
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.583516   0.152057  -3.837 0.000257 ***
## negative     0.014161   0.004165   3.400 0.001082 **
## positive    -0.006533   0.003408  -1.917 0.059051 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7039 on 75 degrees of freedom
##   (81 observations deleted due to missingness)
## Multiple R-squared:  0.1607, Adjusted R-squared:  0.1383
## F-statistic:  7.18 on 2 and 75 DF,  p-value: 0.001402

##
## Call:
## lm(formula = price ~ negative + positive, data = full_bing_close)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.4497 -0.7788 -0.1623  0.6357  1.8860
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.297213   0.174894  -1.699   0.0931 .
## negative     0.001741   0.006787   0.257   0.7982
## positive     0.004999   0.005940   0.842   0.4025
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9149 on 80 degrees of freedom
## Multiple R-squared:  0.05206,    Adjusted R-squared:  0.02836
## F-statistic: 2.197 on 2 and 80 DF,  p-value: 0.1178

##
## Call:
## lm(formula = price ~ negative + positive, data = full_bing_open)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -1.33428 -0.95712 -0.05755  0.63993  2.18249
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.165283   0.254024  -0.651   0.5173
## negative    -0.004004   0.005035  -0.795   0.4291
## positive     0.006616   0.003828   1.728   0.0882 .
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.023 on 73 degrees of freedom
## Multiple R-squared:  0.0446, Adjusted R-squared:  0.01843
## F-statistic: 1.704 on 2 and 73 DF,  p-value: 0.1891

## 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
## -2.17391 -0.55763  0.05258  0.67679  1.11907
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.335215   0.089361  -3.751 0.000342 ***
## sentiment    0.009936   0.076035   0.131 0.896378
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.7632 on 76 degrees of freedom
##   (81 observations deleted due to missingness)
## Multiple R-squared:  0.0002246,  Adjusted R-squared:  -0.01293
## F-statistic: 0.01708 on 1 and 76 DF,  p-value: 0.8964

##
## Call:
## lm(formula = price ~ sentiment, data = full_afinn_close)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3393 -0.8607 -0.2293  0.6310  2.0392
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.03297    0.10242   0.322   0.7484
## sentiment    0.19501    0.11144   1.750   0.0839 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.9167 on 81 degrees of freedom
## Multiple R-squared:  0.03643,    Adjusted R-squared:  0.02453
## F-statistic: 3.062 on 1 and 81 DF,  p-value: 0.08393

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

```
## Residuals:
##     Min      1Q  Median      3Q     Max
## -1.3521 -0.9056 -0.1274  0.6688  2.2356
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.04336    0.12282   0.353    0.725
## sentiment    0.17603    0.09217   1.910    0.060 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.015 on 74 degrees of freedom
## Multiple R-squared:  0.04697,    Adjusted R-squared:  0.03409
## F-statistic: 3.647 on 1 and 74 DF,  p-value: 0.06004
```

Predict the following days

```
## # A tibble: 6 x 3
## # Groups:   date [1]
##   date       time                text
##   <date>     <chr>               <chr>
## 1 2021-04-10 2021-04-10 16:00:00 " Stay ahead with Nasdaq news views amp analys~
## 2 2021-04-10 2021-04-10 17:00:00 " The music is a commodity question is an inte~
## 3 2021-04-10 2021-04-10 18:00:00 " Stocks trending in conversation across FinTw~
## 4 2021-04-10 2021-04-10 19:00:00 " BTC Sentiment Price What s next for Bitcoin ~
## 5 2021-04-10 2021-04-10 20:00:00 " Apple chip partner TSMC to take part in Whit~
## 6 2021-04-10 2021-04-10 21:00:00 " Highest scoring stories for SP under one wat~
```

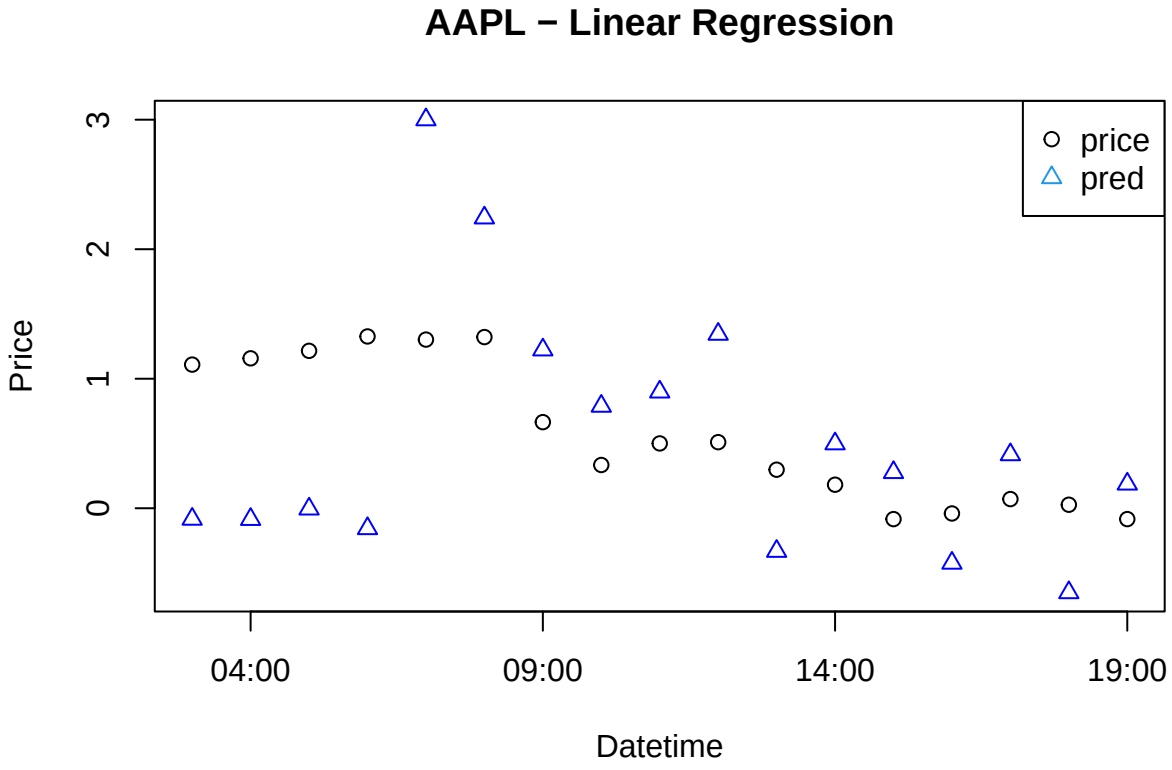
```
## [1] "there are total 145 observation"
```

```
## 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 03:00:00 0.196  2021-04-12 03:00         22           52       3
## 2 2021-04-12 04:00:00 0.221  2021-04-12 04:00         36           54       1
## 3 2021-04-12 05:00:00 0.0949 2021-04-12 05:00         29           48      22
## 4 2021-04-12 06:00:00 0.196  2021-04-12 06:00         39           37      29
## 5 2021-04-12 07:00:00 0.0853 2021-04-12 07:00         23           31      18
## 6 2021-04-12 08:00:00 0.182  2021-04-12 08:00         21           32      12
## # ... with 8 more variables: fear <dbl>, joy <dbl>, negative <dbl>,
## #   positive <dbl>, sadness <dbl>, surprise <dbl>, trust <dbl>, state <chr>
```

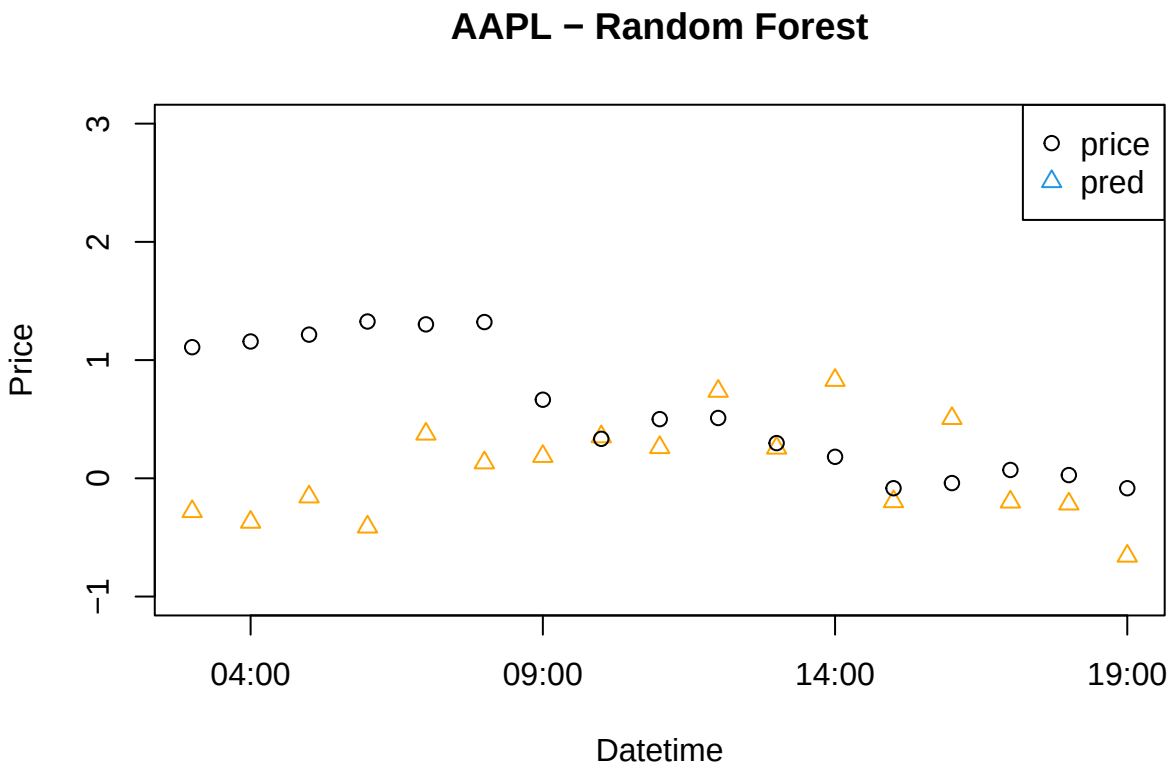
```
## # A tibble: 6 x 15
```

```
##   datetime             price date       time_stock anger anticipation disgust
##   <dttm>               <dbl> <date>     <chr>      <dbl>        <dbl>   <dbl>
## 1 2021-04-12 09:00:00 -0.137 2021-04-12 09:00         34           38      26
## 2 2021-04-12 10:00:00 -0.470 2021-04-12 10:00         42           71      27
## 3 2021-04-12 11:00:00 -0.446 2021-04-12 11:00         32           86      21
## 4 2021-04-12 12:00:00 -0.202 2021-04-12 12:00         42           86      23
## 5 2021-04-12 13:00:00 -0.325 2021-04-12 13:00         67          139      43
## 6 2021-04-12 14:00:00 -0.431 2021-04-12 14:00         56          117      27
## # ... 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")
```



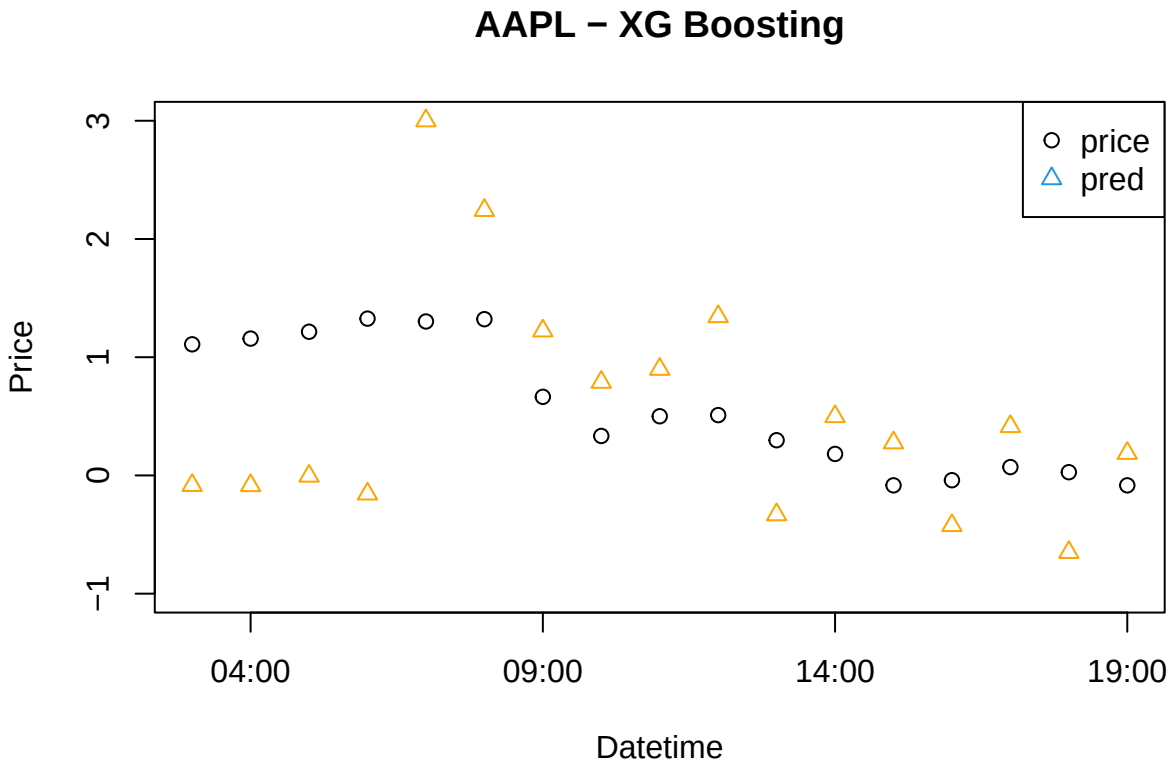

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 1 1
##          1 8 7
##
##                Accuracy : 0.4706
##                  95% CI : (0.2298, 0.7219)
##     No Information Rate : 0.5294
##     P-Value [Acc > NIR] : 0.7671
##
##                   Kappa : -0.0132
##
```

```
##  Mcnemar's Test P-Value : 0.0455
##
##             Sensitivity : 0.11111
##             Specificity : 0.87500
##          Pos Pred Value : 0.50000
##          Neg Pred Value : 0.46667
##              Prevalence : 0.52941
##          Detection Rate : 0.05882
##    Detection Prevalence : 0.11765
##       Balanced Accuracy : 0.49306
##
##        'Positive' Class : 0
##
```




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 6 7
##          1 3 1
##
##                Accuracy : 0.4118
##                  95% CI : (0.1844, 0.6708)
##     No Information Rate : 0.5294
##     P-Value [Acc > NIR] : 0.8878
##
##                   Kappa : -0.2143
##
##  Mcnemar's Test P-Value : 0.3428
```

```
##
##             Sensitivity : 0.6667
##             Specificity : 0.1250
##          Pos Pred Value : 0.4615
##          Neg Pred Value : 0.2500
##              Prevalence : 0.5294
##          Detection Rate : 0.3529
##    Detection Prevalence : 0.7647
##       Balanced Accuracy : 0.3958
##
##        'Positive' Class : 0
##
```




```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction 0 1
##          0 7 5
##          1 2 3
##
##                Accuracy : 0.5882
##                  95% CI : (0.3292, 0.8156)
##     No Information Rate : 0.5294
##     P-Value [Acc > NIR] : 0.4063
##
##                   Kappa : 0.156
##
##  Mcnemar's Test P-Value : 0.4497
##
```

```
##             Sensitivity : 0.7778
##             Specificity : 0.3750
##          Pos Pred Value : 0.5833
##          Neg Pred Value : 0.6000
##              Prevalence : 0.5294
##          Detection Rate : 0.4118
##    Detection Prevalence : 0.7059
##       Balanced Accuracy : 0.5764
##
##        'Positive' Class : 0
##
```