Tweets

2025-01-03

7 Case study: Comparing Twitter data from past presidents (Trump and Obama)

load libraries and dataset

```
library(lubridate)
```

```
## Warning: package 'lubridate' was built under R version 4.4.2
```

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.4.2
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(readr)
library(tidytext)
```

```
## Warning: package 'tidytext' was built under R version 4.4.2
```

```
library(stringr)
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 4.4.2
```

```
tweets_obama <- read_csv("C:\\Users\\knc5576\\Downloads\\obama.csv")
```

```
## Rows: 11539 Columns: 5
```

```
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr  (4): text, Comments, favorites, retweets
## dttm (1): timestamp
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
tweets_trump <- read_csv("C:\\Users\\knc5576\\Downloads\\trump.csv")
```

```
## Rows: 56571 Columns: 5
## ── Column specification ────────────────────────────────────────────────────────
## Delimiter: ","
## chr (3): text, device, timestamp
## dbl (2): favorites, retweets
##
## ℹ Use `spec()` to retrieve the full column specification for this data.
## ℹ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

7.1 Getting the data and distribution of tweets

```
#make favorites and retweets column in tweets_obama numeric
tweets_obama <- tweets_obama %>%
  mutate(favorites = as.numeric(favorites),
         retweets = as.numeric(retweets))
```

```
## Warning: There were 2 warnings in `mutate()`.
## The first warning was:
## ℹ In argument: `favorites = as.numeric(favorites)`.
## Caused by warning:
## ! NAs introduced by coercion
## ℹ Run `dplyr::last_dplyr_warnings()` to see the 1 remaining warning.
```

```
which(is.na(as.numeric(tweets_obama$favorites)))
```

```
##    [1]    32    68    77    78    85    91    95    96   100   128   130   133
##   [13]   136   137   142   143   144   146   148   150   151   152   154   159
##   [25]   165   166   168   173   176   177   179   181   182   185   186   188
##   [37]   189   190   191   192   193   194   195   196   198   199   200   202
##   [49]   203   205   207   208   212   218   219   220   223   224   226   227
##   [61]   231   238   240   242   247   251   252   254   258   262   263   265
##   [73]   267   268   270   272   273   274   275   277   280   281   284   285
##   [85]   454   469   534   555   556   567   568   587   588   591   592   621
##   [97]   624   626   637   653   654   672   733   734   739   745   746   752
##  [109]   764   773   784   786   799   816   843   844   845   871   877   881
##  [121]   888   891   893   902   919   920   921   922   927   928   935   937
##  [133]   943   945   946   966   967   968   970   971   972   973   975   976
##  [145]   977   979   980   981   984   985   987   988  1008  1017  1018  1026
##  [157]  1033  1036  1047  1053  1054  1055  1056  1058  1061  1062  1069  1070
##  [169]  1071  1074  1075  1078  1081  1083  1085  1087  1092  1095  1106  1108
##  [181]  1109  1110  1111  1113  1118  1119  1121  1122  1127  1130  1131  1133
##  [193]  1139  1147  1148  1183  1236  1237  1238  1246  1261  1263  1264  1331
##  [205]  1379  1380  1413  1443  1444  1445  1446  1447  1448  1449  1450  1451
##  [217]  1503  1504  1524  1525  1526  1595  1596  1630  1631  1675  1699  1702
##  [229]  1703  1704  1705  1735  1870  1881  1911  1913  1916  1918  1987  2032
##  [241]  2033  2047  2070  2074  2077  2082  2111  2130  2131  2132  2134  2150
##  [253]  2153  2154  2165  2179  2180  2181  2182  2183  2184  2185  2186  2188
##  [265]  2189  2190  2191  2193  2196  2219  2229  2253  2254  2259  2288  2335
##  [277]  2336  2338  2420  2421  2423  2456  2457  2458  2466  2498  2538  2540
##  [289]  2541  2579  2581  2625  2627  2630  2672  2719  2726  2730  2753  2754
##  [301]  2756  2759  2760  2767  2798  2800  2802  2808  2812  2835  2836  2837
##  [313]  2838  2839  2848  2871  2880  2918  2920  2921  2922  2923  2924  2928
##  [325]  2929  2933  2968  2969  2970  2972  3014  3018  3019  3020  3056  3057
##  [337]  3058  3059  3062  3063  3064  3065  3067  3069  3070  3073  3077  3102
##  [349]  3103  3107  3118  3147  3148  3149  3151  3152  3154  3155  3196  3197
##  [361]  3199  3200  3201  3216  3217  3244  3245  3247  3248  3254  3284  3285
##  [373]  3286  3287  3289  3290  3300  3306  3314  3328  3329  3331  3341  3345
##  [385]  3371  3372  3373  3374  3375  3376  3378  3421  3423  3460  3461  3463
##  [397]  3464  3466  3505  3507  3508  3510  3511  3512  3513  3515  3516  3518
##  [409]  3519  3521  3529  3543  3544  3545  3547  3550  3559  3585  3586  3589
##  [421]  3601  3629  3630  3631  3666  3667  3670  3671  3709  3710  3711  3721
##  [433]  3750  3751  3796  3798  3799  3800  3809  3832  3834  3862  3865  3866
##  [445]  3878  3907  3908  3916  3922  3954  3956  3957  3958  3959  3963  3964
##  [457]  3967  3969  3971  3976  3997  3998  3999  4006  4036  4037  4040  4041
##  [469]  4043  4044  4049  4055  4079  4080  4082  4083  4084  4085  4086  4087
##  [481]  4088  4090  4091  4092  4095  4096  4097  4098  4103  4105  4108  4126
##  [493]  4129  4130  4131  4140  4154  4168  4169  4170  4171  4172  4173  4174
##  [505]  4175  4176  4177  4178  4179  4181  4183  4184  4188  4194  4211  4212
##  [517]  4213  4215  4230  4231  4237  4251  4254  4274  4275  4277  4278  4279
##  [529]  4280  4285  4288  4290  4301  4302  4304  4305  4306  4308  4309  4313
##  [541]  4314  4315  4316  4317  4318  4320  4321  4323  4324  4325  4326  4327
##  [553]  4328  4329  4330  4331  4333  4334  4336  4337  4339  4355  4356  4357
##  [565]  4358  4359  4360  4361  4363  4364  4365  4366  4367  4369  4371  4375
##  [577]  4376  4377  4386  4388  4395  4402  4404  4405  4406  4407  4409  4410
##  [589]  4411  4412  4413  4415  4426  4437  4439  4440  4441  4442  4443  4444
```

```
##  [601]  4447  4453  4469  4471  4473  4474  4478  4480  4491  4505  4506  4507
##  [613]  4508  4509  4510  4511  4512  4513  4514  4515  4516  4517  4518  4519
##  [625]  4520  4521  4522  4523  4524  4526  4527  4528  4529  4530  4531  4532
##  [637]  4533  4534  4535  4537  4538  4541  4542  4546  4547  4552  4561  4562
##  [649]  4563  4564  4565  4570  4571  4578  4579  4581  4584  4596  4614  4615
##  [661]  4616  4639  4640  4641  4642  4666  4667  4669  4670  4671  4672  4674
##  [673]  4675  4676  4677  4679  4681  4682  4686  4687  4688  4689  4691  4692
##  [685]  4693  4694  4695  4697  4698  4699  4701  4707  4710  4717  4718  4719
##  [697]  4720  4721  4722  4723  4725  4728  4738  4741  4742  4743  4744  4745
##  [709]  4746  4747  4748  4749  4753  4754  4758  4764  4765  4766  4767  4768
##  [721]  4769  4770  4771  4772  4774  4775  4776  4777  4778  4779  4780  4781
##  [733]  4782  4783  4784  4785  4786  4787  4788  4789  4790  4791  4792  4793
##  [745]  4794  4795  4796  4797  4798  4799  4800  4801  4802  4804  4805  4806
##  [757]  4807  4808  4814  4815  4816  4817  4819  4825  4826  4827  4828  4847
##  [769]  4848  4849  4850  4851  4860  4861  4862  4865  4866  4867  4868  4869
##  [781]  4876  4879  4880  4881  4882  4884  4885  4886  4887  4890  4891  4892
##  [793]  4893  4894  4896  4897  4899  4903  4917  4918  4919  4920  4922  4930
##  [805]  4943  4944  4945  4946  4948  4949  4967  4968  4969  5004  5005  5006
##  [817]  5007  5008  5009  5010  5029  5030  5031  5032  5035  5036  5037  5041
##  [829]  5042  5043  5054  5055  5056  5057  5060  5077  5078  5079  5080  5082
##  [841]  5087  5088  5103  5104  5105  5106  5112  5120  5121  5122  5123  5125
##  [853]  5128  5130  5132  5133  5136  5144  5145  5146  5147  5148  5149  5150
##  [865]  5154  5172  5173  5174  5175  5195  5196  5197  5198  5199  5200  5201
##  [877]  5202  5205  5206  5207  5208  5209  5210  5211  5212  5213  5214  5219
##  [889]  5220  5221  5222  5224  5227  5228  5229  5233  5234  5237  5238  5239
##  [901]  5240  5242  5243  5249  5250  5251  5252  5253  5261  5264  5265  5266
##  [913]  5267  5268  5269  5275  5276  5277  5278  5287  5288  5289  5290  5292
##  [925]  5293  5296  5303  5304  5305  5306  5307  5308  5309  5310  5311  5312
##  [937]  5323  5324  5325  5326  5327  5328  5332  5341  5342  5343  5344  5346
##  [949]  5347  5348  5349  5351  5353  5360  5377  5378  5379  5380  5381  5382
##  [961]  5383  5384  5385  5386  5388  5389  5390  5391  5392  5396  5399  5414
##  [973]  5415  5416  5417  5419  5420  5421  5424  5426  5429  5448  5449  5450
##  [985]  5451  5453  5454  5455  5459  5462  5467  5493  5494  5495  5496  5497
##  [997]  5498  5499  5500  5501  5504  5505  5525  5528  5529  5530  5531  5532
## [1009]  5533  5534  5539  5540  5542  5545  5562  5563  5564  5566  5567  5568
## [1021]  5570  5577  5578  5579  5580  5581  5582  5583  5584  5585  5586  5588
## [1033]  5617  5618  5619  5620  5621  5659  5660  5693  5694  5695  5718  5719
## [1045]  5720  5721  5722  5723  5724  5727  5731  5732  5757  5758  5759  5760
## [1057]  5788  5790  5791  5795  5811  5812  5814  5815  5816  5818  5819  5820
## [1069]  5851  5852  5853  5855  5856  5858  5860  5861  5862  5863  5865  5892
## [1081]  5893  5895  5904  5909  5922  5924  5925  5927  5930  5931  5932  5933
## [1093]  5950  5951  5952  5953  5954  5955  5968  5969  5972  5973  5976  5987
## [1105]  6008  6009  6010  6011  6012  6013  6014  6015  6016  6017  6018  6019
## [1117]  6020  6021  6022  6024  6025  6026  6027  6028  6030  6031  6032  6034
## [1129]  6035  6037  6039  6049  6050  6051  6052  6053  6054  6055  6056  6057
## [1141]  6059  6060  6066  6067  6069  6070  6083  6084  6085  6086  6087  6088
## [1153]  6090  6091  6092  6097  6101  6102  6103  6104  6105  6106  6107  6108
## [1165]  6109  6112  6115  6116  6117  6119  6120  6123  6124  6136  6137  6139
## [1177]  6141  6166  6167  6169  6170  6192  6193  6194  6195  6196  6197  6207
## [1189]  6224  6225  6226  6227  6228  6229  6231  6233  6258  6259  6260  6261
## [1201]  6270  6279  6281  6282  6286  6304  6305  6306  6307  6309  6332  6333
## [1213]  6337  6344  6358  6359  6360  6362  6375  6376  6377  6378  6380  6382
## [1225]  6383  6386  6389  6395  6398  6425  6426  6427  6447  6452  6497  6498
## [1237]  6512  6517  6519  6540  6554  6558  6559  6562  6594  6595  6627  6628
## [1249]  6629  6630  6631  6632  6634  6647  6648  6650  6670  6671  6672  6673
## [1261]  6674  6675  6676  6677  6678  6679  6682  6686  6697  6718  6719  6722
## [1273]  6751  6755  6766  6768  6769  6770  6771  6783  6813  6814  6815  6816
## [1285]  6817  6822  6824  6834  6878  6879  6880  6882  6883  6885  6887  6917
## [1297]  6919  6920  6924  6925  6930  6971  6988  6989  6990  7016  7018  7027
## [1309]  7049  7050  7077  7119  7120  7123  7124  7126  7152  7153  7154  7155
## [1321]  7159  7160  7161  7174  7177  7178  7179  7202  7203  7204  7205  7227
## [1333]  7228  7229  7244  7245  7246  7248  7256  7264  7265  7266  7267  7268
## [1345]  7270  7292  7293  7295  7310  7311  7313  7314  7315  7316  7321  7328
## [1357]  7329  7337  7340  7344  7361  7362  7367  7368  7369  7370  7384  7385
## [1369]  7386  7396  7397  7398  7401  7418  7419  7420  7421  7422  7439  7440
## [1381]  7441  7442  7443  7444  7445  7447  7448  7465  7466  7467  7468  7469
## [1393]  7471  7483  7484  7494  7510  7511  7514  7515  7516  7517  7518  7543
## [1405]  7544  7545  7546  7554  7564  7587  7588  7589  7590  7605  7606  7636
## [1417]  7645  7646  7659  7683  7692  7693  7695  7697  7721  7729  7732  7756
## [1429]  7757  7758  7759  7763  7776  7788  7789  7790  7791  7800  7801  7802
## [1441]  7803  7810  7826  7827  7828  7829  7830  7849  7850  7851  7856  7858
## [1453]  7859  7873  7874  7882  7896  7898  7899  7900  7903  7916  7917  7918
## [1465]  7920  7922  7923  7925  7927  7928  7930  7940  7947  7952  7960  7962
## [1477]  7964  7965  7966  7968  8004  8005  8006  8007  8017  8025  8026  8043
## [1489]  8048  8064  8065  8089  8091  8092  8093  8096  8097  8099  8111  8112
## [1501]  8113  8114  8119  8120  8123  8126  8127  8128  8132  8138  8143  8155
## [1513]  8174  8175  8176  8177  8178  8179  8180  8181  8182  8183  8184  8185
## [1525]  8186  8187  8188  8190  8191  8192  8193  8196  8197  8205  8206  8207
## [1537]  8209  8211  8213  8219  8223  8225  8228  8235  8243  8246  8257  8258
```

```
## [1549]  8260  8264  8280  8282  8303  8308  8325  8340  8341  8343  8348  8351
## [1561]  8352  8355  8361  8377  8385  8386  8387  8392  8393  8399  8406  8409
## [1573]  8411  8414  8433  8434  8435  8436  8437  8438  8439  8440  8441  8442
## [1585]  8443  8444  8446  8447  8448  8449  8450  8451  8452  8453  8454  8455
## [1597]  8456  8457  8458  8460  8462  8463  8464  8465  8466  8467  8468  8469
## [1609]  8470  8471  8474  8475  8483  8484  8485  8486  8497  8499  8503  8507
## [1621]  8509  8516  8521  8525  8536  8538  8539  8544  8550  8553  8554  8555
## [1633]  8558  8559  8560  8561  8562  8563  8564  8565  8567  8569  8570  8571
## [1645]  8572  8573  8574  8579  8580  8581  8582  8583  8585  8586  8587  8588
## [1657]  8589  8591  8594  8595  8597  8601  8603  8607  8608  8609  8610  8612
## [1669]  8613  8617  8620  8625  8627  8630  8631  8632  8638  8640  8641  8642
## [1681]  8668  8693  8707  8708  8709  8710  8711  8713  8724  8725  8727  8736
## [1693]  8738  8739  8740  8741  8742  8743  8744  8745  8756  8759  8766  8767
## [1705]  8769  8800  8801  8804  8814  8815  8818  8824  8825  8829  8833  8843
## [1717]  8844  8845  8855  8859  8860  8861  8862  8863  8864  8886  8889  8890
## [1729]  8894  8895  8902  8903  8904  8905  8906  8907  8909  8910  8914  8916
## [1741]  8917  8918  8920  8924  8927  8931  8932  8933  8945  8946  8947  8960
## [1753]  8962  8963  8964  8965  8980  8990  8991  8992  8993  8994  8995  8996
## [1765]  8997  8998  8999  9000  9006  9007  9008  9012  9015  9016  9017  9018
## [1777]  9020  9021  9024  9025  9026  9027  9030  9033  9041  9050  9052  9073
## [1789]  9075  9083  9088  9090  9091  9092  9093  9095  9097  9098  9099  9100
## [1801]  9101  9102  9103  9107  9108  9109  9110  9113  9114  9117  9118  9119
## [1813]  9121  9122  9123  9135  9136  9140  9141  9150  9163  9164  9165  9172
## [1825]  9173  9174  9176  9178  9186  9188  9189  9190  9192  9217  9232  9234
## [1837]  9235  9237  9240  9247  9252  9267  9278  9279  9280  9284  9295  9297
## [1849]  9298  9299  9300  9307  9308  9310  9320  9321  9323  9326  9328  9332
## [1861]  9333  9334  9335  9340  9349  9350  9356  9373  9374  9375  9376  9388
## [1873]  9390  9391  9392  9398  9402  9404  9410  9411  9418  9424  9425  9434
## [1885]  9444  9446  9461  9462  9467  9484  9508  9509  9514  9515  9520  9527
## [1897]  9528  9529  9533  9539  9540  9549  9550  9551  9555  9557  9559  9560
## [1909]  9588  9608  9611  9614  9615  9616  9617  9622  9624  9625  9626  9627
## [1921]  9629  9637  9641  9644  9649  9650  9651  9652  9653  9654  9655  9656
## [1933]  9657  9658  9659  9660  9661  9662  9663  9664  9665  9666  9667  9668
## [1945]  9669  9671  9675  9678  9684  9688  9700  9701  9704  9714  9715  9717
## [1957]  9719  9721  9722  9728  9730  9731  9734  9735  9745  9746  9747  9749
## [1969]  9758  9759  9769  9770  9771  9773  9775  9776  9777  9779  9781  9782
## [1981]  9783  9784  9787  9788  9789  9790  9791  9794  9796  9802  9803  9804
## [1993]  9805  9806  9808  9810  9811  9818  9819  9820  9822  9823  9824  9826
## [2005]  9832  9833  9834  9835  9836  9839  9840  9841  9849  9850  9851  9855
## [2017]  9860  9861  9862  9863  9864  9865  9867  9868  9873  9874  9875  9876
## [2029]  9877  9878  9879  9880  9887  9888  9889  9899  9900  9901  9902  9904
## [2041]  9911  9912  9913  9914  9916  9918  9924  9925  9926  9927  9928  9929
## [2053]  9930  9931  9932  9935  9944  9945  9946  9947  9949  9956  9957  9958
## [2065]  9959  9960  9961  9968  9970  9971  9972  9974  9979  9982  9988  9989
## [2077]  9990  9991  9992  9993  9995  9997 10007 10008 10009 10010 10011 10017
## [2089] 10018 10019 10020 10021 10030 10031 10034 10035 10036 10041 10053 10054
## [2101] 10055 10056 10063 10064 10066 10067 10068 10072 10074 10075 10076 10077
## [2113] 10078 10079 10080 10081 10082 10084 10090 10091 10092 10093 10094 10095
## [2125] 10097 10101 10102 10103 10104 10105 10106 10107 10108 10109 10111 10112
## [2137] 10113 10114 10115 10116 10117 10118 10120 10125 10129 10130 10131 10132
## [2149] 10133 10134 10135 10136 10137 10138 10145 10147 10148 10149 10150 10151
## [2161] 10153 10155 10156 10157 10158 10159 10162 10168 10170 10171 10174 10175
## [2173] 10176 10177 10178 10180 10184 10185 10186 10187 10188 10189 10191 10192
## [2185] 10193 10195 10198 10199 10200 10201 10202 10203 10206 10210 10212 10213
## [2197] 10214 10215 10216 10218 10219 10220 10221 10222 10224 10225 10226 10227
## [2209] 10228 10231 10235 10236 10240 10241 10242 10243 10244 10245 10246 10249
## [2221] 10252 10255 10256 10257 10259 10262 10264 10265 10266 10267 10268 10269
## [2233] 10271 10272 10276 10277 10278 10279 10280 10283 10285 10291 10294 10295
## [2245] 10297 10298 10301 10302 10303 10304 10305 10306 10308 10309 10312 10313
## [2257] 10314 10315 10316 10317 10319 10320 10321 10322 10323 10327 10329 10330
## [2269] 10331 10332 10333 10334 10336 10341 10342 10343 10344 10345 10346 10348
## [2281] 10353 10354 10355 10356 10357 10360 10363 10365 10366 10367 10368 10369
## [2293] 10372 10374 10377 10382 10383 10384 10385 10386 10387 10393 10394 10395
## [2305] 10396 10397 10398 10399 10401 10402 10405 10406 10407 10408 10409 10413
## [2317] 10414 10420 10421 10422 10423 10424 10425 10426 10427 10429 10431 10432
## [2329] 10434 10435 10437 10438 10440 10442 10444 10445 10446 10447 10448 10449
## [2341] 10450 10451 10452 10453 10454 10455 10456 10457 10458 10459 10460 10461
## [2353] 10462 10463 10464 10465 10466 10467 10468 10469 10470 10471 10472 10473
## [2365] 10474 10475 10476 10477 10478 10479 10480 10481 10482 10483 10484 10485
## [2377] 10486 10487 10488 10489 10490 10491 10492 10493 10494 10495 10496 10497
## [2389] 10498 10499 10500 10501 10502 10503 10504 10505 10506 10507 10508 10509
## [2401] 10510 10511 10512 10513 10514 10515 10516 10517 10518 10519 10520 10521
## [2413] 10522 10523 10524 10525 10526 10527 10528 10529 10530 10531 10532 10533
## [2425] 10534 10535 10536 10537 10538 10539 10540 10541 10542 10543 10544 10545
## [2437] 10546 10547 10548 10549 10550 10551 10552 10553 10554 10555 10556 10557
## [2449] 10558 10559 10560 10561 10562 10563 10564 10565 10566 10567 10568 10569
## [2461] 10570 10571 10572 10573 10574 10575 10576 10577 10578 10579 10580 10581
## [2473] 10582 10583 10584 10585 10586 10587 10588 10589 10590 10591 10592 10593
## [2485] 10594 10595 10596 10597 10598 10599 10600 10601 10602 10603 10604 10605
```

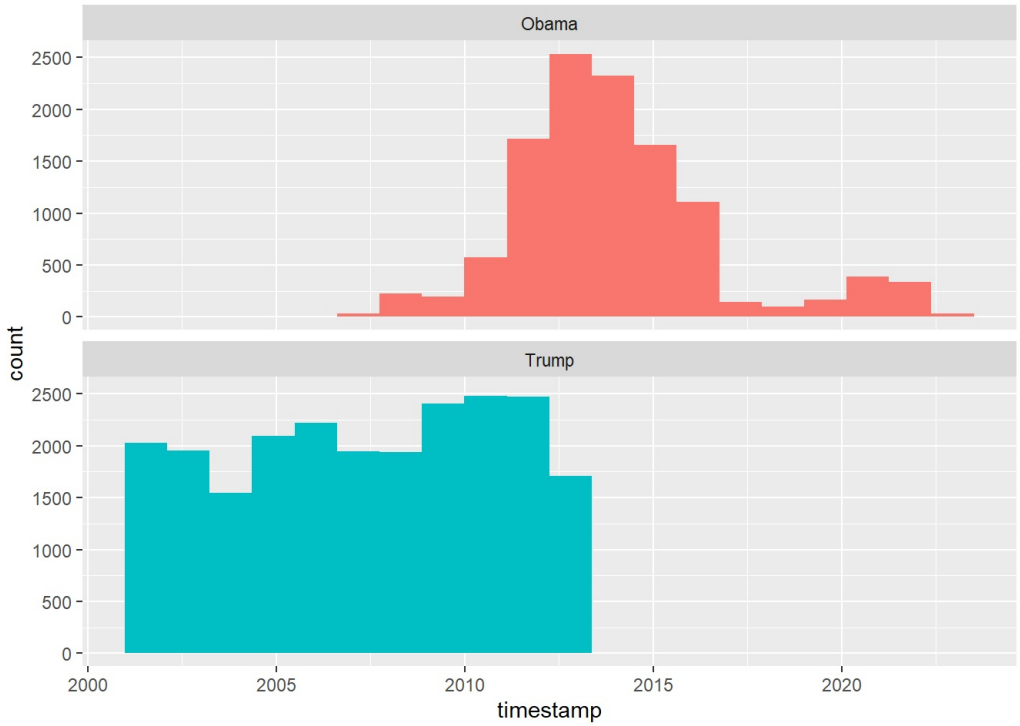
```
## [2497] 10606 10607 10608 10609 10610 10611 10612 10613 10614 10615 10616 10617
## [2509] 10618 10619 10620 10621 10622 10623 10624 10625 10626 10627 10628 10629
## [2521] 10630 10631 10632 10633 10634 10635 10636 10637 10638 10639 10640 10641
## [2533] 10642 10643 10644 10645 10646 10647 10648 10649 10650 10651 10652 10653
## [2545] 10654 10655 10656 10657 10658 10659 10660 10661 10662 10663 10664 10665
## [2557] 10666 10667 10668 10669 10670 10671 10672 10673 10674 10675 10676 10677
## [2569] 10678 10679 10680 10681 10682 10683 10684 10685 10686 10687 10688 10689
## [2581] 10690 10691 10692 10693 10694 10695 10696 10697 10698 10699 10700 10701
## [2593] 10702 10703 10704 10705 10706 10707 10708 10709 10710 10711 10712 10713
## [2605] 10714 10715 10716 10717 10718 10719 10720 10721 10722 10723 10724 10725
## [2617] 10726 10727 10728 10729 10730 10731 10732 10733 10734 10735 10736 10737
## [2629] 10738 10739 10740 10741 10742 10743 10744 10745 10746 10747 10748 10749
## [2641] 10750 10751 10752 10753 10754 10755 10756 10757 10758 10759 10760 10761
## [2653] 10762 10763 10764 10765 10766 10767 10768 10769 10770 10771 10772 10773
## [2665] 10774 10775 10776 10777 10778 10779 10780 10781 10782 10783 10784 10785
## [2677] 10786 10787 10788 10789 10790 10791 10792 10793 10794 10795 10796 10797
## [2689] 10798 10799 10800 10801 10802 10803 10804 10805 10806 10807 10808 10809
## [2701] 10810 10811 10812 10813 10814 10815 10816 10817 10818 10819 10820 10821
## [2713] 10822 10823 10824 10825 10826 10827 10828 10829 10830 10831 10832 10833
## [2725] 10834 10835 10836 10837 10838 10839 10840 10841 10842 10843 10844 10845
## [2737] 10846 10847 10848 10849 10850 10851 10852 10853 10854 10855 10856 10857
## [2749] 10858 10859 10860 10861 10862 10863 10864 10865 10866 10867 10868 10869
## [2761] 10870 10871 10872 10873 10874 10875 10876 10877 10878 10879 10880 10881
## [2773] 10882 10883 10884 10885 10886 10887 10888 10889 10890 10891 10892 10893
## [2785] 10894 10895 10896 10897 10898 10899 10900 10901 10902 10903 10904 10905
## [2797] 10906 10907 10908 10909 10910 10911 10912 10913 10914 10915 10916 10917
## [2809] 10918 10919 10920 10921 10922 10923 10924 10925 10926 10927 10928 10929
## [2821] 10930 10931 10932 10933 10934 10935 10936 10937 10938 10939 10940 10941
## [2833] 10942 10943 10944 10945 10946 10947 10948 10949 10950 10951 10952 10953
## [2845] 10954 10955 10956 10957 10958 10959 10960 10961 10962 10963 10964 10965
## [2857] 10966 10967 10968 10969 10970 10971 10972 10973 10974 10975 10976 10977
## [2869] 10978 10979 10980 10981 10982 10983 10984 10985 10986 10987 10988 10989
## [2881] 10990 10991 10992 10993 10994 10995 10996 10997 10998 10999 11000 11001
## [2893] 11002 11003 11004 11005 11006 11007 11008 11009 11010 11011 11012 11013
## [2905] 11014 11015 11016 11017 11018 11019 11020 11021 11022 11023 11024 11025
## [2917] 11026 11027 11028 11029 11030 11031 11032 11033 11034 11035 11036 11037
## [2929] 11038 11039 11040 11041 11042 11043 11044 11045 11046 11047 11048 11049
## [2941] 11050 11051 11052 11053 11054 11055 11056 11057 11058 11059 11060 11061
## [2953] 11062 11063 11064 11065 11066 11067 11068 11069 11070 11071 11072 11073
## [2965] 11074 11075 11076 11077 11078 11079 11080 11081 11082 11083 11084 11085
## [2977] 11086 11087 11088 11089 11090 11091 11092 11093 11094 11095 11096 11097
## [2989] 11098 11099 11100 11101 11102 11103 11104 11105 11106 11107 11108 11109
## [3001] 11110 11111 11112 11113 11114 11115 11116 11117 11118 11119 11120 11121
## [3013] 11122 11123 11124 11125 11126 11127 11129 11130 11131 11132 11133 11134
## [3025] 11135 11136 11137 11138 11139 11140 11141 11142 11143 11144 11145 11146
## [3037] 11147 11148 11149 11150 11151 11152 11153 11154 11155 11156 11157 11163
## [3049] 11164 11165 11166 11167 11168 11169 11170 11171 11172 11173 11175 11176
## [3061] 11177 11178 11179 11180 11181 11182 11183 11184 11185 11186 11187 11188
## [3073] 11189 11190 11191 11193 11194 11195 11196 11197 11198 11199 11200 11201
## [3085] 11202 11203 11204 11205 11206 11207 11208 11209 11210 11211 11212 11214
## [3097] 11215 11216 11217 11218 11219 11220 11221 11222 11223 11224 11225 11226
## [3109] 11227 11228 11229 11230 11231 11232 11233 11234 11235 11236 11237 11238
## [3121] 11239 11240 11241 11242 11243 11244 11245 11246 11247 11248 11249 11250
## [3133] 11251 11252 11253 11254 11255 11256 11257 11258 11259 11260 11261 11262
## [3145] 11263 11264 11265 11266 11267 11268 11269 11270 11271 11272 11273 11274
## [3157] 11275 11276 11277 11278 11279 11280 11281 11282 11283 11284 11285 11286
## [3169] 11287 11289 11290 11291 11292 11293 11294 11295 11296 11297 11298 11299
## [3181] 11300 11301 11302 11303 11304 11305 11307 11308 11309 11310 11311 11313
## [3193] 11314 11315 11316 11317 11318 11319 11320 11321 11322 11323 11324 11325
## [3205] 11326 11330 11331 11332 11334 11335 11336 11337 11339 11340 11341 11342
## [3217] 11343 11344 11345 11346 11347 11348 11349 11350 11351 11352 11353 11354
## [3229] 11355 11356 11357 11358 11360 11361 11362 11365 11366 11367 11368 11369
## [3241] 11370 11371 11372 11373 11374 11375 11376 11377 11378 11379 11380 11381
## [3253] 11382 11385 11386 11387 11389 11390 11393 11395 11396 11397 11398 11399
## [3265] 11401 11402 11403 11405 11406 11408 11409 11410 11415 11416 11417 11422
## [3277] 11423 11424 11425 11426 11427 11428 11429 11430 11431 11432 11433 11434
## [3289] 11435 11436 11437 11438 11439 11440 11441 11442 11443 11445 11446 11448
## [3301] 11449 11450 11451 11452 11453 11454 11460 11461 11462 11463 11464 11465
## [3313] 11467 11468 11469 11470 11471 11473 11474 11475 11477 11478 11479 11480
## [3325] 11481 11482 11483 11484 11485 11488 11489 11490 11492 11493 11494 11495
## [3337] 11496 11497 11498 11499 11500 11501 11502 11503 11504 11505 11506 11507
## [3349] 11508 11509 11510 11511 11512 11513 11514 11515 11516 11517 11518 11519
## [3361] 11520 11521 11524 11525 11526 11527 11528 11530 11532 11534 11535 11536
## [3373] 11537 11538 11539
```

```
#make timestamp format the same for each dataset
tweets_obama <- tweets_obama %>%
  mutate(timestamp = ymd_hms(timestamp))
#now the same for the other
tweets_trump <- tweets_trump %>%
  mutate(timestamp = ymd_hms(timestamp))
```

```
## Warning: There was 1 warning in `mutate()`.
## ℹ In argument: `timestamp = ymd_hms(timestamp)`.
## Caused by warning:
## !  33770 failed to parse.
```

```
#bind the datasets
tweets <- bind_rows(
  tweets_obama %>% mutate(person = "Obama"),
  tweets_trump %>% mutate(person = "Trump")
)


ggplot(tweets, aes(x = timestamp, fill = person)) +
  geom_histogram(position = "identity", bins = 20, show.legend = FALSE) +
  facet_wrap(~person, ncol = 1)
```

```
## Warning: Removed 33770 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



7.2 Word Frequencies

```
# Removing stop words
numbers_as_words <- as.character(1:500)
custom_stop_words <- tibble(word = numbers_as_words)
tidy_tweets <- tweets %>%
  unnest_tokens(word, text) %>%
  filter(!str_detect(word, "^[0-9]+$")) %>%
  anti_join(stop_words) %>%
  anti_join(custom_stop_words, by = "word")
```

```
## Joining with `by = join_by(word)`
```

```
# join the data with the stop words
tidy_tweets <- tweets %>%
  unnest_tokens(word, text) %>%
  anti_join(stop_words) %>%
  anti_join(custom_stop_words)
```

```
## Joining with `by = join_by(word)`
## Joining with `by = join_by(word)`
```

```
#View the frequency of words from each past president

frequency <- tidy_tweets %>%
  count(person, word, sort = TRUE) %>%
  left_join(tidy_tweets %>%
              count(person, name = "total")) %>%
  mutate(freq = n/total)
```

```
## Joining with `by = join_by(person)`
```

```
frequency <- frequency %>%
  select(person, word, freq) %>%
  pivot_wider(names_from = person, values_from = freq) %>%
arrange(Obama, Trump)


library(scales)
```

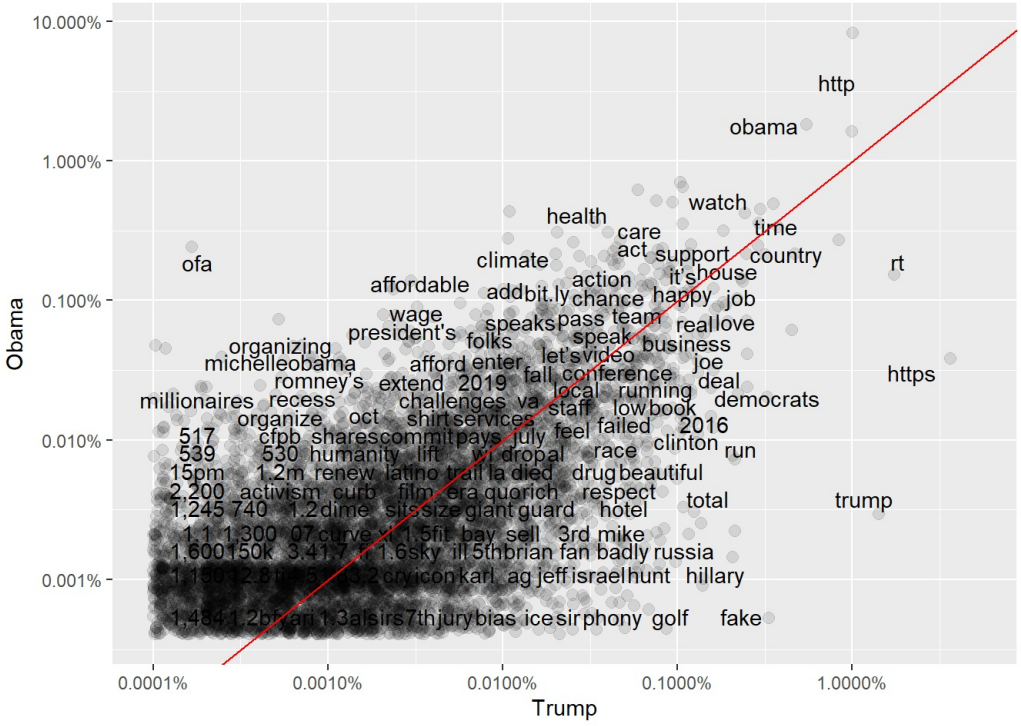
```
## Warning: package 'scales' was built under R version 4.4.2
```

```
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:readr':
##
##     col_factor
```

```
ggplot(frequency, aes(Trump, Obama)) +
  geom_jitter(alpha = 0.1, size = 2.5, width = 0.25, height = 0.25) +
  geom_text(aes(label = word), check_overlap = TRUE, vjust = 1.5) +
  scale_x_log10(labels = percent_format()) +
  scale_y_log10(labels = percent_format()) +
  geom_abline(color = "red")
```

```
## Warning: Removed 66979 rows containing missing values or values outside the scale range
## (`geom_point()`).
```

```
## Warning: Removed 66979 rows containing missing values or values outside the scale range
## (`geom_text()`).
```

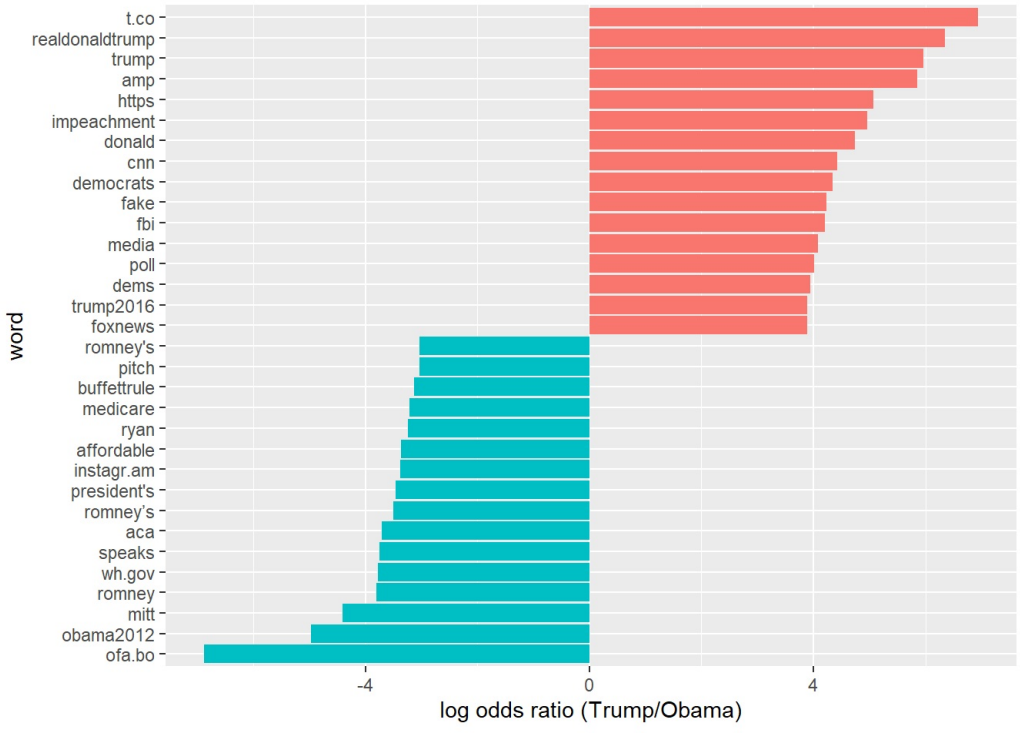


7.3 Comparing word usage

```
tidy_tweets <- tidy_tweets %>%
  filter(timestamp >= as.Date("2012-01-01"),
         timestamp < as.Date("2013-01-01"))

word_ratios <- tidy_tweets %>%
  filter(!str_detect(word, "^@")) %>%
  count(word, person) %>%
  group_by(word) %>%
  filter(sum(n) >= 10) %>%
  ungroup() %>%
  pivot_wider(names_from = person, values_from = n, values_fill = 0) %>%
  mutate_if(is.numeric, list(~(. + 1) / (sum(.) + 1))) %>%
  mutate(logratio = log(Trump / Obama)) %>%
  arrange(desc(logratio))

word_ratios %>%
  arrange(abs(logratio))
```

```
## # A tibble: 819 × 4
##    word         Trump    Obama logratio
##    <chr>        <dbl>    <dbl>    <dbl>
##  1 bill      0.00177  0.00177   0.00309
##  2 can't     0.00202  0.00204  -0.00857
##  3 doesn't   0.000759 0.000773 -0.0175
##  4 senator   0.000759 0.000773 -0.0175
##  5 hampshire 0.000506 0.000497  0.0188
##  6 months    0.000844 0.000828  0.0188
##  7 politics  0.000675 0.000662  0.0188
##  8 save      0.000844 0.000828  0.0188
##  9 saving    0.000506 0.000497  0.0188
## 10 failed    0.000591 0.000607 -0.0277
## # ℹ 809 more rows
```

```
word_ratios %>%
  group_by(logratio < 0) %>%
  slice_max(abs(logratio), n = 15) %>%
  ungroup() %>%
  mutate(word = reorder(word, logratio)) %>%
  ggplot(aes(word, logratio, fill = logratio < 0)) +
  geom_col(show.legend = FALSE) +
  coord_flip() +
  ylab("log odds ratio (Trump/Obama)") +
  scale_fill_discrete(name = "", labels = c("Trump", "Obama"))
```



7.4 Changes in word use

```
words_by_time <- tidy_tweets %>%
  filter(!str_detect(word, "^@")) %>%
  mutate(time_floor = floor_date(timestamp, unit = "1 month")) %>%
  count(time_floor, person, word) %>%
  group_by(person, time_floor) %>%
  mutate(time_total = sum(n)) %>%
  group_by(person, word) %>%
  mutate(word_total = sum(n)) %>%
  ungroup() %>%
  rename(count = n) %>%
  filter(word_total > 30)
# nest the data
nested_data <- words_by_time %>%
  nest(data = c(-word, -person))

library(purrr)
```

```
##
## Attaching package: 'purrr'
```

```
## The following object is masked from 'package:scales':
##
##     discard
```

```
nested_models <- nested_data %>%
  mutate(models = map(data, ~ glm(cbind(count, time_total) ~ time_floor, .,
                                  family = "binomial")))

library(broom)
```

```
## Warning: package 'broom' was built under R version 4.4.2
```

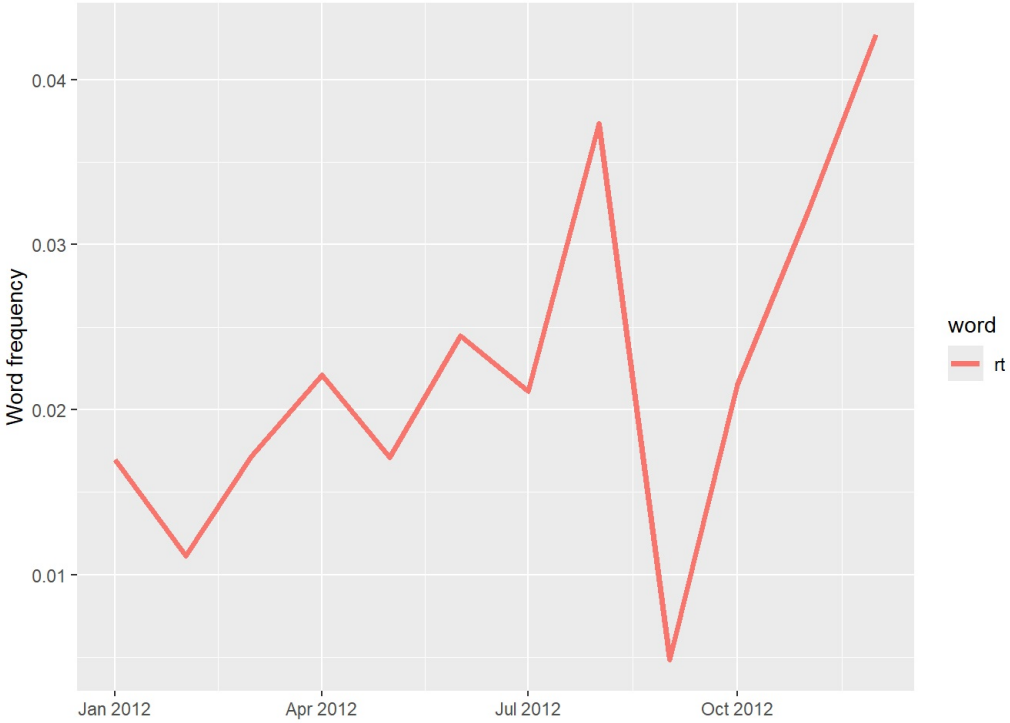
```
slopes <- nested_models %>%
  mutate(models = map(models, tidy)) %>%
  unnest(cols = c(models)) %>%
  filter(term == "time_floor") %>%
  mutate(adjusted.p.value = p.adjust(p.value))

top_slopes <- slopes %>%
  filter(adjusted.p.value < 0.01)

top_slopes
```

```
## # A tibble: 15 × 9
##    person word      data     term          estimate std.error statistic  p.value
##    <chr>  <chr>     <list>   <chr>            <dbl>     <dbl>     <dbl>    <dbl>
##  1 Obama  class     <tibble> time_floor     9.36e-8   1.19e-8      7.86 3.91e-15
##  2 Obama  http      <tibble> time_floor    -1.41e-8   3.13e-9     -4.51 6.53e- 6
##  3 Obama  middle    <tibble> time_floor     1.06e-7   1.24e-8      8.50 1.92e-17
##  4 Obama  obama2012 <tibble> time_floor    -5.43e-8   9.88e-9     -5.49 3.92e- 8
##  5 Obama  plan      <tibble> time_floor     6.31e-8   1.13e-8      5.59 2.23e- 8
##  6 Obama  potus     <tibble> time_floor     7.06e-8   1.43e-8      4.94 7.97e- 7
##  7 Obama  president <tibble> time_floor     1.87e-8   4.06e-9      4.60 4.18e- 6
##  8 Obama  romney    <tibble> time_floor     5.31e-8   9.22e-9      5.76 8.40e- 9
##  9 Obama  rt        <tibble> time_floor    -7.38e-8   9.77e-9     -7.56 4.13e-14
## 10 Obama  tax       <tibble> time_floor     3.72e-8   9.04e-9      4.11 3.92e- 5
## 11 Obama  taxes     <tibble> time_floor     1.17e-7   1.72e-8      6.81 9.86e-12
## 12 Obama  vote      <tibble> time_floor     1.04e-7   1.59e-8      6.53 6.79e-11
## 13 Trump  rt        <tibble> time_floor     3.00e-8   5.07e-9      5.92 3.24e- 9
## 14 Obama  election  <tibble> time_floor     7.43e-8   1.37e-8      5.43 5.52e- 8
## 15 Obama  family    <tibble> time_floor     7.99e-8   1.92e-8      4.17 3.09e- 5
## # i 1 more variable: adjusted.p.value <dbl>
```

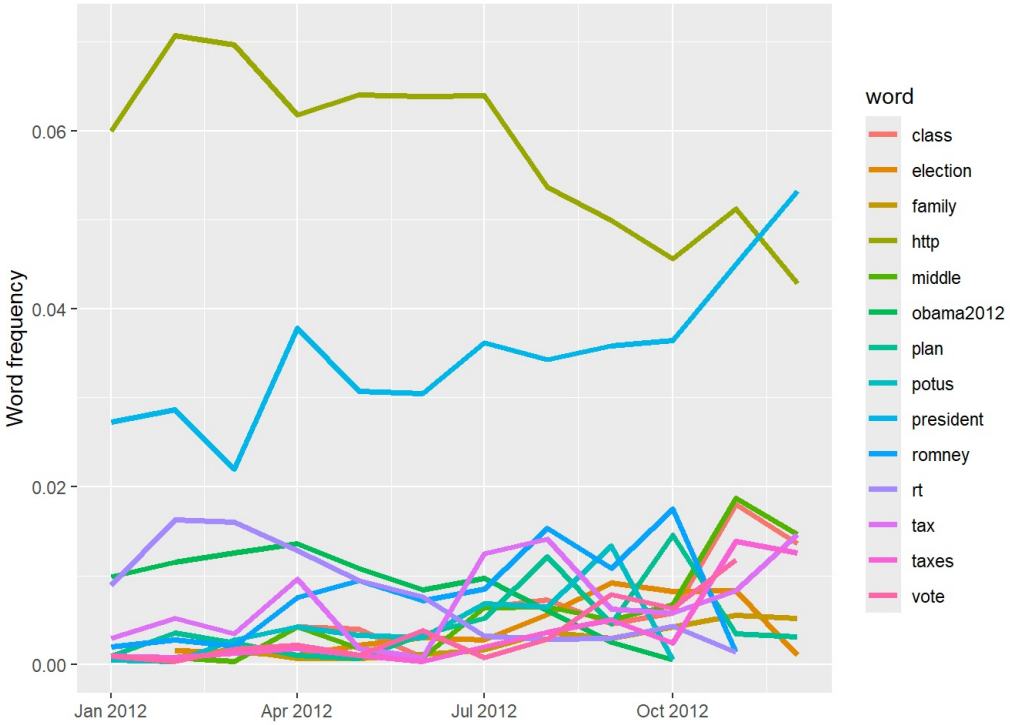
Taking a look at words most tweeted over 2012-2013 for each past president

```
words_by_time %>%
  inner_join(top_slopes, by = c("word", "person")) %>%
  filter(person == "Trump") %>%
  ggplot(aes(time_floor, count/time_total, color = word)) +
  geom_line(size = 1.3) +
  labs(x = NULL, y = "Word frequency")
```

```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## ℹ Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
words_by_time %>%
  inner_join(top_slopes, by = c("word", "person")) %>%
  filter(person == "Obama") %>%
  ggplot(aes(time_floor, count/time_total, color = word)) +
  geom_line(size = 1.3) +
  labs(x = NULL, y = "Word frequency")
```



7.5 Favorites and retweets

preparing the data for analysis of favorites and retweets

```
tidy_tweets <- tidy_tweets %>%
  mutate(id = row_number())

tweets_obama <- tweets_obama %>%
  mutate(retweets = as.numeric(retweets))
tweets_obama <- tweets_obama %>%
  mutate(retweets = ifelse(is.na(retweets), 0, retweets))
which(is.na(as.numeric(tweets_obama$retweets)))
```

```
## integer(0)
```

```
tidy_tweets %>%
  filter(person == "Obama") %>%
  summarise(non_na_retweets = sum(!is.na(retweets)))
```

```
## # A tibble: 1 × 1
##   non_na_retweets
##             <int>
## 1           26836
```

```
totals <- tidy_tweets %>%
  group_by(person, id) %>%
  summarise(rts = first(na.omit(retweets))) %>%
  group_by(person) %>%
  summarise(total_rts = sum(rts, na.rm = TRUE))
```

```
## `summarise()` has grouped output by 'person'. You can override using the
## `.groups` argument.
```

```
totals
```

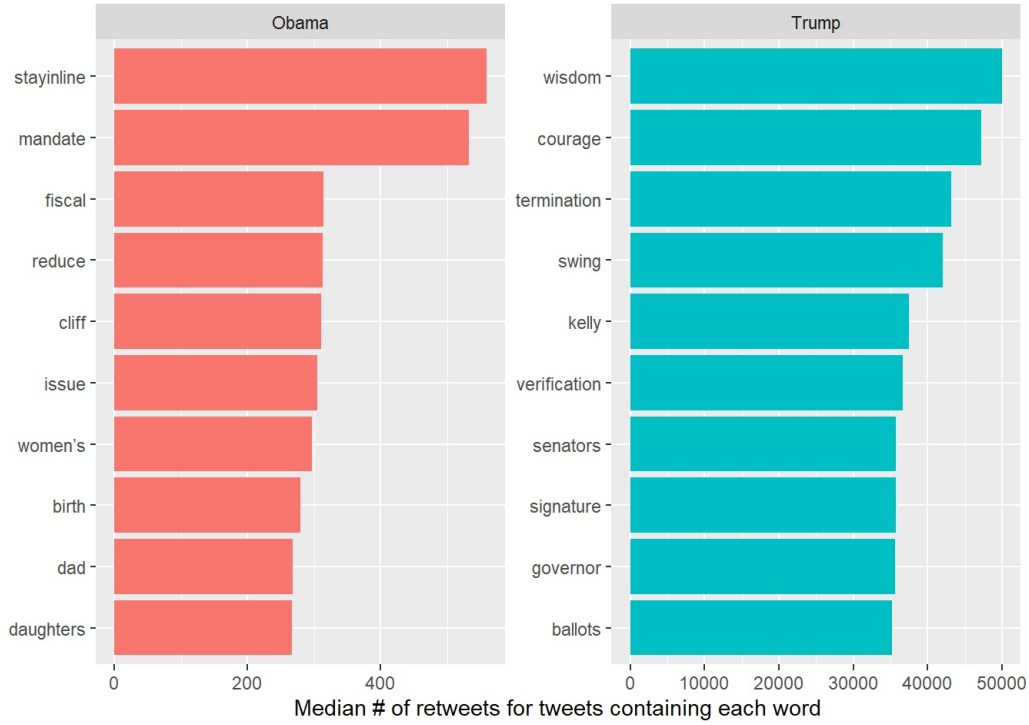
```
## # A tibble: 2 × 2
##   person total_rts
##   <chr>      <dbl>
## 1 Obama    3937563
## 2 Trump  225718674
```

Comparing retweet counts among the two presidents

```
word_by_rts <- tidy_tweets %>%
  group_by(id, word, person) %>%
  summarise(rts = first(retweets)) %>%
  group_by(person, word) %>%
  summarise(retweets = median(rts), uses = n()) %>%
  left_join(totals) %>%
  filter(retweets != 0) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'id', 'word'. You can override using the
## `.groups` argument.
## `summarise()` has grouped output by 'person'. You can override using the
## `.groups` argument.
## Joining with `by = join_by(person)`
```

```
word_by_rts %>%
  filter(uses >= 5) %>%
  group_by(person) %>%
  slice_max(retweets, n = 10) %>%
  arrange(retweets) %>%
  ungroup() %>%
  mutate(word = factor(word, unique(word))) %>%
  ungroup() %>%
  ggplot(aes(word, retweets, fill = person)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ person, scales = "free", ncol = 2) +
  coord_flip() +
  labs(x = NULL,
       y = "Median # of retweets for tweets containing each word")
```



Comparing favorites among the two presidents

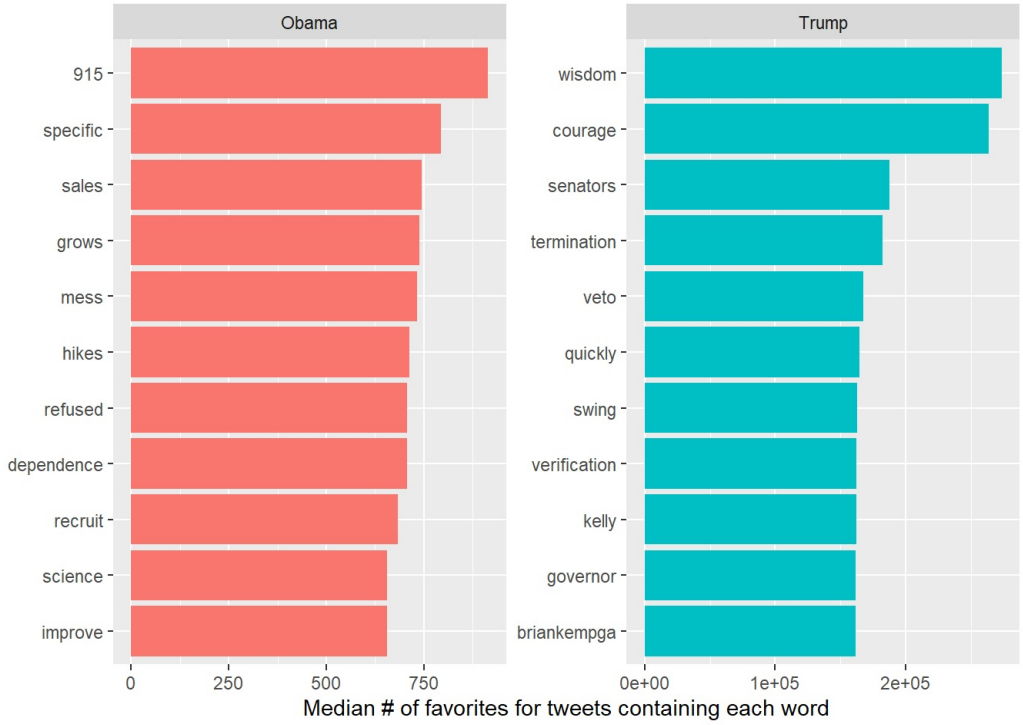
```
totals <- tidy_tweets %>%
  group_by(person, id) %>%
  summarise(favs = first(favorites)) %>%
  group_by(person) %>%
  summarise(total_favs = sum(favs))
```

```
## `summarise()` has grouped output by 'person'. You can override using the
## `.groups` argument.
```

```
word_by_favs <- tidy_tweets %>%
  group_by(id, word, person) %>%
  summarise(favs = first(favorites)) %>%
  group_by(person, word) %>%
  summarise(favorites = median(favs), uses = n()) %>%
  left_join(totals) %>%
  filter(favorites != 0) %>%
  ungroup()
```

```
## `summarise()` has grouped output by 'id', 'word'. You can override using the
## `.groups` argument.
## `summarise()` has grouped output by 'person'. You can override using the
## `.groups` argument.
## Joining with `by = join_by(person)`
```

```
word_by_favs %>%
  filter(uses >= 5) %>%
  group_by(person) %>%
  slice_max(favorites, n = 10) %>%
  arrange(favorites) %>%
  ungroup() %>%
  mutate(word = factor(word, unique(word))) %>%
  ungroup() %>%
  ggplot(aes(word, favorites, fill = person)) +
  geom_col(show.legend = FALSE) +
  facet_wrap(~ person, scales = "free", ncol = 2) +
  coord_flip() +
  labs(x = NULL,
       y = "Median # of favorites for tweets containing each word")
```


Obama
915
specific
sales
grows
mess
hikes
refused
dependence
recruit
science
improve
0
250
500
750
Trump
wisdom
courage
senators
termination
veto
quickly
swing
verification
kelly
governor
briankempga
0e+00
1e+05
2e+05
Median # of favorites for tweets containing each word