BlueWave

Aaron Shaffer

May 18, 2018

To start of my project I scraped real clear politics to gather data on polls over the past few months using 'scrapeRCP.py' this script created a 'senate.csv' and a 'house.csv'

This data was then scraped for pollnames to compare to the other website using getUniquePolls.py, which took the data from the above script and extracted all of the pollnames from RCP. But it turned out that there was very little overlap between these polls and the 538 data so this was not used.

Data in the 'senate.csv' and 'house.csv' output files also had to be manually cleaned by hand because the website structure of RCP is a microsoft word document saved as .html so their tables are not structured cleanly for easy scrapeing.

I also attempted to gather sentaors party affiliations from senate.gov using 'GetSenators.py', but instead the candidate summary action csv from the other data set had all of the information that was needed to get the part, so this script ended up not being used.

```
CSA <- read.csv('~/math485/BlueWaveProject/data/CandidateSummaryAction.csv', stringsAsFactors = FALSE)
senate <- read.csv("~/math485/BlueWaveProject/senate.csv")
house  <- read.csv("~/math485/BlueWaveProject/house.csv")
pander(head(senate))
```

Table 1: Table continues below

Date	Race	Poll
Wednesday April 25	Tennessee Senate - Blackburn vs. Bredesen	Mason-Dixon
Wednesday April 25	Tennessee Senate - Blackburn vs. Bredesen	Mason-Dixon
Wednesday April 25	Nevada Senate - Heller vs. Rosen	Nevada Independent/Mellman
Wednesday April 25	Nevada Senate - Heller vs. Rosen	Nevada Independent/Mellman
Tuesday April 24	West Virginia Senate - Republican Primary	FOX News
Tuesday April 24	West Virginia Senate - Republican Primary	FOX News

Results	Votes	Spread
Bredesen	46	Bredesen +3
Blackburn	43	Bredesen +3
Heller	40	Heller +1
Rosen	39	Heller +1
Jenkins	25	Jenkins +4
Morrisey	21	Jenkins +4

```
pander(head(house))
```

Table 3: Table continues below

Date	Race	Poll
Monday April 23	Arizona 8th District Special Election - Lesko vs. Tipirneni	Emerson
Monday April 23	Arizona 8th District Special Election - Lesko vs. Tipirneni	Emerson
Monday April 16	Arizona 8th District Special Election - Lesko vs. Tipirneni	Emerson
Monday April 16	Arizona 8th District Special Election - Lesko vs. Tipirneni	Emerson
Thursday April 12	Arizona 8th District Special Election - Lesko vs. Tipirneni	OH Predictive Insights
Thursday April 12	Arizona 8th District Special Election - Lesko vs. Tipirneni	OH Predictive Insights

Results	Votes	Spread
Lesko	49	Lesko +6
Tipirneni	43	Lesko +6
Lesko	45	Tipirneni +1
Tipirneni	46	Tipirneni +1
Lesko	53	Lesko +10
Tipirneni	43	Lesko +10

The columns from RCP are multiple columns in one so I used tidyr to split the columns into their sub categories.

Date was split into, "Weekday", "Month", and "Day".

Spread is a bit more confusing. This is also a column that had to be manually edited a bunch because the data never showed +0 if people tied in a poll. Spread was split into "Victor", and "Difference". Difference is the differnce separating the top two people in the poll. So if there were 100 votes and 1st place got 43 and 2nd place got 42 then the spread would be +1. Even if there were N other candidates who all got well under 40 votes only the top spread was recorded.

Additionally on their website if you tied then instead of saing "Winner +0" or "Tie +0", it simply says "Tie", and provided no number. Now thats what I call real clear. So, any NA for this value is 0. because they couldn't be consitent and say Tie +0

```
senate <- separate(senate,Date, into = c("Weekday","Month","Day"), convert = TRUE, sep = " ")
house <- separate(house,Date, into = c("Weekday","Month","Day"), convert = TRUE, sep = " ")
senate <- separate(senate,Spread, into = c("Victor","Difference"), convert = TRUE, sep = " ")
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 2 rows [108,
## 109].
```

```
house <- separate(house,Spread, into = c("Victor","Difference"), convert = TRUE, sep = " ")
```

```
## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 4 rows [23,
## 24, 31, 32].
```

```
senate$Difference[is.na(senate$Difference)] <- 0
house$Difference[is.na(house$Difference)] <- 0
```

Next I had to join the CSA and the RCP datasets in order to get party affiliations

I only used the candidate name, state and party affiliation columns from the CSA datset to do so.

```
CSA.small <- CSA[,c('can_nam','can_sta','can_par_aff')]
head(CSA.small)
```

```
##                  can_nam can_sta can_par_aff
## 1   ZIEGLER, EDWARD RAY      TX         REP
## 2 AALDERS, TIMOTHY NOEL      UT         CON
## 3       AARESTAD, DAVID      CO         DEM
## 4       ABATECOLA, BILL      AZ         REP
## 5        ABBOUD, DEEDRA      AZ         DEM
## 6   ABDULAHI, JAMAL MR.      MN         DEM
```

```
CSA.split <- separate(CSA.small,can_nam,into = c("Results","first_name"),sep=", ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 13 rows [228,
## 491, 495, 588, 589, 831, 1146, 1632, 2212, 2328, 2662, 3020, 3033].

## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 2 rows
## [2245, 2392].
```

```
CSA.split$Results <- toTitleCase(tolower(CSA.split$Results))

senate_new <- left_join(senate, CSA.split, by="Results")
```

```
## Warning: Column `Results` joining factor and character vector, coercing
## into character vector
```

```
house_new <- left_join(house,CSA.split, by="Results")
```

```
## Warning: Column `Results` joining factor and character vector, coercing
## into character vector
```

```
head(senate_new)
```

```
##     Weekday Month Day                                      Race
## 1 Wednesday April  25 Tennessee Senate - Blackburn vs. Bredesen
## 2 Wednesday April  25 Tennessee Senate - Blackburn vs. Bredesen
## 3 Wednesday April  25          Nevada Senate - Heller vs. Rosen
## 4 Wednesday April  25          Nevada Senate - Heller vs. Rosen
## 5   Tuesday April  24 West Virginia Senate - Republican Primary
## 6   Tuesday April  24 West Virginia Senate - Republican Primary
##                          Poll   Results Votes   Victor Difference
## 1                 Mason-Dixon  Bredesen    46 Bredesen          3
## 2                 Mason-Dixon Blackburn    43 Bredesen          3
## 3 Nevada Independent/Mellman    Heller    40   Heller          1
## 4 Nevada Independent/Mellman     Rosen    39   Heller          1
## 5                    FOX News   Jenkins    25  Jenkins          4
## 6                    FOX News   Jenkins    25  Jenkins          4
##         first_name can_sta can_par_aff
## 1           PHILIP      TN         DEM
## 2       MARSHA MRS      TN         REP
## 3             DEAN      NV         REP
## 4            JACKY      NV         DEM
## 5 ABE LINCOLN BRIAN      UT         REP
## 6           EVAN H      WV         REP
```

```
head(house_new)
```

```
##    Weekday Month Day
## 1   Monday April  23
## 2   Monday April  23
## 3   Monday April  16
## 4   Monday April  16
## 5 Thursday April  12
## 6 Thursday April  12
##                                                            Race
## 1 Arizona 8th District Special Election - Lesko vs. Tipirneni
## 2 Arizona 8th District Special Election - Lesko vs. Tipirneni
## 3 Arizona 8th District Special Election - Lesko vs. Tipirneni
## 4 Arizona 8th District Special Election - Lesko vs. Tipirneni
## 5 Arizona 8th District Special Election - Lesko vs. Tipirneni
## 6 Arizona 8th District Special Election - Lesko vs. Tipirneni
##                     Poll   Results Votes    Victor Difference first_name
## 1                Emerson     Lesko    49     Lesko          6     DEBBIE
## 2                Emerson Tipirneni    43     Lesko          6 HIRAL VYAS
## 3                Emerson     Lesko    45 Tipirneni          1     DEBBIE
## 4                Emerson Tipirneni    46 Tipirneni          1 HIRAL VYAS
## 5 OH Predictive Insights     Lesko    53     Lesko         10     DEBBIE
## 6 OH Predictive Insights Tipirneni    43     Lesko         10 HIRAL VYAS
##   can_sta can_par_aff
## 1      AZ         REP
## 2      AZ         DEM
## 3      AZ         REP
## 4      AZ         DEM
## 5      AZ         REP
## 6      AZ         DEM
```

In order for the columns to be merged since the way names and words were capitalized were different I had to do some fenageling to get the 'can_name' column to match the readable RCP dataset. The CSA can_nam column had to be split into first and last name which I renamed to match RCP for the left_join.

This part is what gave me the most headache.

```r
senate_new$Votes <- as.numeric(senate_new$Votes)
house_new$Votes <- as.numeric(house_new$Votes)

senate_aggvotes <- na.omit(senate) %>% group_by(Weekday, Month, Day,Race,Poll,Victor,Difference) %>% mu
house_aggvotes <- na.omit(house) %>% group_by(Weekday, Month, Day,Race,Poll,Victor,Difference) %>% muta

senate.final <- na.omit(left_join(senate_new,senate_aggvotes))
```

```
## Joining, by = c("Weekday", "Month", "Day", "Race", "Poll", "Results", "Votes", "Victor", "Difference

## Warning: Column `Results` joining character vector and factor, coercing
## into character vector
```

```r
house.final <- na.omit(left_join(house_new,house_aggvotes))
```

```
## Joining, by = c("Weekday", "Month", "Day", "Race", "Poll", "Results", "Votes", "Victor", "Difference

## Warning: Column `Results` joining character vector and factor, coercing
## into character vector
```

```r
senate.final$perVotes <- senate.final$Votes/senate.final$Total.Votes
house.final$perVotes <- house.final$Votes/house.final$Total.Votes
```

It was in the above code that I learned that I needed to manually touch up both RCP datasets. But this accomplished was aggreagating the total number of votes in each poll so that a % of support by candidate/state/party etc could be calculated.

```
senate.model <- glm(perVotes ~ 0 + Month + Day + can_sta + can_par_aff, data = senate.final)
summary(senate.model)
```

```
##
## Call:
## glm(formula = perVotes ~ 0 + Month + Day + can_sta + can_par_aff,
##     data = senate.final)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.36267  -0.04312   0.00225   0.05093   0.61411
##
## Coefficients:
##                  Estimate Std. Error t value Pr(>|t|)
## MonthApril      0.4962515  0.0386216  12.849  < 2e-16 ***
## MonthDecember   0.5281163  0.0263094  20.073  < 2e-16 ***
## MonthFebruary   0.4794102  0.0343060  13.975  < 2e-16 ***
## MonthJanuary    0.3612164  0.0415101   8.702  < 2e-16 ***
## MonthMarch      0.4652554  0.0368910  12.612  < 2e-16 ***
## Day            -0.0011964  0.0010664  -1.122  0.26260
## can_staAZ      -0.0310745  0.0461827  -0.673  0.50145
## can_staCA      -0.0475473  0.0360051  -1.321  0.18745
## can_staFL      -0.0314936  0.0333486  -0.944  0.34559
## can_staGA       0.0001939  0.0407284   0.005  0.99620
## can_staID      -0.1233448  0.1168390  -1.056  0.29180
## can_staIL       0.0031763  0.0336534   0.094  0.92486
## can_staIN      -0.0737548  0.0385158  -1.915  0.05627 .
## can_staKS      -0.0038468  0.0463539  -0.083  0.93391
## can_staMA      -0.2434333  0.1169199  -2.082  0.03802 *
## can_staMD      -0.1948187  0.0852572  -2.285  0.02287 *
## can_staMI       0.0130683  0.0400402   0.326  0.74432
## can_staMN       0.0567721  0.0545335   1.041  0.29853
## can_staMO       0.0788875  0.0638553   1.235  0.21746
## can_staMS       0.0632962  0.0506471   1.250  0.21217
## can_staMT      -0.1001000  0.1169199  -0.856  0.39247
## can_staNC      -0.0122573  0.0395044  -0.310  0.75652
## can_staND       0.0623657  0.0723494   0.862  0.38924
## can_staNJ       0.0458243  0.0638233   0.718  0.47322
## can_staNV      -0.0036647  0.0416169  -0.088  0.92988
## can_staNY       0.0262495  0.0348319   0.754  0.45156
## can_staOH      -0.0355004  0.0330007  -1.076  0.28274
## can_staOK      -0.1375436  0.0657742  -2.091  0.03719 *
## can_staOR      -0.2241096  0.0715220  -3.133  0.00186 **
## can_staPA       0.0030457  0.0399773   0.076  0.93931
## can_staSC      -0.0684191  0.0551373  -1.241  0.21543
## can_staTN       0.0048116  0.0505829   0.095  0.92427
## can_staTX      -0.0140056  0.0308613  -0.454  0.65022
## can_staUT       0.0091768  0.0539562   0.170  0.86504
## can_staVA      -0.0103686  0.0316663  -0.327  0.74352
## can_staWA      -0.0147453  0.0857310  -0.172  0.86353
## can_staWI       0.0010876  0.0353720   0.031  0.97549
```

```
## can_staWV      -0.1082444  0.0644060  -1.681  0.09367 .
## can_par_affGRE -0.1220367  0.0846985  -1.441  0.15047
## can_par_affIND -0.2597723  0.0824330  -3.151  0.00176 **
## can_par_affLIB -0.0101150  0.0336826  -0.300  0.76411
## can_par_affNNE -0.2678553  0.0854337  -3.135  0.00185 **
## can_par_affNOP  0.0378483  0.0492538   0.768  0.44272
## can_par_affREP -0.0232448  0.0134429  -1.729  0.08461 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.01277163)
##
##     Null deviance: 90.8642  on 417  degrees of freedom
## Residual deviance:  4.7638  on 373  degrees of freedom
## AIC: -591.44
##
## Number of Fisher Scoring iterations: 2
```

```r
senatepred.votes <- predict(senate.model,newdata = senate.final)
senate.END <- cbind(senate.final,senatepred.votes)
senate.total.votes <- senate.END %>% group_by(can_par_aff) %>% summarise(mean(senatepred.votes))
names(senate.total.votes) <- c("Party","MeanVotes")
pander(rbind(senate.total.votes$Party,round(senate.total.votes$MeanVotes * 33, digits = 3)))
```

DEM	GRE	IND	LIB	NNE	NOP	REP
15.273	9.955	6.27	15.373	2.896	16.787	14.539

```r
house.model <- glm(perVotes ~ 0 + Month + Day + can_sta + can_par_aff, data = house.final)
summary(house.model)
```

```
##
## Call:
## glm(formula = perVotes ~ 0 + Month + Day + can_sta + can_par_aff,
##     data = house.final)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.22079  -0.04556  -0.00138   0.02097   0.38855
##
## Coefficients: (3 not defined because of singularities)
##                  Estimate Std. Error t value Pr(>|t|)
## MonthApril      1.045e-01  4.751e-02   2.199 0.029048 *
## MonthFebruary   1.009e-01  5.911e-02   1.708 0.089311 .
## MonthJanuary    1.240e-01  8.070e-02   1.536 0.126128
## MonthJune       4.125e-01  5.590e-02   7.379 4.41e-12 ***
## MonthMarch      1.112e-01  5.549e-02   2.003 0.046540 *
## MonthMay        4.131e-01  6.828e-02   6.050 7.23e-09 ***
## MonthNovember   3.438e-01  7.267e-02   4.731 4.26e-06 ***
## MonthOctober    1.389e-01  1.040e-01   1.336 0.183056
## Day            -6.903e-04  1.767e-03  -0.391 0.696533
## can_staAK       2.347e-01  1.364e-01   1.721 0.086830 .
## can_staAL       4.368e-01  1.223e-01   3.570 0.000449 ***
## can_staAR       4.500e-02  6.128e-02   0.734 0.463620
## can_staAZ       4.308e-01  6.069e-02   7.097 2.27e-11 ***
```

```
## can_staCA       2.423e-16  5.740e-02   0.000 1.000000
## can_staCO       4.500e-02  6.128e-02   0.734 0.463620
## can_staGA       1.191e-01  4.551e-02   2.616 0.009579 **
## can_staIA       2.367e-01  1.096e-01   2.160 0.032017 *
## can_staIN       3.965e-01  6.651e-02   5.961 1.15e-08 ***
## can_staKS       1.923e-01  1.365e-01   1.409 0.160376
## can_staKY       6.425e-02  5.616e-02   1.144 0.254019
## can_staMD       3.723e-01  1.134e-01   3.285 0.001210 **
## can_staMI       9.320e-02  4.822e-02   1.933 0.054682 .
## can_staMS      -9.360e-17  5.740e-02   0.000 1.000000
## can_staMT      -1.018e-02  1.008e-01  -0.101 0.919661
## can_staNC       1.897e-01  1.353e-01   1.403 0.162347
## can_staNH       4.678e-01  1.409e-01   3.321 0.001068 **
## can_staNY       3.537e-01  1.159e-01   3.052 0.002587 **
## can_staPA       4.216e-01  5.482e-02   7.690 6.97e-13 ***
## can_staTX       1.667e-01  1.235e-01   1.350 0.178684
## can_staUT       4.818e-01  1.242e-01   3.879 0.000143 ***
## can_staVA       4.500e-02  6.128e-02   0.734 0.463620
## can_par_affGRE  9.654e-02  1.291e-01   0.748 0.455426
## can_par_affIND -6.151e-02  6.128e-02  -1.004 0.316731
## can_par_affLIB -1.182e-01  1.094e-01  -1.080 0.281378
## can_par_affREF         NA         NA      NA       NA
## can_par_affREP -4.500e-02  2.146e-02  -2.097 0.037257 *
## can_par_affUN          NA         NA      NA       NA
## can_par_affW           NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.01317746)
##
##     Null deviance: 25.1396  on 231  degrees of freedom
## Residual deviance:  2.5828  on 196  degrees of freedom
## AIC: -310.46
##
## Number of Fisher Scoring iterations: 2
```

```r
housepred.votes <- predict(house.model,newdata = house.final)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```r
house.END <- cbind(house.final,housepred.votes)
house.total.votes <- house.END %>% group_by(can_par_aff) %>% summarise(mean(housepred.votes))
names(house.total.votes) <- c("Party","MeanVotes")
pander(rbind(house.total.votes$Party,round(house.total.votes$MeanVotes * 435, digits = 3)))
```

DEM	GRE	IND	LIB	REF	REP	UN	W
131.002	230.602	34.033	76.958	41.215	118.656	41.215	233.036

Finally I looked at what variables I had left to build a model with and realized that the uniqueness of so many of the variables are completely meangless if you want to use the data to make predictions for other candidates so I ended up left with only the Month, Day of the month, State and Political party from all of the above data to predict support. In theory I could have left in additional columns from the CSA dataset to use as predictors but I didn't really realize that until just now doing this write up.

As far as predictions goes this is the mean % of votes across all states (in the dataset) expected of each party * number of seats available. Obviously our election system doesn't work like this. In order to estimate the true number of seats you would need to use this model to predict the expect % of support by party by state, and then use that to determine a victor on a state by state basis, and then extrapolate that to the number of seats per state. That would give you the expected number of seats of a given party by state. Atleast thats the theory behind the model(s).