Final Project - US Election 2020 - Race to Presidential Election 2020 by County

Rajasekhar Reddy Karna

2020-11-20

**##Session 1:**

##• Introduction:

##Watching news from sometimes and most of the times the discussions are on 2020 elections related. Even while watching YouTube or any other online video stuff, so many promotions of 2020 year elections and representative. So, thought to work on Presidential Election 2020 dataset, which inheritance helps me also to learn how USA election system work. This would also help me to understand geographic details of USA too along with each political party influence in respective states / districts / county level.

##• Research questions:

##Planning to identify the following questions to help investigate my problem statement. ##o How many presidential votes exist at count level?

##o How many candidate’s context in elections for presidential position from each party and number of votes individual got to be elected as presidential?

##o How many votes cumulatively used by citizen to elect presidential and any elector votes if exist any?

##o Are there any states / county level numbers so close by to determine the close proximity votes the losing party missed, which might help in future to recover from such counties.

##o Along with governor elections data analysis,

## o How does data analysis look like for congress house representative elections? ## How does data analysis look like for governor representative elections?

## o How does data analysis look like for senate representative elections?

##• Approach:

##- Perform dataset clean and identify variables.

##- Generate regression, correlation, basic summary, and descriptive statistics findings. ##- Plot findings to visualize the data results.

##• How your approach addresses (fully or partially) the problem:

##- Approach may give data analysis at state level and county level each political party got votes. Also see if any cross votes between Presidential votes vs Governor votes vs Senate votes vs Congress house representative votes.

##• Data:

##Reference: <https://www.kaggle.com/unanimad/us-election-2020>

##File Names:

## governors\_county\_candidate.csv

## president\_county\_candidate.csv

##• Required Packages:

##o “dplyr” – to filter, arrange or select variables

##o “readxl” / “readr” – to read data from csv files

##o “ggplot2” – to visualize the data

##o “lubridate” – to work with the dates ##o “colorspace” – to add colors to my visualizations

##• Plots and Table Needs:

##Scatterplots and Bar graphs ma provide most user friendly data analysis to give illustrate examples.

##• Questions for future steps:

##Need to do some exploratory search on how to apply a data analysis to see if any cross voting happened, as example presidential vote is to Republican party and Senate / Congress House vote is to Democratic party or other regional party based on available data. Since I am not very good on R programming yet with list of techniques and features it provides, will have to further research to determine which functions / libraries will provide best results to complete this project.

**##Section 2**

##• How to import and clean my data:

##Response:

##o Uploaded the data set and importing county / state level shapefiles (.shp).

##o Capturing vector points based on available data in the sets.

##o Based on read data and conditional logics, identify the results outcome of each geo.state wide details.

##o Created data frame based on the relevant data.

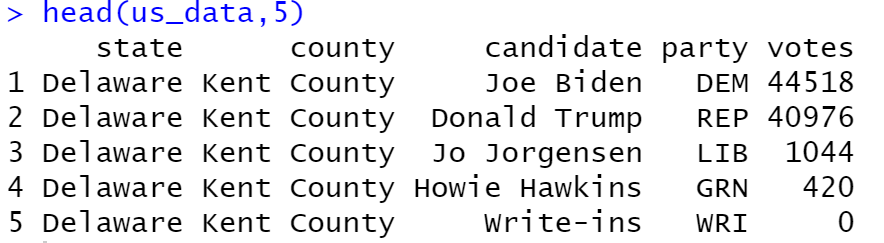
##o On demand basis, converting all string data into lower cases to use for data comparison between 2 different data sets.

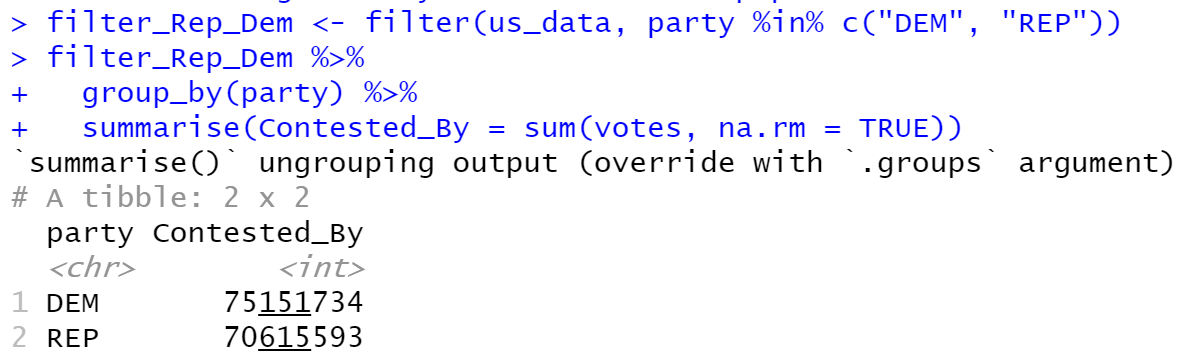
##o On demand basis changing the column headers to match between dataset for easy results plotting.

##• What does the final data set look like?

##Response:

##o Capturing different data parameters for analysis, sample as below:

##o 

##o 

##o 

##• What information is not self-evident?

##Response:

##o The data sets I picked has sample basis data. Means, data between 2 data sets has many mismatches which resulting difficult to plot data as expected on maps. Looking for other possible options to capture data that is available and display blank/white cells data on maps. From my experience the very important points are to view the entire picture while working with data. Planning to take some peer help from individual on expertise on analyzing the data.

##• What are different ways you could look at this data?

##Response:

##o Data can be looked at many ways, like in USA elections, we can see data as winner in each geo.state and same time the same geo.state can be used to replicate the looser party too. Also identifying the data portions sharing by at each geo.state level, margin of winning and losing elections, etc.. All these parameters can be better used to display the right values on ggplots.

##• How do you plan to slice and dice the data?

##Response:

##o Goal is to perform calculations and determine the elections winner at county level in each state and state level in the country. Also platter the party with color coding for better visibility. Also performing merging the data frames because of importance of data variable to use in data analysis.

##• How could you summarize your data to answer key questions?

##Response:

##o My plans to summarize the data based on calculated functions like sum(), levels(), max.col(), is.na(), group\_by(), summarise(), etc… These all are providing good analysis information that I required to plot the details.

##• What types of plots and tables will help you to illustrate the findings to your questions?

##Response:

##o For this project, my plan is to show scatterplots and bar graphs to provide most user friendly data analysis to give illustrate examples. Also goal is to provide a markdown file and associative graphs along with narration of analysis.

##• Do you plan on incorporating any machine learning techniques to answer your research questions? Explain.

##Response:

##o I wish I could invest time on ML techniques. But as the chosen data is more of existing data based on completed results and still there is a provision that can be used for predictive analysis. Due to time constraint and current chosen taks does not necessarily require to add machine learning to this data set. I think the statistical procedures would give all expected outcome based on available data.

##• Questions for future steps.

##Response:

##o Goals is based on USA elections 2020 results data is plot two types of graphs. Questions are primarily how to meet the objective with less number of coding lines. ## At county level, with mouse hover should display votes Trump or Biden got along with Margin both had with the name of winner.

## Display final winner at state level with Blue and Red colors with mouse hover as well.

##o Challenges:

## Currently having challenges to identify the winners of dataset based on different columns and rows specific wise. ## Currently having challenges to plot all details on a single plot to display 2 parties data at a time.

### Coding for Presidential Elections

options(warn=-1)  
#Defining all library files required to execute programming  
library(tidyr)  
library(dplyr)

##   
## 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(tibble)  
library(ggplot2)  
library(usmap)  
#Defining the data set path and importing data to perform analysis  
setwd("C:/Users/vahin/Documents/GitHub/dsc520/assignments/Final Project/")  
president\_county <- read.csv("data/president\_county\_candidate.csv")  
president\_county$party <- as.factor(president\_county$party)  
  
# List total political parties contested in 2020 elections  
levels(president\_county$party)

## [1] "ALI" "APV" "ASP" "BAR" "BFP" "BMP" "CST" "DEM" "GOP" "GRN" "IAP" "IND"  
## [13] "LIB" "LLC" "LLP" "NON" "OTH" "PRG" "PRO" "PSL" "REP" "SEP" "SWP" "UNA"  
## [25] "UTY" "WRI"

#Removing candidates data from system of records for now  
president\_county <- select(president\_county, -candidate)  
president\_county <- subset(president\_county, party %in% c("DEM","REP","LIB","GRN"))  
  
#Total Votes gained by Democratic vs Republican parties  
filter\_Rep\_Dem <- filter(president\_county, party %in% c("DEM", "REP"))  
filter\_Rep\_Dem %>%   
 group\_by(party) %>%   
 summarise(Total\_Votes = sum(votes, na.rm = TRUE))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 2 x 2  
## party Total\_Votes  
## <fct> <int>  
## 1 DEM 75151734  
## 2 REP 70615593

# Since there are multiple counties named the same way, adding a new attribute that will combine the first two columns together and then parse them our later.  
president\_county$state\_county <- as.factor(paste(president\_county$state,president\_county$county, sep = "\_and\_"))  
president\_county <- select(president\_county, -state, -county)  
#view the holistic data of state and county level details  
head(president\_county)

## party votes state\_county  
## 1 DEM 44518 Delaware\_and\_Kent County  
## 2 REP 40976 Delaware\_and\_Kent County  
## 3 LIB 1044 Delaware\_and\_Kent County  
## 4 GRN 420 Delaware\_and\_Kent County  
## 6 DEM 194238 Delaware\_and\_New Castle County  
## 7 REP 87685 Delaware\_and\_New Castle County

summary(president\_county)

## party votes state\_county   
## DEM :4633 Min. : 0 Alaska\_and\_ED 1 : 4   
## LIB :4633 1st Qu.: 47 Alaska\_and\_ED 10: 4   
## REP :4633 Median : 498 Alaska\_and\_ED 11: 4   
## GRN :3034 Mean : 8730 Alaska\_and\_ED 12: 4   
## ALI : 0 3rd Qu.: 4010 Alaska\_and\_ED 13: 4   
## APV : 0 Max. :2486527 Alaska\_and\_ED 14: 4   
## (Other): 0 (Other) :16909

#Spread the data in such a way each party received total number of votes at county level  
new\_president\_county <- spread(president\_county, key = party, value = votes)  
head(new\_president\_county)

## state\_county DEM GRN LIB REP  
## 1 Alabama\_and\_Autauga County 7450 NA 346 19764  
## 2 Alabama\_and\_Baldwin County 24344 NA 1221 83055  
## 3 Alabama\_and\_Barbour County 4772 NA 68 5605  
## 4 Alabama\_and\_Bibb County 1982 NA 73 7508  
## 5 Alabama\_and\_Blount County 2627 NA 209 24595  
## 6 Alabama\_and\_Bullock County 3439 NA 19 1143

#Now parse the state and county individually  
new\_president\_county <- separate(new\_president\_county, state\_county, c("State","County") , sep = "\_and\_")  
head(new\_president\_county, 50)

## State County DEM GRN LIB REP  
## 1 Alabama Autauga County 7450 NA 346 19764  
## 2 Alabama Baldwin County 24344 NA 1221 83055  
## 3 Alabama Barbour County 4772 NA 68 5605  
## 4 Alabama Bibb County 1982 NA 73 7508  
## 5 Alabama Blount County 2627 NA 209 24595  
## 6 Alabama Bullock County 3439 NA 19 1143  
## 7 Alabama Butler County 3953 NA 59 5448  
## 8 Alabama Calhoun County 15118 NA 548 34964  
## 9 Alabama Chambers County 6356 NA 140 8748  
## 10 Alabama Cherokee County 1619 NA 82 10562  
## 11 Alabama Chilton County 3056 NA 123 16052  
## 12 Alabama Choctaw County 3126 NA 38 4294  
## 13 Alabama Clarke County 5730 NA 48 7310  
## 14 Alabama Clay County 1262 NA 53 5589  
## 15 Alabama Cleburne County 672 NA 58 6472  
## 16 Alabama Coffee County 5050 NA 262 16832  
## 17 Alabama Colbert County 7057 NA 239 17311  
## 18 Alabama Conecuh County 2951 NA 32 3435  
## 19 Alabama Coosa County 1794 NA 46 3626  
## 20 Alabama Covington County 2717 NA 97 14579  
## 21 Alabama Crenshaw County 1700 NA 44 4864  
## 22 Alabama Cullman County 4454 NA 418 36804  
## 23 Alabama Dale County 5154 NA 232 14281  
## 24 Alabama Dallas County 12228 NA 90 5523  
## 25 Alabama DeKalb County 4271 NA 250 24744  
## 26 Alabama Elmore County 10304 NA 397 30089  
## 27 Alabama Escambia County 4894 NA 94 10844  
## 28 Alabama Etowah County 11487 NA 528 35343  
## 29 Alabama Fayette County 1390 NA 56 7295  
## 30 Alabama Franklin County 2085 NA 105 10364  
## 31 Alabama Geneva County 1592 NA 90 10844  
## 32 Alabama Greene County 3880 NA 8 875  
## 33 Alabama Hale County 4687 NA 37 3190  
## 34 Alabama Henry County 2589 NA 69 6593  
## 35 Alabama Houston County 12738 NA 525 32384  
## 36 Alabama Jackson County 3709 NA 208 19644  
## 37 Alabama Jefferson County 180936 NA 3568 138443  
## 38 Alabama Lamar County 978 NA 33 6168  
## 39 Alabama Lauderdale County 11872 NA 533 31578  
## 40 Alabama Lawrence County 3544 NA 126 12266  
## 41 Alabama Lee County 27600 NA 1012 42019  
## 42 Alabama Limestone County 13510 NA 736 34337  
## 43 Alabama Lowndes County 4968 NA 26 1835  
## 44 Alabama Macon County 7084 NA 52 1539  
## 45 Alabama Madison County 86885 NA 3678 102395  
## 46 Alabama Marengo County 5476 NA 59 5337  
## 47 Alabama Marion County 1457 NA 115 12190  
## 48 Alabama Marshall County 5880 NA 432 33094  
## 49 Alabama Mobile County 78754 NA 1920 100605  
## 50 Alabama Monroe County 4425 NA 53 6129

#if any party didn't get vote, mark such values (NA) to 0  
new\_president\_county[is.na(new\_president\_county)] <- 0  
parties <- c("DEM","GRN","LIB","REP")  
#Capture the winning party from each county and add to the new column as 'winner'  
new\_president\_county$winner <- parties[max.col(new\_president\_county[3:ncol(new\_president\_county)])]  
head(new\_president\_county, 50)

## State County DEM GRN LIB REP winner  
## 1 Alabama Autauga County 7450 0 346 19764 REP  
## 2 Alabama Baldwin County 24344 0 1221 83055 REP  
## 3 Alabama Barbour County 4772 0 68 5605 REP  
## 4 Alabama Bibb County 1982 0 73 7508 REP  
## 5 Alabama Blount County 2627 0 209 24595 REP  
## 6 Alabama Bullock County 3439 0 19 1143 DEM  
## 7 Alabama Butler County 3953 0 59 5448 REP  
## 8 Alabama Calhoun County 15118 0 548 34964 REP  
## 9 Alabama Chambers County 6356 0 140 8748 REP  
## 10 Alabama Cherokee County 1619 0 82 10562 REP  
## 11 Alabama Chilton County 3056 0 123 16052 REP  
## 12 Alabama Choctaw County 3126 0 38 4294 REP  
## 13 Alabama Clarke County 5730 0 48 7310 REP  
## 14 Alabama Clay County 1262 0 53 5589 REP  
## 15 Alabama Cleburne County 672 0 58 6472 REP  
## 16 Alabama Coffee County 5050 0 262 16832 REP  
## 17 Alabama Colbert County 7057 0 239 17311 REP  
## 18 Alabama Conecuh County 2951 0 32 3435 REP  
## 19 Alabama Coosa County 1794 0 46 3626 REP  
## 20 Alabama Covington County 2717 0 97 14579 REP  
## 21 Alabama Crenshaw County 1700 0 44 4864 REP  
## 22 Alabama Cullman County 4454 0 418 36804 REP  
## 23 Alabama Dale County 5154 0 232 14281 REP  
## 24 Alabama Dallas County 12228 0 90 5523 DEM  
## 25 Alabama DeKalb County 4271 0 250 24744 REP  
## 26 Alabama Elmore County 10304 0 397 30089 REP  
## 27 Alabama Escambia County 4894 0 94 10844 REP  
## 28 Alabama Etowah County 11487 0 528 35343 REP  
## 29 Alabama Fayette County 1390 0 56 7295 REP  
## 30 Alabama Franklin County 2085 0 105 10364 REP  
## 31 Alabama Geneva County 1592 0 90 10844 REP  
## 32 Alabama Greene County 3880 0 8 875 DEM  
## 33 Alabama Hale County 4687 0 37 3190 DEM  
## 34 Alabama Henry County 2589 0 69 6593 REP  
## 35 Alabama Houston County 12738 0 525 32384 REP  
## 36 Alabama Jackson County 3709 0 208 19644 REP  
## 37 Alabama Jefferson County 180936 0 3568 138443 DEM  
## 38 Alabama Lamar County 978 0 33 6168 REP  
## 39 Alabama Lauderdale County 11872 0 533 31578 REP  
## 40 Alabama Lawrence County 3544 0 126 12266 REP  
## 41 Alabama Lee County 27600 0 1012 42019 REP  
## 42 Alabama Limestone County 13510 0 736 34337 REP  
## 43 Alabama Lowndes County 4968 0 26 1835 DEM  
## 44 Alabama Macon County 7084 0 52 1539 DEM  
## 45 Alabama Madison County 86885 0 3678 102395 REP  
## 46 Alabama Marengo County 5476 0 59 5337 DEM  
## 47 Alabama Marion County 1457 0 115 12190 REP  
## 48 Alabama Marshall County 5880 0 432 33094 REP  
## 49 Alabama Mobile County 78754 0 1920 100605 REP  
## 50 Alabama Monroe County 4425 0 53 6129 REP

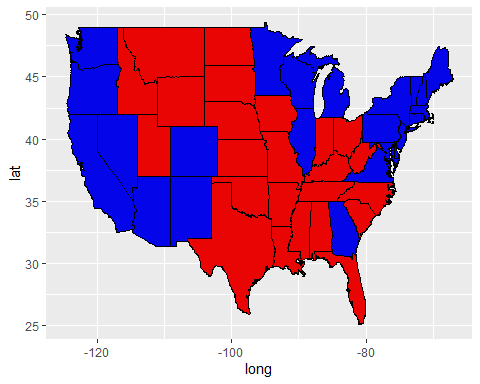
#Plot the graph at state level  
all\_states <- map\_data("state")  
# Taking off the county and winner from the dataset to plot state wide results  
new\_president\_state <- select(new\_president\_county, -County, -winner)  
# Grouping the data at state level based on number of votes got by each party  
new\_president\_state <- group\_by(new\_president\_state, State) %>% summarise\_all(sum)  
head (new\_president\_state)

## # A tibble: 6 x 5  
## State DEM GRN LIB REP  
## <chr> <int> <dbl> <int> <int>  
## 1 Alabama 843473 0 24994 1434159  
## 2 Alaska 45758 910 3478 80999  
## 3 Arizona 1571723 0 45394 1533268  
## 4 Arkansas 417898 2947 13039 755820  
## 5 California 8390640 53509 120136 4286488  
## 6 Colorado 1753416 8535 50469 1335253

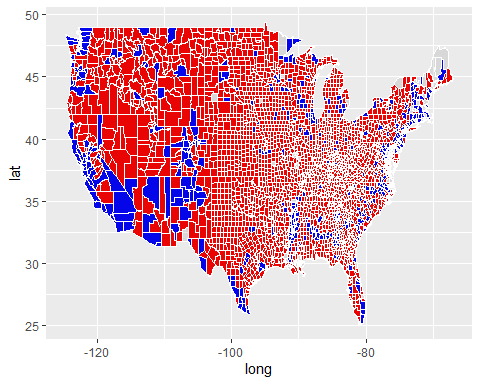
# Capture the winning party from each state and add to the new column as 'winner'  
new\_president\_state$winner <- parties[max.col(new\_president\_state[2:ncol(new\_president\_state)])]  
head(new\_president\_state, 50)

## # A tibble: 50 x 6  
## State DEM GRN LIB REP winner  
## <chr> <int> <dbl> <int> <int> <chr>   
## 1 Alabama 843473 0 24994 1434159 REP   
## 2 Alaska 45758 910 3478 80999 REP   
## 3 Arizona 1571723 0 45394 1533268 DEM   
## 4 Arkansas 417898 2947 13039 755820 REP   
## 5 California 8390640 53509 120136 4286488 DEM   
## 6 Colorado 1753416 8535 50469 1335253 DEM   
## 7 Connecticut 1045263 7369 20086 697599 DEM   
## 8 Delaware 295413 2135 4979 199857 DEM   
## 9 District of Columbia 258561 1302 1449 14449 DEM   
## 10 Florida 5286727 14669 70058 5659910 REP   
## # ... with 40 more rows

#Creating subset data with the winner as Democratic or Repulican party at state level  
demwins <- subset(new\_president\_state, winner == "DEM")  
repwins <- subset(new\_president\_state, winner == "REP")  
  
# To match the state names in map\_data vs data set actual data, defining all geo.states into lower cases  
blu <- tolower(as.vector(unique(demwins$State)))  
rd <- tolower(as.vector(unique(repwins$State)))  
  
#Plot the state wide election results got by DEM or REP parties  
state\_results <- ggplot(all\_states, aes(x=long, y=lat, group = group)) +  
 geom\_polygon(fill="blue", data = filter(all\_states, region %in% blu)) +  
 geom\_polygon(fill="red", data = filter(all\_states, region %in% rd)) +   
 geom\_polygon(color = "black", alpha = 0.1)  
state\_results



#Plot the graph at county level  
all\_counties <- map\_data("county")  
# grouping details at county level in each state based on subregion values  
all\_counties$merge <- paste(all\_counties$region,all\_counties$subregion, sep = "\_and\_")  
all\_counties$merge <- gsub(" ","",all\_counties$merge)  
  
# To match the county and state names in map\_data vs data set actual data, defining all counties and states into lower cases  
new\_president\_county$County <- tolower(gsub(paste0(c("County","Parish"),collapse = "|"),"", new\_president\_county$County))  
new\_president\_county$merge <- tolower(paste(new\_president\_county$State, new\_president\_county$County, sep = "\_and\_"))  
  
# grouping details at county level based on merged state & county data  
new\_president\_county$merge <- gsub(" ","",new\_president\_county$merge)  
  
#Creating subset data with the winner as Democratic or Repulican party at county level  
demwins <- subset(new\_president\_county, winner == "DEM")  
repwins <- subset(new\_president\_county, winner == "REP")  
  
# To match the county names in map\_data vs data set actual data, defining all counties into lower cases  
blu <- tolower(as.vector(unique(demwins$merge)))  
rd <- tolower(as.vector(unique(repwins$merge)))  
  
#Plot the county wide election results got by DEM or REP parties  
county\_results <- ggplot(all\_counties, aes(x=long, y=lat, group = group)) +  
 geom\_polygon(fill="blue", data = filter(all\_counties, merge %in% blu)) +  
 geom\_polygon(fill="red", data = filter(all\_counties, merge %in% rd)) +  
 geom\_polygon(color = "white", alpha = 0.1)  
county\_results



### Coding for Governor Elections in USA occured along with Presidential Elections. Happened in 12 geo.states:

# Delaware, Indiana, Missouri, Montana, New Hampshire, North Carolina, North Dakota, Utah, Vermont, Washington, West Virginia

options(warn=-1)  
#Defining all library files required to execute programming  
library(tidyr)  
library(dplyr)  
library(tibble)  
library(ggplot2)  
library(usmap)  
#Defining the data set path and importing data to perform analysis  
setwd("C:/Users/vahin/Documents/GitHub/dsc520/assignments/Final Project/")  
governor\_county <- read.csv("data/governors\_county\_candidate.csv")  
governor\_county$party <- as.factor(governor\_county$party)  
  
# List total political parties contested in 2020 elections  
levels(governor\_county$party)

## [1] "CST" "DEM" "GRN" "IAP" "IND" "IPD" "LIB" "MNP" "REP" "TM" "UNA" "WRI"

#Removing candidates data from system of records for now  
governor\_county <- select(governor\_county, -candidate)  
governor\_county <- subset(governor\_county, party %in% c("DEM","REP","LIB","GRN"))  
  
#Total Votes gained by Democratic vs Republican parties  
filter\_Rep\_Dem <- filter(governor\_county, party %in% c("DEM", "REP"))  
filter\_Rep\_Dem %>%   
 group\_by(party) %>%   
 summarise(Total\_Votes = sum(votes, na.rm = TRUE))

## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 2 x 2  
## party Total\_Votes  
## <fct> <int>  
## 1 DEM 8749024  
## 2 REP 10339911

# Since there are multiple counties named the same way, adding a new attribute that will combine the first two columns together and then parse them our later.  
governor\_county$state\_county <- as.factor(paste(governor\_county$state,governor\_county$county, sep = "\_and\_"))  
governor\_county <- select(governor\_county, -state, -county)  
#view the holistic data of state and county level details  
head(governor\_county)

## party votes state\_county  
## 1 DEM 44343 Delaware\_and\_Kent County  
## 2 REP 39325 Delaware\_and\_Kent County  
## 4 LIB 616 Delaware\_and\_Kent County  
## 5 DEM 190941 Delaware\_and\_New Castle County  
## 6 REP 81969 Delaware\_and\_New Castle County  
## 8 LIB 2013 Delaware\_and\_New Castle County

summary(governor\_county)

## party votes state\_county   
## DEM :1025 Min. : 0 Missouri\_and\_Adair County : 4   
## REP :1025 1st Qu.: 159 Missouri\_and\_Andrew County : 4   
## LIB : 740 Median : 775 Missouri\_and\_Atchison County: 4   
## GRN : 115 Mean : 6772 Missouri\_and\_Audrain County : 4   
## CST : 0 3rd Qu.: 3559 Missouri\_and\_Barry County : 4   
## IAP : 0 Max. :866430 Missouri\_and\_Barton County : 4   
## (Other): 0 (Other) :2881

#Spread the data in such a way each party received total number of votes at county level  
new\_governor\_county <- spread(governor\_county, key = party, value = votes)  
head(new\_governor\_county)

## state\_county DEM GRN LIB REP  
## 1 Delaware\_and\_Kent County 44343 NA 616 39325  
## 2 Delaware\_and\_New Castle County 190941 NA 2013 81969  
## 3 Delaware\_and\_Sussex County 56869 NA 622 68427  
## 4 Indiana\_and\_Adams County 2143 NA 2570 9440  
## 5 Indiana\_and\_Allen County 42790 NA 15324 82249  
## 6 Indiana\_and\_Bartholomew County 9013 NA 5064 21959

#Now parse the state and county individually  
new\_governor\_county <- separate(new\_governor\_county, state\_county, c("State","County") , sep = "\_and\_")  
head(new\_governor\_county, 50)

## State County DEM GRN LIB REP  
## 1 Delaware Kent County 44343 NA 616 39325  
## 2 Delaware New Castle County 190941 NA 2013 81969  
## 3 Delaware Sussex County 56869 NA 622 68427  
## 4 Indiana Adams County 2143 NA 2570 9440  
## 5 Indiana Allen County 42790 NA 15324 82249  
## 6 Indiana Bartholomew County 9013 NA 5064 21959  
## 7 Indiana Benton County 671 NA 614 2815  
## 8 Indiana Blackford County 932 NA 1017 3330  
## 9 Indiana Boone County 9661 NA 5094 23737  
## 10 Indiana Brown County 2349 NA 1601 4913  
## 11 Indiana Carroll County 1569 NA 1864 6077  
## 12 Indiana Cass County 3370 NA 2940 8831  
## 13 Indiana Clark County 19072 NA 3677 34662  
## 14 Indiana Clay County 1777 NA 2242 8163  
## 15 Indiana Clinton County 2406 NA 2514 7971  
## 16 Indiana Crawford County 1256 NA 555 3080  
## 17 Indiana Daviess County 1531 NA 2016 8313  
## 18 Indiana Dearborn County 4551 NA 2370 18272  
## 19 Indiana Decatur County 1560 NA 2787 7905  
## 20 Indiana DeKalb County 3293 NA 3158 13042  
## 21 Indiana Delaware County 15518 NA 5470 26710  
## 22 Indiana Dubois County 4586 NA 2599 14402  
## 23 Indiana Elkhart County 20097 NA 7641 45462  
## 24 Indiana Fayette County 1707 NA 1414 6933  
## 25 Indiana Floyd County 14945 NA 1669 24956  
## 26 Indiana Fountain County 1148 NA 1613 5186  
## 27 Indiana Franklin County 1690 NA 1499 8633  
## 28 Indiana Fulton County 1601 NA 2077 5411  
## 29 Indiana Gibson County 2990 NA 1533 11607  
## 30 Indiana Grant County 5849 NA 4325 16847  
## 31 Indiana Greene County 2511 NA 2604 9579  
## 32 Indiana Hamilton County 58699 NA 17116 117723  
## 33 Indiana Hancock County 8495 NA 8568 25484  
## 34 Indiana Harrison County 4508 NA 1599 14060  
## 35 Indiana Hendricks County 23060 NA 14226 50571  
## 36 Indiana Henry County 4127 NA 4442 12591  
## 37 Indiana Howard County 9832 NA 6294 24284  
## 38 Indiana Huntington County 2803 NA 3168 11759  
## 39 Indiana Jackson County 3234 NA 4188 11714  
## 40 Indiana Jasper County 2904 NA 2089 10378  
## 41 Indiana Jay County 1240 NA 1938 5227  
## 42 Indiana Jefferson County 3710 NA 1748 9074  
## 43 Indiana Jennings County 1930 NA 2676 7544  
## 44 Indiana Johnson County 17626 NA 12177 47453  
## 45 Indiana Knox County 2809 NA 1821 11209  
## 46 Indiana Kosciusko County 5774 NA 6838 23007  
## 47 Indiana LaGrange County 1690 NA 1365 7520  
## 48 Indiana Lake County 112301 NA 10036 94802  
## 49 Indiana LaPorte County 18126 NA 4356 26123  
## 50 Indiana Lawrence County 3616 NA 3881 13371

#if any party didn't get vote, mark such values (NA) to 0  
new\_governor\_county[is.na(new\_governor\_county)] <- 0  
parties <- c("DEM","GRN","LIB","REP")  
#Capture the winning party from each county and add to the new column as 'winner'  
new\_governor\_county$winner <- parties[max.col(new\_governor\_county[3:ncol(new\_governor\_county)])]  
head(new\_governor\_county, 50)

## State County DEM GRN LIB REP winner  
## 1 Delaware Kent County 44343 0 616 39325 DEM  
## 2 Delaware New Castle County 190941 0 2013 81969 DEM  
## 3 Delaware Sussex County 56869 0 622 68427 REP  
## 4 Indiana Adams County 2143 0 2570 9440 REP  
## 5 Indiana Allen County 42790 0 15324 82249 REP  
## 6 Indiana Bartholomew County 9013 0 5064 21959 REP  
## 7 Indiana Benton County 671 0 614 2815 REP  
## 8 Indiana Blackford County 932 0 1017 3330 REP  
## 9 Indiana Boone County 9661 0 5094 23737 REP  
## 10 Indiana Brown County 2349 0 1601 4913 REP  
## 11 Indiana Carroll County 1569 0 1864 6077 REP  
## 12 Indiana Cass County 3370 0 2940 8831 REP  
## 13 Indiana Clark County 19072 0 3677 34662 REP  
## 14 Indiana Clay County 1777 0 2242 8163 REP  
## 15 Indiana Clinton County 2406 0 2514 7971 REP  
## 16 Indiana Crawford County 1256 0 555 3080 REP  
## 17 Indiana Daviess County 1531 0 2016 8313 REP  
## 18 Indiana Dearborn County 4551 0 2370 18272 REP  
## 19 Indiana Decatur County 1560 0 2787 7905 REP  
## 20 Indiana DeKalb County 3293 0 3158 13042 REP  
## 21 Indiana Delaware County 15518 0 5470 26710 REP  
## 22 Indiana Dubois County 4586 0 2599 14402 REP  
## 23 Indiana Elkhart County 20097 0 7641 45462 REP  
## 24 Indiana Fayette County 1707 0 1414 6933 REP  
## 25 Indiana Floyd County 14945 0 1669 24956 REP  
## 26 Indiana Fountain County 1148 0 1613 5186 REP  
## 27 Indiana Franklin County 1690 0 1499 8633 REP  
## 28 Indiana Fulton County 1601 0 2077 5411 REP  
## 29 Indiana Gibson County 2990 0 1533 11607 REP  
## 30 Indiana Grant County 5849 0 4325 16847 REP  
## 31 Indiana Greene County 2511 0 2604 9579 REP  
## 32 Indiana Hamilton County 58699 0 17116 117723 REP  
## 33 Indiana Hancock County 8495 0 8568 25484 REP  
## 34 Indiana Harrison County 4508 0 1599 14060 REP  
## 35 Indiana Hendricks County 23060 0 14226 50571 REP  
## 36 Indiana Henry County 4127 0 4442 12591 REP  
## 37 Indiana Howard County 9832 0 6294 24284 REP  
## 38 Indiana Huntington County 2803 0 3168 11759 REP  
## 39 Indiana Jackson County 3234 0 4188 11714 REP  
## 40 Indiana Jasper County 2904 0 2089 10378 REP  
## 41 Indiana Jay County 1240 0 1938 5227 REP  
## 42 Indiana Jefferson County 3710 0 1748 9074 REP  
## 43 Indiana Jennings County 1930 0 2676 7544 REP  
## 44 Indiana Johnson County 17626 0 12177 47453 REP  
## 45 Indiana Knox County 2809 0 1821 11209 REP  
## 46 Indiana Kosciusko County 5774 0 6838 23007 REP  
## 47 Indiana LaGrange County 1690 0 1365 7520 REP  
## 48 Indiana Lake County 112301 0 10036 94802 DEM  
## 49 Indiana LaPorte County 18126 0 4356 26123 REP  
## 50 Indiana Lawrence County 3616 0 3881 13371 REP

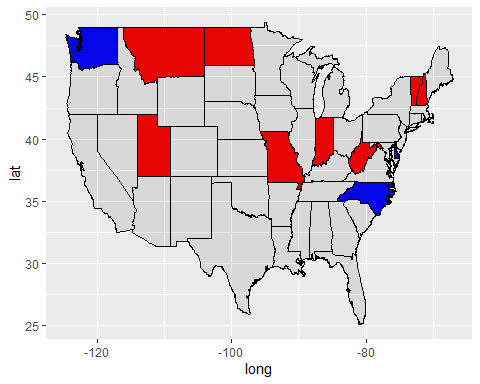
#Plot the graph at state level  
all\_states <- map\_data("state")  
# Taking off the county and winner from the dataset to plot state wide results  
new\_governor\_state <- select(new\_governor\_county, -County, -winner)  
# Grouping the data at state level based on number of votes got by each party  
new\_governor\_state <- group\_by(new\_governor\_state, State) %>% summarise\_all(sum)  
head (new\_governor\_state)

## # A tibble: 6 x 5  
## State DEM GRN LIB REP  
## <chr> <int> <dbl> <dbl> <int>  
## 1 Delaware 292153 0 3251 189721  
## 2 Indiana 955314 0 344353 1688047  
## 3 Missouri 1216392 17121 48843 1713153  
## 4 Montana 247777 0 23614 321603  
## 5 New Hampshire 261817 0 11182 508782  
## 6 North Carolina 2804293 0 58833 2563585

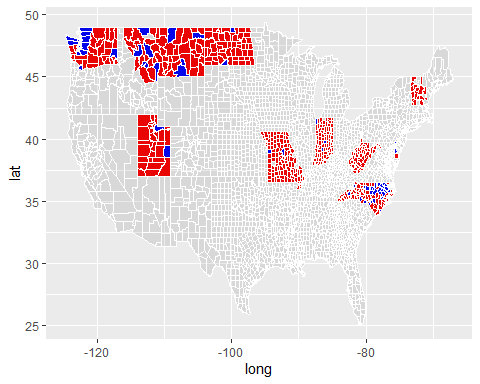
# Capture the winning party from each state and add to the new column as 'winner'  
new\_governor\_state$winner <- parties[max.col(new\_governor\_state[2:ncol(new\_governor\_state)])]  
head(new\_governor\_state, 50)

## # A tibble: 11 x 6  
## State DEM GRN LIB REP winner  
## <chr> <int> <dbl> <dbl> <int> <chr>   
## 1 Delaware 292153 0 3251 189721 DEM   
## 2 Indiana 955314 0 344353 1688047 REP   
## 3 Missouri 1216392 17121 48843 1713153 REP   
## 4 Montana 247777 0 23614 321603 REP   
## 5 New Hampshire 261817 0 11182 508782 REP   
## 6 North Carolina 2804293 0 58833 2563585 DEM   
## 7 North Dakota 90595 0 13940 234905 REP   
## 8 Utah 373828 0 40803 786336 REP   
## 9 Vermont 99134 0 0 248185 REP   
## 10 Washington 2174123 0 0 1593591 DEM   
## 11 West Virginia 233598 0 22112 492003 REP

#Creating subset data with the winner as Democratic or Repulican party at state level  
demwins <- subset(new\_governor\_state, winner == "DEM")  
repwins <- subset(new\_governor\_state, winner == "REP")  
  
# To match the state names in map\_data vs data set actual data, defining all geo.states into lower cases  
blu <- tolower(as.vector(unique(demwins$State)))  
rd <- tolower(as.vector(unique(repwins$State)))  
  
#Plot the state wide election results got by DEM or REP parties  
state\_results <- ggplot(all\_states, aes(x=long, y=lat, group = group)) +  
 geom\_polygon(fill="blue", data = filter(all\_states, region %in% blu)) +  
 geom\_polygon(fill="red", data = filter(all\_states, region %in% rd)) +   
 geom\_polygon(color = "black", alpha = 0.1)  
state\_results



#Plot the graph at county level  
all\_counties <- map\_data("county")  
# grouping details at county level in each state based on subregion values  
all\_counties$merge <- paste(all\_counties$region,all\_counties$subregion, sep = "\_and\_")  
all\_counties$merge <- gsub(" ","",all\_counties$merge)  
  
# To match the county and state names in map\_data vs data set actual data, defining all counties and states into lower cases  
new\_governor\_county$County <- tolower(gsub(paste0(c("County","Parish"),collapse = "|"),"", new\_governor\_county$County))  
new\_governor\_county$merge <- tolower(paste(new\_governor\_county$State, new\_governor\_county$County, sep = "\_and\_"))  
  
# grouping details at county level based on merged state & county data  
new\_governor\_county$merge <- gsub(" ","",new\_governor\_county$merge)  
  
#Creating subset data with the winner as Democratic or Repulican party at county level  
demwins <- subset(new\_governor\_county, winner == "DEM")  
repwins <- subset(new\_governor\_county, winner == "REP")  
  
# To match the county names in map\_data vs data set actual data, defining all counties into lower cases  
blu <- tolower(as.vector(unique(demwins$merge)))  
rd <- tolower(as.vector(unique(repwins$merge)))  
  
#Plot the county wide election results got by DEM or REP parties  
county\_results <- ggplot(all\_counties, aes(x=long, y=lat, group = group)) +  
 geom\_polygon(fill="blue", data = filter(all\_counties, merge %in% blu)) +  
 geom\_polygon(fill="red", data = filter(all\_counties, merge %in% rd)) +  
 geom\_polygon(color = "white", alpha = 0.1)  
county\_results



**##Section 3**

##• A story / narrative that emerged from your data. Follow this structure.

##o Introduction:

#Watching USA news from sometime and most of the times the discussions are on 2020 USA presidential elections related. Even while watching YouTube or any other online videos, many promotions are associated to 2020 year elections and representative. So, thought to work on Presidential Election 2020 data set, which inheritance helped me to learn how USA election system working. This also helped me to understand geographic details of USA along with each political party influence in respective states / districts / county level in age of 2020.

#The data set has been captured from Kaggle in CSV file format and it contains data of presidential and governor elections happened in few geo.states across USA. The provided data contains information like geo.state with in USA, county in each geo.state, candidate contested in elections, associate candidate polical party details along with total votes gained by each party to win the 2020 elections.

#Considering the fact of provided data has more insights and metrics are more helpful to create relationship between attributes. By developing various plots at state and county level, summary tables, visualization got an opportunity to show case various dimension of election results data analysis. Through this clear graphical observations were able to be to tell the winning party at county, state and county level both for Presidential and Governor elections.

##o The problem statement you addressed:

# Initially I faced separate the data based on political party and total number of votes captured in each county level. Since the political party data present in row wise, converting it to column level become little hurdle. To plot the data at each state level and county level winning parties based on number of votes, summed data with group by function which eventually printed same in plot with color codes.

##o How you addressed this problem statement:

#A step by step process has been take into consideration to address the problem statements and plot the details in graphical visible format.

#1. Analysis and Data Preparation:

# - Captured data from Kaggela and loaded the same through read.csv function in R # - Analyzed the data to get initial understand of avaiable attributes and possible solutions that can be articulated with R programming

# - While programming, intalled and loaded all the required Libraries needed for project. # - On needed named the columns names and also converted into lower case letters to make comparision success.

#2. Clean Up:

# - Removed not requried data columns

# - Spread the data based on political party and total votes gained at rount level # - Initially combined state and county level since there is possibility of having same county in differnt state. By combining, would be able to create unique identified and able to separate data for each political party.

# - Defining 0, if any particular political party hasn’t got any vote in a given county region # - Captured winning party based on max votes gainied by the party at state or county level and added the winner in a new column.

#3. Data Analysis and Plots:

# - Created subset of each party details (picked major leading parties only into consideration)

# - Also converted state and county names into lower case to match with the usa mapping details.

# - Plotted the details with party color symbols to make more realistic representation.

##o Analysis:

# The main reference and objective of this project is to plot the winning party of USA 2020 elections in each county and state level based on number of votes gaining by each party (predominently either Democratic or Republican party)

#Presidential Elections:

# Based on statistical analysis made on available data, the clear victory is given to Democratic party in majority geo.state.

# In terms of total votes gained at party level for Presidential elections, Democratic party earned total of 75151734 votes & Republican party earned total of 70615593 votes.

# There is a margin of 4536131 votes Republican party lost the winning USA 2020 elections against Democratic party.

# Out of 50 geo.states in USA, 24 geo.states given winning numbers to Republican party and remaining 26 geo.states given winnding numbers to Democratic party in Presidential elections.

# Based on county level graph for Presidential elections, though the color is showing more of red (Republican party), based on density of people total votes turned to blue (Domocratic party) overal.

#Governer Elections:

# Based on statistical analysis made on available data, the clear victory is given to Republican party in majority geo.state.

# In terms of total votes gained at party level for Governer elections, Democratic party earned total of 8749024 votes & Republican party earned total of 10339911 votes.

# There is a margin of 1590887 votes Democratic party lost the winning USA 2020 Governer elections against Republican party for vote casting happened in 11 geo.states.

# Out of 11 geo.states in USA, 8 geo.states given winning numbers to Republican party and remaining 3 geo.states given winnding numbers to Democratic party in Governer elections.

# Based on county level graph for Governer elections, Republican party showed clear majority at many places.

# Even in 3 geo.state lost by Republican party, total counties owned by Republican party. But based on people density in remaining county regions is more compared with regions own by republican party, win goes to Democratic party.

##o Implications.

# Observed the overall counties won by Replican party in Presidential elections, due to people density living in other counties is more than geographic location. Offcourse the people density depends on many parameters like city and suburban regions etc..

# In 2 geo states though Presidential elections got majority for Democratic party, the Governor elections results winning given to Repulican party. Same way in 1 geo state though Presidential electiosn got majority for Republican party, the Governor elections results winning given to Democratic party. This is giving a clear representation of cross voting happened between Presidential elections vs Governor elections. This also giving a clear representation of people casting votes based on candidate contesting in elections, not only on basis of policatical party symbol / color.

##o Limitations.

# The main point, this project analysis, data preparation and execution empowered me to study and understand the USA elections 2020 results based on geo.states as well as governor elections. This gave me the insight about how geo.graphically a political party taking opportunities to keep their winning strategies. # I feel that, we can leverage this project analysis further to perform cross voting happened at each county level by comparing Presidential elections and Governor elections. This might give an increased predictive model anaysis for future elections for Governor elections in geo.states. # Also I feel that, we can leverage this project analysis data to plot different types of graphs with mouse cursor data display at each county and geo.state level with the margin votes own by each contested candidate for both PResidential and Governor elections.

##o Concluding Remarks

# Yes, a very good opportunity to learn R programming and convert some foundational level learning points from the class course into project needs in real time. Since data analysis has boundaryless opportunities, has ways to plot data in many ways to read different combinations for future data references. A bottom line, a lot more to learn in data analysis and it is continous journey and it changes based on business needs and demands.