Master file

2023-10-17

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ dplyr     1.1.2     ✓ readr     2.1.4
## ✓ forcats   1.0.0     ✓ stringr   1.5.0
## ✓ ggplot2   3.4.2     ✓ tibble    3.2.1
## ✓ lubridate 1.9.2     ✓ tidyr     1.3.0
## ✓ purrr     1.0.1
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to becom
e errors
```

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
##
## The following object is masked from 'package:ggplot2':
##
##     last_plot
##
## The following object is masked from 'package:stats':
##
##     filter
##
## The following object is masked from 'package:graphics':
##
##     layout
```

```
library(RColorBrewer)
library(sf)
```

```
## Linking to GEOS 3.11.2, GDAL 3.6.2, PROJ 9.2.0; sf_use_s2() is TRUE
```

```
library(readxl)
```

Loading in data files

```
load("Master.RData")
map21 <- st_zm(st_read("PRECINCT2021_0311.shp"))
```

```
## Reading layer `PRECINCT2021_0311' from data source
##   `C:\Users\nddan\OneDrive\Documents\210democracy\Final Presentation\PRECINCT2021_0311.shp'
##   using driver `ESRI Shapefile'
## Simple feature collection with 564 features and 1 field
## Geometry type: MULTIPOLYGON
## Dimension:     XY, XYZ
## Bounding box:  xmin: -84.8203 ymin: 39.02153 xmax: -84.25651 ymax: 39.31206
## z_range:       zmin: 0 zmax: 0
## Geodetic CRS:  NAD83
```

```
results21 <-  read_excel("G21_Official_Canvass.xlsx", sheet = "Cincinnati", skip = 2)
voters <- read_csv("VoterListExport-20231019-no.csv")
```

```
## New names:
## Rows: 597289 Columns: 43
## ── Column specification
## ──────────────────────────────────────────────────────── Delimiter: "," chr
## (32): VoterIdent, PrecinctName, RegisteredDate, FirstName, LastName, Sta... dbl
## (5): PrecinctNum, PrecinctSplit, BirthYear, AddressNumber, AddressZip lgl (6):
## MiddleName...7, SuffixName...9, MiddleName...10, SuffixName...11, ...
## 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.
## • `MiddleName` -> `MiddleName...7`
## • `SuffixName` -> `SuffixName...9`
## • `MiddleName` -> `MiddleName...10`
## • `SuffixName` -> `SuffixName...11`
```

Cleaning and wrangling the data.

```
results21[51, "PRECINCT"] <- "0605 CIN 6-E"

temp <- right_join(map21, results21, by = c("NAME" = "PRECINCT"))
```

Creating percentages for how Mark Jeffreys and Jan-Michele Lemon Kearney preformed in each Cincinnati precinct. Averaging those percentage to represent how we believe our candidate will preform. Adding that percentage to the original data frame for further analysis.

```
temp <- temp %>%
  mutate(`Mark Jeffreys percentage` = `Mark       Jeffreys`/`BALLOTS CAST TOTAL`)

temp <- temp %>%
  mutate(`Kearney percentage` = `Jan-Michele Lemon Kearney`/`BALLOTS CAST TOTAL`)

temp <- temp %>%
  mutate(Jeff_Kearney = (`Mark Jeffreys percentage` + `Kearney percentage`) / 2)

temp <- temp %>%
  mutate(base_swing = cut(Jeff_Kearney, breaks = c(-.1, 0.3, 0.5, 1), label = c("Residual", "Swi
ng", "Base")))

temp2 <- temp %>%
  select(Jeff_Kearney, base_swing, NAME)

combined5 <-
  left_join(combined5,
                     st_drop_geometry(temp2),
                     by = c("NAME"= "NAME"))
```

Cleaning and wrangling the voters data frame

```
voters3 <- voters %>%
  mutate(`PRC #` = PrecinctNum) %>%
  select(`PRC #`, PartyCode)
```

Calculating a percentage for democrats in each precinct

```
voters2 <- voters3 %>%
  group_by(`PRC #`, PartyCode) %>%
  summarize(count = n(), .groups = 'drop')


voters2 <- voters2 %>%
  pivot_wider(names_from = `PartyCode`, values_from = count, values_fill = 0)

combined5 <- left_join(combined5, voters2, by = c("PRC #"= "PRC #"))

combined5 <- combined5 %>%
  mutate(reg_voters = D + R)

combined5 <- combined5 %>%
  mutate(dem_percent = D/reg_voters)

combined5[79, 29] <- NA

combined5[51, "dem_percent"] <- 0.68556701
combined5 <- combined5 %>%
  mutate(dem_percent = as.numeric(dem_percent))
```

Building a formula that predicts if a voter will vote in the upcoming election based on their voting patterns from the last 11 elections

```
voters3 <- voters %>%
  filter(grepl("CIN", PrecinctName))
voters3 <- voters3 %>%
  select(VoterIdent, GENERAL_NOV_2023, `2023 August Election`, PRIMARY_MAY_2023, GENERAL_NOV_202
2, `AUG PRIMARY ELECTION 2022`,
         PRIMARY_MAY_2022, GENERAL_NOV_2021, PRIMARY_MAY_2021, GENERAL_NOV_2020, SPECIAL_AUG_202
0, PRIMARY_MARCH_2020, GENERAL_NOV_2019,
         FirstName, LastName, PartyCode, AddressNumber, AddressStreet, AddressSuffix, AddressZi
p)
voters3 <- voters3 %>%
  mutate(`2023 August Election` = ifelse(is.na(`2023 August Election`), 0, 3),
         PRIMARY_MAY_2023 = ifelse(is.na(PRIMARY_MAY_2023), 0, 2),
         GENERAL_NOV_2022 = ifelse(is.na(GENERAL_NOV_2022), 0, 1.5),
         `AUG PRIMARY ELECTION 2022` = ifelse(is.na(`AUG PRIMARY ELECTION 2022`), 0, 1.5),
         PRIMARY_MAY_2022 = ifelse(is.na(PRIMARY_MAY_2022), 0, 1.5),
         GENERAL_NOV_2021 = ifelse(is.na(GENERAL_NOV_2021), 0, 1.25),
         PRIMARY_MAY_2021 = ifelse(is.na(PRIMARY_MAY_2021), 0, 1.25),
         GENERAL_NOV_2020 = ifelse(is.na(GENERAL_NOV_2020), 0, 1),
         SPECIAL_AUG_2020 = ifelse(is.na(SPECIAL_AUG_2020), 0, 1.25),
         PRIMARY_MARCH_2020 = ifelse(is.na(PRIMARY_MARCH_2020), 0, 1.25),
         GENERAL_NOV_2019 = ifelse(is.na(GENERAL_NOV_2019), 0, 1)
         )

subset_data <- voters3[, 3:13]


row_means <- rowMeans(subset_data, na.rm = TRUE)
voters3$Row_Means_3_to_13 <- row_means

get_voters <- voters3 %>%
  filter(Row_Means_3_to_13 >= 0.3 & Row_Means_3_to_13 <= 0.6) %>%
  filter(PartyCode %in% c("D", "U"))


exp_voters <- voters3 %>%
  filter(Row_Means_3_to_13 >= 0.4)
votes <- 72550
```

```
combined5 %>%
  filter(base_swing == "Base") %>%
  mean(dem_percent, na.rm = T)
```

```
## Warning in mean.default(., dem_percent, na.rm = T): argument is not numeric or
## logical: returning NA
```

```
## [1] NA
```

A scatter plot that shows the relationship between the percentage of white people in each precinct, and our predicted performance in each precinct

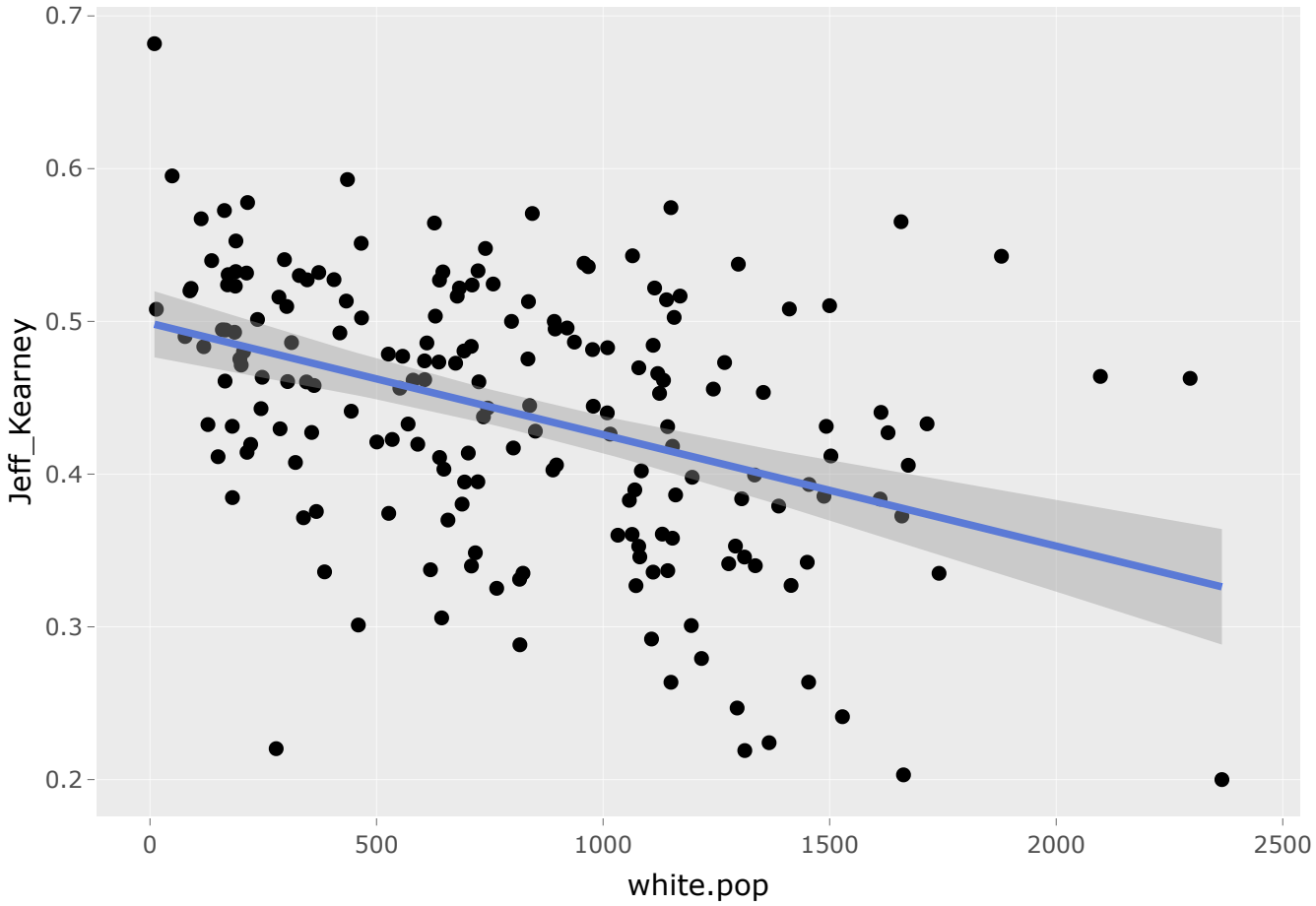
```
p<-
combined5 %>%
  ggplot(aes(x=white.pop,y=Jeff_Kearney))+
  geom_point(aes(text=NAME))+
  geom_smooth(method = "lm", se = TRUE)
```

```
## Warning in geom_point(aes(text = NAME)): Ignoring unknown aesthetics: text
```

```
ggplotly(p)
```

```
## `geom_smooth()` using formula = 'y ~ x'
```

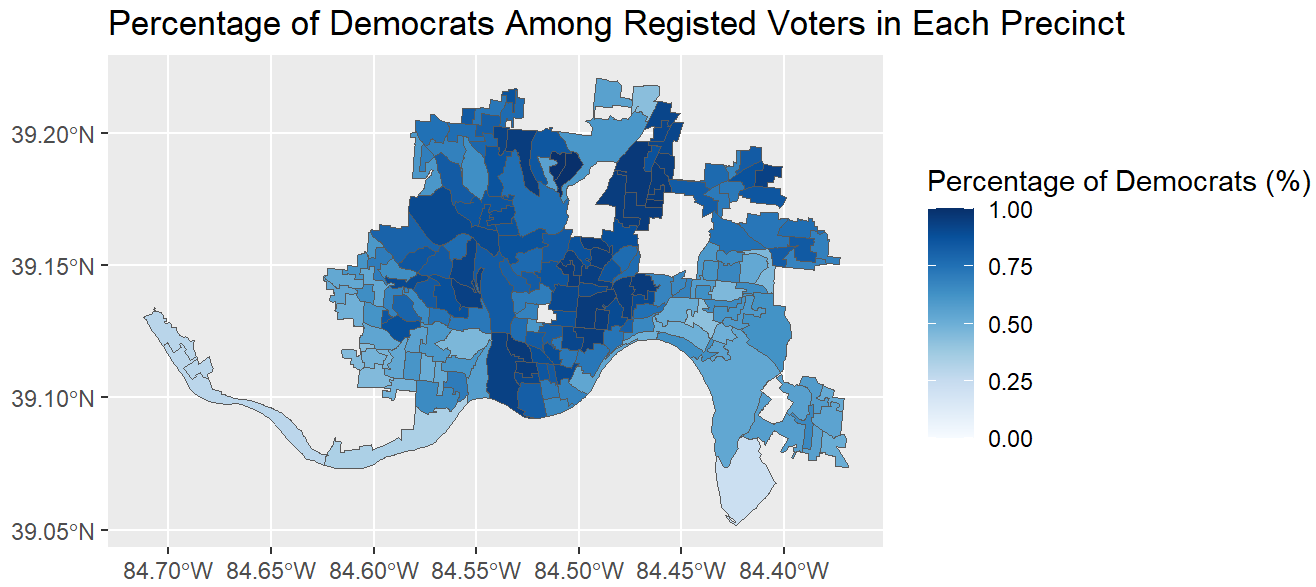
```
## Warning: Removed 374 rows containing non-finite values (`stat_smooth()`).
```



A graph of Cincinnati that shows the percentage of democrats in each precinct

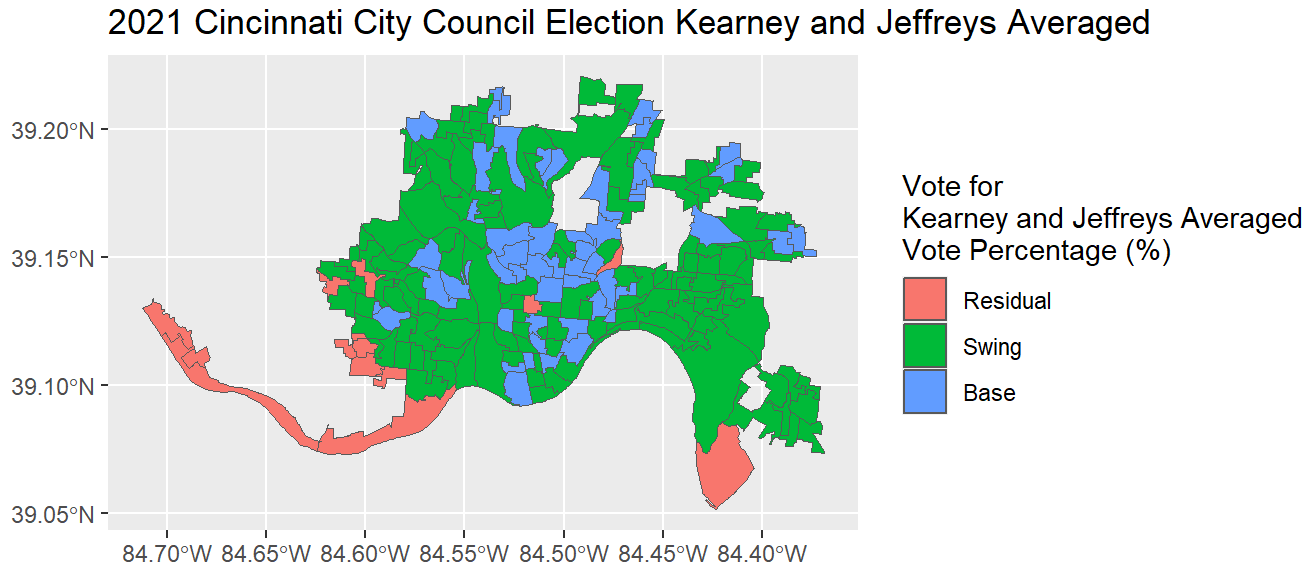
```
combined5 %>%
  filter(grepl("CIN", NAME)) %>%
  ggplot(aes(fill = dem_percent))+
  geom_sf()+
  labs(title = "Percentage of Democrats Among Registed Voters in Each Precinct",
       fill = "Percentage of Democrats (%)")+
  scale_fill_gradientn(colours = brewer.pal(n = 10, name = "Blues"), na.value = "transparent",
                       limits = c(0, 1))
```

```
## Warning in brewer.pal(n = 10, name = "Blues"): n too large, allowed maximum for palette Blues
is 9
## Returning the palette you asked for with that many colors
```



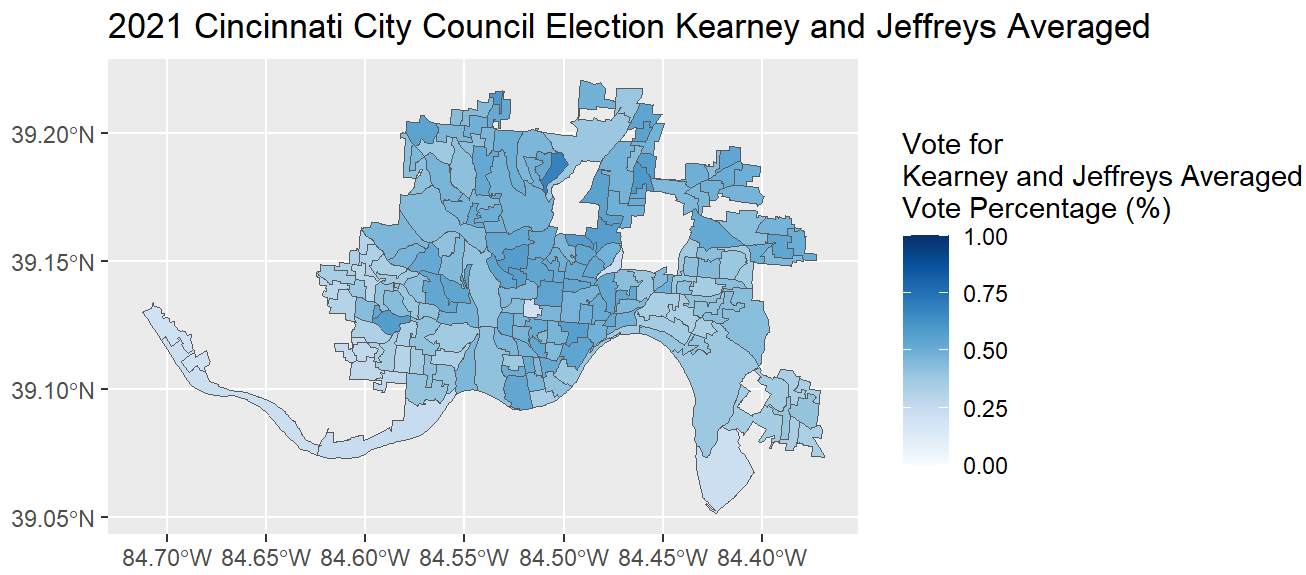
A base swing residual map of the Cincinnati voting precincts. Base precincts represent precincts we are confident our candidate will win. Residual precincts are those that our candidate probably won't win. Swing precincts could go either way

```
combined5 %>%
  filter(grepl("CIN", NAME)) %>%
  ggplot(aes(fill = base_swing)) +
  geom_sf() +
  labs(title = "2021 Cincinnati City Council Election Kearney and Jeffreys Averaged",
       fill = "Vote for \nKearney and Jeffreys Averaged \nVote Percentage (%)",
       caption = "")
```



A graph of Cincinnati that shows the predicted percentage of votes our candidate will receive in each precinct

```
combined5 %>%
  filter(grepl("CIN", NAME)) %>%
  ggplot(aes(fill = Jeff_Kearney)) +
  geom_sf() +
  labs(title = "2021 Cincinnati City Council Election Kearney and Jeffreys Averaged",
       fill = "Vote for \nKearney and Jeffreys Averaged \nVote Percentage (%)",
       caption = "") +
  scale_fill_gradientn(colours = brewer.pal(n = 9, name = "Blues"), na.value = "transparent",
                       limits = c(0, 1))
```

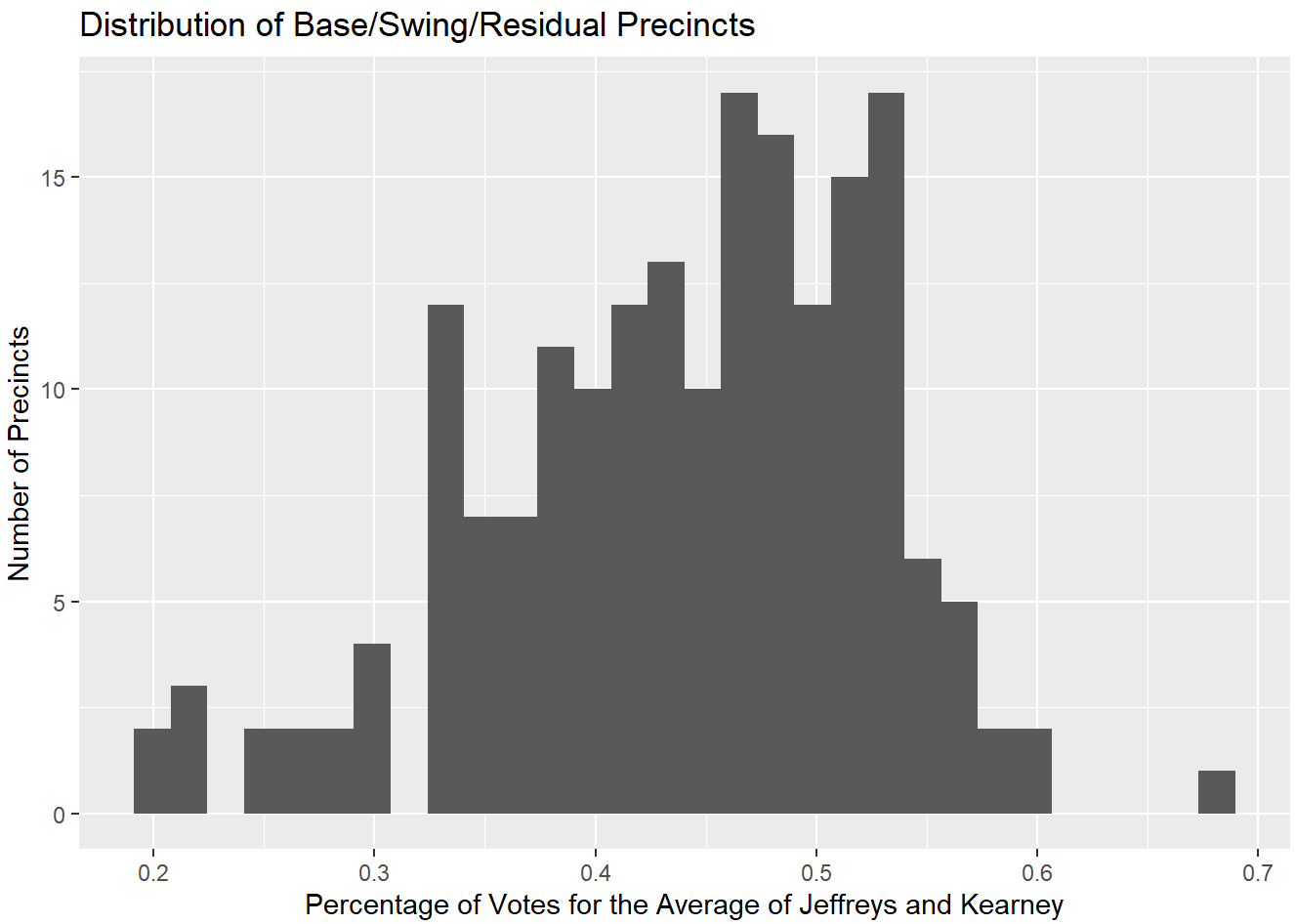


Distribution of the percentage of predicted votes among the precincts

```
combined5 %>%
  ggplot(aes(Jeff_Kearney))+
  geom_histogram()+
  labs(title = "Distribution of Base/Swing/Residual Precincts",
       x = "Percentage of Votes for the Average of Jeffreys and Kearney",
       y = "Number of Precincts")
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 374 rows containing non-finite values (`stat_bin()`).
```

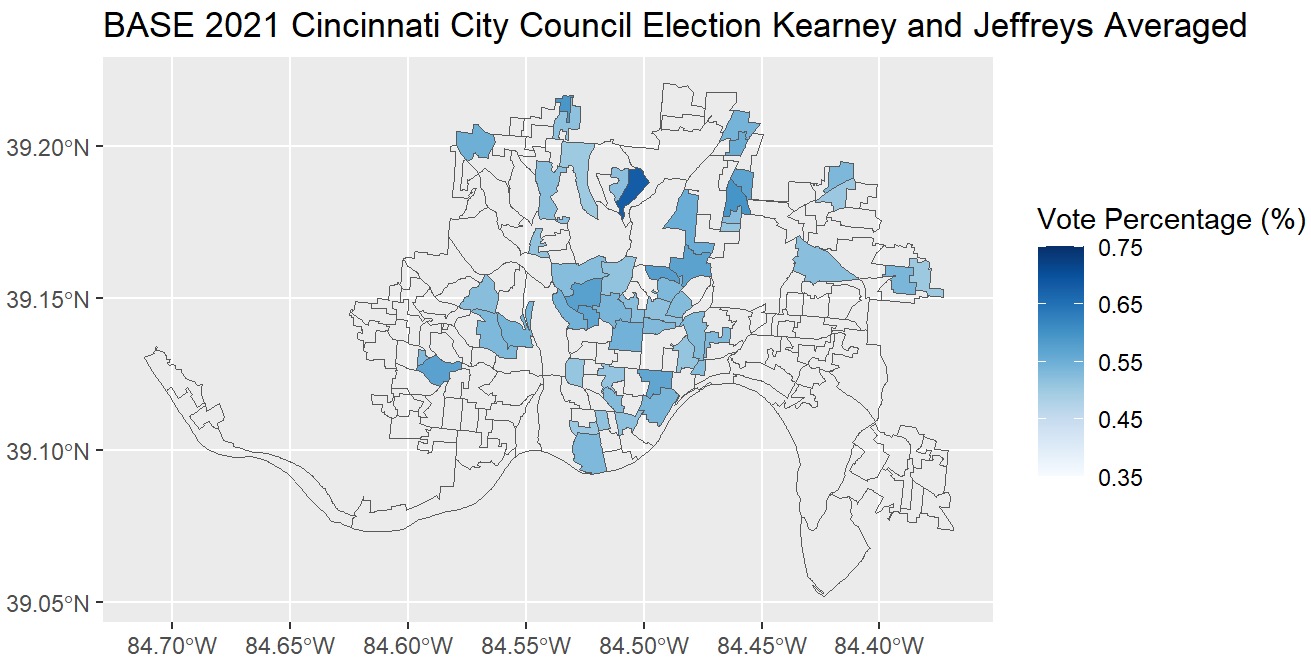


Identifying the top base and swing precincts

```
temp <- combined5 %>%
  filter(base_swing == "Base")

temp <- temp %>% arrange(desc(Jeff_Kearney))


temp2 <- combined5 %>%
  filter(base_swing == "Swing")

temp2 <- temp2 %>% arrange(desc(`REGISTERED VOTERS TOTAL`))
```

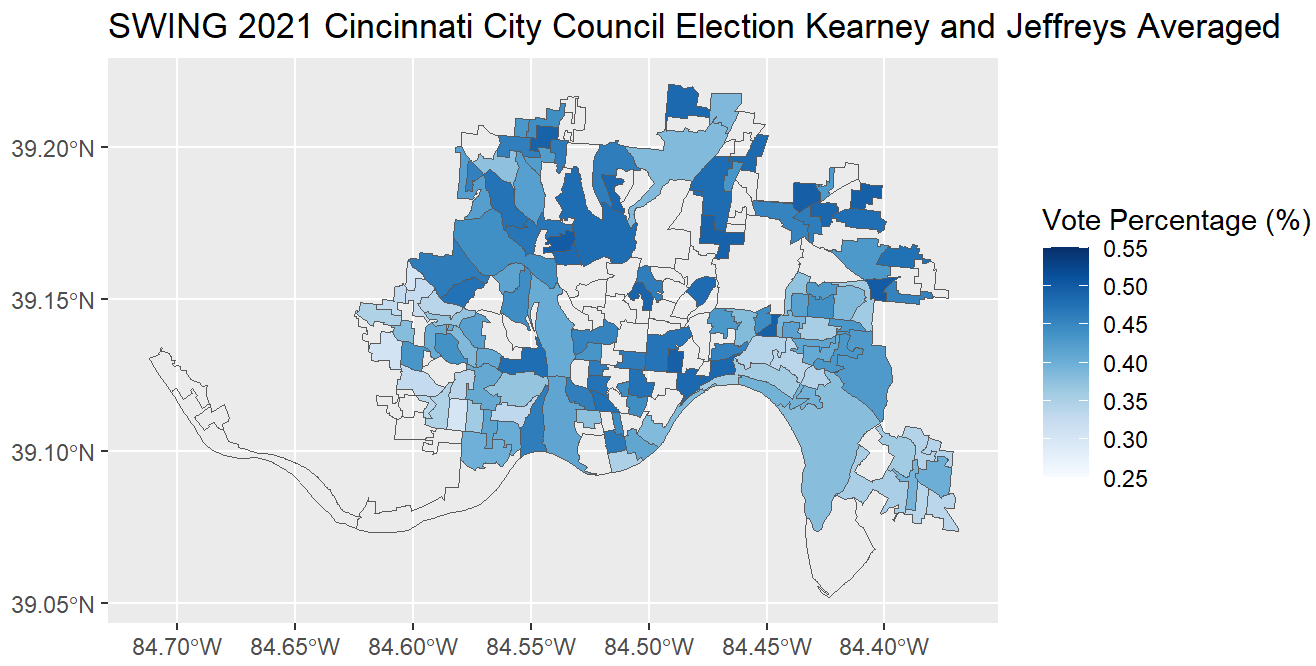
Map of the base precincts

```
combined5 %>%
  filter(grepl("CIN", NAME)) %>%
  mutate(Jeff_Kearney = ifelse(base_swing == "Base", Jeff_Kearney, NA)) %>%
  ggplot(aes(fill = Jeff_Kearney))+
  geom_sf()+
  labs(title = "BASE 2021 Cincinnati City Council Election Kearney and Jeffreys Averaged",
       fill = "Vote Percentage (%)",
       caption = "") +
  scale_fill_gradientn(colours = brewer.pal(n = 9, name = "Blues"),
                       limits = c(0.35, 0.75),
                       na.value = "transparent")
```



Map of the swing precincts

```
combined5 %>%
  filter(grepl("CIN", NAME)) %>%
  mutate(Jeff_Kearney = ifelse(base_swing == "Swing", Jeff_Kearney, NA)) %>%
  ggplot(aes(fill = Jeff_Kearney))+
  geom_sf()+
  labs(title = "SWING 2021 Cincinnati City Council Election Kearney and Jeffreys Averaged",
       fill = "Vote Percentage (%)",
       caption = "") +
  scale_fill_gradientn(colours = brewer.pal(n = 9, name = "Blues"),
                       limits = c(0.25, 0.55),
                       na.value = "transparent")
```


SWING 2021 Cincinnati City Council Election Kearney and Jeffreys Averaged
39.20°N
39.15°N
39.10°N
39.05°N
84.70°W
84.65°W
84.60°W
84.55°W
84.50°W
84.45°W
84.40°W
Vote Percentage (%)
0.55
0.50
0.45
0.40
0.35
0.30
0.25