

CT5100 Assignment 3

Conor O'Donoghue

Introduction

For the third assignment, we were tasked with visualizing the results of the Irish 2020 general election. Given election data for the 2016 and 2020 elections at the level of both the Galway West constituency and the National totals, we were asked to show:

- The vote per party in each election for Galway West
- The change in vote per party from 2016-2020 for Galway West
- A comparison of Galway West to the national average for party share of the vote for 2016 and 2020
- The change in vote for the most significant candidates in both elections in Galway West

Each of the four prompts will be addressed with a single plot, as well as an explanation of the choices made in methods of visualization. In addition, one of the plots will be accompanied by a CVD-friendly version.

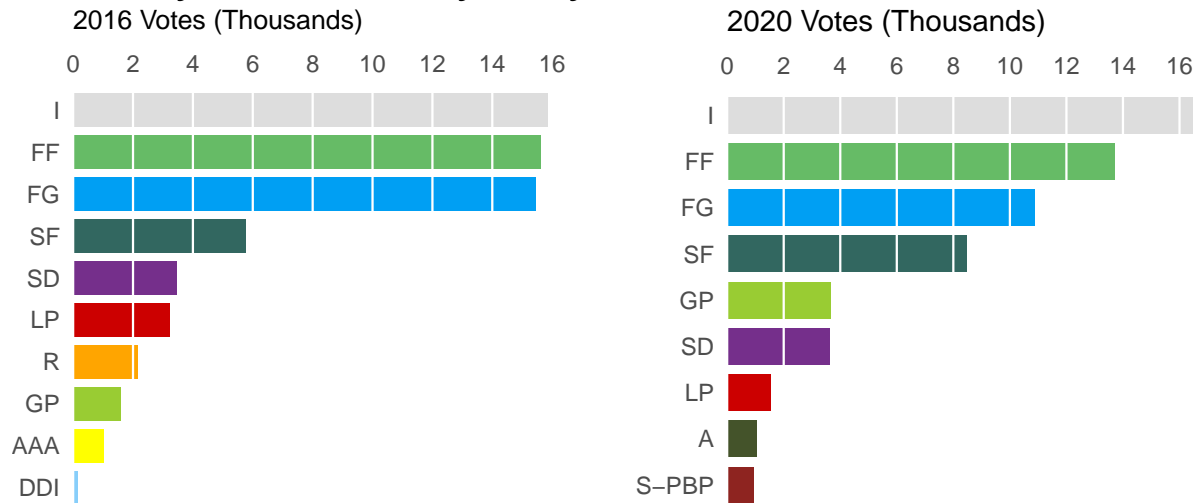
Results by Party

As future sections will represent this same data in terms of proportion of the total vote, and this section is merely concerned with vote per party in each election, I believe it better to emphasize the differences between parties vote counts rather than the percentage of the total vote. This is best done with a bar chart as opposed to a stacked bar or Pareto chart. As the electorate population of Galway West hasn't moved much in 2016 vs 2020 (roughly 100,000), I think visualizing the total votes rather than percentage of total votes is suitable.

To minimize ink and make it easier to compare values between parties, I utilize Tufte lines as the gridlines. Breaking the bars up into units of 2000 make general comparisons in proportion easier (eg: Sinn Féin has a value of three units while Labour has around one and a half).

Though the purpose of this visualization is to compare the differences between each party within each election as opposed to between the elections, I chose to arrange the plots side-by-side. Not only does this save space, but it does make it easier to compare the two elections in ways beyond the difference for each party (eg: Green Party and Labour roughly swapped total votes between the two elections, and some parties present in 2016 are not in 2020). This is also why I flipped the axes and made the bars horizontal: it makes comparison between the two plots much easier than if they were each vertical.

Galway West Votes by Party: 2016 vs 2020 General Election

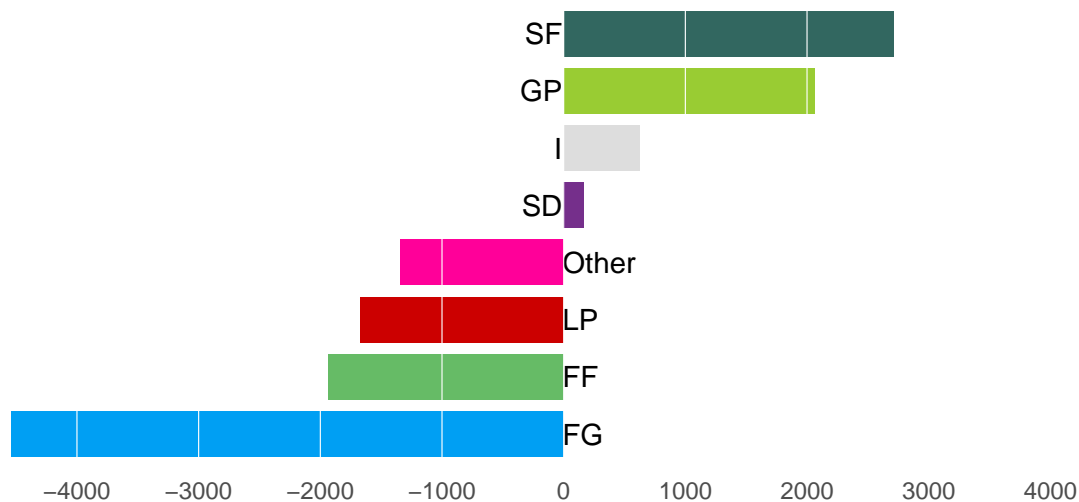


Change in Party Vote from 2016-2020 for Galway West

My interpretation of this prompt is to emphasize the difference between votes in 2016 and 2020 for each party. Rather than plot the votes for each party side-by-side on the same set of axes, I instead chose to plot the gain or loss of votes in 2020 compared to 2016 in a diverging discrete bar chart.

One of the issues to consider is how to approach parties that were present in one election and not the other – like Renua’s lack of presence in Galway West in 2020 and Aontú not existing before 2019. If the value to compare to is 0, the differences for even small parties will appear large enough to dwarf the longstanding parties. For this reason, I argue it makes the most sense to group parties unique to each election into an “Other” category, and visualize the difference in “Other” votes instead.

Galway West Elections: Vote Differential by Party 2020 vs 2016



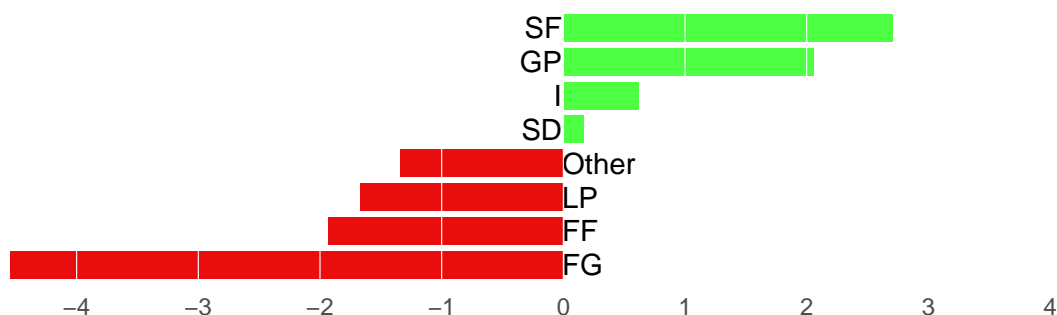
Galway West Party Vote Difference: Colorblind Alternative

Differentiating between party colors can be difficult for those with colorblindness, particularly for those with tritanomaly (which makes blue and green look the same, and makes Fianna Fáil and Fine Gael difficult to tell apart). For the previous bar charts, a simple and effective way to make the plots more accessible would be to remove the different party colors altogether, and only encode party with the axis labels. For a diverging bar chart, however, it may be more appropriate to have diverging colors as well – one color for positive values and another for negative values. If we opt for this instead of party colors to accommodate those with CVD, it would be essential to use a pair of colors that would be discernible for people with CVD.

The colors I used in the following graph were made using the tool at <https://davidmathlogic.com/colorblind/>

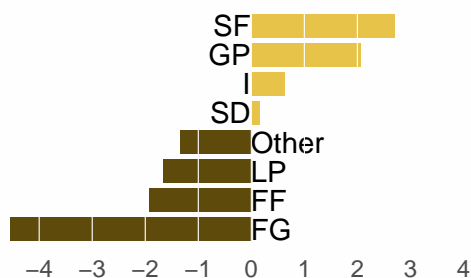
I wanted to maintain using green for positive values and red for negative values due to the traditional association, but added white to the green and black to the red to make them distinguishable for those with CVD. Below is the chart as it would appear to those with varying types of CVD using the colorblindnr package.

Vote Differential (thousands)



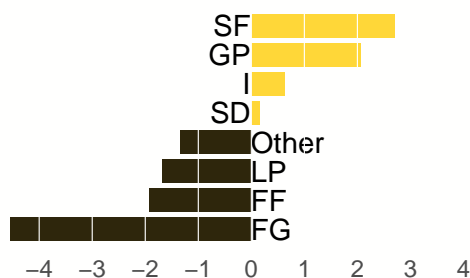
Deutanomaly

Vote Differential (thousands)



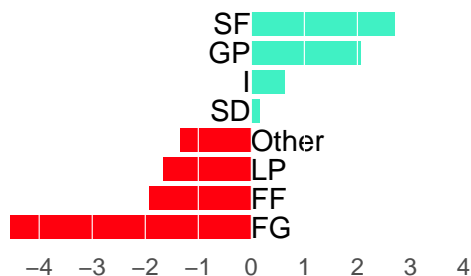
Protanomaly

Vote Differential (thousands)



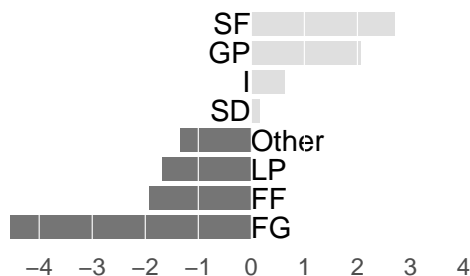
Tritanomaly

Vote Differential (thousands)



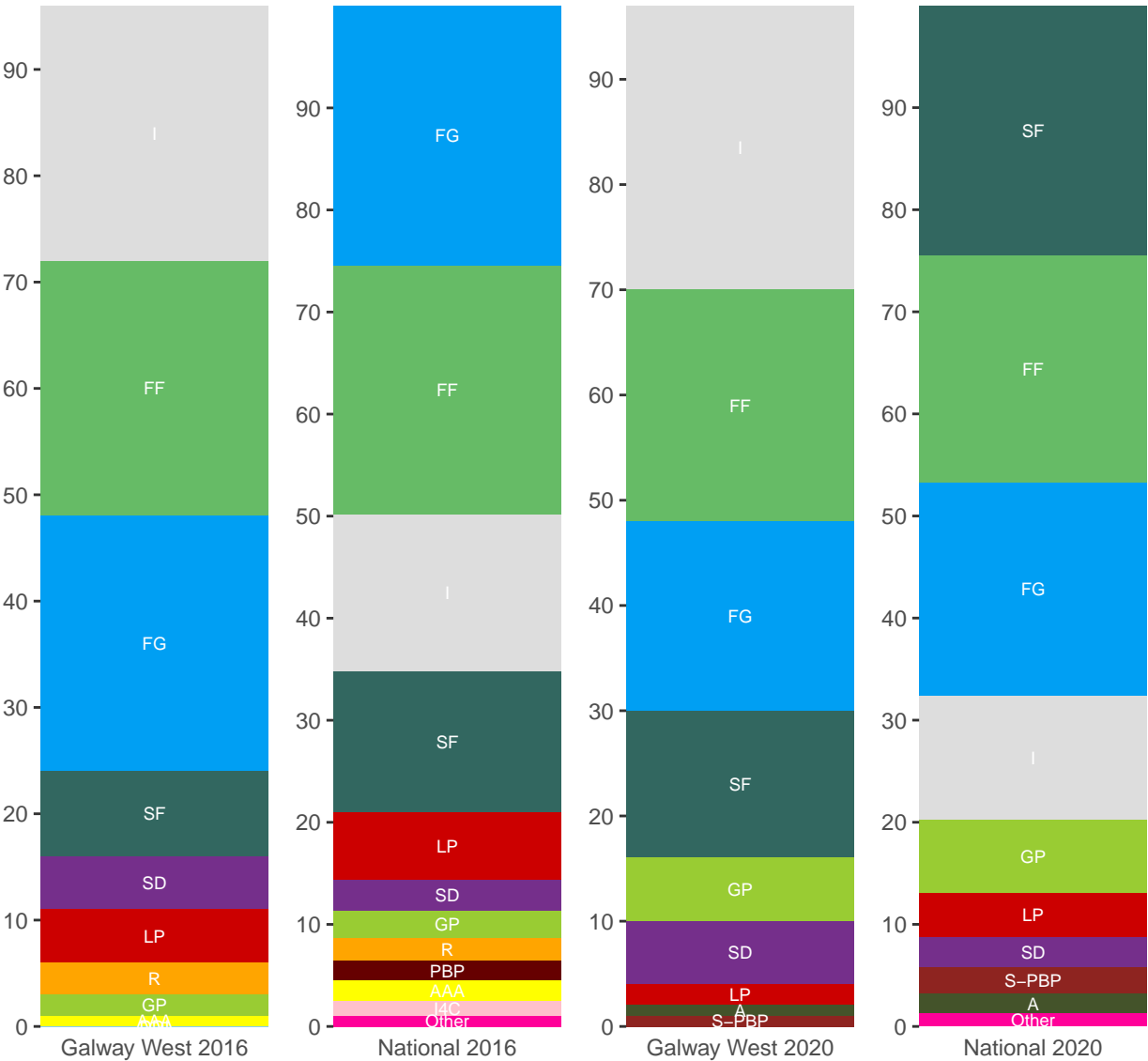
Desaturated

Vote Differential (thousands)



Comparison of Party Vote: Galway West vs National Average

When comparing elections at the constituency and national level, it is essential to represent each party in terms of the proportion of total vote to make them comparable. I chose to visualize each of the Galway West and National elections for each of 2016 and 2020 as separate stacked bar charts. While I could have visualized the differences in total proportion of the vote for each party between Galway West and National elections for each of 2016 and 2020 using a pair of diverging bar charts similar to the previous plot, that would provide less information. Plotting each of the elections as side-by-side stacked bar charts allows one to compare any election to any other, which I believe makes the following plot more useful.



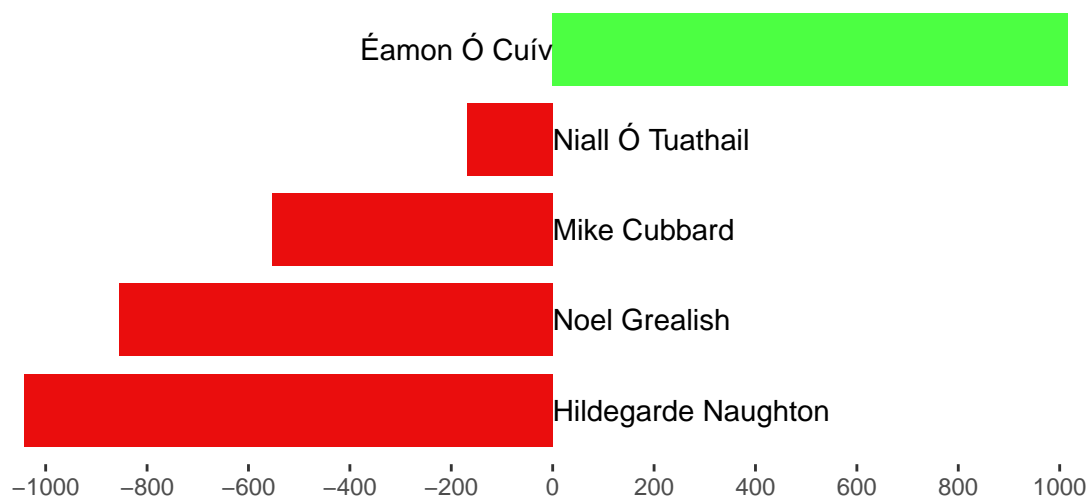
Change in vote for most significant candidates in Galway West

Each of the 5 Galway West candidates that ran in both 2016 and 2020 obtained votes in the thousands, which in my opinion makes each of the five worth including. If one had received a negligible amount of votes compared to the others I may have excluded him or her, but they are close enough to compare.

Since we are once again concerned with the change in votes from 2016 to 2020, I once again think it is most appropriate to use a diverging bar chart the same way as I had previously done with the parties. This time I do not believe coloring the bars by party to be appropriate, as two are Independent and different candidates should not have the same color. Instead, I will use the discrete positive-negative coloring from the CVD-friendly version of the previous diverging bar chart as the default.

While I would have preferred to utilize Tufte lines like in previous plots, the candidate names are long enough that the white gridlines also go through the names, and the difference in votes for each candidate is easy enough to read without them that I found it acceptable to remove the tufte lines altogether.

Galway West 2020 Repeat Candidates Vote Change from 2016



Appendix: Code

```
knitr::opts_chunk$set(echo = TRUE)

# Loading libraries
library(ggplot2)
library(dplyr)
library(forcats)
library(grid)
library(gridExtra)
library(colorblindr)

# Galway West 2020 Data Wrangling
# I removed the extra cells in GalwayWest-2020.xlsx, kept just the data, and saved as csv format.
# Each one of the 'Party' values has an extra space (' ') at the end. Removing so party.color will work
GW2020 <- read.csv('./GalwayWest-2020_simple.csv', stringsAsFactors=FALSE, strip.white = TRUE)
# Only need the party, candidate, and first preference vote count.
GW2020 <- GW2020[,c('Party', 'Candidate', 'Count.1')]
# Change Count.1 column name to match 2016 data.
colnames(GW2020)[which(names(GW2020) == "Count.1")] <- "Votes"

# Galway West 2016 Data Wrangling
GW2016 <- read.csv('./2016-04-28_general-election-count-details-galway-west-csv_en.csv', stringsAsFactors=FALSE)
# Select rows where Count.Number is 1
GW2016 <- GW2016[which(GW2016$Count.Number==1),]
# Combine first name and surname cols into name
GW2016$Candidate <- paste(GW2016$Candidate.First.Name, GW2016$Candidate.surname)
# Only include Candidate name, Party, and Votes
GW2016 <- GW2016[,c("Candidate", "Party", "Votes")]

# Various name changes for consistency between 2016 and 2020 dataframes:
GW2016$Party[GW2016$Party == 'Fianna Fail'] <- "Fianna Fáil"
GW2016$Party[GW2016$Party == 'Ind'] <- "Independent"
GW2016$Party[GW2016$Party == 'Social Democratic Party'] <- "Social Democrats"
GW2016$Party[GW2016$Party == 'Sinn Fein'] <- "Sinn Féin"
GW2016$Party[GW2016$Party == 'Direct Democracy Ireland '] <- "Direct Democracy Ireland"
GW2016$Party[GW2016$Party == 'Fine Gael'] <- "Fine Gael"
GW2020$Party[GW2020$Party == 'Solidarity-PBP'] <- "Solidarity-PBP"
# In the 2016 dataset, Ó is replaced with O'. To keep names consistent between 2016 and 2020,
# names that use Ó in the 2020 dataset will be edited in this one.
GW2016$Candidate[GW2016$Candidate == "Éamon O'Cuív"] <- "Éamon Ó Cuív"
GW2016$Candidate[GW2016$Candidate == "Niall O' Tuathail"] <- "Niall Ó Tuathail"

# Party colors and short names
# Party colors obtained from:
# https://en.wikipedia.org/wiki/Category:Ireland_political_party_colour_templates
# 'Other' party color chosen as a bright magenta, which is distinct from the other colors.
party.colors = c("Fianna Fáil" = "#66BB66", "Fine Gael" = "#009FF3",
                 "Sinn Féin" = "#326760", "Independent" = "#DDDDDD",
                 "Social Democrats" = "#752F8B", "Green Party" = "#99CC33",
                 "Labour Party" = "#CC0000", "Aontú" = "#44532A",
                 "Solidarity-PBP" = "#8E2420", "AAA" = "#FFFF00",
                 "Direct Democracy Ireland" = "#87CEFA", "Renua" = "#FFA500",
```

```

      "Other" = "#FF0099", "People before Profit" = "#660000",
      "Independents 4 Change" = "#FFC0CB")
party.short = c("Fianna Fáil" = "FF", "Fine Gael" = "FG",
  "Sinn Féin" = "SF", "Independent" = "I",
  "Social Democrats" = "SD", "Green Party" = "GP",
  "Labour Party" = "LP", "Aontú" = "A",
  "Solidarity-PBP" = "S-PBP", "AAA" = "AAA",
  "Direct Democracy Ireland" = "DDI", "Renua" = "R",
  "Other" = "Other", "People before Profit" = "PBP",
  "Independents 4 Change" = "I4C")

# Obtaining party totals for 2016 and 2020 datasets
GW2016_Party <- GW2016 %>% group_by(Party) %>%
  summarise(Num_Candidates = n(),
            Total_Votes = sum(Votes)) %>%
  arrange(desc(Total_Votes))
GW2020_Party <- GW2020 %>% group_by(Party) %>%
  summarise(Num_Candidates = n(),
            Total_Votes = sum(Votes)) %>%
  arrange(desc(Total_Votes))

# Adding Party_short column to dataframes
GW2016_Party$Party_short <- party.short[GW2016_Party$Party]
GW2020_Party$Party_short <- party.short[GW2020_Party$Party]

# 2016 Galway West Party results subplot
theme_set(theme_classic())
plot2016 <- ggplot(GW2016_Party, (aes(x=reorder(Party,Total_Votes), y=Total_Votes, fill=Party))) +
  geom_col(width=0.8) +
  scale_fill_manual(values=party.colors) +
  scale_y_continuous(sec.axis = dup_axis(), limits = c(0, 18000),
    expand = c(0, 0),
    breaks = seq(0,16000,2000),
    # Labels in units of thousands to save space
    labels = as.character(seq(0,16,2)),
    name = "Votes") +
  scale_x_discrete(labels=GW2016_Party$Party_short) +
  ggtitle("2016 Votes (Thousands)") +
  coord_flip(clip = "off") +
  theme(
    axis.title = element_blank(),
    axis.line.y = element_blank(),
    axis.ticks.y = element_blank(),
    axis.line.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.title.y = element_blank(),
    axis.title.x.bottom = element_blank(),
    axis.text.x.bottom = element_blank(),
    legend.position = "none",
    plot.title = element_text(size = 10),
    plot.margin = margin(3, 6, 3, 3),
    panel.background = element_blank(),
    panel.grid.major.x = element_line(size = 0.4, linetype = 'solid', colour = "white"),

```

```

    panel.ontop = TRUE
  )

# 2020 Galway West Party results subplot
plot2020 <- ggplot(GW2020_Party, (aes(x=reorder(Party,Total_Votes), y=Total_Votes, fill=Party))) +
  geom_col(width=0.8) +
  scale_fill_manual(values=party.colors) +
  scale_y_continuous(sec.axis = dup_axis(), limits = c(0, 18000),
                     expand = c(0, 0),
                     breaks = seq(0,16000,2000),
                     labels = as.character(seq(0,16,2)),
                     name = "Votes") +
  scale_x_discrete(labels=GW2020_Party$Party_short) +
  ggtitle("2020 Votes (Thousands)") +
  coord_flip(clip = "off") +
  theme(
    axis.title = element_blank(),
    axis.line.y = element_blank(),
    axis.ticks.y = element_blank(),
    axis.line.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.title.y = element_blank(),
    axis.title.x.bottom = element_blank(),
    axis.text.x.bottom = element_blank(),
    legend.position = "none",
    plot.title = element_text(size = 11),
    plot.margin = margin(3, 6, 3, 3),
    panel.background = element_blank(),
    panel.grid.major.x = element_line(size = 0.4, linetype = 'solid', colour = "white"),
    panel.ontop = TRUE
  )

# Arrange GW Party subplots into grid
grid.arrange(plot2016, plot2020, ncol=2,
             top=textGrob("Galway West Votes by Party: 2016 vs 2020 General Election",
                          gp=gpar(fontsize=16,font=3)))

# GW Party Vote Difference Data Wrangling
# Find parties in both datasets
common_parties <- GW2020_Party$Party[GW2020_Party$Party %in% GW2016_Party$Party]
# Change all parties not in both datasets to "Other", as well as short name
GW2016_Party_0 <- GW2016_Party
GW2016_Party_0$Party[!GW2016_Party_0$Party %in% common_parties] <- "Other"
GW2016_Party_0[GW2016_Party_0$Party == "Other","Party_short"] <- "Other"
GW2020_Party_0 <- GW2020_Party
GW2020_Party_0$Party[!GW2020_Party_0$Party %in% common_parties] <- "Other"
GW2020_Party_0[GW2020_Party_0$Party == "Other","Party_short"] <- "Other"
# Combine rows with "Other" using dplyr summarize->groupby->sum
GW2016_Party_0 <- GW2016_Party_0 %>% group_by(Party, Party_short) %>%
  summarise(Total_Votes = sum(Total_Votes)) %>%
  arrange(desc(Total_Votes))
GW2020_Party_0 <- GW2020_Party_0 %>% group_by(Party, Party_short) %>%
  summarise(Total_Votes = sum(Total_Votes)) %>%

```



```

                                arrange(desc(Total_Votes))
# Finally, combine the two into a single df that has vote differences
GWdiff <- inner_join(GW2020_Party_0, GW2016_Party_0, by="Party")
GWdiff$Diff <- GWdiff$Total_Votes.x - GWdiff$Total_Votes.y
GWdiff <- GWdiff[, c("Party", "Party_short.x", "Diff")]
colnames(GWdiff) <- c("Party", "Short", "Diff")

# Plotting GW party vote differentials from 2016 to 2020
# Will put party names at base of each bar,
# and which side to put name on depends on whether the bar is negative or positive.
# So here, hjust values are set based on whether Difference is negative or positive.
GWdiff$pos <- ifelse(GWdiff$Diff < 0, 0, 1)
theme_set(theme_classic())
ggplot(GWdiff, (aes(x=reorder(Party, Diff), y=Diff, fill=Party))) +
  geom_col(position="identity", width=0.8) +
  scale_fill_manual(values=party.colors) +
  scale_y_continuous(limits = c(-5000, 5000),
                     expand = c(0, 0),
                     breaks = seq(-4000, 4000, 1000),
                     labels = as.character(seq(-4000, 4000, 1000))) +
  ggtitle("Galway West Elections: Vote Differential by Party 2020 vs 2016") +
  coord_flip() +
  geom_text(aes(x=Party, y=0, label=Short), hjust=GWdiff$pos, nudge_y=-10) +
  theme(
    axis.line.y = element_blank(),
    axis.ticks.y = element_blank(),
    axis.text.y = element_blank(),
    axis.title.y = element_blank(),
    axis.line.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.title.x = element_blank(),
    plot.margin = margin(6, 6, 3, 3),
    panel.background=element_blank(),
    panel.grid.major.x = element_line(size = 0.2, linetype = 'solid', colour = "white"),
    panel.ontop = TRUE,
    legend.position = "none"
  )

# CVD-friendly GW Party Vote Differential plot
cvd_colors <- c("#EA0D0D", "#4CFF42")
theme_set(theme_classic())
cvd_g <- ggplot(GWdiff, (aes(x=reorder(Party, Diff), y=Diff, fill=as.factor(pos)))) +
  geom_col(position="identity", width=0.8) +
  scale_fill_manual(values=cvd_colors) +
  scale_y_continuous(limits = c(-5000, 5000),
                     expand = c(0, 0),
                     breaks = seq(-4000, 4000, 1000),
                     labels = as.character(seq(-4, 4, 1))) +
  ggtitle("Vote Differential (thousands)") +
  coord_flip() +
  geom_text(aes(x=Party, y=0, label=Short), hjust=GWdiff$pos, nudge_y=-10) +
  theme(
    axis.line.y = element_blank(),

```

```

axis.ticks.y = element_blank(),
axis.text.y = element_blank(),
axis.title.y = element_blank(),
axis.line.x = element_blank(),
axis.ticks.x = element_blank(),
axis.title.x = element_blank(),
plot.margin = margin(6, 6, 3, 3),
panel.background=element_blank(),
panel.grid.major.x = element_line(size = 0.2, linetype = 'solid',colour = "white"),
panel.ontop = TRUE,
legend.position = "none"
)
cvd_g
cvd_grid(cvd_g)

# National Election Data Wrangling
# Data loading
Nat2016 <- read.csv('./National_2016.csv', stringsAsFactors = FALSE,
strip.white = TRUE)
Nat2020 <- read.csv('./National_2020.csv', stringsAsFactors = FALSE,
strip.white = TRUE)
# First, remove redundant rows.
# 2016: remove AAA-PBP row. This combines AAA and PBP rows, which are separate in the GW
Nat2016 <- Nat2016[Nat2016$Party != "AAA-PBP",]
# In 2020, remove individual PBP, Solidarity, and RISE rows.
# Total is in "Solidarity-PBP", which is what's present in GW 2020 data.
Nat2020 <- Nat2020[!Nat2020$Party %in% c("People Before Profit","Solidarity","RISE"),]
# Want to combine any party that got less than 10000 votes (none of which present in GW data) into "Other"
# Best way is to make a new column with unique values,
# Reassign all columns I want to combine with the same value,
# use dplyr's groupby and summarize functions to sum up votes within the group,
# and then change the values back to party names by left joining and removing extraneous columns
Nat2016$Combine <- seq(1:length(rownames(Nat2016)))
Nat2016_Uniq <- Nat2016 # keep a copy with all unique values for joining later
Nat2016[Nat2016$Votes < 10000,"Combine"] <- 50
Nat2020$Combine <- seq(1:length(rownames(Nat2020)))
Nat2020_Uniq <- Nat2020
Nat2020[Nat2020$Votes < 10000,"Combine"] <- 60
# "Independents 4 Change is a party, not independents.
# But "Independent Alliance" is actually made up of independents.
# So for 2016, want to combine "Independent Alliance" and "Other Independents" rows.
# Hence, also change their combine value to be the same.
Nat2016[Nat2016$Party %in% c("Independent Alliance", "Other Independents"),"Combine"] <- 70

Nat2016_Party <- Nat2016 %>% group_by(Combine) %>%
summarise(Votes = sum(Votes),
Vote_Pct = sum(Vote_Pct)) %>%
arrange(desc(Votes))

Nat2016_Party <- left_join(x=Nat2016_Party, y=Nat2016_Uniq, by="Combine")
# Change Party names for groups to 'Independent' and 'Other'
Nat2016_Party[Nat2016_Party$Combine==50,"Party"] <- "Other"

```

```

Nat2016_Party[Nat2016_Party$Combine==70,"Party"] <- "Independent"
# Select only relevant columns and rename
Nat2016_Party <- Nat2016_Party[,c("Party", "Votes.x", "Vote_Pct.x")]
colnames(Nat2016_Party) <- c("Party", "Votes", "Vote_Pct")

Nat2020_Party <- Nat2020 %>% group_by(Combine) %>%
  summarise(Votes = sum(Votes),
            Vote_Pct = sum(Vote_Pct)) %>%
  arrange(desc(Votes))

Nat2020_Party <- left_join(x=Nat2020_Party, y=Nat2020_Uniq, by="Combine")
# Change Party names for groups to 'Independent' and 'Other'
Nat2020_Party[Nat2020_Party$Combine==60,"Party"] <- "Other"
# Select only relevant columns and rename
Nat2020_Party <- Nat2020_Party[,c("Party", "Votes.x", "Vote_Pct.x")]
colnames(Nat2020_Party) <- c("Party", "Votes", "Vote_Pct")
# Lastly, add short party names
Nat2016_Party$Party_short <- party.short[Nat2016_Party$Party]
Nat2020_Party$Party_short <- party.short[Nat2020_Party$Party]

# GW data wrangling
# Need percentage of the total vote, which we haven't calculated yet for GW datasets.
GW2016_Party$Vote_Pct <- as.integer(100*GW2016_Party$Total_Votes / sum(GW2016_Party$Total_Votes))
GW2020_Party$Vote_Pct <- as.integer(100*GW2020_Party$Total_Votes / sum(GW2020_Party$Total_Votes))

# Lastly, need to turn "Party" into a factor and reorder by vote
# Otherwise, placements in the stacked bar won't be ordered by size.
GW2016_Party_F <-GW2016_Party%>%mutate(Party = fct_reorder(Party, -Total_Votes))
GW2020_Party_F <-GW2020_Party%>%mutate(Party = fct_reorder(Party, -Total_Votes))
Nat2016_Party_F <-Nat2016_Party%>%mutate(Party = fct_reorder(Party, -Votes))
Nat2020_Party_F <-Nat2020_Party%>%mutate(Party = fct_reorder(Party, -Votes))

# Plotting Galway West vs National data (4 subplots)
# Galway West 2016 subplot
gw16_g <- ggplot(GW2016_Party_F, aes(x = "Galway West 2016", y = Vote_Pct, fill = Party))+
  geom_col(width = 0.4) +
  geom_text(aes(label = Party_short),
            position = position_stack(vjust = 0.5), size=2.6, colour = "white") +
  scale_y_continuous(breaks = seq(0,90, by = 10), expand=c(0,0))+
  scale_x_discrete( expand = c(0, 0))+
  scale_fill_manual(values = party.colors)+
  theme(legend.position = "None",
        legend.title = element_blank()) +
  theme(axis.title.y = element_blank(),
        axis.title.x = element_blank(),
        axis.line.x = element_blank(),
        axis.line.y = element_blank(),
        axis.ticks.x = element_blank())

# Galway West 2020 subplot
gw20_g <- ggplot(GW2020_Party_F, aes(x = "Galway West 2020", y = Vote_Pct, fill = Party))+
  geom_col(width = 0.4) +
  geom_text(aes(label = Party_short),

```

```

        position = position_stack(vjust = 0.5), size=2.6, colour = "white") +
scale_y_continuous(breaks = seq(0,90, by = 10), expand=c(0,0))+
scale_x_discrete( expand = c(0, 0))+
scale_fill_manual(values = party.colors)+
theme(legend.position = "None",
      legend.title = element_blank()) +
theme(axis.title.y = element_blank(),
      axis.title.x = element_blank(),
      axis.line.x = element_blank(),
      axis.line.y = element_blank(),
      axis.ticks.x = element_blank())

# National 2016 subplot
n16_g <- ggplot(Nat2016_Party_F, aes(x = "National 2016", y = Vote_Pct, fill = Party))+
  geom_col(width = 0.4) +
  geom_text(aes(label = Party_short),
            position = position_stack(vjust = 0.5), size=2.6, colour = "white") +
  scale_y_continuous(breaks = seq(0,90, by = 10), expand=c(0,0))+
  scale_x_discrete( expand = c(0, 0))+
  scale_fill_manual(values = party.colors)+
  theme(legend.position = "None",
        legend.title = element_blank()) +
  theme(axis.title.y = element_blank(),
        axis.title.x = element_blank(),
        axis.line.x = element_blank(),
        axis.line.y = element_blank(),
        axis.ticks.x = element_blank())

# National 2020 subplot
n20_g <- ggplot(Nat2020_Party_F, aes(x = "National 2020", y = Vote_Pct, fill = Party))+
  geom_col(width = 0.4) +
  geom_text(aes(label = Party_short),
            position = position_stack(vjust = 0.5), size=2.6, colour = "white") +
  scale_y_continuous(breaks = seq(0,90, by = 10), expand=c(0,0))+
  scale_x_discrete( expand = c(0, 0))+
  scale_fill_manual(values = party.colors)+
  theme(legend.position = "None",
        legend.title = element_blank()) +
  theme(axis.title.y = element_blank(),
        axis.title.x = element_blank(),
        axis.line.x = element_blank(),
        axis.line.y = element_blank(),
        axis.ticks.x = element_blank())
grid.arrange(gw16_g, n16_g, gw20_g, n20_g, ncol=4)

# Galway West candidate vote change data wrangling
# First, inner join 2016 and 2020 dataframes by candidate
GWdiff_C <- inner_join(GW2016, GW2020, by="Candidate")
# Next, obtain vote differential using dplyr groupby->summarize
GWdiff_C <- GWdiff_C %>% group_by(Candidate) %>%
  summarize(Diff = Votes.x - Votes.y)
# Lastly, add flag for positive or negative
GWdiff_C$pos <- ifelse(GWdiff_C$Diff < 0, 0, 1)

```

```

# Plotting GW candidate vote differential
theme_set(theme_classic())
ggplot(GWdiff_C, (aes(x=reorder(Candidate,Diff), y=Diff, fill=as.factor(pos)))) +
  geom_col(position="identity",width=0.8) +
  scale_fill_manual(values=cvd_colors) +
  scale_y_continuous(limits = c(-1200, 1200),
                     expand = c(0, 0),
                     breaks = seq(-1000,1000,200),
                     labels = as.character(seq(-1000,1000,200))) +

  ggtitle("Galway West 2020 Repeat Candidates Vote Change from 2016") +
  coord_flip() +
  theme(
    axis.line.y = element_blank(),
    axis.ticks.y = element_blank(),
    axis.text.y = element_blank(),
    axis.title.y = element_blank(),
    axis.line.x = element_blank(),
    #axis.ticks.x = element_blank(),
    axis.title.x = element_blank(),
    plot.margin = margin(6, 6, 3, 3),
    panel.background=element_blank(),
    panel.grid.major.x = element_blank(),
    #panel.grid.major.x = element_line(size = 0.2, linetype = 'solid',colour = "white"),
    panel.ontop = TRUE,
    legend.position = "none"
  ) +
  geom_text(aes(x=Candidate, y=0, label=Candidate), hjust=GWdiff_C$pos, nudge_y=0)

```