

02/03/2020

## Introduction

For this data visualisation assignment we are required to analyse the 2016 and 2020 Galway West constituency Irish Election Results and create visualisations that will compare the vote per party for each election, compare results with the national results and also compare some of the significant candidates in these elections.

## Data Cleaning

In order to prepare the data for visualisation, had to carry out the following steps:

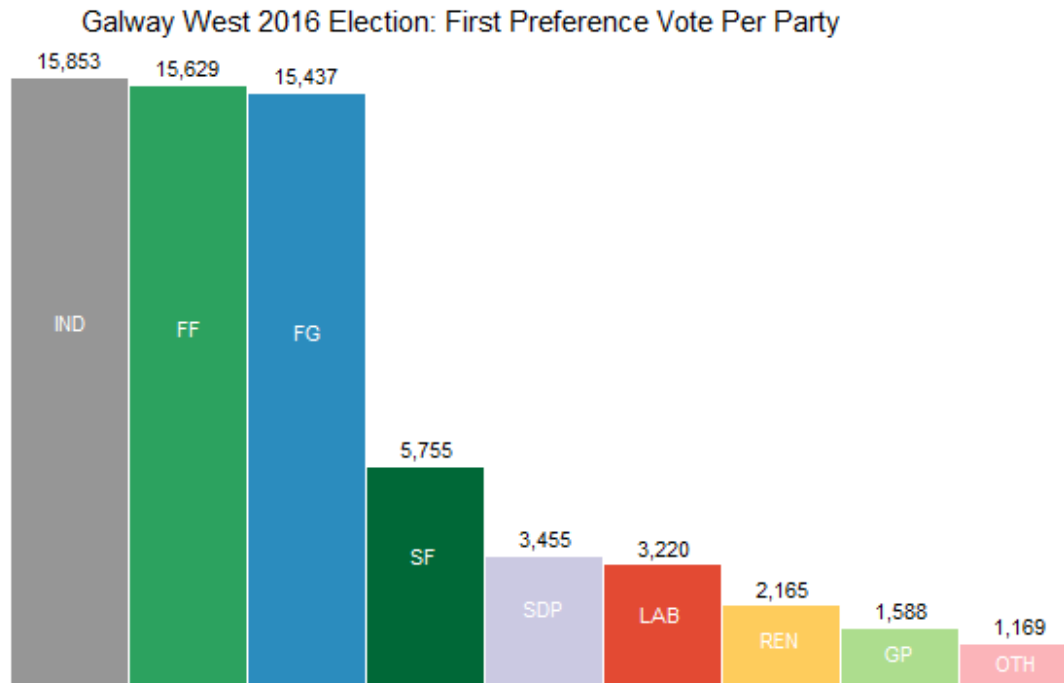
- merged the candidate first name and surname columns into one candidate field (*2016 dataset only*).
- filtered only the first preference votes (count number = 1)
- any candidate that had votes less than 1500, set party to Other and summed their votes (*2016 dataset only*)
- aggregated data per party, creating new column to hold percentage share of vote per party
- created column for abbreviated version of party name. This is required for displaying party name on visualisations. There are many variations of the party names across the different data sources. This abbreviated version of party name makes joining of datasets easier for visualisation.
- created some formatting columns required for improving the look of the visualisations
- some candidates names had to be renamed in order to match between datasets.
- in order to get national vote results per party per election, created a csv file using data from the following website:  
[https://www.tcd.ie/Political\\_Science/people/michael\\_gallagher/Election2020.php](https://www.tcd.ie/Political_Science/people/michael_gallagher/Election2020.php)
- independents were split into different parties in this dataset so merged these into one party called 'Independent', summing their votes.
- merged all the party data into one dataframe and candidate data into another dataframe in order to do visualisations across parties/elections/candidates.

## Vote Per Party in Each Election

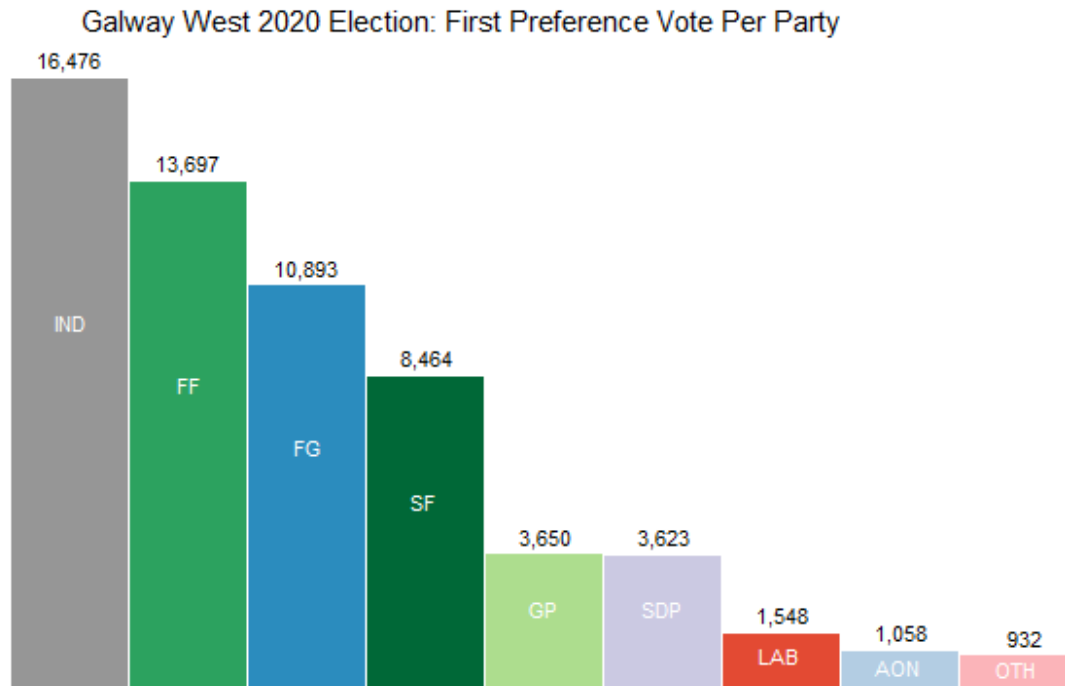
To represent the comparison of number of votes per party in the elections, have used two bar plots. Both plots have the following characteristics:

- number of votes is visualised using vertical bars of different lengths. Have displayed the amount of votes on the top of each bar to give more precision to the user.
- as the exact number of votes is displayed on top of the bars, there is no need for a y-axis.
- ordered the parties in the bar plot, going from highest number of votes to lowest.
- have used colour to represent the different parties, a different colour per party.

- have used colours from the colorbrewer palette <http://colorbrewer2.org>. Have selected colour friendly colours but have also made sure to stay as close as possible to the colours that are usually associated with the particular party.
- The abbreviated version of the party name is displayed in the middle of the bar. This eliminates the need for a legend and x-axis.
- overall these bar plots have a high data to ink ratio. To achieve this have used white grid-lines, removed ticks and values on both axis. This results in minimalist visualisations which still provides the user with the necessary insight into the results of the elections.



*Galway West 2016 Election: First Preference Vote Per Party*



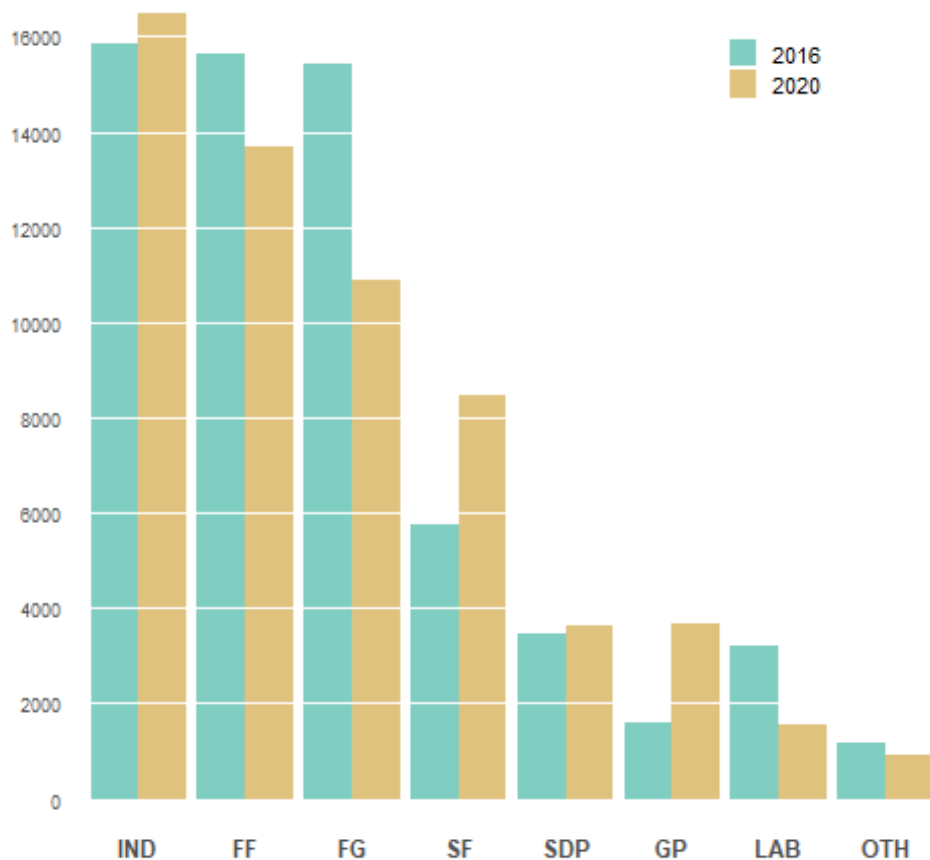
*Galway West 2020 Election: First Preference Vote Per Party*

#### Change in vote per party from 2016-2020

To show the change in number of votes per party in both elections, used a stacked bar chart. Using this, can visualise how the number of votes vary in each election. This plot has the following characteristics:

- number of votes is visualised using vertical bars of different lengths.
- ordered the parties in the bar plot, going from highest number of votes to lowest.
- use colour here to distinguish the different election years.
- have also included a legend on the top right of the plot to specify which colour represents which year.
- have used 'Tufte' lines to help guide the readers eye from the top of each bar to the corresponding horizontal tick on the y-axis
- have used the abbreviated version of the party name on the x-axis. This makes the visualisation more readable as many of the party names are very long.
- have used colours from the colorbrewer palette <http://colorbrewer2.org>. Have selected colour friendly colour from this palette.
- this visualisation has a high data to ink ratio. Have used white grid-lines, 'Tufte' lines and white background.

Change in vote per party between 2016 and 2020 elections



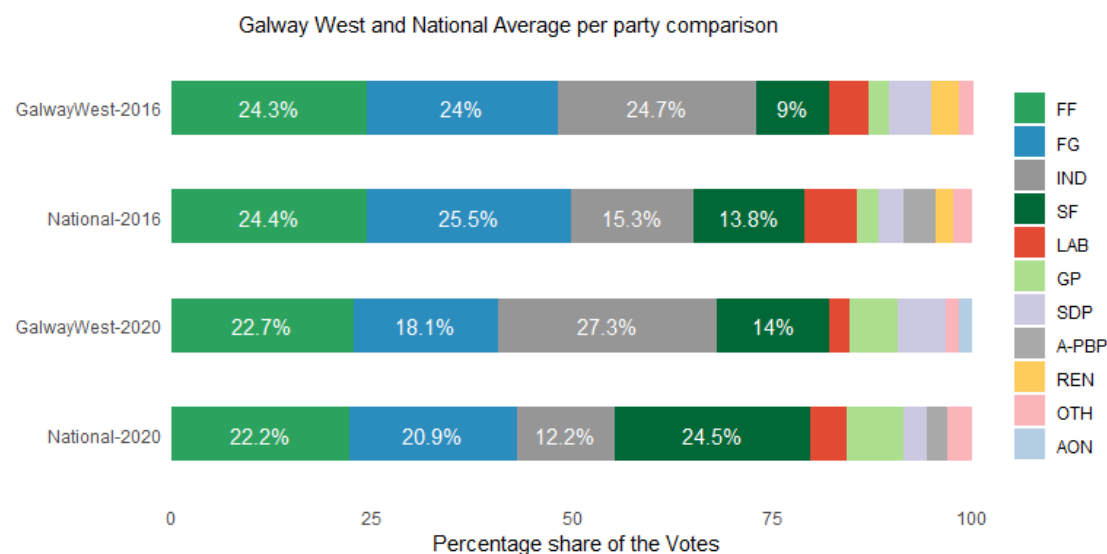
Change in vote per party between 2016 and 2020 elections

### National average per party comparison

To compare the Galway West constituency number of votes per party in both elections, with the overall national results for the same parties, used a horizontal stacked bar chart. This plot can give us a good visual sense of proportions of each bar within the whole bar. As each of the results are displayed under each other can easily compare the different proportions. The parties are displayed in the same order in each bar so that comparisons can easily be made. This plot has the following characteristics:

- parties are ordered with their number of votes in descending order.
- have displayed the Election (2016 or 2020) and Area (Galway West or National) on the left of each bar.
- displayed the percentage proportion of votes per party within the bar. Do not show the percentage on the small parties as the percentages would not fit in the small area and they were added outside the bar char it would clutter the visualisation.
- use colour here to distinguish the different parties and have used the same colours for each bar.

- have also included a legend to the right of the plot to specify which colour represents which party.
- have used the abbreviated version of the party name on the legend. This makes the visualisation more readable as many of the party names are very long.
- have used colours from the colorbrewer palette <http://colorbrewer2.org>. Have selected colour friendly colour for this plot. Have used the same colours to represent party colours as on previous visualisations in this report.
- this visualisation has a high data to ink ratio. Have used white grid-lines, no x-axis or y-axis ticks.



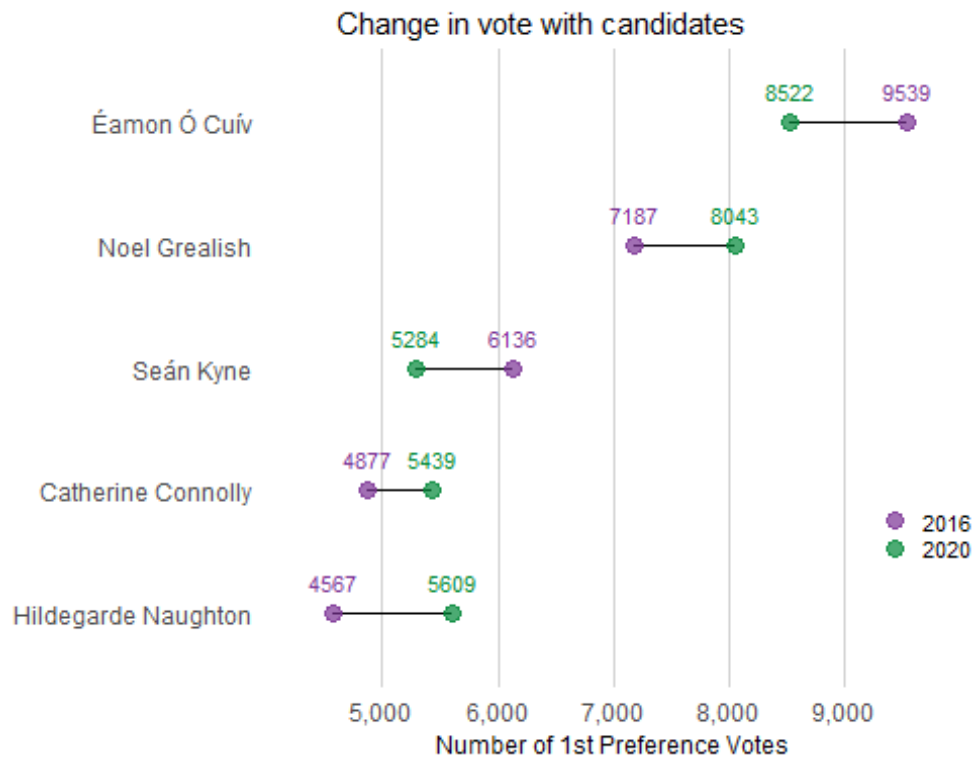
### *Galway West and National results for 2016 and 2020 Elections Compared*

#### **Candidate Change in Vote between 2016 and 2020.**

To compare the performance of significant candidates in the Galway West constituency between the 2016 and 2020 elections, used the dot plot.

Selected the candidates: Eamon O Cuiv, Noel Grealish, Sean Kyne, Catherine Connolly and Hildegard Naughton, as these were candidates that were common between the two elections and where the ones with the highest vote and also the most significant change in vote. This plot has the following characteristics:

- added the candidates with the most votes in descending order. i.e highest at the top.
- use colour to represent the election years 2016 and 2020. Added legend in order to distinguish which colour represents which year.
- have used colours from the colorbrewer palette <http://colorbrewer2.org>. Have selected colour friendly colour for this plot.
- display the number of votes above each dot.
- this visualisation has a high data to ink ratio. Have used white grid-lines, no x-axis or y-axis ticks and white background.



*Galway West General Election: Change in vote per party between 2016 and 2020*

## Conclusion

For all visualisations in this report, have use similar design decision throughout. i.e. consistent colours to represent the political parties, colours that are friendly for readers with CVD and also high data to ink ratio. These design decisions provide clarity and simplicity while still giving the reader some valuable insights into the results of the two recent elections in Irish history.

## Code Appendix

```
library(ggplot2)
library(viridis)
library(colorblindr)
library(dplyr)
library(utils)
library(tidyr)
library(readxl)
library(stringr)
library(forcats)

### Data Preperation Step

# Load the Galway West General Election data for 2016 from csv file
gw16 <- read.csv("../data/2016-04-28_general-election-count-details-galway-
west-csv_en.csv")
```

```

# Load the Galway West General Election data for 2016 from csv file (skip
first line)
gw20 <- read_excel("../data/GalwayWest-2020.xlsx", skip=1)

# extrac data per candidate
gw16PerCandidate <- gw16 %>%
  # merge candidate first and last names
  unite(Candidate, c(Candidate.First.Name, Candidate.surname), sep = " ",
remove = FALSE)%>%
  group_by(Candidate) %>%
  filter(Count.Number == 1) %>% # filter 1st preference votes
  select(Candidate, Votes) %>%
  arrange(desc(Votes))

# Extract and sort date per party (2016) - just going to look at first
preference votes (i.e. Count = 1)
gw16PerParty <- gw16 %>%
  group_by(Party) %>%
  filter(Count.Number == 1) %>%
  summarise(TotalVotes = sum(Votes)) %>%
  #filter(TotalVotes >= 1000) %>%
  arrange(desc(TotalVotes))

# create subset of dataframe for parties that have less than 1500 votes
other<-gw16PerParty %>%
  group_by(Party) %>%
  filter(TotalVotes<1500) %>%
  select(Party, TotalVotes) %>%
  summarise(sum(TotalVotes))

# Set these parties to have party name = Other
other$Party <- "Other"
# Sum the values of these other parties merging rows together
other <- other %>% rename("TotalVotes" = "sum(TotalVotes)") %>%
  group_by(Party) %>%
  select(Party, TotalVotes) %>%
  summarise(TotalVotes = sum(TotalVotes))

# Add back other to dataframe
gw16PerParty <- gw16PerParty %>% filter(TotalVotes >= 1500)
gw16PerParty <- rbind(gw16PerParty, other)

# Create new columns to identify the election, percentage vote and different
details required for formatting of the visualisations
gw16PerParty$Election <- "2016"
gw16PerParty$PercentageVote <-
round(gw16PerParty$TotalVotes*100/sum(gw16PerParty$TotalVotes),1)

```

```

gw16PerParty$PartyShort <- c("IND", "FF", "FG", "SF", "SDP", "LAB", "REN",
"GP", "OTH")#, "OTH")
gw16PerParty$Label <- format(gw16PerParty$TotalVotes, big.mark = ",", )
gw16PerParty$Label2 <- paste0(gw16PerParty$PercentageVote, "%" )

# Prepare Galway West 2020 dataframe for analysis
# Extract and sort date per candidate
# drop first column as it just contains N/A values
gw20$Party <- NULL
# Rename new first column to Party
colnames(gw20)[1] <- "Party"
colnames(gw20)[4] <- "Votes"

# extract data per candidate
gw20PerCandidate <- gw20 %>%
  select(Party, Candidate, Votes)%>%
  arrange(desc(Votes))

# Extract and sort date per party
gw20PerParty <- gw20 %>%
  group_by(Party) %>%
  summarise(TotalVotes = sum(Votes)) %>%
  arrange(desc(TotalVotes))

# remove any NA values
gw20PerParty <- na.omit(gw20PerParty)

# Create new columns to identify the election, percentage vote and different
details required for formatting of the visualisations
gw20PerParty$Election <- "2020"
gw20PerParty$PercentageVote <-
round(gw20PerParty$TotalVotes*100/sum(gw20PerParty$TotalVotes),1)
gw20PerParty$PartyShort <- c("IND", "FF", "FG", "SF", "GP", "SDP", "LAB",
"AON", "OTH")
gw20PerParty$Label <- format(gw20PerParty$TotalVotes, big.mark = ",", )
gw20PerParty$Label2 <- paste0(gw20PerParty$PercentageVote, "%" )

# Merge election 2016 and 2020 into one dataframe
allPerParty <- rbind(gw16PerParty,gw20PerParty)
# create new column to set the type of data
allPerParty$Type <- paste0("GalwayWest-", allPerParty$Election )

# prepare national data
natData <- read.csv("../data/NationalResults.csv")
# select out required columns
natData<-natData%>%select(c(1,2,4))

# create subset of dataframe for parties that have less than 2% of the vote
other<-natData %>%

```



```

group_by(Election) %>%
filter(PercentageVote<2) %>%
select(Election, PercentageVote) %>%
summarise(sum(PercentageVote))

# set party on this subset to be Other
other$Party <- "Other"

other <- other %>%
  rename("PercentageVote" = "sum(PercentageVote)")

# add these other rows back to dataframe
natData <- natData %>% filter(PercentageVote>=2)
natData <- rbind(natData,other)

# Merge Independents into one party type ( if independent is in there name)
natDataNew2 <- mutate(natData, Party = ifelse(str_detect(Party,
"Independent"), "Independents", as.character(Party)))

# group this data and aggregate using sume
natDataNew <- natDataNew2 %>%
  group_by(Election,Party) %>%
  summarise_all(sum)

# filter out 2016 data, create columns required for visualisation (and to
help match with galway west data)
natData16 <- natDataNew %>%
  filter(Election == 2016) %>%
  select(Party, PercentageVote) %>%
  arrange(desc(PercentageVote))
natData16$PartyShort <- c("FG", "FF", "IND", "SF", "LAB", "A-PBP", "SDP",
"GP", "OTH", "REN")
natData16$Type <- paste0("National-", natData16$Election )
natData16$PercentageVote <- round(natData16$PercentageVote,1)
natData16$Label2 <- paste0(natData16$PercentageVote, "%" )

# filter out 2020 data, create columns required for visualisation (and to
help match with galway west data)
natData20 <- natDataNew %>%
  filter(Election == 2020) %>%
  select(Party, PercentageVote) %>%
  arrange(desc(PercentageVote))
natData20$PartyShort <- c("SF", "FF", "FG", "IND", "GP", "LAB", "OTH", "SDP",
"A-PBP")
natData20$Type <- paste0("National-", natData20$Election )
natData20$PercentageVote <- round(natData20$PercentageVote,1)
natData20$Label2 <- paste0(natData20$PercentageVote, "%" )

# merge these two back into a common dataframe

```

```

natDataAll <- rbind(natData16, natData20)

# merge galway west and national data into one dataframe
gwDataAll <- allPerParty %>% select(Election, PartyShort, PercentageVote,
Label2, Type)
natDataAll2 <- natDataAll %>% select(Election, PartyShort, PercentageVote,
Label2, Type) %>% ungroup()
allData = rbind(gwDataAll,natDataAll2 )

# prepare candidate data
# select out significant candidates from 2016 data
gw16PerCandidate <- gw16PerCandidate[c(1:3, 5,7),]
# carry out data cleaning procedures of names
gw16PerCandidate <- gw16PerCandidate %>% ungroup() %>%
  mutate(Candidate = recode(Candidate, "Catherine Martina Ann Connolly" =
"Catherine Connolly",
                             "Sean Kyne" = "Seáin Kyne"),
Election="2016") %>%
gw16PerCandidate[1,1] <- "Dámon Á Cuá-v"
# select out significant candidates from 2020 data
gw20PerCandidate <- gw20PerCandidate[c(1,3:6),2:3]
gw20PerCandidate <- gw20PerCandidate %>% ungroup() %>%
  mutate(Election="2020")

# merge all candidate data into one dataframe
allPerCandidate = rbind(gw16PerCandidate, gw20PerCandidate)

# define party colours
# used colours from colourbrewer website: http://colorbrewer2.org which are
colour-blind friendly

party.colours <- c('FG' = '#2b8cbe', 'FF' = '#2ca25f', 'SF' = '#006837',
'LAB' = '#e34a33', 'GP' = '#add8e6', 'SDP' = '#c43c29', 'IND' = '#969696',
'REN' = '#feca5c', 'A-PBP' = 'darkgrey', "OTH"='#fbb4b9', 'AON' = '#b3cde3')

# plot to display vote per party for the 2016 election
theme_set(theme_classic())

# reorder the bars so that highest vote shows first
gw16PerParty <- gw16PerParty %>% mutate(Party = fct_reorder(Party, -
TotalVotes))

ggplot(gw16PerParty, (aes(x= Party, y=TotalVotes, fill=PartyShort))) +
  geom_col(width=1, colour="white") +

```

```

geom_text(aes(label=PartyShort, group = Party), colour="white", size =3,
position = position_stack(vjust = .6))+

  geom_text(aes(label=Label, group = Party), colour="black", size =3, vjust =
-0.5 )+

  scale_fill_manual(values = party.colours)+

  ggtitle("Galway West 2016 Election: First Preference Vote Per Party") +

  theme(
    legend.position = "none",
    axis.title.y = element_blank(),
    axis.line.y = element_blank(),
    axis.line.x = element_blank(),
    axis.ticks = element_blank(),
    axis.text.y = element_blank(),
    axis.text.x = element_blank(),
    axis.title.x = element_blank(),
    plot.title = element_text(vjust = 0, hjust = 0.25, size = 11),
    plot.margin = margin(6, 6, 3, 3),
    panel.background = element_blank(),
    panel.grid.major.y = element_blank(),
    panel.ontop = TRUE
  )

# plot to display vote per party for the 2020 election
theme_set(theme_classic())

# reorder the bars so that highest vote shows first
gw20PerParty <- gw20PerParty %>% mutate(Party = fct_reorder(Party, -
TotalVotes))

ggplot(gw20PerParty, (aes(x= Party, y=TotalVotes, fill=PartyShort))) +
  geom_col(width=1, colour="white") +

  geom_text(aes(label=PartyShort, group = Party), colour="white", size =3,
position = position_stack(vjust = .6))+
  geom_text(aes(label=Label, group = Party), colour="black", size =3, vjust =
-0.5 )+

  scale_fill_manual(values = party.colours)+

  ggtitle("Galway West 2020 Election: First Preference Vote Per Party") +

  theme(
    legend.position = "none",
    axis.title.y = element_blank(),
    axis.line.y = element_blank(),

```

```

axis.line.x = element_blank(),
axis.ticks = element_blank(),
axis.text.y = element_blank(),
axis.text.x = element_blank(),
axis.title.x = element_blank(),
plot.title = element_text(vjust = 0, hjust = 0.25, size = 11),
plot.margin = margin(6, 6, 3, 3),
panel.background = element_blank(),
panel.grid.major.y = element_blank(),
panel.ontop = TRUE
)

# Plot Change in vote per party between 2016 and 2020 elections
allPerPartyPlot <- allPerParty %>%
  mutate(PartyShort = fct_reorder(PartyShort, -TotalVotes)) %>%
  filter(!PartyShort %in% c("REN", "AON", "A-PBP"))

ggplot(allPerPartyPlot, aes(x = PartyShort, y=TotalVotes, fill=Election)) +
  geom_bar(position="dodge", stat="identity") +

  scale_y_continuous(name = "", breaks = seq(0,18000, by=2000)) +

  scale_fill_manual(values = c( "#80cdc1", "#dfc27d"), name = NULL) +

  ggtitle("Change in vote per party between 2016 and 2020 elections") +

  theme_classic() +

  theme(
    axis.line.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.ticks.y = element_blank(),
    axis.line.y= element_blank(),
    axis.title.x=element_blank(),
    axis.text.x = element_text( vjust = 5, size=9, face="bold"),
    axis.text.y = element_text( size=7),
    axis.title.y = element_text(size=9, face="bold"),
    legend.text = element_text(size=8),
    legend.title = element_blank(),
    legend.position = c(0.8,0.9),
    legend.key.size = unit(0.8,"line"),
    plot.title = element_text(vjust = 0, hjust = 0.25, size = 11),
    plot.margin = margin(6, 0, 3, 0),
    panel.background = element_blank(),
    panel.grid.major.y = element_line(size = 0.1, linetype = 'solid', colour
= "white"),
    panel.ontop = TRUE
  )

```

```

# Plot "Galway West and National Average per party comparison"
allData<-allData%>%mutate(PartyShort = fct_reorder(PartyShort,
PercentageVote))%>%arrange(desc(PercentageVote))

theme_set(theme_classic())

annotate_size <- 2.5
v_just= -5

# Reverse the order as follow
allData$Type <- factor(allData$Type, levels = rev(c('GalwayWest-2016',
'National-2016', 'GalwayWest-2020', 'National-2020'))))

ggplot(allData, aes(x=Type, y=PercentageVote, fill=PartyShort), size=10) +
  geom_col(width = 0.5) +

geom_text(data=subset(allData, PercentageVote >8), aes(label=Label2, group =
PartyShort), colour="white", size =4, position = position_stack(vjust = .5))
+

scale_y_continuous(limits = c(0, 101),
  expand = c(0, 0),
  breaks = seq(0,100, by = 25),
  name = "Percentage share of the Votes") +

  scale_fill_manual(values = party.colours, guide = guide_legend(reverse =
TRUE)) +

  ggtitle("Galway West and National Average per party comparison") +

  coord_flip() +

  theme(
    axis.title.y = element_blank(),
    axis.line.y = element_blank(),
    axis.ticks.y = element_blank(),
    axis.line.x = element_blank(),
    axis.ticks.x = element_blank(),
    plot.title = element_text(vjust = 0, hjust = 0.25, size = 11),
    plot.margin = margin(6, 6, 3, 3),
    panel.grid.major.y = element_line(size = 0.1, linetype = 'solid', colour
= "white"),
    legend.position = "right",
    legend.title = element_blank(),

  )

allPerCandidate<-allPerCandidate%>%mutate(Candidate = fct_reorder(Candidate,
Votes))%>%arrange(Votes)

```

```

ggplot(allPerCandidate , aes(x = Votes, y= Candidate), size=9) +
  geom_path(aes(group = Candidate), Candidate = "grey", size=0.5) +
  geom_point(aes(colour = Election), size = 3, alpha = 0.7) +

  geom_text(aes(label= Votes, colour = Election), vjust=-1.2, size=3,
na.rm = TRUE, show.legend = FALSE)+

  scale_colour_manual(values= c("#7b3294", "#008837"), name = "") +

  scale_x_continuous(limits = c(4000, 10000),
                     expand = c(0, 0),
                     breaks = seq(5000, 9000, by = 1000),
                     labels = c("5,000", "6,000", "7,000", "8,000",
"9,000"),
                     name = "Number of 1st Preference Votes") +

  ggtitle("Change in vote with candidates") +

  theme(axis.title.y = element_blank(),
        panel.grid.major.x =element_line(size=0.01, linetype='solid',
colour="lightgrey"),
        panel.grid.minor.x = element_blank(),
        panel.grid.major.y = element_blank(),
        axis.line.y = element_blank(),
        axis.line.x = element_blank(),
        axis.ticks.y = element_blank(),
        axis.ticks.x = element_blank(),
        axis.title.x = element_text(size=9),
        plot.margin = margin(6, 6, 3, 3),
        plot.title = element_text(vjust = 0, hjust = 0.25, size = 11),
        legend.position = c(0.95, 0.25),
        legend.text = element_text(size = 8),
        legend.key.size = unit(0.7, "lines"),
        legend.title = element_blank())

```