

## Data Visualization Assignment 3

### Introduction:

The 2020 Irish General Elections took place on 8<sup>th</sup> February and this report analyses and compares the past and present performance of the Parties and their Candidates. This report closely analyses the performance in the Galway West region but also compares the local performance of the parties against the national average.

### 1. Vote per party in each Election.

The following graphs provide detailed analysis of performance of the Parties in the 2016 and 2020 General Elections in the Galway West Region.

For plotting the graph, I used the colours which represent the Party as it is easy for association. The names of the parties were big, so I used the short names for representation. The bars are plotted in sorted order based on performance so that we can see the best performers first. We can see how many votes each party has received and also see their percent share. This allows us to read the data efficiently with exact numbers. I removed the legend from the plot as it was unnecessarily catching the attention. The same colour and style format is used for both the 2016 and 2020 Graphs in order to avoid any ambiguity.

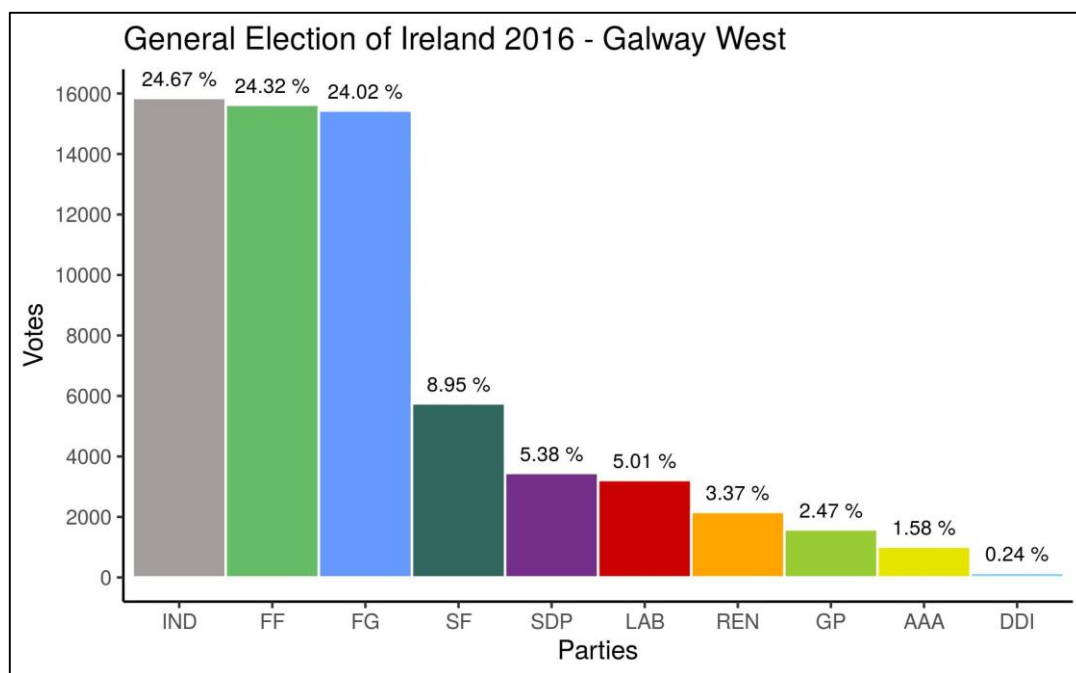


Figure 1 General Election of Ireland 2016 – Galway West

On the X-axis we can see the Parties name and on the Y-axis we can see the votes. I removed the gridlines from the background as they add no information and plain background gives better clarity. I also locked the scale from 0 to 16000 as the amount of votes was not more than 16000.

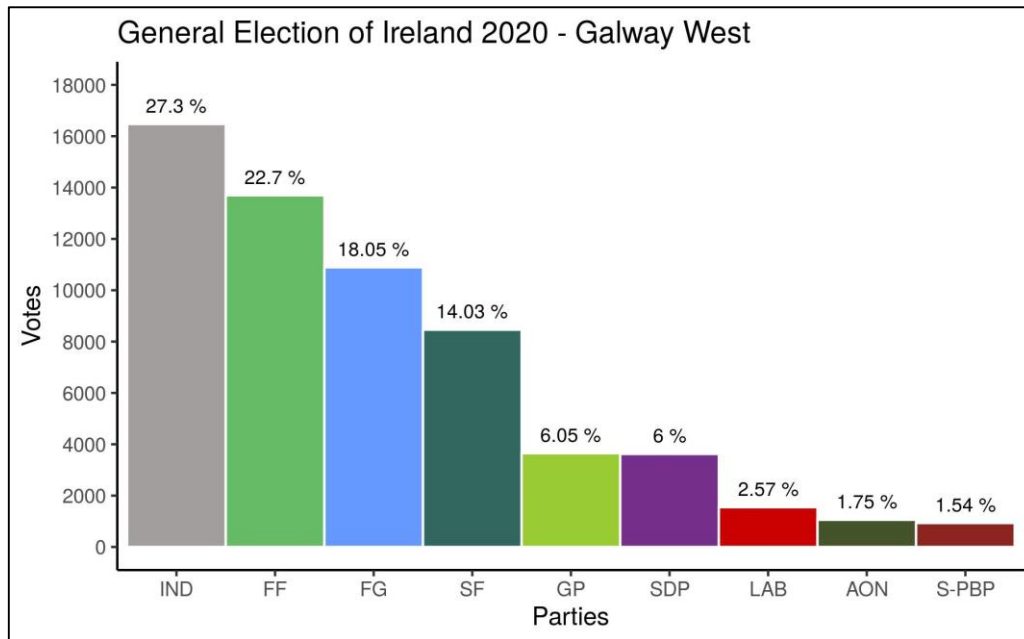


Figure 2 General Election of Ireland 2020 - Galway West

## 2. The change in vote per party from 2016-2020:

The following graph gives us comparison of the past and present performances of the parties in 2016 and 2020 in the Galway West Region.

For showcasing the comparisons, I used dodged bar plots. Dodged bar plots are besides each other and we can easily notice the changes in the performance between two elections. I used different colours for the votes on 2016 and 2020 elections so that each bar can be differentiated and information can be processed correctly. As the party names were big I again used the short names to make the plot more readable. The legend is required in this plot as it is key to understand which bar signify which year. I used dashed lines so that we can get a fair idea of number of votes. I am a colour blind person so I chose all the colour blind friendly colours and referred to the cvd friendly palette.

Palette: <https://www.color-hex.com/color-palette/49436>

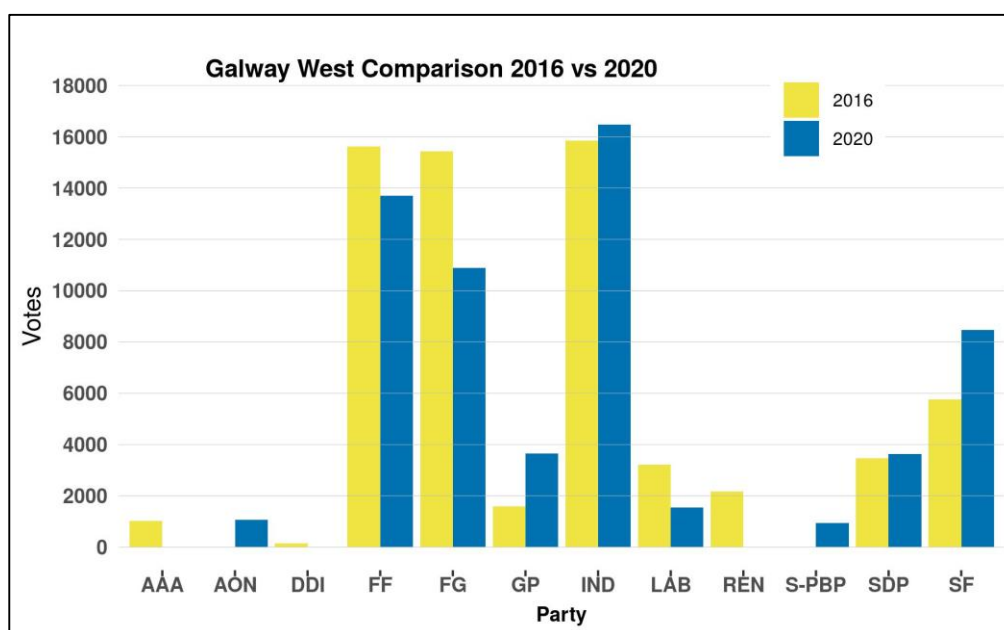


Figure 3 Galway West Comparison 2016 vs 2020

### 3. A comparison to the national average for party share of the vote:

This plot shows the comparison of the parties in Galway West Region against their National Performance in 2020 General Elections

The National Average data is fetched from the **RTE** website. I again chose dodged barplots as they do a very good job in displaying comparison. On the Y-axis there is the percentage of the votes and on the X-axis there are short names of the Parties. The colours are used for the plot belong to CVD friendly palette. The Legend signifies the local and national performance. I removed the Y-axis line and kept the plot minimal.

**References:** <https://www.rte.ie/news/election-2020/results/#/national>

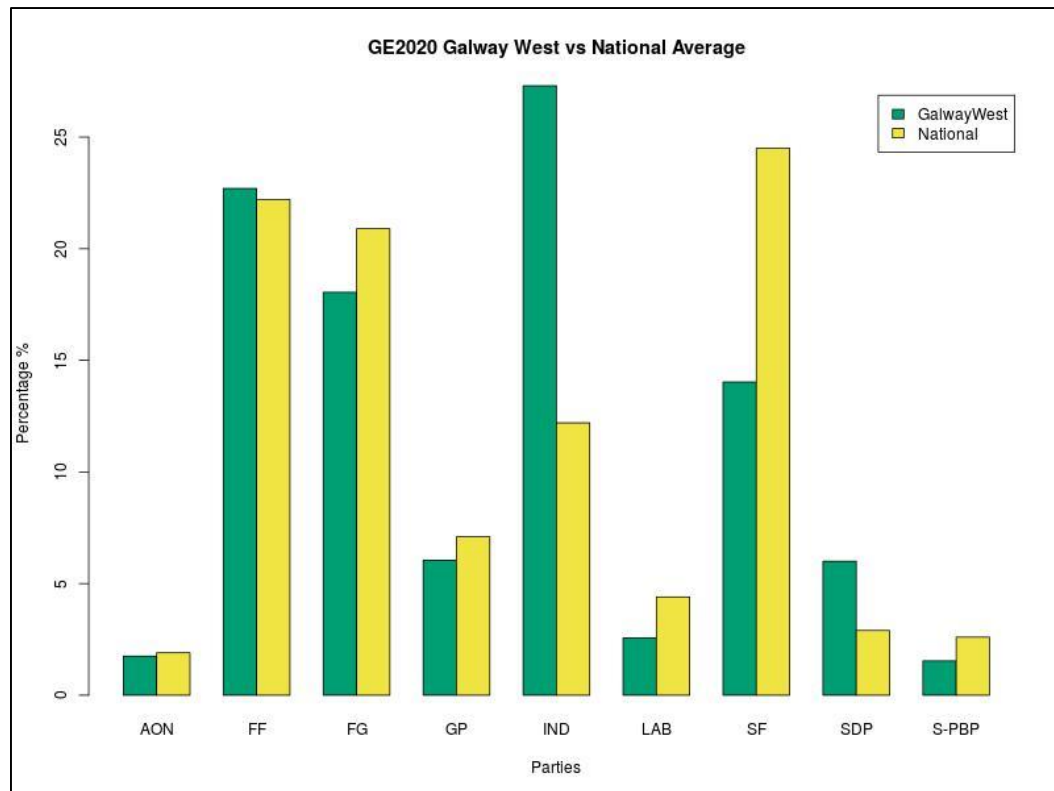


Figure 4 GE2020 Galway West vs National Average

### 4. The change in vote for the most significant candidates in both elections. It is up to you to judge what candidates meet that criteria.

In the following graph, we can see the performance of the top performing candidates from 2016 to 2020 General Elections. I filtered the top performing candidates by choosing only those candidates who have received more than 5000 Votes in 2020 General Election.

Dot plots are excellent for continuous and quantitative data. The names of the candidates are essential while comparing their performance; therefore I used their complete names on the in order to avoid any ambiguity with short names. On the Y-axis we can see the names of the top performing candidates and on the X-axis we can see the number of Votes. With this alignment, full names of the candidates can be shown. The line connecting dots shows the increase or decrease in the number of votes and each dot has a label (Year) so the information does not mislead. The gridlines in the background are important as they give

better understanding of the votes and the performance. I used the colour blind friendly Viridis palette for colour representation.

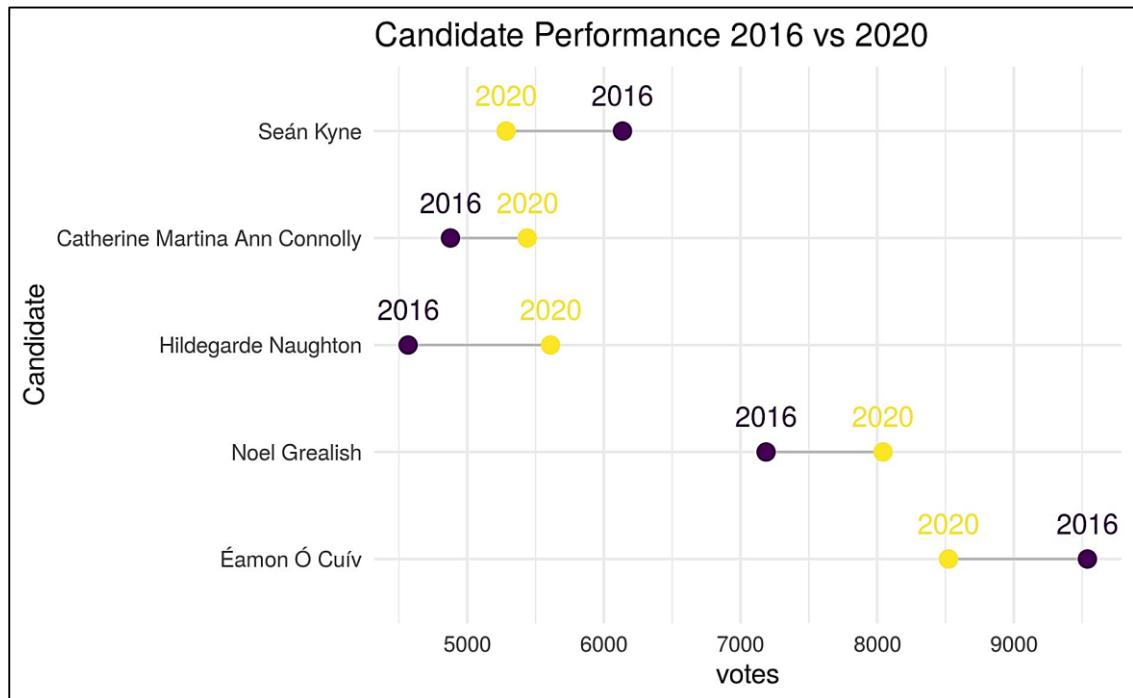


Figure 5 Candidate Performance 2016 vs 2020

### Conclusion:

From the above analysis we can understand that the Independent Party has performed better in 2020 General Elections as compared to 2016 Elections and retained its Prime Position in the Galway West Region. From the plots it is also evident that Independent party performs better in the Galway West Region as its performance in this part is better than the National Average. Sinn Féin is the best performing party on National Scale in the 2020 General Elections.

## Appendix:

### Data Importing and Pre-processing:

```
# Loading required packages
```

```
library(tidyverse)
```

```
library(ggplot2 )
```

```
#Column Names for the 2016 Data
```

```
column_names2016 <- c("Constituency", "Candidate_surname", "Candidate_firstname", "Result",  
"Count", "Transfers", "Votes", "Total_votes", "Candidate_id", "Party")
```

```
# Loading the 2016 General Election Data
```

```
general_elections_2016 <- read.csv('./2016-04-28_general-election-count-details-galway-west-  
csv_en.csv', col.names = column_names2016)
```

```
#Filtering the 2016 Data
```

```
#The Party names are big and they occupy lot of space on plot so using the #shortnames for them:
```

```
party_short <- c('AAA'="AAA", 'Direct Democracy Ireland'="DDI", 'Fianna Fail'="FF", 'Fine  
Gael'="FG", 'Green Party'="GP", 'Independent'="IND", 'Labour Party'="LAB", 'Renua'="REN", 'Sinn  
Fein'="SF", 'Social Democrats'="SDP", 'Aontú'="AON", 'Solidarity-PBP'="S-PBP")
```

```
#As per the Readme file we are just focusing on the 1st Count so filtering data #for first count only  
and updating the dataframe
```

```
general_elections_2016 <- filter(general_elections_2016, general_elections_2016$Count ==1)
```

```
#Merging Name and Surname of candidate into 1 column
```

```
general_elections_2016 <- mutate(general_elections_2016, Candidate = paste(Candidate_firstname,  
Candidate_surname)) %>%  
select(-Candidate_firstname, -Candidate_surname)
```

```
party_names_2016 <- c("Social Democratic Party"="Social Democrats", "Ind"="Independent", 'Fine  
Gael'="Fine Gael")
```

```
general_elections_2016 <- mutate(general_elections_2016, Party =  
str_trim(str_remove_all(general_elections_2016$Party, party_names_2016)))
```

```
#Adding the Shortname column to the dataframe
```

```
general_elections_2016$'Short' <- party_short[general_elections_2016$Party]
```

```
#Selecting only required columns
```

```
general_elections_2016 <- select(general_elections_2016, Party, Short, Candidate, Votes)
```

```
party.colours <- c('FG' = '#6699FF', 'FF' = '#66BB66', 'SF' = '#326760', 'LAB' = '#CB0000', 'GP' =  
'#99CC33', 'SDP' = '#752F8B', '#fb9a99', 'IND' = '#a39e9e', 'REN' = '#FFA500', 'DDI' = '#87CEFA', 'AAA'  
='#E5E500', 'S-PBP' = '#8E2420', 'AON' = '#44532A')
```

```

#Fetched the National Average data from RTE.ie website and put it into a dataframe

Party <- c('Fine Gael','Fianna Fail','Sinn Fein','Labour Party','Solidarity–PBP','Green Party','Social
Democrats','Aontú','Independent')

#National average vote percentage
Votes_NA <- c(20.9,22.2,24.5,4.4,2.6,7.1,2.9,1.9,12.2)

#Combining and making one dataframe
national_avg_20 <- data.frame(Party, Votes_NA)

#https://www.rte.ie/news/election-2020/results/#/national

#Importing 2020 data
general_elections_2020 <- read.xlsx("/GalwayWest-2020.xlsx", sheetIndex = 1, startRow=2)

#Deleting the Party Column as it had NA values for colours
general_elections_2020 <- subset( general_elections_2020, select = -Party)

#Renaming the NA column to Party
colnames(general_elections_2020)[1] <- "Party"

#As per the Readme file we are just focusing on the 1st Count so filtering data #for first count only
and renaming it to Votes
general_elections_2020 <- general_elections_2020 %>% rename(Votes=Count.1)
general_elections_2020 <- select(general_elections_2020, Party, Candidate, Votes)

party_names_2020 <- c("Fianna Fáil" = "Fianna Fail","Sinn Féin" = "Sinn Fein","Solidarity PBP" =
"Solidarity PBP",'Aontú'="Aontú")

candidate_names_en <- c("Áine O'Connell" = "Áine O'Connell", "Catherine Connolly"="Catherine Martina
Ann Connolly", "Niall Ó Tuathail"="Niall O' Tuathail")

general_elections_2020 <- general_elections_2020 %>%
mutate(Party = str_trim(str_replace_all(general_elections_2020$Party, party_names_2020))) %>%
drop_na()

#Votes are in Double format in 2020 dataset; we need to make them to int so that #they can be
merged later with 2016 dataset
general_elections_2020[, 'Votes'] <- strtoi(gsub(",","",general_elections_2020[, 'Votes']))

general_elections_2020 <- mutate(general_elections_2020, Candidate =
str_trim(str_replace_all(general_elections_2020$Candidate, candidate_names_en)))

#Adding the Shortname column to the dataframe
general_elections_2020$Short <- party_short[general_elections_2020$Party]

```

## **First Plot**

#Plot 1: The vote per party in each election.

#For Plotting the first plot we dont need all the data from the dataframe hence I # am putting the data into new dataframe so it can be modified.

```
GE16_votes <- general_elections_2016
```

#Grouping the data based on party

```
GE16_votes <- general_elections_2016 %>% group_by(Party, Short) %>% summarise(Votes =  
sum(Votes))
```

# Calculating the percentage of the votes received by each party

```
GE16_votes$Votes_Percent <- paste(round(GE16_votes$Votes*100/sum(  
GE16_votes$Votes),2), "%")
```

# Plotting the data on the plot in descending order of votes percent

```
ggplot(GE16_votes, aes(x=reorder(Short, -Votes), y=Votes, fill=Short)) +  
  geom_col(width=1, colour='White') +  
  scale_y_continuous(limits = c(0, 16000),  
    breaks = seq(0,16000, by = 2000),  
    name = "Votes") +  
  geom_text(aes(label=Votes_Percent, group = Party), size =3, vjust=-0.8) +  
  scale_fill_manual(values = party.colours)+  
  ggtitle("General Election of Ireland 2016 - Galway West")+  
  xlab("Parties")+  
  theme(legend.position="none")
```

```
ggsave("Graph-1_2016.jpeg")
```

## **Second Plot**

#2020 General Election data

```
GE20_votes <- general_elections_2020
```

#Grouping the data based on party

```
GE20_votes <- general_elections_2020 %>% group_by(Party, Short) %>% summarise(Votes =  
sum(Votes))
```

# Calculating the percentage of the votes received by each party

```
GE20_votes$Votes_Percent <- paste(round(GE20_votes$Votes*100/sum(  
GE20_votes$Votes),2), "%")
```

#Allocating data to be used in later plotting

```
GE20_votes$Votes_Percent_nosign <- paste(round(GE20_votes$Votes*100/sum(  
GE20_votes$Votes),2))
```

# Plotting the data on the plot in descending order of votes percent

```
ggplot(GE20_votes, aes(x=reorder(Short, -Votes), y=Votes, fill=Short)) +
```

```

geom_col(width=1, colour='White') +
scale_y_continuous(limits = c(0, 18000),
  breaks = seq(0,18000, by = 2000),
  name = "Votes") +
geom_text(aes(label=Votes_Percent, group = Party), size =3, vjust=-0.8) +
scale_fill_manual(values = party.colours)+
ggtitle("General Election of Ireland 2020 - Galway West")+
xlab("Parties")+
theme(legend.position="none")

ggsave("Graph-1_2020.jpeg")

```

### **Third Plot**

```

#Merging both 2016 and 2020 dataset

#The change in vote per party from 2016-2020
#Summarizing the 2016 data
GEV_2016 <- general_elections_2016 %>% group_by(Party,Short) %>%
summarise('2016'=sum(Votes))

#Summarizing the 2020 data
GEV_2020 <- general_elections_2020 %>% group_by(Party,Short) %>%
summarise('2020'=sum(Votes))

#Merging the data
mergedData <- full_join(GEV_2016, GEV_2020, by=c("Party", "Short"))
mergedData1 <- mergedData %>% gather(year, votes, '2016':'2020')

#Custom colors used color blind freindly colors
custom_palette <- c("#f0e442", "#0072b2")
theme_set(theme_classic())
ggplot(mergedData1, aes(x=Short, y=votes, fill=year)) +
  geom_bar(position="dodge", stat="identity") +
  scale_y_continuous(limits = c(0, 18000),
    breaks = seq(0,18000, by = 2000),
    name = "Votes") +
  scale_fill_manual(values=custom_palette) +
  ggtitle("Galway West Comparison 2016 vs 2020") +
  labs(x = "Party") +
  theme(
    legend.text = element_text(size=8),
    legend.title = element_blank(),
    legend.position = c(0.8, 0.9),
    axis.line.y = element_blank(),
    axis.ticks.y = element_blank(),
    axis.line.x = element_blank(),
    axis.text.x = element_text( vjust = 2, size=9, face="bold"),
    axis.text.y = element_text( size=9, face="bold"),
    axis.title.x = element_text( size=9, face="bold"),

```



```

panel.background = element_blank(),
panel.grid.major.y = element_line(size = 0.1, linetype = 'solid', colour = "grey"),
panel.ontop = TRUE,
plot.title = element_text(vjust=-5, hjust=0.2, face='bold', size = 11)
)

```

```
ggsave("Graph-2.jpeg")
```

#### **Fourth Plot**

```
#A comparison to the national average for party share of the vote
```

```
#Getting the National Average data from RTE and preprocessing it
national_avg_20$Party <- trimws(national_avg_20$Party)
```

```
#2020 votes
```

```
GE2020Votes = select(GE20_votes, Party, Short, Votes_Percent_nosign)
```

```
#Adding the Shortname column to the dataframe
```

```
national_avg_20$Short <- party_short[national_avg_20$Party]
```

```
#Merging both datasets
```

```
mergedData_Comp <- full_join(GE2020Votes, national_avg_20, by=c("Party", "Short"))
```

```
#Renaming the column names
```

```
colnames(mergedData_Comp)[3] <- "GalwayWest"
```

```
colnames(mergedData_Comp)[4] <- "National"
```

```
#The column was in cha format so converted to double
```

```
mergedData_Comp$GalwayWest <- as.double(mergedData_Comp$GalwayWest)
```

```
mergedData_Comp
```

```
# Grouped Bar Plot
```

```
#Plotting the graphs
```

```
#jpeg(file="Graph-3.jpeg", width=800, height=600)
```

```
plotData <- rbind(mergedData_Comp$GalwayWest, mergedData_Comp$National)
```

```
barplot(plotData, beside=T, names.arg = mergedData_Comp$Short, ylab = "Percentage %", xlab =
"Parties", col=c("#009e73", "#f0e442"), legend = c("GalwayWest", "National"), main="GE2020 Galway
West vs National Average")
```

```
#dev.off()
```

#### **Fifth Plot**

```
#Preprocessing the data for Task 4
```

```
#Handling special characters in the name
```

```
GE16_4 <- general_elections_2016 %>% mutate(Candidate = str_replace(Candidate, "O'", "Ó ")) %>%
```

```

mutate(Candidate = str_replace(Candidate, "Sean", "Seán")) %>%
select(Candidate, Party, Votes)

#Renaming the columns and adding the trimmed data to df
GE16_4$Votes2016 <- GE16_4$Votes
GE16_4$Candidate = trimws(GE16_4$Candidate)

#Selecting only required columns 2016
Candidates2016 <- GE16_4 %>% select(Candidate, Votes2016)

#Selecting only required columns 2020
GE20_4 <- general_elections_2020 %>% select(Candidate, Party, Votes)
#Same formatting for 2020 data
GE20_4$Votes2020 <- GE20_4$Votes
GE20_4$Candidate = trimws(GE20_4$Candidate)

#selecting required columns
Candidates2020 <- GE20_4 %>% select(Candidate, Votes2020)

#Comparison of candidates merged together
CandidateComparison <- full_join(Candidates2016, Candidates2020)
CandidateComparison <- CandidateComparison %>% filter(CandidateComparison$Votes2020 >=
5000) %>% drop_na()

#Plotting the 4th Plot

#preprocessing the data
dataProcess <- CandidateComparison %>% gather(year, votes, Votes2016:Votes2020)
dataProcess$year[dataProcess$year=="Votes2016"] <- '2016'
dataProcess$year[dataProcess$year=="Votes2020"] <- '2020'

#Nameorder
nameorder <- dataProcess %>% filter(year == "2020") %>%
  arrange(-votes) %>%
  mutate(Candidate = factor(Candidate, levels = .$Candidate))

#Putting levels based on nameorder
dataProcess$Candidate = factor(dataProcess$Candidate, levels = nameorder$Candidate)

#Plotting the graph
ggplot(dataProcess , aes(x = votes, y= Candidate)) +
  geom_line(aes(group = Candidate), colour = "grey", size=0.5) +
  geom_point(aes(colour = year), size = 3, alpha = 0.7) +
  scale_color_viridis_d()+
  geom_text(aes(label=year, colour = year), vjust=-1.2, size=4, na.rm = TRUE, show.legend = FALSE)
+
  theme_minimal() + theme(legend.position = "none")+
  ggtitle("Candidate Performance 2016 vs 2020")

ggsave("Graph-4.jpeg")

```