Data Visualisation Project

Assignment 2 – Report

Ireland Census 2011 - 2016: Changing Socio-Economic Factors

Module code : SPEC9995: 2022-23

Ciaran Finnegan
Student No : d21124026
TU060 Yr. 2
Data Science MSc

9/12/2022

Table of Contents

1 Assignment Overview

1.1 Introduction

This document covers the design, implementation, and observations on all parts of the December 2022 CA(2) for the Data Visualisation module (TU060 – DS – Year 2).

This assignment is based on Irish Census data from 2011 and 2016.

The three sets of visualisations provide a comparison on a number of social and economic data points gathered by both census.

1.2 Problem Statement

The 2011 Census took place in the midst of significant turmoil, when Ireland required external support to avoid economic collapse. This period of austerity had a significant impact on employment and prosperity for the Irish population.

Five years later, in 2016, Ireland was considered, by some analysts, to be starting to emerge from this period of uncertainty.

The visualisations in this assignment attempt to show graphically, using a certain subset of information, that 2016 data points show an improving situation for the Irish population since the difficult days of 2011.

1.3 Datasets and Intended Audience

The datasets for the visualisations in the assignment come from two sources;

- Graphical information on Irish county boundaries from the Ordnance Survey of Ireland (OSI) website.
- Census 'theme' data for 2011 and 2011, compiled at Irish county level by the Central Statistics Office (CSO), in the following domains;
    - Employment rates per county.
    - The size of various socio-economic groups in Ireland.
    - The perceived health/wellbeing rating of the population in each county.

The intended audience is anyone looking to understand if economic actions taken in Ireland in 2011 resulted in a measureable improvement in employment and prosperity five years later.

2 Part 1: Data Pre-processing

2.1 Data Exploration

2.1.1 Geographical Data

The first visualisation uses a map of Ireland with county boundaries. Code libraries are available within the ***R*** language to read detailed geographical information and then augment with additional attributes.

The OSI provide a 2019 data file with Irish country boundaries, in a ***.shp*** format, at his location;

https://data-osi.opendata.arcgis.com/maps/osi::counties-osi-national-statutory-boundaries-2019-generalised-20m

The *readShapePoly* function in **R** reads this format and allows manipulation and addition of data points for use in the *ggplot* graphing function.

The map data is 'clean', but it is combined with additional data elements, as described in Section 0 below, to provide the richness of detail in Visualisation One.

2.1.2 'Themes' from Census Data

The 2011 and 2016 Census used a largely similar grouping of numerical data points under 15 categories or 'Themes'. Within many of these were sub-tables of data, breaking the Themes down into lower levels of granularity.

A glossary of the census themes is located here;
https://www.cso.ie/en/media/csoie/census/census2016/census2016boundaryfiles/SAPS_2016_Glossary.xlsx

This assignment looks at the Census numbers collated at the Irish county level for 2011 and 2016 for the following themes;

- **Theme 8:** *Principal Status*. The focus in on the population of unemployed in the workforce for each county.
- **Theme 9**: *Social Class and Socio-Economic Group*. This captures the number of people in each county who identify as members of a given employment category. As an example, Class A and B refer to Employers and Higher Professionals respectively while Class G is for 'Unskilled'.
- **Theme 12**: *Disability, Carers and General Health*. For this assignment, we only consider sub-table 3 that captures numbers of people by how they perceive their general health/wellbeing.

The 2011 Census theme data is available from the CSO here:
https://www.cso.ie/en/media/csoie/census/documents/saps2011files/AllThemesTablesCTY.csv

2016 theme data was less straightforward to find on the CSO website, but a copy is stored on Kaggle at;

https://www.kaggle.com/datasets/stephenofarrell/irish-census-2016

2.2 Data Wrangling

As expected, the map and census data has no missing values. However, in order to improve the visualisations in this assignment the following 'data wrangling' steps were required;

- The OSI map data file contains all the information to render the 26 counties of the Republic of Ireland but contains no data points on attributes of the actual human population in each county. Thus the 'theme' files were loaded into a separate dataset for both 2011 and 2016 and ***joined*** separately to the county map dataframe, based on county name.

- There are 26 counties in the Republic of Ireland and therefore 26 rows in the OSI map ***.shp*** file read in the assignment code. The 2011 and 2016 Census county files break down certain metropolitan areas into further subdivisions. For example, Dublin County census data is split into the following sub areas; *Fingal*, *South Dublin*, *Dublin City*, and *Dún Laoghaire-Rathdown*. Thus, it is necessary to ***re-group*** these metropolitan areas back into a single county row first before merging with the map dataset. This reduces both 'theme' dataframes from 30+ rows back to 26.

- To add complexity, the 2011 and 2016 Census differ slightly on the metropolitan divisions. For example, Limerick City and Limerick County are different areas (with separate rows in the dataset) in the 2011 Census but are treated as a single area in 2016. Visualisation One had to cater for the slight structural changes in data capture between these census years.

- The 'theme' files only capture absolute numbers of peoples or households. It was necessary to add ***calculated fields*** to determine the ***rate*** of certain attributes per county. For example, the columns named ***T8_1_LFFJT*** and ***T8_1_ULGUPJT*** capture the population in each county looking for a first job or unemployed, respectively. These values have to be divided by the total possible workforce (***T8_1_TT***) to find the rate of unemployment in each county.

- A similar calculation is applied to find the ratio of Socio-economic group A and B in each county (as a proportion of total population).

- The percentage of those in '*Very Good Health'* is derived from the number recorded in that column for each county divided by the overall county population.

- 26 separate unemployment rates generated an unwieldy legend and graph for Visualisation One. It was necessary to add another derived field that broke down this continuous set of values into a manageable, and more meaningful, range of ***categorical*** percentage descriptions.

- The 2016 theme ***.csv*** file stores the *Dún Laoghaire-Rathdown* region for Dublin with a different text format than the 2011 file. The Irish language *fada* over the 'u' caused a ***data corruption*** error with the rendering of 2016 *ggplot* graphs. Thus, it is necessary to directly update this dataframe cell in **R** code before rendering the graphs in Visualisation Three (*Dún Laoghaire-Rathdown* is specifically labelled in the Scatter Plot).

3 Part 2: Visualisations

3.1 Visualisation One – Employment Rates per County

How did rates of employment change across Ireland between 2011 and 2016?

The OSI map data provides an outline of county boundaries in the Republic of Ireland and the rates of employment in each county for 2011 and 2016 are represented below.

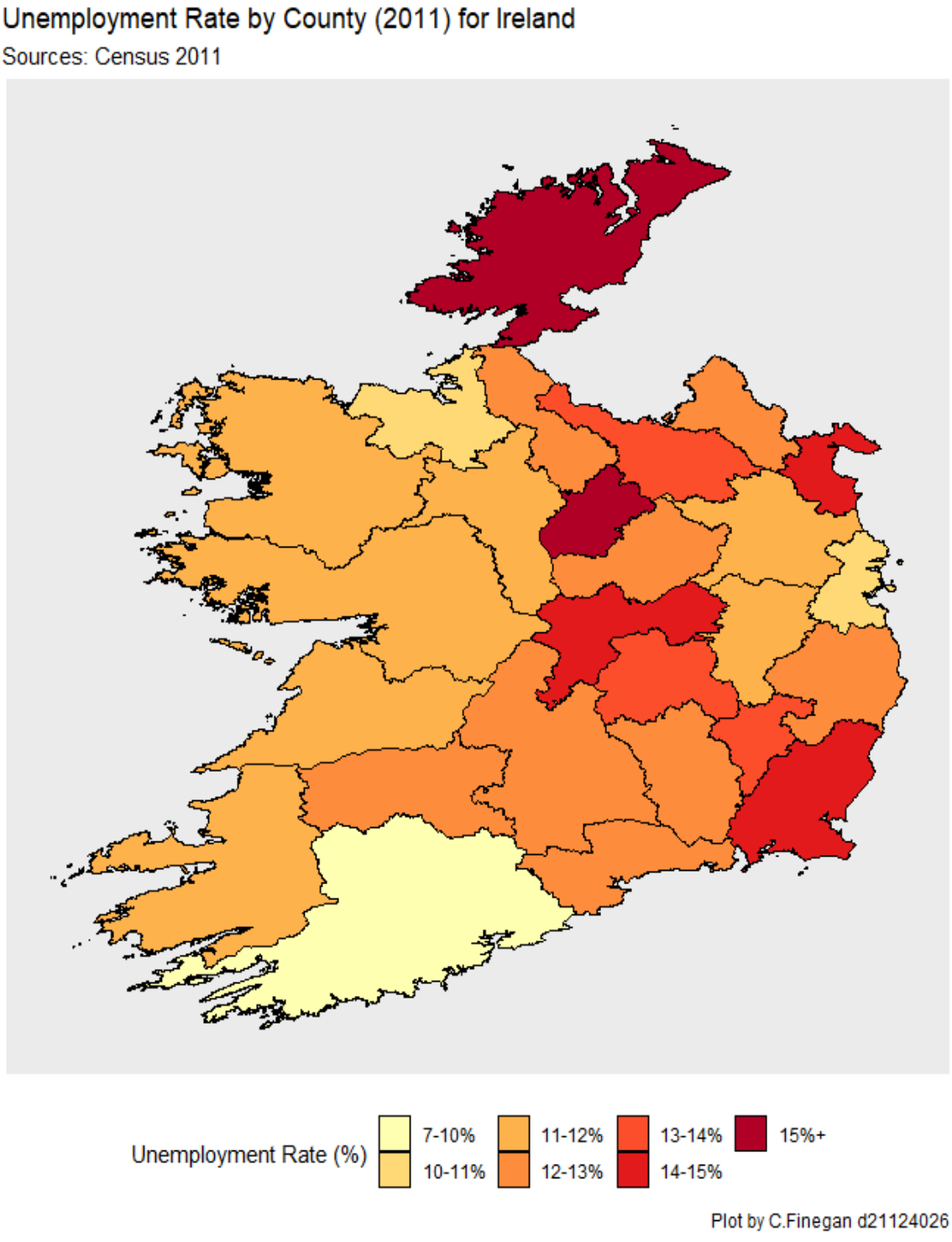


Figure – Irish Unemployment Rates by County (2011)

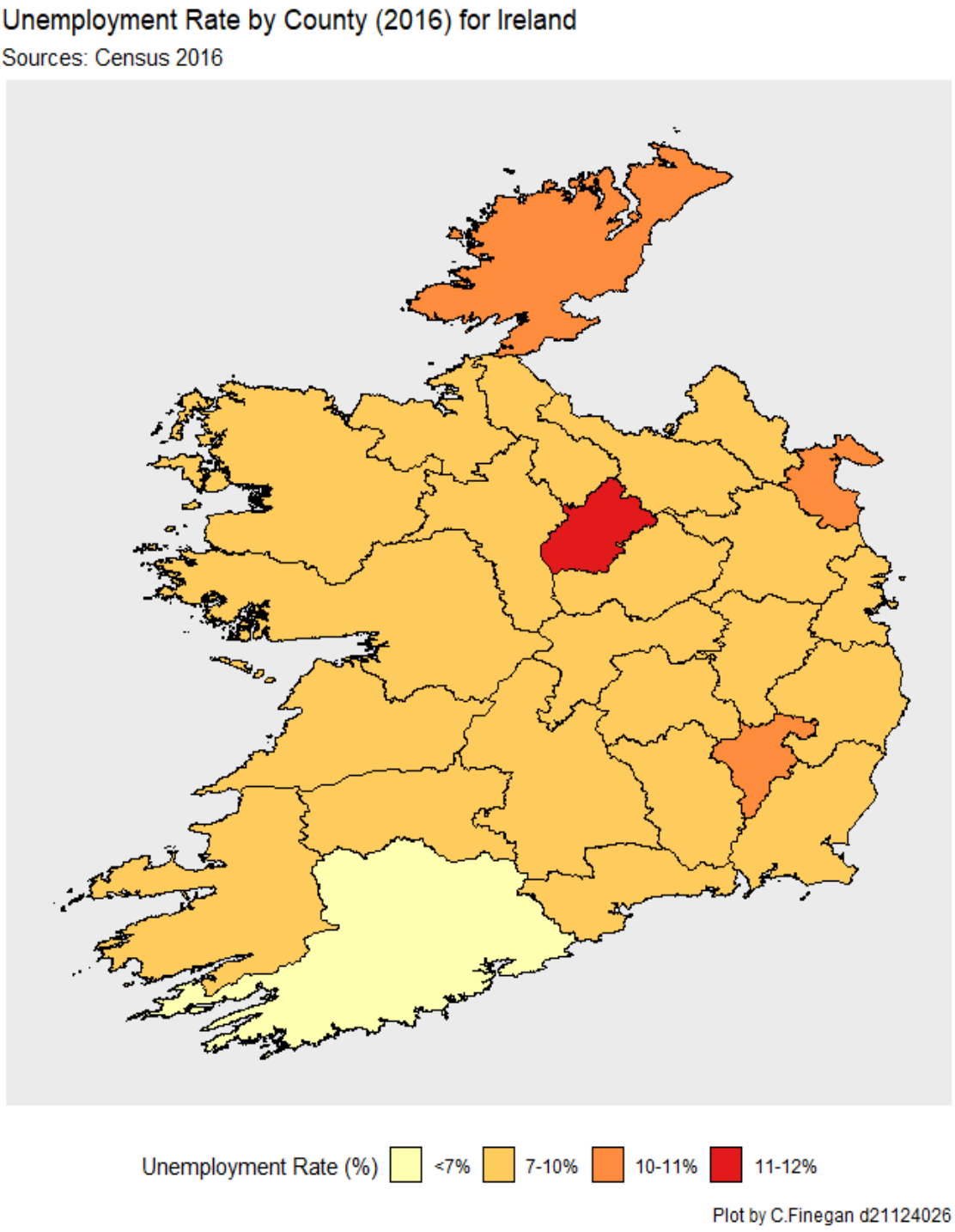


Figure – Irish Unemployment Rates by County (2016)

A graded colour scale is used to show the areas of highest unemployment (darkest colours).

The scale to measure the range of unemployment is kept as close as possible between 2011 and 2016 graphs.

It is evident that employment statistics improved in the five years after 2011. That said, areas like Longford continue to rank lowest in employment metrics.

3.2 Visualisation 2 – Socio-economic Groups in Ireland

Did the population of Ireland, as a whole, perceive that their social and economic status had improved in 2016, compared with 2011?

In the Irish Census questionnaire each respondent elects the socio-economic group into which they believe they belong. This reflects a person's sense of their status, and possible affluence.

The bar chart below is a comparison of the population numbers in each socio-economic group in both 2011 and 2016.

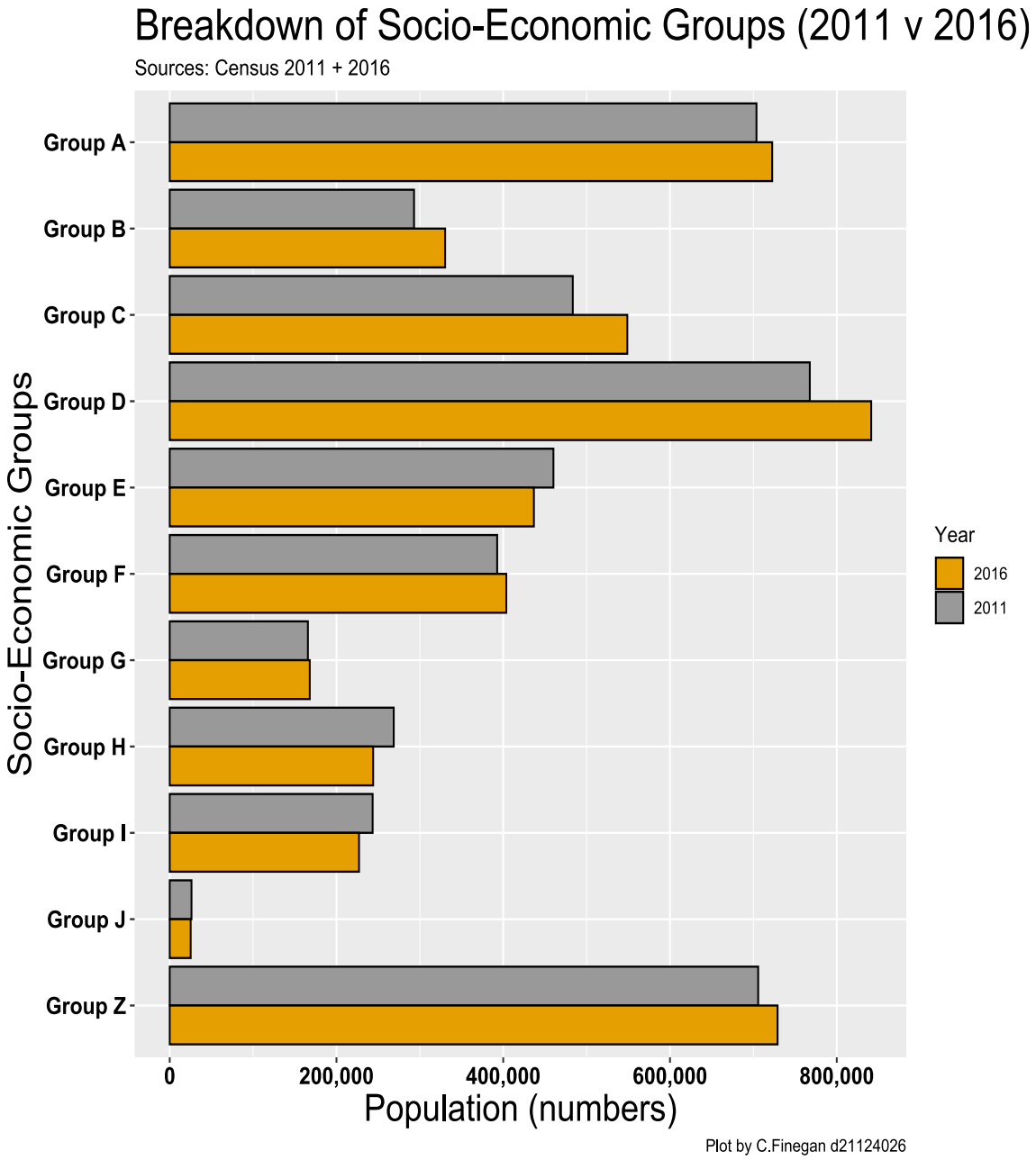


Figure – Irish Socio-economic Groups (2011 + 2016)

Of note in this graph is that these are absolute population numbers, aggregated to the national level from the individual county data.

Overall population in Ireland would have risen slightly in the period from 2011 to 2016.Thus, if other factors remained static, one would expect to see a general increase for all groups in 2016.

In fact, the graph shows that lower skilled employment groups (E, F, G) show near zero or negative growth from 2011. The A and B categories appear to be slightly larger than one might expect.

Thus, one could argue that there is an apparent increase in the general perceptions of prosperity across the Irish population in 2016.

3.3 Visualisation 3 – Impact of Socio-economic Group on Health

Does social standing impact on an individual's health? Has this relationship changed between 2011 and 2016?

The Irish Census does not directly record income levels, but it does measure the population by 'social class', which is an indicator of general affluence.

Do those who are 'better off' financially generally have better health (possibly either through lifestyle or access to health care)?

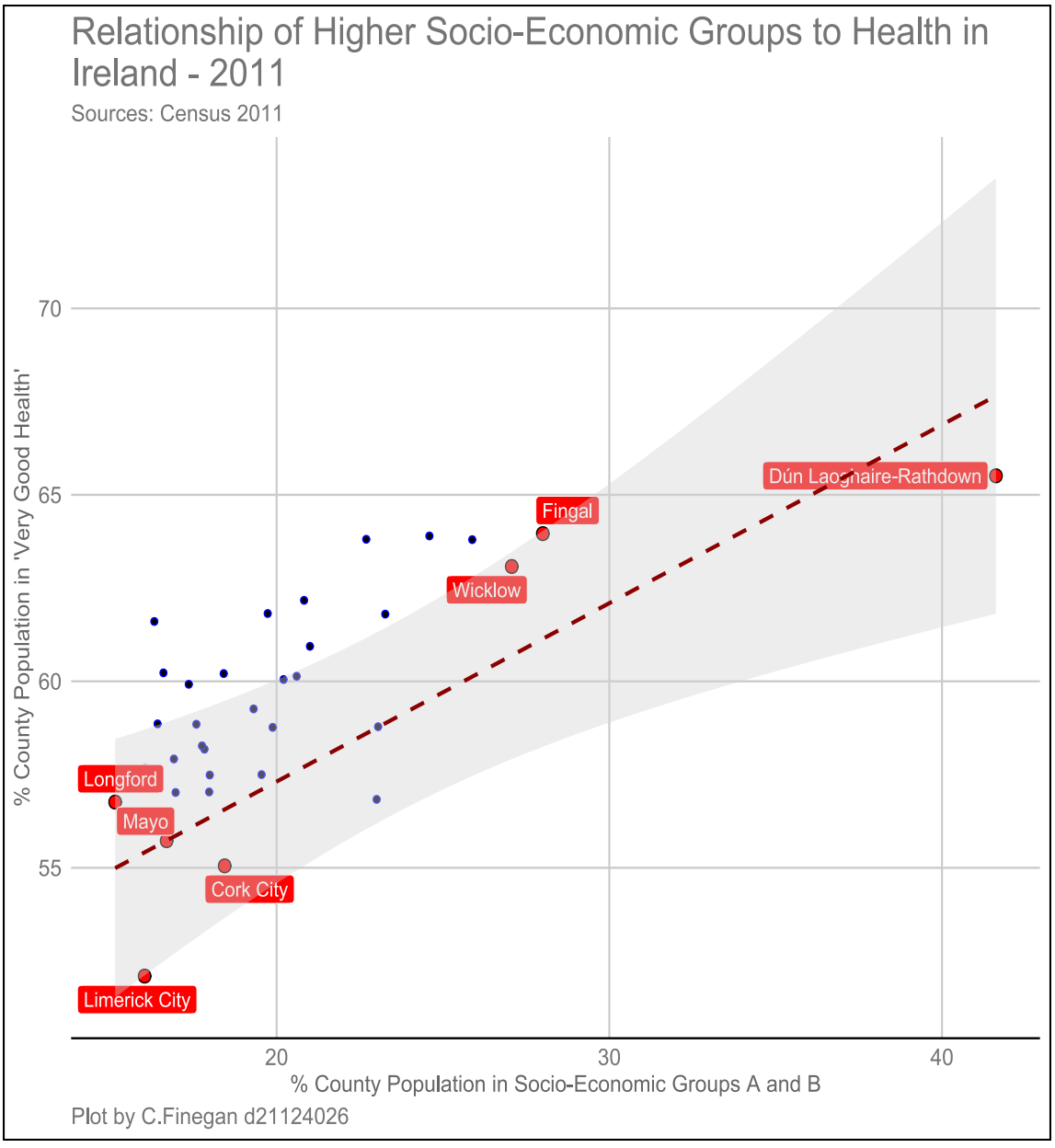


Figure – Irish Socio-economic Groups and Health (2011)

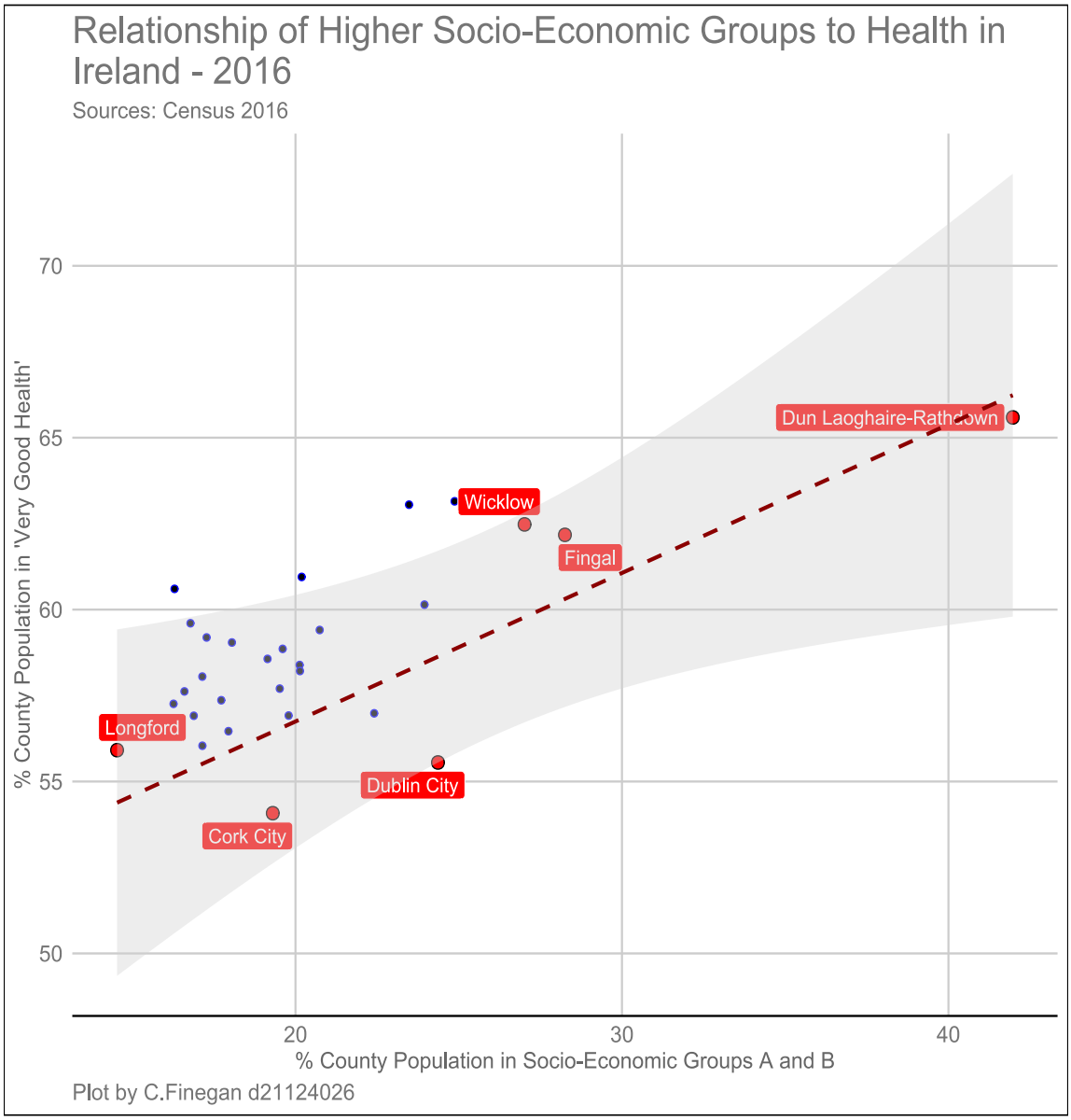


Figure – Irish Socio-economic Groups and Health (2016)

Both scatter plots show a relationship between those who identify in Socio-economic groups A and B and how many consider themselves in 'Very Good Health', according to the question on the Irish Census forms.

The data is represented for each county in Ireland, and we use this 26 rows of data points to determine if a correlation exists.

So, in Ireland, are the county areas with the highest percentage of population in Class A and B, the places with the healthiest people?

That positive correlation can be seen in both graphs, but it is not very strong. It has also not changed in any significant magnitude between 2011 and 2016. Such changes may take much longer to manifest.

For illustration, the county areas at the top and bottom of the scales are highlighted. Dún Laoghaire-Rathdown is certainly the area with the wealthiest and healthiest people but is somewhat of an outlier.

3.4 Previous Iterations with Assignment Visualisations

3.4.1 Visualisation One

The first map graph just used the ***absolute numbers*** of unemployed in each county to generate the colour gradients. This distorted the true picture because counties with low populations will have lower numbers of unemployed workers but may still have high ***rates*** of unemployment.

3.4.2 Visualisation Two

Originally this was going to be another two-graph comparison of data, similar to Visualisation One and Two.

However, comparing this type of bar chart data is much more effective with one graph that groups twin bars together by year/socio-economic group.

3.4.3 Visualisation Three

The original scatter pot contained just points for all Irish counties. This was sufficient for the general message of the visualisation but lacked 'sparkle'.

Adding labels to represent the county areas at the lower and upper areas of the graph added some useful context in terms of the spread of affluence across Ireland.

Initially two R files were used to generate the 2011 and 2016 graph data for Visualisation Three. However, as there was no complication with joining the OSI boundary data in this visualisation, the code was combined into a single file with a ***for…loop*** to generate multiple graphs.

4 Appendices

4.1 Appendix 1 – R Source Code – Visualisation One

4.1.1 2011 Unemployment Map Graph Code

```
# ------------------------------------------------ #

# Data Visualisation Assignment 2
# Visualisations in R
#
# Student No. d21124026
# Name : Ciaran Finnegan
# TU060 Data Science MSc
#
# December 2022


# Visualisation One - Comparison of Unemployment Rates
# Per County in Ireland in Census 2011 data
# ------------------------------------------------ #
# ------------------------------------------------ #


# load required libraries
library(rgeos)

library(maptools)

library(curl)

library(readr)

library(ggplot2)

library(sqldf)

library(tidyverse)

library(viridis)

library(dplyr)
```

```
# There were challenges with accessing data file locations for all datasets.
# To allow the R code to run anywhere the required data files have been hosted
# on GitHub where required.

# Build URL connecting string to download OSI country boundaries data file
# for counties in the Republic of Ireland
sGitHub_Datasource1 <-"https://github.com/JackDaedalus/DataVizLabs/raw/"
sGitHub_Datasource2 <- paste(sGitHub_Datasource1,"dfa3d486a5ea74a588e9768141b35f570eff3c57/CA2/", sep = "", collapse=NULL)
sGitHub_Datafile <- "Counties_-_OSi_National_Statutory_Boundaries_-_2019_-_Generalised_20m.zip"
sGitHub_Datasource <- paste(sGitHub_Datasource2,sGitHub_Datafile, sep = "", collapse=NULL)

county_map_source <- sGitHub_Datasource

# Download zip file from from GitHub and extract 2019 OSI County data for 26 Irish counties
temp_1 <- tempfile()
temp_2 <- tempfile()
source <- county_map_source
temp_1 <- curl_download(url = source, destfile = temp_1, quiet = FALSE)
unzip(temp_1, exdir = temp_2)



# Read the shape file extracted from the downloaded zip file
spdf <- readShapePoly(file.path(temp_2,"Counties___OSi_National_Statutory_Boundaries___Generalised_20m.shp"))

# Generate a new dataframe that will work with ggplot to generate Ireland county map
spdf@data$id <- rownames(spdf@data)
spdf.points <- fortify(spdf, region="id")
counties <- inner_join(spdf.points, spdf@data, by="id")
```

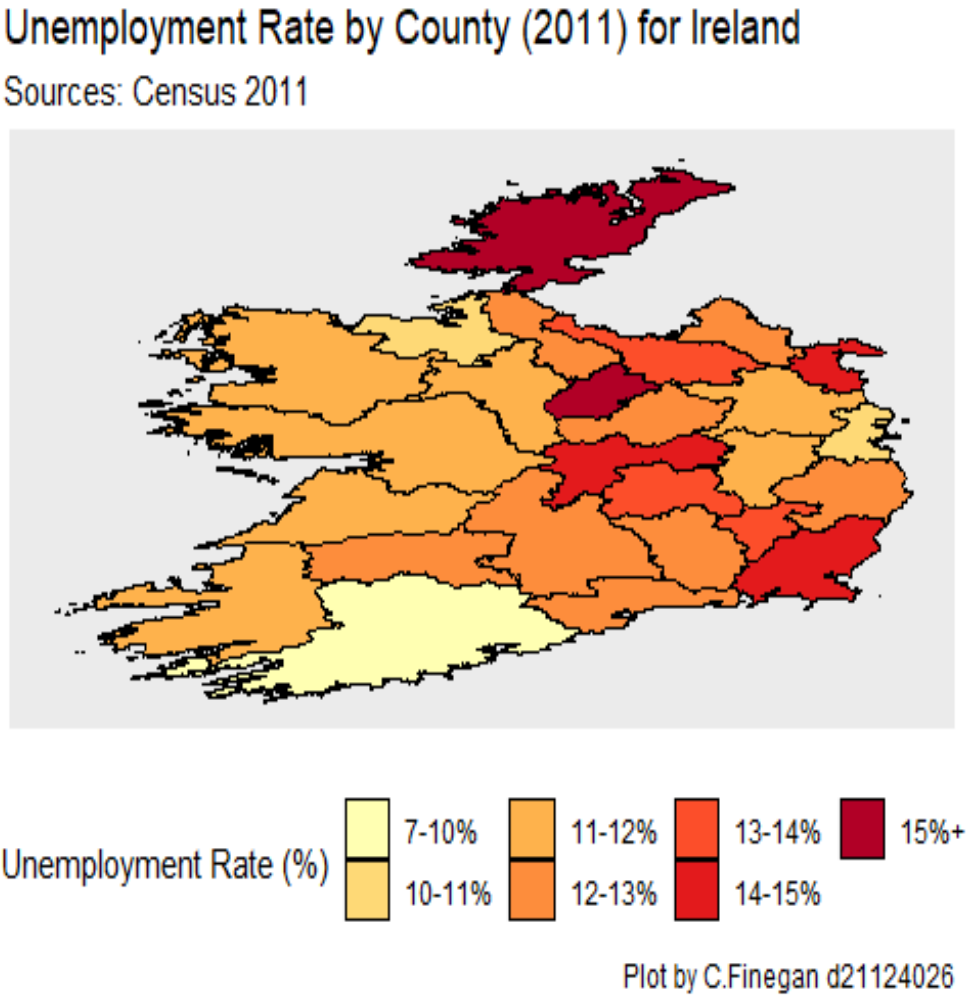
```
# ------------------------------------------------------#
# Load Census Theme Data for 2011 for Irish counties
# ------------------------------------------------------#


# This data is available directly from the CSO website
# Select only the required unemployment data
# Rename the columns to increase understanding of the data
df2011CountyThemes <- read_delim("https://www.cso.ie/en/media/csoie/census/documents/saps2011files/AllThemesTablesCTY.csv",show_col_types = FALSE) %>%
  select(GEOGID, GEOGDESC, T8_1_LFFJT, T8_1_ULGUPJT, T8_1_TT) %>%
  rename(Looking_for_Work = T8_1_LFFJT, Unemployed = T8_1_ULGUPJT, Total_Workforce = T8_1_TT)



# ------------------------------------------------------------#
# The Census theme data breaks down the counties in certain cases
# for metropolitan areas. This data needs to be re-merged to match
# the county boundaries in the OSI dataframe
# ------------------------------------------------------------#


# Start with Dublin...
df2011DublinThemes <- df2011CountyThemes %>%
  filter(GEOGID %in% c("C02","C03","C04","C05"))

# Group the country regions and sum all unemployment data for Dublin overall
df2011DublinThemesTotal <- sqldf("Select 'C35' as GEOGID,
                                 'Dublin' as Dublin,
                                 sum(Looking_for_Work),
                                 sum(Unemployed),
                                 sum(Total_Workforce)
                                 from df2011DublinThemes
                                 group by Dublin")



# The Cork areas are combined next...
df2011CorkThemes <- df2011CountyThemes %>%
  filter(GEOGID %in% c("C17","C18"))

# Group the country regions and sum all unemployment data for Dublin overall
df2011CorkThemesTotal <- sqldf("Select 'C36' as GEOGID,
                                 'Cork' as Cork,
                                 sum(Looking_for_Work),
                                 sum(Unemployed),
                                 sum(Total_Workforce)
                                 from df2011CorkThemes
                                 group by Cork")
```

```
# The Limerick areas are combined next...
df2011LimerickThemes <- df2011CountyThemes %>%
  filter(GEOGID %in% c("C20","C21"))

# Group the country regions and sum all unemployment data for Limerick overall
df2011LimerickThemesTotal <- sqldf("Select 'C36' as GEOGID,
                                    'Limerick' as Limerick,
                                    sum(Looking_for_Work),
                                    sum(Unemployed),
                                    sum(Total_Workforce)
                                    from df2011LimerickThemes
                                    group by Limerick")



# The Tipperary areas are combined next...
df2011TipperaryThemes <- df2011CountyThemes %>%
  filter(GEOGID %in% c("C22","C23"))

# Group the country regions and sum all unemployment data for Tipperary overall
df2011TipperaryThemesTotal <- sqldf("Select 'C37' as GEOGID,
                                    'Tipperary' as Tipperary,
                                    sum(Looking_for_Work),
                                    sum(Unemployed),
                                    sum(Total_Workforce)
                                    from df2011TipperaryThemes
                                    group by Tipperary")



# The Waterford areas are combined next...
df2011WaterfordThemes <- df2011CountyThemes %>%
  filter(GEOGID %in% c("C24","C25"))

# Group the country regions and sum all unemployment data for Waterford overall
df2011WaterfordThemesTotal <- sqldf("Select 'C38' as GEOGID,
                                    'Waterford' as Waterford,
                                    sum(Looking_for_Work),
                                    sum(Unemployed),
                                    sum(Total_Workforce)
                                    from df2011WaterfordThemes
                                    group by Waterford")



# The Galway areas are combined next...
df2011GalwayThemes <- df2011CountyThemes %>%
  filter(GEOGID %in% c("C26","C27"))

# Group the country regions and sum all unemployment data for Galway overall
```

```
df2011GalwayThemesTotal <- sqldf("Select 'C39' as GEOGID,
                                  'Galway' as Galway,
                                  sum(Looking_for_Work),
                                  sum(Unemployed),
                                  sum(Total_Workforce)
                                  from df2011GalwayThemes
                                  group by Galway")




# Add Collated County data to revised county theme data
df2011CountyThemes <- data.frame(rbind(as.matrix(df2011CountyThemes), as.matrix(d
f2011DublinThemesTotal)))    # Dublin
df2011CountyThemes <- data.frame(rbind(as.matrix(df2011CountyThemes), as.matrix(d
f2011CorkThemesTotal)))      # Cork
df2011CountyThemes <- data.frame(rbind(as.matrix(df2011CountyThemes), as.matrix(d
f2011LimerickThemesTotal)))  # Limerick
df2011CountyThemes <- data.frame(rbind(as.matrix(df2011CountyThemes), as.matrix(d
f2011TipperaryThemesTotal))) # Tipperary
df2011CountyThemes <- data.frame(rbind(as.matrix(df2011CountyThemes), as.matrix(d
f2011WaterfordThemesTotal))) # Waterford
df2011CountyThemes <- data.frame(rbind(as.matrix(df2011CountyThemes), as.matrix(d
f2011GalwayThemesTotal)))    # Galway
```

```
# ------------------------------------------------------------#
# Additional data manipulation routines to reformat data and add
# calculated fields to 2011 dataframe
# ------------------------------------------------------------#


# Convert county names to upper case to match map dataframe
df2011CountyThemes <- mutate_all(df2011CountyThemes, .funs=toupper)


# Reconvert County unemployment data columns back to numeric
df2011CountyThemes$Looking_for_Work = as.numeric(as.character(df2011CountyThemes$
Looking_for_Work))
df2011CountyThemes$Unemployed = as.numeric(as.character(df2011CountyThemes$Unempl
oyed))
df2011CountyThemes$Total_Workforce = as.numeric(as.character(df2011CountyThemes$T
otal_Workforce))



# Remove the redundant county sub-breakdowns for unemployment data
df2011CountyThemes <- df2011CountyThemes %>%
  filter(!GEOGID %in% c("C02","C03","C04","C05","C17","C18","C20","C21","C22","C2
3","C24","C25","C26","C27")) %>%
  rename(COUNTY = GEOGDESC)



# Calculate Unemployment rate by County and add to Dataframe
df2011CountyThemes$Unemploy_Rate <- ((df2011CountyThemes$Looking_for_Work + df201
1CountyThemes$Unemployed) / df2011CountyThemes$Total_Workforce) * 100



# Using cut() function to create 2011 categorical bands for rates of unemployment
df2011CountyThemes$Unemploy_Pct <- cut(df2011CountyThemes$Unemploy_Rate,
                                       breaks = c(0, 6.99, 9.99, 10.99, 11.99, 12
.99, 13.99, 14.99, 99),
                                       labels = c("<7%", "7-10%", "10-11%", "11-1
2%", "12-13%", "13-14%", "14-15%", "15%+"))



# Join dataframe on county names
dfCountyMap <- left_join(counties, df2011CountyThemes, by = "COUNTY")

# Set up factor in dataframe for visualisation
dfCountyMap$COUNTY <- factor(dfCountyMap$COUNTY)
```

```
#-----------------------------------------------------------------------#
# Generate the 2011 Country Map Plot with Unemployment Rates
#-----------------------------------------------------------------------#

ggplot(dfCountyMap) +
  geom_polygon(colour="black", aes(x=long, y=lat, group=group,
fill=Unemploy_Pct)) +
  labs(x = NULL, y = NULL,
       title = "Unemployment Rate by County (2011) for Ireland",
       subtitle = "Sources: Census 2011",
       caption = "Plot by C.Finegan d21124026") +
  theme(axis.line=element_blank(),
        axis.ticks=element_blank(),
        axis.text=element_blank(),
        axis.title=element_blank(),
        panel.grid = element_blank(),
        plot.caption.position = 'plot',
        plot.title.position = 'plot',
        legend.position = "bottom") +
  labs(fill = "Unemployment Rate (%)") +
  # Colour scale for unemployment rate
  scale_fill_brewer(palette="YlOrRd")
```



4.1.2 2016 Unemployment Map Graph Code

```
# ------------------------------------------------ #
# ------------------------------------------------ #

# Data Visualisation Assignment 2
# Visualisations in R
#
# Student No. d21124026
# Name : Ciaran Finnegan
# TU060 Data Science MSc
#
# December 2022


# Visualisation One - Comparison of Unemployment Rates
# Per County in Ireland in Census 2016 data
# ------------------------------------------------ #
# ------------------------------------------------ #



# load required libraries
library(rgeos)

library(maptools)

library(curl)

library(readr)

library(ggplot2)

library(sqldf)

library(tidyverse)

library(viridis)

library(dplyr)



# There were challenges with accessing data file locations for all datasets.
# To allow the R code to run anywhere the required data files have been hosted
# on GitHub where required.

# Build URL connecting string to download OSI country boundaries data file
# for counties in the Republic of Ireland
sGitHub_Datasource1 <-"https://github.com/JackDaedalus/DataVizLabs/raw/"
sGitHub_Datasource2 <- paste(sGitHub_Datasource1,"dfa3d486a5ea74a588e9768141b35f5
70eff3c57/CA2/", sep = "", collapse=NULL)
sGitHub_Datafile <- "Counties_-_OSi_National_Statutory_Boundaries_-_2019_-_Genera
lised_20m.zip"
sGitHub_Datasource <- paste(sGitHub_Datasource2,sGitHub_Datafile, sep = "", colla
pse=NULL)
```

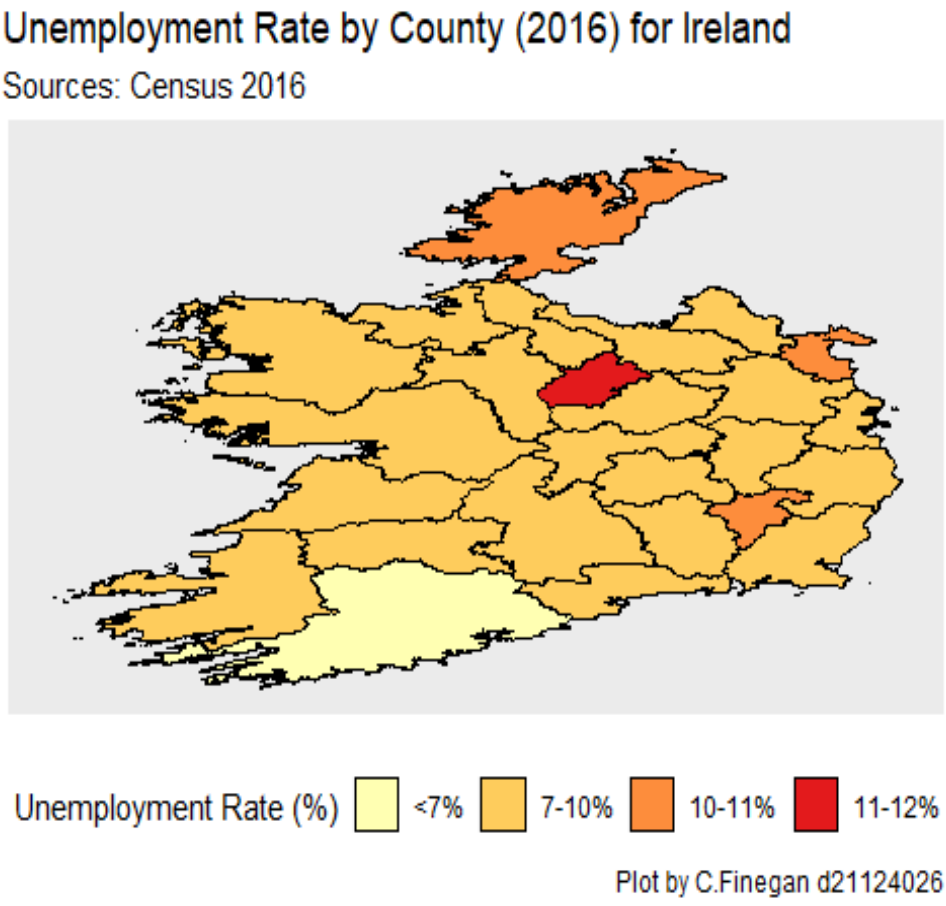
```
county_map_source <- sGitHub_Datasource

# Download zip file from from GitHub and extract 2019 OSI County data for 26 Iris
h counties
temp_1 <- tempfile()
temp_2 <- tempfile()
source <- county_map_source
temp_1 <- curl_download(url = source, destfile = temp_1, quiet = FALSE)
unzip(temp_1, exdir = temp_2)




# Read the shape file extracted from the downloaded zip file
spdf <- readShapePoly(file.path(temp_2,"Counties___OSi_National_Statutory_Boundar
ies___Generalised_20m.shp"))

## Warning: shapelib support is provided by GDAL through the sf and terra package
s
## among others

# Generate a new dataframe that will work with ggplot to generate Ireland county
map
spdf@data$id <- rownames(spdf@data)
spdf.points <- fortify(spdf, region="id")
counties <- inner_join(spdf.points, spdf@data, by="id")







# Build URL connection string to pull 2016 theme data (zip file)
# from GitHub repository
sGitHub_Datasource1_2016 <-"https://github.com/JackDaedalus/DataVizLabs/raw/"
sGitHub_Datasource2_2016 <- paste(sGitHub_Datasource1_2016,"main/CA2/", sep = "",
collapse=NULL)
sGitHub_Datafile_2016 <- "SAPS2016_CTY31.zip"
sGitHub_Datasource_2016 <- paste(sGitHub_Datasource2_2016,sGitHub_Datafile_2016,
sep = "", collapse=NULL)

f2016CTY_data <- sGitHub_Datasource_2016

# Download zip file from from GitHub and extract 2016 theme data for 26 Irish cou
nties
temp_3 <- tempfile()
temp_4 <- tempfile()
source <- f2016CTY_data
temp_3 <- curl_download(url = source, destfile = temp_3, quiet = FALSE)
unzip(temp_3, exdir = temp_4)


# ------------------------------------------------------#
#  Load Census Theme Data for 2016 for Irish counties
```

```
# ------------------------------------------------------#

# Select only the required unemployment data
# Rename the columns to increase understanding of the data
# Read the downloaded CSV file
df2016CountyThemes <- read_delim(file.path(temp_4,"SAPS2016_CTY31.csv"),show_col_
types = FALSE) %>%
  select(GEOGID, GEOGDESC, T8_1_LFFJT, T8_1_ULGUPJT, T8_1_TT) %>%
  rename(Looking_for_Work = T8_1_LFFJT, Unemployed = T8_1_ULGUPJT, Total_Workforc
e = T8_1_TT)



# ------------------------------------------------------------#
# The Census theme data breaks down the counties in certain cases
# for metropolitan areas. This data needs to be re-merged to match
# the county boundaries in the OSI dataframe
#
# The 2016 division of metropolitan areas is slightly different to
# the categorization in the 2011 census
#
# ------------------------------------------------------------#

# Start with Dublin...
df2016DublinThemes <- df2016CountyThemes %>%
  filter(GEOGID %in% c("CTY31_DC","CTY31_DR","CTY31_FL","CTY31_SD"))



# Group the country regions and sum all unemployment data for Dublin overall
df2016DublinThemesTotal <- sqldf("Select 'CTY32_DC' as GEOGID,
                                  'Dublin' as Dublin,
                                  sum(Looking_for_Work),
                                  sum(Unemployed),
                                  sum(Total_Workforce)
                                  from df2016DublinThemes
                                  group by Dublin")



# The Cork areas are combined next...
df2016CorkThemes <- df2016CountyThemes %>%
  filter(GEOGID %in% c("CTY31_CC","CTY31_CK"))

# Group the country regions and sum all unemployment data for Dublin overall
df2016CorkThemesTotal <- sqldf("Select 'CTY33_CC' as GEOGID,
                                  'Cork' as Cork,
                                  sum(Looking_for_Work),
                                  sum(Unemployed),
                                  sum(Total_Workforce)
                                  from df2016CorkThemes
                                  group by Cork")
```

```
# The Galway areas are combined next...
df2016GalwayThemes <- df2016CountyThemes %>%
  filter(GEOGID %in% c("CTY31_GC","CTY31_GY"))

# Group the country regions and sum all unemployment data for Galway overall
df2016GalwayThemesTotal <- sqldf("Select 'CTY34_GC' as GEOGID,
                                  'Galway' as Galway,
                                  sum(Looking_for_Work),
                                  sum(Unemployed),
                                  sum(Total_Workforce)
                                  from df2016GalwayThemes
                                  group by Galway")



# Add Collated County data to revised county theme data
df2016CountyThemes <- data.frame(rbind(as.matrix(df2016CountyThemes), as.matrix(d
f2016DublinThemesTotal)))    # Dublin
df2016CountyThemes <- data.frame(rbind(as.matrix(df2016CountyThemes), as.matrix(d
f2016CorkThemesTotal)))      # Cork
df2016CountyThemes <- data.frame(rbind(as.matrix(df2016CountyThemes), as.matrix(d
f2016GalwayThemesTotal)))    # Galway



# ------------------------------------------------------------#
# Additional data manipulation routines to reformat data and add
# calculated fields to 2016 dataframe
# ------------------------------------------------------------#

# Reconvert County unemployment data columns back to numeric
df2016CountyThemes$Looking_for_Work = as.numeric(as.character(df2016CountyThemes$
Looking_for_Work))
df2016CountyThemes$Unemployed = as.numeric(as.character(df2016CountyThemes$Unempl
oyed))
df2016CountyThemes$Total_Workforce = as.numeric(as.character(df2016CountyThemes$T
otal_Workforce))



# Remove the redundant county sub-breakdowns for unemployment data
df2016CountyThemes <- df2016CountyThemes %>%
  filter(!GEOGID %in% c("CTY31_CC","CTY31_CK","CTY31_DC","CTY31_DR","CTY31_FL","C
TY31_SD","CTY31_GC","CTY31_GY")) %>%
  rename(COUNTY = GEOGDESC)



# Calculate Unemployment rate by County and add to Dataframe
df2016CountyThemes$Unemploy_Rate <- ((df2016CountyThemes$Looking_for_Work + df201
```

```
6CountyThemes$Unemployed) / df2016CountyThemes$Total_Workforce) * 100


# Convert county names to upper case to match map dataframe
df2016CountyThemes <- mutate_all(df2016CountyThemes, .funs=toupper)


# Reconvert County unemployment data columns back to numeric
df2016CountyThemes$Looking_for_Work = as.numeric(as.character(df2016CountyThemes$
Looking_for_Work))
df2016CountyThemes$Unemployed = as.numeric(as.character(df2016CountyThemes$Unempl
oyed))
df2016CountyThemes$Total_Workforce = as.numeric(as.character(df2016CountyThemes$T
otal_Workforce))
df2016CountyThemes$Unemploy_Rate = as.numeric(as.character(df2016CountyThemes$Une
mploy_Rate))



# Using cut() function to create 2016 categorical bands for rates of unemployment
df2016CountyThemes$Unemploy_Pct <- cut(df2016CountyThemes$Unemploy_Rate,
                                       breaks = c(0, 6.99, 9.99, 10.99, 11.99, 12
.99, 13.99, 14.99, 99),
                                       labels = c("<7%", "7-10%", "10-11%", "11-1
2%", "12-13%", "13-14%", "14-15%", "15%+"))



# Change name for Limerick and Waterford to allow dataframe sot join on County Na
me
#myDataFrame["rowName", "columnName"] <- value
df2016CountyThemes[10, "COUNTY"] <- "LIMERICK"
df2016CountyThemes[20, "COUNTY"] <- "WATERFORD"


# Join dataframe on county names
dfCountyMap <- left_join(counties, df2016CountyThemes, by = "COUNTY")

# Set up factor in dataframe for visualisation
dfCountyMap$COUNTY <- factor(dfCountyMap$COUNTY)
```

```
#-------------------------------------------------------------------------------#
# Generate the 2016 Country Map Plot with Unemployment Rates
#-------------------------------------------------------------------------------#

ggplot(dfCountyMap) +
  geom_polygon(colour="black", aes(x=long, y=lat, group=group, fill=Unemploy_Pct)
) +
  labs(x = NULL, y = NULL,
       title = "Unemployment Rate by County (2016) for Ireland",
       subtitle = "Sources: Census 2016",
       caption = "Plot by C.Finegan d21124026") +
  theme(axis.line=element_blank(),
        axis.ticks=element_blank(),
        axis.text=element_blank(),
        axis.title=element_blank(),
        panel.grid = element_blank(),
        plot.caption.position = 'plot',
        plot.title.position = 'plot',
        legend.position = "bottom")+
  labs(fill = "Unemployment Rate (%)") +
  # Colour scale for unemployment rate
  scale_fill_brewer(palette="YlOrRd")
```



4.2 Appendix 2 – R Source Code – Visualisation Two

```
# ------------------------------------------------ #
# ------------------------------------------------ #

# Data Visualisation Assignment 2
# Visualisations in R
#
# Student No. d21124026
# Name : Ciaran Finnegan
# TU060 Data Science MSc
#
# December 2022


# Visualisation Two - Socio-economic group comparison
# Census 2011 v Census 2016
# ------------------------------------------------ #
# ------------------------------------------------ #


# load required libraries
library(curl)

library(readr)

library(ggplot2)

library(sqldf)

library(tidyverse)

library(viridis)

library(dplyr)


library(scales)
```

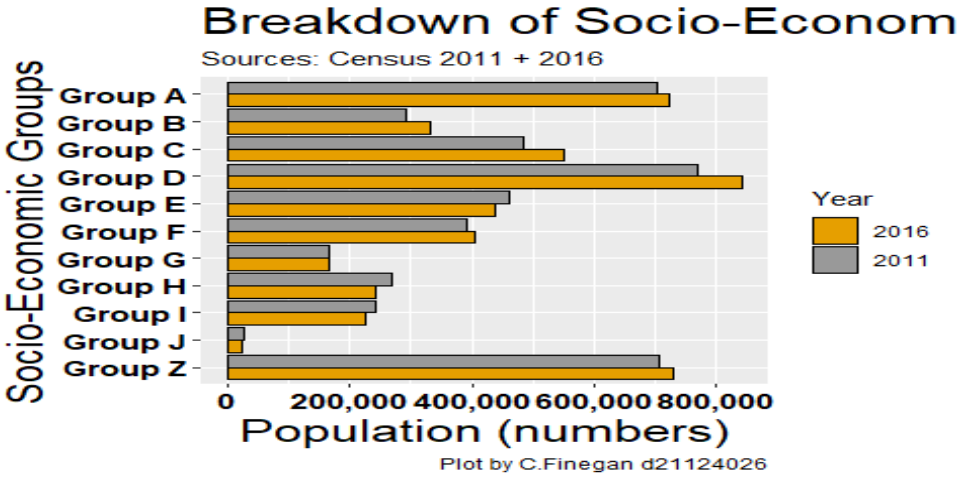
```
# ------------------------------------------------- #
# Download Census Theme Data for 2016 for Irish counties
# ------------------------------------------------- #

# Prepare URL string with location on GitHub of ZIP file with Census 2016 'Theme'
data
sGitHub_Datasource1_2016 <-"https://github.com/JackDaedalus/DataVizLabs/raw/"
sGitHub_Datasource2_2016 <- paste(sGitHub_Datasource1_2016,"main/CA2/", sep = "",
collapse=NULL)
sGitHub_Datafile_2016 <- "SAPS2016_CTY31.zip"
sGitHub_Datasource_2016 <- paste(sGitHub_Datasource2_2016,sGitHub_Datafile_2016,
sep = "", collapse=NULL)
f2016CTY_data <- sGitHub_Datasource_2016


# Download zip file from from GitHub and extract 2016 Theme data for Irish counti
es
temp_3 <- tempfile()
temp_4 <- tempfile()
source <- f2016CTY_data
temp_3 <- curl_download(url = source, destfile = temp_3, quiet = FALSE)
unzip(temp_3, exdir = temp_4)

# Prepare location string of downloaded 2016 Census data CSV file
f2016CensusData <- "\\SAPS2016_CTY31.csv"
f2016CensusData <- paste(temp_4,f2016CensusData, sep = "", collapse=NULL)



# ------------------------------------------------- #
# Prepare URL for download of Census Theme Data for 2011
# for Irish counties
# ------------------------------------------------- #

# Prepare URL string for Census Theme Data for 2011 for Irish counties
f2011CensusData <- "https://www.cso.ie/en/media/csoie/census/documents/saps2011fi
les/AllThemesTablesCTY.csv"


# -------------------------------------------------#
# Prepare sequence of Census data to be downloaded and
# read into dataframes for processing
# -------------------------------------------------#

# Set up array of Census file names to be loaded in sequence
arrCensusThemeFiles <- c(f2011CensusData, f2016CensusData)
arrCensusThemeYears <- c('2011','2016') # 2011 data is downloaded first for manip
ulation



# Set up dataframe array to hold the 2011 and 2016 Census Theme data
# Both sets of data will undergo the same transformation before
# being merged in advance of graph generation
```

```
arrDFYrCountySocioThemes          <- list()  # start with empty array for data lo
aded from files
arrDFYrCountySocioThemes_Reshaped <- list()  # array to store dataframes after th
e data wrangling process



# ----------------------------------------------------#
# Iterate through the 2011 and 2016 files and manipulate
# the socio-economic data for visualisation
# ----------------------------------------------------#

for (i in 1:(length(arrCensusThemeFiles))) {

  # Select only the required Socio-economic data
  # Rename the columns to increase understanding of the data
  arrDFYrCountySocioThemes[[i]] <- read_delim(arrCensusThemeFiles[i],
                                      show_col_types = FALSE) %>%   # Read Cens
us data
    select(GEOGID, GEOGDESC, # Only select the county identifier and the numbers
of people in each
           T9_2_PA,         # Socio-economic group
           T9_2_PB,
           T9_2_PC,
           T9_2_PD,
           T9_2_PE,
           T9_2_PF,
           T9_2_PG,
           T9_2_PH,
           T9_2_PI,
           T9_2_PJ,
           T9_2_PZ) %>%
    rename(GroupA = T9_2_PA,  # Rename Columns to improve readability
           GroupB = T9_2_PB,
           GroupC = T9_2_PC,
           GroupD = T9_2_PD,
           GroupE = T9_2_PE,
           GroupF = T9_2_PF,
           GroupG = T9_2_PG,
           GroupH = T9_2_PH,
           GroupI = T9_2_PI,
           GroupJ = T9_2_PJ,
           GroupZ = T9_2_PZ)


  # Create dataframe for year in array - 2011 or 2016
  dfThisYrCountySocioThemes <- arrDFYrCountySocioThemes[[i]]

  # Add Year Value as label to dataframe
  dfThisYrCountySocioThemes$Year <- arrCensusThemeYears[i]


  # Sum County Data into a single row
  # Group the counties and sum all socio-economic data for Ireland overall
```

```
  dfThisYrCtyThemesSocioTotals <- sqldf("Select 'CTT' as GEOGID,
                                        Year,
                                        sum(GroupA) as GrpA,
                                        sum(GroupB) as GrpB,
                                        sum(GroupC) as GrpC,
                                        sum(GroupD) as GrpD,
                                        sum(GroupE) as GrpE,
                                        sum(GroupF) as GrpF,
                                        sum(GroupG) as GrpG,
                                        sum(GroupH) as GrpH,
                                        sum(GroupI) as GrpI,
                                        sum(GroupJ) as GrpJ,
                                        sum(GroupZ) as GrpZ
                                        from dfThisYrCountySocioThemes
                                        group by Year")


  # Pivot County Data for numbers in each Socio-economic Group
  dfThisYrCtyThemesSocioTotals_Reshape <- dfThisYrCtyThemesSocioTotals %>%
    pivot_longer(c(GrpA,
                   GrpB,
                   GrpC,
                   GrpD,
                   GrpE,
                   GrpF,
                   GrpG,
                   GrpH,
                   GrpI,
                   GrpJ,
                   GrpZ), # values to pivot or reshape
                 names_to = "SocioEcon_Group", # Rename column for Social Class G
roup
                 values_to = "Numbers_in_Group") # Re-name column containing pop.
numbers


  #Sort x-axis variable in alphabetical order for each table - top down from Grou
p A to Z
  level_order <- c('GrpZ',
                   'GrpJ',
                   'GrpI',
                   'GrpH',
                   'GrpG',
                   'GrpF',
                   'GrpE',
                   'GrpD',
                   'GrpC',
                   'GrpB',
                   'GrpA')


  # Set up dataframe in array after data manipulation complete
  arrDFYrCountySocioThemes_Reshaped[[i]] <- dfThisYrCtyThemesSocioTotals_Reshape
```

```
}



# ------------------------------------------------#
# Merge the 2011 and 2016 dataframes and plot the
# comparisons in a horizontal bar chart
# ------------------------------------------------#

# Merge dataframes
dfFinal2011_2016SocEconCensus <- merge(arrDFYrCountySocioThemes_Reshaped[[1]], ar
rDFYrCountySocioThemes_Reshaped[[2]],all=TRUE)


# Set up legend so that '2016' is on top
dfFinal2011_2016SocEconCensus$Year <- factor(dfFinal2011_2016SocEconCensus$Year,
levels = c("2016", "2011"))
```

```
# ------------------------------------------------------------------#
# Generate Horizontal Bar Chart
# Contains bars for each Social Class
# Grouped by year
# ------------------------------------------------------------------#

gg1 <- ggplot(data=dfFinal2011_2016SocEconCensus, aes(x = factor(SocioEcon_Group,
level = level_order),
                                                      y=Numbers_in_Group, fi
ll=Year)) +
  geom_bar(stat="identity", position=position_dodge(), colour="black") +
  labs(x = "Socio-Economic Groups", y = "Population (numbers)",
       title = "Breakdown of Socio-Economic Groups (2011 v 2016) for Ireland",
       subtitle = "Sources: Census 2011 + 2016",
       caption = "Plot by C.Finegan d21124026") +
  theme(legend.position = "right",
        plot.title = element_text(size = 22),
        axis.title = element_text(size = 19),
        axis.text.y = element_text(size=12, face="bold", colour = "black"),
        axis.text.x = element_text(size=12, face="bold", colour = "black")) +
  # Rename legend
  scale_fill_discrete(labels=c('2016', '2011'),name = "Year") +
  # Make population axis more readable
  scale_y_continuous(labels = comma) +
  # Tidy up axis descriptions of socio-economic groups
  scale_x_discrete(labels = c("Group Z","Group J","Group I","Group H",
      "Group G","Group F","Group E","Group D","Group C","Group B","Group A")) +
  coord_flip() +
  scale_fill_manual(values=c("#E69F00","#999999"))

## Scale for fill is already present.
## Adding another scale for fill, which will replace the existing scale.

print(gg1)
```



4.3 Appendix 3 – R Source Code – Visualisation Three

```
# ------------------------------------------------ #
# Data Visualisation Assignment 2
# Visualisations in R
#
# Student No. d21124026
# Name : Ciaran Finnegan
# TU060 Data Science MSc
#
# December 2022

# Visualisation Three - Comparison of population
# health against Socio-economic groups in Census 2011
# and Census 2016 data for Irish counties
# ------------------------------------------------ #

# Additional packages to improve information display
# on scatter plot graphs
#install.packages("gghighlight")
#install.packages("ggthemes")

# load required libraries
library(curl)

library(readr)

library(ggplot2)

library(sqldf)

library(tidyverse)

library(viridis)

library(dplyr)
library(scales)

library(gghighlight)

library(ggthemes)

library(stringr)

# ------------------------------------------------ #
# Download Census Theme Data for 2016 for Irish counties
# ------------------------------------------------ #

# Prepare URL string with location on GitHub of ZIP file with Census 2016 'Theme'
data
sGitHub_Datasource1_2016 <-"https://github.com/JackDaedalus/DataVizLabs/raw/"
sGitHub_Datasource2_2016 <- paste(sGitHub_Datasource1_2016,"main/CA2/", sep = "",
collapse=NULL)
sGitHub_Datafile_2016 <- "SAPS2016_CTY31.zip"
sGitHub_Datasource_2016 <- paste(sGitHub_Datasource2_2016,sGitHub_Datafile_2016,
sep = "", collapse=NULL)
```

```
f2016CTY_data <- sGitHub_Datasource_2016


# Download zip file from from GitHub and extract 2016 Theme data for Irish counti
es
temp_3 <- tempfile()
temp_4 <- tempfile()
source <- f2016CTY_data
temp_3 <- curl_download(url = source, destfile = temp_3, quiet = FALSE)
unzip(temp_3, exdir = temp_4)

# Prepare location string of downloaded 2016 Census data CSV file
f2016CensusData <- "\\SAPS2016_CTY31.csv"
f2016CensusData <- paste(temp_4,f2016CensusData, sep = "", collapse=NULL)



# ------------------------------------------------- #
# Prepare URL for download of Census Theme Data for 2011
# for Irish counties
# ------------------------------------------------- #

# Prepare URL string for Census Theme Data for 2011 for Irish counties
f2011CensusData <- "https://www.cso.ie/en/media/csoie/census/documents/saps2011fi
les/AllThemesTablesCTY.csv"
```

```
# ------------------------------------------------#
# Prepare sequence of Census data to be downloaded and
# read into dataframes for processing
# ------------------------------------------------#

# Set up array of Census file names to be loaded in sequence
arrCensusThemeFiles <- c(f2011CensusData, f2016CensusData)
arrCensusThemeYears <- c('2011','2016') # 2011 data is downloaded first for manip
ulation



# Set up dataframe array to hold the 2011 and 2016 Census Theme data
# Both sets of data will undergo the same transformation before
# being merged in advance of graph generation
arrDFYrCountySocioThemes          <- list()  # start with empty array for data
                                             # loaded from files
arrDFYrCountySocioThemes_Modified <- list()  # array to store dataframes after
                                             # the data wrangling process
dfCensusGraph <- list()


# ------------------------------------------------#
# Iterate through the 2011 and 2016 files and manipulate
# the socio-economic data for visualisation
# ------------------------------------------------#
#i <- 1
for (i in 1:(length(arrCensusThemeFiles))) {


  if (arrCensusThemeYears[i] == '2011') { # The 2011 Census Theme columns for
                                          # Health data have a different format

    # Select only the required Socio-economic and Health data
    # Rename the columns to increase understanding of the data

    # Read Census 2011 data from CSO website
    arrDFYrCountySocioThemes[[i]] <- read_delim(arrCensusThemeFiles[i],
                                          show_col_types = FALSE) %>%

      select(GEOGID, GEOGDESC, # Select the county identifier and the following..
.
             T9_2_PA,          # Socio-economic Group A
             T9_2_PB,          # Socio-economic Group B
             T9_2_PT,          # Total population in Socio-economic Groups
             T12_3VGT,         # Population in 'Very Good Health'
             T12_3GT,          # Population in 'Good Health'
             T12_3TT           # Total Population in health census
                    ) %>%
      rename(GroupA         = T9_2_PA,  # Rename Columns to improve readability
             GroupB         = T9_2_PB,
             GroupsTotal    = T9_2_PT,
             VeryGoodHealth = T12_3VGT,
             GoodHealth     = T12_3GT,
```

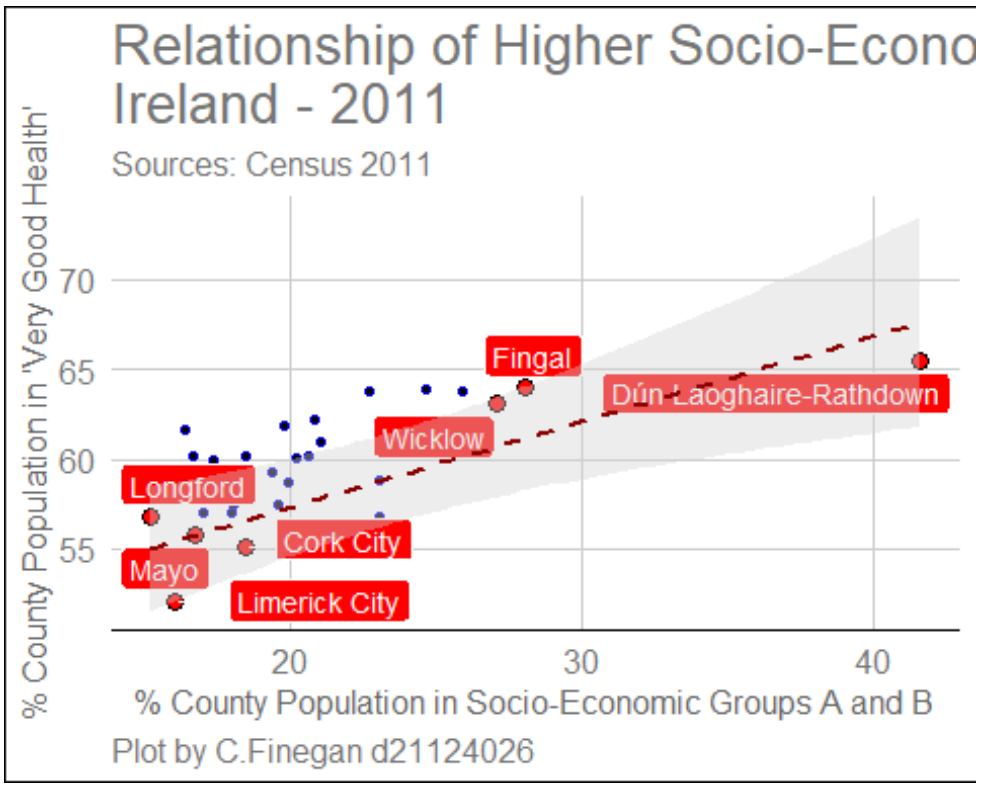
```
            HealthTotal    = T12_3TT)



  } else {

    # Read 2016 file - formats of Theme 12 column headings are different than tho
se in 2011
    arrDFYrCountySocioThemes[[i]] <- read_delim(arrCensusThemeFiles[i],
                                                show_col_types = FALSE) %>%
                                  # Read Census 2011 data from CSO website
      select(GEOGID, GEOGDESC, # Select the county identifier and the following
             T9_2_PA,          # Socio-economic Group A
             T9_2_PB,          # Socio-economic Group B
             T9_2_PT,          # Total population in Socio-economic Groups
             T12_3_VGT,        # Population in 'Very Good Health'
             T12_3_GT,         # Population in 'Good Health'
             T12_3_TT          # Total Population in health census
      ) %>%
      rename(GroupA         = T9_2_PA,  # Rename Columns to improve readability
             GroupB         = T9_2_PB,
             GroupsTotal    = T9_2_PT,
             VeryGoodHealth = T12_3_VGT,
             GoodHealth     = T12_3_GT,
             HealthTotal    = T12_3_TT)


    # Correct for Irish language fada in Dun-Laoighaire throwing a text error
    arrDFYrCountySocioThemes[[i]][8, "GEOGDESC"] <- "Dun Laoghaire-Rathdown"

  }
```

```
# ------------------------------------------------------------#
  # Data manipulation routines to add calculated field for
  # each county for percentage of population in Social
  # Class A ad B, and in 'Very God Health'
  # ------------------------------------------------------------#

  # Create dataframe for year in array - 2011 or 2016
  dfThisYrCountySocioThemes <- arrDFYrCountySocioThemes[[i]]

  # Add Year Value as label to dataframe
  dfThisYrCountySocioThemes$Year <- arrCensusThemeYears[i]


  # Calculate Ratio by County of Population in Socio-economic groups A and B.
  dfThisYrCountySocioThemes$GroupAB_Pct  <- ((dfThisYrCountySocioThemes$GroupA
                                              + dfThisYrCountySocioThemes$Group
B)
                                             / dfThisYrCountySocioThemes$Groups
Total) * 100


  # Calculate Ratio by County of Population in 'Very Good' health.
  dfThisYrCountySocioThemes$VGHealth_Pct  <- (dfThisYrCountySocioThemes$VeryGoodH
ealth
                                            / dfThisYrCountySocioThemes$HealthTo
tal) * 100



  # Set up dataframe in array after data manipulation complete
  arrDFYrCountySocioThemes_Modified[[i]] <- dfThisYrCountySocioThemes
```

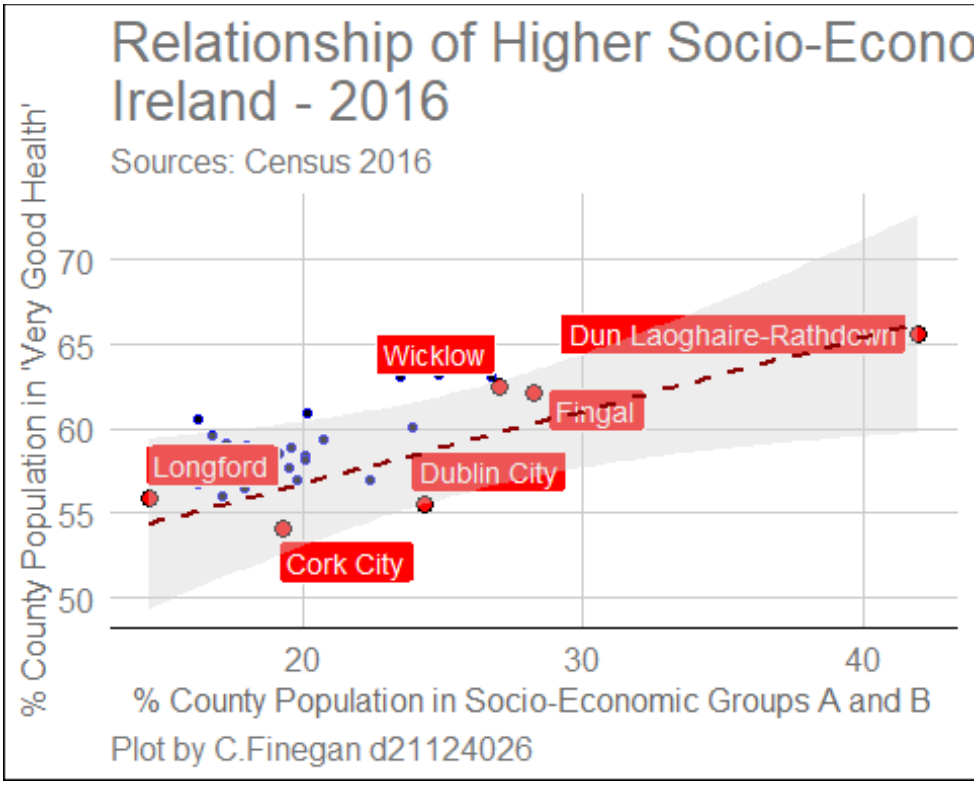
```
# ------------------------------------------------------------------#
  # Prepare elements to generate and improve Scatter Plot
  # rendering of relationship between Social Class A/B and
  # 'Very Good Health'
  # ------------------------------------------------------------------#

  # Copy data to a dataframe with a shorter name to simplify
  # the code generating the graphs
  dfCensusGraph[[i]] <- arrDFYrCountySocioThemes_Modified[[i]]

  # Format graph titles - avoids distortion when rendering the graph
  sGraphTitle <- "Relationship of Higher Socio-Economic Groups to Health in Irela
nd - "
  sGraphSubTitle <- "Sources: Census "
  # Add 'Year' for given dataframe in this loop
  sGraphTitle <- paste(sGraphTitle, arrCensusThemeYears[i],
                       sep = "", collapse=NULL)
  sGraphSubTitle <- paste(sGraphSubTitle, arrCensusThemeYears[i],
                          sep = "", collapse=NULL)


  # These parameters work best for highlighting specific data points
  # for county regions in the 2011/2016 Scatterplot graphs
  UpperGroupAB_Pct  <- 27
  LowerGroupAB_Pct  <- 16
  LowerVGHealth_Pct <- 56


  # Generate Scatter Plot Grah
  gg1 <- ggplot(data=dfCensusGraph[[i]], aes(x = GroupAB_Pct, y=VGHealth_Pct)) +
    geom_point(colour="black", size=3, shape=21, fill="red") +
    # Highlight on Scatter plot the county areas at the upper and lower end of
    # the socio-economic groups and health rating.
    gghighlight(((GroupAB_Pct>=UpperGroupAB_Pct|GroupAB_Pct<=LowerGroupAB_Pct)
                 | VGHealth_Pct<=LowerVGHealth_Pct),
                label_key = GEOGDESC,
                unhighlighted_params
                = list(colour = "blue",
                       fill="black",
                       size=1.5),
                label_params = list(fill="red",
                                    colour="white",
                                    size=3.75))+
    # Label the graph
    labs(x = "% County Population in Socio-Economic Groups A and B",
         y = "% County Population in 'Very Good Health'",
         # Format title to avoid distortion of the graph
         title = str_wrap(sGraphTitle, 60),
         subtitle = sGraphSubTitle,
         caption = "Plot by C.Finegan d21124026") +
    # Add a dashed regression line and a confidence interval for relationship
    geom_smooth(method='lm', se=TRUE, linetype="dashed",
                color="darkred", fullrange=TRUE, fill="light grey") +
    # Use GDocs theme from ggplot themes library
```

```
    theme_gdocs()

  print(gg1)

}
```

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




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


Relationship of Higher Socio-Econo
Ireland - 2016
Sources: Census 2016
% County Population in 'Very Good Health'
70
65
60
55
50
Wicklow
Dun Laoghaire-Rathdown
Fingal
Longford
Dublin City
Cork City
20
30
40
% County Population in Socio-Economic Groups A and B
Plot by C.Finegan d21124026