DataViz2

D21124026

2022-12-09

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

# -------------------------------------------------- #  
# -------------------------------------------------- #  
  
# 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)

## Warning: package 'curl' was built under R version 4.2.2

## Using libcurl 7.64.1 with Schannel

library(readr)

## Warning: package 'readr' was built under R version 4.2.2

##   
## Attaching package: 'readr'

## The following object is masked from 'package:curl':  
##   
## parse\_date

library(ggplot2)

## Warning: package 'ggplot2' was built under R version 4.2.2

library(sqldf)

## Warning: package 'sqldf' was built under R version 4.2.2

## Loading required package: gsubfn

## Warning: package 'gsubfn' was built under R version 4.2.2

## Loading required package: proto

## Warning: package 'proto' was built under R version 4.2.2

## Loading required package: RSQLite

## Warning: package 'RSQLite' was built under R version 4.2.2

library(tidyverse)

## Warning: package 'tidyverse' was built under R version 4.2.2

## ── Attaching packages  
## ───────────────────────────────────────  
## tidyverse 1.3.2 ──

## ✔ tibble 3.1.8 ✔ dplyr 1.0.10  
## ✔ tidyr 1.2.1 ✔ stringr 1.4.1   
## ✔ purrr 0.3.5 ✔ forcats 0.5.2

## Warning: package 'tibble' was built under R version 4.2.2

## Warning: package 'tidyr' was built under R version 4.2.2

## Warning: package 'purrr' was built under R version 4.2.2

## Warning: package 'dplyr' was built under R version 4.2.2

## Warning: package 'stringr' was built under R version 4.2.2

## Warning: package 'forcats' was built under R version 4.2.2

## ── Conflicts ────────────────────────────────────────── tidyverse\_conflicts() ──  
## ✖ dplyr::filter() masks stats::filter()  
## ✖ dplyr::lag() masks stats::lag()  
## ✖ readr::parse\_date() masks curl::parse\_date()

library(viridis)

## Warning: package 'viridis' was built under R version 4.2.2

## Loading required package: viridisLite

## Warning: package 'viridisLite' was built under R version 4.2.2

library(dplyr)  
library(scales)

## Warning: package 'scales' was built under R version 4.2.2

##   
## Attaching package: 'scales'  
##   
## The following object is masked from 'package:viridis':  
##   
## viridis\_pal  
##   
## The following object is masked from 'package:purrr':  
##   
## discard  
##   
## The following object is masked from 'package:readr':  
##   
## col\_factor

# ---------------------------------------------------- #  
# 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 counties  
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/saps2011files/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 manipulation  
  
  
  
# 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\_Reshaped <- list() # array to store dataframes after the 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 Census 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 Group  
 values\_to = "Numbers\_in\_Group") # Re-name column containing pop. numbers  
  
   
 #Sort x-axis variable in alphabetical order for each table - top down from Group 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]], arrDFYrCountySocioThemes\_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, fill=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)

