DataViz1b

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 One - Comparison of Unemployment Rates   
# Per County in Ireland in Census 2016 data   
# -------------------------------------------------- #  
# -------------------------------------------------- #  
  
  
  
# load required libraries  
library(rgeos)

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

## Loading required package: sp

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

## rgeos version: 0.5-9, (SVN revision 684)  
## GEOS runtime version: 3.9.3-CAPI-1.14.3   
## Please note that rgeos will be retired by the end of 2023,  
## plan transition to sf functions using GEOS at your earliest convenience.  
## GEOS using OverlayNG  
## Linking to sp version: 1.5-1   
## Polygon checking: TRUE

library(maptools)

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

## Checking rgeos availability: TRUE  
## Please note that 'maptools' will be retired during 2023,  
## plan transition at your earliest convenience;  
## some functionality will be moved to 'sp'.

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)  
  
  
  
# 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"))

## Warning: shapelib support is provided by GDAL through the sf and terra packages  
## 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 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)  
  
  
# ----------------------------------------------------------#  
# 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\_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  
#  
# 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"))  
  
head(df2016DublinThemes)

## # A tibble: 4 × 5  
## GEOGID GEOGDESC Looking\_for\_Work Unemployed Total\_Workf…¹  
## <chr> <chr> <dbl> <dbl> <dbl>  
## 1 CTY31\_DC "Dublin City" 4686 34514 471341  
## 2 CTY31\_DR "D\xfan Laoghaire-Rathdown" 927 6789 177979  
## 3 CTY31\_FL "Fingal" 1850 13565 223407  
## 4 CTY31\_SD "South Dublin" 2030 16235 214679  
## # … with abbreviated variable name ¹​Total\_Workforce

# 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(df2016DublinThemesTotal))) # Dublin  
df2016CountyThemes <- data.frame(rbind(as.matrix(df2016CountyThemes), as.matrix(df2016CorkThemesTotal))) # Cork  
df2016CountyThemes <- data.frame(rbind(as.matrix(df2016CountyThemes), as.matrix(df2016GalwayThemesTotal))) # 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$Unemployed = as.numeric(as.character(df2016CountyThemes$Unemployed))  
df2016CountyThemes$Total\_Workforce = as.numeric(as.character(df2016CountyThemes$Total\_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","CTY31\_SD","CTY31\_GC","CTY31\_GY")) %>%  
 rename(COUNTY = GEOGDESC)  
  
  
  
# Calculate Unemployment rate by County and add to Dataframe  
df2016CountyThemes$Unemploy\_Rate <- ((df2016CountyThemes$Looking\_for\_Work + df2016CountyThemes$Unemployed) / df2016CountyThemes$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$Unemployed = as.numeric(as.character(df2016CountyThemes$Unemployed))  
df2016CountyThemes$Total\_Workforce = as.numeric(as.character(df2016CountyThemes$Total\_Workforce))  
df2016CountyThemes$Unemploy\_Rate = as.numeric(as.character(df2016CountyThemes$Unemploy\_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%", "11-12%", "12-13%", "13-14%", "14-15%", "15%+"))  
  
  
  
# Change name for Limerick and Waterford to allow dataframe sot join on County Name  
#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 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 (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")

