Assignment 3

Caolan McDonagh

21/02/2022

Data Visualisation - Assignment 3

by Caolan McDonagh

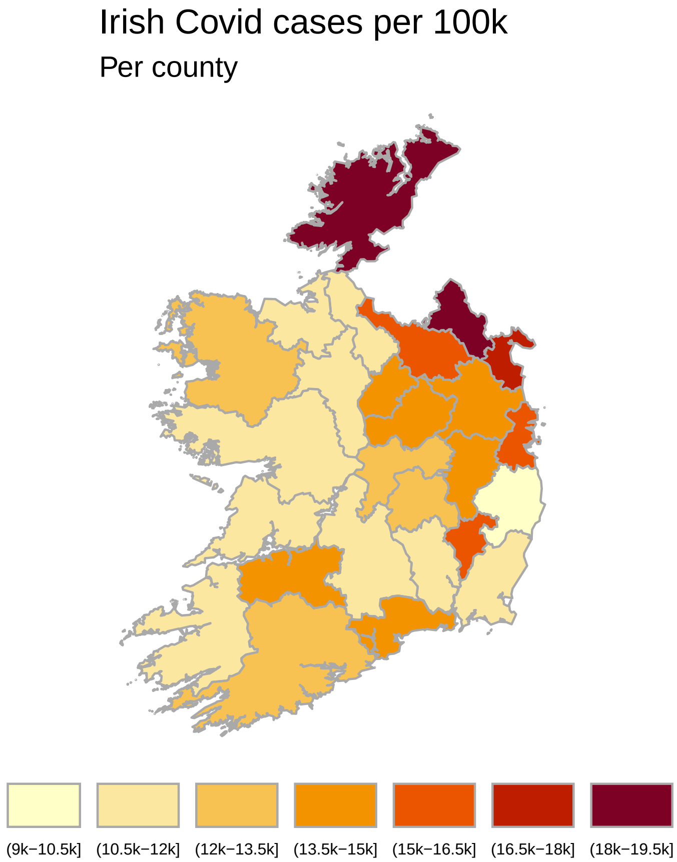
*Skeleton r code adapted from week 6/7 exercise PDFs

A visualisation that allows the reader to accurately read and compare the number of cases per 100,000 of population per county on the 21 December 2021.

The below uses a choropleth visualization to display the cases per 100k for each county on 21/12/2021. I chose this as the static variable for date allows us to accurately show the cases per county on a map of Ireland. This makes it much easier for a reader to compare and contrast counties to one another. The per 100k normalization also allows us to compare them on a level scale, such as Dublin (over one million population) to Donegal (over 150k). Looking at the choropleth we can see Donegal actually was much more infectious than Dublin, even though populations were vastly different. Each coloured county is attached to the case scale, using intervals of 1500, reaching both ends of the scale in enough (not too many or too few) break points to cause differentiation between colour/counties to become too difficult if too many, or too meaningless if too few. Keeping the plot both quick to analyse and accurate at the same time.

```r
library(tinytex)
library(ggplot2)
library(sf)
library(plyr)
library(dplyr)
library(colorspace)
library(RColorBrewer)
library(patchwork)
library(ggridges)
library(lubridate)
library(ggrepel)
library(knitr)
library(tidyr)
library(kableExtra)
library(scales)
library(colorblindr)
library(e1071)
library(lubridate)
library(plyr)
options(scipen=999) #Get rid of scientific notation for large numbers.

#Read out shapefile.
```

```r
shapeFile <- "./CovidCountyStatisticsIreland_v2.shp"

#Read data to a usable df, rather than constantly trying to read the 200mb+ shapefile.
Ireland_Counties <- st_read(shapeFile, quiet = TRUE)

#A visualisation that allows the reader to accurately read and compare the number
#of cases per 100,000 of population per county on the 21 December 2021.

#Create Per 100k column on given date (21st December 2021)
per100kResult <- Ireland_Counties %>% filter(TimeStamp == "2021-12-21") %>%
  mutate(infectionsPer100k = round(((ConfirmedC/Population)*100000),1))

#Use scale and breaks/labs to give a more accurate plot for the reader.
scale_minimum<-round_any(min(per100kResult$infectionsPer100k), 1000, f = floor)
scale_maximum<- round_any(max(per100kResult$infectionsPer100k), 1000, f = ceiling)

#Scaling our breaks
breaks<-seq(scale_minimum-1000,scale_maximum+1000, by =1500)

per100kResult$infectionsPer100k <- cut(per100kResult$infectionsPer100k,
                                       breaks = breaks, dig.lab = 3)

nlevels<- nlevels(per100kResult$infectionsPer100k)

pal <- hcl.colors(nlevels, "YlOrRd", rev = TRUE)

labs <- breaks/1000
labs_plot <- paste0("(", labs[1:nlevels], "k-", labs[1:nlevels+1], "k]")

ggplot(per100kResult) +
  geom_sf(aes(fill = infectionsPer100k),
          color = "darkgrey",
          linetype = 1,
          lwd = 0.4) +

  labs(title = "Irish Covid cases per 100k",
       subtitle = "Per county",) +

  scale_fill_manual(values = pal,
                    drop = FALSE,
                    na.value = "grey80",
                    label = labs_plot,

                    guide = guide_legend(direction = "horizontal", nrow = 1,
                                         label.position = "bottom")) +

  theme_void() +
  theme(legend.title = element_blank(),
        legend.text = element_text(size=5.5),
        plot.caption = element_text(size = 7, face = "italic"),
        legend.position = "bottom")
```

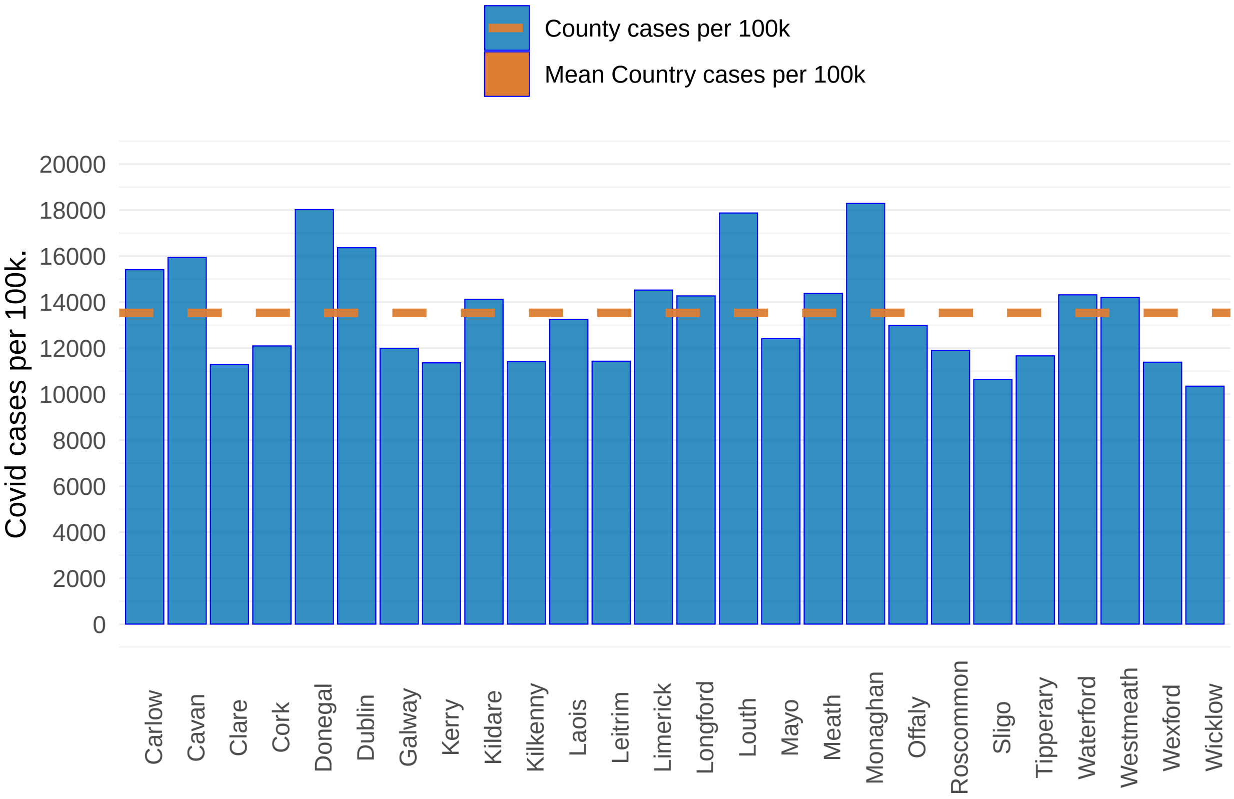


A visualisation that allows the reader to read how each county differs from the mean number of cases (per 100,000) in the country as at the 21 December 2021.

If we want to compare each county effectively against the mean of the country's cases, the best plot in my opinion is a simple bar chart, using a horizontal (dotted) line denoting the mean. This allows us to quickly see how a given county compares with the country's mean. I toyed with a few solutions to this, but after 4+plots and a few hundred lines of r, the bar chart was definitely the easiest to interpret while keeping said interpretation clean and concise.

```r
#A visualisation that allows the reader to read and compare the difference from the
#mean number of  cases per 100,000 of population for each county on the 21 December 2021.


#Mean total per county

#Create a new dataset for meanPer100k, filtering to requested date.
meanPer100kResult <- Ireland_Counties %>% filter(TimeStamp == "2021-12-21") %>%
  mutate(infectionsPer100k = round(((ConfirmedC/Population)*100000),1))
meanCasesTwentyFirst <- mean(meanPer100kResult$infectionsPer100k)
meanPer100kResult <- meanPer100kResult %>% mutate(meanCasesTwentyFirst)



#Barplot comparing mean cases as on 21/12/2021 vs the current mean for that county
#over the entire period.
```

```r
p <- ggplot(meanPer100kResult,
            aes(CountyName, infectionsPer100k, fill =
                  as.factor("County cases per 100k")))+

  geom_col(position="dodge",
           colour="blue", size=0.2, alpha=0.8)  +

  geom_hline(aes(yintercept = meanCasesTwentyFirst, fill =
                   as.factor("Mean Country cases per 100k")), alpha = .95,
             linetype = 2, color="#dd7e33", size = 1.5,  show.legend = TRUE) +

  scale_linetype_manual(values=c("twodash"))+


  scale_y_continuous(breaks=seq(0,20000,2000), limits = c(0, 20000),
                     name = "Covid cases per 100k.") +
  scale_x_discrete(name = "CountyName") +

  scale_fill_manual(values = palette_OkabeIto[c(5,6,3,1,8)],
                    name = NULL) +

  theme_minimal() +

  theme(
    legend.position = "top",
    legend.direction = "vertical",
    panel.grid.major.x = element_blank(),
    panel.grid.minor.x = element_blank(),
    panel.grid.major.y = element_line(size=0.3),
    panel.grid.minor.y = element_line(size=0.15),
    axis.title.x = element_blank(),
    axis.text.y = element_text(angle = 0),
    axis.text.x = element_text(angle = 90),
    axis.text.y.right = element_text(margin =
                                       margin(0, 0, 0, 0),
                                     size = 8),
    plot.margin = margin(14, 7, 3, 1.5))

p
```

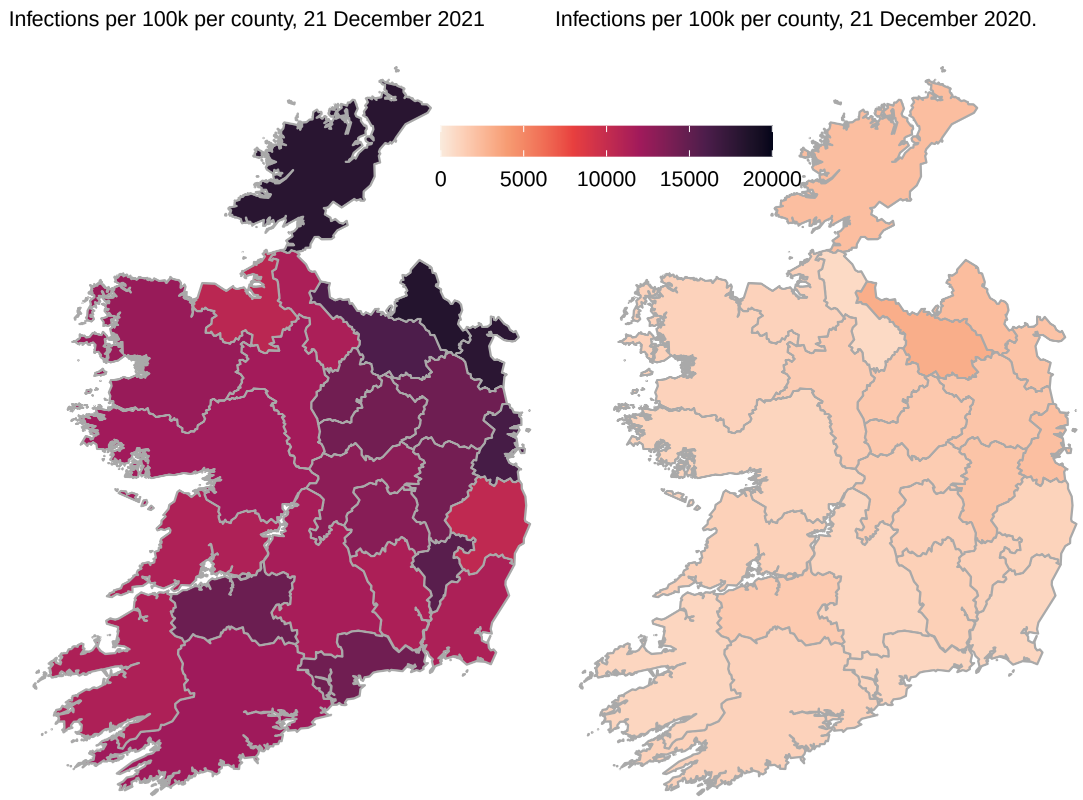


A choreopleth visualisation of the number of cases per 100,000 on the 21 December 2021 and on 21 December 2020. These should be placed side by side on the page and must use the same scale so that they can be directly compared.

The below is simply two choropleths working off two "mirror" datasets. I created two datasets, one filtered to 21/12/2021 and the other to 21/12/2020. With these choropleths plotted I can force the scale on them, in this case it is from 0 to 20,000. scale_fill_viridis_c() allows us to insure both plots use the same scale, and can be directly compared to one another. I believe a more detailed scale would make readability here quite a bit nicer as it is a bit too vague trying to pinpoint case numbers on the 21/12/2020 choropleth. You can assume most of the counties are sub 5000 cases, but beyond that it is difficult to make an accurate assumption to the case count. This plot is great however for make a broad comparison in that the per100k cases was definitely worse/larger in 2021, showing the virus has inded made its mark on the country in that year.

```
#A choreopleth visualisation of the number of cases per 100,000  on the 21
#December 2021 and on 21 December 2020. These should be placed side by side
#on the page and must use the same scale so that they can be directly compared.



#New table with date filter and new infectionsPer100ka
per100kResultA <- Ireland_Counties %>% filter(TimeStamp == "2021-12-21") %>%
  mutate(infectionsPer100ka = round(((ConfirmedC/Population)*100000),1))

PlotA <- ggplot(per100kResultA) +
  geom_sf(aes(fill = infectionsPer100ka),
          color = "darkgrey",
          linetype = 1,
          lwd = 0.4) +
  ggtitle("Covid cases_per_100k") +

  scale_fill_viridis_c(option = "rocket", direction =-1,
                       limits = c(0, 20000),
                       guide = guide_colorbar(
                         direction = "vertical",
                         label.position = "right",
                         title.position = "bottom",
                         ticks = T,
                         barwidth = grid::unit(.4, "cm"),
                         barheight = grid::unit(4.4, "cm")))+
  ggtitle("Infections per 100k per county, 21 December 2021") +
  theme_void() +
  theme(legend.title = element_blank(),
        plot.title = element_text(size = 8),
        legend.text.align = 0.5,
        legend.justification = c(0, 0),
        legend.position = c(11.005, 0.18))


#New table with different filtered date
per100kResultB <- Ireland_Counties %>% filter(TimeStamp == "2020-12-21") %>%
  mutate(infectionsPer100kb = round(((ConfirmedC/Population)*100000),1))

PlotB <- ggplot(per100kResultB) +
  geom_sf(aes(fill = infectionsPer100kb),
          color = "darkgrey",
          linetype = 1,
          lwd = 0.4) +
  ggtitle("Covid cases_per_100k") +

  scale_fill_viridis_c(option = "rocket", direction =-1,
                       limits = c(0, 20000),
                       guide = guide_colorbar(
                         direction = "horizontal",
                         label.position = "bottom",
                         title.position = "top",
                         ticks = T,
                         barwidth = grid::unit(4.4, "cm"),
                         barheight = grid::unit(.4, "cm")))+
  ggtitle("Infections per 100k per county, 21 December 2020.") +
  theme_void() +
  theme(legend.title = element_blank(),
        plot.title = element_text(size = 8),
        legend.text.align = 0.5,
        legend.justification = c(0, 0),
        legend.text = element_text(size=8),
        legend.position = c(-0.21, .8))

#Print both plots.
```

```
PlotA + PlotB
```


Infections per 100k per county, 21 December 2021
Infections per 100k per county, 21 December 2020.
0
5000
10000
15000
20000

A time series bar graph of the daily number of confirmed covid cases in one county in Ireland for one of the following periods: 3-month, 6 months or 1 year (1 year ~ January 1st 2021 to December 21 2021). This bar graph should also have a line representing the 7-day average for this period

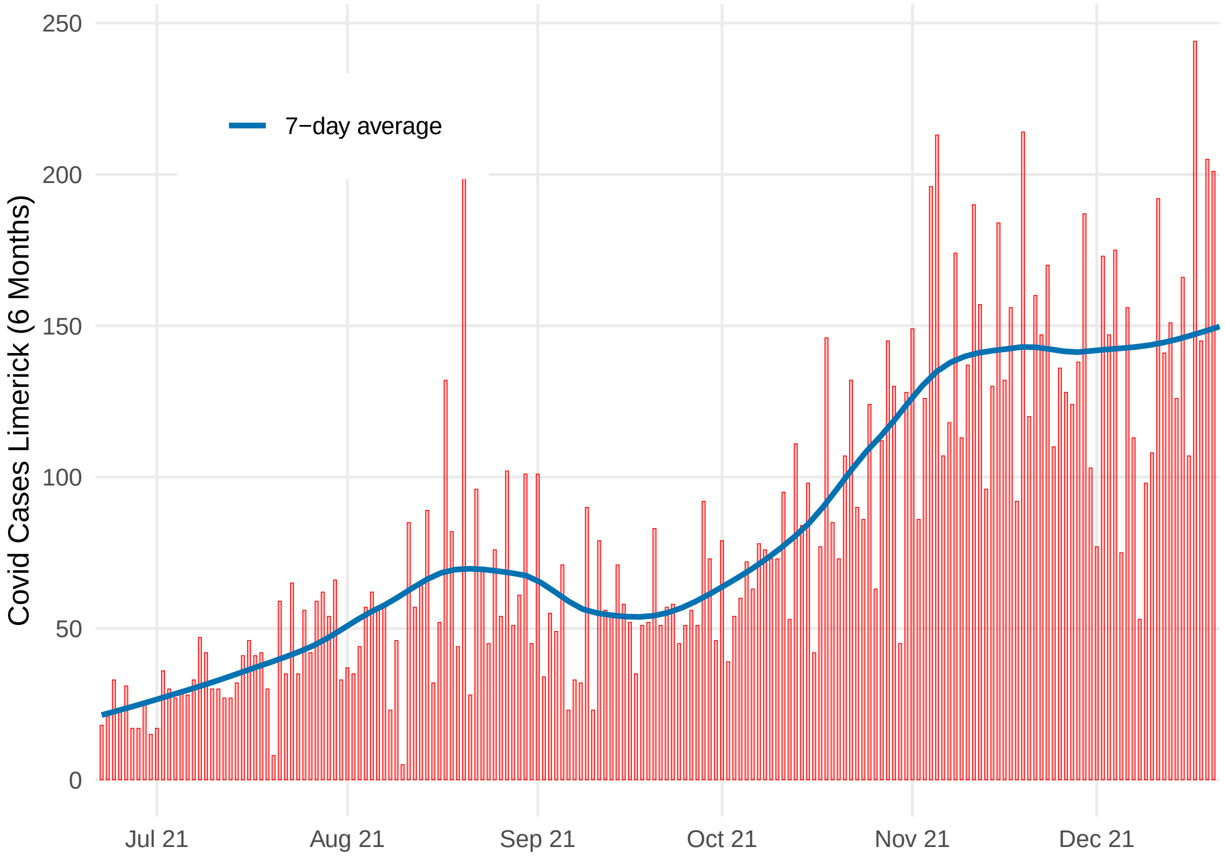
For our below bar graph we are using the 6-month period and a 7day average. With the field DailyCCase plotted in time series, we can see how cases increased over the course of 6 month. The bar graph allows us to see exceptional spikes in the cases, e.g around the start of September there is a sharp but short lived spiked in new cases. This is reflected in the 7-day average line we see in blue. You can quickly make out the peakes and valleys present in waves/spikes in case counts. September and November saw two big spikes, with the November keep a steady course. This was originally plotted against ConfirmedC and it was much more difficult to interpret. This is a good point to make in showing the value of the data you chose to plot. While both DailyCCase and ConfirmedC showed the rise in cases, DailyCCase is much more valuable in showing the important details such as spikes/waves in the cases as well as allowing the 7-day average to be much more useful. When plotting ConfirmedC, the 7-day average basically crept along the top of the plot due to the small difference seen between ConfirmedC and the 7-day average based on ConfirmedC. A 7-day average that made use of DailyCCase was much easier to interept and useful in the plot overall.

```
startdate<-"21-06-2021"
enddate<-"21-12-2021"
#Dataset for timescale tasks, includes filter to within given date range and
#preferred county (Limerick)
```

```
totalCasesTimeScale <- Ireland_Counties %>% filter(TimeStamp > dmy(startdate) &
                TimeStamp <= dmy(enddate))  %>% filter(CountyName == "Limerick")

source("moving_ave.R", echo = T)
```

```
##
## > moving_ave <- function(date, value, range, center = TRUE) {
## +     if (isTRUE(center)) {
## +         offset <- ceiling(range/2)
## +     }
## +     else {
## +  .... [TRUNCATED]
```

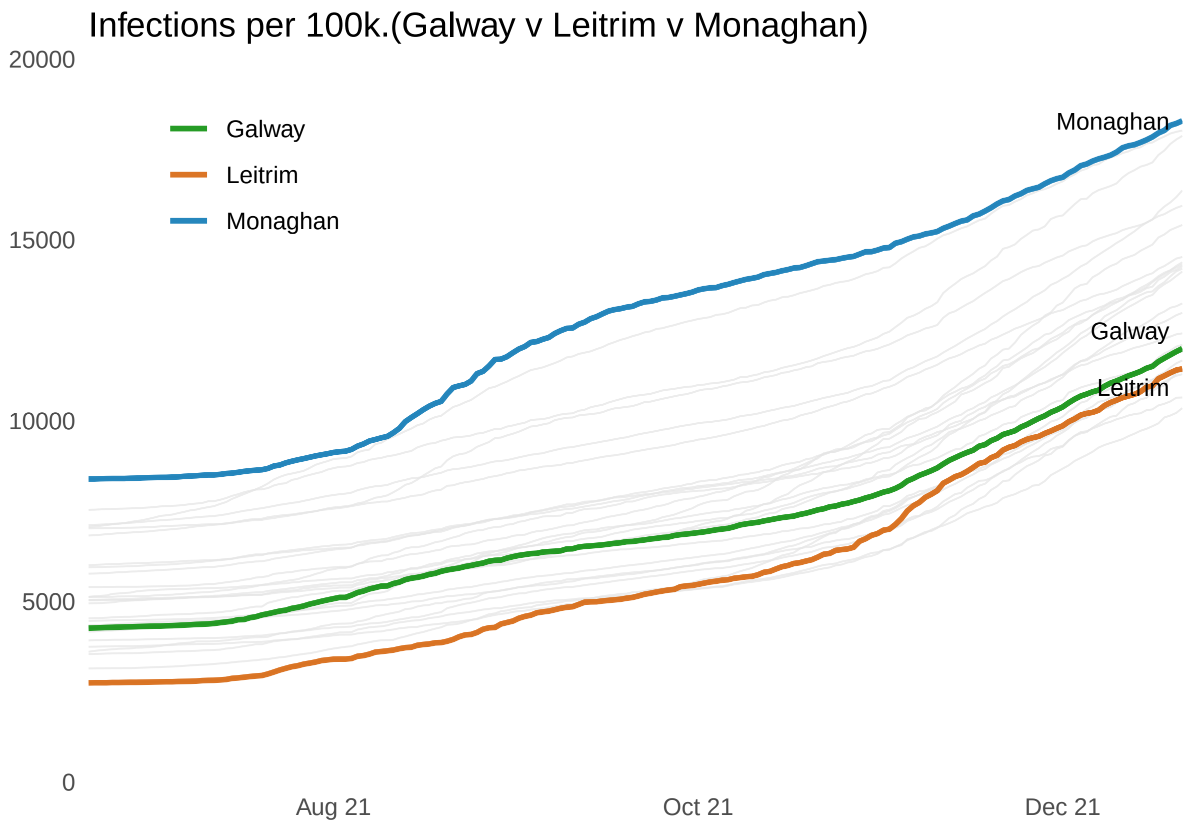
```
#7 day average.
sevenDayAverage <- totalCasesTimeScale %>%
  mutate(sevenDayAverage = moving_ave(TimeStamp, DailyCCase, 7, center = TRUE))

totalCasesTimeScale <-  mutate(sevenDayAverage)

totalCasesTimeScale<-totalCasesTimeScale[,c(3,4,7)] %>%
  pivot_longer(names_to ="covidCase",
               values_to ="TotalCases", cols=2:3) %>%
  rename(month = 'TimeStamp')%>%
  mutate(month=paste0(month, "2021-31"))%>%
  mutate(month= as.Date(month, format="%Y-%m-%d"))

totalCasesTimeScale = select(totalCasesTimeScale, -geometry)



#Dataset for timescale tasks, including filtered date range and county.
totalCasesTimeScale <- Ireland_Counties %>% filter(TimeStamp > dmy(startdate) &
              TimeStamp <= dmy(enddate))  %>% filter(CountyName == "Limerick")


#7 day average.
sevenDayAverage <- totalCasesTimeScale %>% mutate(sevenDayAverage =
                            moving_ave(TimeStamp, DailyCCase, 7, center = TRUE))

totalCasesTimeScale <-  mutate(sevenDayAverage)

loess_span1 = 0.4

sevenDayAverage %>%
ggplot(aes(TimeStamp, DailyCCase)) +
  geom_col(alpha=0.25, colour="red", fill = "red", size=.15, width = .5)  +

  geom_smooth(aes(color = "smooth1"), method="loess", span= loess_span1,
              size = 1.0, alpha = .5, na.rm = TRUE, se = FALSE) +

  scale_color_manual(
    values = c(
      `7d` = palette_OkabeIto[5],
      smooth1 = palette_OkabeIto[5]
    ),
    breaks = c("7d"),
    labels = c("7-day average"),
    name = NULL
  ) +
  scale_x_date(limits = c(dmy(startdate), dmy(enddate)), expand = c(0, 0),
               date_breaks = "1 month", date_labels="%b %y") +
  xlab(NULL) + ylab("Covid Cases Limerick (6 Months)") +
  theme_minimal()+
  theme(

    legend.justification = c(1, 0.5),
    legend.box.background = element_rect(fill = "white", color = NA),
    legend.box.margin = margin(0, 12, 6, 12),
    plot.margin = margin(3, 12, 3, 1.5),
    panel.grid.minor.x = element_blank(),
    panel.grid.minor.y = element_blank(),
    legend.position = c(0.35,0.85)
  )
```



A time series line graph that shows the cumulative number of cases per 100,000 in Galway and two other counties representing counties that have had the lowest and highest number of cases per 100,000. This time series line graph must also show the time series of all other counties in Ireland. However, the three selected counties (Galway and two other must be highlighted)

The below line plot is for cases per 100k in Ireland, with Galway, Leitrim (lowest) and Monaghan (Highest) coloured separately. This covers the timeseries from start to finish of our dataset. Every county can be seen in the background as a faint grey, for comparison. This is a nice graph for visualizing your chosen counties to the rest of the country at a glance. As I purposely chose the lowest infected county per 100k (Leitrim), highest (Monaghan) and Galway, we can easily distinguish and spot on the plot where they stand in comparison to the rest of the country. Monoaghan is exceptionally higher than most other counties for the most part, Leitrim is the lowest for most of the time series, slightly rising above some other counties towards the most recent dates. Galway looks to trend somewhere in the middle, if I plot the mean line I would imagine it would follow a similar path.

```r
#Create a new dataset for meanPer100k.
meanPer100kResult <- Ireland_Counties

meanPer100kResult <- meanPer100kResult %>% mutate(infectionsPer100k =
                                round(((ConfirmedC/Population)*100000),1))

#Table containing just the relevant data for our 3 counties.
threeCountyData <- meanPer100kResult %>%
  filter(CountyName %in% c("Galway", "Leitrim", "Monaghan"))

dataEnd <- threeCountyData %>% filter(TimeStamp == '2021-12-21')

background <- ggplot(meanPer100kResult,
                   aes(x =TimeStamp,
                       y=infectionsPer100k,
                       group = CountyName )) +
  geom_line( aes(group = CountyName),size= 0.35, na.rm = TRUE, color="grey90",
             alpha =0.7, show.legend = FALSE )+

  scale_y_continuous(name="Covid Cases per 100k",
                     limits = c(0, 20000),
                     expand=c(0,0)) +

  scale_x_date(limits = c(dmy(startdate), dmy(enddate)), date_breaks = "2 month",
               date_labels="%b %y", expand=c(0,0)) +


  theme_classic() +

  theme(panel.grid.major = element_blank(),
        panel.background = element_blank(),
        axis.line = element_blank(),
        axis.title.x=element_blank(),
        axis.text.x = element_text(  vjust = .5),
        legend.key = element_rect(fill = NA, colour = NA, size = 0.25),
        plot.margin = margin(14, 14, 8, 14))

#Overlay our coloured lines (3 counties)
foreground <- background +

  geom_line(data=threeCountyData, size =1, alpha=0.85, show.legend = TRUE,
            (aes(x =TimeStamp, y=infectionsPer100k, colour= CountyName,
                 group = CountyName))) +

  scale_colour_manual(values = c("green4","#D55E00", "#0072b2"),name = NULL,
                      limits = c("Galway", "Leitrim", "Monaghan")) +

  ##Label our data
  geom_text_repel(
    aes(label = CountyName), data = dataEnd,
    fontface ="plain", color = "black", size = 3)+


  ggtitle("Infections per 100k.(Galway v Leitrim v Monaghan)") +

  theme(
    axis.ticks.y.right = element_blank(),
    axis.ticks.y = element_blank(),
    axis.ticks.x = element_blank(),
    axis.title.y= element_blank(),
    axis.text.y.right = element_text(colour="black", size =8),
    legend.key = element_rect(fill = NA, colour = NA, size = 0.25),
    legend.position = c(0.15, .85))

foreground
```


Infections per 100k.(Galway v Leitrim v Monaghan)
Galway
Leitrim
Monaghan
Monaghan
Galway
Leitrim
Aug 21
Oct 21
Dec 21