Assignment 3 - Visualizing Covid Data in Irish Counties

Amogh M. Agnihotri Student ID - 21236437 (MSc-AI)

09/03/2022

Visualizing Covid Data for Irish Counties

In this Assignment we will visualize various aspects of spread covid across Ireland in different counties. The data we are visualizing ranges from early 2020 till the end of Year 2021. We will plot various diagrams to show increase / decrease in number of covid cases in various counties. We will also plot Choreopleths to visualize heat maps of Irish counties during 2 different times and visualize the growth of pandemic in the time series. Finally, we will also plot the time series line graph for all counties across all the time frame to visualize cumulative covid cases from start towards the end of pandemic and follow the trend. For all plotting, we will normalize the data to per hundred thousand of the population to have a fair comparisons among the counties.

Loading required Libraries

```
library(sf)
library(dplyr)
library(ggplot2)
library(RColorBrewer)
library(plyr)
library(colorspace)
library(gridExtra)
library(zoo)
library(scales)
library(lubridate)
```

Data Preparations

1. For Preparing the data, I took a decision to add a column in an intial Data Frame which contains the cumulative cases per Hundred Thousand people in the county, fore all the counties and all the time. This decision was made as many of the tasks require this data on different time steps.
2. Further, I made subsets of the Data frame in new data frames holding the particular data needed for that explicit task. I have made the data frames in a way that the same DF can be used in multiple tasks without any further changes. The below code contains most of the data pre-processing for upcoming tasks.

```
#Importing data from .shp file
file <- "E:/College Wiki/Data Visualisation/Assignment 3/Assignment Details//CovidCountyStati
sticsIreland_v2.shp"
covid_data <- st_read(file, quiet = TRUE)

#Creating a column to store cumulative cases per 100K people
covid_data <- covid_data %>%
  mutate(Cases_per_100K = (ConfirmedC/Population) * 100000)

#subseting the covid dataset for a particular time frame
cases_per_capita <- subset(covid_data, TimeStamp =="2021-12-21")
#Adding a new column to store cumulative cases perr 100K people
cases_per_capita$Cases_per_100K <- round(cases_per_capita$Cases_per_100K, digit = 2)


#Similar subsetting as above for time frame in year 2020
cases_per_capita_2020 <- subset(covid_data, TimeStamp =="2020-12-21")
cases_per_capita_2020$Cases_per_100K <- round(cases_per_capita_2020$Cases_per_100K, digit = 2
)

#Calculating mean and creating a new DF for storing the difference from mean
mean_per_100k = mean(cases_per_capita$Cases_per_100K)
cases_per_capita_2021 <- cases_per_capita %>%
  mutate(Difference_mean = Cases_per_100K - mean_per_100k)
```

Task 1 - Plotting the flipped bar chart for showing cumulative Covid-19 cases for all counties on 21-DEC-2021

Task description -

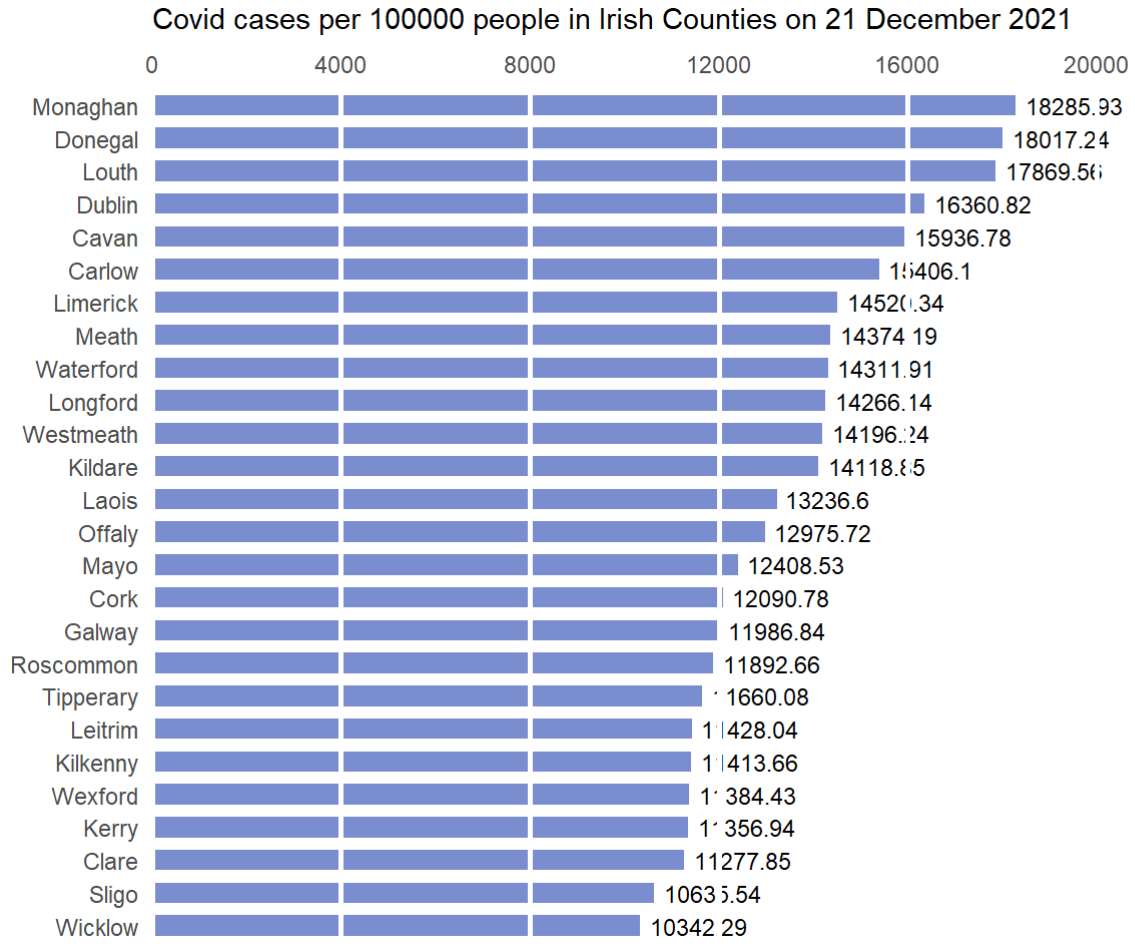
1. The task was to develop a visualization to allow reader to accurately read and compare cumulative number of cases per 100,000 of population per county on the 21 December 2021.
2. As shown above, we subset the data as per the given date for all the 26 counties.
3. We already have a column for cumulative number of cases per 100k people for each county

Plotting Visualization (Design Decisions) - Flipped Bar Graph

1. Bar graphs are generally used to present and compare categorical set of values.
2. I saw that no other visualization like pie chart, line graphs, etc will be able to communicate the required message correctly as they lack overall sense of comparisons.
3. The proportional graphs will show the figures with whole proportion(100%) and divide them as per the each categories value. Which is not ideal in this case.
4. Initially I started with bar graphs, but as there are more than 6 categories it is better to flip it horizontally for better readability and easy way of comparison.
5. In the flipped graph show, the name of the counties are on left side which is the X-axis and and I have moved Y-axis label to Top so that the values can be easily scene and compared.
6. I have removed X and Y axis lines as the don’t contribute anything to the graph at this stage and will cause clustering.
7. For that, I have plotted grid lines at appropriate intervals of 4000 cases ranging from 0 to 20000, which will help reader gauge the number for a particular county without again going to Y axis label at top.

8. As the categories have natural ordering, I have reorders the data in descending order from top.
9. Finally, as the ask was to have the reader accurate notion of numbers, I have printed exact value in front of the bar. With this, I have also chosen to keep Y-axis labels as I think they are still necessary to provide flying information for reader reading it fast. 10.As there is only one color, there should not be any issue for color blind people to read the graph.

```
ggplot(cases_per_capita,
        aes(x = reorder(CountyName, Cases_per_100K),
            y = Cases_per_100K)) +
  geom_col(fill = "#5069be", width = 0.63, alpha = 0.76) +
  geom_text(aes(label=Cases_per_100K), size = 3, hjust=-0.1) +
  scale_y_continuous(limits = c(0, 2.5e4),
                     expand = c(0,0),
                     breaks = seq(0,2e4,by=4e3),
                     labels = as.character(seq(0,20000,by=4000)),
                     name = "Covid cases per 100000 people",
                     sec.axis = dup_axis(),
                     )+

  ggtitle("Covid cases per 100000 people in Irish Counties on 21 December 2021 ")+

  coord_flip(clip = "off")+

  theme(
    axis.title = element_blank(),
    axis.line.y = element_blank(),
    axis.ticks.y = element_blank(),
    axis.line.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.title.y = element_blank(),
    axis.title.x.bottom = element_blank(),
    axis.text.x.bottom = element_blank(),
    plot.title = element_text(size = 11),
    plot.margin = margin(3, 6, 3, 3),
    panel.background = element_blank(),
    panel.grid = element_blank(),
    panel.grid.major.x = element_line(size = 0.9,
                                      linetype = 'solid',
                                      colour = "white"),
    panel.ontop = TRUE
    )
```



Task 2 - Plotting a graph to show Distance from mean of cumulative covid cases per 100K people in irish counties on 21-DEC-2021

Task description -

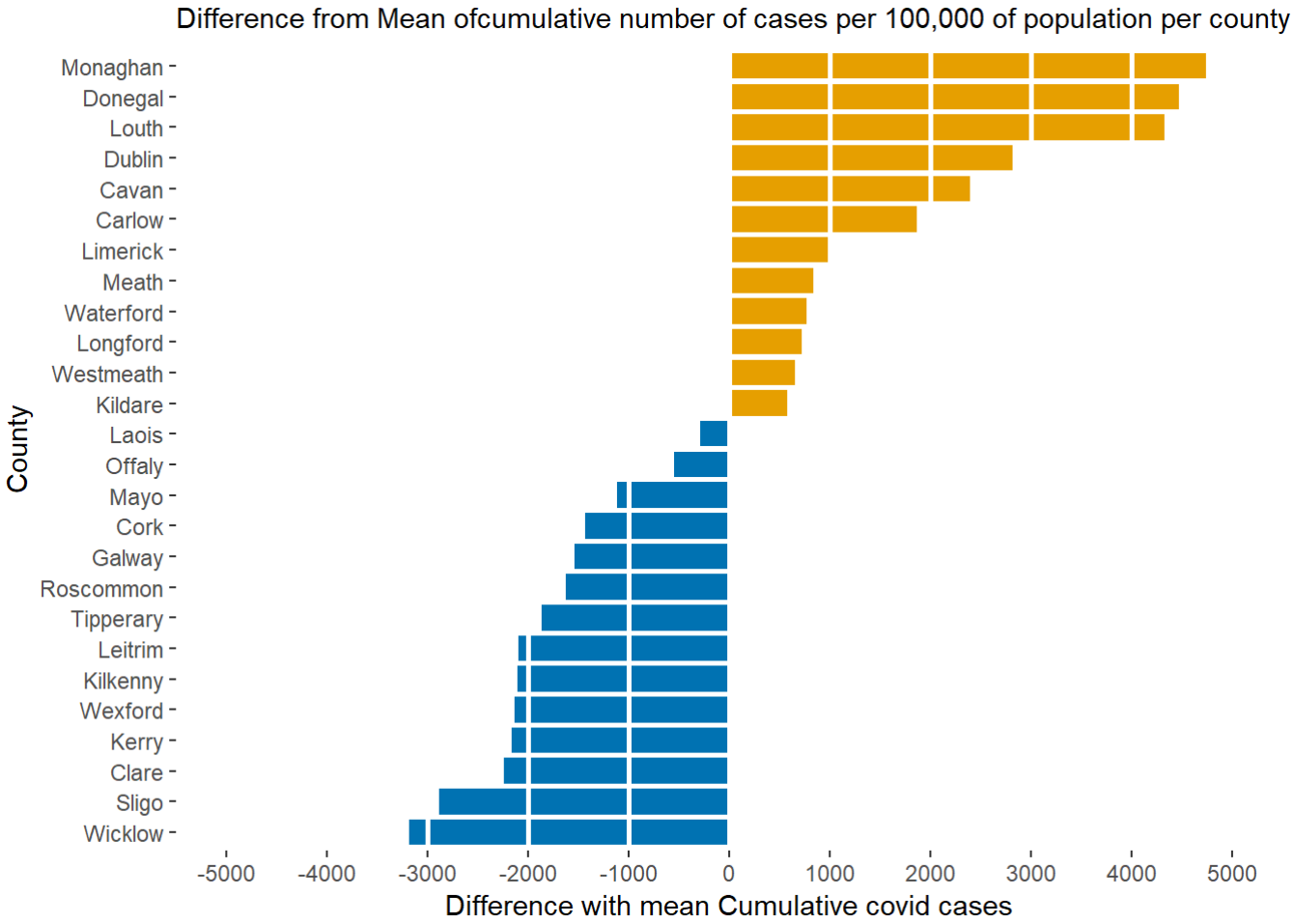
1. This task was to create a visualization that allows reader to see how each county differs from the mean cumulative number of cases (per 100,000) in the country as at the 21 December 2021.
2. For this we calculate a mean of cumulative mean of all counties per 100,000 people at the given date and store it in variable.
3. Further we create a column in dataframe which gives us difference between cumulative number of cases and the mean value, at the given date for all counties per 100,000 people.

Plotting Visualization (Design Decisions) - Diverging Bar Graph

1. As we know the distance from mean might also be a negative value. Bar graphs are not the best ones to handle and show negative values to reader.
2. The even if we convert the negative values to positive with a threshold it would be falsifying the exact data.
3. Normal Dot plots and Pareto chart are not the best one to use for such visualizations and carry similar drawbacks for this task as the bar graphs.
4. The two visualizations that I though would be best for this were - Diverging Bar Charts and Diverging Dot Plots. The diverging dot plots are used when the range between values are very narrow or hard to compare or the range of values is really short. Hence the dot plots use position as an aesthetic while Bar Graphs will use length as the aesthetics.

5. Both uses of dot plots mentioned above are not the cases and diverging bar plot can easily show divergence from mean in both positive and negative directions. hence the decision was to taken to plot a Diverging bar graph showing variance from the mean.
6. As seen in the bar above bar graph, her as well the axis are flipped for the same reasons, as categories are more than 6.
7. With monitoring the column of difference from mean, the range of Y-Axis on the bottom is plotted from -5000 till +5000, with appropriate breaks at every 1000.
8. Grid lines are provided so that reader can gauge the value of a particular bar even if its away from the Y label.
9. I have explicitly avoided the exact values of distance from mean in front of the bar as they won't be adding much information and the bar will be clustered.
10. The color combination is specified for Negative and Positive distances from mean i.e 0 and again the categories are ordered in descending way as they posses natural ordering.
11. The color combination is adapted from colorblind scales with slight tweaking in HCL color picker tool. It carries distance of more than 90 units for all 4 major types of colorblindness.
12. With this visualization, the reader can easily guess how far from mean a particular county was on 21-Dec-2021, in terms of cumulative covid cases per 100,000 people in each county.

```
# Choosing colorblind friendly colors with maximum distance (Selected by checking on 'I want
 hue')
color<-ifelse(cases_per_capita_2021$Difference_mean < 0, "#0072B2","#E69F00")

ggplot(cases_per_capita_2021,aes(x=reorder(CountyName, Difference_mean),
                         y=Difference_mean)) +
  geom_bar(stat = "identity",
           show.legend = TRUE,
           fill = color,      # Background color
           color = "white") +
  geom_hline(yintercept = 0, color = 1, lwd = 0.2) +
  ggtitle("Difference from Mean ofcumulative number of cases per 100,000 of population per co
unty on 21 Dec 2021") +
  xlab("County") +
  ylab("Difference with mean Cumulative covid cases") +
  scale_y_continuous(breaks= seq(-5000, 5000, by = 1000),
                     limits = c(-5000, 5000)) +
  coord_flip() +
  theme(
    plot.title = element_text(size = 11),
    plot.margin = margin(3, 6, 3, 3),
    panel.background = element_blank(),
    panel.grid.major.x = element_line(size = 0.9,
                                      linetype = 'solid',
                                      colour = "white"),
    panel.grid = element_blank(),
    legend.text = element_text("Mean = 13529"),
    panel.ontop = TRUE
  )
```



Task 3 - Plotting Choreopleth at 2 Time frames of Cumulative Covid cases per 100,000 people in Irish counties

Task description -

1. The task was to create a choreopleth visualization of Irish conties for number of cumulative cases per 100,000 people on 21-Dec-2020 and 21-Dec-2021. 2.For both the dates, we had data subseted as shown above in different time frames which had cumulative cases. 3.We had to show both the Choreopleth side by side with continuous color gradient for comparing the situation on different times as given.
2. We had already imported shape coordinates initially from the .shp file.

Plotting Visualization (Design Decisions) - Choreopleth Maps

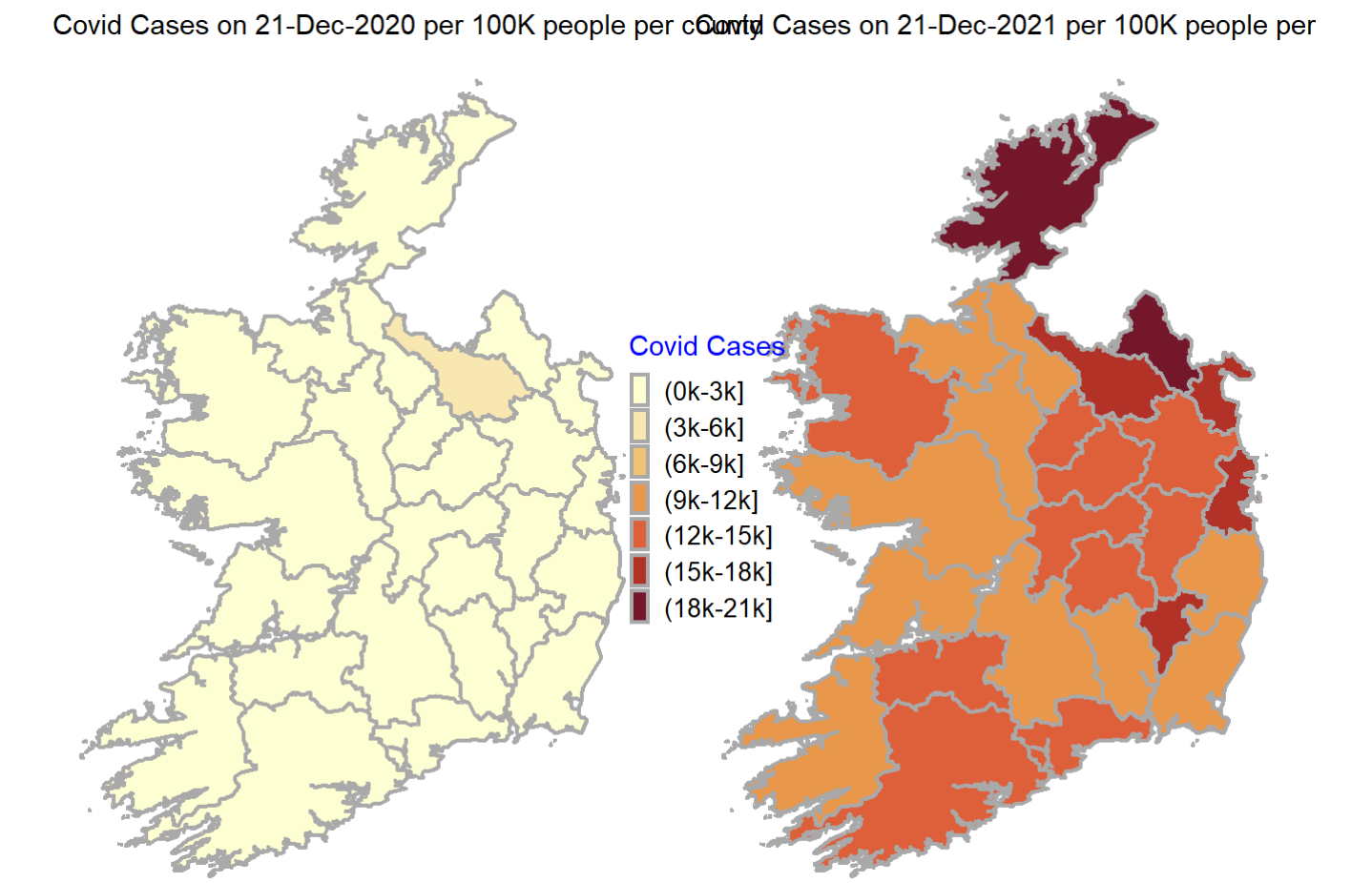
1. The main design decision was to select proper intervals which are common for both the time frames.
2. As the choreopleth maps are mainly used to show changes in area for particular variable, as here it is number of covid cases, for two different time frames, we had to develop a discrete scale with appropriate intervals of range.
3. As we I already saved the data for 21-Dec-2020 I knew that values ranged from 500 till 4000 for cumulative cases per 100,000 people for every county. So i had my minimum limit figured out.
4. As we already visualized the same thing for 21-12-2021, I knew the values were ranging from 8000 till 19000.
5. I created a scale and stored maximum and minimum value for the range and floored and fenced it to nearest thousand respectively. By doing so I wont miss any any category. 6.I created a new column in

already existing data frames which held the required data and populated new column with the appropriate intervals based on their value for cumulative cases per 100,000 people for each county.

6. I save the number of intervals in a variable and created a color pallet from existing pallet which is “YlOrRd”. I will take the first 7 colors from it. I checked the pallet on I want hue for colorblind score and it has shown more than 60 unit distance for all 4 major colorblindness, which is good enough. 8.Further we map discrete intervals to appropriate scales. That is low value gets fainter color and high value gets the darker color.
7. We perform actions for second data frame by adding intervals and mapping them. The we plot both the Choreopleth as shown below side by side with grid.arrange function and have single legend in middle for easy readability and comparison.
8. It is evident from the figure that t=on 21-December-2020, there were very less cases per 100,000 people on all the counties with exception of 1 county.but in Dec 2021 the cases skyrocketed across the nation with counties like Dublin, Donegal, etc. topping the charts which is indicated by very dark shade of the color.
9. The color orange and its shades are chosen as a resemblance to heat map for spreading of the deadly disease.

```
#Storing munimum and Maximum for scaling the heatmap colors
scale_minimum <- round_any(min(cases_per_capita_2020$Cases_per_100K), 1000, f = floor)
scale_maximum<- round_any(max(cases_per_capita$Cases_per_100K), 1000, f = ceiling)

#Adding breaks as appropriate
breakss<-seq(scale_minimum,scale_maximum+2000, by =3000)

#Adding intervals to DF for Splitting the counties -- year 2021
cases_per_capita$Case_interval_D <- cut(cases_per_capita$Cases_per_100K,
                                        breaks = breakss,
                                        dig.lab = 7)

#Adding intervals to DF for Splitting the counties -- year 2020
cases_per_capita_2020$Case_interval_C <-cut(cases_per_capita_2020$Cases_per_100K,
                                            breaks = breakss,
                                            dig.lab = 7)

# Selecting YlOrRd color pallet and number of colors equal to intervals
nlevels <- nlevels(cases_per_capita$Case_interval_D )
pal <- hcl.colors(nlevels, "YlOrRd", rev = TRUE)
pal_desat<-desaturate(pal,amount = 0.2)


labs <- breakss/1000
labs_plot <- paste0("(", labs[1:nlevels], "k-", labs[1:nlevels+1], "k]")
##############################################################################################
##############
#Plotting the Choreopleth for year 21-Dec-2021 with implemented intervals

map2021 <- ggplot(cases_per_capita) +
  geom_sf(aes(fill = Case_interval_D),
          color = "darkgrey",
          linetype = 1,
          lwd = 0.8) +

  labs(subtitle = "Covid Cases on 21-Dec-2021 per 100K people per county")+
    # Custom palette
  scale_fill_manual(values = pal_desat,
                    drop = FALSE,
                    na.value = "grey80",
                    label = labs_plot,
                    # Legend
                    guide = guide_legend(direction = "vertical",
                                         ncol = 1,
                                         label.position = "right",
                                         title = "Covid Cases"
                    )) +
   # Theme
  theme_void() +
    theme(legend.title = element_text(color = "blue", size = 11),
        legend.text = element_text(size=10),
        legend.key.height = grid::unit(0.5, "cm"),
        legend.key.width = grid::unit(0.3, "cm"),
        plot.caption = element_text(size = 9, face = "italic"),
        legend.position = c(0.020,0.5))
```

```
#############################################################################################
#####
#Plotting the Choreopleth for year 21-Dec-2020 with implemented intervals


map2020 <- ggplot(cases_per_capita_2020) +
  geom_sf(aes(fill = Case_interval_C),
          color = "darkgrey",
          linetype = 1,
          lwd = 0.8) +
  labs(
       subtitle = "Covid Cases on 21-Dec-2020 per 100K people per county",
       caption = "source: central statistics office") +
  # Custom palette
  scale_fill_manual(values = pal_desat,
                    drop = FALSE,
                    na.value = "grey80",
                    label = labs_plot)+

  # Theme
  theme_void() +
  theme(legend.title = element_blank(),
        legend.text = element_blank(),
        legend.key.height = element_blank(),
        plot.caption = element_blank(),
        legend.position = "c(-0.03,0.5)")


#Arranging both the Choreopleth side by side
grid.arrange(map2020, map2021, ncol=2)
```



Task 4 - Plotting Daily number of cases for 3 months from September 2021 - December 2021 for County Galway as well as Plotting a line which show rolling mean of past seven days at a point

Task description -

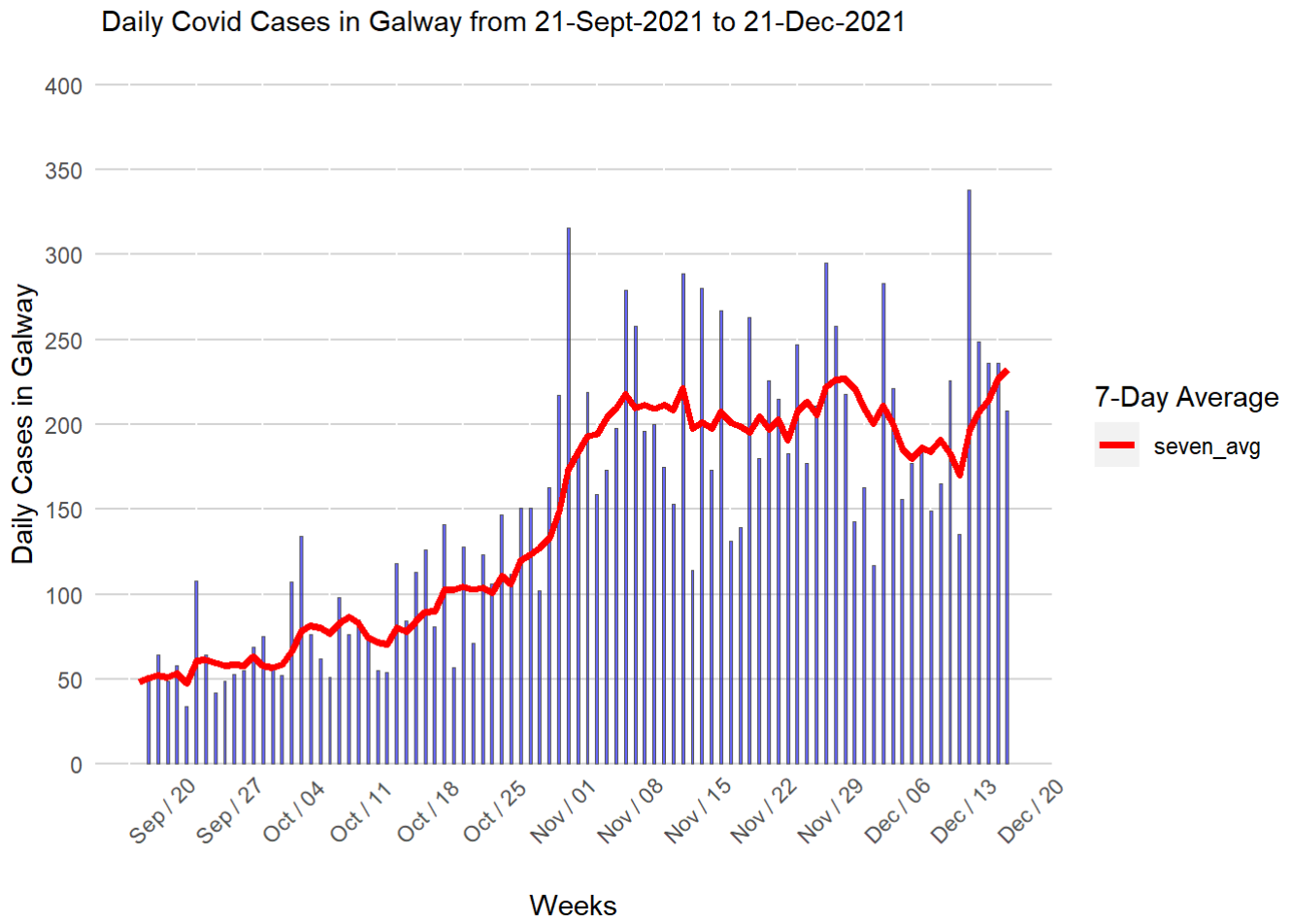
1. The task was plot the a time series bar graph of the daily number of confirmed covid cases in one county in Ireland for period on 3 months for any particular county.
2. Also to plot a line on the graph which shows a rolling mean or average number of cases for last 7 days at any given point.

Plotting Visualization (Design Decisions) - Time Series Bar Graph

1. If our primary aim is to compare values on monthly or weekly basis rather than showing a trend then the bar graph is better than line graph for plotting based on time series.
2. The classical verticals bar here can show discrete, numerical comparisons across the categories.
3. The length is such bars in time series allows accurate comparison of individual values to one and other.
4. The role of rolling mean line is to show average of past 7-days which gives the reader idea of the trend about how everyday cases are compared with the past days. With this reader can gauge information and decide upon multiple decisions.
5. For this I created a separate data frame from original one carrying on all the values and filtered it county Galway and the 3 months time period form Sept 2021 to December 2021.

6. I added a new column for adding a rolling mean at all the days from past 7 days. This will be used to draw the rolling mean line.
7. As I saw the data set I scaled Y axis ranging from 0 to 400 with break at every 50. The horizontal grid lines will help readr to find the value quickly and accurately.
8. The X-axis scaled with weekly basis with scale_x_date and labels are printed in MM/DD format with a little twist to adjust them as shown in the figure.
9. The mean average line can be seen and daily cases can be compared for every day to rolling mean.
10. We can easily see the trend going upward as we approach the month of December after seven day average sunk pretty low in the month on November.
11. The color combination used here for graph and the line again maintains pretty good distance, not causing any issues for colorblind people with 4 major colorblindness.

```
#Subsetting data for County Galway
cases_galway <- subset(covid_data, CountyName=="Galway")
#Calculating rolling average in new column
cases_galway <- cases_galway %>%
  mutate(seven_avg = rollmean(DailyCCase, 7, align="right", fill=0))

#Subsetting cases for 3 months timeframe
cases_galway <- subset(cases_galway, TimeStamp>="2021-09-21")

# Plotting the graph with mean line
ggplot(cases_galway,
       aes(x=TimeStamp, y=DailyCCase))+
  geom_col(fill="Blue", position = "dodge", size = 0.1, alpha=0.6, colour = "#555555", width
 = 0.3)+
  scale_y_continuous(limits = c(0, 400),
                     breaks = seq(0,400, by =50),
                     labels = as.character(seq(0,400,by=50)),
                     name = "Daily Cases in Galway") +
  scale_x_date(date_breaks = "1 week",
               date_labels = "%b / %d",
               limits = c(as.Date("2021-09-21"), NA),
               name = "Weeks") +

  ggtitle("Daily Covid Cases in Galway from 21-Sept-2021 to 21-Dec-2021") +

  geom_line(aes(y=seven_avg,color = "seven_avg"), size = 1.25) +
  scale_color_manual(name="7-Day Average",values = c("seven_avg"="red"))+
  #labs(color = "7-Day Average")+

  theme(
    #axis.title.y = element_blank(),
    axis.line.y = element_blank(),
    axis.ticks.y = element_blank(),
    axis.line.x = element_blank(),
    axis.ticks.x = element_blank(),
    axis.text.x = element_text(angle = 45,
                               vjust = 0.65, hjust = 0),
    #axis.title.x = element_blank(),
    plot.title = element_text(hjust = 0.04, size = 11),
    plot.margin = margin(6, 6, 3, 3),
    panel.background = element_blank(),
    panel.grid.major.y =
      element_line(size = 0.4,
                   linetype = 'solid',
                   colour = "light gray")
    )
```



Task 5 – Plotting cumulative covid cases per 100K population for 2 years (start of 2020 till end of 2021) For all Irish counties, highlighting County Galway and 2 other counties with Least number and maximum number of cases

Task description -

1. The task is to plot 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. The other counties need to be in background and faded a bit.

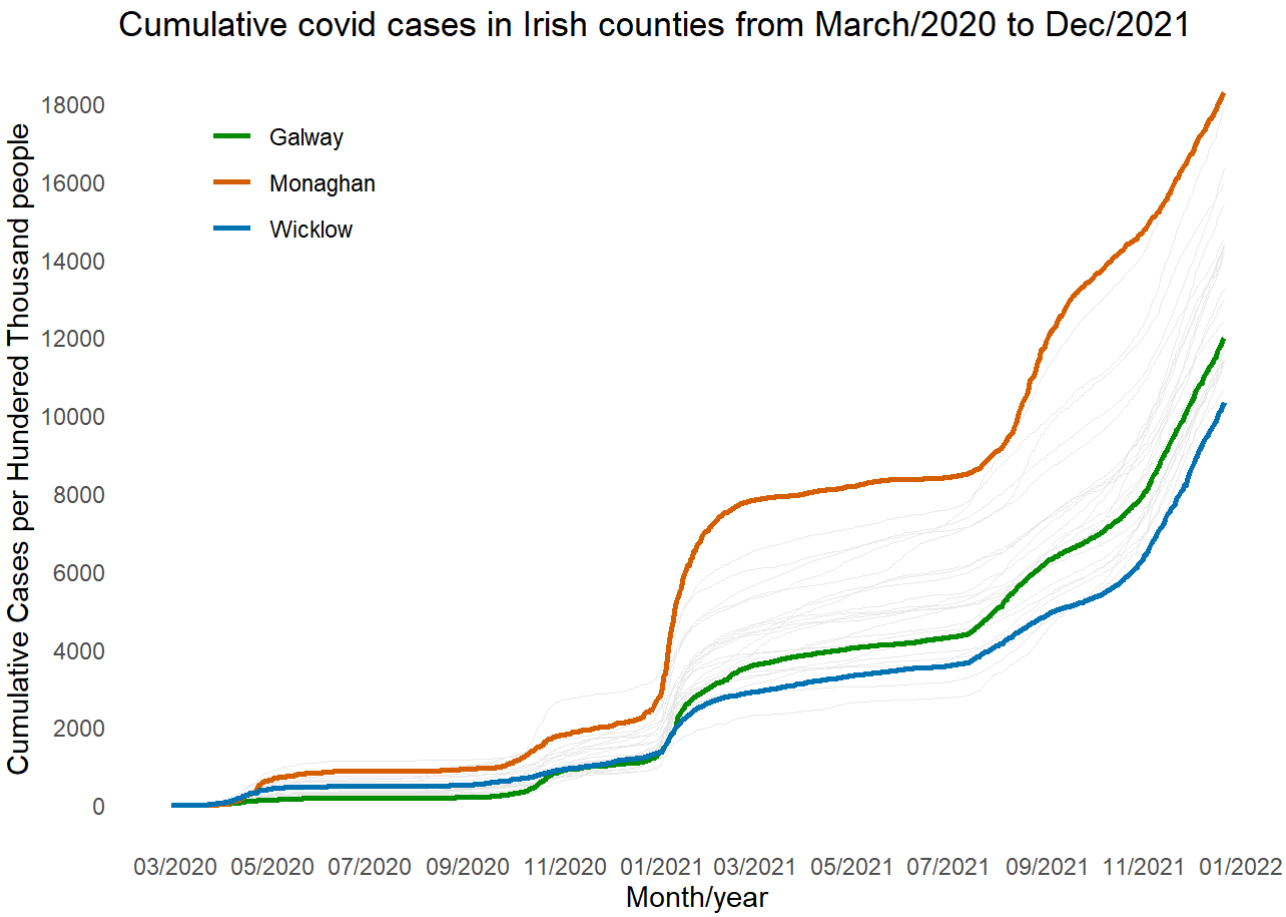
Plotting Visualization (Design Decisions) - Time Series line graph Graph with emphasize on select catrgories

1. Here we basically plot all the cumulative cases for all counties across the 2-year time frame.
2. The line graphs are useful here as we don’t need to know the exact values but we just need to see the trend and compare it with other counties.
3. As there are 26 counties, the line graph gets clustered and makes very less sense in terms of providing any information.
4. As we are only interested in 3 counties - Galway(county of my choice), Monaghan(County with highest cumulative cases per 100,000 people) and Wicklow(County with lowest cumulative cases per 100,000 people). We will plot these lines in forefront and highlight them with distinguishing colors. We will keep

rest of the data in background which is seen just as a reference and has very little importance in terms of providing valuable information.

5. As we know from previous plotting the highest cumulative cases are till 18 to 19k and starting from 0, I have scaled the y axis with that range and break at every 2000.
6. The dates on X-axis are scaled similarly as previous plot, here with 2 months interval between them and labeled in MM/YYYY format.
7. The grid lines are not necessary as we are not looking for any accurate value but the comparison of trends in time series of 2 years. The legend gives appropriate notations for lines of counties.
8. The process was to create a foreground first with all the gray lines.
9. Then I created a custom DF containing all the details about 3 counties of our interest and plotted in forefront with appropriate colors.
10. The plot successfully shows the trends between our counties of intrest from start of the time till the end

```
# Saving the counties to be showed in Forefront
target_counties <- c("Galway", "Monaghan", "Wicklow")

# Storing data of Target counties in new DF
county_data_target <- subset(covid_data, CountyName %in% target_counties)

# Plotting background
background <- ggplot(covid_data, aes(x=TimeStamp, y = Cases_per_100K)) +
  scale_x_date(name = "Month/year", breaks = "2 month",
               labels=date_format("%m/%Y") ) +

  scale_y_continuous(breaks=seq(0,20000, by = 2000),
                     name = "Cumulative Cases per Hundered Thousand people",
                     labels = seq(0,20000, by = 2000))
# Modifying Background in a way that foreground can be highlithed
background <- background +
  geom_line(aes(group = CountyName),size= 0.35, na.rm = TRUE, color = "grey90", alpha =0.8, s
how.legend = FALSE) +
  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)
  )

# Plotting foreground along with highlighting our target counties
P2 <- background +

  geom_line(data=county_data_target, size =1, show.legend = TRUE,
            (aes(x =TimeStamp, y=Cases_per_100K, colour= CountyName, group = CountyName))) +


  scale_colour_manual(values = c("green4","#D55E00", "#0072b2"),name = NULL,
                      limits = c("Galway", "Monaghan", "Wicklow")) +
  ggtitle("Cumulative covid cases in Irish counties from March/2020 to Dec/2021") +
  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)
  )

P2
```


Cumulative covid cases in Irish counties from March/2020 to Dec/2021
Cumulative Cases per Hundered Thousand people
18000
16000
14000
12000
10000
8000
6000
4000
2000
0
Galway
Monaghan
Wicklow
03/2020
05/2020
07/2020
09/2020
11/2020
01/2021
03/2021
05/2021
07/2021
09/2021
11/2021
01/2022
Month/year