PSY6422 Final Project- COVID-19 Deaths in the UK in 2021

Mc.N

2022/4/25

Contents

1 Introduction

It has been more than two years since the first confirmed case of the COVID-19 virus was detected. We have paid a terrible price in money, resources, freedom and countless lives. The global health system and governments have been greatly challenged during this time. Although we are not confident that COVID-19 has been wholly conquered, things are looking up. The project aims to visualize what we will lose in 2021. We are reminded that what we have achieved so far is precious.

The project uses data from the ***Office for National Statistics*** website on the number of deaths registered weekly in England and Wales 2021 edition.The dataset was updated weekly during the pandemic and was counted whenever COVID-19 was mentioned or suspected in a death certificate.The number of deaths and age groupings from 52 weeks in 2021 were used.

2 Data Visualization

2.1 Check packages

In this visualization project, I use the following packages, the *"require"* and *"if"* function to help us check if it is installed.

```
# For Example
if(!require("openxlsx")) install.packages("openxlsx")
library(openxlsx)
```

2.2 Visualization 1 — Weekly Deaths in 2021

I'll go into more detail about how I process the data and chart it, but I'll skip the repetitions. Read raw data from the /Raw_Data folder. The worksheet we use in this section is sheet 9.

```
data1 <- read.xlsx(here("Raw_Data/publishedweek522021.xlsx"), sheet = 9)

head(data1,2)
```

```
##                                                                        Contents
## 1 Weekly provisional figures on deaths registered in the UK1,2,3,4,5,6,7,8,9,10
## 2                                                                  Week number
##     X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13  X14 X15  X16 X17 X18 X19 X20 X21
## 1 <NA> NA NA NA NA NA NA NA  NA  NA  NA  NA <NA>  NA <NA>  NA  NA  NA  NA  NA
## 2 <NA>  1  2  3  4  5  6  7   8   9  10  11   12  13   14  15  16  17  18  19
##   X22 X23 X24 X25 X26 X27 X28 X29 X30 X31 X32 X33 X34 X35 X36 X37 X38 X39 X40
## 1  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## 2  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38
##   X41 X42 X43 X44 X45 X46 X47 X48 X49 X50 X51 X52 X53 X54
## 1  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA  NA
## 2  39  40  41  42  43  44  45  46  47  48  49  50  51  52
```

The content of this worksheet is weekly provisional figures on deaths registered where COVID-19 was mentioned on the death certificate in England and Wales. It contains too much information, and the data set is not in the right format for R to plot. So we need to select and preprocess the data in the worksheet.

In this section, we are only interested in weeks and total weekly deaths, so only rows 2 and 4 of the worksheet are selected for visualization.

```
Weekly_Death <- slice(data1,2L,4L)
```

This data is not directly used to draw a bar chart. So there are operations to be performed on the data set, such as removing empty values, transposing, and changing data type.

```
rownames(Weekly_Death)=Weekly_Death[,1]
Weekly_Death=Weekly_Death[,-1]

# Transposing
Weekly_Death <-t(Weekly_Death)
# Changing data type
```

```
Weekly_Death <-as.data.frame(Weekly_Death)
# Removing empty values
Weekly_Death<- na.omit(Weekly_Death)

colnames(Weekly_Death)
```

```
## [1] "Week number"                    "UK deaths involving COVID-19 7"
```

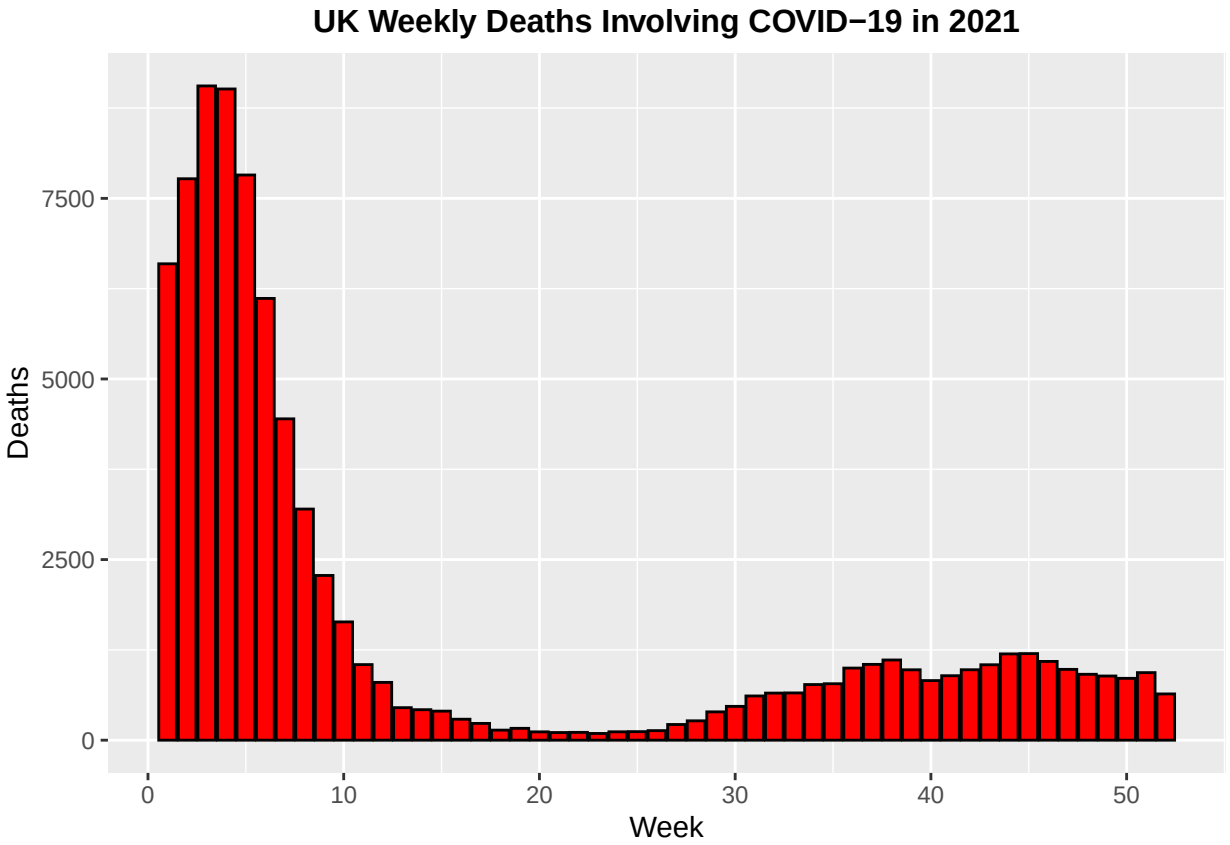
```
# Rename
names(Weekly_Death)[names(Weekly_Death) =="Week number"] <-"Week"

names(Weekly_Death)[names(Weekly_Death) =="UK deaths involving COVID-19 7"] <-"Deaths"


Weekly_Death = as.data.frame(lapply(Weekly_Death,as.numeric))
```

```
head(Weekly_Death,3)
```

```
##   Week Deaths
## 1    1   6594
## 2    2   7771
## 3    3   9056
```

This set of data becomes concise and can be used to plot. Use ggplot2 to plot a bar graph with *Week number* on the horizontal and *Deaths* on the vertical. Use *"fill"* = to control the fill color and *"color ="* to control the border color.

```
p1 <- ggplot(Weekly_Death,aes(x=Week, y=Deaths))+

  geom_bar(stat='identity',fill="red", colour="black")+

#add a title for the figure.
  ggtitle("UK Weekly Deaths Involving COVID-19 in 2021") +

  theme(plot.title = element_text(size = 12, face = "bold", hjust = 0.5))

plot(p1)
```


UK Weekly Deaths Involving COVID−19 in 2021
Deaths
7500
5000
2500
0
0
10
20
30
40
50
Week

2.3 Visualization 2 — Weekly Deaths Grouping by Age

In Visualization 2, I want to show death in different age groups. By grouping by age, we can find out which age groups are at greater risk when facing COVID-19.

```
## [1] "Week"         "Under.1.year" "X01.14"       "X15.44"       "X45.64"
## [6] "X65.74"       "X75.84"       "X85."
```

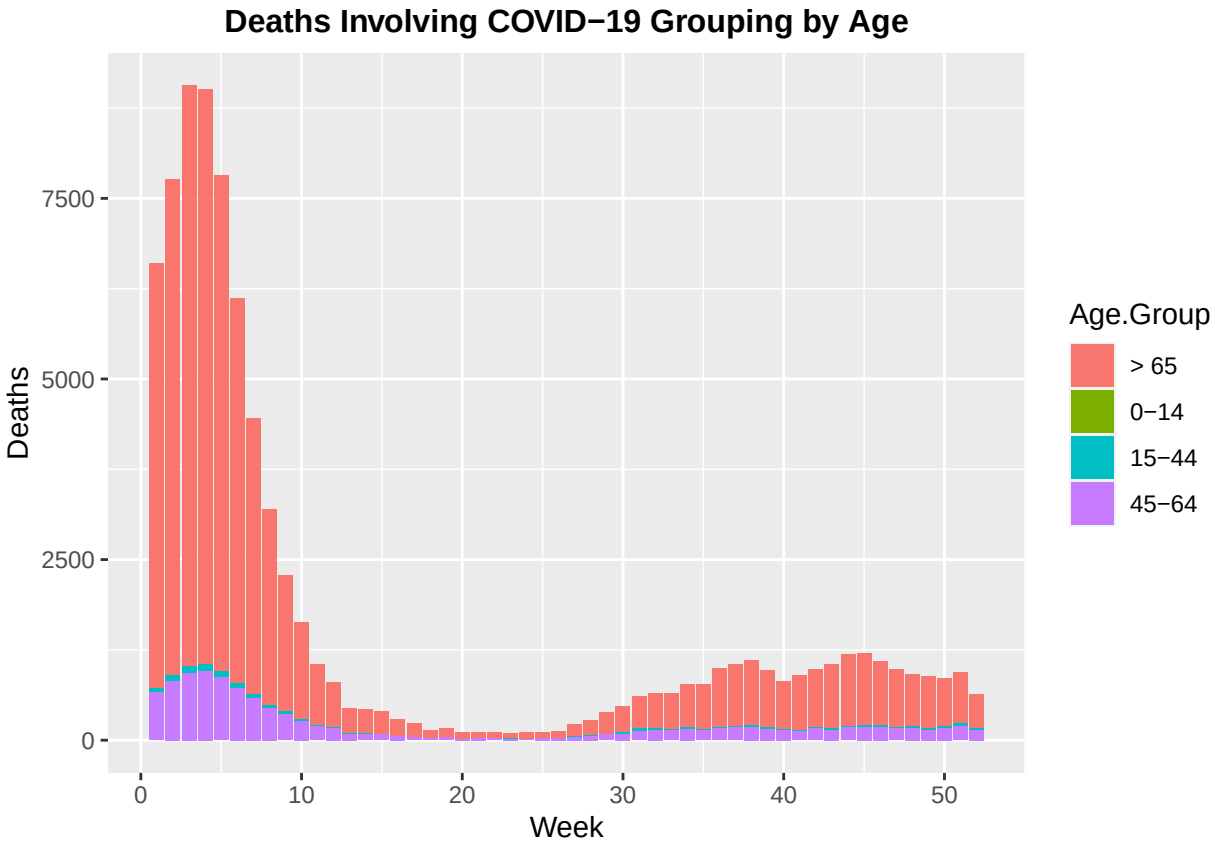
Like Visualization 1, the raw data needs some tweaking before it can be used for plotting. The difference is that **Age** will be saved as a variable.

```
head(AgeGroup_Deaths)
```

```
##   Week Age.Group Deaths
## 1    1      0-14      2
## 2    2      0-14      0
## 3    3      0-14      1
## 4    4      0-14      0
## 5    5      0-14      0
## 6    6      0-14      1
```

"fill =" and *"position = 'stack'"* enable the bar chart to be stacked vertically grouped by age.

```
p2 <- ggplot(data = AgeGroup_Deaths,
             mapping = aes(
               x =Week,
               y= Deaths,
               fill = Age.Group))+
  geom_bar(stat = 'identity', position = 'stack')+

  ggtitle("Deaths Involving COVID-19 Grouping by Age") +

  theme(plot.title = element_text(size = 12, face = "bold", hjust = 0.5))

plot(p2)
```


Deaths Involving COVID−19 Grouping by Age
Deaths
7500
5000
2500
0
0
10
20
30
40
50
Week
Age.Group
> 65
0−14
15−44
45−64

2.3.1 Pie Chart

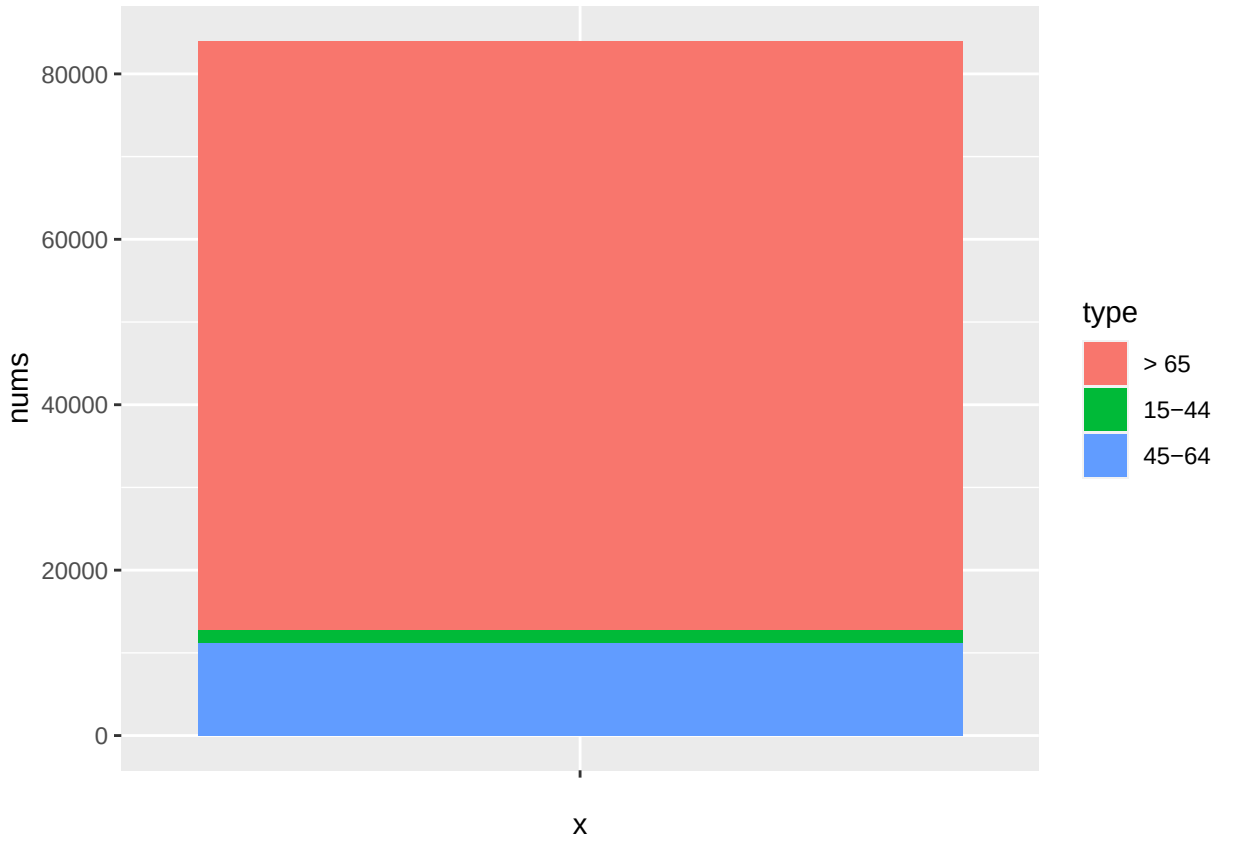
In addition to the bar chart, we can also use *pie charts* to show the proportion between different age groups. *"ggplot2" package* does not include a function to draw a pie chart directly. So I did it by drawing a vertical stacked bar and polar transformation.

After renaming, summing and transpose, a data set suitable for plotting pie charts is obtained.

```
##   Age_Group  nums
## 1     15-44  1588
## 2     45-64 11234
## 3      > 65 71185
```

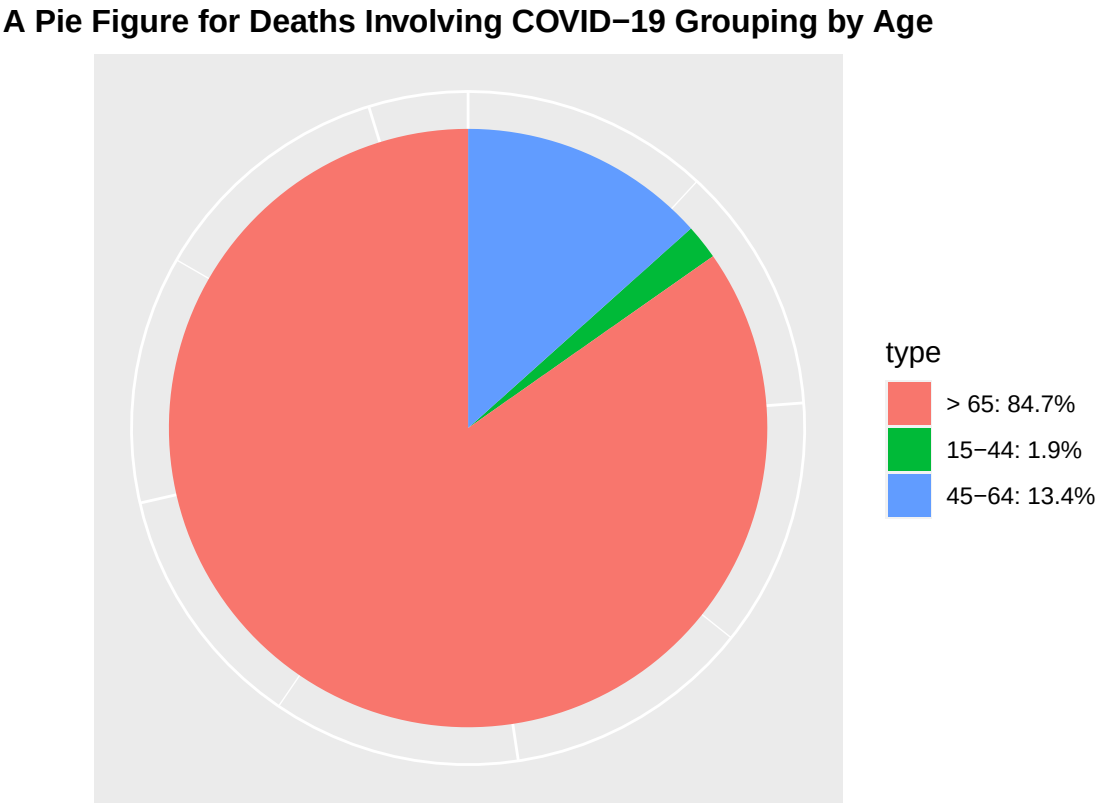
The first step is to draw a vertical stacked bar chart. The most critical of these is the adjustment of the **"width"** parameter, which is set to **1** to make the pie chart a solid circle.

```
p4_1 <- ggplot(data = Pie_Data_Set, mapping = aes(x = ' ', y = nums, fill = type)) +
  geom_bar(stat = 'identity', position = 'stack', width = 1)

plot(p4_1)
```



The next step is to do the polar transformation and remove the superfluous elements.

```
p4_1 <- p4_1 +
  coord_polar(theta = 'y') +
  labs(x = '', y = '', title = '')+

  # Remove unnecessary elements and add labels.
  theme(axis.text = element_blank())+
  theme(axis.ticks = element_blank())+

  # Add new labels
  scale_fill_discrete(labels = label)+
  ggtitle("A Pie Figure for Deaths Involving COVID-19 Grouping by Age") +
  theme(plot.title = element_text(size = 12, face = "bold", hjust = 0.5))

plot(p4_1)
```



2.4 Visualization 3 — Excess Deaths Group by Place of Occurrence

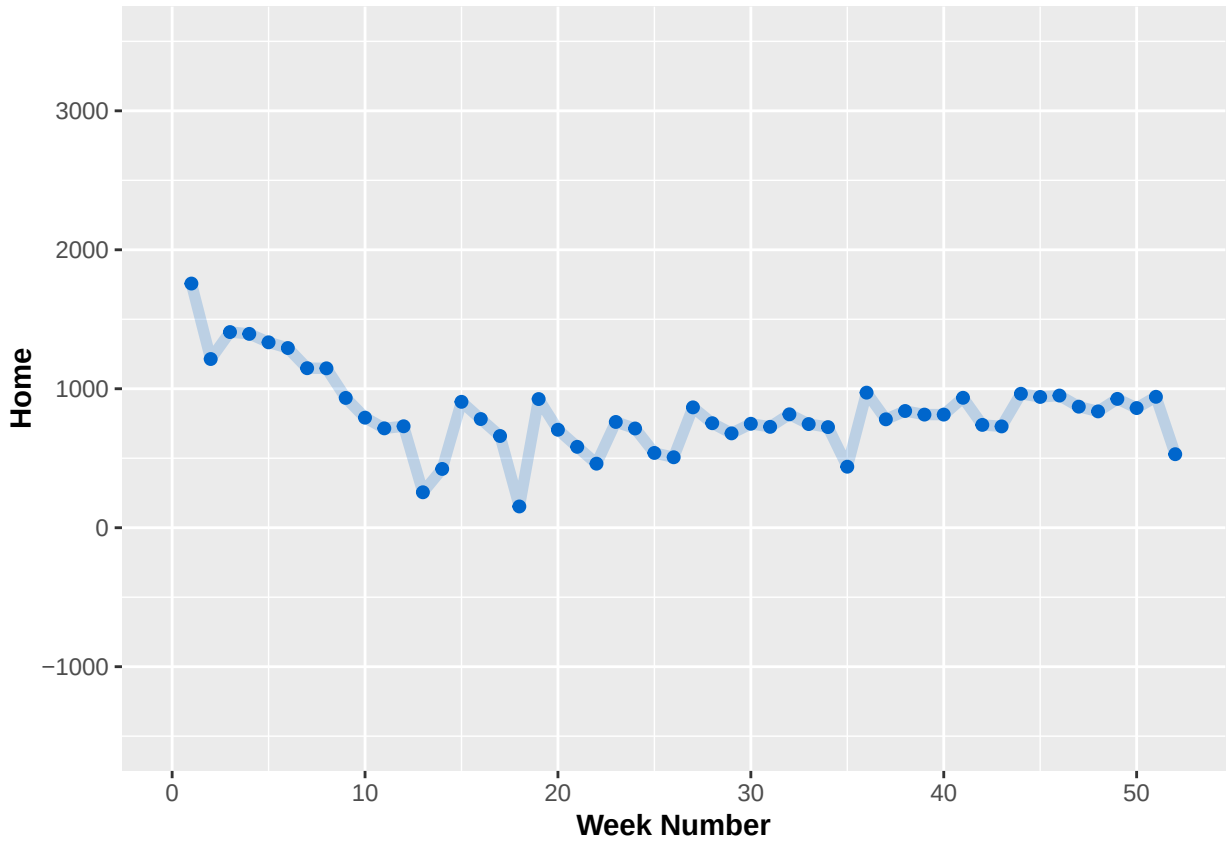
The website that published the raw data counted the number of excess deaths per week in 2021 compared to the previous five years. Excess deaths partly reflect the impact on the health of citizens and the health system at different levels.

```
data2<- read.xlsx(here("Raw_Data/publishedweek522021.xlsx"), sheet = 14)
```

```
##   Home Hospital CareHome Other Week InTotal
## 1 1757     3185      560    45    1    5547
## 2 1214     2600      405     2    2    4221
## 3 1408     3136      852    61    3    5457
## 4 1395     3221     1064     1    4    5681
## 5 1334     2740      954   -44    5    4984
## 6 1293     1834      339   -39    6    3427
```

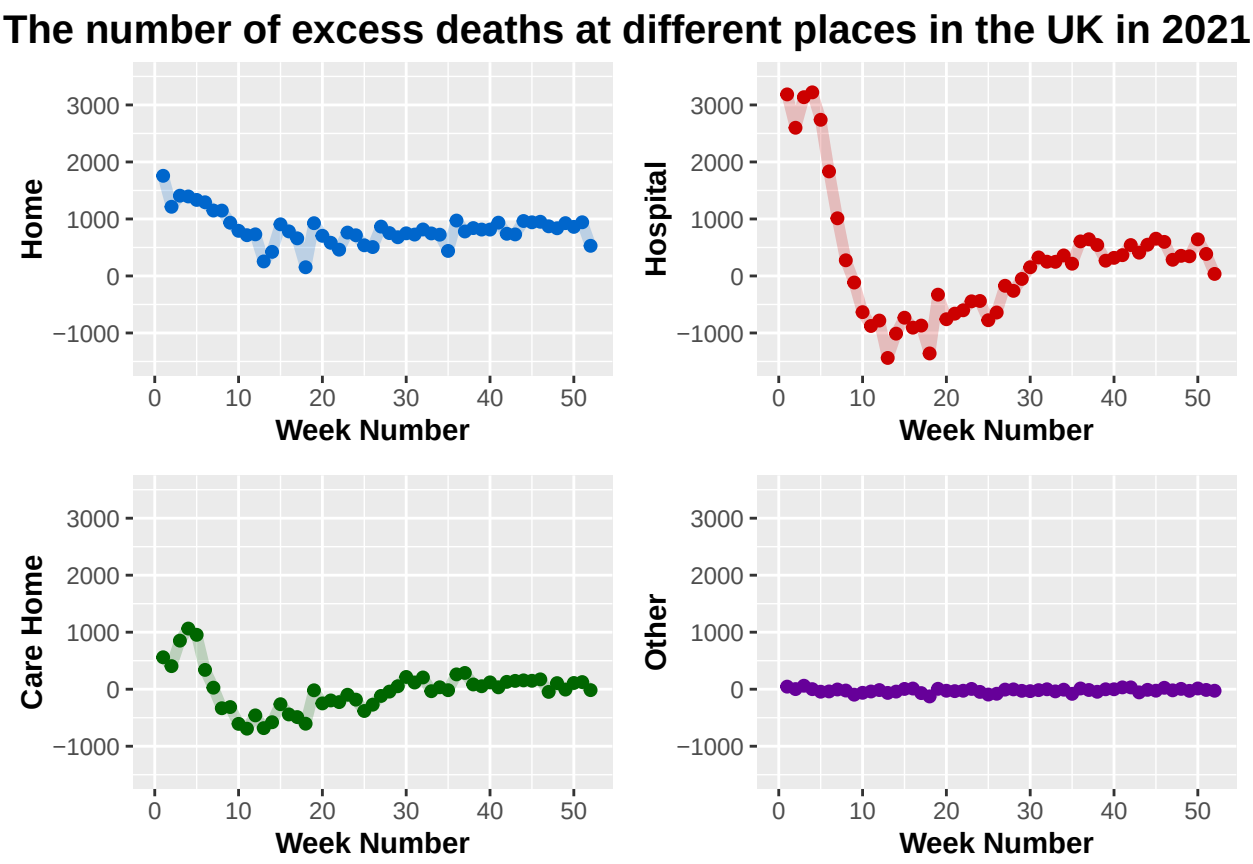
After exactly the same data processing as before, the data set can be used to draw line charts. Use the code that draws one of these diagrams as an example.I control the scope of the coordinate system by using *scale_y_continuous* and *scale_x_continuous*, so that each condition can be seen in the same frame of reference.In addition, by using hexadecimal color codes, we can give the statistical graph rich colors.

```
p3_1 <- ggplot(data = Excess_Deaths, mapping = aes(x = Week, y = Home))+
  xlab("Week Number") +
  theme(axis.title.x = element_text(size = 11, face = "bold"))+
  ylab("Home")+
  theme(axis.title.y = element_text(size = 11, face = "bold", vjust = 0.5, hjust = 0.5))+
  #and "scale_ _continuous" to control the scope of the axes
  scale_y_continuous(limits = c(-1500,3500))+
  scale_x_continuous(limits = c(0,52))+
  #Color" and "fill" define the color of the line and point.
  geom_line(size = 2, color = "#0066CC", alpha = 0.2)+
  geom_point(shape = 20, size = 2, color = "#0066CC",fill = "#0066CC", stroke = 1 )

plot(p3_1)
```


3000
2000
1000
0
−1000
Home
0
10
20
30
40
50
Week Number

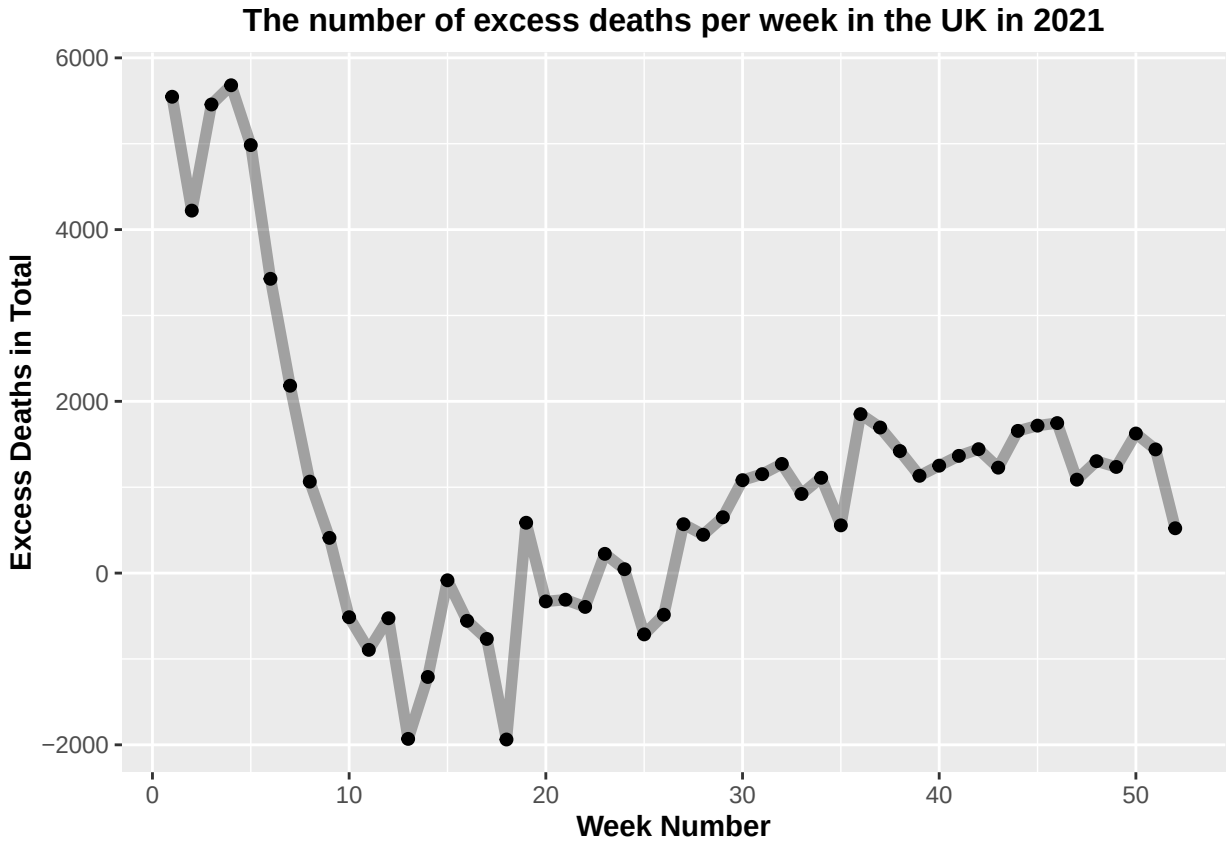
Finally use *ggarrange package* to combine 4 figures.We can control the number of images on rows and columns using *ncol* and *nrow*.

```
p3 <- ggarrange(p3_1,p3_2,p3_3,p3_4, ncol= 2, nrow = 2)+

  ggtitle("The number of excess deaths at different places in the UK in 2021") +

  theme(plot.title = element_text(size = 15, face = "bold", hjust = 0.5))

plot(p3)
```



2.5 Visualization 4 — Weekly Excess Deaths in 2021

Plotting a total figure by using the "sum" data.



3 Discussion and Conclusion

3.1 Discussion of the analysis research

People seem to have forgotten what the world was like without it, and human beings are getting used to COVID-19. This data shows us that the situation is getting better. It also shows us how devastating COVID-19 is. Data are cold, but they offer hope and a reminder of how much human sacrifice has been made.

3.2 Discussion and conclusion of this project

It was only after I had written the basic R file version that I realized that this theme had already appeared in previous assignments. But the good news is that my data is from 2021, and it covers 52 weeks. When choosing a theme, I actually struggled because the data was cruel. Every number is a life. So I take this analysis as seriously as POSSIBLE. In the process of realizing this project, I think I met a lot of difficulties. Raw data doesn't lend itself to plotting directly, so I spend a lot of energy processing it, and it takes up a lot of lines of code. And when checking the results, I also found that the total amount of data was very large, although I only selected part of the data. This reduces the readability of the final result of the visualization. The end results don't look cool, they're a little too easy. This is probably due to my technique, and there are issues I haven't solved in the un-adopted version below.

This visualization project gave me a great sense of achievement. Since I didn't have any Python experience, I was also poor at programming other software such as MATLAB. In the process of implementation, I successfully solved some technical problems by looking for mutual help information on the network. Just like the pie chart above with polar coordinates. I really want to apply what I've learned in this class elsewhere, and I have a lot of confidence now.

3.2.1 The Version Not Adopted

For the total excess death line chart, my original version filled the area with positive and negative values. However, because week data is discrete, if the data are divided according to the change of positive and negative values, separated groups will be obtained. This results in a blank fill area when the number of excess deaths in the two weeks is either positive and negative.

```
Excess_Deaths$tag <- Excess_Deaths$InTotal >= 0

cat.rle = rle(Excess_Deaths$tag)

Excess_Deaths$group = rep.int(1:length(cat.rle$lengths), times=cat.rle$lengths)

p5_1 <- ggplot(Excess_Deaths, aes(x = Week, y = InTotal, fill = tag, group = group)) +
geom_area(alpha = 0.5) +
scale_fill_manual(values = c('green','red'), guide = "none"  )

plot(p5_1)
```