GY7702 2021/22 Assignment 1

Student Number: 219031729

19/11/2021

Part 1 Part 1.1 #Loading the data into a variable, filtering it, converting it into wid format and replacing NA with 0

Part 1.2

Joining Covid_19 data with population per Local Authority District data

```
#Loading the population per Local Authority District data
ukpopest <-
  readr::read_csv("GY7702_2021-22_Assignment_1_v1-0/ukpopestimatesmid2020on2021geography.csv")
```

```
## Rows: 426 Columns: 4
```

```
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (3): Code, Name, Geography
```

```
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
#Joining Covid_19 data with population per Local Authority District data to contain all in the former
join<- covid19_vaccinations %>%
  dplyr::filter(area_name == ("East Suffolk")
                | area_name == ("Derby") | area_name == ("Cambridge")
  )  %>%
  dplyr::left_join(
    ukpopest,
    by = c("area_name" = "Name")
  ) %>%
  #remove column code becuase it is replicated
  dplyr::select(-Code)
```

```

#creating highest number of new covid cases in each LAD,
#and the new cases as a ratio of total population
 join %>%
  tidyr::pivot_wider(
    names_from = metric,
    values_from = value
  ) %>%
  #Replacing NA values in the date with 0
  tidyr::replace_na(list(new_cases = 0, cumulative_cases = 0,
                         new_first_vaccine_dose =0,
                         new_second_vaccine_dose = 0) ) %>%
  dplyr::mutate(New_cases_perc_of_pop = new_cases/`All ages` *100 ) %>%
  dplyr::select(date_reported, area_name, new_cases,New_cases_perc_of_pop) %>%
  dplyr::arrange(date_reported) %>%
  dplyr::group_by(area_name) %>%
  dplyr::slice_max(new_cases)%>%
  knitr::kable()
```

Part 2 Part 2.1

date_reported	area_name	new_cases	New_cases_perc_of_pop
2020-12-29	Cambridge	166	0.1327331
2021-01-11	Derby	335	0.1304446
2021-01-04	East Suffolk	222	0.0886677

```
# creating table for new cases in each LAD from 2021-06-24 to 2021-07-31
 gv <-  covid19_vaccinations %>%
   dplyr::filter(between( date_reported, as.Date("2021-06-24"), as.Date("2021-07-31")))%>%
   dplyr::filter(metric == "new_cases") %>%
   dplyr::filter(area_code == ("E07000244")
                 | area_code == ("E06000015") | area_code == ("E07000008")) %>%
   dplyr::select(date_reported, area_code, value)%>%
    tidyr::pivot_wider(
      names_from = area_code,
      values_from = value
    )

#Creating new variable to store present new cases value minus 7 days ago value
# creating table for new cases in each LAD from 2021-06-24 to 2021-07-31
 gv <-  covid19_vaccinations %>%
   dplyr::filter(between( date_reported, as.Date("2021-06-24"), as.Date("2021-07-31")))%>%
   dplyr::filter(metric == "new_cases") %>%
   dplyr::filter(area_name == ("East Suffolk") | area_name == ("Derby")
                 | area_name == ("Cambridge")) %>%
   dplyr::select(date_reported, area_name, value)%>%
    tidyr::pivot_wider(
      names_from = area_name,
      values_from = value
    )

#Creating new variable to store present new cases value minus 7 days ago value
mmk <- gv

 mmk$bbb <- 0
 mmk$ccc <- 0
 mmk$vvv <- 0

 # Running a loop to create table showing change in new cases as a percentage of 7 days before
for (i in 1: length(mmk$date_reported) ) {
  if(i >7){
 mmk$bbb[i] <- paste((((mmk$Derby[i] - mmk$Derby[i-7])/mmk$Derby[i-7]) *100),"%")
 mmk$ccc[i] <-  paste((((mmk$`East Suffolk`[i] - mmk$`East Suffolk`[i-7])
                        /mmk$`East Suffolk`[i-7]) *100),"%")
  mmk$vvv[i] <- paste((((mmk$Cambridge[i] - mmk$Cambridge[i-7])
                        /mmk$Cambridge[i-7]) *100),"%")
  }
  print(paste( mmk$ccc[i], mmk$bbb[i], mmk$vvv[i]))
}
```

Part 2 Part 2.2

```
## [1] "0 0 0"
## [1] "0 0 0"
## [1] "0 0 0"
## [1] "0 0 0"
## [1] "0 0 0"
## [1] "0 0 0"
## [1] "0 0 0"
## [1] "0 % 46.875 % 31.1111111111111 %"
## [1] "112.5 % 28.3018867924528 % 43.2432432432432 %"
## [1] "45.4545454545455 % 11.5384615384615 % 27.6595744680851 %"
## [1] "58.3333333333333 % 62.5 % -9.23076923076923 %"
## [1] "64.7058823529412 % -31 % -23.6111111111111 %"
## [1] "92.8571428571429 % -14.8936170212766 % -44.3037974683544 %"
## [1] "138.095238095238 % 18.9189189189189 % -35.0649350649351 %"
## [1] "216.666666666667 % -20.2127659574468 % -28.8135593220339 %"
## [1] "258.823529411765 % 23.5294117647059 % -15.0943396226415 %"
## [1] "137.5 % 79.3103448275862 % -35 %"
## [1] "157.894736842105 % 43.5897435897436 % -47.4576271186441 %"
## [1] "135.714285714286 % 136.231884057971 % -12.7272727272727 %"
## [1] "166.666666666667 % 87.5 % 40.9090909090909 %"
## [1] "120 % 122.727272727273 % 54 %"
## [1] "223.684210526316 % 129.333333333333 % 123.809523809524 %"
## [1] "72.1311475409836 % 125 % 57.7777777777778 %"
## [1] "121.052631578947 % 24.0384615384615 % 28.2051282051282 %"
## [1] "44.8979591836735 % -7.14285714285714 % 48.3870967741936 %"
## [1] "39.3939393939394 % 1.22699386503067 % 33.3333333333333 %"
## [1] "5.55555555555556 % -13.3333333333333 % 1.61290322580645 %"
## [1] "-32.7272727272727 % -40.8163265306122 % -24.6753246753247 %"
## [1] "-41.4634146341463 % -46.5116279069767 % -25.531914893617 %"
## [1] "-46.6666666666667 % -51.3227513227513 % -50.7042253521127 %"
## [1] "-45.2380952380952 % -38.7596899224806 % -8 %"
## [1] "-40.8450704225352 % -29.8076923076923 % -23.9130434782609 %"
## [1] "-18.4782608695652 % -44.8484848484849 % -6.25 %"
## [1] "-10.5263157894737 % -36.9230769230769 % 23.8095238095238 %"
## [1] "43.2432432432432 % -13.7931034482759 % 3.44827586206897 %"
## [1] "54.1666666666667 % -6.52173913043478 % 0 %"
## [1] "62.5 % 4.34782608695652 % 54.2857142857143 %"
## [1] "89.1304347826087 % 39.2405063291139 % 17.3913043478261 %"
```

```
 # Filtering out to have only the dates needed
gbbcv <- mmk %>%
dplyr::select(date_reported, ccc, bbb, vvv) %>%
dplyr::filter(between(date_reported, as.Date("2021-07-01"),
                      as.Date("2021-07-31")))

#Renaming the variable back to normal LAD names
 colnames(gbbcv)[2] = "East Suffolk"
```

```r
colnames(gbbcv)[3] = "Derby"
colnames(gbbcv)[4] = "Cambridge"
gbbcv %>%
  knitr::kable()
```

date_reported	East Suffolk	Derby	Cambridge
2021-07-01	0 %	46.875 %	31.1111111111111 %
2021-07-02	112.5 %	28.3018867924528 %	43.2432432432432 %
2021-07-03	45.4545454545455 %	11.5384615384615 %	27.6595744680851 %
2021-07-04	58.3333333333333 %	62.5 %	-9.23076923076923 %
2021-07-05	64.7058823529412 %	-31 %	-23.6111111111111 %
2021-07-06	92.8571428571429 %	-14.8936170212766 %	-44.3037974683544 %
2021-07-07	138.095238095238 %	18.9189189189189 %	-35.0649350649351 %
2021-07-08	216.666666666667 %	-20.2127659574468 %	-28.8135593220339 %
2021-07-09	258.823529411765 %	23.5294117647059 %	-15.0943396226415 %
2021-07-10	137.5 %	79.3103448275862 %	-35 %
2021-07-11	157.894736842105 %	43.5897435897436 %	-47.4576271186441 %
2021-07-12	135.714285714286 %	136.231884057971 %	-12.7272727272727 %
2021-07-13	166.666666666667 %	87.5 %	40.9090909090909 %
2021-07-14	120 %	122.727272727273 %	54 %
2021-07-15	223.684210526316 %	129.333333333333 %	123.809523809524 %
2021-07-16	72.1311475409836 %	125 %	57.7777777777778 %
2021-07-17	121.052631578947 %	24.0384615384615 %	28.2051282051282 %
2021-07-18	44.8979591836735 %	-7.14285714285714 %	48.3870967741936 %
2021-07-19	39.3939393939394 %	1.22699386503067 %	33.3333333333333 %
2021-07-20	5.55555555555556 %	-13.3333333333333 %	1.61290322580645 %
2021-07-21	-32.7272727272727 %	-40.8163265306122 %	-24.6753246753247 %
2021-07-22	-41.4634146341463 %	-46.5116279069767 %	-25.531914893617 %
2021-07-23	-46.6666666666667 %	-51.3227513227513 %	-50.7042253521127 %
2021-07-24	-45.2380952380952 %	-38.7596899224806 %	-8 %
2021-07-25	-40.8450704225352 %	-29.8076923076923 %	-23.9130434782609 %
2021-07-26	-18.4782608695652 %	-44.8484848484849 %	-6.25 %
2021-07-27	-10.5263157894737 %	-36.9230769230769 %	23.8095238095238 %
2021-07-28	43.2432432432432 %	-13.7931034482759 %	3.44827586206897 %
2021-07-29	54.1666666666667 %	-6.52173913043478 %	0 %
2021-07-30	62.5 %	4.34782608695652 %	54.2857142857143 %
2021-07-31	89.1304347826087 %	39.2405063291139 %	17.3913043478261 %

```r
# Creating function that returns weekly mean for first vaccine doses,
# for the month that with the highest first vaccine doses

H_ <- function(lad){
  H_first_dose <-
    #extracting the month with the highest daily new cases
  lubridate::ymd(
    covid19_vaccinations %>%
      dplyr::filter(area_name == lad) %>%
      dplyr::filter(metric == "new_first_vaccine_dose") %>%
      dplyr::slice_max(value) %>%
      dplyr::select(date_reported) %>%
      dplyr::pull(1, 1)
```

```
   )%>%
   month()

   #Framework function for analyzing the weekly average
   av_first_week <- covid19_vaccinations %>%
     dplyr::select(date_reported, metric, area_name, value) %>%
     dplyr::filter(area_name == lad) %>%
     dplyr::filter(metric == "new_first_vaccine_dose") %>%
     dplyr::filter(lubridate::month(date_reported)
                   == H_first_dose)%>%
     dplyr::group_by(Week_st = cut(date_reported,
                                   seq(date_reported[1],
                                       last(date_reported), by = 7))) %>%
     dplyr::group_by(Week_st)%>%
     dplyr::mutate(weeks = paste( date_reported[[1]], "to",
                                  date_reported[[1]]+ lubridate::ddays(6)))%>%
     dplyr::ungroup()%>%
     dplyr::select(weeks, value) %>%
     dplyr::group_by(weeks)%>%
     dplyr::summarise(Average_number_of_first_doses = mean(value)) %>%
     knitr::kable()
   return(av_first_week)
 }

####### Weekly first doses average for LAD Derby in the
#month with the highest first dosage
H_("East Suffolk")
```

Part 2 Part 2.3

weeks	Average_number_of_first_doses
2021-01-01 to 2021-01-07	164.2857
2021-01-08 to 2021-01-14	468.1429
2021-01-15 to 2021-01-21	2648.7143
2021-01-22 to 2021-01-28	2143.0000
2021-01-29 to 2021-02-04	2547.3333

```
####### Weekly first doses average for LAD Derby in the
#month with the highest first dosage
H_("Derby")
```

weeks	Average_number_of_first_doses
2021-01-01 to 2021-01-07	107.5714
2021-01-08 to 2021-01-14	711.2857
2021-01-15 to 2021-01-21	1097.7143
2021-01-22 to 2021-01-28	1521.7143
2021-01-29 to 2021-02-04	2796.6667

```
####### Weekly first doses average for LAD
#Cambridge in the month with the highest first dosage
H_("Cambridge")
```

weeks	Average_number_of_first_doses
2021-01-01 to 2021-01-07	65.42857
2021-01-08 to 2021-01-14	253.85714
2021-01-15 to 2021-01-21	883.57143
2021-01-22 to 2021-01-28	1140.42857
2021-01-29 to 2021-02-04	800.66667

```
#function for sum of monthly incidents in East_Suffolk
East_Suffolk_monthly_sum <- function( x){
  East_Suffolk_monthly <- covid19_vaccinations %>%
    dplyr::filter(area_name == ("East Suffolk") )%>%
    dplyr::filter(metric == x) %>%
    dplyr::mutate(monts = month(date_reported)) %>%
    dplyr::group_by(monts) %>%
    dplyr::summarise(mont_new = sum(value))
  return(East_Suffolk_monthly$mont_new)
}

#function for sum of monthly incidents in England
covid19_monthly_sum <- function(x) {
  covid19_monthly <- covid19_vaccinations %>%
    dplyr::filter(metric == x) %>%
    dplyr::mutate(monts = month(date_reported)) %>%
    dplyr::group_by(monts) %>%
    dplyr::summarise(mont_new = sum(value))
  return(covid19_monthly$mont_new)
}


#function for average monthly incidents in East_Suffolk
East_Suffolk_monthly_average <- function(x){
  East_Suffolk_monthly <- covid19_vaccinations %>%
    dplyr::filter(area_name == ("East Suffolk") )%>%
    dplyr::filter(metric == x) %>%
    dplyr::mutate(monts = month(date_reported)) %>%
    dplyr::group_by(monts) %>%
    dplyr::summarise(mont_new = mean(value))
  return(East_Suffolk_monthly$mont_new)
}

#function for average monthly incidents in England
covid19_monthly_average <- function(x) {
  covid19_monthly <- covid19_vaccinations %>%
    dplyr::filter(metric == x) %>%
    dplyr::mutate(monts = month(date_reported)) %>%
    dplyr::group_by(monts) %>%
    dplyr::summarise(mont_new = mean(value))
  return(covid19_monthly$mont_new)
}


#function for total sum of  incidents in East_Suffolk
```

```
East_Suffolk_overall_sum <- function(x){
  East_Suffolk_overall <- covid19_vaccinations %>%
    dplyr::filter(area_name == ("East Suffolk") )%>%
    dplyr::filter(metric == x) %>%
    dplyr::summarise(suum =  sum(value))
  return(East_Suffolk_overall)
}

#function for total sum of  incidents in East_Suffolk
covid19_overall_sum <- function(x){
  covid19_overall <- covid19_vaccinations %>%
    dplyr::filter(metric == x) %>%
    dplyr::summarise(suum =  sum(value))
  return(covid19_overall)
}

# #Comparing the overall number of new cases in my
#allocated LAD to overall new cases in England
East_Suffolk_overall_sum("new_cases")/
  covid19_overall_sum("new_cases")*100
```

Part 3

```
##        suum
## 1 0.2388966
```

```
# #Comparing the overall number of new_first_vaccine_dose
#in my allocated LAD to overall new cases in England
East_Suffolk_overall_sum("new_first_vaccine_dose")/
  covid19_overall_sum("new_first_vaccine_dose")*100
```

```
##        suum
## 1 0.4386245
```

```
#
# #Comparing the overall number of cumulative_cases
#in my allocated LAD to overall new cases in England
East_Suffolk_overall_sum("cumulative_cases")/
  covid19_overall_sum("cumulative_cases")*100
```

```
##        suum
## 1 0.1972975
```

```
#
# #Comparing the overall number of new_second_vaccine_dose
#in my allocated LAD to overall new cases in England
East_Suffolk_overall_sum("new_second_vaccine_dose")/
  covid19_overall_sum("new_second_vaccine_dose")*100
```

```
##       suum
## 1 0.449581
```

```
#Comparing the number of new cases in my
#allocated LAD to overall new cases in England
East_Suffolk_monthly_sum("new_cases")/
  covid19_monthly_sum ("new_cases") *100
```

```
##  [1] 0.29808792 0.21506549 0.17478942 0.27606711 0.09689279 0.05741605
## [7] 0.19040519 0.24364924 0.33165271 0.33081984 0.20513952
```

```
#Comparing the number of new_first_vaccine_dose in
#my allocated LAD to overall new cases in England
East_Suffolk_monthly_sum("new_first_vaccine_dose")/
  covid19_monthly_sum ("new_first_vaccine_dose") *100
```

```
##  [1] 0.6090893 0.5670439 0.3798733 0.3008464 0.5142684 0.1939472 0.2452393
##  [8] 0.3094109 0.4142719 0.1646105 0.2675719
```

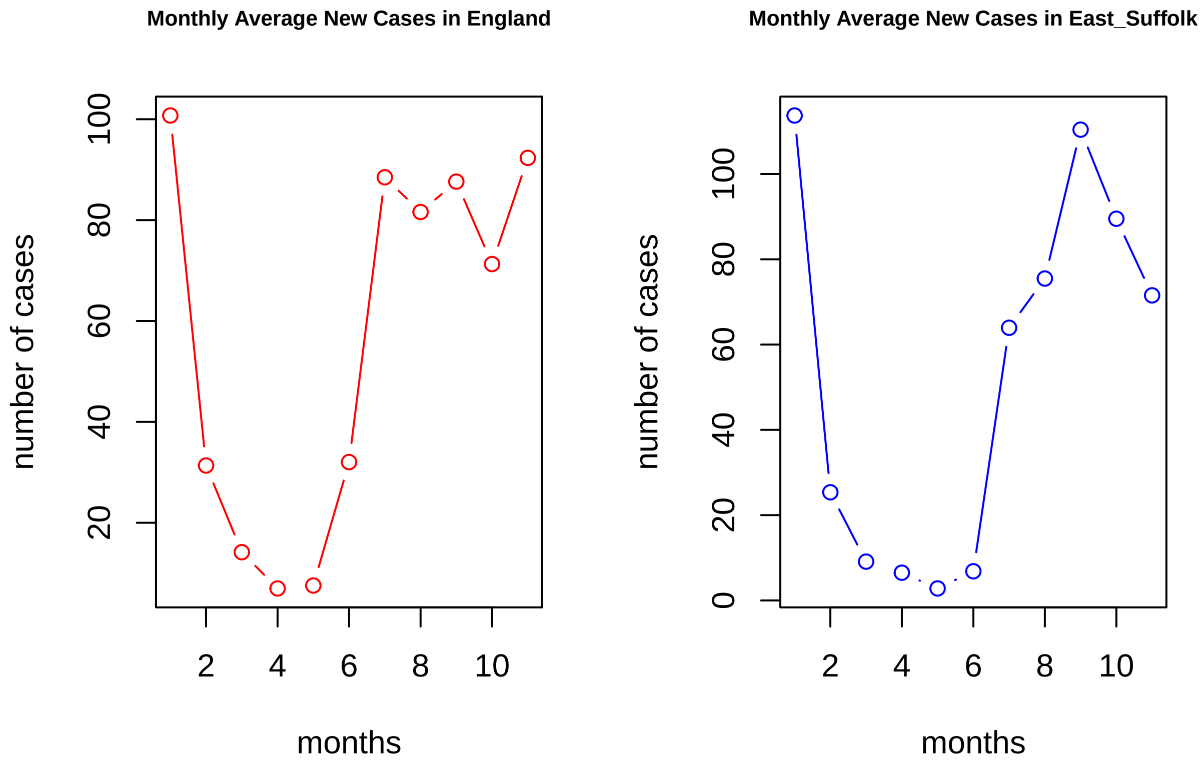
```
#Comparing the number of cumulative_cases in my
#allocated LAD to overall new cases in England
East_Suffolk_monthly_sum("cumulative_cases")/
  covid19_monthly_sum ("cumulative_cases") *100
```

```
##  [1] 0.1911635 0.2039952 0.2031716 0.2029686 0.2032442 0.1978306 0.1872246
##  [8] 0.1948550 0.2073642 0.2162235 0.1446534
```

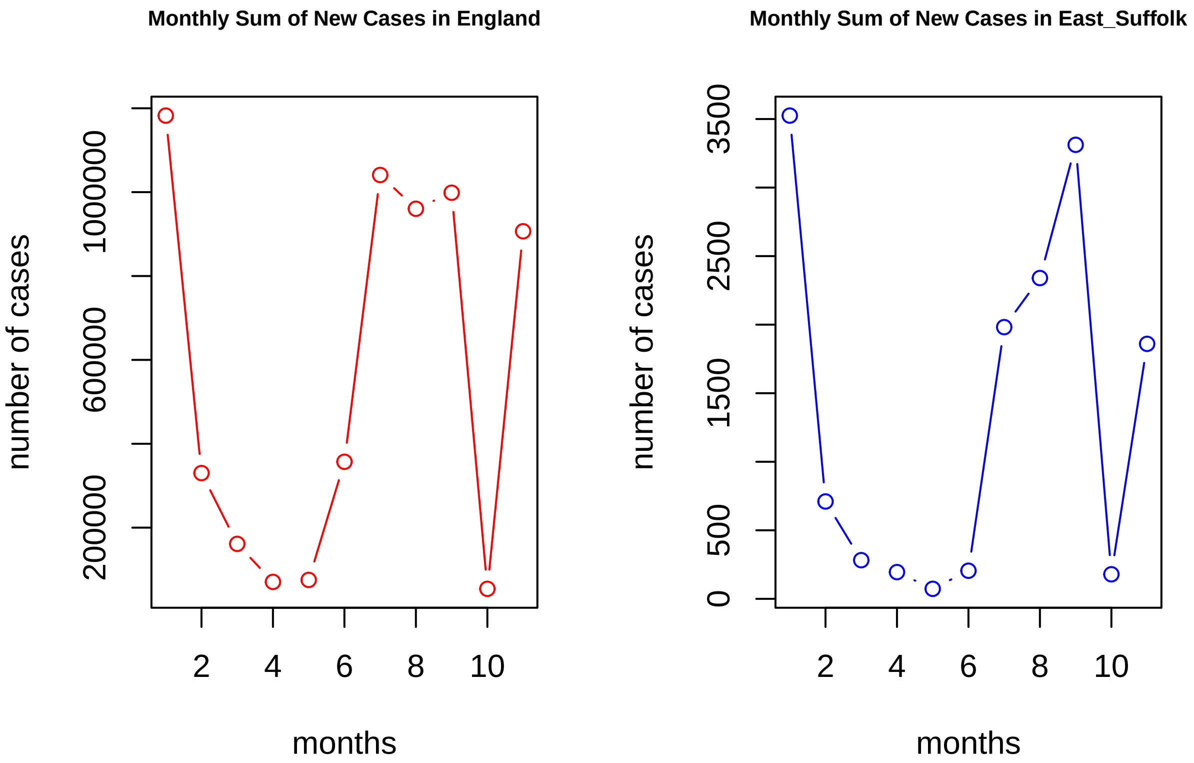
```
#Comparing the number of new_second_vaccine_dose
#in my allocated LAD to overall new cases in England
East_Suffolk_monthly_sum("new_second_vaccine_dose")/
  covid19_monthly_sum ("new_second_vaccine_dose") *100
```

```
##  [1] 0.34242218 0.46853468 0.34042982 0.68020629 0.46601863 0.32791763
##  [7] 0.44684948 0.24556840 0.27687352 0.33922151 0.06192498
```

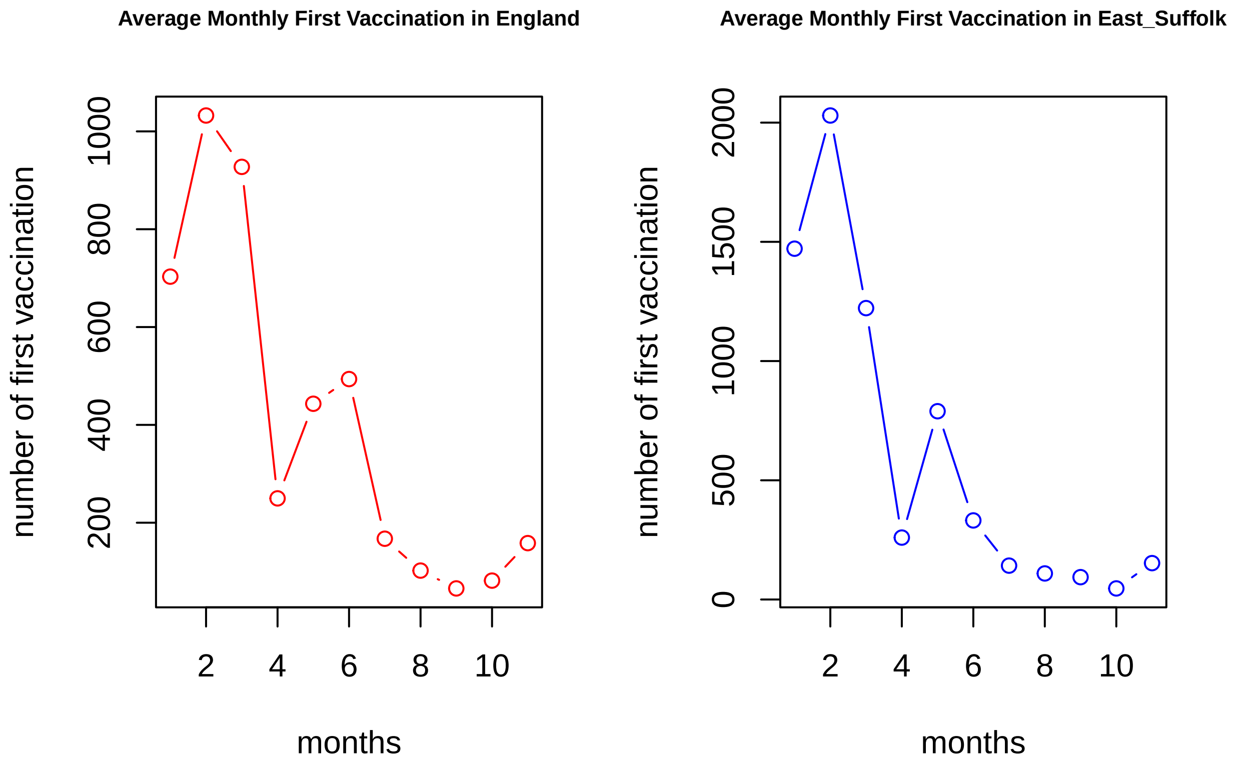
```
#Figure 1: Creating side by side plot of monthly
#average new cases in my allocated LAD and in England
options(scipen = 1000)
par(mfrow = c (1, 2))
plot(covid19_monthly_average ("new_cases"), col = "red",
     type = "b", main = "Monthly Average New Cases in England",
     xlab = "months", ylab = "number of cases", cex.main=0.7)
plot(East_Suffolk_monthly_average("new_cases"), col = "blue",
     type = "b", main = "Monthly Average New Cases in East_Suffolk",
     xlab = "months", ylab = "number of cases", cex.main=0.7)
```



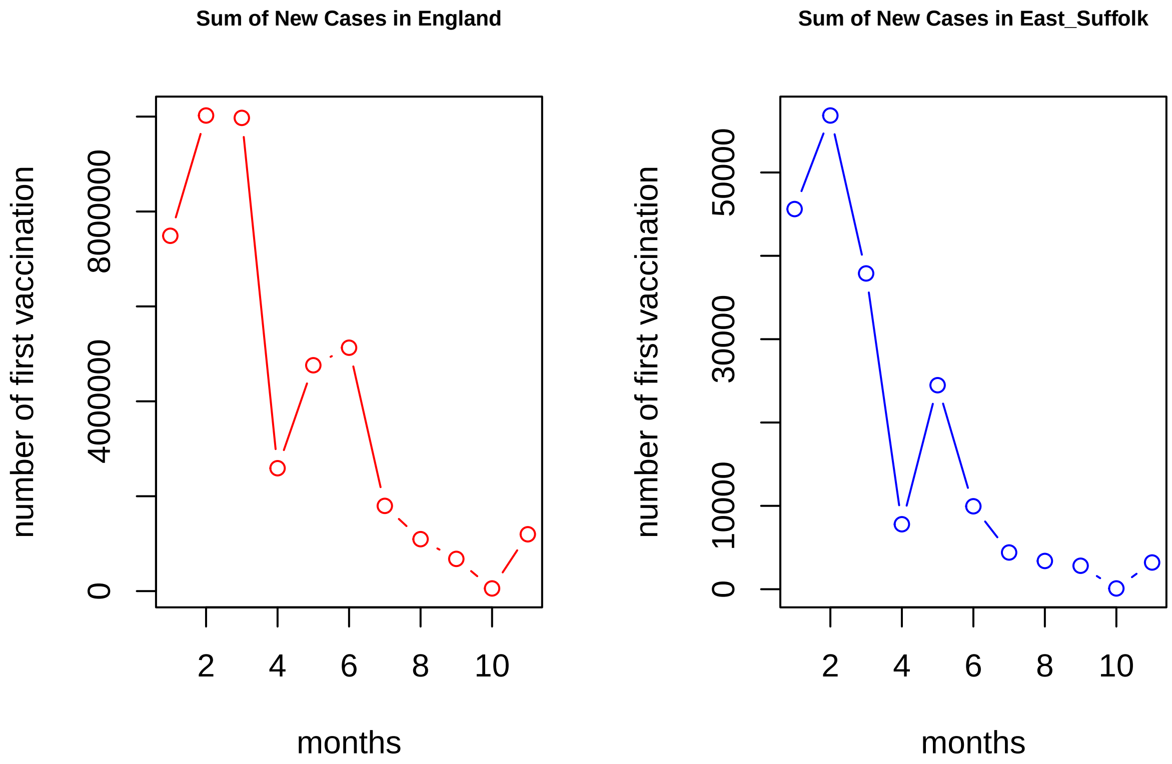
```
#Figure 2: Creating side by side plot of monthly sum of new
#cases in my allocated LAD and in England
par(mfrow = c (1, 2))
plot(covid19_monthly_sum ("new_cases"), col = "red", type = "b",
     main = "Monthly Sum of New Cases in England",
     xlab = "months", ylab = "number of cases", cex.main=0.7)
plot(East_Suffolk_monthly_sum("new_cases"), col = "blue", type = "b",
     main = "Monthly Sum of New Cases in East_Suffolk",
     xlab = "months", ylab = "number of cases", cex.main=0.7)
```



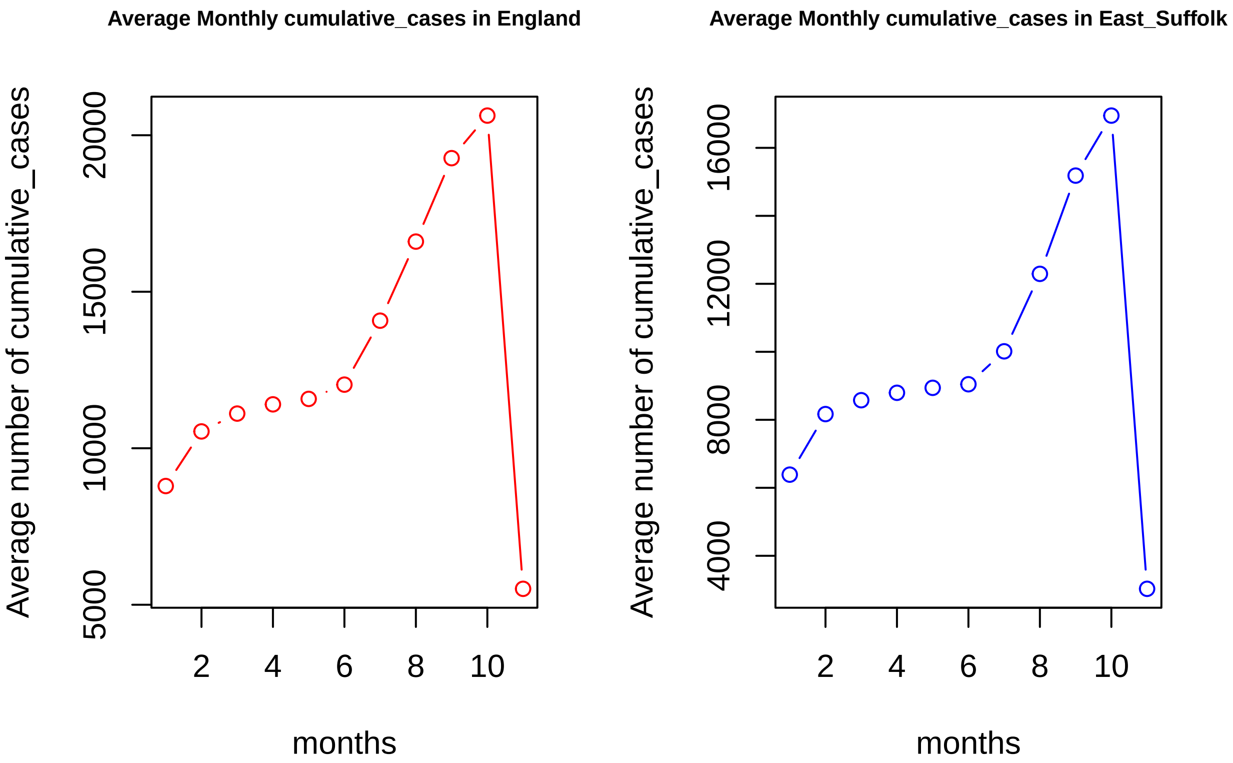
```
#Figure 3: Creating side by side plot of monthly average new
#first vaccination collected in my allocated LAD and in England
par(mfrow = c (1, 2))
plot(covid19_monthly_average ("new_first_vaccine_dose"), col = "red", type = "b",
     main = "Average Monthly First Vaccination in England", xlab = "months",
     ylab = "number of first vaccination", cex.main=0.7)
plot(East_Suffolk_monthly_average("new_first_vaccine_dose"),
     col = "blue", type = "b",
     main = "Average Monthly First Vaccination in East_Suffolk",
     xlab = "months", ylab = "number of first vaccination", cex.main=0.7)
```



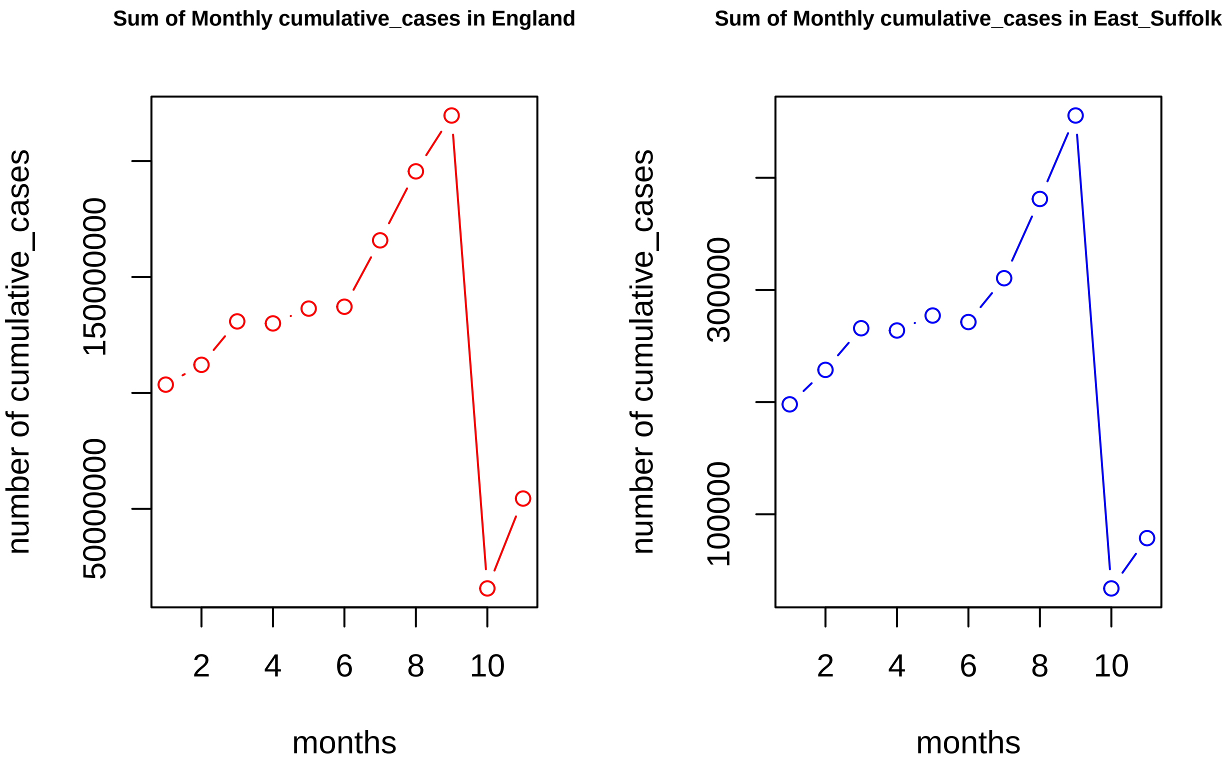
```
#Figure 4: Creating side by side plot of monthly sum of new
#first vaccination collected in my allocated LAD and in England
par(mfrow = c (1, 2))
plot(covid19_monthly_sum ("new_first_vaccine_dose"), col = "red", type = "b",
     main = "Sum of New Cases in England", xlab = "months",
     ylab = "number of first vaccination", cex.main=0.7)
plot(East_Suffolk_monthly_sum("new_first_vaccine_dose"), col = "blue", type = "b",
     main = "Sum of New Cases in East_Suffolk", xlab = "months",
     ylab = "number of first vaccination", cex.main=0.7)
```



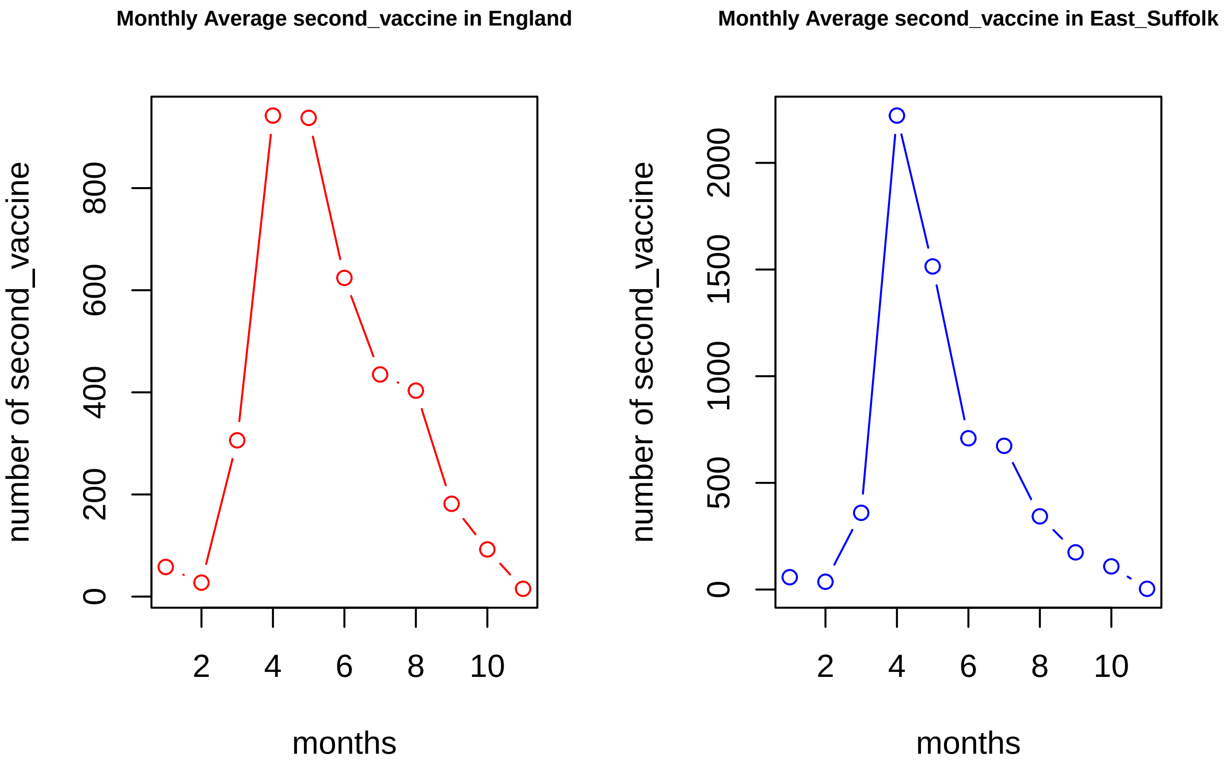
```
#Figure 5: Creating side by side plot of monthly average
#cumulative COVID cases in my allocated LAD and in England
par(mfrow = c (1, 2))
plot(covid19_monthly_average ("cumulative_cases"), col = "red", type = "b",
     main = "Average Monthly cumulative_cases in England", xlab = "months",
     ylab = "Average number of cumulative_cases", cex.main=0.7)
plot(East_Suffolk_monthly_average("cumulative_cases"), col = "blue", type = "b",
     main = "Average Monthly cumulative_cases in East_Suffolk", xlab = "months",
     ylab = "Average number of cumulative_cases", cex.main=0.7)
```



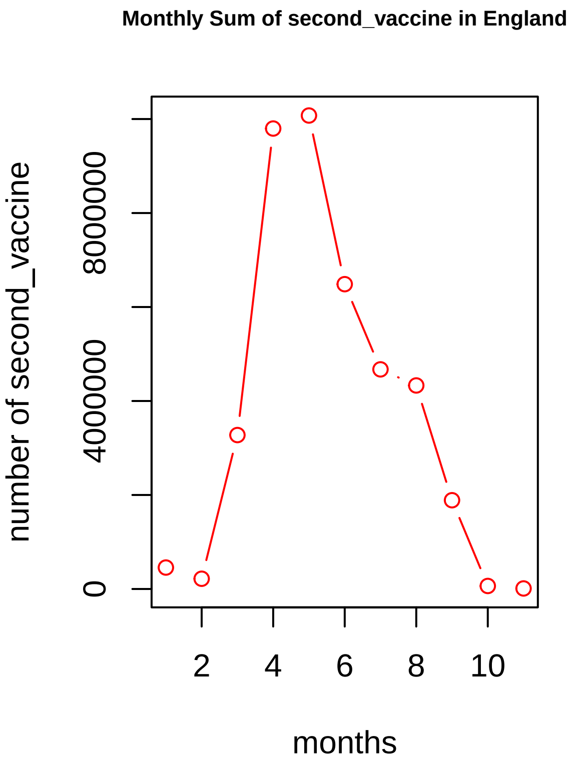
```
#Figure 6: Creating side by side plot of monthly sum
#of cumulative COVID cases in East_Suffolk and England
par(mfrow = c (1, 2))
plot(covid19_monthly_sum ("cumulative_cases"), col = "red", type = "b",
      main = "Sum of Monthly cumulative_cases in England", xlab = "months",
     ylab = "number of cumulative_cases", cex.main=0.7)
plot(East_Suffolk_monthly_sum("cumulative_cases"), col = "blue", type = "b",
      main = "Sum of Monthly cumulative_cases in East_Suffolk", xlab = "months",
     ylab = "number of cumulative_cases", cex.main=0.7)
```



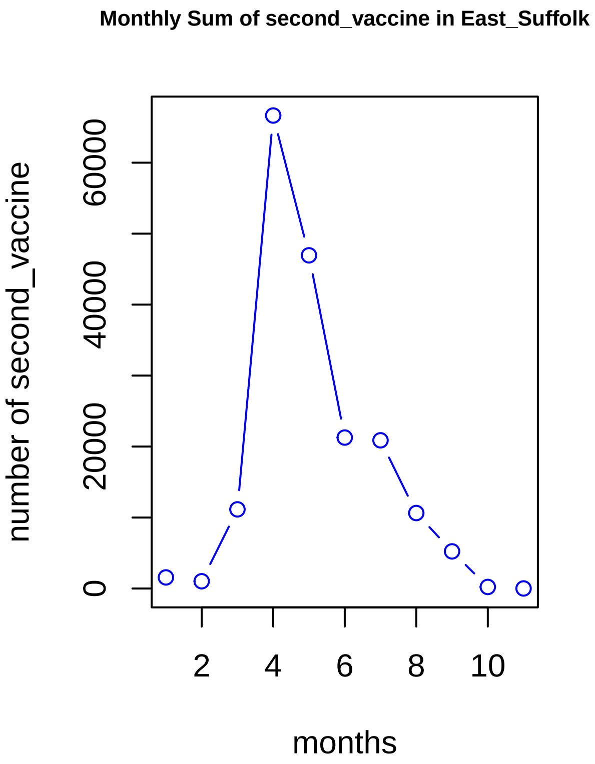
```
#Figure 7: Creating side by side plot of monthly average of
#new second vaccination collected in my allocated LAD and in England
par(mfrow = c (1, 2))
plot(covid19_monthly_average ("new_second_vaccine_dose"), col = "red", type = "b",
     main = "Monthly Average second_vaccine in England", xlab = "months",
     ylab = "number of second_vaccine", cex.main=0.7)
plot(East_Suffolk_monthly_average("new_second_vaccine_dose"), col = "blue", type = "b",
     main = "Monthly Average second_vaccine in East_Suffolk",
     xlab = "months", ylab = "number of second_vaccine", cex.main=0.7)
```



```
#Figure 8: Creating side by side plot of monthly sum of new second vaccination collected in East_Suffol
options(scipen = 1000)
par(mfrow = c (1, 2))
plot(covid19_monthly_sum ("new_second_vaccine_dose"),
     col = "red", type = "b", main = "Monthly Sum of second_vaccine in England",
     xlab = "months", ylab = "number of second_vaccine", cex.main=0.7)
plot(East_Suffolk_monthly_sum("new_second_vaccine_dose"),
     col = "blue", type = "b", main = "Monthly Sum of second_vaccine in East_Suffolk",
     xlab = "months", ylab = "number of second_vaccine", cex.main=0.7)
```



Monthly Sum of second_vaccine in England



Monthly Sum of second_vaccine in East_Suffolk

The data a covid data for the entire England, consisting of 380 LAD in England. I am required to compared the incidents or activity in one the LAD assigned to me (East_Suffolk).

To compare both areas, I built a two functions using dyplyr library functions to analyze the cases to avoid replication. The input argurmnet are new cases, new first covid vaccine and others.

For Visual comparison, line charts were also built.

The total number of cases in East_Suffolk is about 0.24% of the total new covid cases in England. The number of new vaccines given in East_Suffolk represents 0.4% of all vaccines given in England for the whole period. Result shows that East_Suffolk is responsible about 0.2% of all cumulative covid cases in the UK. 0.4% of all second vaccines in ENgland was administered in East_Suffolk.

According to figure 1, the highest average new cases was witnessed in January 2021 in East_Suffolk and in England, and the lowest was seen around May for both regions.

According to figure 2, the highest total covid new cases was seen in January 2021 for both East_Suffolk and entire England. The sum of covid cases monthly bottomed to lowest in May and October 2021 for both regions.

Although, there are differences in the trend for average monthly first vaccine doses in the entire England and East_Suffolk, both regions administered the highest in February 2021, and lowest in September 2021, figure 3 shows.

Figure 4 shows the trend of first vaccines given in both regions for the period.

Figure 5 and 6, shows the trend of the average number of cumulative cases and the total number of cumulative cases in the two regions which are the same.

In figure 7 and 8, shows the trend of second vaccine dosage administered bith regions.

Part 4 Thanks to some of the practical classes we did and support from online platforms, I was able to produce this analysis and document. In order to achieve the required task, I use several based, tidyverse and dplyr library functions, among others. Several analysis were performed including filtering out, selection, converting table to wide or long format and joining of two data. Also, I built functions and group by select by specific criterion. It took me several weeks to complete the task becuase i was doing it in badges, and I had problem in some of the phases so I had to search on internet.

References

1. dates?, H., Borck, C. and Cardoso, P., 2021. How to get week numbers from dates?. [online] Stack Overflow. Available at: https://stackoverflow.com/questions/22439540/how-to-get-week-numbers-from-dates [Accessed 19 November 2021].

2. ggplot2, S., DuBois, C., LeBauer, D. and Wilke, C., 2021. Side-by-side plots with ggplot2. [online] Stack Overflow. Available at: https://stackoverflow.com/questions/1249548/side-by-side-plots-with-ggplot2 [Accessed 19 November 2021].

3. notation?, H. and Epskamp, S., 2021. How to disable scientific notation?. [online] Stack Overflow. Available at: https://stackoverflow.com/questions/5352099/how-to-disable-scientific-notation [Accessed 19 November 2021].

4. Rdocumentation.org. 2021. seq function - RDocumentation. [online] Available at: https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/seq [Accessed 19 November 2021].