Health Status of Residents in European and the Impact of Covid-19 Sequelae

Introduction:

Post-acute sequelae of COVID-19 (PASC), also known as Long COVID syndrome, is the persistence of COVID-19 symptoms long after viral infection (Lamontagne et al,2021). More than one year since its emergence, corona virus disease 2019 (COVID-19) is still looming large with a paucity of treatment options. To add to this burden, a sizeable subset of patients who have recovered from acute COVID-19 infection have reported lingering symptoms, leading to significant disability and impairment of their daily life activities. These patients are considered to suffer from what has been termed as "chronic" or "long" COVID-19 or a form of post-acute sequelae of COVID-19, and patients experiencing this syndrome have been termed COVID-19 long-haulers. Despite recovery from infection, the persistence of atypical chronic symptoms, including extreme fatigue, shortness of breath, joint pains, brain fogs, anxiety, and depression, that could last for months implies an underlying disease pathology that persist beyond the acute presentation of the disease (Ramakrishnan et al,2021)

The purpose of this study was to find out which sequelae can significantly impact physical health in infected individuals. In addition, this study explored whether medication use, and hospitalization could affect the health of infected individuals.

This study will obtain the data required for the study from SHARE data and perform data analysis by using R programming. This study will focus on the following sections: Data cleaning, EDA, Visualization, Modelling, and resampling method.

In the modelling part of the research, our selected variable as below:

Dependent Variable:

cah102_ Health: change in health in the last 3 months

Independent Variable:

cac120_1 COVID-19: fatigue attributed to respondent's Covid illness

cac120_2 COVID-19: cough, congestion, shortness of breath attributed to respondent's Covid

cac120_3 COVID-19: loss of taste or smell attributed to respondent's Covid illness

cac120_4 COVID-19: headache attributed to respondent's Covid illness

cac120_5 COVID-19: body aches, joint pain attributed to respondent's Covid illness

cac120_6 COVID-19: chest or abdominal pain attributed to respondent's Covid illness

cac120_7 COVID-19: diarrhea, nausea attributed to respondent's Covid illness

cac120_8 COVID-19: confusion attributed to respondent's Covid illness

At the same time, we will study the health changes and health status of the respondents (different place of residence, gender, country, age). Mainly use Crosstab and ggplot to display.

Data Cleaning

```
#Plot Package to provide interactivity to ggplot2 charts

library(tidyverse)

library(plotly)

data <- as_tibble(read_csv("share8.csv"))

# head(data)
```

We have also converted the data dictionary as csv so that we were able to quickly lookup and identify the definitions of the column variables.

```
#Read Dictionary CSV and Clean Trim all the Columns
dictionary <- read_csv("DataDictionary/SHARE8DataDictionary-converted.csv")

dictionary$Varname <-str_trim(dictionary$Varname, side = c("both"))
dictionary$Label <-str_trim(dictionary$Label, side = c("both"))
dictionary$Categories <-str_trim(dictionary$Categories, side = c("both"))
# head(dictionary)
```

With importing the data dictionary into a tibble, we also created this custom function below to identify definition for our data.

```
# This is a function to lookup and retrieve details of a variable

Vardetails <- function(column,details){
    tryCatch(
    expr = {
    as.character(column)
    a <- colnames(data[,column])
    },
    error = function(e){
        message("Vardetails Function: Column Not Found, Please Input a single
 valid variable name i.e.- 'mergeid'")
    }
    )
    tryCatch(
    expr = {
                if(missing(details)) {
                    as.character(dictionary[match(colnames(data[,column]),dic
tionary$Varname),2])
                } else {
```

```
                    if(details == 1){
                        as.character(dictionary[match(colnames(data[,colum
n]),dictionary$Varname),2])
                    } else if (details == 2) {
                        as.character(dictionary[match(colnames(data[,colum
n]),dictionary$Varname),3])
                    } else {
                        message("Vardetails Function: Error - Opt Arg: 1 for
Label, 2 for Category ")
                    }
                }
            }
)
}

#Example Usage of the Var detail's function
Vardetails('mergeid')

## [1] "Person identifier (fix across modules and waves)"

Vardetails('cait104_',1)

## [1] "Social: use of internet for e-mailing, etc. since outbreak"

Vardetails('cait104_',2)

## [1] "Social"
```

General Tidying:

Prior to starting off our exploratory data analysis process, we were able to identify specific rules and methods in order to clean the data in general. We will start off by 1) tidying missing codings and 2) Transforming Binary Values

1) General Tidying: Missing Codes

Based on the share8 pdf documentation provided. Missing Codes are generally tagged as a negative coded value. ~page 16 of SHARE_release_guide_8-0-0.pdf.

With this notion, we are able to filter all negative coded value in the raw file first, to identify whether it is approapriate to be converted to "NA".

```
#Identify the index for Coded Variables In The File
CodedVar <- c(9,11:308)
```

We will start by creating an empty tibble, next we looped through the coded variable and filter for unique negative coded values. The results are appended onto the empty tibble.

```
#Initiating an empty tibble
tbl_colnames = c('variable','value')
```

```
UniqueValues=as_tibble(matrix(nrow = 0, ncol = length(tbl_colnames)), .name_r
epair = ~ tbl_colnames)
UniqueValues=UniqueValues %>%
  mutate(across(everything(), as.character))
UniqueValues %>% summarise_all(class)
```

```
## # A tibble: 1 x 2
##   variable  value
##   <chr>     <chr>
## 1 character character
```

```
#Looping Though Coded Variables
for (val in CodedVar)
    {

#Identify the unique values in individual columns, subseqeuntly filter for ne
gative codes with regular expression. "-\\d\\."
    B <- unique(data[, val]) %>%
      filter(grepl("-\\d\\.",!!as.symbol(colnames(data[, val]))))

#Pivot the data into 2 columns and unite to the main tibble
    C <- pivot_longer(B, cols =colnames(B), names_to = "variable", values_to
= "value")
    UniqueValues <- union(UniqueValues,C)
}
```

Unique Negative Coded Values can be seeing in the tibble below. We only found 3 negative coded variables in the file.

```
# head(UniqueValues)
unique(UniqueValues[, 'value'])
```

```
## # A tibble: 3 x 1
##   value
##   <chr>
## 1 -2. Refusal
## 2 -1. Don't know
## 3 -9. Not applicable (qqn routing)
```

It is appropriate to convert these negative coding to “NA”. We then assigned the negative codes identified to a “MissingCoding” List. Additionally, we have also identified other convention in the raw data that is indicating NA result, (‘Not applicable (qnn routing)’,‘99. Not applicable’) these values are also added to the MissingCoding List.

```
NegativeCoding <- unique(UniqueValues[,2])
MissingCoding <- c(NegativeCoding$value,'Not applicable (qnn routing)','99. N
ot applicable')
MissingCoding
```

```
## [1] "-2. Refusal"                       "-1. Don't know"
## [3] "-9. Not applicable (qqn routing)" "Not applicable (qnn routing)"
## [5] "99. Not applicable"
```

We will now 1) look through the coded variable and 2) convert any values that is in the missing coding list into "NA"

```
#Looping Though The Coded Variables
for (val in CodedVar)
    {
    #Replace Any Missing Coding to NA
    for (x in MissingCoding)
    {

        data[, val][data[, val] == as.symbol(x)] <- NA

        }
}
```

We have transformed the negative coded and other applicable values into NA. Now we will proceed to step 2, which is the handling of binary values.

2. General Tidying: Binary Values.

Prior to delving into the analysis, we can also identify and transform the columns containing only binary values into 1s and 0s. We can I identify binary values by starting with the function below.

```
#Function identify Columns containing only 2 unique values after removing NAs
is.binary <- function(v) {
  x <- length(na.omit(unique(v)))==2
}

#The function can be vapply on the data to return us TRUE and FALSE results f
or each columns
B <- vapply(data, is.binary, logical(1))
B

Example of output result

##     mergeid     hhid9ca mergeidp9ca coupleid9ca     country language_ca
##       FALSE       FALSE       FALSE       FALSE       FALSE       FALSE
##      exrate    cadn042_    cadn002_    cadn003_    cas140_    caho100_
##       FALSE        TRUE       FALSE       FALSE       FALSE        TRUE
##    caho037_    caho136_    caho032_    caph003_    cah102_    cah004_1
##       FALSE       FALSE       FALSE       FALSE       FALSE        TRUE
##    cah004_2    cah004_3    cah004_4    cah004_5    cah004_6    cah004_7
##        TRUE        TRUE        TRUE        TRUE        TRUE        TRUE
##    caph105_   caph089_1   caph089_2   caph089_3   caph089_4     cah006_
##       FALSE        TRUE        TRUE        TRUE        TRUE        TRUE
```

Subsequently we can filter for the column index which have only unique 2 values and assigned it to a new list.

```
C <- B %>% grep(TRUE,.)
length(C)
```

```
## [1] 192
```

```
#Removing Gender Variable
BinaryVar <- C[-1]
BinaryVar
```

```
##[1]  12  18  19  20  21  22  23  24  26  27  28  29  30  31  32  33  34  35
##[19] 36  37  38  47  49  50  52  53  54  56  57  59  60  62  65  66  67  68
##[37] 69  70  71  72  73  74  82  83  84  85  86  87  88  89  90  91  99 100
##[55]101 102 103 104 105 106 107 108 109 111 112 113 114 115 116 117 118 119
##[73]120 121 122 129 130 131 132 133 134 135 136 137 138 139 143 145 146 147
##[91]148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165
##[109]166167 168 170 171 172 173 174 175 176 177 179 180 181 182 183 184 190
##[127]192194 195 196 197 200 203 205 206 209 211 212 213 214 215 216 217 218
##[145]219220 221 222 223 226 228 231 233 234 235 236 237 238 239 240 241 242
##[163]245257 258 259 260 265 266 267 268 273 275 276 277 278 283 284 285 286
##[181]287 288 290 294 295 296 297 298 299 300 301
```

Similar to the approach earlier for missing code, we will 1) initiate an empty tibble, 2) loop through and unite unique values in the binary columns. In order to identify whether it is approapriate to transform the data.

```
#Initiating an empty tibble
tbl_colnames = c('variable','value')
BinaryValues=as_tibble(matrix(nrow = 0, ncol = length(tbl_colnames)), .name_r
epair = ~ tbl_colnames)
BinaryValues=BinaryValues %>%
  mutate(across(everything(), as.character))
BinaryValues %>% summarise_all(class)
```

```
## # A tibble: 1 x 2
##   variable  value
##   <chr>     <chr>
## 1 character character
```

```
#Loop Though the Binary Variables list with only 2 values
for (val in BinaryVar)
    {

# Unite all the unique results into a new tibble "BinaryValues"
    B <- unique(data[, val])

    C <- pivot_longer(B, cols =colnames(B), names_to = "variable", values_to
= "value")
    BinaryValues <- union(BinaryValues,C)
```

```
}
BinaryValues <- BinaryValues%>% filter(!is.na(BinaryValues$value))
head(BinaryValues)
```

```
## # A tibble: 6 x 2
##   variable value
##   <chr>    <chr>
## 1 caho100_ 1. Yes
## 2 caho100_ 5. No
## 3 cah004_1 5. No
## 4 cah004_1 1. Yes
## 5 cah004_2 5. No
## 6 cah004_2 1. Yes
```

We can now export the resulting tibble to csv to explore in excel whether the columns should be transformed.

```
# write.csv(BinaryValues, file = "BinaryValues.csv", row.names =FALSE)
```

We are also able to aggregate and count all the occurence of elements in the binary columns.

```
BinElements <- BinaryValues  %>%
  count(value, sort = TRUE)
BinElements
```

```
## # A tibble: 14 x 2
##    value                                                  n
##    <chr>                                              <int>
##  1 1. Yes                                                90
##  2 5. No                                                 90
##  3 0. Not selected                                       83
##  4 1. Selected                                           83
##  5 2020                                                   9
##  6 2021                                                   9
##  7 1                                                      4
##  8 2. About the same                                      4
##  9 1. Less so                                             3
## 10 2                                                      3
## 11 1. Trouble with sleep or recent change in pattern      1
## 12 2. No trouble sleeping                                 1
## 13 3. More often                                          1
## 14 30                                                     1
```

By exploring the csv exported and the count of elements above, we decided to only transformed the binary variables column containing the elements ("1. Yes", "5. No", "0. Not selected", "1. Selected").

```
#Excluding Element that should not be transformed
ExcludeBin <- BinElements%>% filter(BinElements$n<10)
ExcludeBin$value
```

```
##  [1] "2020"
##  [2] "2021"
##  [3] "1"
##  [4] "2. About the same"
##  [5] "1. Less so"
##  [6] "2"
##  [7] "1. Trouble with sleep or recent change in pattern"
##  [8] "2. No trouble sleeping"
##  [9] "3. More often"
## [10] "30"
```

```
#Subsequently Identifying Binary Column Not Eligible or Advisable for Data Tr
ansformation
ExcludeBinCols <- BinaryValues %>% filter(value%in%ExcludeBin$value) %>% sele
ct(variable) %>% {unique(.$variable)}
ExcludeBinCols
```

```
##  [1]"cah121_2" "camh113_2" "camh007_"  "camh118_2" "cac111_3b" "cac114_3b"
##  [7]"cac114_4b""cac114_8b" "caep101_1" "caw123_2"  "caw123_4"  "caw126_2"
##  [13]"caw126_4" "cae106_2"  "cae106_4"  "cae108_2"  "cae108_4"  "cas131_2"
```

From the BinaryValues tibble created earlier, we are able to assign all the variable into a new list.

```
BinCols <- BinaryValues %>% {unique(.$variable)}
```

By Comparing the set different between all binary columns and the binary columns to be excluded, we can now create a third list which are only binary column that is eligible for transformation.

```
# Compare Sets of Variable and Remove Not Elligible Columns

EligBinCols <- setdiff(BinCols,ExcludeBinCols)
#Compare Lengths of List for Validation
length(BinCols)               # Total Binary Variables
```

```
## [1] 191
```

```
length(ExcludeBinCols)        # Binary Variables to be excluded
```

```
## [1] 18
```

```
length(EligBinCols)           # Remaining Binary Variables For Transformation
```

```
## [1] 173
```

After we have obtained the final list of Columns to be transformed, we can then quickly mutate and recode all with the code below.

```
data <- data %>%
mutate_at(vars(colnames(data[EligBinCols])),
          ~as.numeric(recode(.,
                             "1. Yes"=1,
                             "5. No"=0,
                             "1. Selected"=1,
                             "0. Not selected"=0)))
```

3. General Tidying: Renaming Useful Columns

Prior to in depth analysis, we have also identified appropriate columns to be renamed, in order to smoothen the process ahead.

```
#With the function created earlier, we are able to quickly lookup column defi
nition on the fly.
#Below are the dictionary definition for the columns we used in our analysis

for (i in c('country','cadn042_','cadn003_','caho037_','cah102_','caph003_','
cac105_1','cac120_1','cac120_2','cac120_3','cac120_4','cac120_5','cac120_6','
cac120_7','cac120_8','cac122_','cac111_1'))
{
     print(paste(Vardetails(i,2)," - ", i," - ", Vardetails(i)))
    }
```

```
## [1] "Intro  -  country  -  Country identifier"
## [1] "Intro  -  cadn042_  -  Intro: sex"
## [1] "Intro  -  cadn003_  -  Intro: year of birth"
## [1] "Intro  -  caho037_  -  Intro: area lived in"
## [1] "Health  -  cah102_  -  Health: change in health in the last 3 months"
## [1] "Health  -  caph003_  -  Health: rating of subjective health"
## [1] "COVID-19  -  cac105_1  -  COVID-19: respondent tested positive"
## [1] "COVID-19  -  cac120_1  -  COVID-19: fatigue attributed to respondent'
s Covid illness"
## [1] "COVID-19  -  cac120_2  -  COVID-19: cough, congestion, shortness of b
reath attributed to respondent's Covi"
## [1] "COVID-19  -  cac120_3  -  COVID-19: loss of taste or smell attributed
 to respondent's Covid illness"
## [1] "COVID-19  -  cac120_4  -  COVID-19: headache attributed to respondent
's Covid illness"
## [1] "COVID-19  -  cac120_5  -  COVID-19: body aches, joint pain attributed
 to respondent's Covid illness"
## [1] "COVID-19  -  cac120_6  -  COVID-19: chest or abdominal pain attribute
d to respondent's Covid illness"
## [1] "COVID-19  -  cac120_7  -  COVID-19: diarrhoea, nausea attributed to r
espondent's Covid illness"
## [1] "COVID-19  -  cac120_8  -  COVID-19: confusion attributed to responden
t's Covid illness"
## [1] "COVID-19  -  cac122_  -  COVID-19: drugs taken by respondent to allev
iate these [long-term or lingering]"
## [1] "COVID-19  -  cac111_1  -  COVID-19: respondent hospitalized"
```

Based on the definitions above, we can also first rename some keep variables to be used in our analysis.

```
data <- data %>%
rename(.,
       country=country,
       gender=cadn042_,
       yearborn=cadn003_,
       living_area=caho037_,
       health_change=cah102_,
       health_rate=caph003_,
       test_positive=cac105_1,
       sequelae1=cac120_1,
       sequelae2=cac120_2,
       sequelae3=cac120_3,
       sequelae4=cac120_4,
       sequelae5=cac120_5,
       sequelae6=cac120_6,
       sequelae7=cac120_7,
       sequelae8=cac120_8,
       drugs_taken = cac122_,
       hospitalized = cac111_1,)
```

Additionally, we are also able to estimate the respondents age, by substracting "2020", (which was the main period for the share8 survey) ~release survey page 10 against the respondents "cadn003_" year of birth data. The result can then be grouped into different age bins.

```
#Estimate Respondents Age
data$age <- 2020-data$yearborn
#Grouped into Age Bins
data <- data %>%
  mutate(agegroup = cut(age, seq(0, max(age) + 6, 10), right = FALSE))
head(data[,c('yearborn','age','agegroup')])
```

```
## # A tibble: 6 x 3
##   yearborn   age agegroup
##      <dbl> <dbl> <fct>
## 1     1952    68 [60,70)
## 2     1951    69 [60,70)
## 3     1924    96 [90,100)
## 4     1942    78 [70,80)
## 5     1942    78 [70,80)
## 6     1951    69 [60,70)
```

```
#Quick Recoding of Gender Variable
data <- data %>%
mutate(gender=recode(gender,
                             "1. Male"='Male',
                             "2. Female"='Female'))
```

```
#BACKUP
# write.csv(data, file = "Share8MissingValuesBinaryRenamedCleaned.csv", row.n
ames =FALSE)
```

Exploratory Data Analysis

As a whole, the share8 dataset provided contains 318 columns (+2 derived age variable) with 49253 observations.

```
# data
attach(data)

dim(data)

## [1] 49253   310
```

In this section, we will show all the variables used in this report. Including: country, gender, age, agegroup, living_area, health_change, health_rate, test_positive, eight major sequelaes. And these variables will be used in the EDA and Modelling sections respectively.

```
#Create the data frame for this report
df<-data.frame(country,gender,age,agegroup,living_area,health_change,health_r
ate,test_positive,sequelae1,sequelae2,sequelae3,sequelae4,sequelae5,sequelae
6,sequelae7,sequelae8,drugs_taken,hospitalized)

#Check size of data frame
dim(df)

## [1] 49253    18

#Check variable names
head(df)

##   country gender age agegroup living_area     health_change  health_rate
## 1 Austria Female  68  [60,70)        <NA> 2. About the same 2. Very good
## 2 Austria   Male  69  [60,70)        <NA> 2. About the same 2. Very good
## 3 Austria   Male  96 [90,100)        <NA> 2. About the same      3. Good
## 4 Austria Female  78  [70,80)        <NA>       3. Worsened      4. Fair
## 5 Austria   Male  78  [70,80)        <NA>       1. Improved      4. Fair
## 6 Austria Female  69  [60,70)        <NA> 2. About the same      3. Good
## test_positive sequelae1 sequelae2 sequelae3 sequelae4 sequelae5 sequelae6
## 1      NA        NA        NA        NA        NA        NA        NA
## 2       0        NA        NA        NA        NA        NA        NA
## 3      NA        NA        NA        NA        NA        NA        NA
## 4      NA        NA        NA        NA        NA        NA        NA
## 5      NA        NA        NA        NA        NA        NA        NA
## 6      NA        NA        NA        NA        NA        NA        NA
##   sequelae7 sequelae8 drugs_taken hospitalized
## 1        NA        NA          NA           NA
## 2        NA        NA          NA            0
```

```
## 3         NA         NA          NA           NA
## 4         NA         NA          NA           NA
## 5         NA         NA          NA           NA
## 6         NA         NA          NA            0
```

Some general demographic variables are available and derived from the dataset.

Firstly, we can understand the background of our share8 respondents as a whole.

```
library(maps)

##
## Attaching package: 'maps'

## The following object is masked from 'package:purrr':
##
##     map

#Count Respondents Occurence by Country and rename country to region to join
against map data
countrydf <- data %>% group_by(country) %>% tally() %>% rename(., region = co
untry)
# countrydf

# 1) Load Map Dataset from Library 2) Left Join Against Share8 Data 3) Filter
 out unrelated countries
mapdata1 <- map_data("world") %>%
    left_join(., countrydf, by="region") %>%
    filter(!is.na(.$n))

map2 <- ggplot(mapdata1, aes( x = long, y = lat, group=group)) +
  geom_polygon(color = "white",aes(fill = n, text = region)) +
scale_fill_gradient(name = "Participants",
  high = munsell::mnsl("5P 2/12"),
  low = munsell::mnsl("5P 7/12")
    , na.value = "grey50"
)+
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks = element_blank(),
        axis.title.y=element_blank(),
        axis.title.x=element_blank(),
       rect = element_rect(fill = "transparent")
       )+

theme(
  panel.background = element_rect(fill = "transparent",
                                  colour = NA_character_),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  plot.background = element_rect(fill = "transparent",
```

```
                                    colour = NA_character_),
  legend.background = element_rect(fill = "transparent"),
  legend.box.background = element_rect(fill = "transparent"),
  legend.key = element_rect(fill = "transparent"),
  legend.title=element_text(color="white")
)

## Warning: Ignoring unknown aesthetics: text

Map3 <- ggplotly(map2,tooltip = c("fill","text"))
Map3
```

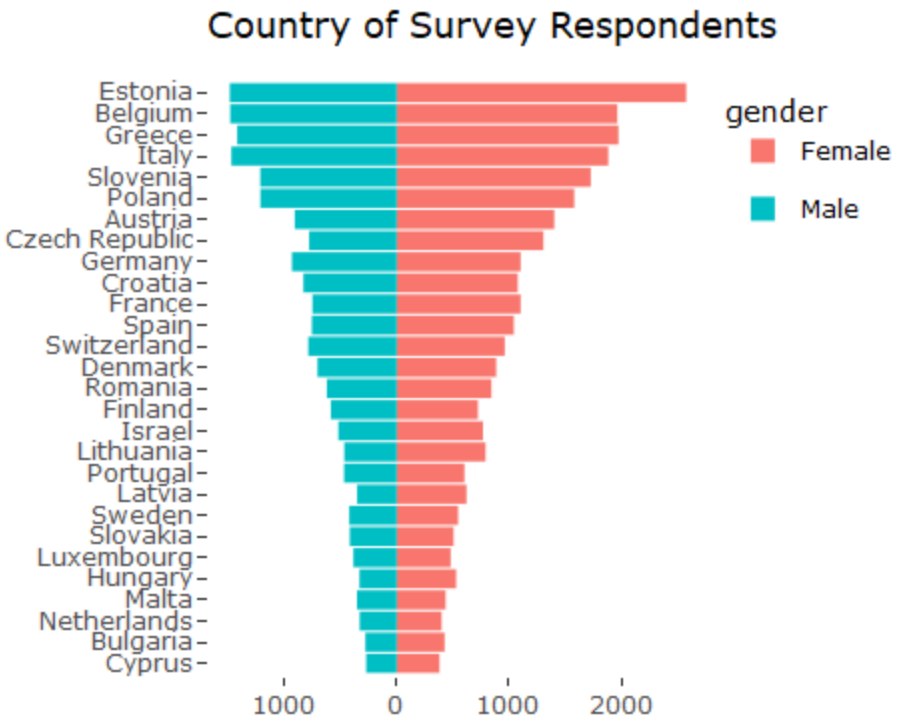

```
# options(browser = 'false')
# api_create(Map3, filename = "map1")
#https://chart-studio.plotly.com/~plotlys2022355/14
```

Share8 respondents resides in 28 European countries.

Alternatively, we can also understand the respondent demographic in terms of country and gender. The population pyramid below observed that there are in general more female than male respondents.

```
countrygp <- data %>% group_by(country, gender)  %>% tally()
countrygp$norg <-countrygp$n

#Male Values are ploted as negative, with labels showing absolute values to p
roduce population pyramid chart)
countrygp$n[countrygp$gender == "Male"]<-countrygp$n[countrygp$gender == "Mal
```

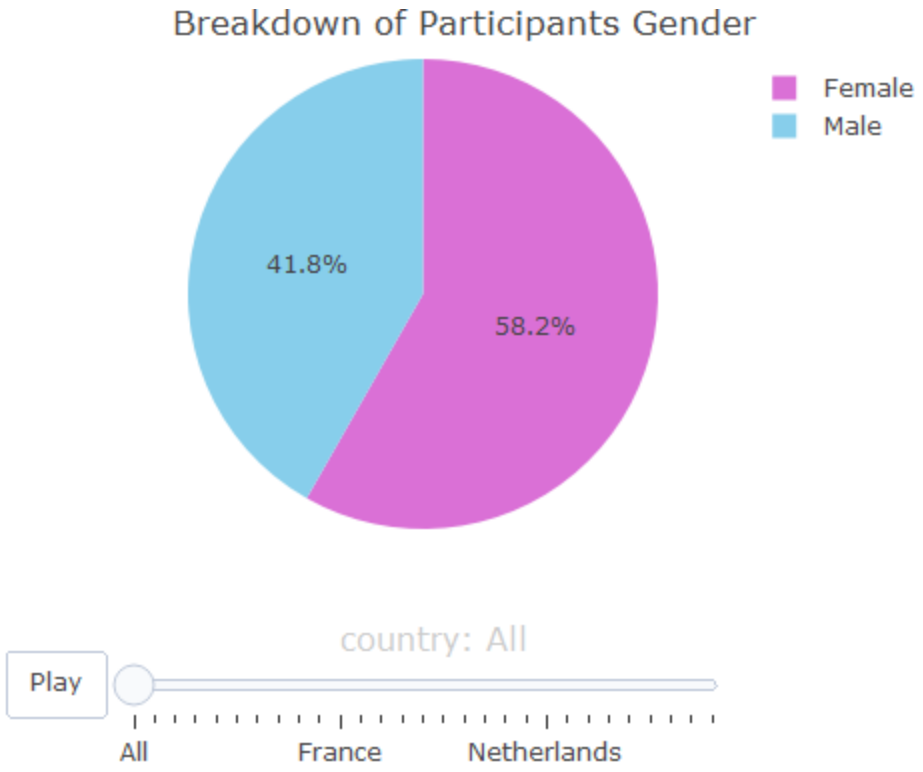
```
e"]*-1
B =    ggplot(countrygp,aes(x = reorder(country, norg),text=(country) ,y = n,
 fill=gender)) +
    geom_bar(stat = "identity") +
    coord_flip()+
   labs(title = "Country of Survey Respondents", x = "",
        y = "")+
  scale_y_continuous(labels = abs)+

theme(
  panel.background = element_rect(fill = "transparent", colour = NA_character
_),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  plot.background = element_rect(fill = "transparent", colour = NA_character
_),
  legend.background = element_rect(fill = "transparent"),
  legend.box.background = element_rect(fill = "transparent"),
  legend.key = element_rect(fill = "transparent")
  # ,legend.title=element_text(color="white")
  # ,legend.text=element_text(color="white")
  # ,plot.title=element_text(color="white")
  # ,axis.text.y=element_text(color="white")
  # ,axis.text.x=element_text(color="white")
)

Bplotly <- ggplotly(B,tooltip = c("y","text"))
Bplotly
```




```
# options(browser = 'false')
# api_create(Bplotly, filename = "CountryDistribution")
#https://chart-studio.plotly.com/~plotlys2022355/20
```

From the pie chart below, female generally consist of 60% of respondents for all countries

```
genderdf = data[,c('country','gender','agegroup')]

#Created and Union additional Tibble to create additional country category 'A
ll'
allgenderdf = genderdf
allgenderdf$country = 'All'
genderdf = union_all(genderdf,allgenderdf)

genderdf <- genderdf %>% group_by(country, gender)  %>% tally()
# head(genderdf)

plotpie <- plot_ly(data=genderdf,labels=~factor(gender), values=~n, frame = ~
country, marker = list(colors = c('orchid', 'skyblue')), type="pie"
        ) %>%
    layout(title = 'Breakdown of Participants Gender',
 xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
 yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels = FALSE),
 plot_bgcolor = "rgba(0, 0, 0, 0)",
 paper_bgcolor = "rgba(0, 0, 0, 0)")#%>% layout(font = list(color = 'white'))

plotpie
```

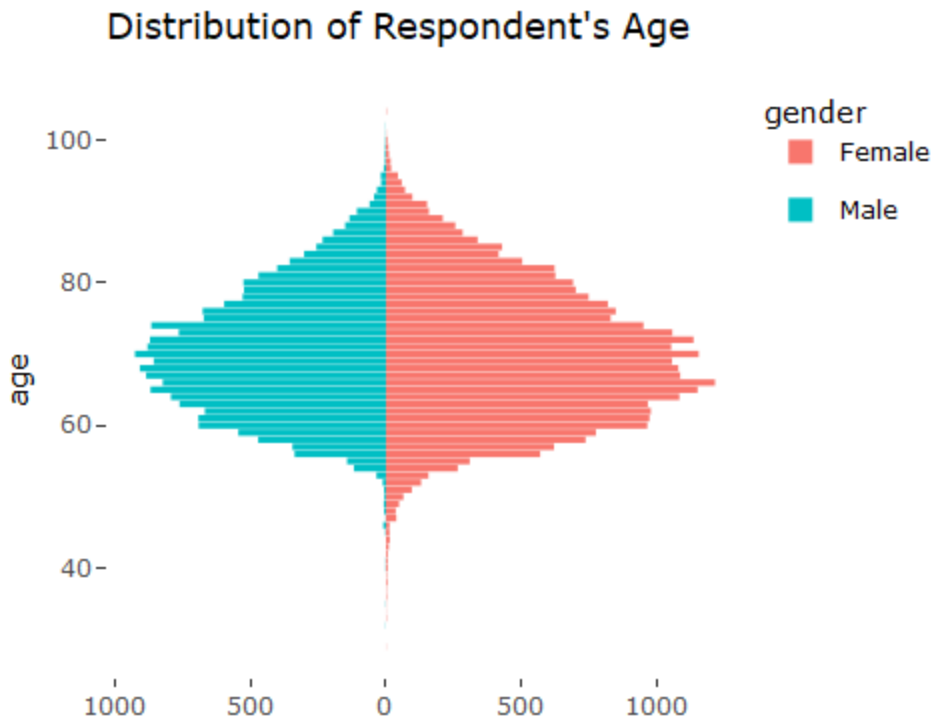



```
# options(browser = 'false')
# api_create(plotpie, filename = "GenderPieChart")
#https://chart-studio.plotly.com/~plotlys2022355/38
```

As we have already derived earlier, we can also explore our data in terms of age.

```
#Population Pyramid of Respondents Age

#Age Related Tibbles for Chart Constuction
Agedf <- data[,c('country','gender','age')]
agegp <- Agedf %>% group_by(age, gender)  %>% tally()
agegp$norg <-agegp$n

#Male Plotted on negative x-axis to achieve the population pyramid view
agegp$n[agegp$gender == "Male"]<-agegp$n[agegp$gender == "Male"]*-1


C =  ggplot(agegp,aes(x = age ,y = n, fill=gender)) +
    geom_bar(stat = "identity") +
    coord_flip()+
   labs(title = "Distribution of Respondent's Age", x = "age",
        y = "")+
  scale_y_continuous(labels = abs)+
# Transparent Aesthetics Themes
theme(
```

```
  panel.background = element_rect(fill="transparent", colour=NA_character_),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  plot.background = element_rect(fill="transparent", colour = NA_character_),
  legend.background = element_rect(fill = "transparent"),
  legend.box.background = element_rect(fill = "transparent"),
  legend.key = element_rect(fill = "transparent")
)

#Wrapping GGPlot Charts to Provide interactivity
Cplotly<-ggplotly(C,tooltip = c("y","x"))
Cplotly
```

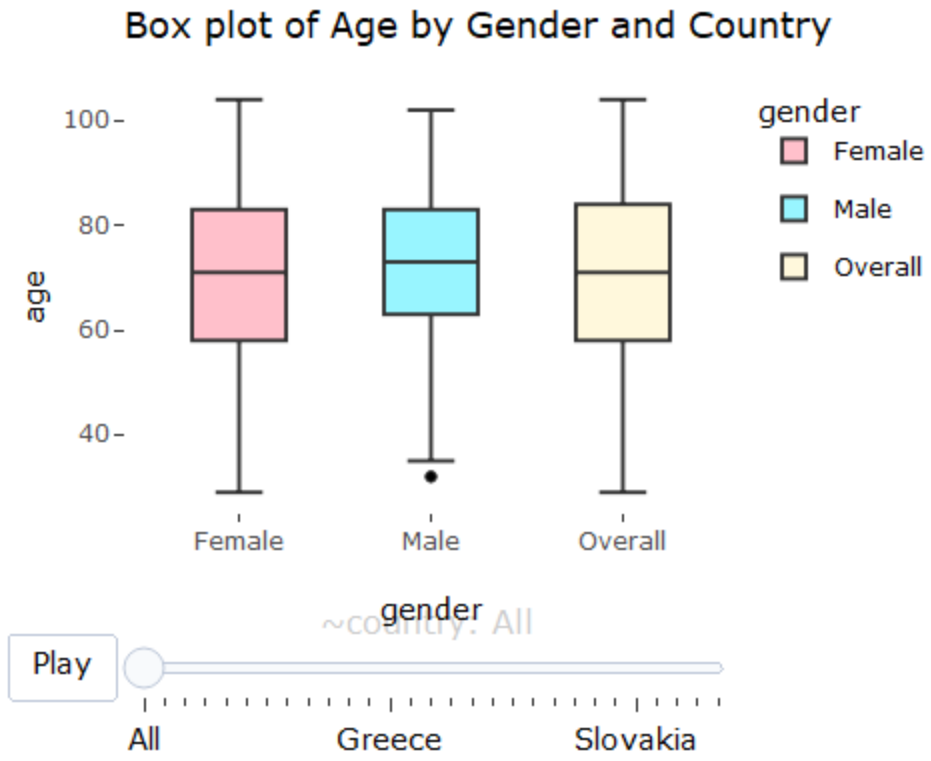



```
# options(browser = 'false')
# api_create(Cplotly, filename = "AgeDistribution") #PlotlyStudioUpload
#https://chart-studio.plotly.com/~plotlys2022355/18
```

We can check age distribution with a box plot.

```
agegender = data[,c('country','gender','age')]

#Created and Union additional Tibble to create additional gender category 'Al
l'
ageall = data[,c('country','age')]
ageall$gender = 'Overall'

agegender = union(agegender,ageall)
```

```
#Created and Union additional Tibble to create additional country category 'A
ll'
allcountryagegender = agegender
allcountryagegender$country = 'All'
agegender = union_all(agegender,allcountryagegender)

# Plot
g <- ggplot(agegender, aes(gender, age,fill=gender, frame=country))+
    geom_boxplot() +
    labs(title="Box plot of Age by Gender and Country",
         subtitle="boxplot by Gender",
         caption="Source: boxplot",
         x="gender",
         y="age")+
scale_fill_manual(values=c("pink",
                           "cadetblue1",
                           "cornsilk"))+
# Transparent Aesthetics Themes
theme(
  panel.background = element_rect(fill = "transparent",
                                  colour = NA_character_),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  plot.background = element_rect(fill = "transparent",
                                 colour = NA_character_),
  legend.background = element_rect(fill = "transparent"),
  legend.box.background = element_rect(fill = "transparent"),
  legend.key = element_rect(fill = "transparent")
)

Dplotly <- ggplotly(g,tooltip = c("y","text"))
Dplotly
```




```
# options(browser = 'false')
# api_create(Dplotly, filename = "AgeGenderCountryBoxPlot")

#Output Link: https://chart-studio.plotly.com/~plotlys2022355/36
```

Exploration of Main Variable of our study

```
#Convert the character and numeric variables in the data frame to factor for
further analysis
df[sapply(df, is.character)] <- lapply(df[sapply(df, is.character)], as.facto
r)
df[sapply(df, is.numeric)] <- lapply(df[sapply(df, is.numeric)], as.factor)

#Browse the basic information of the data frame
summary(df)
```

```
##      country         gender           age           agegroup
##  Estonia : 4069   Female:28684   70     : 2073   [60,70) :18434
##  Belgium : 3451   Male  :20569   66     : 2031   [70,80) :16541
##  Greece  : 3399                  65     : 2012   [80,90) : 7371
##  Italy   : 3360                  72     : 1999   [50,60) : 5721
##  Slovenia: 2946                  68     : 1980   [90,100):  934
##  Poland  : 2798                  67     : 1964   [40,50) :  220
##  (Other) :29230                  (Other):37194   (Other) :   32
```

```
##
##                    living_area                              health_change
##  1. A big city                              :  619   1. Improved      : 2917
##  2. The suburbs or outskirts of a big city:  375   2. About the same:39182
##  3. A large town                            :  564   3. Worsened      : 7104
##  4. A small town                            :  791   NA's             :   50
##  5. A rural area or village                 :  958
##  NA's                                       :45946
##
##     health_rate       test_positive sequelae1    sequelae2    sequelae3
##  1. Excellent: 2143   0   :15099    0   : 2016   0   : 2593   0   : 2805
##  2. Very good: 7648   1   : 3167    1   : 2056   1   : 1479   1   : 1267
##  3. Good     :19706   NA's:30987    NA's:45181   NA's:45181   NA's:45181
##  4. Fair     :14879
##  5. Poor     : 4842
##  NA's        :   35
##
##  sequelae4    sequelae5    sequelae6    sequelae7    sequelae8    drugs_ta
ken
##  0: 2910   0   : 2728   0   : 3478   0   : 3668   0   : 3733   0   : 1500
##  1: 1162   1   : 1344   1   :  594   1   :  404   1   :  339   1   : 1444
##  NA's:45181 NA's:45181   NA's:45181   NA's:45181   NA's:45181  NA's:46309
##
##  hospitalized
##  0   : 5435
##  1   :  594
##  NA's:43224
```

1.1 Changes in the health and health status of respondents living in different area

```
#Import the source and packages required by analysis
source("http://pcwww.liv.ac.uk/~william/R/crosstab.r")
library(ggplot2)
library(dplyr)
options(dplyr.summarise.inform = FALSE)

#Exhibit the change of health in different living area by crosstab
living_change<-data.frame(df$living_area,df$health_change)
#remove the missing value
living1<-drop_na(rename(living_change, living_area1 =df.living_area,health_ch
ange1 = df.health_change))
#build crosstab with count and row precentage
crosstab(living1, row.vars = "living_area1", col.vars = "health_change1",type
 = c("f", "r"), style = "long",addmargins = FALSE)
```

##

		health_change1 1. Improved	2. About the same	3. Worsened
living_area1				
1. A big city	Count	33.00	482.00	104.00
	Row %	5.33	77.87	16.80
2. The suburbs or outskirts of a big city	Count	27.00	295.00	52.00
	Row %	7.22	78.88	13.90
3. A large town	Count	41.00	447.00	76.00
	Row %	7.27	79.26	13.48
4. A small town	Count	74.00	600.00	116.00
	Row %	9.37	75.95	14.68
5. A rural area or village	Count	77.00	749.00	132.00
	Row %	8.04	78.18	13.78

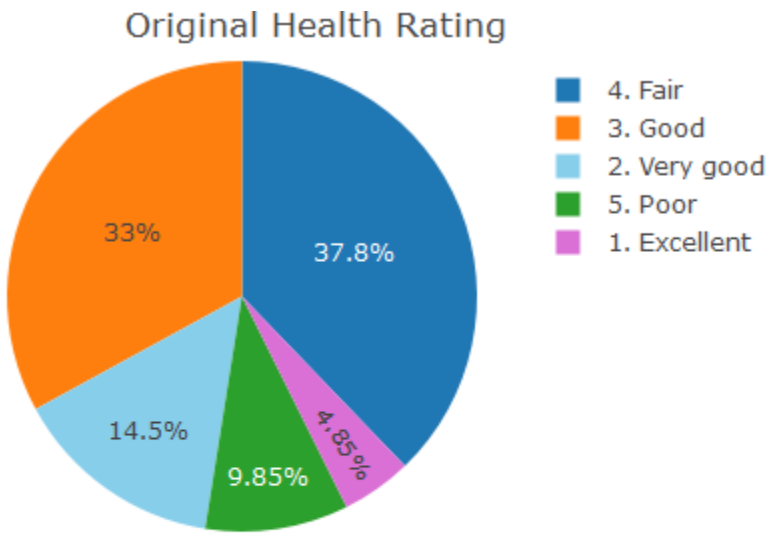
```
#Exhibit the health status in different living area by crosstab
living_rate<-data.frame(df$living_area,df$health_rate)
#remove the missing value
living2<-drop_na(rename(living_rate, living_area1 =df.living_area,health_rate
1 = df.health_rate))
#build crosstab with count and row precentage
crosstab(living2, row.vars = "living_area1", col.vars = "health_rate1",type =
 c("f", "r"), style = "long",addmargins = FALSE)
```

##

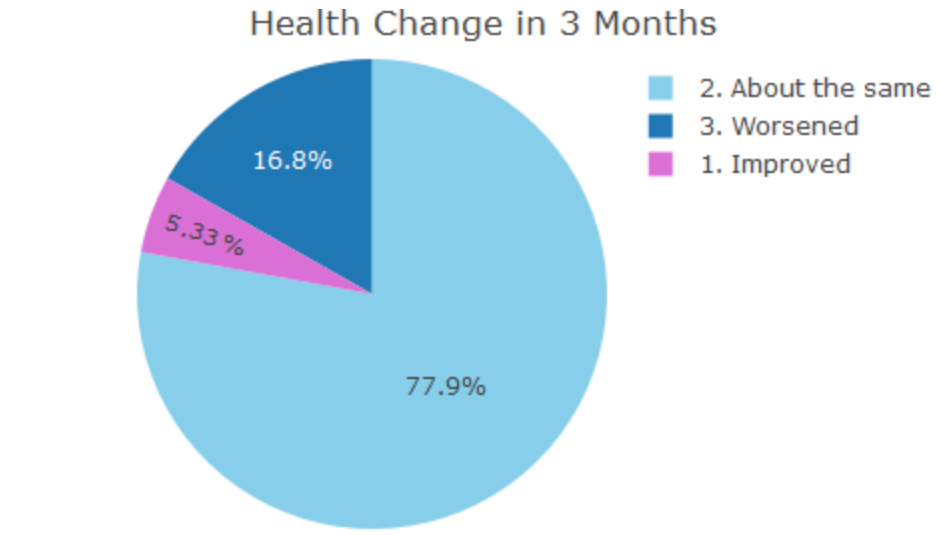
		health_rate1 1. Excellent	2. Very good	3. Good	4. Fair	5. Poor
living_area1						
1. A big city	Count	30.00	90.00	204.00	234.00	61.00
	Row %	4.85	14.54	32.96	37.80	9.85
2. The suburbs or outskirts of a big city	Count	30.00	77.00	147.00	94.00	27.00
	Row %	8.00	20.53	39.20	25.07	7.20
3. A large town	Count	45.00	85.00	201.00	174.00	59.00
	Row %	7.98	15.07	35.64	30.85	10.46
4. A small town	Count	48.00	122.00	293.00	250.00	78.00
	Row %	6.07	15.42	37.04	31.61	9.86
5. A rural area or village	Count	60.00	158.00	341.00	295.00	103.00
	Row %	6.27	16.51	35.63	30.83	10.76

```
living_areadf = data[,c('country','living_area','health_change','health_rate
')]%>% na.omit(.)
living_areadfhealthrate <- living_areadf %>% group_by(health_rate, living_are
a)  %>% tally()
living_areadfhealthchange <- living_areadf %>% group_by(health_change, living
_area)  %>% tally()

living_areadfhealthratefig1 <- plot_ly(data=living_areadfhealthrate,labels=~f
actor(health_rate), values=~n, frame = ~living_area, marker = list(colors = c
('orchid', 'skyblue')), type="pie"
        ) %>%
    layout(title = 'Original Health Rating',
           xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels =
FALSE),
```

```
            yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels =
FALSE),
            plot_bgcolor = "rgba(0, 0, 0, 0)",
            paper_bgcolor = "rgba(0, 0, 0, 0)")#%>% layout(font = list(color =
 'white'))
living_areadfhealthchangefig2 <- plot_ly(data=living_areadfhealthchange,label
s=~factor(health_change), values=~n, frame = ~living_area, marker = list(colo
rs = c('orchid', 'skyblue')), type="pie"
        ) %>%
    layout(title = 'Health Change in 3 Months',
            xaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels =
FALSE),
            yaxis = list(showgrid = FALSE, zeroline = FALSE, showticklabels =
FALSE),
            plot_bgcolor = "rgba(0, 0, 0, 0)",
            paper_bgcolor = "rgba(0, 0, 0, 0)")#%>% layout(font = list(color =
 'white'))

living_areadfhealthratefig1
```




```
living_areadfhealthchangefig2
```




```
# options(browser = 'false')
# api_create(living_areadfhealthratefig1, filename = "living_areadfhealthrate
fig1")
# https://chart-studio.plotly.com/~plotlys2022355/52/#/
# options(browser = 'false')
# api_create(living_areadfhealthchangefig2, filename = "living_areadfhealthch
angefig2")
# https://chart-studio.plotly.com/~plotlys2022355/54/#/
```

According to crosstab and Figures, regardless of which area the respondents lived in, those who considered themselves health status is good, or fair accounted for two-thirds of the overall living area. Moreover, 15% to 20% of people think they are in very good health, and only a few people think their health status are excellent or poor. Similarly, the number of respondents living in big cities who thought their health was excellent/very good/good was significantly lower than in other regions. On the other hand, respondents living in the suburbs or outskirts of a big city were more likely to feel healthy. And respondents in the rest of the regions were similar in health.

1.2 Changes in health and health status of respondents by Gender

```
#Exhibit the change of health by gender via crosstab
gender_change<-data.frame(df$gender,df$health_change)
#remove the missing value
gender1<-drop_na(rename(gender_change, gender1 =df.gender,health_change1 = d
f.health_change))
```

```
#build crosstab with count and row precentage
crosstab(gender1, row.vars = "gender1", col.vars = "health_change1",type = c
("f", "r"), style = "long",addmargins = FALSE)
```

```
##               health_change1 1. Improved 2. About the same 3. Worsened
## gender1
## Female  Count                    1796.00          22419.00     4431.00
##         Row %                       6.27             78.26       15.47
## Male    Count                    1121.00          16763.00     2673.00
##         Row %                       5.45             81.54       13.00
```

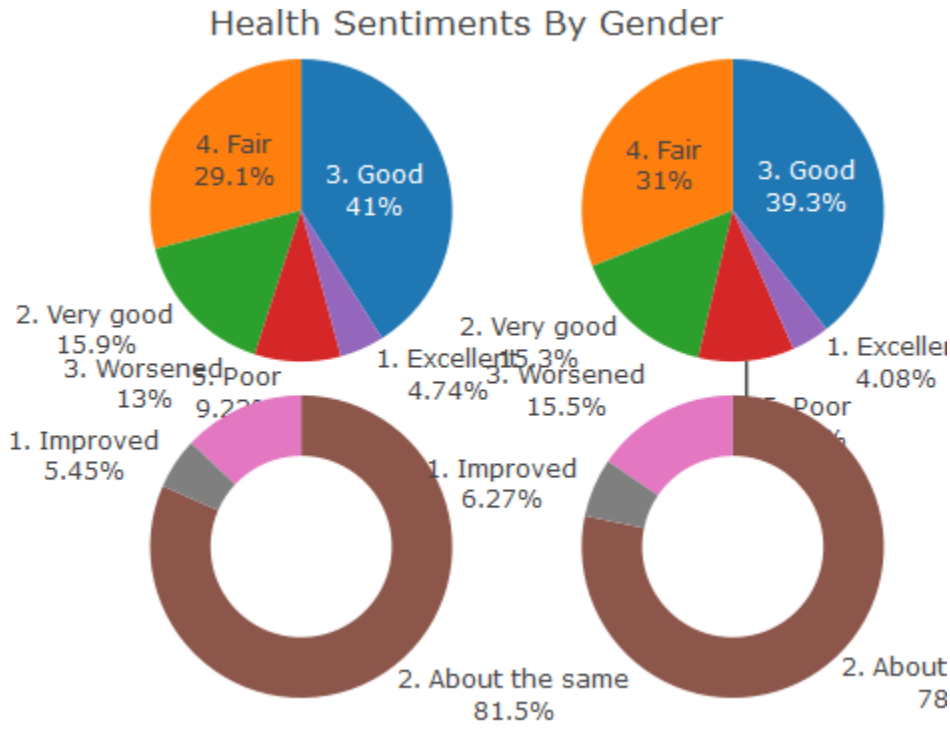
```
#Exhibit the change of health by gender via crosstab
gender_rate<-data.frame(df$gender,df$health_rate)
#remove the missing value
gender2<-drop_na(rename(gender_rate, gender1 =df.gender,health_rate1 = df.hea
lth_rate))
#build crosstab with count and row precentage
crosstab(gender2, row.vars = "gender1", col.vars = "health_rate1",type = c("f
", "r"), style = "long",addmargins = FALSE)
```

```
##
```

		health_rate1 1. Excellent	2. Very good	3. Good	4. Fair	5. Poor
gender1						
Female	Count	1167.00	4383.00	11274.00	8890.00	2946.00
	Row %	4.07	15.29	39.34	31.02	10.28
Male	Count	976.00	3265.00	8432.00	5989.00	1896.00
	Row %	4.75	15.88	41.02	29.13	9.22

```
genderdf = data[,c('country','gender','health_change','health_rate')]%>% na.o
mit(.)
genderdfhealthrate <- genderdf %>% group_by(health_rate, gender)  %>% tally()

genderdfhealthchange <- genderdf %>% group_by(health_change, gender)  %>% tal
ly()

fig <- plot_ly()
fig <- fig %>% add_pie(data = genderdfhealthrate %>% filter(gender == 'Male
'), labels = ~factor(health_rate), values = ~n,
                       name = "Male: Start", textinfo='label+percent',   do
main = list(row = 0, column = 0))
fig <- fig %>% add_pie(data = genderdfhealthrate %>% filter(gender == 'Female
'), labels = ~factor(health_rate), values = ~n,
                       name = "Female: Start",  textinfo='label+percent',
 domain = list(row = 0, column = 1))
fig <- fig %>% add_pie(data = genderdfhealthchange %>% filter(gender == 'Male
'), labels = ~factor(health_change), values = ~n,
                     name = "Male: 3 Months", textinfo='label+percent', dom
ain = list(row = 1, column = 0),hole = 0.6)
```

```
fig <- fig %>% add_pie(data = genderdfhealthchange %>% filter(gender == 'Fema
le'), labels = ~factor(health_change), values = ~n,
                       name = "Female: 3 Months", textinfo='label+percent', d
omain = list(row = 1, column = 1),hole = 0.6)
fig <- fig %>% layout(title = "Health Sentiments By Gender", showlegend = F,
                      grid=list(rows=2, columns=2),
                      xaxis = list(showgrid = FALSE, zeroline = FALSE, showti
cklabels = FALSE),
                      yaxis = list(showgrid = FALSE, zeroline = FALSE, showti
cklabels = FALSE))

fig%>%
    layout(
           plot_bgcolor = "rgba(0, 0, 0, 0)",
           paper_bgcolor = "rgba(0, 0, 0, 0)")
```




```
# options(browser = 'false')
# api_create(fig, filename = "genderpiesubplots")
# https://chart-studio.plotly.com/~plotlys2022355/56
```

According to crosstab and Figures, regardless of male or female, the vast majority of respondents felt that their physical health had not changed, followed by worse and better. And female is more likely to consider their health is better or worse than male, possibly because women are more concerned about their health.

According to crosstab the Figures, overall, there were no significant differences in health between male and female. However, women are more likely to feel that their health status is fair and poor relatively.

1.3 Changes in health and health status of respondents by country

```
#Exhibit the change of health in different country by crosstab
country_change<-data.frame(df$country,df$health_change)
#remove the missing value
country1<-drop_na(rename(country_change, country1 =df.country,health_change1
= df.health_change))
#build crosstab with count and row precentage
crosstab(country1, row.vars = "country1", col.vars = "health_change1",type =
c("f", "r"), style = "wide",addmargins = FALSE)
```

```
##
```

	Count			Row %		
health_change1	1. Improved	2. About the same	3. Worsened	1. Improved	2. About the same	3. Worsened
country1						
Austria	147.00	1839.00	326.00	6.36	79.54	14.10
Belgium	302.00	2690.00	454.00	8.76	78.06	13.17
Bulgaria	11.00	520.00	174.00	1.56	73.76	24.68
Croatia	84.00	1495.00	331.00	4.40	78.27	17.33
Cyprus	16.00	537.00	100.00	2.45	82.24	15.31
Czech Republic	146.00	1636.00	307.00	6.99	78.31	14.70
Denmark	134.00	1380.00	76.00	8.43	86.79	4.78
Estonia	287.00	3155.00	623.00	7.06	77.61	15.33
Finland	165.00	1033.00	113.00	12.59	78.79	8.62
France	140.00	1491.00	224.00	7.55	80.38	12.08
Germany	196.00	1568.00	272.00	9.63	77.01	13.36
Greece	64.00	2987.00	342.00	1.89	88.03	10.08
Hungary	27.00	737.00	98.00	3.13	85.50	11.37
Israel	41.00	913.00	328.00	3.20	71.22	25.59
Italy	81.00	2812.00	466.00	2.41	83.72	13.87
Latvia	27.00	753.00	195.00	2.77	77.23	20.00
Lithuania	80.00	920.00	258.00	6.36	73.13	20.51
Luxembourg	63.00	698.00	104.00	7.28	80.69	12.02
Malta	23.00	624.00	143.00	2.91	78.99	18.10
Netherlands	85.00	578.00	67.00	11.64	79.18	9.18
Poland	137.00	2101.00	557.00	4.90	75.17	19.93
Portugal	74.00	803.00	194.00	6.91	74.98	18.11
Romania	67.00	1157.00	243.00	4.57	78.87	16.56
Slovakia	40.00	748.00	137.00	4.32	80.86	14.81
Slovenia	194.00	2361.00	391.00	6.59	80.14	13.27
Spain	73.00	1417.00	305.00	4.07	78.94	16.99
Sweden	84.00	802.00	82.00	8.68	82.85	8.47
Switzerland	129.00	1427.00	194.00	7.37	81.54	11.09

```
#Exhibit the health status in different country by crosstab
country_rate<-data.frame(df$country,df$health_rate)
#remove the missing value
country2<-drop_na(rename(country_rate, country1 =df.country,health_rate1 = d
f.health_rate))
#build crosstab with count and row precentage
```

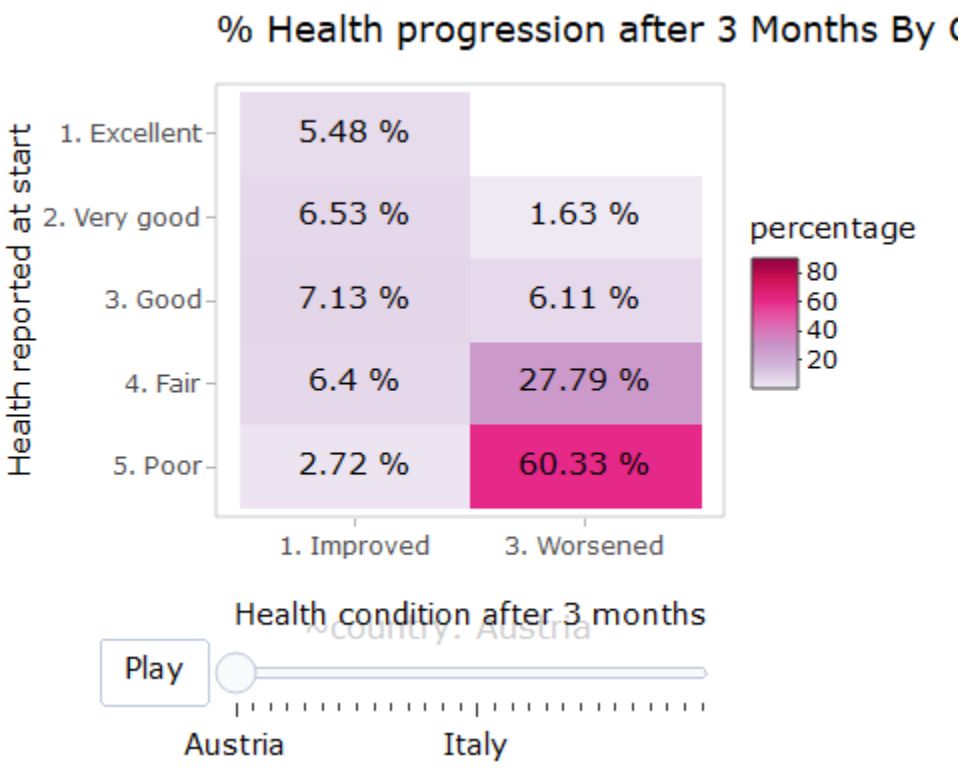
```
crosstab(country2, row.vars = "country1", col.vars = "health_rate1",type = c
("f", "r"), style = "wide",addmargins = FALSE)

##
```

	Count					Row %				
health_rate1	1. Excellent	2. Very good	3. Good	4. Fair	5. Poor	1. Excellent	2. Very good	3. Good	4. Fair	5. Poor
country1										
Austria	146.00	552.00	884.00	548.00	184.00	6.31	23.85	38.20	23.68	7.95
Belgium	218.00	811.00	1491.00	762.00	168.00	6.32	23.51	43.22	22.09	4.87
Bulgaria	14.00	65.00	299.00	250.00	78.00	1.98	9.21	42.35	35.41	11.05
Croatia	102.00	394.00	706.00	480.00	228.00	5.34	20.63	36.96	25.13	11.94
Cyprus	26.00	123.00	277.00	183.00	44.00	3.98	18.84	42.42	28.02	6.74
Czech Republic	61.00	248.00	1051.00	566.00	162.00	2.92	11.88	50.34	27.11	7.76
Denmark	237.00	621.00	383.00	274.00	74.00	14.92	39.08	24.10	17.24	4.66
Estonia	44.00	195.00	1102.00	2028.00	699.00	1.08	4.79	27.09	49.85	17.18
Finland	81.00	215.00	589.00	352.00	74.00	6.18	16.40	44.93	26.85	5.64
France	86.00	286.00	833.00	476.00	173.00	4.64	15.43	44.93	25.67	9.33
Germany	82.00	320.00	839.00	629.00	166.00	4.03	15.72	41.21	30.89	8.15
Greece	130.00	745.00	1353.00	871.00	300.00	3.82	21.92	39.81	25.63	8.83
Hungary	24.00	118.00	349.00	283.00	88.00	2.78	13.69	40.49	32.83	10.21
Israel	59.00	225.00	386.00	433.00	176.00	4.61	17.59	30.18	33.85	13.76
Italy	86.00	361.00	1412.00	1190.00	310.00	2.56	10.75	42.04	35.43	9.23
Latvia	1.00	13.00	292.00	507.00	162.00	0.10	1.33	29.95	52.00	16.62
Lithuania	25.00	45.00	467.00	595.00	125.00	1.99	3.58	37.15	47.33	9.94
Luxembourg	37.00	179.00	405.00	179.00	67.00	4.27	20.65	46.71	20.65	7.73
Malta	18.00	59.00	346.00	328.00	39.00	2.28	7.47	43.80	41.52	4.94
Netherlands	99.00	148.00	299.00	154.00	30.00	13.56	20.27	40.96	21.10	4.11
Poland	18.00	201.00	1324.00	927.00	327.00	0.64	7.19	47.34	33.14	11.69
Portugal	24.00	51.00	285.00	462.00	249.00	2.24	4.76	26.61	43.14	23.25
Romania	17.00	170.00	688.00	349.00	243.00	1.16	11.59	46.90	23.79	16.56
Slovakia	83.00	149.00	491.00	172.00	30.00	8.97	16.11	53.08	18.59	3.24
Slovenia	116.00	452.00	1339.00	740.00	299.00	3.94	15.34	45.45	25.12	10.15
Spain	30.00	175.00	706.00	648.00	237.00	1.67	9.74	39.31	36.08	13.20
Sweden	130.00	219.00	348.00	226.00	45.00	13.43	22.62	35.95	23.35	4.65
Switzerland	149.00	508.00	762.00	267.00	65.00	8.51	29.01	43.52	15.25	3.71

The health outcomes for each country.

```
plottb <- data[,c('country','gender','agegroup','health_rate','health_change
','test_positive')] %>%
                group_by(health_rate, country, health_change)%>% tally()%>%
                rename(., total='n')%>%na.omit(.)

#Percentage Computation
plottbgroup <- plottb %>% group_by(health_rate, country) %>%
  summarise(Frequency = sum(total))
plottb <- left_join(plottb, plottbgroup, by=c("health_rate","country"))
plottb$percentage <- plottb$total / plottb$Frequency *100
plottb$percentage <- round(plottb$percentage, digits = 2)
plottb <- plottb %>% filter(health_change != '2. About the same')

# plot
E<- plottb %>%
ggplot(mapping = aes(y = reorder(health_rate, desc(health_rate)), x = health_
change, text=paste(percentage,"%"), frame=country)) +
geom_tile(mapping = aes(fill = percentage, frame=country)) +
  scale_fill_distiller(palette = "PuRd", direction = 1) +
  theme_light() +
  labs(title="% Health progression after 3 Months By Country",
```

```
       y="Health reported at start",
       x="Health condition after 3 months",
       caption ='test',
       color=NULL)+

theme(
  panel.background = element_rect(fill = "transparent",
                                  colour = NA_character_), # necessary to avo
id drawing panel outline
  panel.grid.major = element_blank(), # get rid of major grid
  panel.grid.minor = element_blank(), # get rid of minor grid
  plot.background = element_rect(fill = "transparent",
                                 colour = NA_character_), # necessary to avoi
d drawing plot outline
  legend.background = element_rect(fill = "transparent"),
  legend.box.background = element_rect(fill = "transparent"),
  legend.key = element_rect(fill = "transparent")
) + geom_text(aes(label = paste(percentage,"%")))

## Warning: Ignoring unknown aesthetics: frame

Eplotly<-ggplotly(E,tooltip = c("x", "text"))
Eplotly
```




```
# options(browser = 'false')
# api_create(Eplotly, filename = "HealthChangeCountry")
#https://chart-studio.plotly.com/~plotlys2022355/44
```

1.4 Changes in health and health status of respondents by agegroup

```
#Exhibit the change of health in different agegroup by crosstab
agegroup_change<-data.frame(df$agegroup,df$health_change)
#remove the missing value
agegroup1<-drop_na(rename(agegroup_change, agegroup1 =df.agegroup,health_change1 = df.health_change))
#build crosstab with count and row precentage
crosstab(agegroup1, row.vars = "agegroup1", col.vars = "health_change1",type = c("f", "r"), style = "wide",addmargins = FALSE)
```

##

	Count			Row %		
health_change1	1. Improved	2. About the same	3. Worsened	1. Improved	2. About the same	3. Worsened
agegroup1						
[0,10)	0.00	0.00	0.00	0.00	0.00	0.00
[10,20)	0.00	0.00	0.00	0.00	0.00	0.00
[20,30)	0.00	1.00	0.00	0.00	100.00	0.00
[30,40)	0.00	19.00	0.00	0.00	100.00	0.00
[40,50)	14.00	192.00	14.00	6.36	87.27	6.36
[50,60)	372.00	4762.00	583.00	6.51	83.30	10.20
[60,70)	1169.00	15232.00	2022.00	6.35	82.68	10.98
[70,80)	982.00	13031.00	2510.00	5.94	78.87	15.19
[80,90)	352.00	5330.00	1675.00	4.78	72.45	22.77
[90,100)	28.00	608.00	295.00	3.01	65.31	31.69
[100,110)	0.00	7.00	5.00	0.00	58.33	41.67

```
#Exhibit the health status in different agegroup by crosstab
agegroup_rate<-data.frame(df$agegroup,df$health_rate)
#remove the missing value
agegroup2<-drop_na(rename(agegroup_rate, agegroup1 =df.agegroup,health_rate1 = df.health_rate))
#build crosstab with count and row precentage
crosstab(agegroup2, row.vars = "agegroup1", col.vars = "health_rate1",type = c("f", "r"), style = "wide",addmargins = FALSE)
```

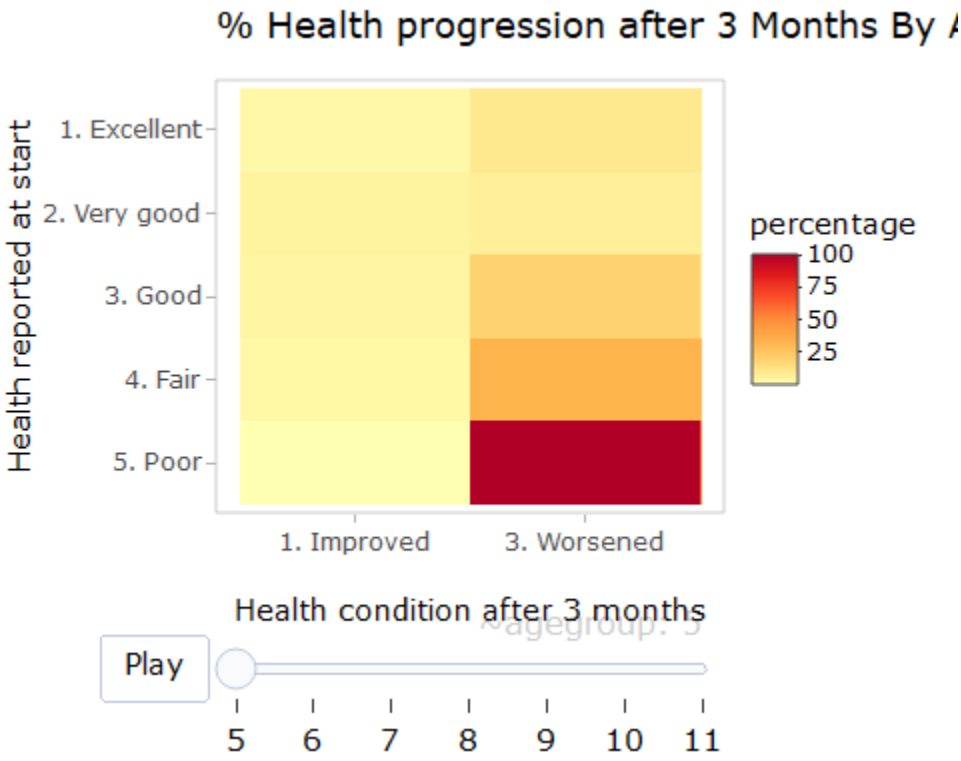
##

	Count					Row %				
health_rate1	1. Excellent	2. Very good	3. Good	4. Fair	5. Poor	1. Excellent	2. Very good	3. Good	4. Fair	5. Poor
agegroup1										
[0,10)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
[10,20)	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
[20,30)	1.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00
[30,40)	8.00	9.00	2.00	0.00	0.00	42.11	47.37	10.53	0.00	0.00
[40,50)	25.00	69.00	83.00	32.00	11.00	11.36	31.36	37.73	14.55	5.00
[50,60)	409.00	1350.00	2489.00	1173.00	296.00	7.15	23.61	43.54	20.52	5.18
[60,70)	1042.00	3532.00	8035.00	4701.00	1115.00	5.66	19.17	43.61	25.51	6.05
[70,80)	523.00	2134.00	6607.00	5507.00	1758.00	3.16	12.91	39.97	33.32	10.64
[80,90)	125.00	500.00	2264.00	3067.00	1407.00	1.70	6.79	30.75	41.65	19.11
[90,100)	10.00	53.00	221.00	396.00	252.00	1.07	5.69	23.71	42.49	27.04
[100,110)	0.00	1.00	5.00	3.00	3.00	0.00	8.33	41.67	25.00	25.00

We can explore health outcomes by age groups

```
plottb <- data[,c('country','gender','age','agegroup','health_rate','health_c
hange','test_positive')]%>%
                group_by(health_rate, agegroup, health_change)%>% tally()%>%
                rename(., total='n')%>%na.omit(.)%>%
                mutate(agegroup=recode(agegroup,
                            "[40,50]"='1. 40-50',
                            "[50,60]"='2. 50-60',
                            "[60,70]"='3. 60-70',
                            "[70,80]"='4. 70-80',
                            "[80,90]"='5. 80-90',
                            "[90,100]"='6. 90-100',
                            "[100,110]"='7. 100-110' ]]


#Percentage Computation
plottbgroup <- plottb %>% group_by(health_rate, agegroup) %>%
  summarise(Frequency = sum(total))
plottb <- left_join(plottb, plottbgroup, by=c("health_rate","agegroup"))
plottb$percentage <- plottb$total / plottb$Frequency *100
plottb$percentage <- round(plottb$percentage, digits = 2)
plottb <- plottb %>% filter(health_change != '2. About the same')

# plot
G<- plottb %>%
ggplot(mapping = aes(y = reorder(health_rate, desc(health_rate)), x = health_
change, text=paste(percentage,"%"), frame=agegroup)) +
geom_tile(mapping = aes(fill = percentage, frame=agegroup)) +
  scale_fill_distiller(palette = "YlOrRd", direction = 1) +
  theme_light() +
  labs(title="% Health progression after 3 Months By Age Group",
       y="Health reported at start",
       x="Health condition after 3 months",
       caption ='test',
       color=NULL)+

theme(
  panel.background = element_rect(fill = "transparent",
                                  colour = NA_character_), # necessary to avo
id drawing panel outline
  panel.grid.major = element_blank(), # get rid of major grid
  panel.grid.minor = element_blank(), # get rid of minor grid
  plot.background = element_rect(fill = "transparent",
                                 colour = NA_character_), # necessary to avoi
d drawing plot outline
  legend.background = element_rect(fill = "transparent"),
  legend.box.background = element_rect(fill = "transparent"),
  legend.key = element_rect(fill = "transparent")
)

## Warning: Ignoring unknown aesthetics: frame
```

```
# + geom_text(aes(label = paste(percentage,"%")))

Fplotly<-ggplotly(G,tooltip = c("x", "text"))
Fplotly
```




```
# options(browser = 'false')
# api_create(Eplotly, filename = "HealthChangeAgeGroup")
#https://chart-studio.plotly.com/~plotlys2022355/46
```

Next, this report translates the respondents' physical health status into a point scale. The better the health, the higher the score (5 points for "1. Excellent" and 1 point for "5. Poor"). And we will also analyze the average health scores of respondents across different living area, gender, country and age group.

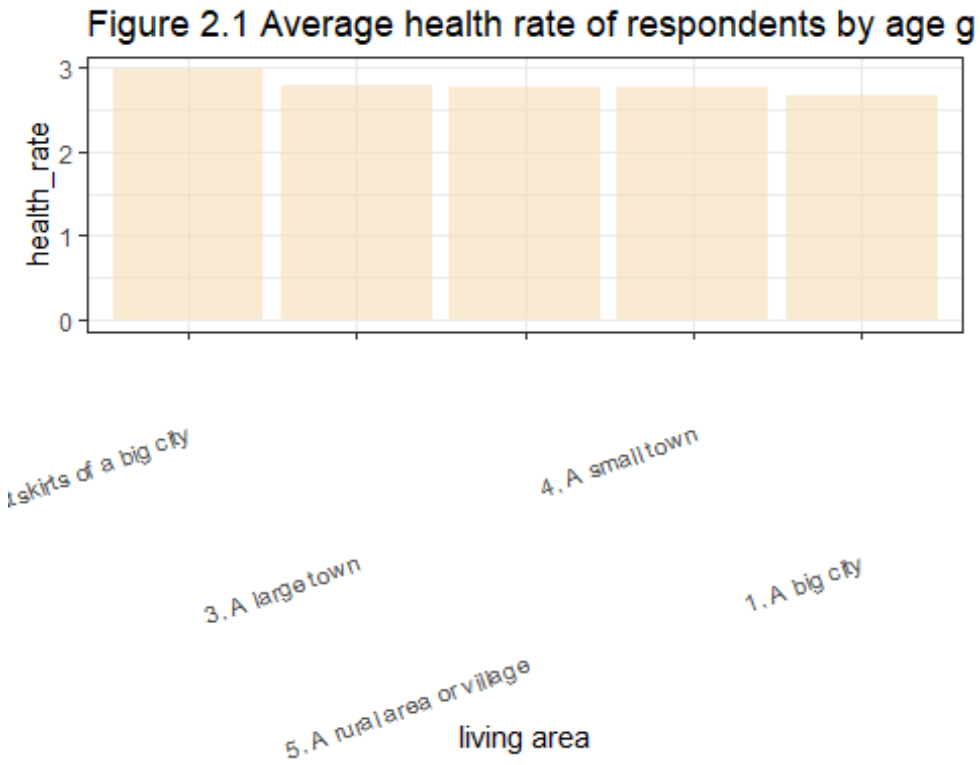
```
#Convert health_rate to numeric variable and recode the content
df$health_rate<-as.character(df$health_rate)
df$health_rate[df$health_rate == "1. Excellent"]<-5
df$health_rate[df$health_rate == "2. Very good"]<-4
df$health_rate[df$health_rate == "3. Good"]<-3
df$health_rate[df$health_rate == "4. Fair"]<-2
df$health_rate[df$health_rate == "5. Poor"]<-1
df$health_rate<-as.numeric(df$health_rate)
```

2.1 Average health rate of respondents living in different area

```r
#Exhibit the average health rate in different living area
living_rate1<-data.frame(df$living_area,df$health_rate)
#remove the missing value
living3<-drop_na(rename(living_rate1, living_area1 =df.living_area,health_rat
e1 = df.health_rate))
#calculate the mean value, ranked as descending
arrange(living3 %>%
  group_by(living_area1) %>%
  summarise_at(vars(health_rate1), list(health_rate = mean)),desc(health_rat
e))
```

```
## # A tibble: 5 x 2
##   living_area1                                health_rate
##   <fct>                                             <dbl>
## 1 2. The suburbs or outskirts of a big city          2.97
## 2 3. A large town                                    2.79
## 3 5. A rural area or village                         2.77
## 4 4. A small town                                    2.76
## 5 1. A big city                                      2.67
```

```r
#Exhibit the average health rate in different living area by ggplot
living3_a<-living3 %>%
  group_by(living_area1) %>%
  summarise_at(vars(health_rate1), list(health_rate = mean))
#Build the barchart, ranked as descending
livingchart <- living3_a %>%
  ggplot(aes(x=reorder(living_area1,-health_rate), y=health_rate))+
  geom_bar(stat="identity", fill="wheat", alpha=.6, width=.9)+
  scale_x_discrete(guide = guide_axis(n.dodge=3))+
  labs(title="Figure 2.1 Average health rate of respondents by age group")+
  xlab("living area") +
  theme_bw()+
    theme(axis.text.x = element_text(angle = 20, vjust = 0.5, hjust=1,(size=
5)))
suppressWarnings(livingchart)
```

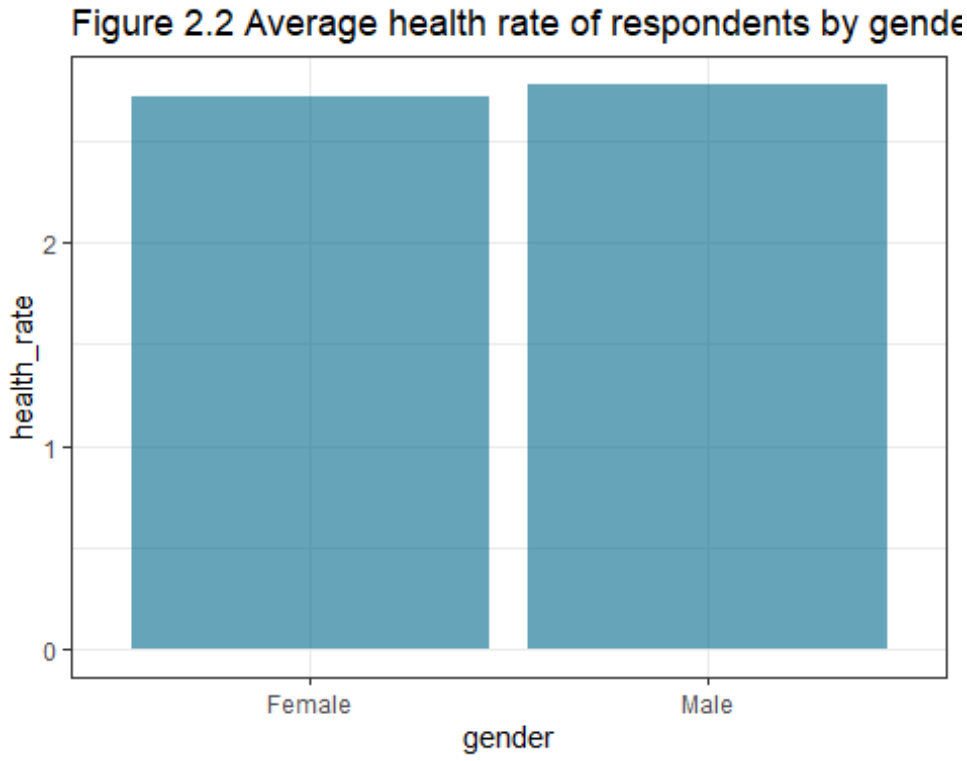


Respondents living in the suburbs or outskirts of a big city had the highest average health score (2.97), while those living in big cities had the lowest average health score (2.66). And the average score for the rest of the regions is around 2.75.

2.2 Average health rate of respondents by gender

```
#Exhibit the average health rate by gender
gender_rate1<-data.frame(df$gender,df$health_rate)
#remove the missing value
gender3<-drop_na(rename(gender_rate1, gender1 =df.gender,health_rate1 = df.he
alth_rate))
#calculate the mean value
gender3 %>%
  group_by(gender1) %>%
  summarise_at(vars(health_rate1), list(health_rate = mean))

## # A tibble: 2 x 2
##   gender1 health_rate
##   <fct>         <dbl>
## 1 Female         2.72
## 2 Male           2.78

#Exhibit the average health rate by gender via ggplot
gender3_a<-gender3 %>%
  group_by(gender1) %>%
```

```
  summarise_at(vars(health_rate1), list(health_rate = mean))
#build the bar chart
genderchart <- gender3_a %>%
  ggplot(aes(x=gender1, y=health_rate))+
  geom_bar(stat="identity", fill="deepskyblue4", alpha=.6, width=.9)+
  labs(title="Figure 2.2 Average health rate of respondents by gender")+
  xlab("gender") +
  theme_bw()
genderchart
```



Similar to the results of "1.2 Changes in health and health status of respondents by Gender", there was no significant difference in the average health scores of males and female. And the average score for men was only about 0.06 higher than that for women.

2.3 Average health rate of respondents in different countries

```
#Exhibit the average health rate in different country
country_rate1<-data.frame(df$country,df$health_rate)
#remove the missing value
country3<-drop_na(rename(country_rate1, country1 =df.country,health_rate1 = d
f.health_rate))
#calculate the mean value, show countries with the highest scores
head(arrange(country3 %>%
  group_by(country1) %>%
```

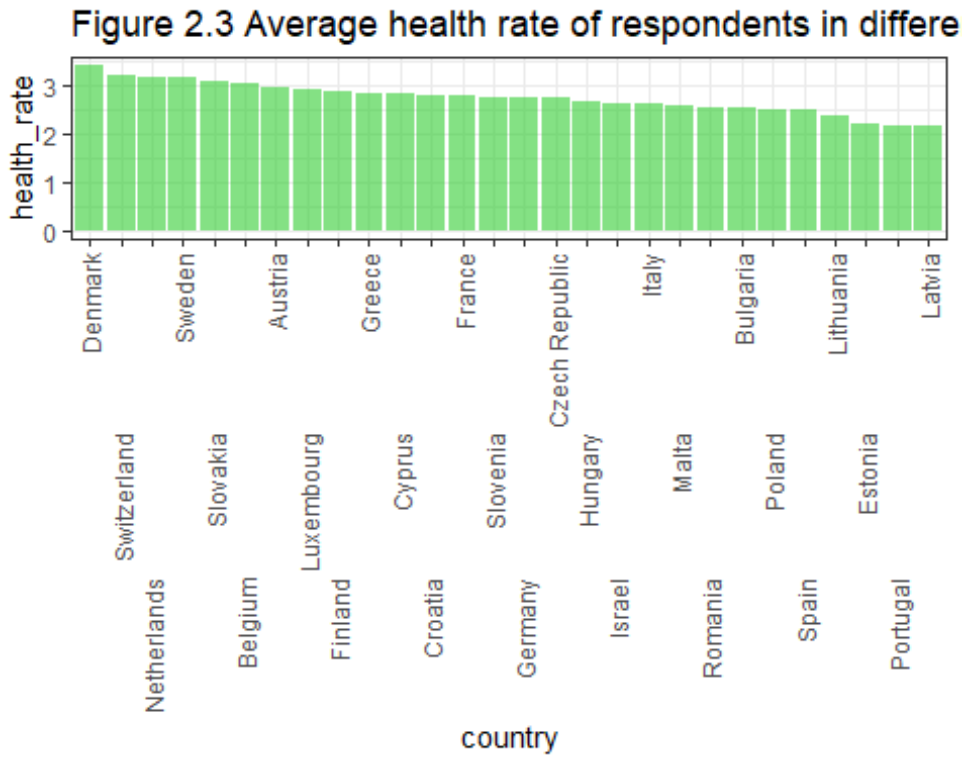
```
  summarise_at(vars(health_rate1), list(health_rate = mean)),desc(health_rat
e)))
```

```
## # A tibble: 6 x 2
##   country1    health_rate
##   <fct>             <dbl>
## 1 Denmark            3.42
## 2 Switzerland        3.23
## 3 Netherlands        3.18
## 4 Sweden             3.17
## 5 Slovakia           3.09
## 6 Belgium            3.04
```

```
#calculate the mean value, show countries with the lowest scores
head(arrange(country3 %>%
  group_by(country1) %>%
  summarise_at(vars(health_rate1), list(health_rate = mean)),desc(-health_rat
e)))
```

```
## # A tibble: 6 x 2
##   country1  health_rate
##   <fct>           <dbl>
## 1 Latvia           2.16
## 2 Portugal         2.20
## 3 Estonia          2.23
## 4 Lithuania        2.40
## 5 Spain            2.51
## 6 Poland           2.52
```

```
#Exhibit the average health rate in different country by ggplot
country3_a <- country3 %>%
  group_by(country1) %>%
  summarise_at(vars(health_rate1), list(health_rate = mean))
#Build the barchart, ranked as descending
countrychart <- country3_a %>%
  ggplot(aes(x=reorder(country1,-health_rate), y=health_rate))+
  geom_bar(stat="identity", fill="limegreen", alpha=.6, width=.9)+
  scale_x_discrete(guide = guide_axis(n.dodge=3))+
  labs(title="Figure 2.3 Average health rate of respondents in different coun
tries")+
  xlab("country") +
  theme_bw()+
    theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1,(size=
5)))
suppressWarnings(countrychart)
```

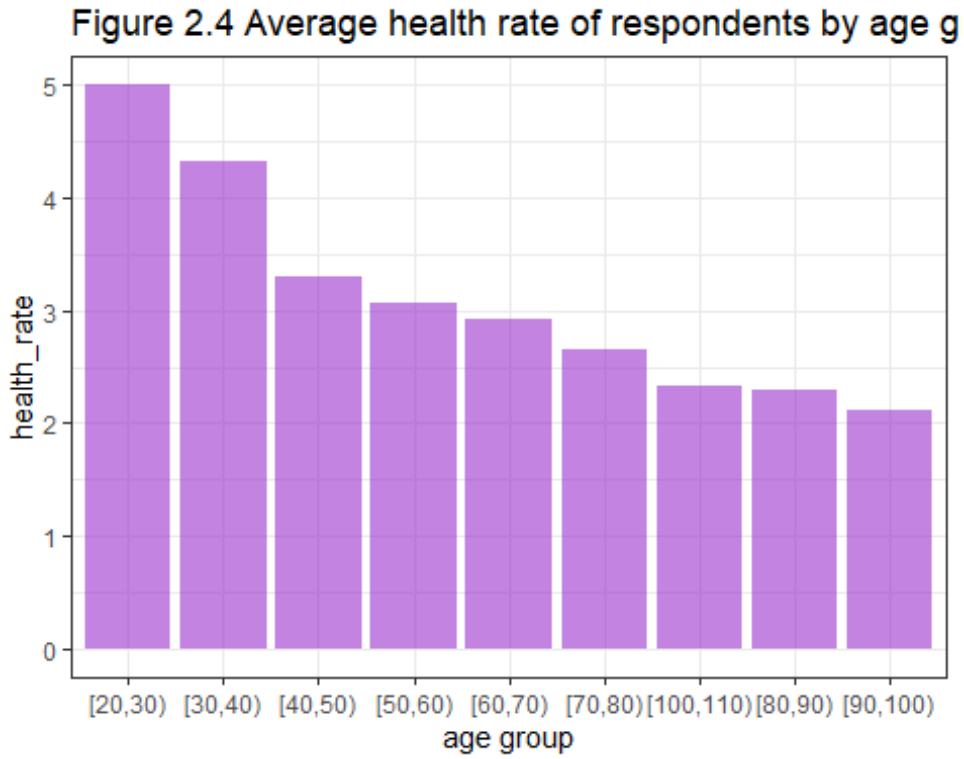


The six countries with the highest average health scores are: Denmark, Switzerland, Netherlands, Sweden, Slovakia, Belgium. The average score for the above countries is over 3 points. And the six countries with the lowest average health scores were: Latvia, Portugal, Estonia, Lithuania, Spain, Poland. Among them, the average score of Latvia, Portugal and Estonia is only about 2 points.

2.4 Average health rate of respondents by age group

```
#Exhibit the average health rate in different agegroup
agegroup_rate1<-data.frame(df$agegroup,df$health_rate)
#remove the missing value
agegroup3<-drop_na(rename(agegroup_rate1, agegroup1 =df.agegroup,health_rate1
 = df.health_rate))
#calculate the mean value
arrange(agegroup3 %>%
  group_by(agegroup1) %>%
  summarise_at(vars(health_rate1), list(health_rate = mean)))

## # A tibble: 9 x 2
##   agegroup1 health_rate
##   <fct>           <dbl>
## 1 [20,30)          5
## 2 [30,40)          4.32
## 3 [40,50)          3.30
## 4 [50,60)          3.07
```

```
## 5 [60,70)          2.93
## 6 [70,80)          2.65
## 7 [80,90)          2.30
## 8 [90,100)         2.11
## 9 [100,110)        2.33

#Exhibit the average health rate in different agegroup by ggplot
agegroup3_a <- agegroup3 %>%
  group_by(agegroup1) %>%
  summarise_at(vars(health_rate1), list(health_rate = mean))
#Build the barchart, ranked as descending
agechart <- agegroup3_a %>%
  ggplot(aes(x=reorder(agegroup1,-health_rate), y=health_rate))+
  geom_bar(stat="identity", fill="darkorchid", alpha=.6, width=.9)+
  labs(title="Figure 2.4 Average health rate of respondents by age group")+
  xlab("age group") +
  theme_bw()
agechart
```

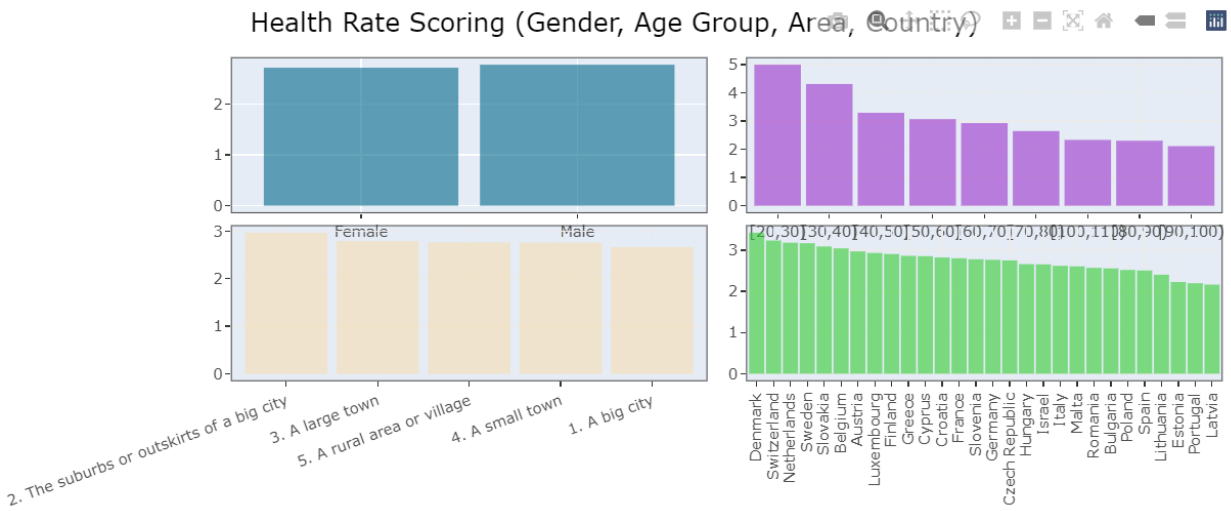


As can be seen from Figure 2.4, the older the respondents, the lower the average health score.

Interactive Sub-Plot

```
plotsubs <- subplot(ggplotly(genderchart,tooltip = c("y", "text")),
                    ggplotly(agechart,tooltip = c("y", "text")),
                    ggplotly(livingchart,tooltip = c("y", "text")),
```

```
                    ggplotly(countrychart,tooltip = c("y", "text"))
                    , nrows = 2) %>%
  layout(title = 'Health Rate Scoring (Gender, Age Group, Area, Country)',
         plot_bgcolor='#e5ecf6',
         xaxis = list(
           zerolinecolor = '#ffff',
           zerolinewidth = 2,
           gridcolor = 'ffff'),
         yaxis = list(
           zerolinecolor = '#ffff',
           zerolinewidth = 2,
           gridcolor = 'ffff'))

plotsubs
```

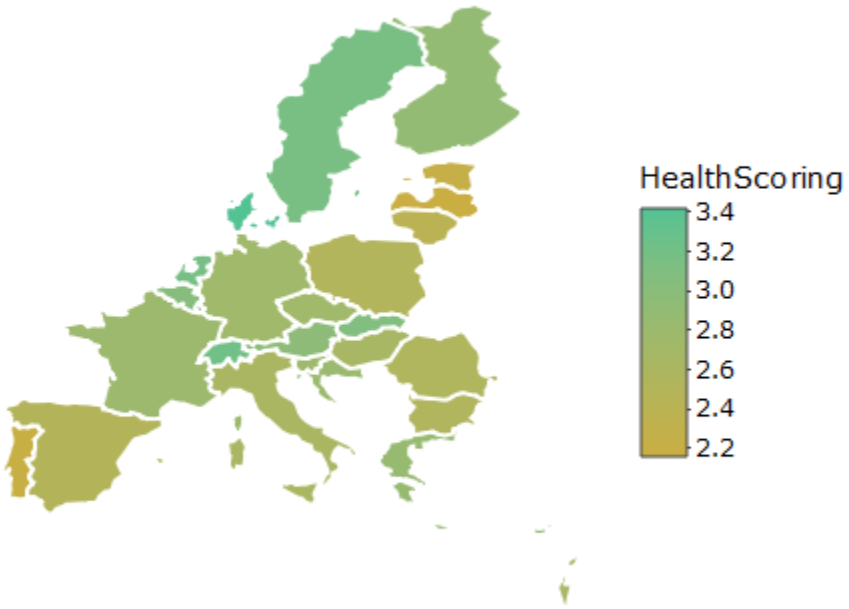



```
# options(browser = 'false')
# api_create(plotsubs, filename = "plotsubs")
# https://chart-studio.plotly.com/~plotlys2022355/58/#/
```

Map of Health Scores

```
# 1) Load Map Dataset from Library 2) Left Join Against Share8 Data 3) Filter
 out unrelated countries
mapdata3 <- map_data("world") %>%
    left_join(., country3_a %>% rename(., region = country1), by="region")
 %>%
    filter(!is.na(.$health_rate))

map3 <- ggplot(mapdata3, aes( x = long, y = lat, group=group, text=region)) +
  geom_polygon(color = "white",aes(fill = health_rate)) +
scale_fill_gradient(name = "HealthScoring",
  low = munsell::mnsl("5Y 7/8"),
  high = munsell::mnsl("5G 7/8"),
```

```
  na.value = "grey50"
)+
  theme(axis.text.x = element_blank(),
        axis.text.y = element_blank(),
        axis.ticks = element_blank(),
        axis.title.y=element_blank(),
        axis.title.x=element_blank(),
       rect = element_rect(fill = "transparent")
       )+

theme(
  panel.background = element_rect(fill = "transparent", colour = NA_character
_),
  panel.grid.major = element_blank(),
  panel.grid.minor = element_blank(),
  plot.background = element_rect(fill = "transparent",
                                 colour = NA_character_),
  legend.background = element_rect(fill = "transparent"),
  legend.box.background = element_rect(fill = "transparent"),
  legend.key = element_rect(fill = "transparent")
)
MapScore <- ggplotly(map3,tooltip = c("fill","text"))
MapScore
```



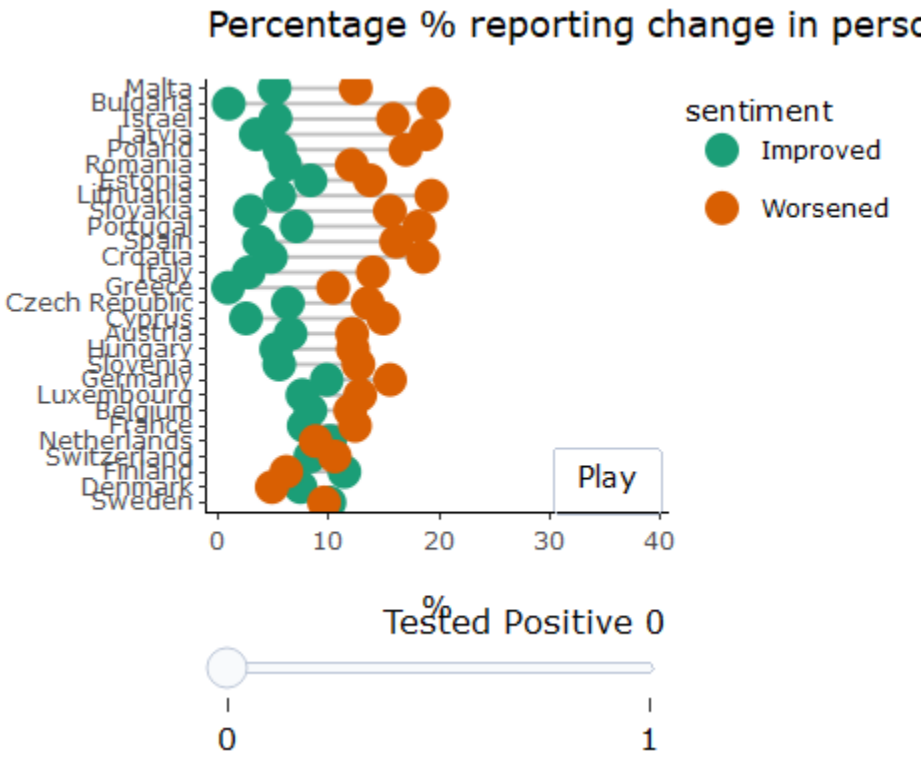

```
# options(browser = 'false')
# api_create(map3, filename = "healthmap")
# https://chart-studio.plotly.com/~plotlys2022355/60
```

Other visualization

We seek to understand how covid diagnosis influence Health Sentiment After 3 Months

```
#Create and Join Additional Tibble for percentage calculation
HealthChange <- data %>% group_by(country,health_change,test_positive) %>% ta
lly() %>% na.omit(.)
HealthChange2 <- HealthChange %>% group_by(country,test_positive) %>%
  summarise(Frequency = sum(n))
HealthChange <- left_join(HealthChange, HealthChange2, by=c("country","test_p
ositive"))

#Compute Percentages
HealthChange$percentage <- HealthChange$n / HealthChange$Frequency *100
HealthChange$percentage <- round(HealthChange$percentage, digits = 2)

#Filter irrelevant data point
HealthChange <- HealthChange %>% filter(health_change != '2. About the same
')%>% select(country,health_change,test_positive,percentage)
HealthChange <- rename(HealthChange, sentiment='health_change')
HealthChange$pctsort <- HealthChange$percentage
HealthChange$pctsort[HealthChange$sentiment=='1. Improved'] <- 0

#Rename
HealthChange <- HealthChange%>%
mutate(sentiment=recode(sentiment,
                            "1. Improved"='Improved',
                            "3. Worsened"='Worsened'))

p <- HealthChange %>%
  ggplot(aes(x= percentage, y= reorder(country,pctsort),text=(country), frame
 = test_positive)) +
  geom_line(aes(group = country, frame = test_positive),color="grey")+
    geom_point(aes(color=sentiment, frame = test_positive), size=4) +
  labs(y="country")+
  theme_classic()+
scale_color_brewer(palette = "Dark2")+
   labs(title = "Percentage % reporting change in personal health condition",
 x = "%",
       y = "")+

theme(
  panel.background = element_rect(fill = "transparent",
                                  colour = NA_character_), # necessary to avo
id drawing panel outline
  panel.grid.major = element_blank(), # get rid of major grid
  panel.grid.minor = element_blank(), # get rid of minor grid
  plot.background = element_rect(fill = "transparent",
                                 colour = NA_character_), # necessary to avoi
```

```
d drawing plot outline
  legend.background = element_rect(fill = "transparent"),
  legend.box.background = element_rect(fill = "transparent"),
  legend.key = element_rect(fill = "transparent")
)

## Warning: Ignoring unknown aesthetics: frame
## Ignoring unknown aesthetics: frame

fig <- ggplotly(p,tooltip = c("x","text")) %>%
  animation_opts(
    1000, easing = "elastic", redraw = FALSE
  ) %>%
  animation_button(
    x = 1, xanchor = "right", y = 0, yanchor = "bottom"
  ) %>%
  animation_slider(
    currentvalue = list(prefix = "Tested Positive ", font = list(color="black
"))
  )

fig
```




```
# options(browser = 'false')
# api_create(fig, filename = "HealthConditionSliders")
#https://chart-studio.plotly.com/~plotlys2022355/26
```

Chart shows % of respondents reporting their health outcomes and whether are they improving or worsening.

It is interesting to note for covid positive resposdents residing in “north-western” european countries,

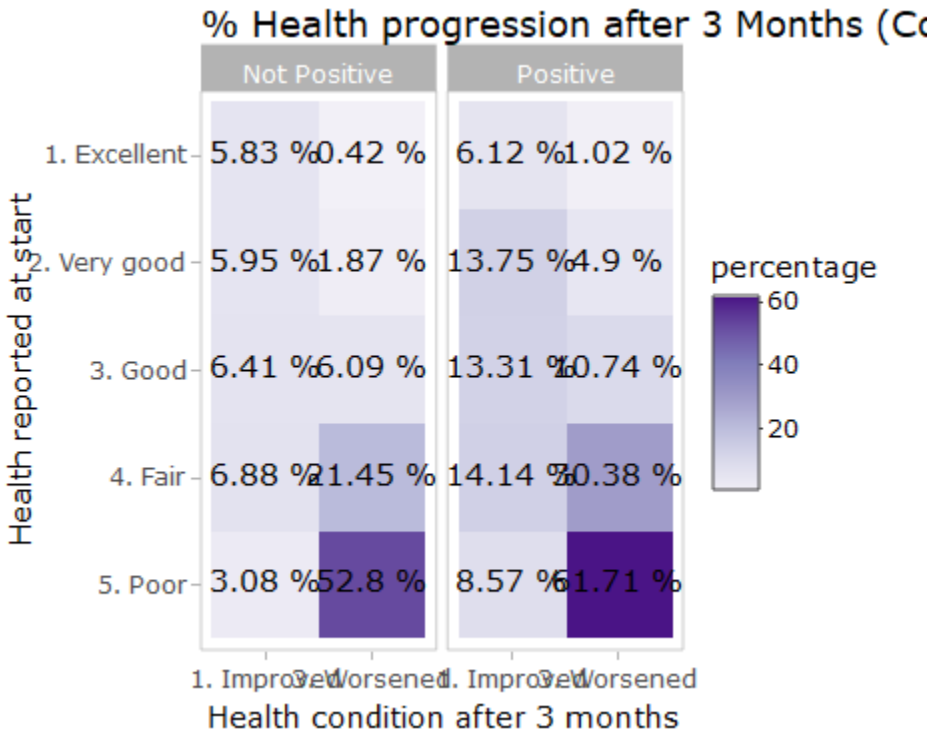
There are greater proportion of responsdents that reported improving health as compared to worsening.

GEOM TILE HEAT MAP

Alternatively, we also compare what respondents reported during the start of the survey and how are they feeling after 3 months. We subplot between covide diagnosed and non respectively. Result are plot on a tile heatmat.

```
A <- data[,c('country','gender','agegroup','health_rate','health_change','tes
t_positive')]

Otherdf <- A%>% filter(test_positive!=1) %>%
            group_by(health_rate, health_change)%>% tally()%>%
            rename(., total='n')%>%na.omit(.)
Otherdf$covidstatus = 'Not Positive'

positivedf <- A%>% filter(test_positive==1)%>%
                group_by(health_rate, health_change)%>% tally()%>%
                rename(., total='n')%>%na.omit(.)
positivedf$covidstatus = 'Positive'

plottb <- union_all(Otherdf, positivedf)

#Percentage Computation
plottbgroup <- plottb %>% group_by(health_rate,covidstatus) %>%
  summarise(Frequency = sum(total))
plottb <- left_join(plottb, plottbgroup, by=c("health_rate","covidstatus"))
plottb$percentage <- plottb$total / plottb$Frequency *100
plottb$percentage <- round(plottb$percentage, digits = 2)
plottb <- plottb %>% filter(health_change != '2. About the same')

# plot
E<- plottb %>%
ggplot(mapping = aes(y = reorder(health_rate, desc(health_rate)), x = health_
change, text=paste(percentage,"%"))) +
geom_tile(mapping = aes(fill = percentage)) +
  scale_fill_distiller(palette = "Purples", direction = 1) +
  theme_light() +
  labs(title="% Health progression after 3 Months (Covid Negative vs Positive
 respondents)",
       y="Health reported at start",
       x="Health condition after 3 months",
       color=NULL)+
```

```
theme(
  panel.background = element_rect(fill = "transparent",
                                  colour = NA_character_), # necessary to avo
id drawing panel outline
  panel.grid.major = element_blank(), # get rid of major grid
  panel.grid.minor = element_blank(), # get rid of minor grid
  plot.background = element_rect(fill = "transparent",
                                 colour = NA_character_), # necessary to avoi
d drawing plot outline
  legend.background = element_rect(fill = "transparent"),
  legend.box.background = element_rect(fill = "transparent"),
  legend.key = element_rect(fill = "transparent")
)
F <- E + facet_grid(cols = vars(covidstatus)) + geom_text(aes(label = paste(p
ercentage,"%")))
Eplotly<-ggplotly(F,tooltip = c("x", "text"))
Eplotly
```



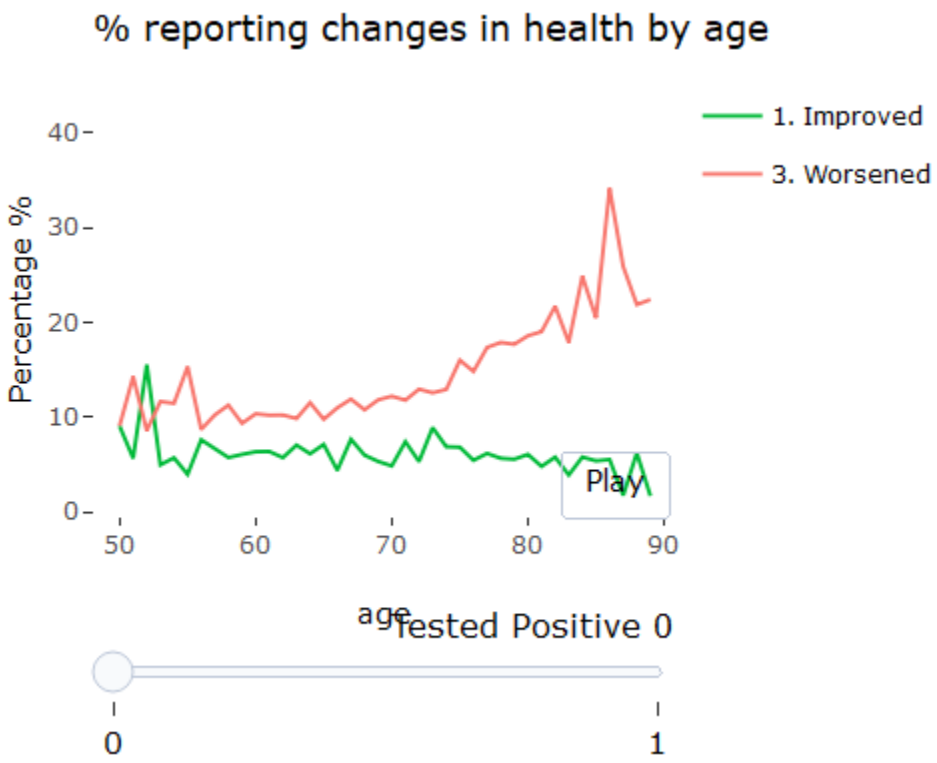

```
# options(browser = 'false')
# api_create(Eplotly, filename = "HealthChangeSubplots")
#https://chart-studio.plotly.com/~plotlys2022355/40
```

As expected, covid diagnosis magnifies the impact of worsening health

We can also look at how age and covid both affects health outcomes

```r
A <- data[,c('country','gender','age','health_change','test_positive')]%>%na.
omit(.)
HlthChangegp <- A %>% group_by(age, health_change, test_positive)  %>% tally
()

#% Computation
Hlthgp <- A %>% group_by(age, test_positive)  %>% tally()%>% rename(., total=
'n')
HlthChangegp <- merge(HlthChangegp, Hlthgp, by=c('age','test_positive'), all.
x=TRUE)
HlthChangegp$percentage <- HlthChangegp$n/HlthChangegp$total * 100
HlthChangegp$percentage <- round(HlthChangegp$percentage, digits = 2)

#Filter out irrelevant data points and outliers
HlthChangegp <- HlthChangegp %>% filter(health_change != '2. About the same
')%>% select(age,health_change,n, total, percentage,test_positive)
HlthChangegp <- HlthChangegp%>% filter(age>=50 & age<90)



# plot
D <- ggplot(HlthChangegp, aes(x=age)) +
  geom_line(aes(y=percentage, col=health_change, frame = test_positive)) +
  labs(title="% reporting changes in health by age",
       # subtitle="Drawn from Long Data format",
       # caption="Source: Economics",
       y="Percentage %",
       color=NULL) +
  scale_color_manual(labels = c("1. Improved", "3. Worsened"),
                     values = c("1. Improved"="#00ba38", "3. Worsened"="#f876
6d"))+

theme(
  panel.background = element_rect(fill = "transparent",
                                  colour = NA_character_), # necessary to avo
id drawing panel outline
  panel.grid.major = element_blank(), # get rid of major grid
  panel.grid.minor = element_blank(), # get rid of minor grid
  plot.background = element_rect(fill = "transparent",
                                 colour = NA_character_), # necessary to avoi
d drawing plot outline
  legend.background = element_rect(fill = "transparent"),
  legend.box.background = element_rect(fill = "transparent"),
  legend.key = element_rect(fill = "transparent")
  # ,legend.title=element_text(color="white")
  # ,legend.text=element_text(color="white")
  # ,plot.title=element_text(color="white")
  # ,axis.text.y=element_text(color="white")
  # ,axis.text.x=element_text(color="white")
  # ,axis.title.y=element_text(color="white")
```

```
  # ,axis.title.x=element_text(color="white")
)  # turn off minor grid

## Warning: Ignoring unknown aesthetics: frame

Dplotly<-ggplotly(D,tooltip = c("x","text","text")) %>%
  animation_opts(
    1000, easing = "elastic", redraw = FALSE
  ) %>%
  animation_button(
    x = 1, xanchor = "right", y = 0, yanchor = "bottom"
  ) %>%
  animation_slider(
    currentvalue = list(prefix = "Tested Positive ", font = list(color="black
"))
  )

Dplotly
```




```
# options(browser = 'false')
# api_create(Dplotly, filename = "AgeAnimationDistribution")
#https://chart-studio.plotly.com/~plotlys2022355/24
```

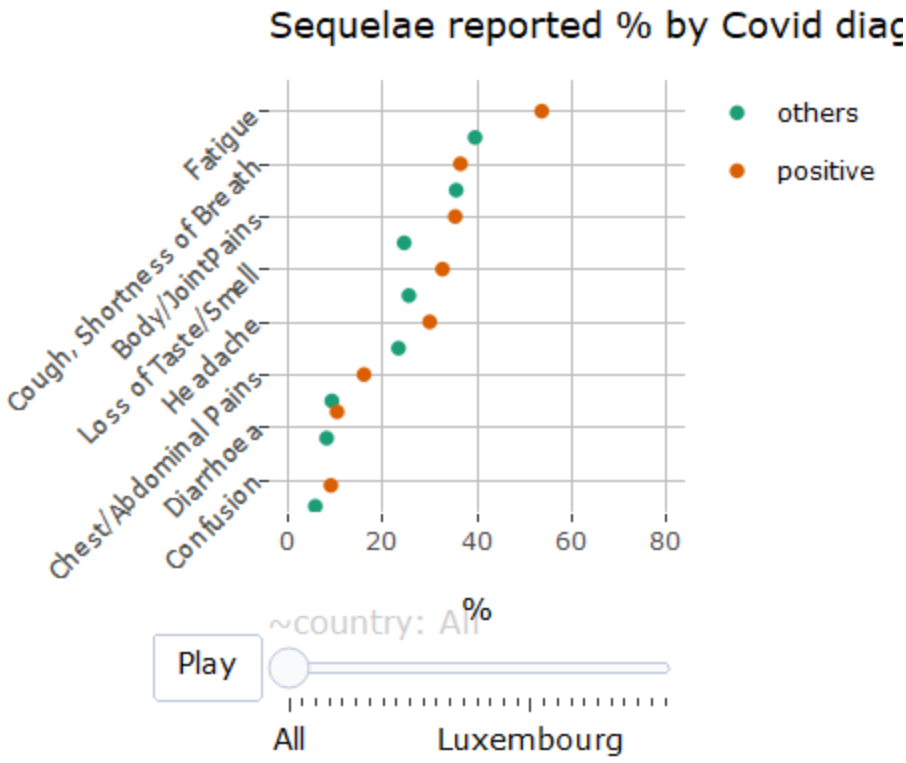
Naturally, health is going to progression worsen with age. This effect is magnified with covid diagnosis. However, we also noted increase % of improved health outcome for older covid patient. Could this be related to medical facilities in different countries?

Finally, we can look at the prevalence of Sequaia against Covid Status

```
sequelaedf <- data[,c('country','agegroup','test_positive','sequelae1','seque
lae2','sequelae3','sequelae4','sequelae5','sequelae6','sequelae7','sequelae8
')]

#Assign 0 to NA, in order gather sequalae sample results for participant that
 did not fill in covid test surveys.
sequelaedf$test_positive[is.na(sequelaedf$test_positive)]=0

# Result for covid positive respondents
sequelaepos <- sequelaedf %>%
filter(test_positive==1) %>%
select(c('country','sequelae1','sequelae2','sequelae3','sequelae4','sequelae5
','sequelae6','sequelae7','sequelae8')) %>%
na.omit
# dim(sequelaepos)

#1 Aggregate the occurence of each sequalae by country
#2 Tally number of row by country to act as denominator to compute percentage
s.
#3 left join both tibbles together

sequelaeposgroup<- left_join(
aggregate(sequelaepos[2:9], by=list(sequelaepos$country), FUN=c('sum')) %>% r
ename(., country='Group.1'),
sequelaepos %>% count(country, sort = TRUE),
by="country")


#Summarize additional result "sum of all countries together", tibble are unio
n together.
sequelaeposgroup <- union_all(
sequelaeposgroup,

sequelaeposgroup[2:10] %>%
summarise(sequelae1 = sum(sequelae1), sequelae2 = sum(sequelae2), sequelae3 =
 sum(sequelae3), sequelae4 = sum(sequelae4),
          sequelae5 = sum(sequelae5), sequelae6 = sum(sequelae6), sequelae7 =
 sum(sequelae7), sequelae8 = sum(sequelae8),
          n = sum(n)) %>%
       mutate(country = 'All')
    )

# dim(sequelaeposgroup)

#Similar Approach to the Code Row Above, except this is done for covid negati
ve / other participants
```

```
sequelaenotpos <- sequelaedf %>%
filter(test_positive!=1) %>%
select(c('country','sequelae1','sequelae2','sequelae3','sequelae4','sequelae5
','sequelae6','sequelae7','sequelae8'))%>%
na.omit
# dim(sequelaenotpos)

sequelaenotposgroup<- left_join(
aggregate(sequelaenotpos[2:9], by=list(sequelaenotpos$country), FUN=c('sum'))
 %>% rename(., country='Group.1'),
sequelaenotpos %>% count(country, sort = TRUE),
by="country")


sequelaenotposgroup <- union_all(
sequelaenotposgroup,
sequelaenotposgroup[2:10] %>%
summarise(sequelae1 = sum(sequelae1), sequelae2 = sum(sequelae2), sequelae3 =
 sum(sequelae3), sequelae4 = sum(sequelae4),
          sequelae5 = sum(sequelae5), sequelae6 = sum(sequelae6), sequelae7 =
 sum(sequelae7), sequelae8 = sum(sequelae8),
          n = sum(n)) %>%
       mutate(country = 'All')
    )
# dim(sequelaenotposgroup)

#Union the covid positive and other groupings together
sequelaegroupcombined <- union_all(
    sequelaeposgroup%>%mutate(covid_status = 'positive')
    ,
    sequelaenotposgroup%>%mutate(covid_status = 'others')
    )

# head(sequelaegroupcombined)

#Compute Percentage Occurence for all 8 sequalae, assign to new columns
for (val in (1:8))

{   calccol = paste('sequelae',val, sep = "")
    newcol = paste('s',val, sep = "")
    sequelaegroupcombined[newcol] <- round(sequelaegroupcombined[calccol]/seq
uelaegroupcombined$n*100,1)
    }

# dim(sequelaegroupcombined)
# head(sequelaegroupcombined)


sequelaegroupchart <- sequelaegroupcombined %>%
```

```
select(country,covid_status,s1,s2,s3,s4,s5,s6,s7,s8) %>%
pivot_longer(., cols =c(s1,s2,s3,s4,s5,s6,s7,s8), names_to = "Sequelae", valu
es_to = "percentage") %>%
mutate(Sequelae=recode(Sequelae, #Recoded for chart interpretation
                               "s1"='Fatigue',
                               "s2"='Cough, Shortness of Breath',
                               "s3"='Loss of Taste/Smell',
                               "s4"='Headache',
                               "s5"='Body/JointPains',
                               "s6"='Chest/Abdominal Pains',
                               "s7"='Diarrhoea',
                               "s8"='Confusion')) %>%
rename(., Covid_Status='covid_status')

G<-ggplot(sequelaegroupchart)+
    geom_point(aes(x = reorder(Sequelae,percentage), y = percentage, colour =
 Covid_Status, frame=country, text=paste(Sequelae,": ", percentage,"%")),
               position = position_dodge2(width = 1))+
    coord_flip() +
    theme(axis.text.y = element_text(angle = 45, vjust = 0.5, hjust=1,(size=
5)))+
    labs(title="Sequelae reported % by Covid diagnosis",
       y="%",
       x="",
       color=NULL)+
    scale_color_brewer(palette = "Dark2")+
    theme(panel.background = element_rect(fill = "transparent", colour = NA_c
haracter_),
          panel.grid = element_line(color = "Grey",size = 0.55,linetype = 2),
          plot.background = element_rect(fill = "transparent", colour = NA_ch
aracter_)
         )

## Warning: Ignoring unknown aesthetics: frame, text

Gplotly<-ggplotly(G,tooltip = c("colour", "text"))
Gplotly
```




```
# options(browser = 'false')
# api_create(Gplotly, filename = "SequelaePercentage")
```

In General, Covid Positive respondents reported higher occurrence of sequelae symptoms as opposed to those who are not diagnosed with covid. We do not that this might not be the case when looking at individual countries.

Logistic Modelling & Interpretation

As a whole, the share8 dataset provided contains 318 columns (+2 derived age variable) with 49253 observations. Because of the dependent variable(health_change) with three responses (1. Improved 2. About the same 3. Worsened), we use Multinomial Logistic Regression to build model.

```
attach(data)

#select variables
variable<-data.frame(health_change,sequelae1,sequelae2,sequelae3,sequelae4,se
quelae5,sequelae6,sequelae7,sequelae8)

#check missing value
head(is.na(variable))

##  health_change sequelae1 sequelae2 sequelae3 sequelae4 sequelae5 sequelae6
## [1,] FALSE      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE
## [2,] FALSE      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE
```

```
## [3,] FALSE      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE
## [4,] FALSE      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE
## [5,] FALSE      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE
## [6,] FALSE      TRUE      TRUE      TRUE      TRUE      TRUE      TRUE
##      sequelae7 sequelae8
## [1,]      TRUE      TRUE
## [2,]      TRUE      TRUE
## [3,]      TRUE      TRUE
## [4,]      TRUE      TRUE
## [5,]      TRUE      TRUE
## [6,]      TRUE      TRUE
```

```
#drop missing values
used<-na.omit(variable)
#check again
head(is.na(used))
```

```
## health_change sequelae1 sequelae2 sequelae3 sequelae4 sequelae5 sequelae6
## 8   FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE
## 9   FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE
## 24  FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE
## 44  FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE
## 45  FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE
## 69  FALSE     FALSE     FALSE     FALSE     FALSE     FALSE     FALSE
##    sequelae7 sequelae8
## 8      FALSE     FALSE
## 9      FALSE     FALSE
## 24     FALSE     FALSE
## 44     FALSE     FALSE
## 45     FALSE     FALSE
## 69     FALSE     FALSE
```

```
#change the value
used <- used %>%
mutate(health_change=recode(health_change,
                            "1. Improved"=2,
                            "2. About the same"=1,
                            "3. Worsened"=0))
```

The values for individual sequelae were already recoded to 1s and 0s earlier during data cleaning, hence it is not needed to be handled on this step.

```
str(used)
```

```
## 'data.frame':    4071 obs. of  9 variables:
##  $ health_change: num  1 1 1 1 1 1 1 0 1 1 ...
##  $ sequelae1    : num  0 0 0 0 0 1 1 1 0 0 ...
##  $ sequelae2    : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ sequelae3    : num  1 1 0 0 0 0 0 0 0 1 ...
##  $ sequelae4    : num  0 0 0 0 0 0 0 0 1 0 ...
##  $ sequelae5    : num  0 0 0 0 0 0 0 0 1 0 ...
```

```
##  $ sequelae6    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ sequelae7    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ sequelae8    : num  0 0 0 0 0 0 0 0 0 0 ...
##  - attr(*, "na.action")= 'omit' Named int [1:45182] 1 2 3 4 5 6 7 10 11 12
 ...
##   ..- attr(*, "names")= chr [1:45182] "1" "2" "3" "4" ...
```

```
#change the data type
used[ colnames(used) ] <- lapply(used[colnames(used) ], factor)
```

```
#check the change successfully or not
str(used)
```

```
## 'data.frame':    4071 obs. of  9 variables:
##  $ health_change: Factor w/ 3 levels "0","1","2": 2 2 2 2 2 2 2 1 2 2 ...
##  $ sequelae1    : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 2 1 1 ...
##  $ sequelae2    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
##  $ sequelae3    : Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 1 1 2 ...
##  $ sequelae4    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
##  $ sequelae5    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
##  $ sequelae6    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sequelae7    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sequelae8    : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "na.action")= 'omit' Named int [1:45182] 1 2 3 4 5 6 7 10 11 12
 ...
##   ..- attr(*, "names")= chr [1:45182] "1" "2" "3" "4" ...
```

```
#Load the multinom package
library(nnet)
```

```
#Since we are going to use health_change="3. Worsened" as the reference grou
p, we need relevel the group.
typeof(used$health_change)
```

```
## [1] "integer"
```

```
levels(used$health_change)
```

```
## [1] "0" "1" "2"
```

```
#define new variable and make the reference
used$newhealth_change<-relevel(used$health_change,ref = 1)
```

```
#include nothing
null_model<-multinom(used$newhealth_change~1)
```

```
## # weights:  6 (2 variable)
## initial  value 4472.450627
## iter  10 value 3469.508515
## iter  10 value 3469.508513
## iter  10 value 3469.508513
```

```
## final  value 3469.508513
## converged

summary(null_model)

## Call:
## multinom(formula = used$newhealth_change ~ 1)
##
## Coefficients:
##   (Intercept)
## 1   1.1610967
## 2  -0.5462245
##
## Std. Errors:
##   (Intercept)
## 1   0.0392357
## 2   0.0565387
##
## Residual Deviance: 6939.017
## AIC: 6943.017

#include all independent variables
full_model<-multinom(used$newhealth_change~used$sequelae1+used$sequelae2+used
$sequelae3+used$sequelae4+used$sequelae5+used$sequelae6+used$sequelae7+used$s
equelae8)

## # weights:  30 (18 variable)
## initial  value 4472.450627
## iter  10 value 3362.484693
## iter  20 value 3325.982340
## final  value 3324.788672
## converged

summary(full_model)

## Call:
## multinom(formula = used$newhealth_change ~ used$sequelae1 + used$sequelae2
 +
##     used$sequelae3 + used$sequelae4 + used$sequelae5 + used$sequelae6 +
##     used$sequelae7 + used$sequelae8)
##
## Coefficients:
##(Intercept) used$sequelae11 used$sequelae21 used$sequelae31 used$sequelae41
##1 1.7029409     -0.47844972      -0.3240059       0.3943957      -0.1666516
##2-0.4382988      0.08900326      -0.2559248       0.2998691      -0.6068302
## used$sequelae51 used$sequelae61 used$sequelae71 used$sequelae81
## 1    -0.1539485     -0.4322652     -0.25862549      -0.7510225
## 2    0.1839554      -0.1678917      0.02382543       0.0435898
##
## Std. Errors:
##(Intercept) used$sequelae11 used$sequelae21 used$sequelae31 used$sequelae41
```

```
## 1  0.06755890    0.09670772     0.09239625      0.09708501       0.1020431
## 2  0.09897365    0.13838066     0.13048941      0.13349540       0.1460793
##   used$sequelae51 used$sequelae61 used$sequelae71 used$sequelae81
## 1       0.1020096       0.1191702       0.1334782       0.1416943
## 2       0.1419204       0.1643516       0.1773703       0.1775956
##
## Residual Deviance: 6649.577
## AIC: 6685.577
```

```
#three methods of selecting variable
step(full_model,direction = "both",trace = 0)
```

```
## trying - used$sequelae1
## trying - used$sequelae2
## trying - used$sequelae3
## trying - used$sequelae4
## trying - used$sequelae5
## trying - used$sequelae6
## trying - used$sequelae7
## trying - used$sequelae8

## Call:
## multinom(formula = used$newhealth_change ~ used$sequelae1 + used$sequelae2
 +
##     used$sequelae3 + used$sequelae4 + used$sequelae5 + used$sequelae6 +
##     used$sequelae7 + used$sequelae8)
##
## Coefficients:
##(Intercept) used$sequelae11 used$sequelae21 used$sequelae31 used$sequelae41
##1 1.7029409     -0.47844972      -0.3240059       0.3943957      -0.1666516
##2-0.4382988      0.08900326      -0.2559248       0.2998691      -0.6068302
##   used$sequelae51 used$sequelae61 used$sequelae71 used$sequelae81
## 1      -0.1539485      -0.4322652     -0.25862549      -0.7510225
## 2       0.1839554      -0.1678917      0.02382543       0.0435898
##
## Residual Deviance: 6649.577
## AIC: 6685.577
```

```
step(full_model,direction = "forward",trace = 0)
```

```
## Call:
## multinom(formula = used$newhealth_change ~ used$sequelae1 + used$sequelae2
 +
##     used$sequelae3 + used$sequelae4 + used$sequelae5 + used$sequelae6 +
##     used$sequelae7 + used$sequelae8)
##
## Coefficients:
##(Intercept) used$sequelae11 used$sequelae21 used$sequelae31 used$sequelae41
##1 1.7029409     -0.47844972      -0.3240059       0.3943957      -0.1666516
##2-0.4382988      0.08900326      -0.2559248       0.2998691      -0.6068302
##   used$sequelae51 used$sequelae61 used$sequelae71 used$sequelae81
```

```
## 1      -0.1539485      -0.4322652     -0.25862549      -0.7510225
## 2       0.1839554      -0.1678917      0.02382543       0.0435898
##
## Residual Deviance: 6649.577
## AIC: 6685.577
```

```
step(full_model,direction = "backward",trace = 0)
```

```
## trying - used$sequelae1
## trying - used$sequelae2
## trying - used$sequelae3
## trying - used$sequelae4
## trying - used$sequelae5
## trying - used$sequelae6
## trying - used$sequelae7
## trying - used$sequelae8
```

```
## Call:
## multinom(formula = used$newhealth_change ~ used$sequelae1 + used$sequelae2
 +
##     used$sequelae3 + used$sequelae4 + used$sequelae5 + used$sequelae6 +
##     used$sequelae7 + used$sequelae8)
##
## Coefficients:
##(Intercept) used$sequelae11 used$sequelae21 used$sequelae31 used$sequelae41
##1 1.7029409     -0.47844972      -0.3240059       0.3943957      -0.1666516
##2-0.4382988      0.08900326      -0.2559248       0.2998691      -0.6068302
##   used$sequelae51 used$sequelae61 used$sequelae71 used$sequelae81
## 1      -0.1539485      -0.4322652     -0.25862549      -0.7510225
## 2       0.1839554      -0.1678917      0.02382543       0.0435898
##
## Residual Deviance: 6649.577
## AIC: 6685.577
```

```
#Check the Z-score (wald Z)
z <- summary(full_model)$coefficients/summary(full_model)$standard.errors
z
```

```
##(Intercept) used$sequelae11 used$sequelae21 used$sequelae31 used$sequelae41
##1 25.206759       -4.947379       -3.506700        4.062375       -1.633150
##2 -4.428439        0.643177       -1.961269        2.246288       -4.154116
##   used$sequelae51 used$sequelae61 used$sequelae71 used$sequelae81
## 1       -1.509157       -3.627294      -1.9375854      -5.3003006
## 2        1.296187       -1.021540       0.1343259       0.2454441
```

```
#2-tailed z test
p <- (1 - pnorm(abs(z), 0, 1)) * 2
p
```

```
##(Intercept) used$sequelae11 used$sequelae21 used$sequelae31 used$sequelae41
##1 0.00000e+00  7.521952e-07    0.0004537001    4.857601e-05    1.024375e-01
```

```
##2 9.49175e-06  5.201092e-01    0.0498477057    2.468557e-02    3.265482e-05
##   used$sequelae51 used$sequelae61 used$sequelae71 used$sequelae81
## 1       0.1312586    0.0002864071      0.05267383    1.156122e-07
## 2       0.1949111    0.3069987307      0.89314485    8.061126e-01

#sequelae5 and sequelae7 are not significant at 5% in any level,try to remove
 them.
model1<-multinom(used$newhealth_change~used$sequelae1+used$sequelae2+used$seq
uelae3+used$sequelae4+used$sequelae6+used$sequelae8)

## # weights:  24 (14 variable)
## initial  value 4472.450627
## iter  10 value 3362.004536
## iter  20 value 3331.830755
## final  value 3331.830674
## converged

summary(model1)

## Call:
## multinom(formula = used$newhealth_change ~ used$sequelae1 + used$sequelae2
 +
##     used$sequelae3 + used$sequelae4 + used$sequelae6 + used$sequelae8)
##
## Coefficients:
##(Intercept) used$sequelae11 used$sequelae21 used$sequelae31 used$sequelae41
##1 1.6910659      -0.5194292      -0.3362381       0.3604355      -0.2184148
##2  -0.4265207     0.1392990      -0.2432110       0.3229901      -0.5672804
##   used$sequelae61 used$sequelae81
## 1      -0.4955024     -0.81367586
## 2      -0.1279803      0.06699625
##
## Std. Errors:
##(Intercept) used$sequelae11 used$sequelae21 used$sequelae31 used$sequelae41
##1  0.06722951     0.0938762      0.09207328      0.09587301      0.09913518
##2  0.09838977     0.1337368      0.13009589      0.13192214      0.14249512
##   used$sequelae61 used$sequelae81
## 1       0.1162881       0.1392235
## 2       0.1603715       0.1742403
##
## Residual Deviance: 6663.661
## AIC: 6691.661

z1<- summary(model1)$coefficients/summary(model1)$standard.errors
z1

##(Intercept) used$sequelae11 used$sequelae21 used$sequelae31 used$sequelae41
##1   25.153624     -5.533129       -3.651853        3.759509       -2.203201
##2   -4.335011      1.041591       -1.869475        2.448339       -3.981051
##   used$sequelae61 used$sequelae81
```

```
## 1      -4.2609911      -5.8443877
## 2      -0.7980241       0.3845048

p1<- (1 - pnorm(abs(z1), 0, 1)) * 2
p1

##(Intercept) used$sequelae11 used$sequelae21 used$sequelae31 used$sequelae41
##1 0.000000e+00  3.145677e-08   0.0002603549    0.0001702468    2.758055e-02
##2 1.457531e-05  2.976013e-01   0.0615567469    0.0143516534    6.861113e-05
##   used$sequelae61 used$sequelae81
## 1    2.035223e-05    5.084349e-09
## 2    4.248565e-01    7.006043e-01
```

Interpretation

We build null_model at first as reference, and then we build full_model. The AIC decreases from 6943.017 to 6685.577, which means the full_model is better than null_model. The three methods of selecting variables to build the best model tell us the same results. The outputs of "both""forward""backward" are full_model with the lowest AIC.

g1(x)=ln[p(Y=1|X)/p(Y=0|x)]=1.70-0.48sequelae1-0.32sequelae2+0.39sequelae3-0.17sequelae4-0.15sequelae5-0.43sequelae6-0.26sequelae7-0.75sequelae8

g2(x)=ln[p(Y=2|X)/p(Y=0|x)]=-0.44+0.09sequelae1-0.26sequelae2+0.30sequelae3-0.61sequelae4+0.18sequelae5-0.17sequelae6+0.02sequelae7+0.04sequelae8

The meaning of coefficient is that for example β11 represents x1 changes 1 unit, odds ratio[p(Y=1|X)/p(Y=0|x)] changes e(-0.48) times, which means x1 increases, odds ratio decreases because the value of e(-0.48) is between 0 and 1. The research aims to find out which sequelae can significantly impact physical health in infected individuals. From the p-value table, we can see sequelae5 and sequelae7 are not significant at 5%. We try to remove them to build model1. All independent variables are significant in model1, though the p-value of sequelae 2 in g2(x) is little more than 0.05, which is 0.06, it can be accepted. Overall, body aches, joint pain attributed to respondent's Covid illness(sequelae5) and diarrhoea, nausea attributed to respondent's Covid illness(sequelae7) can't significantly impact physical health in infected individuals.

Resampling

Resampling methods and show how it improve your model performance Usually, the bootstrap method process is: 1. specify the number of samples "k". obtain a new sample of the same size as the original sample by resampling from the original sample. 2. estimate the intercept term and slop of the sample's "k".

```
#load the boot package
library(boot)
```

```r
#check the dataset
str(used)
```

```
## 'data.frame':    4071 obs. of  10 variables:
##  $ health_change   : Factor w/ 3 levels "0","1","2": 2 2 2 2 2 2 2 1 2 2
 ...
##  $ sequelae1       : Factor w/ 2 levels "0","1": 1 1 1 1 1 2 2 2 1 1 ...
##  $ sequelae2       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
##  $ sequelae3       : Factor w/ 2 levels "0","1": 2 2 1 1 1 1 1 1 1 2 ...
##  $ sequelae4       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
##  $ sequelae5       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 2 1 ...
##  $ sequelae6       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sequelae7       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ sequelae8       : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 1 1 1 ...
##  $ newhealth_change: Factor w/ 3 levels "0","1","2": 2 2 2 2 2 2 2 1 2 2
 ...
##  - attr(*, "na.action")= 'omit' Named int [1:45182] 1 2 3 4 5 6 7 10 11 12
 ...
##   ..- attr(*, "names")= chr [1:45182] "1" "2" "3" "4" ...
```

```r
summary(used)
```

```
##  health_change sequelae1 sequelae2 sequelae3 sequelae4 sequelae5 sequelae6
##  0: 853        0:2016    0:2593    0:2804    0:2910    0:2727    0:3478
##  1:2724        1:2055    1:1478    1:1267    1:1161    1:1344    1: 593
##  2: 494
##  sequelae7 sequelae8 newhealth_change
##  0:3667    0:3732    0: 853
##  1: 404    1: 339    1:2724
##                      2: 494
```

```r
head(used,10)
```

```
##  health_change sequelae1 sequelae2 sequelae3 sequelae4 sequelae5 sequelae6
## 8       1         0         0         1         0         0         0
## 9       1         0         0         1         0         0         0
## 24      1         0         0         0         0         0         0
## 44      1         0         0         0         0         0         0
## 45      1         0         0         0         0         0         0
## 69      1         1         0         0         0         0         0
## 70      1         1         0         0         0         0         0
## 102     0         1         0         0         0         0         0
## 112     1         0         1         0         1         1         0
## 180     1         0         0         1         0         0         0
##     sequelae7 sequelae8 newhealth_change
## 8           0         0                1
## 9           0         0                1
## 24          0         0                1
## 44          0         0                1
## 45          0         0                1
## 69          0         0                1
```

```
## 70          0         0                1
## 102         0         0                0
## 112         0         0                1
## 180         0         0                1
```

```
#check the dataset's type
lapply(used,class)
```

```
## $health_change
## [1] "factor"
##
## $sequelae1
## [1] "factor"
##
## $sequelae2
## [1] "factor"
##
## $sequelae3
## [1] "factor"
##
## $sequelae4
## [1] "factor"
##
## $sequelae5
## [1] "factor"
##
## $sequelae6
## [1] "factor"
##
## $sequelae7
## [1] "factor"
##
## $sequelae8
## [1] "factor"
##
## $newhealth_change
## [1] "factor"
```

```
#factor change into numeric
used$sequelae1<-as.numeric(as.character(used$sequelae1))
used$sequelae2<-as.numeric(as.character(used$sequelae2))
used$sequelae3<-as.numeric(as.character(used$sequelae3))
used$sequelae4<-as.numeric(as.character(used$sequelae4))
used$sequelae6<-as.numeric(as.character(used$sequelae6))
used$sequelae8<-as.numeric(as.character(used$sequelae8))
```

```
#check again
lapply(used,class)
```

```
## $health_change
## [1] "factor"
```

```
##
## $sequelae1
## [1] "numeric"
##
## $sequelae2
## [1] "numeric"
##
## $sequelae3
## [1] "numeric"
##
## $sequelae4
## [1] "numeric"
##
## $sequelae5
## [1] "factor"
##
## $sequelae6
## [1] "numeric"
##
## $sequelae7
## [1] "factor"
##
## $sequelae8
## [1] "numeric"
##
## $newhealth_change
## [1] "factor"
```

we need to create a simple function and put the dataset into the function.

```
#create a function
boot.f <- function(used, index){
    fit <- glm(used$newhealth_change~used$sequelae1+used$sequelae2+used$seque
lae3+used$sequelae4+used$sequelae6+used$sequelae8, family = binomial(), data
= used, subset = index)
    return( coef(fit))
}

#Use boot()to calculate 1000times' interception and slops
set.seed(1)
boot(used,boot.f,1000)

##
## ORDINARY NONPARAMETRIC BOOTSTRAP
##
##
## Call:
## boot(data = used, statistic = boot.f, R = 1000)
##
##
## Bootstrap Statistics :
```

```
##       original       bias    std. error
## t1*  1.8040229  0.001371493  0.06833692
## t2* -0.4200263 -0.003586279  0.09084659
## t3* -0.3197376  0.004802877  0.09082546
## t4*  0.3531647  0.001189192  0.09415876
## t5* -0.2846220 -0.001685399  0.09499409
## t6* -0.4161676  0.002531271  0.11781765
## t7* -0.5881859 -0.000680805  0.12796815
```

fitting by sampling from observations with putbacks.

```
summary(glm(used$newhealth_change~used$sequelae1+used$sequelae2+used$sequelae
3+used$sequelae4+used$sequelae6+used$sequelae8,family = binomial(),data = use
d))$coef
```

```
##                   Estimate Std. Error   z value      Pr(>|z|)
## (Intercept)     1.8040229 0.06649776 27.129079 4.470860e-162
## used$sequelae1 -0.4200263 0.09212672 -4.559224  5.134302e-06
## used$sequelae2 -0.3197376 0.08983161 -3.559299  3.718457e-04
## used$sequelae3  0.3531647 0.09339101  3.781570  1.558421e-04
## used$sequelae4 -0.2846220 0.09613040 -2.960791  3.068502e-03
## used$sequelae6 -0.4161676 0.11136556 -3.736950  1.862657e-04
## used$sequelae8 -0.5881859 0.12910410 -4.555904  5.216073e-06
```

Interpretation estimate SE SE(resampling) 1.8040229 0.06833692 0.06649776 -0.4200263 0.09084659 0.09212672 -0.3197376 0.09082546 0.08983161 0.3531647 0.09415876 0.09339101 -0.2846220 0.09499409 0.09613040 -0.4161676 0.11781765 0.11136556 -0.5881859 0.12796815 0.12910410 compare SE and SE(resampling), The numbers obtained are very close and we can say that the model we built performs very well.

Conclusion

The research aims to find out which sequelae can significantly impact physical health in infected individuals. By modeling multinomial logistic regression in the study, it was concluded that sequelae5(body aches, joint pain attributed to respondent's Covid illness) and sequelae7(diarrhea, nausea attributed to respondent's Covid illness) were not significant to Dependent variable (change in health in the last 3 months. And the other sequelae 1,2,3,4,6,8 were significant to dependent variable.

At the same time, the sequelae of covid-19 also show in the EDA those different countries, genders, ages, etc., will have different health situation. The big city respondents were more likely to feel healthy. The more developed countries, both men and women, will pay more attention to health. At the same time, the more developed countries have better medical standards, higher people's happiness index, happy mood, and comfortable life, which directly affect the health level. Similarly, women's health level is higher than that of men, because women's living habits are better than men. In short, when faced with the advent of Covid-19, different physical conditions will also reflect different consequences.

As new variants of SARS-CoV-2 emerge, these viral strains continue to cause long-term complications. One variant may cause more damaging long-term effects than other strains, and infected individuals could require additional support and more rapid and intense treatment. This makes managing the aftereffects of COVID-19 infection even more difficult for healthcare providers in the next stage of the pandemic. Continued monitoring of these individuals is necessary to comprehend the extent and severity of these long-term symptoms. A greater understanding of the pathogenesis and methods for treating long COVID is necessary to decrease the burden. Post-hospital care of COVID-19 survivors has become a significant research priority, and adequate guidelines for the management of these patients are still being developed (Sanyaolu et al,2022). Always take precautions until the COVID-19 pandemic is over. The number of people with long-term symptoms of Covid-19 appears to be decreasing over time. But the new coronavirus only emerged at the end of 2019 and began a global pandemic the following year, so there is a lack of long-term data studies. Even if Covid-19 patients appear to be recovering now, they may be at risk for life. At the same time, with the passage of time, the amount of data will expand, and it may become clearer in future research.

References:

Lamontagne, S. J., Winters, M. F., Pizzagalli, D. A., & Olmstead, M. C. (2021). Post-acute sequelae of COVID-19: evidence of mood & cognitive impairment. Brain, Behavior, & Immunity-Health, 17, 100347.

Ramakrishnan RK, Kashour T, Hamid Q, Halwani R and Tleyjeh IM (2021) Unraveling the Mystery Surrounding Post-Acute Sequelae of COVID-19. Front. Immunol. 12:686029. doi: 10.3389/fimmu.2021.686029

Sanyaolu, A., Marinkovic, A., Prakash, S. et al. Post-acute Sequelae in COVID-19 Survivors: an Overview. SN Compr. Clin. Med. 4, 91 (2022). https://doi.org/10.1007/s42399-022-01172-7