FIT3152 Assignment1

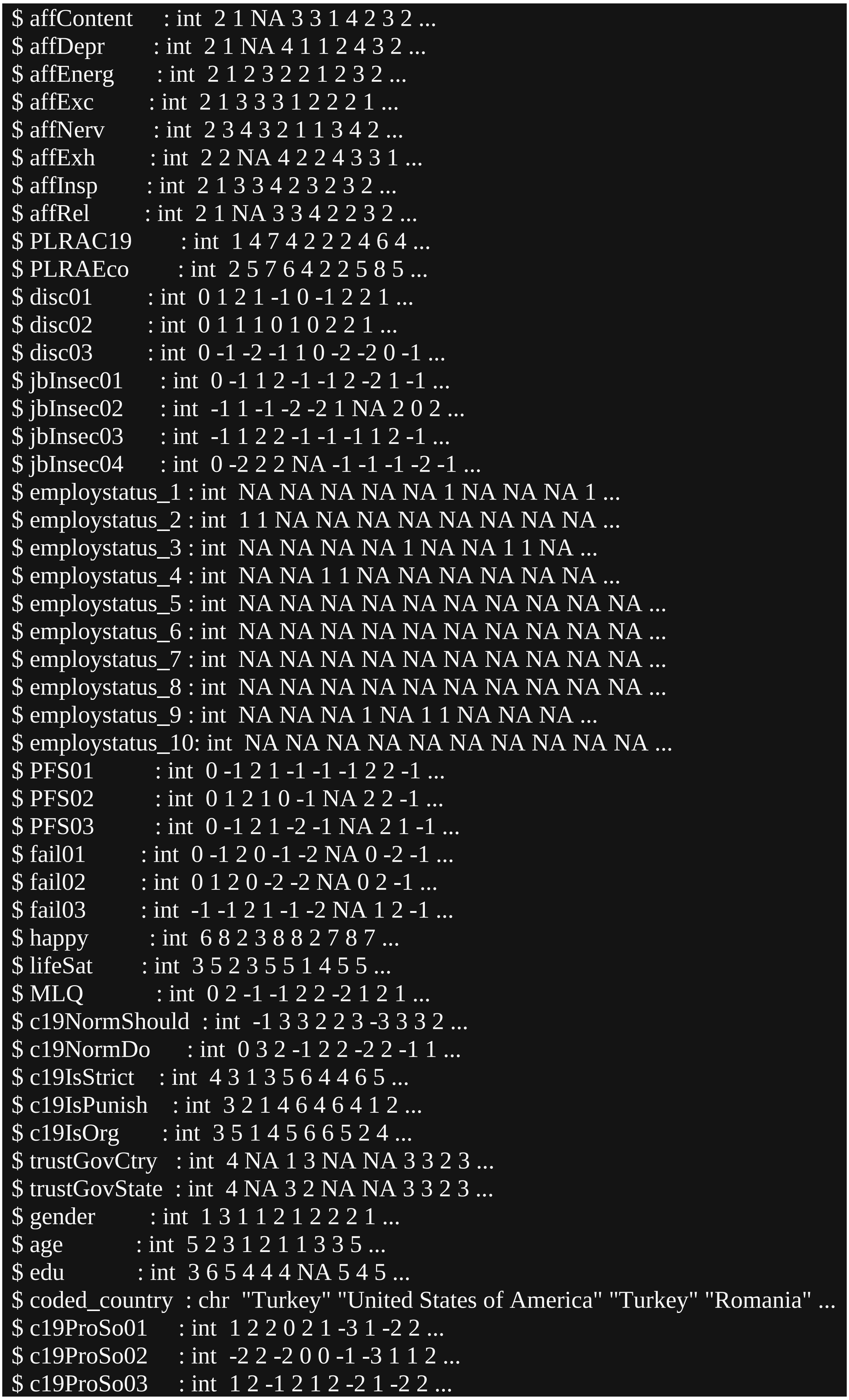
Student Name: DIZHEN LIANG

Student ID: 31240291

1(a)

```
#Instal necessary packages(dplyr, ggplot2)
install.packages("dplyr");
install.packages("ggplot2");

#import libraries
library(dplyr);
library(ggplot2);

#to make output reproducibale
set.seed(31240291)

#set working directory
setwd("C:/Users/DavidL/OneDrive/CS/FIT3152/A1")

#read csv file
covid <- read.csv("PsyCoronaBaselineExtract.csv", header =T)

#randomly sample 40000 rows of data
covid<-covid[sample(nrow(covid),40000),]

#structure of data (The codes and results are pasted together)
```

```
> str(covid)
'data.frame':	40000 obs. of  54 variables:
 $ affAnx        : int  2 3 2 5 3 2 2 4 4 3 ...
 $ affBor        : int  2 2 4 4 2 5 3 3 2 3 ...
 $ affCalm       : int  2 1 2 3 4 4 1 2 2 2 ...
```



```
 $ affContent     : int  2 1 NA 3 3 1 4 2 3 2 ...
 $ affDepr        : int  2 1 NA 4 1 1 2 4 3 2 ...
 $ affEnerg       : int  2 1 2 3 2 2 1 2 3 2 ...
 $ affExc         : int  2 1 3 3 3 1 2 2 2 1 ...
 $ affNerv        : int  2 3 4 3 2 1 1 3 4 2 ...
 $ affExh         : int  2 2 NA 4 2 2 4 3 3 1 ...
 $ affInsp        : int  2 1 3 3 4 2 3 2 3 2 ...
 $ affRel         : int  2 1 NA 3 3 4 2 2 3 2 ...
 $ PLRAC19        : int  1 4 7 4 2 2 2 4 6 4 ...
 $ PLRAEco        : int  2 5 7 6 4 2 2 5 8 5 ...
 $ disc01         : int  0 1 2 1 -1 0 -1 2 2 1 ...
 $ disc02         : int  0 1 1 1 0 1 0 2 2 1 ...
 $ disc03         : int  0 -1 -2 -1 1 0 -2 -2 0 -1 ...
 $ jbInsec01      : int  0 -1 1 2 -1 -1 2 -2 1 -1 ...
 $ jbInsec02      : int  -1 1 -1 -2 -2 1 NA 2 0 2 ...
 $ jbInsec03      : int  -1 1 2 2 -1 -1 -1 1 2 -1 ...
 $ jbInsec04      : int  0 -2 2 2 NA -1 -1 -1 -2 -1 ...
 $ employstatus_1 : int  NA NA NA NA NA 1 NA NA NA 1 ...
 $ employstatus_2 : int  1 1 NA NA NA NA NA NA NA NA ...
 $ employstatus_3 : int  NA NA NA NA 1 NA NA 1 1 NA ...
 $ employstatus_4 : int  NA NA 1 1 NA NA NA NA NA NA ...
 $ employstatus_5 : int  NA NA NA NA NA NA NA NA NA NA ...
 $ employstatus_6 : int  NA NA NA NA NA NA NA NA NA NA ...
 $ employstatus_7 : int  NA NA NA NA NA NA NA NA NA NA ...
 $ employstatus_8 : int  NA NA NA NA NA NA NA NA NA NA ...
 $ employstatus_9 : int  NA NA NA 1 NA 1 1 NA NA NA ...
 $ employstatus_10: int  NA NA NA NA NA NA NA NA NA NA ...
 $ PFS01          : int  0 -1 2 1 -1 -1 -1 2 2 -1 ...
 $ PFS02          : int  0 1 2 1 0 -1 NA 2 2 -1 ...
 $ PFS03          : int  0 -1 2 1 -2 -1 NA 2 1 -1 ...
 $ fail01         : int  0 -1 2 0 -1 -2 NA 0 -2 -1 ...
 $ fail02         : int  0 1 2 0 -2 -2 NA 0 2 -1 ...
 $ fail03         : int  -1 -1 2 1 -1 -2 NA 1 2 -1 ...
 $ happy          : int  6 8 2 3 8 8 2 7 8 7 ...
 $ lifeSat        : int  3 5 2 3 5 5 1 4 5 5 ...
 $ MLQ            : int  0 2 -1 -1 2 2 -2 1 2 1 ...
 $ c19NormShould  : int  -1 3 3 2 2 3 -3 3 3 2 ...
 $ c19NormDo      : int  0 3 2 -1 2 2 -2 2 -1 1 ...
 $ c19IsStrict    : int  4 3 1 3 5 6 4 4 6 5 ...
 $ c19IsPunish    : int  3 2 1 4 6 4 6 4 1 2 ...
 $ c19IsOrg       : int  3 5 1 4 5 6 6 5 2 4 ...
 $ trustGovCtry   : int  4 NA 1 3 NA NA 3 3 2 3 ...
 $ trustGovState  : int  4 NA 3 2 NA NA 3 3 2 3 ...
 $ gender         : int  1 3 1 1 2 1 2 2 2 1 ...
 $ age            : int  5 2 3 1 2 1 1 3 3 5 ...
 $ edu            : int  3 6 5 4 4 4 NA 5 4 5 ...
 $ coded_country  : chr  "Turkey" "United States of America" "Turkey" "Romania" ...
 $ c19ProSo01     : int  1 2 2 0 2 1 -3 1 -2 2 ...
 $ c19ProSo02     : int  -2 2 -2 0 0 -1 -3 1 1 2 ...
 $ c19ProSo03     : int  1 2 -1 2 1 2 -2 1 -2 2 ...
```

```
$ c19ProSo04     : int  0 2 1 2 2 2 -1 1 3 2 ...
```

This data is a long format consisting of 40000 rows and 54 columns, and there is an only one character type variable (text attribute) named coded_country.

The rest are all integer data type, and each has multiple categorical variables.

There are a lot of NAs (missing values) in multiple columns.

#Number of unique countries

```
unique(covid$coded_country)
  [1] "Turkey"                   "United States of America"    "Romania"                   "Chi
na"                      "Argentina"
  [6] "Thailand"                 "Greece"                   "Kosovo"                   "Hungary"
                 "Germany"
 [11] "Malaysia"                 "Republic of Serbia"         "Spain"                    "Hong
Kong S.A.R."           "Pakistan"
 [16] "Japan"                    ""                         "France"                   "Taiwan"
          "Kazakhstan"
 [21] "Philippines"              "South Korea"              "Netherlands"              "Austr
alia"                  "Peru"
 [26] "Tunisia"                  "Egypt"                    "Indonesia"                "Italy"
            "Canada"
 [31] "United Kingdom"           "South Africa"             "Singapore"                "Uk
raine"                   "Russia"
 [36] "Saudi Arabia"             "Brazil"                   "Poland"                   "Croatia"
               "Algeria"
 [41] "Israel"                   "Cyprus"                   "Iran"                     "United Arab E
mirates"        "Bosnia and Herzegovina"
 [46] "Vietnam"                  "Chile"                    "Jamaica"                  "Morocco"
                 "Finland"
 [51] "Bangladesh"               "Colombia"                 "India"                    "Palestine
"                  "Switzerland"
 [56] "Austria"                  "Nigeria"                  "Venezuela"                "Albania"
                "Luxembourg"
 [61] "Mongolia"                 "Sweden"                   "Belgium"                  "Mexico
"                      "Norway"
 [66] "Lebanon"                  "Portugal"                 "Iraq"                     "Trinidad an
d Tobago"        "Botswana"
 [71] "Mali"                     "Ireland"                  "New Zealand"              "El Salvad
or"              "Denmark"
 [76] "Dominican Republic"       "Slovakia"                 "Moldova"                  "Slo
venia"                   "Jordan"
 [81] "Estonia"                  "Czech Republic"           "Costa Rica"               "Monte
negro"                 "Libya"
 [86] "Iceland"                  "Kuwait"                   "Malta"                    "Bahrain"
            "Guatemala"
 [91] "Myanmar"                  "Uruguay"                  "Uzbekistan"               "Kyrgy
zstan"                   "Bulgaria"
```

```
 [96] "Georgia"                    "Latvia"                    "Lithuania"                 "Kenya"
               "Benin"
[101] "Oman"                      "Belarus"                    "Nepal"                     "Andorra"
                "United Republic of Tanzania"
[106] "Qatar"                    "Brunei"                    "Cambodia"                   "Panama"
                  "Armenia"
```

Boxplot to view the data range of one example from each concept

```
> #set arragement of plot
> par(mfrow=c(1,1))
> #select one example from each concept
> cod_imp <- covid %>%
+   select(affInsp, PLRAC19, disc02, jbInsec02, employstatus_10,
+          PFS01, fail02, lifeSat, c19NormShould,trustGovState, edu,
+          gender,age,c19ProSo01, c19ProSo02, c19ProSo03, c19ProSo04)
> #boxplotting with text on x-axis in specific orientation
> boxplot(cod_imp, las =2)
> #title of boxplot
> title("Boxplot of viewing value range one example from each concept")
```

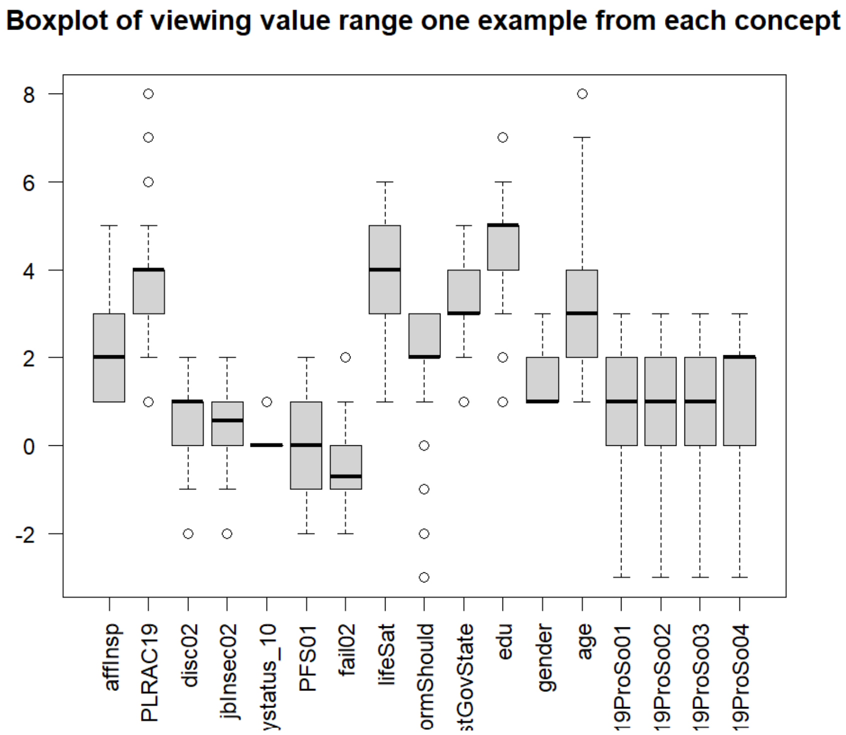


Figure 1.1.1 Boxplot of viewing data range

All the data range of the predictors and response variables in this dataset is from -4 to 8.

There are 111 unique countries

#Summary of dataset

```
> summary(covid)
    affAnx          affBor          affCalm         affContent       affDepr          affEnerg           a
ffExc
 Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.0
00   Min.   :1.000
 1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:
2.000   1st Qu.:1.000
 Median :3.000   Median :3.000   Median :3.000   Median :3.000   Median :2.000   Me
dian :3.000   Median :2.000
 Mean   :2.717   Mean   :2.707   Mean   :2.924   Mean   :2.682   Mean   :2.237   Mean
 :2.578   Mean   :2.146
 3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.:3.000   3rd Q
u.:3.000   3rd Qu.:3.000
 Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :
5.000   Max.   :5.000
 NA's   :533     NA's   :538     NA's   :545     NA's   :636     NA's   :629     NA's   :668
  NA's   :715
   affNerv         affExh          affInsp         affRel          PLRAC19         PLRAEco
disc01
 Min.   :1.000   Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.00
0   Min.   :-2.0000
 1st Qu.:2.000   1st Qu.:1.000   1st Qu.:1.00   1st Qu.:2.000   1st Qu.:3.000   1st Qu.:3.
000   1st Qu.: 0.0000
 Median :2.000   Median :2.000   Median :2.00   Median :3.000   Median :4.000   Med
ian :4.000   Median : 1.0000
 Mean   :2.584   Mean   :2.502   Mean   :2.44   Mean   :2.735   Mean   :3.547   Mean   :
4.397   Mean   : 0.6348
 3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.:3.00   3rd Qu.:4.000   3rd Qu.:4.000   3rd Q
u.:6.000   3rd Qu.: 1.0000
 Max.   :5.000   Max.   :5.000   Max.   :5.00   Max.   :5.000   Max.   :8.000   Max.   :8.
000   Max.   : 2.0000
 NA's   :578     NA's   :670     NA's   :694    NA's   :618     NA's   :147     NA's   :155
 NA's   :135
    disc02          disc03          jbInsec01        jbInsec02        jbInsec03        jbInsec04
 employstatus_1
 Min.   :-2.000   Min.   :-2.0000   Min.   :-2.000   Min.   :-2.000   Min.   :-2.000   Min.
 :-2.000   Min.   :1
 1st Qu.: 0.000   1st Qu.:-1.0000   1st Qu.:-2.000   1st Qu.: 0.000   1st Qu.:-1.000   1st
Qu.:-2.000   1st Qu.:1
 Median : 1.000   Median : 0.0000   Median :-1.000   Median : 1.000   Median : 0.000
  Median :-2.000   Median :1
 Mean   : 0.838   Mean   :-0.4084   Mean   :-0.599   Mean   : 0.565   Mean   : 0.059   M
ean   :-0.987   Mean   :1
 3rd Qu.: 1.000   3rd Qu.: 0.0000   3rd Qu.: 0.000   3rd Qu.: 1.000   3rd Qu.: 1.000   3r
d Qu.: 0.000   3rd Qu.:1
 Max.   : 2.000   Max.   : 2.0000   Max.   : 2.000   Max.   : 2.000   Max.   : 2.000   Ma
x.   : 2.000   Max.   :1
 NA's   :133      NA's   :134       NA's   :11061    NA's   :9971     NA's   :8492     NA's
 :13078    NA's   :34387
 employstatus_2  employstatus_3  employstatus_4  employstatus_5  employstatus_6  e
mploystatus_7  employstatus_8
 Min.   :1       Min.   :1       Min.   :1       Min.   :1       Min.   :1       Min.   :1       Min.   :
1
 1st Qu.:1       1st Qu.:1       1st Qu.:1       1st Qu.:1       1st Qu.:1       1st Qu.:1       1st
Qu.:1
 Median :1       Median :1       Median :1       Median :1       Median :1       Median :1
   Median :1
 Mean   :1       Mean   :1       Mean   :1       Mean   :1       Mean   :1       Mean   :1       M
ean   :1
 3rd Qu.:1       3rd Qu.:1       3rd Qu.:1       3rd Qu.:1       3rd Qu.:1       3rd Qu.:1       3r
d Qu.:1
 Max.   :1       Max.   :1       Max.   :1       Max.   :1       Max.   :1       Max.   :1       Max.
   :1
 NA's   :33279   NA's   :29113   NA's   :36517   NA's   :37977   NA's   :36897   NA's
 :36379   NA's   :39243
 employstatus_9  employstatus_10     PFS01             PFS02            PFS03           fail
01
 Min.   :1       Min.   :1       Min.   :-2.00000   Min.   :-2.0000   Min.   :-2.000   Min.   :-
2.00000
 1st Qu.:1       1st Qu.:1       1st Qu.:-1.00000   1st Qu.: 0.0000   1st Qu.:-1.000   1st Q
u.:-1.00000
 Median :1       Median :1       Median : 0.00000   Median : 1.0000   Median : 0.000
Median : 0.00000
 Mean   :1       Mean   :1       Mean   :-0.03258   Mean   : 0.5704   Mean   :-0.254   Mea
n   :-0.06322
 3rd Qu.:1       3rd Qu.:1       3rd Qu.: 1.00000   3rd Qu.: 1.0000   3rd Qu.: 1.000   3rd
Qu.: 1.00000
 Max.   :1       Max.   :1       Max.   : 2.00000   Max.   : 2.0000   Max.   : 2.000   Max.
 : 2.00000
 NA's   :31813   NA's   :39049   NA's   :162        NA's   :143       NA's   :143      NA's
 :138
     fail02           fail03           happy          lifeSat          MLQ         c19NormShould
   c19NormDo
 Min.   :-2.0000   Min.   :-2.0000   Min.   : 1.000   Min.   :1.000   Min.   :-3.0000   Mi
n.   :-3.000   Min.   :-3.0
 1st Qu.:-1.0000   1st Qu.: 0.0000   1st Qu.: 5.000   1st Qu.:3.000   1st Qu.: 0.0000   1s
t Qu.: 2.000   1st Qu.: 1.0
```

```
 Median :-1.0000   Median : 1.0000   Median : 7.000   Median :4.000   Median : 1.000
0   Median : 2.000   Median : 2.0  
 Mean   :-0.4126   Mean   : 0.3537   Mean   : 6.341   Mean   :4.147   Mean   : 0.8517
Mean   : 2.004   Mean   : 1.3  
 3rd Qu.: 0.0000   3rd Qu.: 1.0000   3rd Qu.: 8.000   3rd Qu.:5.000   3rd Qu.: 2.0000
3rd Qu.: 3.000   3rd Qu.: 2.0  
 Max.   : 2.0000   Max.   : 2.0000   Max.   :10.000   Max.   :6.000   Max.   : 3.0000   M
ax.   : 3.000   Max.   : 3.0  
 NA's   :142       NA's   :131       NA's   :516      NA's   :118     NA's   :122       NA's   :1
40      NA's   :135  
  c19IsStrict    c19IsPunish      c19IsOrg      trustGovCtry   trustGovState     gender  
       age       
 Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.00   Min.   :1.00
0   Min.   :1.000  
 1st Qu.:3.000   1st Qu.:2.000   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:2.00   1st Qu.:1.
000   1st Qu.:2.000  
 Median :4.000   Median :4.000   Median :4.000   Median :3.000   Median :3.00   Med
ian :1.000   Median :3.000  
 Mean   :4.117   Mean   :3.496   Mean   :3.896   Mean   :3.013   Mean   :3.08   Mean   :
1.387   Mean   :2.894  
 3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:4.000   3rd Qu.:4.00   3rd Q
u.:2.000   3rd Qu.:4.000  
 Max.   :6.000   Max.   :6.000   Max.   :6.000   Max.   :5.000   Max.   :5.00   Max.   :3.
000   Max.   :8.000  
 NA's   :161     NA's   :166     NA's   :158     NA's   :9330    NA's   :9416   NA's   :223  
   NA's   :247  
     edu        coded_country       c19ProSo01       c19ProSo02       c19ProSo03      c1
9ProSo04     
 Min.   :1.000   Length:40000       Min.   :-3.0000   Min.   :-3.0000   Min.   :-3.000   Mi
n.   :-3.000  
 1st Qu.:4.000   Class :character   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.000   1
st Qu.: 0.000  
 Median :5.000   Mode  :character   Median : 1.0000   Median : 1.0000   Median : 1.00
0   Median : 2.000  
 Mean   :4.409                      Mean   : 0.9735   Mean   : 0.6797   Mean   : 0.544   Mean
  : 1.283  
 3rd Qu.:5.000                      3rd Qu.: 2.0000   3rd Qu.: 2.0000   3rd Qu.: 2.000   3rd Q
u.: 2.000  
 Max.   :7.000                      Max.   : 3.0000   Max.   : 3.0000   Max.   : 3.000   Max.   :
3.000  
 NA's   :280                        NA's   :128       NA's   :135       NA's   :149      NA's   :156  
```

For the concepts of Affect (affAnx, affBor, affCalm, affContent, affDepr, affEnerg, affExc, affNerv, affExh, affInsp, and affRel), the range is from 1 (minimum) to 5 (maximum), with means ranging from 2.151 to 2.928

#and medians ranging from 2 to 3.

For the concept of Likelihood (PLRAC19 and PLRAEco), the range is from 1 (minimum) to 8 (maximum), with means of 3.554 and 4.402 and medians of 4 for both.

For the concepts of Social Discontent (disc01, disc02, and disc03), the range is from -2 (minimum) to 2 (maximum), with means ranging from -0.4027 to 0.8355 and medians of 1 for all.

For the concept of Job Insecurity (jbInsec01, jbInsec02, jbInsec03, and jbInsec04), the range is from -2 (minimum) to 2 (maximum), with means ranging from -0.982 to 0.56 and medians ranging from -2 to 0.
For the concept of Employment Status (employstatus_1 to employstatus_10), all values are 1 and there are no missing values.

For the concept of Employment Status (employstatus_1 to employstatus_10), all values are 1 and there are no missing values.

For the concepts of Perceived Financial Strain (PFS01, PFS02, and PFS03), the range is from -2 (minimum) to 2 (maximum), with means ranging from -0.2513 to 0.5716 and medians ranging from 0 to 1.

For the concept of Disempowerment (fail01, fail02, and fail03), the range is from -2 (minimum) to 2 (maximum), with means ranging from -0.4099 to 0.3569 and medians ranging from -1 to 1.

For the variable of Happy, the range is from 1 (minimum) to 10 (maximum), with a mean of 6.333 and a median of 7.

For the variable of Life Satisfaction (lifeSat), the range is from 1 (minimum) to 6 (maximum), with a mean of 4.139 and a median of 4.

For the concept of MLQ, the range is from -3 (minimum) to 3 (maximum), with a mean of 0.8434 and a median of 1.

For the concepts of Corona Community Injunctive norms (c19NormShould and c19NormDo), the range is from -3 (minimum) to 3 (maximum), with means of 2.002 and 1.298 and medians of 2 for both.

For the concepts of Corona Community Injunctive norms (c19IsStrict, c19IsPunish, and c19IsOrg),the range is from 1 (minimum) to 6 (maximum), with means ranging from 3.499 to 4.121 and medians of 4 for all.

For the concepts of Trust in Government Country(trustGovCtry and trustGovState), the range is from 1 (minimum) to 5 (maximum), with means of 3.02 and 3.083 and medians of 3 for both.

For the concept of Gender, the range is from 1 (minimum) to 3 (maximum), with a mean of 1.389 and a median of 1.

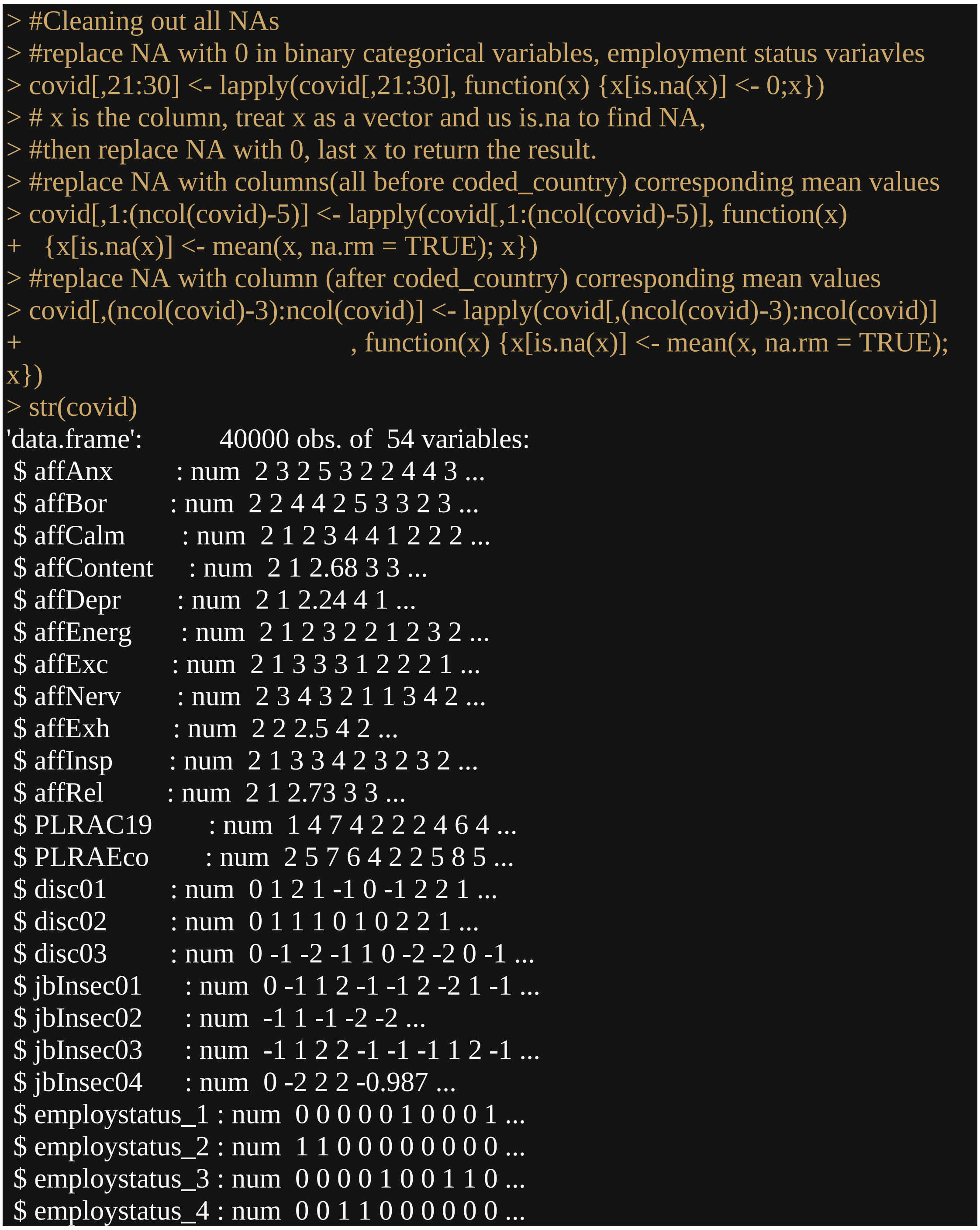
For the concept of Age, the range is from 1 (minimum) to 8 (maximum), with a mean of 2.895 and a median of 3.

For the concept of Education (edu), the range is from 1 (minimum) to 7 (maximum), with a mean of 4.403 and a median of 5.

For the concept of Coded Country (coded_country), it is a character variable with a length of 40000.

For the concepts of Covid-19 Pro-Social Behavior (c19ProSo01, c19ProSo02, and c19ProSo03), the range is from -3 (minimum) to 3 (maximum), with means ranging from 0.5434 to 0.9681 and medians of 1 for all.

1(b)



```
> #Cleaning out all NAs
> #replace NA with 0 in binary categorical variables, employment status variavles
> covid[,21:30] <- lapply(covid[,21:30], function(x) {x[is.na(x)] <- 0;x})
> # x is the column, treat x as a vector and us is.na to find NA,
> #then replace NA with 0, last x to return the result.
> #replace NA with columns(all before coded_country) corresponding mean values
> covid[,1:(ncol(covid)-5)] <- lapply(covid[,1:(ncol(covid)-5)], function(x)
+   {x[is.na(x)] <- mean(x, na.rm = TRUE); x})
> #replace NA with column (after coded_country) corresponding mean values
> covid[,(ncol(covid)-3):ncol(covid)] <- lapply(covid[,(ncol(covid)-3):ncol(covid)]
+                                               , function(x) {x[is.na(x)] <- mean(x, na.rm = TRUE);
x})
> str(covid)
'data.frame':	40000 obs. of  54 variables:
 $ affAnx        : num  2 3 2 5 3 2 2 4 4 3 ...
 $ affBor        : num  2 2 4 4 2 5 3 3 2 3 ...
 $ affCalm       : num  2 1 2 3 4 4 1 2 2 2 ...
 $ affContent    : num  2 1 2.68 3 3 ...
 $ affDepr       : num  2 1 2.24 4 1 ...
 $ affEnerg      : num  2 1 2 3 2 2 1 2 3 2 ...
 $ affExc        : num  2 1 3 3 3 1 2 2 2 1 ...
 $ affNerv       : num  2 3 4 3 2 1 1 3 4 2 ...
 $ affExh        : num  2 2 2.5 4 2 ...
 $ affInsp       : num  2 1 3 3 4 2 3 2 3 2 ...
 $ affRel        : num  2 1 2.73 3 3 ...
 $ PLRAC19       : num  1 4 7 4 2 2 2 4 6 4 ...
 $ PLRAEco       : num  2 5 7 6 4 2 2 5 8 5 ...
 $ disc01        : num  0 1 2 1 -1 0 -1 2 2 1 ...
 $ disc02        : num  0 1 1 1 0 1 0 2 2 1 ...
 $ disc03        : num  0 -1 -2 -1 1 0 -2 -2 0 -1 ...
 $ jbInsec01     : num  0 -1 1 2 -1 -1 2 -2 1 -1 ...
 $ jbInsec02     : num  -1 1 -1 -2 -2 ...
 $ jbInsec03     : num  -1 1 2 2 -1 -1 -1 1 2 -1 ...
 $ jbInsec04     : num  0 -2 2 2 -0.987 ...
 $ employstatus_1 : num  0 0 0 0 0 1 0 0 0 1 ...
 $ employstatus_2 : num  1 1 0 0 0 0 0 0 0 0 ...
 $ employstatus_3 : num  0 0 0 0 1 0 0 1 1 0 ...
 $ employstatus_4 : num  0 0 1 1 0 0 0 0 0 0 ...
$ employstatus_5 : num  0 0 0 0 0 0 0 0 0 0 ...
$ employstatus_6 : num  0 0 0 0 0 0 0 0 0 0 ...
$ employstatus_7 : num  0 0 0 0 0 0 0 0 0 0 ...
$ employstatus_8 : num  0 0 0 0 0 0 0 0 0 0 ...
$ employstatus_9 : num  0 0 0 1 0 1 1 0 0 0 ...
$ employstatus_10: num  0 0 0 0 0 0 0 0 0 0 ...
$ PFS01          : num  0 -1 2 1 -1 -1 -1 2 2 -1 ...
$ PFS02          : num  0 1 2 1 0 ...
$ PFS03          : num  0 -1 2 1 -2 ...
$ fail01         : num  0 -1 2 0 -1 ...
$ fail02         : num  0 1 2 0 -2 ...
$ fail03         : num  -1 -1 2 1 -1 ...
$ happy          : num  6 8 2 3 8 8 2 7 8 7 ...
$ lifeSat        : num  3 5 2 3 5 5 1 4 5 5 ...
$ MLQ            : num  0 2 -1 -1 2 2 -2 1 2 1 ...
$ c19NormShould  : num  -1 3 3 2 2 3 -3 3 3 2 ...
$ c19NormDo      : num  0 3 2 -1 2 2 -2 2 -1 1 ...
$ c19IsStrict    : num  4 3 1 3 5 6 4 4 6 5 ...
$ c19IsPunish    : num  3 2 1 4 6 4 6 4 1 2 ...
$ c19IsOrg       : num  3 5 1 4 5 6 6 5 2 4 ...
$ trustGovCtry   : num  4 3.01 1 3 3.01 ...
$ trustGovState  : num  4 3.08 3 2 3.08 ...
$ gender         : num  1 3 1 1 2 1 2 2 2 1 ...
$ age            : num  5 2 3 1 2 1 1 3 3 5 ...
$ edu            : num  3 6 5 4 4 ...
$ coded_country  : chr  "Turkey" "United States of America" "Turkey" "Romania" ...
$ c19ProSo01     : num  1 2 2 0 2 1 -3 1 -2 2 ...
$ c19ProSo02     : num  -2 2 -2 0 0 -1 -3 1 1 2 ...
$ c19ProSo03     : num  1 2 -1 2 1 2 -2 1 -2 2 ...
$ c19ProSo04     : num  0 2 1 2 2 2 -1 1 3 2 ...
```

Since there are many NAs in the dataset, the is.na would be needed to replace them with mean of each column to have no effect to the dataset. However, for the binary categorical variables (concept of Employment Status (employstatus_1 to employstatus_10) should replace NA with 0, since there are only 0 & 1 (NA usually means 0)

As we have a focus country, it would be appropriate to group data from Germany as a standalone dataset, and rest of the country into one. So, it would be easier to compare between them.

2(a)

#Group Germany and Others as two individual groups and calculate their corresponding mean values of four participant response

```
> germany = covid %>% filter(coded_country == "Germany")
> others = covid %>% filter(coded_country != "Germany")
> #no need na.rm since ,NA values are cleared in previous preocedures
> germany %>% group_by(coded_country)%>%
+   summarise(AC19PS1 = mean(c19ProSo01, na.rm=T), AC19PS2 = mean(c19ProS
o02, na.rm=T),
+             AC19PS3 = mean(c19ProSo03, na.rm=T), AC19PS4 = mean(c19ProSo04, n
a.rm=T))
# A tibble: 1 x 5
  coded_country AC19PS1 AC19PS2 AC19PS3 AC19PS4
  <chr>           <dbl>   <dbl>   <dbl>   <dbl>
1 Germany          1.09   0.171   0.438    1.16
> others %>% group_by(coded_country != "Germany")%>%
+   summarise(AC19PS1 = mean(c19ProSo01, na.rm=T), AC19PS2 = mean(c19ProS
o02, na.rm=T),
+             AC19PS3 = mean(c19ProSo03, na.rm=T), AC19PS4 = mean(c19ProSo04, n
a.rm=T))
# A tibble: 1 x 5
  `coded_country != "Germany"` AC19PS1 AC19PS2 AC19PS3 AC19PS4
  <lgl>                          <dbl>   <dbl>   <dbl>   <dbl>
1 TRUE                           0.965   0.689   0.552    1.29
```

On average, Germany only has higher value in c19ProSo01, the rest are lower than other countries as a group, especially much lower in terms of c19ProSo02 (Germany: 0.162, others: 0.689). Rest of the means in rest of the Response are: c19ProSo01 (Germany: 1.09, Others: 0.965), c19ProSo03 (Germany: 0.438, Others: 0.552)), c19ProSo04 (Germany: 1.16, Others: 1.29)

#Result of t.test for Four Response

```
> #Germany vs Other Countries on c19ProSo Response
> #Make columns to be called by columns names without calling name of data frame
> attach(covid)

> #null hypothesis, Germany'C19ProSo Response = Other Countries' C19ProSo Resp
onse
> #alternative hypothesis: Germany'C19ProSo Response != Other Countries' C19ProS
o
> t.test(c19ProSo01[coded_country == "Germany"], c19ProSo01[coded_country != "
Germany"]
+        , "greater",conf.level = 0.95)

            Welch Two Sample t-test

data:  c19ProSo01[coded_country == "Germany"] and c19ProSo01[coded_country !=
 "Germany"]
t = 2.8038, df = 1060.5, p-value = 0.002571
alternative hypothesis: true difference in means is greater than 0
95 percent confidence interval:
 0.05243358        Inf
sample estimates:
```

```
mean of x mean of y
1.0924749 0.9654689

> t.test(c19ProSo02[coded_country == "Germany"], c19ProSo02[coded_country != "
Germany"]
+        , "less",conf.level = 0.95)

              Welch Two Sample t-test

data:  c19ProSo02[coded_country == "Germany"] and c19ProSo02[coded_country !=
 "Germany"]
t = -9.5448, df = 1052.4, p-value < 2.2e-16
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
       -Inf -0.4281873
sample estimates:
mean of x mean of y
0.1714944 0.6889297

> t.test(c19ProSo03[coded_country == "Germany"], c19ProSo03[coded_country != "
Germany"]
+        , "less",conf.level = 0.95)

              Welch Two Sample t-test

data:  c19ProSo03[coded_country == "Germany"] and c19ProSo03[coded_country !=
 "Germany"]
t = -2.0376, df = 1050.8, p-value = 0.02092
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
        -Inf -0.02191975
sample estimates:
mean of x mean of y
0.4379088 0.5520487

> t.test(c19ProSo04[coded_country == "Germany"], c19ProSo04[coded_country != "
Germany"]
+        , "less",conf.level = 0.95)

              Welch Two Sample t-test

data:  c19ProSo04[coded_country == "Germany"] and c19ProSo04[coded_country !=
 "Germany"]
t = -2.3762, df = 1053.8, p-value = 0.008835
alternative hypothesis: true difference in means is less than 0
95 percent confidence interval:
        -Inf -0.03743899
sample estimates:
mean of x mean of y
 1.163746  1.285630
```

To see whether this result of comparison is consistent by repeating the same experiment, the t.test is used to compare whether the true population mean within calculate hypothesis (95% of the time) of the Germany is greater than true population mean within calculate hypothesis (95% of the time) of the Others. (Other Response: Germany's less than Others)

Since the p-values of c19ProSo01: 0.002571 < 0.05, c19ProSo02: 2.2e-16 < 0.05, c19ProSo03: 0.02092 < 0.05, c19ProSo04: 0.02092 < 0.05, the null hypothesis is rejected and conclude there are significant differences, in terms of four responses, between Germany and Others. The difference between two groups is larger if the t-value is larger (c19ProSo01: 2.8038, c19ProSo02: -9.5448, c19ProSo03: -2.0376, c19ProSo04: -2.3762, signs: + greater, - less)

2(b)

#Fit all four response variables individually to all predictors and find out the important predictors for each response variable

#min_p function to retrieve the best predictor with the smallest p-value for predicting different pro-social attitudes

```
> min_p <- function(p_va){
+   # find the index of the predictor with the smallest p-value
+   min_pvalue_index <- which.min(p_va[-1]) + 1
+ 
+   # get the name of the predictor with the smallest p-value
+   return(names(p_va)[min_pvalue_index])
+ }
> #All predictors for all four reponses for germany
> germany_p = germany[,1:(ncol(germany)-5)]
> #fit to linear model
> #Corona ProSocial Behavior 1 with its predictor
> PS1_fit_g = lm(formula = germany$c19ProSo01~., data = germany_p)
> pvalues_1 <- summary(PS1_fit_g)$coefficients[, 4]
> pvalues_1[pvalues_1 < 0.05]#treat it as list
  (Intercept)      PLRAC19        PFS02       fail02       fail03        happy       lifeSat 
      MLQ     c19NormDo 
 0.0015967803  0.0469865031  0.0079394365  0.0415586049  0.0260622391  0.0270
972707  0.0031738836  0.0240619134  0.0020953608 
  c19IsPunish      c19IsOrg trustGovState          edu 
 0.0375481300  0.0001610899  0.0090477281  0.0371546859 
> min_p(pvalues_1)
[1] "c19IsOrg"
> #Corona ProSocial Behavioure 2 with its predictor
> PS2_fit_g = lm(formula = germany$c19ProSo02~., data = germany_p)
> pvalues_2 <- summary(PS2_fit_g)$coefficients[, 4]
> pvalues_2[pvalues_2 < 0.05]#treat it as list
  (Intercept)       disc03       fail01          MLQ      c19IsOrg trustGovState        gender
         edu
 4.255898e-05  3.643425e-02  2.233895e-02  2.548528e-03  4.035174e-04  4.788259e
-03  4.784323e-02  6.365128e-05
> min_p(pvalues_2)
[1] "edu"
> #Corona ProSocial Behavioure 3 with its predictor
> PS3_fit_g = lm(formula = germany$c19ProSo03~., data = germany_p)
> pvalues_3 <- summary(PS3_fit_g)$coefficients[, 4]
> pvalues_3[pvalues_3 < 0.05]#treat it as list
   (Intercept)        affBor        affExh       PLRAC19 employstatus_4 employstatus_9
       PFS02          happy
  3.221317e-03   1.986326e-02   3.811159e-02   3.335246e-05   4.750683e-03   4.258
995e-02   6.726180e-03   5.937749e-03
      lifeSat      c19NormDo       c19IsOrg  trustGovState            age            edu
  5.087313e-03   1.494871e-03   1.785169e-02   3.966256e-03   2.258166e-03   4.940
743e-02
> min_p(pvalues_3)
[1] "PLRAC19"
> #Corona ProSocial Behavioure 4 with its predictor
> PS4_fit_g = lm(formula = germany$c19ProSo04~., data = germany_p)
> pvalues_4 <- summary(PS4_fit_g)$coefficients[, 4]
> pvalues_4[pvalues_4 < 0.05]#treat it as list
       affBor        PLRAC19        disc03      jbInsec01      jbInsec03 employstatus_9
   fail03          happy
  7.116053e-03   2.230207e-04   9.821026e-04   4.764195e-03   8.069568e-03   1.656
743e-02   3.779644e-02   3.996420e-02
          MLQ  c19NormShould      c19NormDo    c19IsPunish  trustGovState         gen
der            edu
  3.833190e-02   1.416585e-07   1.714410e-03   3.165985e-02   2.386471e-02   1.857
328e-02   4.763298e-02
> min_p(pvalues_4)
[1] "c19NormShould"
> #the predictors that have p-values less than 0.05(enough to reject null hypothesis)
> #in all four fitted linear models
> pvalues_1[pvalues_1 < 0.05 & pvalues_2 < 0.05 & pvalues_3 < 0.05 & pvalues_4 <
 0.05]
trustGovState           edu
  0.009047728   0.037154686
```

Since there are multiple response variables, separating all predictors into germany_p would improve the prediction on the response by avoiding inter-correlation between other responses with some of the predictors. The same reasons are applied to all the other dataset that are about be used to fit linear model in the rest of the report. From the results, most of the predictors have done a poor job of predicting pro-social attitudes for Germany, since their corresponding p-value is more or equal than the

0.05 which is not enough to reject the potential null hypothesis (the presence of predictor would not significantly improve the prediction on response variable). Apart from it, there are multiple predictors are important for pro-social attitudes individually.

For c19ProSo01, there are: PLRAC19, PFS02, fail02, fail03, happy, lifeSat, MLQ, c19NormDo c19IsPunish, c19IsOrg, trustGovState, edu. Best predictor: c19IsOrg

For c19ProSo02: disc03, fail01, MLQ, c19IsOrg trustGovState, gender, edu. Best predictor: "edu"

For c19ProSo03, there are: affBor, affExh, PLRAC19, employstatus_4, employstatus_9, PFS02, happy, lifeSat, c19NormDo, c19IsOr. Best predictor: "PLRAC19"

For c19ProSo04, there are: affBor, PLRAC19, disc03, jbInsec01, jbInsec03 employstatus_9, fail03, happy, MLQ, c19NormShould, c19NormDo. Best predictor: "c19NormShould"

In common, trustGovState, edu are the important predictors common for all four pro-social attitudes.

Under consideration of all four pro-social attitudes as one, the trustGovState, edu are actually the best predictors overall as they all play important roles for all those reponse variables.

There are different best predictors for four pro-social attitudes if they are considered by them individually, which are the predictors that have smallest p-values (like c19NormShould which has p-value: 1.415e-7

, for predicting c19ProSo03. In comparison, edu: 0.0372, trustGovState: 0.000905, their p-values are much larger than the c19NormShould's. A smaller p-value means that the data for the predictor variable is less likely to have occurred under the null hypothesis. Therefore, the predictor variable is more likely to be important in explaining the response variable.

#Plotting linear model of Germany

#arrange 4 graphs in 2x2

```
> par(mfrow=c(2,2))
> plot(PS1_fit_g)
```

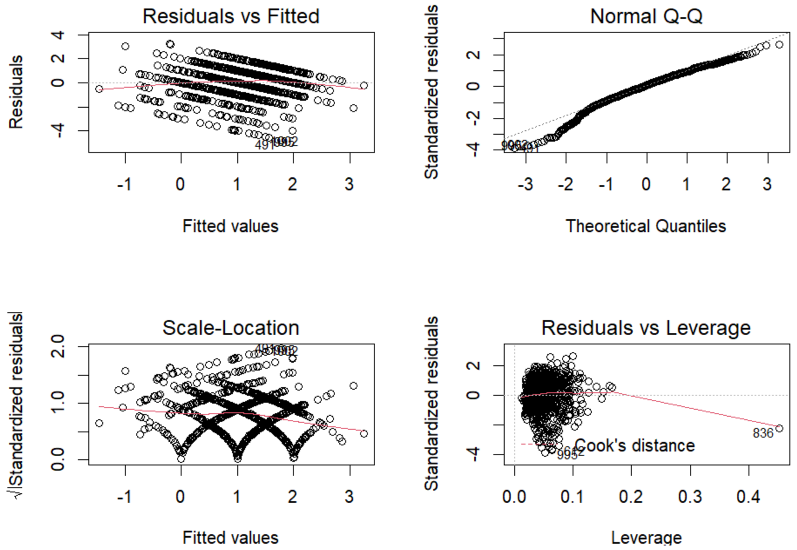


Figure 2.2.1. c19ProSo01

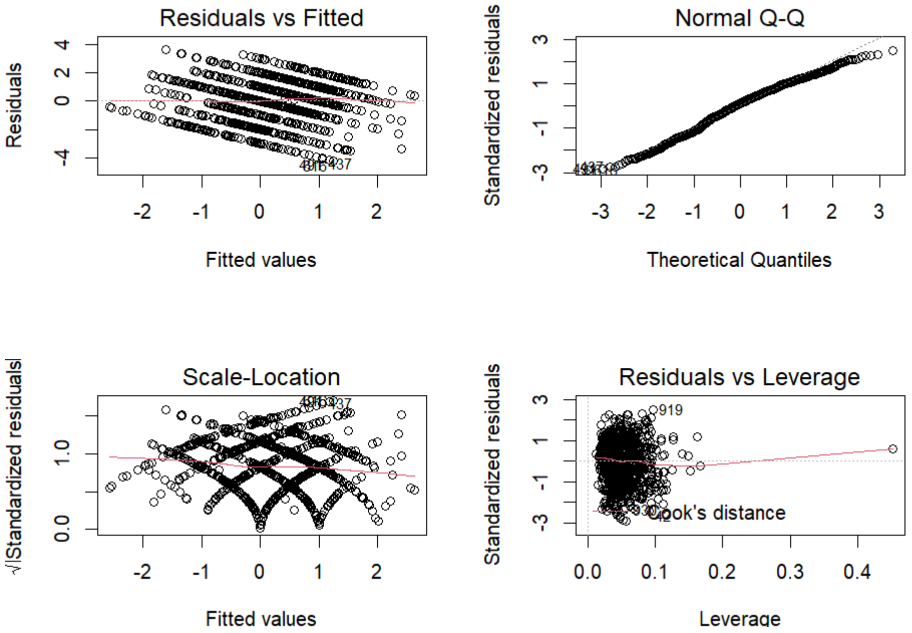


Figure 2.2.2. c19ProSo02

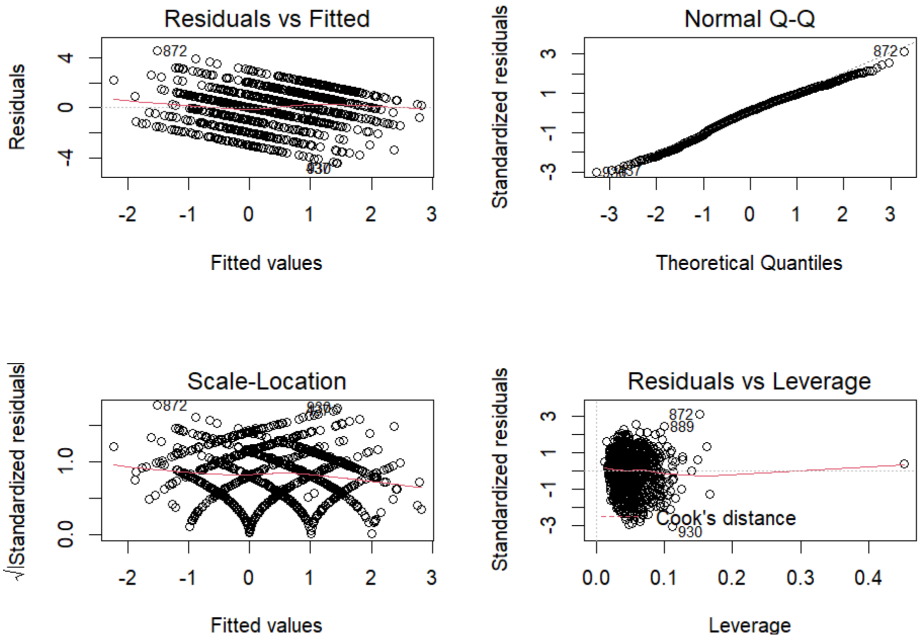


Figure 2.2.3. c19ProSo03

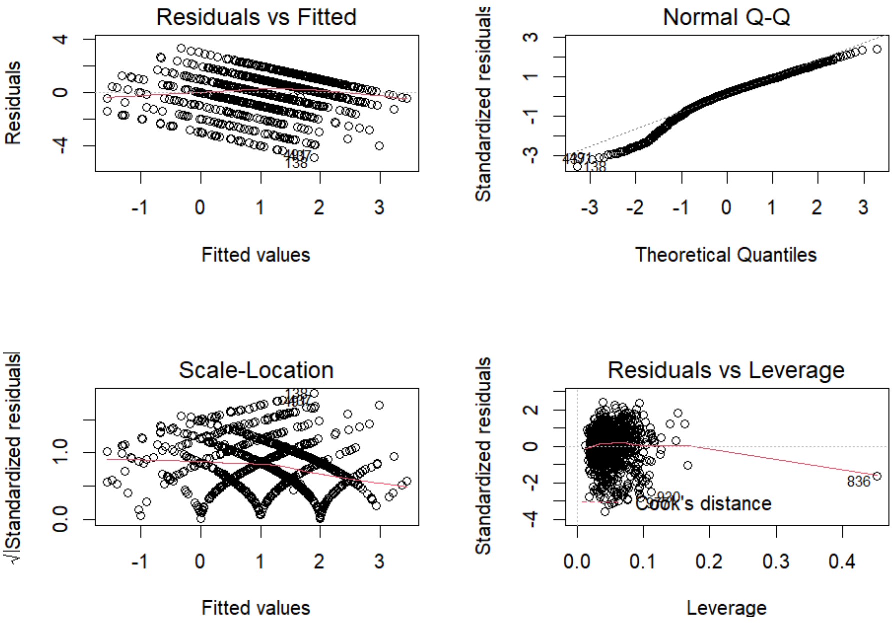
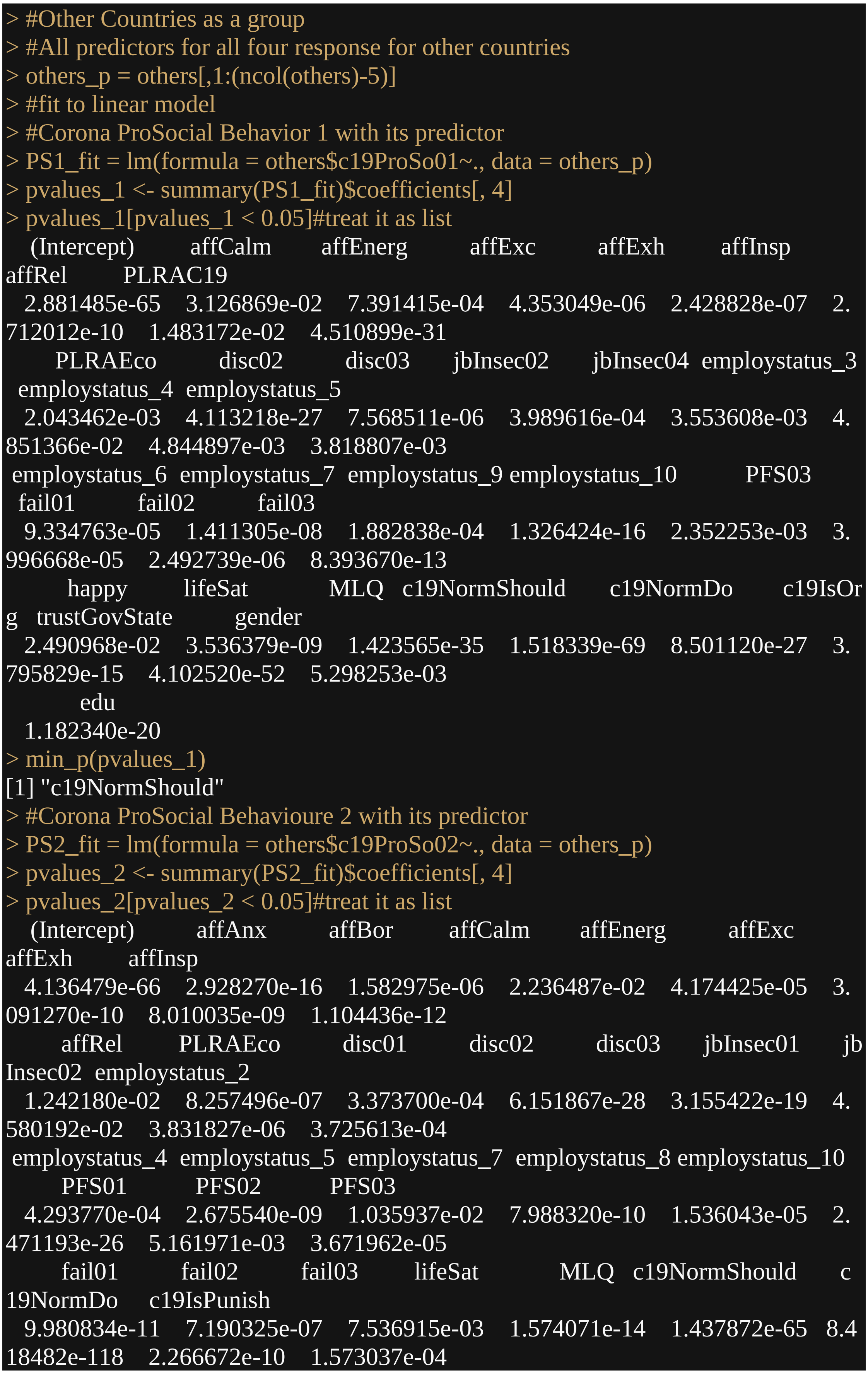


Figure 2.2.4. c19ProSo04

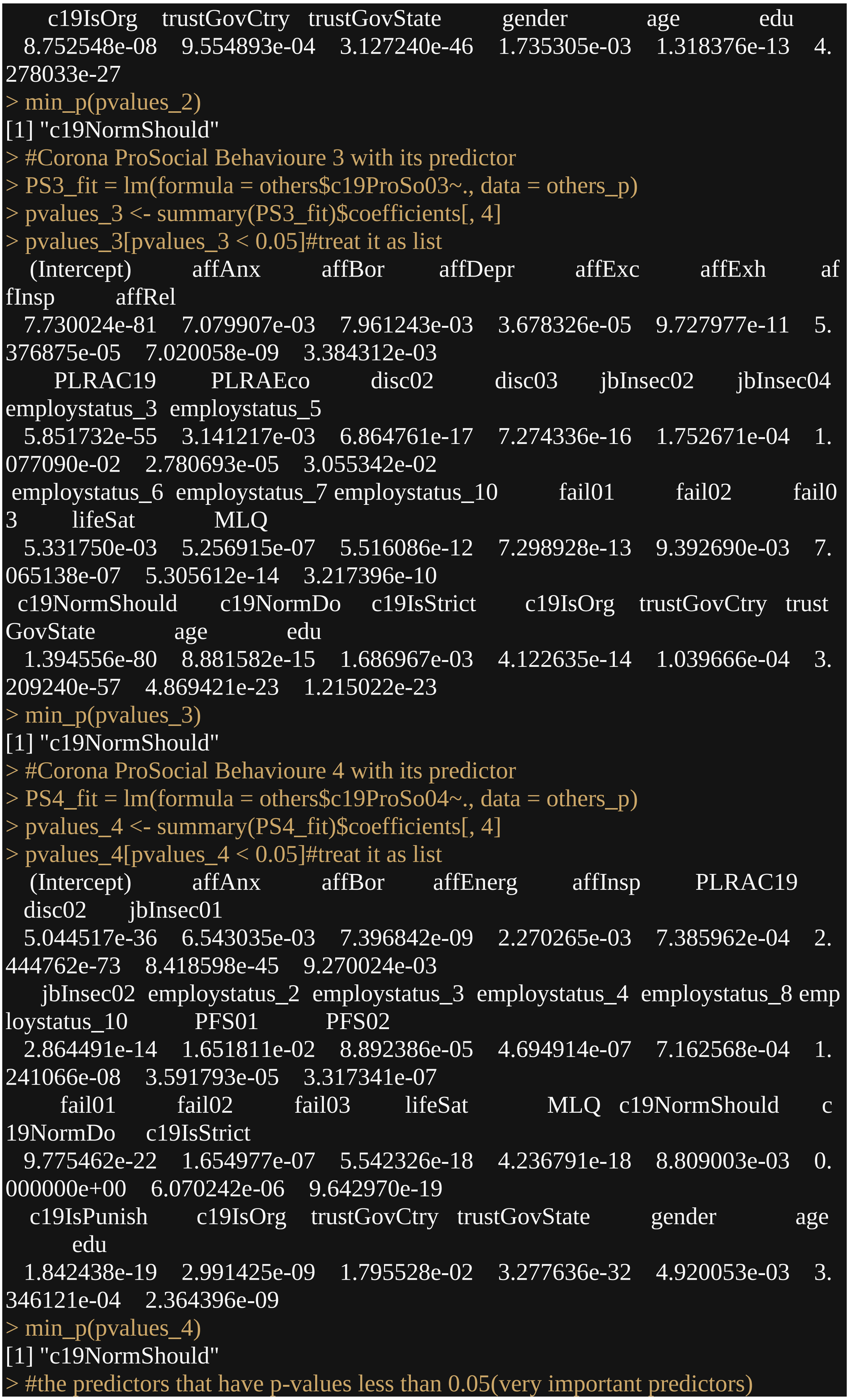
The lines in all four Residuals vs Fitted are almost straight which demonstrate the linearity of suggesting relationship between predictors and responses are linear

Residuals in all four graphs from linear model are normally distributed as the points almost lies on straight diagonal line in Normal Q-Q

2(c)



```
> #Other Countries as a group
> #All predictors for all four response for other countries
> others_p = others[,1:(ncol(others)-5)]
> #fit to linear model
> #Corona ProSocial Behavior 1 with its predictor
> PS1_fit = lm(formula = others$c19ProSo01~., data = others_p)
> pvalues_1 <- summary(PS1_fit)$coefficients[, 4]
> pvalues_1[pvalues_1 < 0.05]#treat it as list
   (Intercept)       affCalm       affEnerg        affExc        affExh       affInsp
affRel        PLRAC19
  2.881485e-65  3.126869e-02  7.391415e-04  4.353049e-06  2.428828e-07  2.
712012e-10  1.483172e-02  4.510899e-31
      PLRAEco        disc02        disc03      jbInsec02      jbInsec04  employstatus_3
  employstatus_4  employstatus_5
  2.043462e-03  4.113218e-27  7.568511e-06  3.989616e-04  3.553608e-03  4.
851366e-02  4.844897e-03  3.818807e-03
 employstatus_6  employstatus_7  employstatus_9 employstatus_10         PFS03
  fail01         fail02         fail03
  9.334763e-05  1.411305e-08  1.882838e-04  1.326424e-16  2.352253e-03  3.
996668e-05  2.492739e-06  8.393670e-13
        happy       lifeSat           MLQ   c19NormShould      c19NormDo       c19IsOr
g   trustGovState        gender
  2.490968e-02  3.536379e-09  1.423565e-35  1.518339e-69  8.501120e-27  3.
795829e-15  4.102520e-52  5.298253e-03
          edu
  1.182340e-20
> min_p(pvalues_1)
[1] "c19NormShould"
> #Corona ProSocial Behavioure 2 with its predictor
> PS2_fit = lm(formula = others$c19ProSo02~., data = others_p)
> pvalues_2 <- summary(PS2_fit)$coefficients[, 4]
> pvalues_2[pvalues_2 < 0.05]#treat it as list
   (Intercept)        affAnx        affBor       affCalm       affEnerg        affExc
affExh        affInsp
  4.136479e-66  2.928270e-16  1.582975e-06  2.236487e-02  4.174425e-05  3.
091270e-10  8.010035e-09  1.104436e-12
       affRel       PLRAEco        disc01        disc02        disc03      jbInsec01      jb
Insec02  employstatus_2
  1.242180e-02  8.257496e-07  3.373700e-04  6.151867e-28  3.155422e-19  4.
580192e-02  3.831827e-06  3.725613e-04
 employstatus_4  employstatus_5  employstatus_7  employstatus_8 employstatus_10
       PFS01         PFS02         PFS03
  4.293770e-04  2.675540e-09  1.035937e-02  7.988320e-10  1.536043e-05  2.
471193e-26  5.161971e-03  3.671962e-05
       fail01        fail02        fail03       lifeSat           MLQ   c19NormShould       c
19NormDo     c19IsPunish
  9.980834e-11  7.190325e-07  7.536915e-03  1.574071e-14  1.437872e-65  8.4
18482e-118  2.266672e-10  1.573037e-04
```



```
      c19IsOrg    trustGovCtry   trustGovState        gender           age           edu 
   8.752548e-08   9.554893e-04   3.127240e-46   1.735305e-03   1.318376e-13   4.
278033e-27 
> min_p(pvalues_2)
[1] "c19NormShould"
> #Corona ProSocial Behavioure 3 with its predictor
> PS3_fit = lm(formula = others$c19ProSo03~., data = others_p)
> pvalues_3 <- summary(PS3_fit)$coefficients[, 4]
> pvalues_3[pvalues_3 < 0.05]#treat it as list
    (Intercept)         affAnx         affBor        affDepr         affExc         affExh         af
fInsp         affRel 
   7.730024e-81   7.079907e-03   7.961243e-03   3.678326e-05   9.727977e-11   5.
376875e-05   7.020058e-09   3.384312e-03 
       PLRAC19        PLRAEco         disc02         disc03      jbInsec02      jbInsec04 
employstatus_3  employstatus_5 
   5.851732e-55   3.141217e-03   6.864761e-17   7.274336e-16   1.752671e-04   1.
077090e-02   2.780693e-05   3.055342e-02 
 employstatus_6  employstatus_7 employstatus_10         fail01         fail02         fail0
3        lifeSat            MLQ 
   5.331750e-03   5.256915e-07   5.516086e-12   7.298928e-13   9.392690e-03   7.
065138e-07   5.305612e-14   3.217396e-10 
  c19NormShould      c19NormDo     c19IsStrict       c19IsOrg    trustGovCtry   trust
GovState            age            edu 
   1.394556e-80   8.881582e-15   1.686967e-03   4.122635e-14   1.039666e-04   3.
209240e-57   4.869421e-23   1.215022e-23 
> min_p(pvalues_3)
[1] "c19NormShould"
> #Corona ProSocial Behavioure 4 with its predictor
> PS4_fit = lm(formula = others$c19ProSo04~., data = others_p)
> pvalues_4 <- summary(PS4_fit)$coefficients[, 4]
> pvalues_4[pvalues_4 < 0.05]#treat it as list
    (Intercept)         affAnx         affBor       affEnerg        affInsp        PLRAC19 
   disc02      jbInsec01 
   5.044517e-36   6.543035e-03   7.396842e-09   2.270265e-03   7.385962e-04   2.
444762e-73   8.418598e-45   9.270024e-03 
      jbInsec02  employstatus_2  employstatus_3  employstatus_4  employstatus_8 emp
loystatus_10          PFS01          PFS02 
   2.864491e-14   1.651811e-02   8.892386e-05   4.694914e-07   7.162568e-04   1.
241066e-08   3.591793e-05   3.317341e-07 
         fail01         fail02         fail03        lifeSat            MLQ   c19NormShould       c
19NormDo     c19IsStrict 
   9.775462e-22   1.654977e-07   5.542326e-18   4.236791e-18   8.809003e-03   0.
000000e+00   6.070242e-06   9.642970e-19 
    c19IsPunish       c19IsOrg    trustGovCtry   trustGovState         gender            age 
          edu 
   1.842438e-19   2.991425e-09   1.795528e-02   3.277636e-32   4.920053e-03   3.
346121e-04   2.364396e-09 
> min_p(pvalues_4)
[1] "c19NormShould"
> #the predictors that have p-values less than 0.05(very important predictors)
> #in all four fitted linear models
> pvalues_1[pvalues_1 < 0.05 & pvalues_2 < 0.05 & pvalues_3 < 0.05 & pvalues_4 <
 0.05]
   (Intercept)        affInsp         disc02      jbInsec02 employstatus_10         fail01
   fail02         fail03
  2.881485e-65   2.712012e-10   4.113218e-27   3.989616e-04   1.326424e-16   3.
996668e-05   2.492739e-06   8.393670e-13
      lifeSat            MLQ   c19NormShould      c19NormDo       c19IsOrg   trustGov
State            edu
  3.536379e-09   1.423565e-35   1.518339e-69   8.501120e-27   3.795829e-15   4.
102520e-52   1.182340e-20
```

Since all the other countries are treated as a group, there are many important predictors (p-value < 0.05 for each response). Overall, the important predictors are common in all four responses are: affInsp, disc02,   jbInsec02, employstatus_10, fail01, fail02, fail03, lifeSat, MLQ, c19NormShould, c19NormDo, c19IsOrg, trustGovState, edu.

Among all predictors, c19NormShould is the best predictors for four individual c19ProSo as it has smallest p-value for different extent in different c19ProSo. Explanation about the p-value is the same as the one for Germany.

By comparing with the focus country (Germany), the c19NormShould is only the best predictor for predicting Germany’s c19ProSo04, and not even exist as an important predictor (p-value < 0.05) for the rest of the pro-social attitudes. As for the frequent important predictors (edu, trustGovState) from Germany data, those predictors are also the important predictors for all pro-social attitudes in other countries.

```
> #ploting linear model of other countries,
> par(mfrow=c(2,2))
> plot(PS1_fit)
> plot(PS2_fit)
> plot(PS3_fit)
> plot(PS4_fit)
```

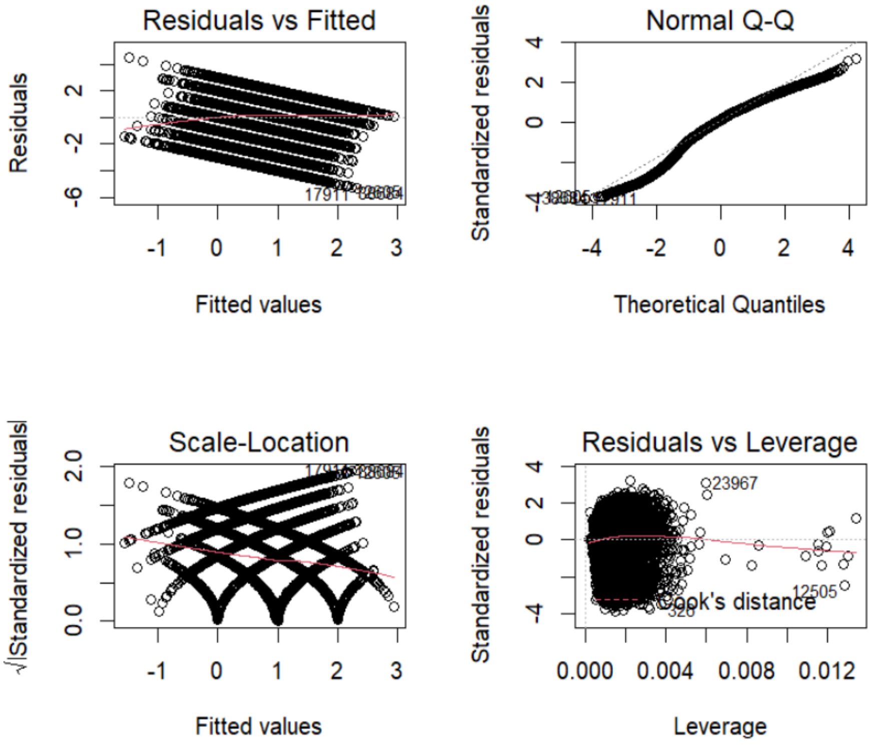


Figure 2.2.5 Linear Model of c19ProSo01 for Other Countries

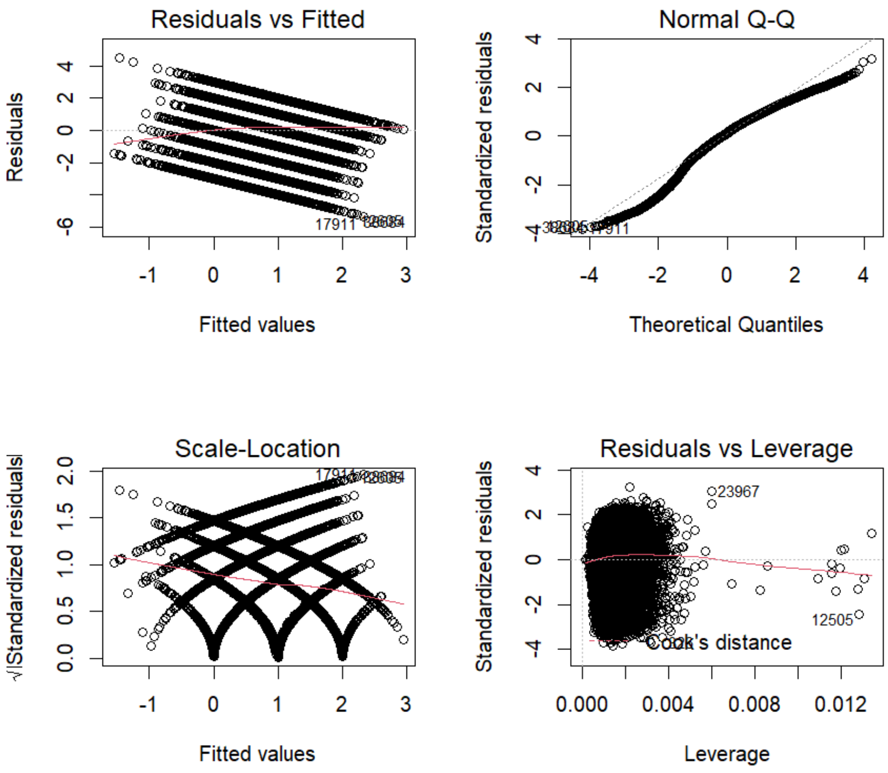


Figure 2.2.6 Linear Model of c19ProSo02 for Other Countries

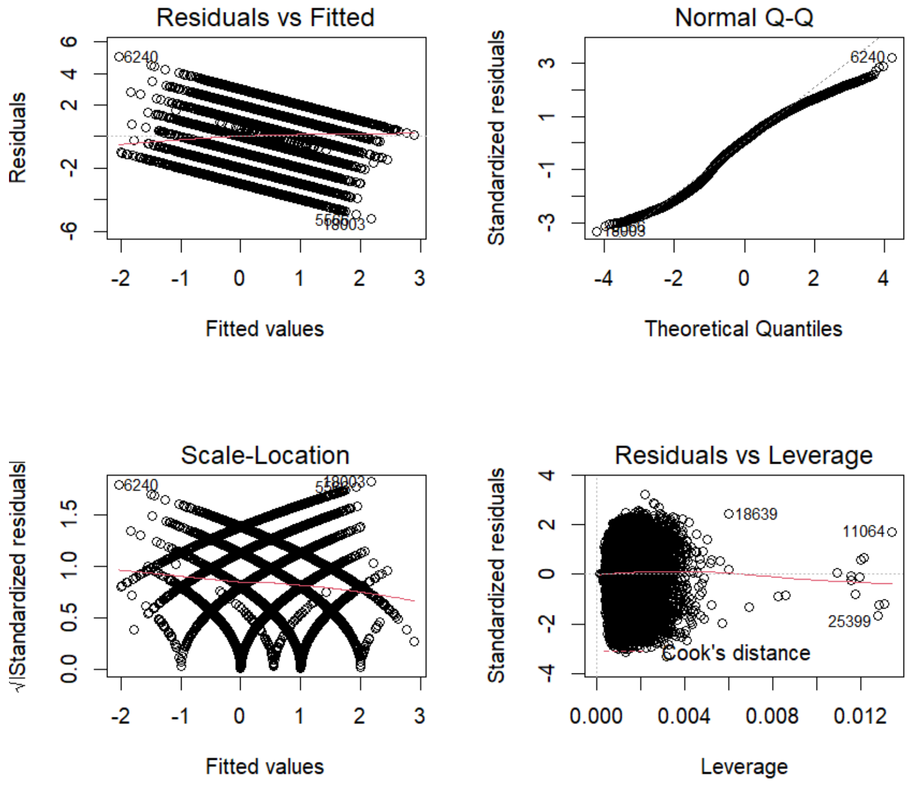


Figure 2.2.7 Linear Model of c19ProSo03 for Other Countries

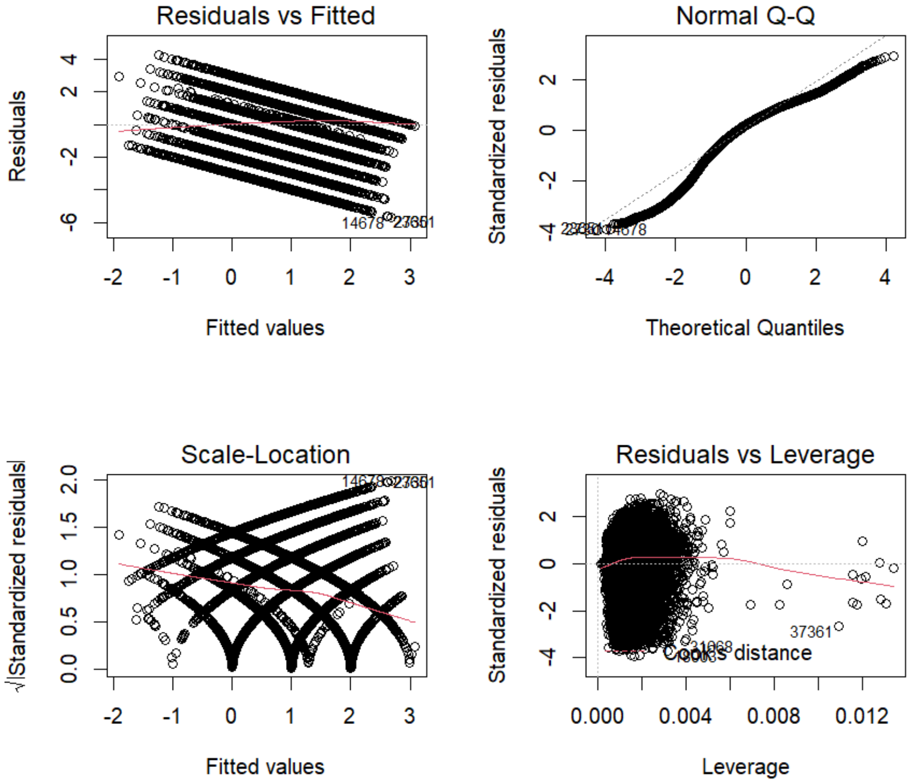


Figure 2.2.8 Linear Model of c19ProSo04 for Other Countries

As for the plotting of four linear model individually corresponding to the pro-social behaviour has the same explanation from the one in Germany.

Q3(a) similar countries in cluster 3

```
> #k-mean clustering to to cluster similiar contries
> library(cluster)
> covid_p = covid[,1:(ncol(others)-5)]
> #fit to linear model
> #Corona ProSocial Behavior 1 with its predictor
> PS1_fit = lm(formula = covid$c19ProSo01~., data = covid_p)
> pvalues_1 <- summary(PS1_fit)$coefficients[, 4]
> min_p(pvalues_1)
[1] "c19NormShould"
> #Corona ProSocial Behavioure 2 with its predictor
> PS2_fit = lm(formula = covid$c19ProSo02~., data = covid_p)
> pvalues_2 <- summary(PS2_fit)$coefficients[, 4]
> min_p(pvalues_2)
[1] "c19NormShould"
> #Corona ProSocial Behavioure 3 with its predictor
> PS3_fit = lm(formula = covid$c19ProSo03~., data = covid_p)
> pvalues_3 <- summary(PS3_fit)$coefficients[, 4]
> min_p(pvalues_3)
[1] "c19NormShould"
> #Corona ProSocial Behavioure 4 with its predictor
> PS4_fit = lm(formula = covid$c19ProSo04~., data = covid_p)
> pvalues_4 <- summary(PS4_fit)$coefficients[, 4]
> min_p(pvalues_4)
[1] "c19NormShould"
> #the predictors that have p-values less than 0.05(very important predictors)
> #in all four fitted linear models
> pvalues_1[pvalues_1 < 0.001 & pvalues_2 < 0.001 & pvalues_3 < 0.001 & pvalues
_4 < 0.001]
   (Intercept)        affInsp         disc02      jbInsec02 employstatus_10
  1.367903e-66   3.911728e-10   4.803830e-28   2.530479e-04    1.575069e-16
        fail01        lifeSat  c19NormShould      c19NormDo        c19IsOrg
  2.537100e-05   1.901627e-10   1.708285e-68   5.614873e-28    4.871623e-17
 trustGovState            edu
  1.689586e-54   4.975676e-21
> # Create imp by selecting most important predictors (pvalue < 0.001) in covid datas
et
> imp <- covid %>%
+   select(coded_country, affInsp, disc02, jbInsec02, employstatus_10,
+          fail01, fail02, lifeSat, c19NormShould,
+          c19NormDo, c19IsOrg, trustGovState, edu,
+          c19ProSo01, c19ProSo02, c19ProSo03, c19ProSo04)
> #aggreagte by countries
> #median_by_group <- aggregate(x ~ group, data = mydata, FUN = median)
> csmall = aggregate(imp[,2:ncol(imp)],list(imp$coded_country),mean)
> colnames(csmall)[1] = "coded_country"
> #scaling to make all indicators have equal weight in the clustering algorithm
> csmall[2:ncol(csmall)]<-scale(csmall[2:ncol(csmall)])
> #choose optimal number of clusters (k) with average silhouette score
> i_silhouette_score <- function(k) {
+                        #start from 2 to avoid coded_country
+   km <- kmeans(csmall[,2:ncol(csmall)], k, nstart = 50)#start from 50 cluster centro
ds
+   #more starts to make clustering more stable
+   ss <- silhouette(km$cluster, dist(csmall[,2:ncol(csmall)]))
+   mean(ss[,3]) #mean of the third column of the silhouette scores
+   #calculates the average silhouette width for all observations in the data
+   #
+   #R returns a matrix with three columns. The first column contains the cluster
+   #assignments for each observation, the second column contains the neighbor
+   #cluster (the second-best cluster assignment for each observation), and the
+   #third column contains the silhouette width for each observation.
+ }
> k <- 2:20  #creates a vector k containing the values from 2 to 20
> #sapply function to apply the i_silhouette_score function to each value of k
> avg_sil <- sapply(k, i_silhouette_score)
> #retrieve k(number of clusters) that has highest average silhouette score
> k[which.max(avg_sil)]
[1] 4
> par(mfrow=c(1,1))
> #create a line plot of the average silhouette scores against number of clusters
> plot(k, type ='b', avg_sil, xlab='Number of clusters', ylab = 'Average Silhouette Scor
e')
> #Add text of number of clustet to every point
> text(k, avg_sil, labels=k, cex=0.8)
> title("Average Silhouette Score against Numebr of clusters (k)")
> set.seed(31240291)
> #fit with kmeans clustering with k = 4, number of centroids start from 20
> zkfit = kmeans(csmall[2:ncol(csmall)], centers = 4, nstart = 20)
> # Add cluster assignments to data frame
> csmall$cluster <- zkfit$cluster
> # Move column coded_country to the first column
> csmall <- cbind(csmall$cluster, csmall[,setdiff(names(csmall), "cluster")])
> #find out Germany in which cluster
> csmall %>% filter(coded_country == "Germany") #cluster 3
  csmall$cluster coded_country   affInsp     disc02 jbInsec02 employstatus_10     fail0
1
1              3       Germany 0.1118166 -0.8131773 0.1599388      -0.2312222 -0.299266
2
      fail02    lifeSat c19NormShould  c19NormDo  c19IsOrg trustGovState        edu
1 -0.1790892 -0.1049388     -1.031809 -0.3342908 0.1022186     0.5068359 -0.89259
47
    c19ProSo01 c19ProSo02 c19ProSo03 c19ProSo04
1 -0.002892592   -1.14121 -0.2381856 -0.1781413
> #find out similar countries in the same cluster
> sim_3 = csmall %>% filter(`csmall$cluster`==3)
```

```
   (Intercept)        affInsp         disc02       jbInsec02 employstatus_10
  1.367903e-66    3.911728e-10    4.803830e-28    2.530479e-04    1.575069e-16
        fail01         lifeSat   c19NormShould       c19NormDo        c19IsOrg
  2.537100e-05    1.901627e-10    1.708285e-68    5.614873e-28    4.871623e-17
trustGovState             edu
  1.689586e-54    4.975676e-21
```

Figure 3.1.1 Table of Predictors for Cluster

Since these are the predictors that have p-value < 0.001 in covid, which mean they are very important for all the countries to cluster each other together base on those predictors.

Also, since clustering would treat every single row as an identical country, the aggregate method is used to calculate their corresponding mean for every country. In total, there are 111 identical countries, hence data of 111 rows is produced.

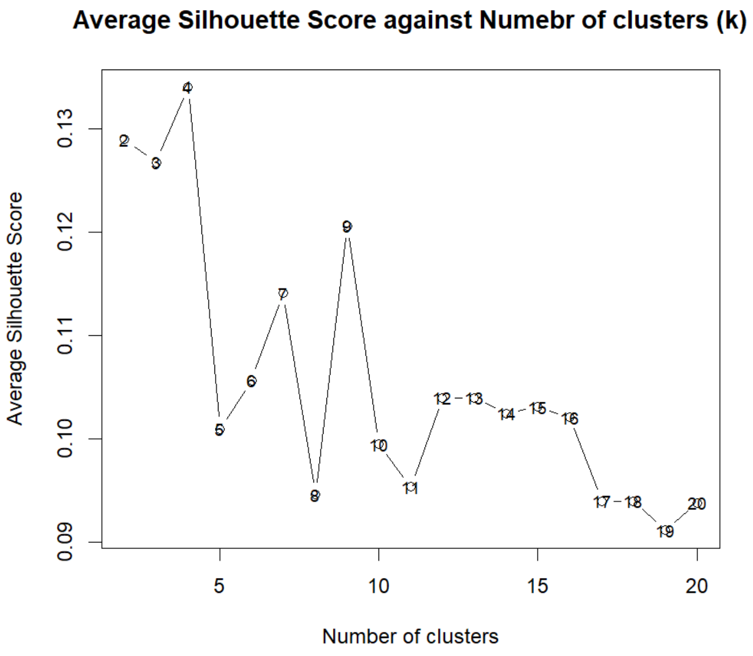


Figure 3.1.2 Graph of Average Silhouette Score against Number of clusters (k)

Before clustering, the silhouette score is used to determine the suitable number of cluster to cluster those countries. Based the Figure 3.1.2, the most suitable number of cluster is 4.

The predictors except the countries are needed to be scaled to avoid one predictor dominate the model as bias the model towards the predictor that has higher scale

Afterward, the kmeans clustering is used to cluster the countries and assign the list of clusters back to the dataset to filter out the similar countries in cluster 3 since Germany is in cluster 3. Alongside with Germany, USA, Spain and Greece are the similar countries in the same cluster

Q3(b)

```
> #table can have frequncies for unique values
>                  #as.character to perserve character type otherwise
> country_count = table(as.character(covid$coded_country)) #changed to factor type
> #change to dataframe to have separate column of countries and frequencies
> cc_frame = as.data.frame(country_count)
> #change back the type to character from factor for the following filtering
> cc_frame$Var1 <- as.character(cc_frame$Var1)
> str(cc_frame)
'data.frame':          111 obs. of  2 variables:
 $ Var1: chr  "" "Albania" "Algeria" "Andorra" ...
 $ Freq: int  157 5 129 1 907 762 28 2 8 98 ...
> # subset dataframe to include only rows where Var1 has those countries in sim_3
> sim_cc <- filter(cc_frame, Var1 %in% sim_3$coded_country)
> sim_cc <- sim_cc[order(sim_cc$Freq, decreasing = TRUE), ]
```

Since table can automatically get the observation for each unique countries, table function is used and as.character() to preserve the character type of coded_country and change back to dataframe to separate into column of countries and column of frequencies. Retrieve the similar countries by checking whether the countries are in the cluster 3 with %in% function, and eventually order them by number observations in descending order. Choosing the top 3 number of observations countries can have more precise estimates of the coefficients and as much more data points to fit into linear model, otherwise might result in overfit as number of predictor variables are more than the observations.

Therefore, USA, Spain and Greece are chosen to be 3 similar countries and used to repeat similar procedures for finding the important predictors.

USA:

```
> #choose USA, Spain and Greece, highest observations in similar countries
> usa = covid %>% filter(coded_country == "United States of America")
> usa_p = usa[,1:(ncol(usa)-5)]
> #fit to linear model
> #Corona ProSocial Behavior 1 with its predictor
> PS1_fit_g = lm(formula = usa$c19ProSo01~., data = usa_p)
> pvalues_1 <- summary(PS1_fit_g)$coefficients[, 4]
> pvalues_1[pvalues_1 < 0.05]#treat it as list
   (Intercept)        affDepr       affEnerg         affExh        affInsp 
  1.415493e-27   3.805229e-02   2.367087e-03   1.942134e-02   4.538286e-04 
       PLRAC19        PLRAEco         disc02       jbInsec04  employstatus_3 
  6.798538e-13   4.325791e-03   1.321155e-08   1.430756e-02   2.325894e-03 
 employstatus_4  employstatus_9 employstatus_10         fail02         fail03 
  1.688514e-02   5.526005e-07   1.562698e-02   5.191264e-05   2.600761e-02 
         happy            MLQ   c19NormShould        c19IsOrg  trustGovState 
  1.774000e-04   2.799756e-02   5.237715e-20   8.237432e-04   1.215129e-14 
```

```
          edu
  7.380564e-07
> min_p(pvalues_1)
[1] "c19NormShould"
> #Corona ProSocial Behavioure 2 with its predictor
> PS2_fit_g = lm(formula = usa$c19ProSo02~., data = usa_p)
> pvalues_2 <- summary(PS2_fit_g)$coefficients[, 4]
> pvalues_2[pvalues_2 < 0.05]#treat it as list
   (Intercept)       affEnerg        affNerv         affExh        affInsp
  2.763652e-43   5.434225e-03   3.012165e-04   1.024385e-02   2.850656e-06
      PLRAC19         disc02         disc03      jbInsec02  employstatus_3
  3.590205e-03   1.029179e-09   2.394706e-02   1.601566e-02   1.096785e-02
 employstatus_5  employstatus_8  employstatus_9 employstatus_10          PFS01
  9.936549e-03   3.938316e-02   4.069052e-02   1.553553e-02   3.610944e-09
        PFS02        lifeSat   c19NormShould     c19IsPunish       c19IsOrg
  1.024181e-04   1.235761e-06   2.299928e-33   1.205987e-05   3.209242e-02
  trustGovState         gender            age            edu
  4.379942e-19   2.541781e-02   5.200107e-03   4.449322e-10
> min_p(pvalues_2)
[1] "c19NormShould"
> #Corona ProSocial Behavioure 3 with its predictor
> PS3_fit_g = lm(formula = usa$c19ProSo03~., data = usa_p)
> pvalues_3 <- summary(PS3_fit_g)$coefficients[, 4]
> pvalues_3[pvalues_3 < 0.05]#treat it as list
   (Intercept)        affDepr         affExh        affInsp        PLRAC19
  2.929554e-18   1.166236e-02   7.898945e-03   6.989095e-05   6.541094e-15
        disc02         disc03  employstatus_3  employstatus_9 employstatus_10
  6.935386e-06   1.089468e-02   1.004960e-02   3.383730e-03   4.849620e-03
        PFS02         fail02         fail03          happy   c19NormShould
  4.444808e-02   1.022597e-02   4.264985e-02   3.276103e-03   1.372166e-24
      c19IsOrg   trustGovState            age            edu
  2.599533e-03   9.381408e-09   9.540028e-05   3.706055e-09
> min_p(pvalues_3)
[1] "c19NormShould"
> #Corona ProSocial Behavioure 4 with its predictor
> PS4_fit_g = lm(formula = usa$c19ProSo04~., data = usa_p)
> pvalues_4 <- summary(PS4_fit_g)$coefficients[, 4]
> pvalues_4[pvalues_4 < 0.05]#treat it as list
  (Intercept)      affEnerg        affExh       PLRAC19        disc01        disc02
 1.464857e-04   4.800562e-02   7.523156e-03   1.633975e-11   3.027652e-02   7.506
312e-14
employstatus_4         PFS02        fail02        fail03         happy  c19NormShould
 8.650066e-03   2.496333e-03   2.385371e-02   2.357759e-06   1.877402e-05  2.8289
87e-142
  c19IsStrict    c19IsPunish       c19IsOrg  trustGovState        gender           age
 4.993136e-03   1.418391e-09   5.498785e-03   1.093482e-08   9.983222e-03   4.895
026e-03
> min_p(pvalues_4)
[1] "c19NormShould"
> #the predictors that have p-values less than 0.05
> #in all four fitted linear models
> pvalues_1[pvalues_1 < 0.05 & pvalues_2 < 0.05 & pvalues_3 < 0.05 & pvalues_4 <
 0.05]
  (Intercept)       affExh       PLRAC19       disc02 c19NormShould      c19IsOrg
 1.415493e-27  1.942134e-02  6.798538e-13  1.321155e-08  5.237715e-20  8.237432e
-04
trustGovState
 1.215129e-14
```

Spain:

```
> #Spain
> spain = covid %>% filter(coded_country == "Spain")
> spain_p = spain[,1:(ncol(usa)-5)]
> #fit to linear model
> #Corona ProSocial Behavior 1 with its predictor
> PS1_fit_g = lm(formula = spain$c19ProSo01~., data = spain_p)
> pvalues_1 <- summary(PS1_fit_g)$coefficients[, 4]
> pvalues_1[pvalues_1 < 0.05]#treat it as list
   (Intercept)        affExh       PLRAC19       PLRAEco      jbInsec02      jbInsec04
  2.464105e-02   2.616944e-02   8.193299e-05   8.475973e-03   1.589234e-02   3.184
975e-02
employstatus_6 employstatus_9         PFS01         PFS03           MLQ       c19IsOrg
  1.027996e-03   4.421088e-02   3.822661e-02   1.332339e-02   2.891904e-02   2.294
387e-02
 trustGovState
  5.322124e-03
> min_p(pvalues_1)
[1] "PLRAC19"
> #Corona ProSocial Behavioure 2 with its predictor
> PS2_fit_g = lm(formula = spain$c19ProSo02~., data = spain_p)
> pvalues_2 <- summary(PS2_fit_g)$coefficients[, 4]
> pvalues_2[pvalues_2 < 0.05]#treat it as list
   (Intercept)        affExc       PLRAC19        disc02 employstatus_9         PFS02
  5.285668e-07   3.065203e-02   1.153012e-03   1.799042e-02   6.339476e-03   1.325
079e-02
        fail01        lifeSat  trustGovState           edu
  4.188415e-03   1.365616e-06   9.724497e-03   1.052850e-02
> min_p(pvalues_2)
[1] "lifeSat"
> #Corona ProSocial Behavioure 3 with its predictor
> PS3_fit_g = lm(formula = spain$c19ProSo03~., data = spain_p)
> pvalues_3 <- summary(PS3_fit_g)$coefficients[, 4]
> pvalues_3[pvalues_3 < 0.05]#treat it as list
    (Intercept)        affDepr         affExh        affInsp        PLRAC19
   4.465425e-05    8.653964e-04    1.388204e-02    3.680922e-04    1.069310e-07
employstatus_10         lifeSat   trustGovState            edu
   7.780439e-04    4.954978e-02    7.350710e-04    3.442699e-02
```

```
> min_p(pvalues_3)
[1] "PLRAC19"
> #Corona ProSocial Behavioure 4 with its predictor
> PS4_fit_g = lm(formula = spain$c19ProSo04~., data = spain_p)
> pvalues_4 <- summary(PS4_fit_g)$coefficients[, 4]
> pvalues_4[pvalues_4 < 0.05]#treat it as list
      affCalm       affEnerg       PLRAC19        disc02 employstatus_1         PFS01
  2.839595e-02   2.421763e-03   9.332247e-07   4.006242e-02   2.413187e-02   1.298
280e-03
        PFS03         fail03  c19NormShould      c19NormDo       c19IsOrg   trustGovCtr
y
  5.902261e-03   5.261647e-03   2.716305e-03   2.998140e-04   4.141951e-02   1.068
496e-02
       gender
  4.952921e-02
> min_p(pvalues_4)
[1] "PLRAC19"
> #the predictors that have p-values less than 0.001(very important predictors)
> #in all four fitted linear models
> pvalues_1[pvalues_1 < 0.05 & pvalues_2 < 0.05 & pvalues_3 < 0.05 & pvalues_4 <
 0.05]
     PLRAC19
8.193299e-05
```

PLRAC19

Greece:

```
> #Greece
> greece = covid %>% filter(coded_country == "Greece")
> greece_p = greece[,1:(ncol(greece)-5)]
> #fit to linear model
> #Corona ProSocial Behavior 1 with its predictor
> PS1_fit_g = lm(formula = greece$c19ProSo01~., data = greece_p)
> pvalues_1 <- summary(PS1_fit_g)$coefficients[, 4]
> pvalues_1[pvalues_1 < 0.05]#treat it as list
   (Intercept)        affDepr employstatus_3 employstatus_4 employstatus_9         fail03
  1.714339e-04   1.631157e-02   5.195599e-04   1.700113e-02   6.418791e-03   1.604
542e-03
        happy   trustGovCtry  trustGovState            edu
  2.543496e-03   1.107094e-05   9.287791e-07   3.438260e-02
> min_p(pvalues_1)
[1] "trustGovState"
> #Corona ProSocial Behavioure 2 with its predictor
> PS2_fit_g = lm(formula = greece$c19ProSo02~., data = greece_p)
> pvalues_2 <- summary(PS2_fit_g)$coefficients[, 4]
> pvalues_2[pvalues_2 < 0.05]#treat it as list
  (Intercept)        affAnx       affInsp         PFS02       c19IsOrg trustGovState
```

```
 7.879302e-05  3.147238e-03  4.063698e-02  1.565154e-02  2.707186e-02  5.386760e
-05
         edu
 2.571570e-02
> min_p(pvalues_2)
[1] "trustGovState"
> #Corona ProSocial Behavioure 3 with its predictor
> PS3_fit_g = lm(formula = greece$c19ProSo03~., data = greece_p)
> pvalues_3 <- summary(PS3_fit_g)$coefficients[, 4]
> pvalues_3[pvalues_3 < 0.05]#treat it as list
   (Intercept)        affDepr         affExc  employstatus_1  employstatus_3
   0.005902301     0.021922853     0.034448277     0.017373145     0.004677116
 employstatus_5  employstatus_9 employstatus_10          PFS01        c19IsOrg
   0.019397929     0.004084845     0.002820166     0.044584096     0.020341432
           age
   0.025475294
> min_p(pvalues_3)
[1] "employstatus_10"
> #Corona ProSocial Behavioure 4 with its predictor
> PS4_fit_g = lm(formula = greece$c19ProSo04~., data = greece_p)
> pvalues_4 <- summary(PS4_fit_g)$coefficients[, 4]
> pvalues_4[pvalues_4 < 0.05]#treat it as list
  (Intercept) employstatus_9        fail02        fail03  c19NormShould    c19IsStrict
  3.607187e-04   2.386055e-02   4.193735e-02   3.834298e-03   1.865130e-23   6.003
694e-03
          edu
  7.577489e-03
> min_p(pvalues_4)
[1] "c19NormShould"
> #the predictors that have p-values less than 0.001(very important predictors)
> #in all four fitted linear models
> pvalues_1[pvalues_1 < 0.05 & pvalues_2 < 0.05 & pvalues_3 < 0.05 & pvalues_4 <
 0.05]
 (Intercept)
0.0001714339
```

USA: affExh, PLRAC19, disc02, c19NormShould (Strongest Predictor), c19IsOrg, trustGovState

Spain: PLRAC19 (Strongest Predictor)

Greece: trustGovState (Strongest Predictor)

In comparison between 2(c) and 3(b) for measuring who better match the important attributes for predicting pro-social attitudes, since trustGovState and edu are the best predictors for Germany which also play important roles in other countries's data, whereas the strongest predictor is PLRAC19 as which is an important predictor in USA and Spain but not Greece which is not an important predictor for Germany, the other countries from 2(c) actually have a better match with the Germany from 2(b). However, the strongest predictor from other countries is c19NormShould which is not an important predictor in Germany, and the PLRAC19 from USA and Spain is an important attribute for predicting 3 out of 4 response in Germany.