PROJECT: Analysis of country level predictors of pro-social behaviors to reduce the spread of COVID-19 during the early stages of the pandemic.

BY: Lim Yu Jin
tool used: R studio

```
# install necessary library packages
install.packages("magrittr")
install.packages("dplyr")
install.packages("ggplot2")
install.packages('tidyverse')
install.packages('caret')

library(magrittr)
library(dplyr)
library(ggplot2)
library(tibble)
library(tidyverse)
library(caret)
library(fpc)

# extract the dataset
rm(list = ls())
set.seed(32637888) # student id: 32637888
psy = read.csv("PsyCoronaBaselineExtract.csv", header=TRUE)
psy <- psy[sample(nrow(psy), 40000), ] # 40000 rows

# Question 1.a
########################################################################################
# get a peek of the first 6 lines
head(psy)
```

```
      affAnx affBor affCalm affContent affDepr affEnerg affExc affNerv affExh affInsp affRel PLRAC19 PLRAEco disc01 disc02 disc03 jbInsec01 jbInsec02 jbInsec03
48406      3      4       3          4       4        2      2       2      4       2      3       4       3      1      1      0        -2         2        -1
23747      5      3       1          1       3        1      1       4      2       1      1       4       5      2      2      0         1        -2         2
52572      1      3       4          4       1        3      3       1      1       4      4       4       3     -1     -1      0        -2         2        -1
59046      4      4       2          2       3        2      1       3      4       1      2       2       6      1      0     -1        -1         1         0
45855      1      5       1          2       3        1      2       4      4       3      3       2       1      0      2     -1        -2         0         1
143        2      4       3          4       1        3      2       2      2       1      3       4       2      1      1      0        NA        NA        NA
      jbInsec04 employstatus_1 employstatus_2 employstatus_3 employstatus_4 employstatus_5 employstatus_6 employstatus_7 employstatus_8 employstatus_9
48406        -2             NA             NA              1             NA             NA             NA             NA             NA             NA
23747        -1              1             NA             NA             NA             NA             NA             NA             NA             NA
52572        -2             NA             NA              1             NA             NA             NA             NA             NA             NA
59046        -2             NA             NA              1             NA             NA             NA             NA             NA             NA
45855        -2             NA              1             NA             NA             NA             NA             NA             NA             NA
143          NA             NA             NA             NA             NA              1             NA             NA             NA             NA
      employstatus_10 PFS01 PFS02 PFS03 fail01 fail02 fail03 happy lifeSat MLQ c19NormShould c19NormDo c19IsStrict c19IsPunish c19IsOrg trustGovCtry
48406              NA    -1     1    -2      0     -1     -2     5       5  -3             2        -1           4           3        3            3
23747              NA     2     2     1      1     -1      2     1       2   2             2        -2           4           2        4            3
52572              NA    -1    -1    -1     -1     -1     -1     3       4   3             1         0           4           3        4            3
59046              NA     2     2     2      1      0      1     5       3   0             0        -3           1           1        1            3
45855              NA     1     1     2      1     -2      0     7       3  -1             3         3           4           3        5            4
143                NA    -1     1    -1     -1     -1     -1     9       6   1             2         2           4           4        4            3
      trustGovState gender age edu       coded_country c19ProSo01 c19ProSo02 c19ProSo03 c19ProSo04
48406             3      1   2   6              Poland          1          1          3          1
23747             2      1   2   4 Republic of Serbia          -2         -2         -2          3
52572             3      2   2   6             Romania          1          1          1          1
59046             1      2   3   2            Thailand          1         -3         -3         -3
45855             4      2   2   6         South Korea         -1         -2         -3         -1
143               3      1   1   4               Egypt          0          2          1          0
```

```
# dimension of data set : 40000 rows, 54 columns
dim(psy)
```

```
> dim(psy)
[1] 40000     54
```

```
# see the range of values and distribution of all attributes
summary(psy)
```

```
> summary(psy)
     affAnx         affBor         affCalm        affContent      affDepr         affEnerg        affExc          affNerv          affExh         affInsp
 Min.   :1.00   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.0   Min.   :1.000
 1st Qu.:2.00   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.000   1st Qu.:2.000   1st Qu.:1.0   1st Qu.:1.000
 Median :3.00   Median :3.000   Median :3.000   Median :3.000   Median :2.000   Median :3.000   Median :2.000   Median :2.000   Median :2.0   Median :2.000
 Mean   :2.72   Mean   :2.714   Mean   :2.931   Mean   :2.683   Mean   :2.237   Mean   :2.575   Mean   :2.153   Mean   :2.586   Mean   :2.5   Mean   :2.439
 3rd Qu.:4.00   3rd Qu.:4.000   3rd Qu.:4.000   3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:3.000   3rd Qu.:4.000   3rd Qu.:3.0   3rd Qu.:3.000
 Max.   :5.00   Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.000   Max.   :5.0   Max.   :5.000
 NA's   :524    NA's   :544     NA's   :529     NA's   :621     NA's   :604     NA's   :658     NA's   :703     NA's   :558     NA's   :627   NA's   :668
     affRel         PLRAC19        PLRAEco          disc01            disc02            disc03          jbInsec01         jbInsec02         jbInsec03
 Min.   :1.000   Min.   :1.00   Min.   :1.000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.0000   Min.   :-2.000   Min.   :-2.000   Min.   :-2.000
 1st Qu.:2.000   1st Qu.:3.00   1st Qu.:3.000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.:-1.0000   1st Qu.:-2.000   1st Qu.: 0.000   1st Qu.:-1.000
 Median :3.000   Median :4.00   Median :4.000   Median : 1.0000   Median : 1.0000   Median : 0.0000   Median :-1.000   Median : 1.000   Median : 0.000
 Mean   :2.739   Mean   :3.56   Mean   :4.396   Mean   : 0.6388   Mean   : 0.8379   Mean   :-0.4021   Mean   :-0.594   Mean   : 0.563   Mean   : 0.064
 3rd Qu.:4.000   3rd Qu.:4.00   3rd Qu.:6.000   3rd Qu.: 1.0000   3rd Qu.: 1.0000   3rd Qu.: 0.0000   3rd Qu.: 0.000   3rd Qu.: 1.000   3rd Qu.: 1.000
 Max.   :5.000   Max.   :8.00   Max.   :8.000   Max.   : 2.0000   Max.   : 2.0000   Max.   : 2.0000   Max.   : 2.000   Max.   : 2.000   Max.   : 2.000
 NA's   :620     NA's   :151    NA's   :162     NA's   :148       NA's   :145       NA's   :151       NA's   :11045    NA's   :9945     NA's   :8533
   jbInsec04       employstatus_1  employstatus_2  employstatus_3  employstatus_4  employstatus_5  employstatus_6  employstatus_7  employstatus_8
 Min.   :-2.000   Min.   :1       Min.   :1       Min.   :1       Min.   :1       Min.   :1       Min.   :1       Min.   :1       Min.   :1
 1st Qu.:-2.000   1st Qu.:1       1st Qu.:1       1st Qu.:1       1st Qu.:1       1st Qu.:1       1st Qu.:1       1st Qu.:1       1st Qu.:1
 Median :-2.000   Median :1       Median :1       Median :1       Median :1       Median :1       Median :1       Median :1       Median :1
 Mean   :-0.986   Mean   :1       Mean   :1       Mean   :1       Mean   :1       Mean   :1       Mean   :1       Mean   :1       Mean   :1
 3rd Qu.: 0.000   3rd Qu.:1       3rd Qu.:1       3rd Qu.:1       3rd Qu.:1       3rd Qu.:1       3rd Qu.:1       3rd Qu.:1       3rd Qu.:1
 Max.   : 2.000   Max.   :1       Max.   :1       Max.   :1       Max.   :1       Max.   :1       Max.   :1       Max.   :1       Max.   :1
 NA's   :13127    NA's   :34349   NA's   :33255   NA's   :29145   NA's   :36545   NA's   :37942   NA's   :36922   NA's   :36342   NA's   :39264
 employstatus_9  employstatus_10     PFS01              PFS02             PFS03              fail01             fail02            fail03              happy
 Min.   :1       Min.   :1       Min.   :-2.00000   Min.   :-2.000   Min.   :-2.0000   Min.   :-2.00000   Min.   :-2.0000   Min.   :-2.0000   Min.   : 1.000
 1st Qu.:1       1st Qu.:1       1st Qu.:-1.00000   1st Qu.: 0.000   1st Qu.:-1.0000   1st Qu.:-1.00000   1st Qu.:-1.0000   1st Qu.: 0.0000   1st Qu.: 5.000
 Median :1       Median :1       Median : 0.00000   Median : 1.000   Median : 0.0000   Median : 0.00000   Median :-1.0000   Median : 1.0000   Median : 7.000
 Mean   :1       Mean   :1       Mean   :-0.02574   Mean   : 0.573   Mean   :-0.2545   Mean   :-0.06268   Mean   :-0.4104   Mean   : 0.3522   Mean   : 6.325
 3rd Qu.:1       3rd Qu.:1       3rd Qu.: 1.00000   3rd Qu.: 1.000   3rd Qu.: 1.0000   3rd Qu.: 1.00000   3rd Qu.: 0.0000   3rd Qu.: 1.0000   3rd Qu.: 8.000
 Max.   :1       Max.   :1       Max.   : 2.00000   Max.   : 2.000   Max.   : 2.0000   Max.   : 2.00000   Max.   : 2.0000   Max.   : 2.0000   Max.   :10.000
 NA's   :31867   NA's   :39066   NA's   :173        NA's   :153      NA's   :149       NA's   :161        NA's   :158       NA's   :145       NA's   :511
    lifeSat          MLQ          c19NormShould      c19NormDo       c19IsStrict     c19IsPunish      c19IsOrg       trustGovCtry    trustGovState
 Min.   :1.00   Min.   :-3.0000   Min.   :-3.000   Min.   :-3.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000   Min.   :1.000
 1st Qu.:3.00   1st Qu.: 0.0000   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:3.000   1st Qu.:2.000   1st Qu.:2.000
 Median :4.00   Median : 1.0000   Median : 2.000   Median : 2.000   Median :4.000   Median :4.000   Median :4.000   Median :3.000   Median :3.000
 Mean   :4.14   Mean   : 0.8411   Mean   : 2.007   Mean   : 1.296   Mean   :4.123   Mean   :3.503   Mean   :3.898   Mean   :3.018   Mean   :3.081
 3rd Qu.:5.00   3rd Qu.: 2.0000   3rd Qu.: 3.000   3rd Qu.: 2.000   3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:5.000   3rd Qu.:4.000   3rd Qu.:4.000
 Max.   :6.00   Max.   : 3.0000   Max.   : 3.000   Max.   : 3.000   Max.   :6.000   Max.   :6.000   Max.   :6.000   Max.   :5.000   Max.   :5.000
 NA's   :130    NA's   :133       NA's   :153      NA's   :149      NA's   :184     NA's   :187     NA's   :173     NA's   :9444    NA's   :9523
     gender           age             edu        coded_country        c19ProSo01        c19ProSo02        c19ProSo03        c19ProSo04
 Min.   :1.000   Min.   :1.000   Min.   :1.000   Length:40000       Min.   :-3.0000   Min.   :-3.0000   Min.   :-3.0000   Min.   :-3.000
 1st Qu.:1.000   1st Qu.:2.000   1st Qu.:4.000   Class :character   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.0000   1st Qu.: 0.000
 Median :1.000   Median :3.000   Median :5.000   Mode  :character   Median : 1.0000   Median : 1.0000   Median : 1.0000   Median : 2.000
 Mean   :1.392   Mean   :2.902   Mean   :4.412                      Mean   : 0.9657   Mean   : 0.6744   Mean   : 0.5442   Mean   : 1.283
 3rd Qu.:2.000   3rd Qu.:4.000   3rd Qu.:5.000                      3rd Qu.: 2.0000   3rd Qu.: 2.0000   3rd Qu.: 2.0000   3rd Qu.: 2.000
 Max.   :3.000   Max.   :8.000   Max.   :7.000                      Max.   : 3.0000   Max.   : 3.0000   Max.   : 3.0000   Max.   : 3.000
 NA's   :217     NA's   :243     NA's   :296                        NA's   :142       NA's   :150       NA's   :158       NA's   :161
```

# only coded_country is character class,
# others in integer class
str(psy)

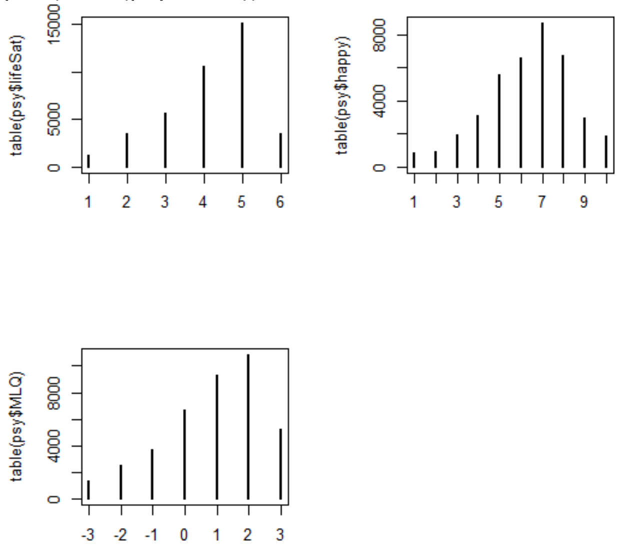
```
> str(psy)
'data.frame':   40000 obs. of  54 variables:
 $ affAnx         : int  3 5 1 4 1 2 4 2 1 2 ...
 $ affBor         : int  4 3 3 4 5 4 2 4 3 1 ...
 $ affCalm        : int  3 1 4 2 1 3 1 3 4 3 ...
 $ affContent     : int  4 1 4 2 2 4 1 3 1 3 ...
 $ affDepr        : int  4 3 1 3 3 1 2 2 3 1 ...
 $ affEnerg       : int  2 1 3 2 1 3 1 3 4 3 ...
 $ affExc         : int  2 1 3 1 2 2 2 3 3 2 ...
 $ affNerv        : int  2 4 1 3 4 2 4 1 1 2 ...
 $ affExh         : int  4 2 1 4 4 2 4 2 2 1 ...
 $ affInsp        : int  2 1 4 1 3 1 2 4 3 3 ...
 $ affRel         : int  3 1 4 2 3 3 1 3 3 3 ...
 $ PLRAC19        : int  4 4 4 2 2 4 6 2 3 4 ...
 $ PLRAEco        : int  3 5 3 6 1 2 7 5 4 5 ...
 $ disc01         : int  1 2 -1 1 0 1 1 2 1 1 ...
 $ disc02         : int  1 2 -1 0 2 1 1 1 0 1 ...
 $ disc03         : int  0 0 0 -1 -1 0 -1 1 -1 0 ...
 $ jbInsec01      : int  -2 1 -2 -1 -2 NA NA 2 0 0 ...
 $ jbInsec02      : int  2 -2 2 1 0 NA NA NA 0 0 ...
 $ jbInsec03      : int  -1 2 -1 0 1 NA NA -1 1 1 ...
 $ jbInsec04      : int  -2 -1 -2 -2 -2 NA 1 NA -1 -1 ...
 $ employstatus_1 : int  NA 1 NA NA NA NA NA NA NA NA ...
 $ employstatus_2 : int  NA NA NA NA 1 NA NA NA NA NA ...
 $ employstatus_3 : int  1 NA 1 1 NA NA NA NA 1 1 ...
 $ employstatus_4 : int  NA NA NA NA NA NA 1 NA NA NA ...
 $ employstatus_5 : int  NA NA NA NA NA 1 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 1 NA NA ...
 $ employstatus_9 : int  NA NA NA NA NA NA 1 NA NA NA ...
 $ employstatus_10: int  NA NA NA NA NA NA NA NA NA NA ...
 $ PFS01          : int  -1 2 -1 2 1 -1 1 1 0 0 ...
 $ PFS02          : int  1 2 -1 2 1 1 2 1 1 1 ...
 $ PFS03          : int  -2 1 -1 2 2 -1 1 1 1 -1 ...
 $ fail01         : int  0 1 -1 1 1 -1 1 1 0 -1 ...
 $ fail02         : int  -1 -1 -1 0 -2 -1 0 1 0 -1 ...
 $ fail03         : int  -2 2 -1 1 0 -1 1 1 1 0 ...
 $ happy          : int  5 1 3 5 7 9 7 10 6 8 ...
 $ lifeSat        : int  5 2 4 3 3 6 4 5 4 5 ...
 $ MLQ            : int  -3 2 3 0 -1 1 0 2 1 1 ...
 $ c19NormShould  : int  2 2 1 0 3 2 3 2 2 3 ...
 $ c19NormDo      : int  -1 -2 0 -3 3 2 1 -2 2 2 ...
 $ c19IsStrict    : int  4 4 4 1 4 4 4 1 6 3 ...
 $ c19IsPunish    : int  3 2 3 1 3 4 1 1 5 2 ...
 $ c19IsOrg       : int  3 4 4 1 5 4 4 1 5 4 ...
 $ trustGovCtry   : int  3 3 3 3 4 3 2 1 5 NA ...
 $ trustGovState  : int  3 2 3 1 4 3 3 4 4 NA ...
 $ gender         : int  1 1 2 2 2 1 1 1 2 2 ...
 $ age            : int  2 2 2 3 2 1 2 4 4 2 ...
 $ edu            : int  6 4 6 2 6 4 5 2 5 5 ...
 $ coded_country  : chr  "Poland" "Republic of Serbia" "Romania" "Thailand" ...
 $ c19ProSo01     : int  1 -2 1 1 -1 0 3 -1 2 2 ...
 $ c19ProSo02     : int  1 -2 1 -3 -2 2 -1 -3 2 1 ...
 $ c19ProSo03     : int  3 -2 1 -3 -3 1 3 -2 1 1 ...
 $ c19ProSo04     : int  1 3 1 -3 -1 0 3 2 2 2 ...
```

# 111 countries in this dataset
n_distinct(psy$coded_country)

```
> n_distinct(psy$coded_country)
[1] 111
```

# Life Satisfaction
# (lifesat, happy, MLQ) have similar distribution, which is a right shift in normal distribution
par(mfrow=c(2,2))
plot(table(psy$lifeSat))
plot(table(psy$happy))

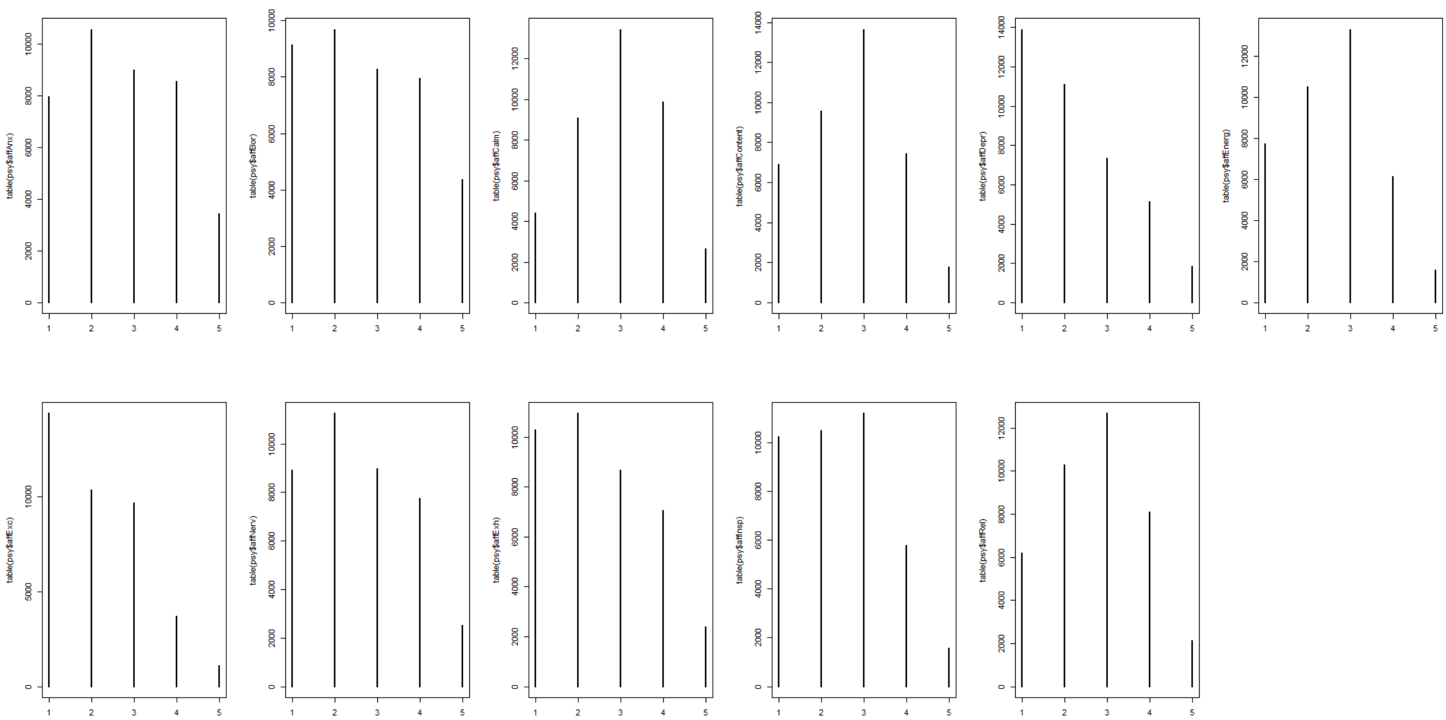
plot(table(psy$MLQ))



Affect

(affAnx, affBor, affExh, affInsp, affNerv) this 5 variables have a similar distribution, which their #median is slightly left from the middle. (affCalm, affContent, affRel, affEnerg) this 4 variables is #likely a normal distrubution. (affDepr, affExc) this 2 variables is decreasing along the x-axes, their #median is more towards the left.

par(mfrow=c(2,6))
plot(table(psy$affAnx))
plot(table(psy$affBor))
plot(table(psy$affCalm))
plot(table(psy$affContent))
plot(table(psy$affDepr))
plot(table(psy$affEnerg))
plot(table(psy$affExc))
plot(table(psy$affNerv))
plot(table(psy$affExh))
plot(table(psy$affInsp))
plot(table(psy$affRel))

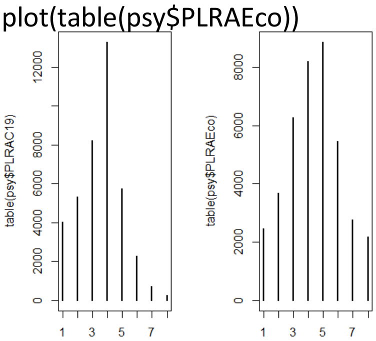


Likelihood

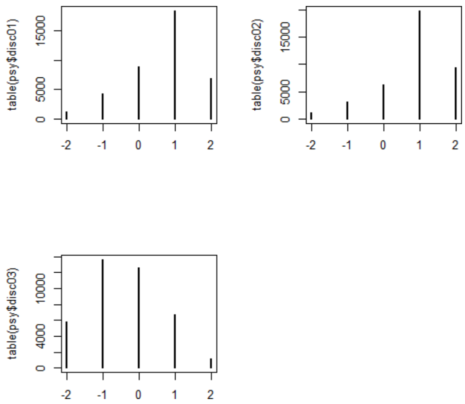
These 2 plot is likely a standard normal distribution, but the PLRAC19 have a left shift in the distribution

par(mfrow=c(1,2))
plot(table(psy$PLRAC19))

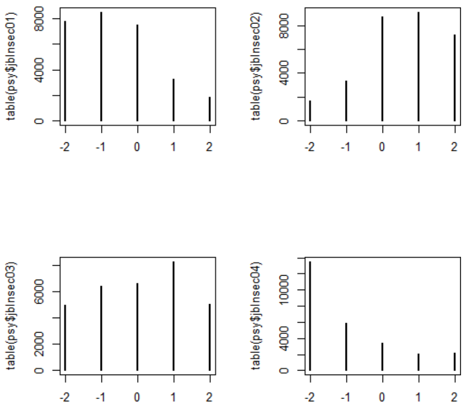
plot(table(psy$PLRAEco))



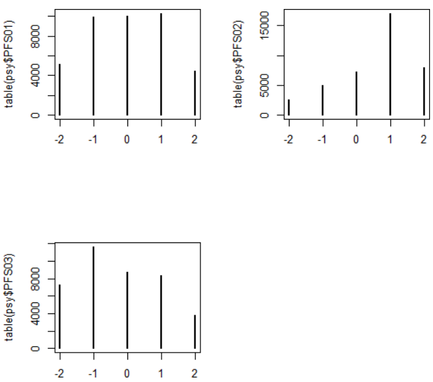
# Societal Discontent
# (disc01, disc02) have similar distribution which has a right shift normal distribution
# (disc03) have a left shift normal distribution
par(mfrow=c(2,2))
plot(table(psy$disc01))
plot(table(psy$disc02))
plot(table(psy$disc03))



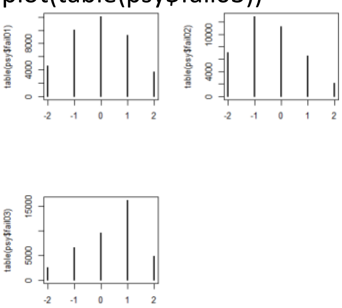
# Job Insecurity
# (jbInsec01) has a left shift normal distribution
# (jbInsec02) has a right shift in distribution
# (jbInsec03) has a almost equally distribution
# (jbInsec04) has a concave up decreasing distribution
par(mfrow=c(2,2))
plot(table(psy$jbInsec01))
plot(table(psy$jbInsec02))
plot(table(psy$jbInsec03))
plot(table(psy$jbInsec04))



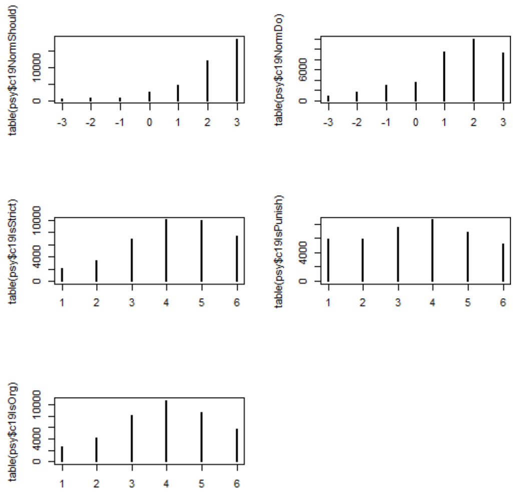
# Perceived Financial Strain
# (PFS01) has a standard normal distribution with a decrease in the center part
# (PFS02) has a right shift normal distribution
# (PFS01) has a left shift normal distribution
par(mfrow=c(2,2))
plot(table(psy$PFS01))
plot(table(psy$PFS02))
plot(table(psy$PFS03))



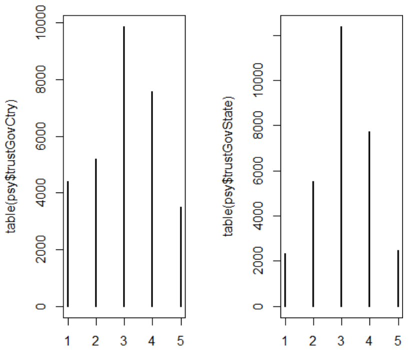
# Disempowerment
# (fail01) has a standard normal distribution
# (fail02) has a left shift normal distribution
# (fail03) has a right shift normal distribution
par(mfrow=c(2,2))
plot(table(psy$fail01))
plot(table(psy$fail02))
plot(table(psy$fail03))



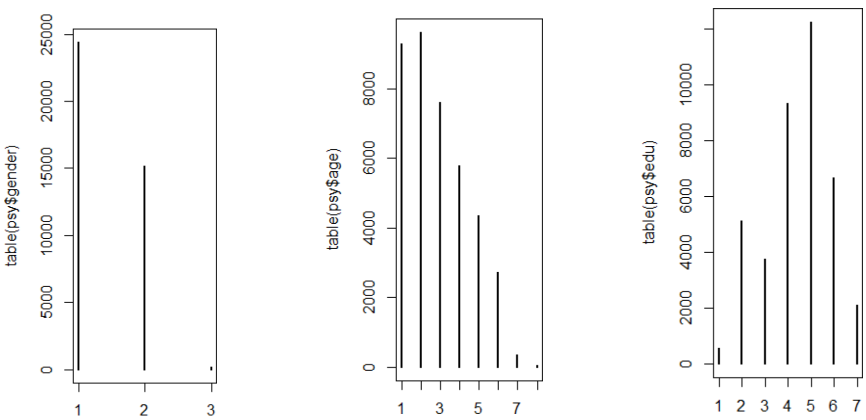
```
# Corona Community Injunctive norms
# ( c19NormShould ) has a concave upwards increasing distribution
# ( c19IsOrg, c19NormDo, c19IsStrict ) have a right shift normal distribution
# ( c19IsPunish ) has a likely equally distribution
par(mfrow=c(3,2))
plot(table(psy$c19NormShould))
plot(table(psy$c19NormDo))
plot(table(psy$c19IsStrict))
plot(table(psy$c19IsPunish))
plot(table(psy$c19IsOrg))
```



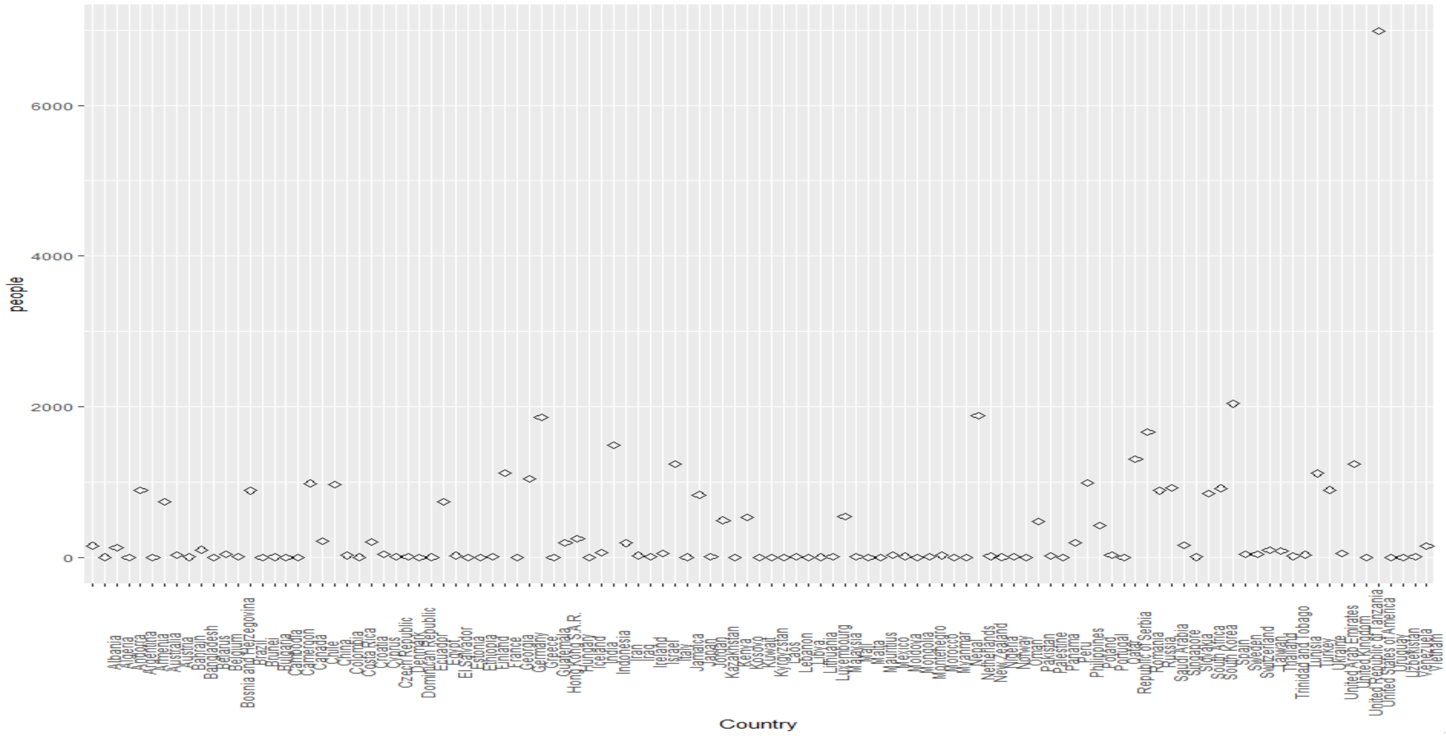

```
# Trust in Government
# ( trustGovCtry, trustGovState ) has a almost standard normal distribution
par(mfrow=c(1,2))
plot(table(psy$trustGovCtry))
plot(table(psy$trustGovState))
```



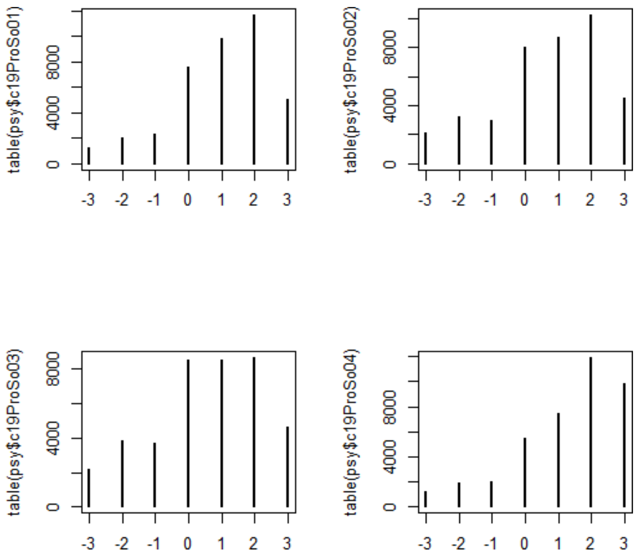

```
# Gender
# in this plot we can see that most of the participants are female,
# the number of male participants are likely half of the females'
# a little of other gender
plot(table(psy$gender))

# Age
# most of the participants are equal or below age of 54
plot(table(psy$age))

# Education
# most of the participants have a higher education
plot(table(psy$edu))
```



```
# Country Self Report
# the maximum participants come from United States of America
tbl = as.data.frame(table(psy$coded_country))
ggplot(tbl, aes(x=Var1,y=Freq))+
  geom_point(size=2, shape=23) +
  labs(x = "Country", y = "people") +
  theme(axis.text.x = element_text(angle = 90))
```




```
# Corona ProSocial Behavior
# ( c19ProSo01, c19ProSo02, c19ProSo03, c19ProSo04 ) have a right shift in normal distribution
par(mfrow=c(2,2))
plot(table(psy$c19ProSo01))
plot(table(psy$c19ProSo02))
plot(table(psy$c19ProSo03))
plot(table(psy$c19ProSo04))
```



```
# top 5 happinest country
# United Republic of Tanzania, Uzbekistan, Iceland, Costa Rica, Oman
psy %>% group_by(coded_country) %>% summarise(happinest = mean(happy)) %>% slice_max(happinest, n =5)
```

```
  coded_country                happinest
  <chr>                            <dbl>
1 United Republic of Tanzania      10
2 Uzbekistan                        9.5
3 Iceland                           8.5
4 Costa Rica                        8.25
5 Oman                              8
```

```
# top 5 most educated countries
# ( Ethiopia, Georgia ), Malta, Albania, ( Armenia, Belarus, Cambodia, Ecuador, Kenya, Mauritius, Mongolia, Panama )
psy %>% group_by(coded_country) %>% summarise(education = mean(edu)) %>% slice_max(education, n =5)
```

```
   coded_country education
   <chr>             <dbl>
 1 Ethiopia           7
 2 Georgia            7
 3 Albania            6.25
 4 Armenia            6
 5 Belarus            6
 6 Cambodia           6
 7 Ecuador            6
 8 Kenya              6
 9 Mauritius          6
10 Mongolia           6
11 Panama             6
```

```
# top 5 countries with most old people
# ( Dominican Republic, Georgia, Myanmar ), Oman, Costa Rica
psy %>% group_by(coded_country) %>% summarise(old = mean(age)) %>% slice_max(old, n =5)
```

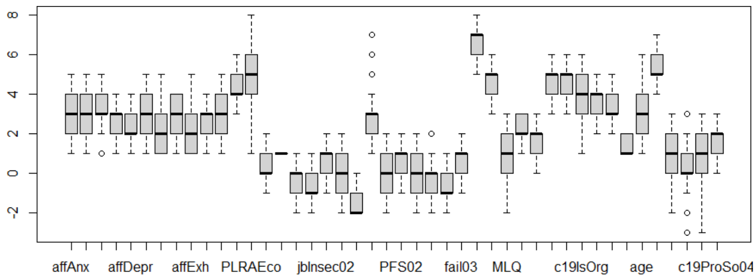
```
  coded_country        old
  <chr>              <dbl>
1 Dominican Republic  6
2 Georgia             6
3 Myanmar             6
4 Oman                5
5 Costa Rica          4.75
```

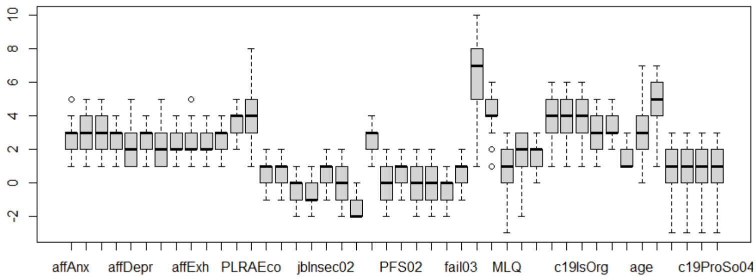
```
# top 5 countries with most teenagers
# Cambodia, Cameroon, Nepal, El Salvador, Moldova
psy %>% group_by(coded_country) %>% summarise(young = mean(age)) %>% slice_min(young, n =5)
```

```
  coded_country young
  <chr>         <dbl>
1 Cambodia       1
2 Cameroon       1
3 Nepal          1
4 El Salvador    1.38
5 Moldova        1.47
```

```
# 1.b ###########################################
# let 'modi' be a copy of the data 'psy'
modi = psy
# Due to the NA values in employment status are meant that boolean values, so can be modified to 0
modi$employstatus_1[is.na(modi$employstatus_1)] = 0
modi$employstatus_2[is.na(modi$employstatus_2)] = 0
modi$employstatus_3[is.na(modi$employstatus_3)] = 0
modi$employstatus_4[is.na(modi$employstatus_4)] = 0
modi$employstatus_5[is.na(modi$employstatus_5)] = 0
modi$employstatus_6[is.na(modi$employstatus_6)] = 0
modi$employstatus_7[is.na(modi$employstatus_7)] = 0
modi$employstatus_8[is.na(modi$employstatus_8)] = 0
modi$employstatus_9[is.na(modi$employstatus_9)] = 0
modi$employstatus_10[is.na(modi$employstatus_10)]  = 0
# Due to every people only can have 1 employment status,
# So if there is multiple selection per pax, it is not logical,
# So filter those rows out.
# data 'modi1' is a copy of modi to drop rows or modifications in columns
 modi1 = modi %>% filter(modi$employstatus_1+modi$employstatus_2+
                   modi$employstatus_3+modi$employstatus_4+
                    modi$employstatus_5+modi$employstatus_6+
                    modi$employstatus_7+modi$employstatus_8+
                    modi$employstatus_9+modi$employstatus_10 == 1)
# Due the 10 empployment status attributes are actually qualitative attributes,
# So it can be combined into 1 attribute
# So change their corresponding value and merge them
# into a new column named 'employstatus'
```

```
modi1$employstatus_2[modi1$employstatus_2 == 1] <- 2
modi1$employstatus_3[modi1$employstatus_3 == 1] <- 3
modi1$employstatus_4[modi1$employstatus_4 == 1] <- 4
modi1$employstatus_5[modi1$employstatus_5 == 1] <- 5
modi1$employstatus_6[modi1$employstatus_6 == 1] <- 6
modi1$employstatus_7[modi1$employstatus_7 == 1] <- 7
modi1$employstatus_8[modi1$employstatus_8 == 1] <- 8
modi1$employstatus_9[modi1$employstatus_9 == 1] <- 9
modi1$employstatus_10[modi1$employstatus_10 == 1] <- 10
employstatus <- rowSums(modi1[,21:30],modi1$employstatus_1 + modi1$employstatus_2 +
                    modi1$employstatus_3 + modi1$employstatus_4 +
                    modi1$employstatus_5 + modi1$employstatus_6 +
                    modi1$employstatus_7 + modi1$employstatus_8 +
                    modi1$employstatus_9 + modi1$employstatus_10)
modi1 <- add_column(modi1, employstatus, .after = 30)
modi1 <- modi1[,-21:-30]
# for age is NA means that they are below 18, so replace by 0
# for edu is NA means that they doesn't have education, so replace by 0
modi1$age[is.na(modi1$age)] = 0
modi1$edu[is.na(modi1$edu)] = 0
modi1 = na.omit(modi1)
# there are no reasons for the other columns to be NA.

# Question 2.a
##########################################################################################
# Focus Country = Greece
# function that detects most outliers
outliers <- function(x) {  Q1 <- quantile(x, probs=.25)
                                Q3 <- quantile(x, probs=.75)
                                iqr = Q3-Q1
                                upper_limit = Q3 + (iqr*1.5)
                                lower_limit = Q1 - (iqr*1.5)
                      x > upper_limit | x < lower_limit}
# function that remove most outliers
remove_outliers <- function(df, cols = names(df)) {
  for (col in cols) {df <- df[!outliers(df[[col]]),]}df}
# 'focus' is data set for Greece, 'focuss' is modification of 'focus',
# that remove the variable coded_country
focus = modi1 %>% filter(coded_country == "Greece")
focuss <- focus[,-41]
# remove outliers in data frame 'focuss'
focuss <- remove_outliers(focuss)
#' other' is data set for all the countries except Greece, 'others' is modification of 'other',
#  that remove the variable coded_country
other = modi1 %>% filter(coded_country != "Greece")
others <- other[,-41]
# remove outliers in data frame 'others'
others <- remove_outliers(others)
# In average,
# we can see there are more people in Greece doesn't have income, compare to other countries.
# We can also see that the people in Greece is more educated.
# All people in Greece felt concerned when think about the future of society
par(mfrow=c(2,1))
boxplot(focuss)
boxplot(others)
```

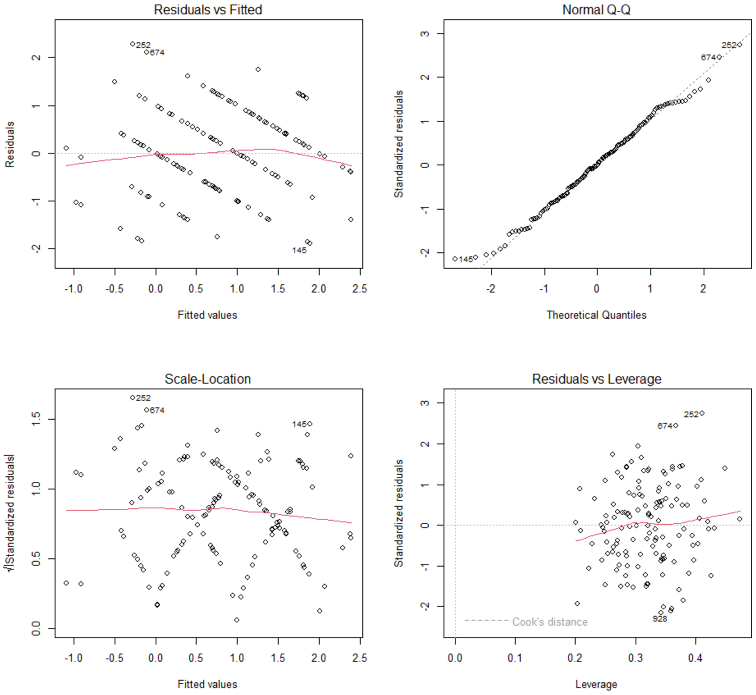





```
# Question 2.b ####################################
# Linear Model used
# Since all the people in Greece have the same value for variable 'disc02',
# so it doesnt make any support to the linear regression,
# so variable 'disc02' dropped.
focuss <- focuss[,-15]
# predict --> c19ProSo01
# partition 80% for training data, 20% for testing data
f1training.samples <- focuss$c19ProSo01 %>% createDataPartition(p = 0.8, list = FALSE)
f1train.data  <- focuss[f1training.samples, ]
f1test.data <- focuss[-f1training.samples, ]
# strongest predictors: c19ProSo02, PLRAC19, affBor, affBor
# p-value: 0.02719
# Multiple R-squared:  0.4292
f1_fit = lm(c19ProSo01 ~ ., data = f1train.data)
summary(f1_fit)
```

```
lm(formula = c19ProSo01 ~ ., data = f1train.data)

Residuals:
     Min       1Q   Median       3Q      Max 
-1.88631 -0.65740 -0.00509  0.63287  2.27582 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)    0.667791   2.047813   0.326 0.745098    
affAnx        -0.117208   0.148608  -0.789 0.432336    
affBor        -0.238843   0.112360  -2.126 0.036237 *  
affCalm       -0.090524   0.153417  -0.590 0.556617    
affContent    -0.095315   0.132077  -0.722 0.472353    
affDepr        0.294568   0.143517   2.052 0.042994 *  
affEnerg       0.146782   0.137961   1.064 0.290169    
affExc         0.123843   0.139205   0.890 0.376002    
affNerv       -0.225678   0.142334  -1.586 0.116308    
affExh        -0.136595   0.123090  -1.110 0.270045    
affInsp        0.101219   0.141245   0.717 0.475444    
affRel         0.014877   0.121790   0.122 0.903045    
PLRAC19        0.246098   0.153529   1.603 0.112412    
PLRAEco        0.064139   0.099189   0.647 0.519499    
disc01        -0.144356   0.173314  -0.833 0.407074    
disc03         0.112085   0.171346   0.654 0.514668    
jbInsec01     -0.267557   0.158429  -1.689 0.094678 .  
jbInsec02     -0.146721   0.193152  -0.760 0.449448    
jbInsec03     -0.136332   0.122964  -1.109 0.270476    
jbInsec04      0.269496   0.207777   1.297 0.197896    
employstatus  -0.006231   0.065957  -0.094 0.924938    
PFS01         -0.035073   0.211359  -0.166 0.868574    
PFS02          0.067024   0.162413   0.413 0.680816    
PFS03         -0.052806   0.211227  -0.250 0.803153    
fail01         0.043085   0.180988   0.238 0.812374    
fail02         0.149101   0.152701   0.976 0.331444    
fail03         0.065088   0.174117   0.374 0.709408    
happy         -0.157351   0.138258  -1.138 0.258067    
lifeSat       -0.107857   0.192593  -0.560 0.576839    
MLQ           -0.143920   0.127962  -1.125 0.263672    
c19NormShould -0.045258   0.168385  -0.269 0.788709    
c19NormDo     -0.235630   0.151671  -1.554 0.123762    
c19IsStrict    0.150784   0.163481   0.922 0.358793    
c19IsPunish   -0.111135   0.172937  -0.643 0.522080    
c19IsOrg       0.102280   0.126989   0.805 0.422677    
trustGovCtry   0.139610   0.171762   0.813 0.418449    
trustGovState  0.036198   0.165757   0.218 0.827621    
gender         0.313406   0.260417   1.203 0.231912    
age            0.013076   0.100299   0.130 0.896559    
edu            0.037908   0.155602   0.244 0.808071    
c19ProSo02     0.340439   0.098058   3.472 0.000793 ***
c19ProSo03     0.160639   0.094946   1.692 0.094087 .  
c19ProSo04    -0.058244   0.132196  -0.441 0.660554    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.083 on 91 degrees of freedom
Multiple R-squared:  0.4292,	Adjusted R-squared:  0.1658 
F-statistic: 1.629 on 42 and 91 DF,  p-value: 0.02719
```

plot(f1_fit)



# make a predict using the test data
f1pred <- f1_fit %>% predict(f1test.data)
# RMSE: 1.580738
RMSE(f1pred, f1test.data$c19ProSo01)

```
> RMSE(f1pred, f1test.data$c19ProSo01)
[1] 1.580738
```

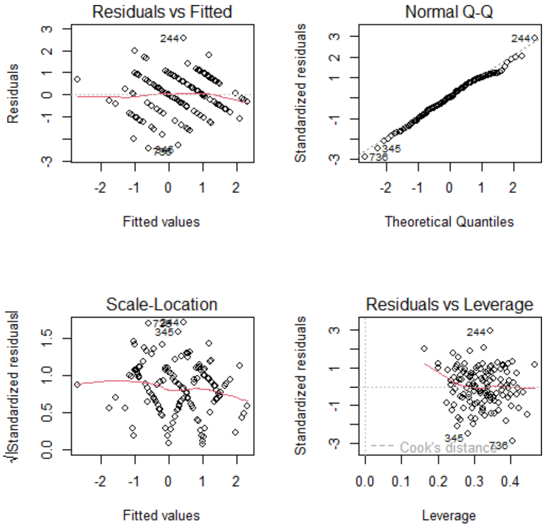
# R-square: 0.01920922
R2(f1pred, f1test.data$c19ProSo01)

```
> R2(f1pred, f1test.data$c19ProSo01)
[1] 0.01920922
```

# predict --> c19ProSo02
# partition 80% for training data, 20% for testing data
f2training.samples <- focuss$c19ProSo02 %>% createDataPartition(p = 0.8, list = FALSE)
f2train.data <- focuss[f1training.samples,]
f2test.data <- focuss[-f1training.samples,]
# strongest predictors: c19ProSo01, c19ProSo03, c19IsStrict, c19NormDo, c19IsPunish
# p-value: 0.0003551
# Multiple R-squared: 0.5202
f2_fit = lm(c19ProSo02 ~ ., data = f2train.data)
summary(f2_fit)

```
lm(formula = c19ProSo02 ~ ., data = f2train.data)

Residuals:
     Min       1Q   Median       3Q      Max
-2.42035 -0.53163  0.03044  0.68335  2.56451

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.129186   2.058354   0.063 0.950094
affAnx        -0.003331   0.149798  -0.022 0.982307
affBor         0.086036   0.115291   0.746 0.457441
affCalm        0.317481   0.150786   2.106 0.038001 *
affContent     0.010813   0.133057   0.081 0.935412
affDepr       -0.213235   0.145770  -1.463 0.146964
affEnerg      -0.012507   0.139446  -0.090 0.928731
affExc        -0.265167   0.137671  -1.926 0.057215 .
affNerv        0.094438   0.144609   0.653 0.515367
affExh         0.313903   0.120060   2.615 0.010457 *
affInsp       -0.207972   0.140611  -1.479 0.142579
affRel         0.008814   0.122354   0.072 0.942729
PLRAC19       -0.152798   0.155572  -0.982 0.328620
PLRAEco        0.027853   0.099830   0.279 0.780870
disc01         0.068642   0.174623   0.393 0.695173
disc03        -0.058485   0.172426  -0.339 0.735251
jbInsec01     -0.069618   0.161464  -0.431 0.667365
jbInsec02     -0.134313   0.194141  -0.692 0.490805
jbInsec03      0.125930   0.123656   1.018 0.311195
jbInsec04     -0.027746   0.210630  -0.132 0.895488
employstatus   0.005005   0.066261   0.076 0.939956
PFS01         -0.317191   0.209741  -1.512 0.133922
PFS02         -0.176638   0.162256  -1.089 0.279190
PFS03          0.376616   0.208564   1.806 0.074262 .
fail01        -0.003671   0.181873  -0.020 0.983938
fail02         0.103683   0.153819   0.674 0.501983
fail03        -0.168770   0.174152  -0.969 0.335067
happy          0.057749   0.139745   0.413 0.680397
lifeSat       -0.313077   0.191010  -1.639 0.104653
MLQ            0.160499   0.128341   1.251 0.214299
c19NormShould  0.088541   0.168969   0.524 0.601547
c19NormDo      0.173343   0.153301   1.131 0.261137
c19IsStrict   -0.279134   0.162380  -1.719 0.089012 .
c19IsPunish    0.430971   0.168160   2.563 0.012023 *
c19IsOrg      -0.135637   0.127233  -1.066 0.289219
trustGovCtry  -0.083321   0.172954  -0.482 0.631140
trustGovState  0.225803   0.164870   1.370 0.174186
gender        -0.302023   0.261776  -1.154 0.251625
age            0.070600   0.100496   0.703 0.484150
edu            0.009697   0.156363   0.062 0.950685
c19ProSo01     0.343566   0.098959   3.472 0.000793 ***
c19ProSo03     0.319281   0.090904   3.512 0.000694 ***
c19ProSo04     0.010238   0.132939   0.077 0.938783
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.088 on 91 degrees of freedom
Multiple R-squared:  0.5202,    Adjusted R-squared:  0.2988
F-statistic: 2.349 on 42 and 91 DF,  p-value: 0.0003551
```

plot(f2_fit)



# make a predict using the test data
f2pred <- f2_fit %>% predict(f2test.data)
#RMSE: 1.516047
RMSE(f2pred, f2test.data$c19ProSo02)

```
> RMSE(f2pred, f2test.data$c19ProSo02)
[1] 1.516047
```

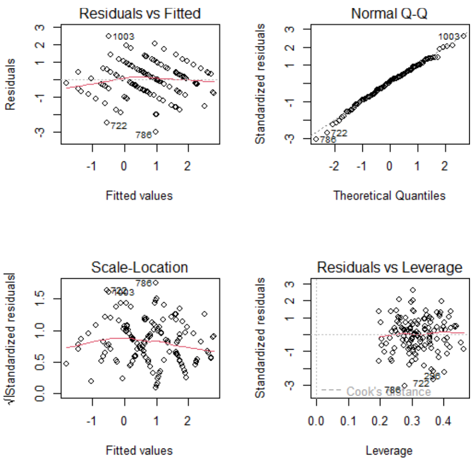
# R-square: 0.001218048
R2(f2pred, f2test.data$c19ProSo02)

```
> R2(f2pred, f2test.data$c19ProSo02)
[1] 0.001218048
```

# predict --> c19ProSo03
# partition 80% for training data
#   20% for testing data
f3training.samples <- focuss$c19ProSo03 %>% createDataPartition(p = 0.8, list = FALSE)
f3train.data <- focuss[f3training.samples,]
f3test.data <- focuss[-f3training.samples,]
# strongest predictors: c19NormDo, edu, lifeSat, c19ProSo04
# p-value: 0.0004024
# Multiple R-squared: 0.5181
f3_fit = lm(c19ProSo03 ~ ., data = f3train.data)
summary(f3_fit)

```
lm(formula = c19ProSo03 ~ ., data = f3train.data)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.98520 -0.64332  0.07926  0.64859  2.47318 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -3.943783   2.110817  -1.868 0.064928 .  
affAnx        -0.088468   0.158685  -0.558 0.578550    
affBor         0.095163   0.121969   0.780 0.437287    
affCalm       -0.262676   0.167330  -1.570 0.119930    
affContent     0.112945   0.156460   0.722 0.472223    
affDepr        0.183333   0.155566   1.178 0.241674    
affEnerg       0.153596   0.156009   0.985 0.327462    
affExc         0.004832   0.150855   0.032 0.974516    
affNerv        0.075596   0.154539   0.489 0.625899    
affExh        -0.192563   0.131537  -1.464 0.146656    
affInsp        0.331883   0.159752   2.077 0.040574 *  
affRel         0.075601   0.138700   0.545 0.587041    
PLRAC19        0.034108   0.148491   0.230 0.818841    
PLRAEco        0.047173   0.107646   0.438 0.662264    
disc01         0.153603   0.178957   0.858 0.392969    
disc03         0.031540   0.186328   0.169 0.865957    
jbInsec01      0.032867   0.203959   0.161 0.872334    
jbInsec02      0.217437   0.230152   0.945 0.347286    
jbInsec03      0.150282   0.133434   1.126 0.263015    
jbInsec04      0.167398   0.224189   0.747 0.457180    
employstatus  -0.008665   0.079885  -0.108 0.913867    
PFS01          0.135483   0.222349   0.609 0.543826    
PFS02         -0.077380   0.174659  -0.443 0.658792    
PFS03         -0.221178   0.206391  -1.072 0.286713    
fail01        -0.043628   0.175821  -0.248 0.804587    
fail02        -0.344950   0.150480  -2.292 0.024192 *  
fail03         0.118193   0.178766   0.661 0.510178    
happy         -0.030208   0.153366  -0.197 0.844295    
lifeSat        0.426184   0.216934   1.965 0.052514 .  
MLQ           -0.249305   0.142406  -1.751 0.083375 .  
c19NormShould -0.237889   0.166180  -1.432 0.155708    
c19NormDo      0.354058   0.148015   2.392 0.018811 *  
c19IsStrict   -0.073342   0.178704  -0.410 0.682468    
c19IsPunish   -0.172515   0.179611  -0.960 0.339354    
c19IsOrg       0.104692   0.141050   0.742 0.459857    
trustGovCtry   0.086660   0.176298   0.492 0.624218    
trustGovState -0.089091   0.172036  -0.518 0.605810    
gender        -0.010607   0.256814  -0.041 0.967145    
age           -0.108961   0.105012  -1.038 0.302204    
edu            0.297334   0.165833   1.793 0.076299 .  
c19ProSo01     0.196677   0.101198   1.943 0.055047 .  
c19ProSo02     0.332141   0.093897   3.537 0.000639 ***
c19ProSo04     0.300239   0.139826   2.147 0.034434 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.148 on 91 degrees of freedom
Multiple R-squared:  0.5181,	Adjusted R-squared:  0.2957 
F-statistic: 2.329 on 42 and 91 DF,  p-value: 0.0004024
```

plot(f3_fit)



# make a predict using the test data
f3pred <- f3_fit %>% predict(f3test.data)
# RMSE: 1.231611
RMSE(f3pred, f3test.data$c19ProSo03)

```
> RMSE(f3pred, f3test.data$c19ProSo03)
[1] 1.231611
```

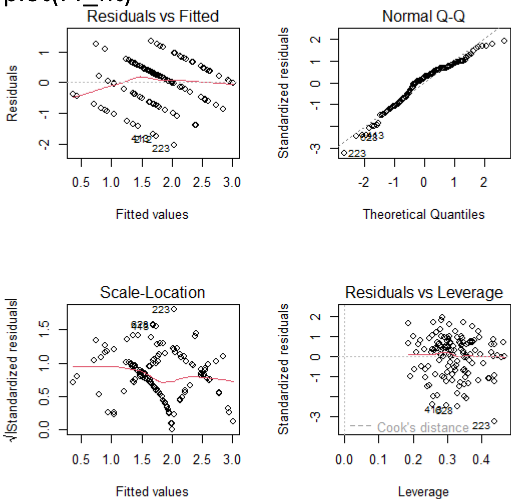
# R-square: 0.278455
R2(f3pred, f3test.data$c19ProSo03)

```
> R2(f3pred, f3test.data$c19ProSo03)
[1] 0.278455
```

# predict --> c19ProSo04
# partition 80% for training data, 20% for testing data
f4training.samples <- focuss$c19ProSo04 %>% createDataPartition(p = 0.8, list = FALSE)
f4train.data <- focuss[f4training.samples,]
f4test.data <- focuss[-f4training.samples,]
# strongest predictors: fail03, PLRAEco
# p-value: 0.1754
# Multiple R-squared: 0.3687
f4_fit = lm(c19ProSo04 ~ ., data = f4train.data)
summary(f4_fit)

```
lm(formula = c19ProSo04 ~ ., data = f4train.data)

Residuals:
    Min      1Q  Median      3Q     Max 
-2.0303 -0.4822  0.1564  0.4632  1.3509 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)  
(Intercept)    2.209342   1.677585   1.317   0.1912  
affAnx         0.023276   0.117603   0.198   0.8435  
affBor        -0.041786   0.091437  -0.457   0.6488  
affCalm        0.070022   0.113789   0.615   0.5399  
affContent     0.017033   0.102671   0.166   0.8686  
affDepr        0.005034   0.110368   0.046   0.9637  
affEnerg      -0.061462   0.111429  -0.552   0.5826  
affExc        -0.176568   0.111219  -1.588   0.1159  
affNerv       -0.142085   0.124210  -1.144   0.2557  
affExh         0.028600   0.095183   0.300   0.7645  
affInsp        0.196198   0.117259   1.673   0.0977 .
affRel        -0.073323   0.093300  -0.786   0.4340  
PLRAC19       -0.173201   0.128295  -1.350   0.1804  
PLRAEco        0.116889   0.079322   1.474   0.1440  
disc01        -0.025943   0.135669  -0.191   0.8488  
disc03         0.062282   0.146294   0.426   0.6713  
jbInsec01     -0.099184   0.139591  -0.711   0.4792  
jbInsec02      0.016995   0.148787   0.114   0.9093  
jbInsec03      0.056900   0.094032   0.605   0.5466  
jbInsec04      0.082982   0.154354   0.538   0.5922  
employstatus  -0.061875   0.053758  -1.151   0.2527  
PFS01          0.162497   0.162172   1.002   0.3190  
PFS02         -0.018457   0.137320  -0.134   0.8934  
PFS03         -0.249994   0.161469  -1.548   0.1250  
fail01         0.111557   0.132066   0.845   0.4005  
fail02        -0.072366   0.118032  -0.613   0.5413  
fail03         0.197934   0.130979   1.511   0.1342  
happy          0.001266   0.109398   0.012   0.9908  
lifeSat       -0.116673   0.151465  -0.770   0.4431  
MLQ            0.128293   0.099981   1.283   0.2027  
c19NormShould  0.248503   0.129383   1.921   0.0579 .
c19NormDo     -0.097250   0.111231  -0.874   0.3843  
c19IsStrict    0.106508   0.128618   0.828   0.4098  
c19IsPunish    0.099205   0.137925   0.719   0.4738  
c19IsOrg      -0.064487   0.102400  -0.630   0.5304  
trustGovCtry   0.253880   0.144983   1.751   0.0833 .
trustGovState -0.159237   0.126400  -1.260   0.2110  
gender        -0.067627   0.196360  -0.344   0.7313  
age           -0.056226   0.086779  -0.648   0.5187  
edu           -0.136689   0.119618  -1.143   0.2562  
c19ProSo01    -0.040365   0.075761  -0.533   0.5955  
c19ProSo02    -0.023476   0.076282  -0.308   0.7590  
c19ProSo03     0.117101   0.073106   1.602   0.1127  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.8335 on 91 degrees of freedom
Multiple R-squared:  0.3687,	Adjusted R-squared:  0.07733 
F-statistic: 1.265 on 42 and 91 DF,  p-value: 0.1754
```

plot(f4_fit)



# make a predict using the test data
f4pred <- f4_fit %>% predict(f4test.data)
# RMSE: 1.024248
RMSE(f4pred, f4test.data$c19ProSo04)

```
> RMSE(f4pred, f4test.data$c19ProSo04)
[1] 1.024248
```

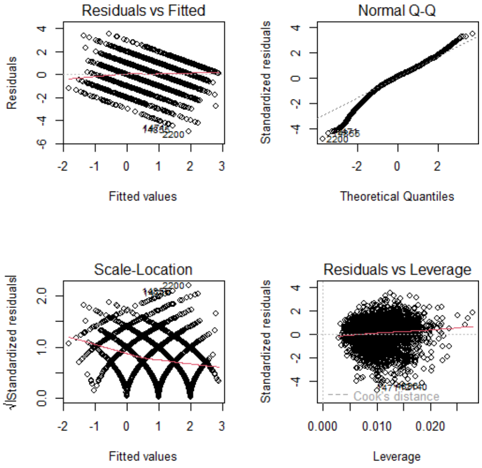
# R-square: 0.003979094
R2(f4pred, f4test.data$c19ProSo04)

```
> R2(f4pred, f4test.data$c19ProSo04)
[1] 0.003979074
```

# Question 2.c ####################################
# predict --> c19ProSo01
# partition 80% for training data, 20% for testing data
o1training.samples <- others$c19ProSo01 %>% createDataPartition(p = 0.8, list = FALSE)
o1train.data <- others[o1training.samples,]
o1test.data <- others[-o1training.samples,]
# strongest predictors: c19ProSo02, c19ProSo03, c19ProSo04
# p-value: < 2.2e-16
# Multiple R-squared: 0.3632
o1_fit = lm(c19ProSo01 ~ ., data = o1train.data)
summary(o1_fit)

```
Residuals:
    Min      1Q  Median      3Q     Max 
-4.9359 -0.5177  0.0936  0.6216  3.5478 

Coefficients:
                Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -0.5120717  0.1994134  -2.568 0.010266 *  
affAnx         0.0056021  0.0194621   0.288 0.773478    
affBor        -0.0163201  0.0145365  -1.123 0.261628    
affCalm        0.0442855  0.0213407   2.075 0.038031 *  
affContent    -0.0024288  0.0214811  -0.113 0.909982    
affDepr       -0.0311983  0.0200790  -1.554 0.120311    
affEnerg       0.0423509  0.0211402   2.003 0.045205 *  
affExc        -0.0229421  0.0191739  -1.197 0.231557    
affNerv       -0.0105221  0.0207781  -0.506 0.612600    
affExh         0.0292594  0.0168668   1.735 0.082860 .  
affInsp        0.0242124  0.0200845   1.206 0.228068    
affRel         0.0069394  0.0223218   0.311 0.755907    
PLRAC19        0.0263763  0.0176916   1.491 0.136064    
PLRAEco        0.0425226  0.0128766   3.302 0.000967 ***
disc01         0.0010440  0.0252183   0.041 0.966981    
disc02        -0.0085583  0.0262730  -0.326 0.744634    
disc03        -0.0122719  0.0211432  -0.580 0.561662    
jbInsec01     -0.0348655  0.0266339  -1.309 0.190584    
jbInsec02      0.0165924  0.0256287   0.647 0.517398    
jbInsec03      0.0026921  0.0193625   0.139 0.889429    
jbInsec04     -0.0617295  0.0274526  -2.249 0.024589 *  
employstatus   0.0085118  0.0188086   0.453 0.650898    
PFS01          0.0149567  0.0240296   0.622 0.533693    
PFS02         -0.0111682  0.0221645  -0.504 0.614372    
PFS03          0.0603874  0.0229614   2.630 0.008570 ** 
fail01         0.0049427  0.0188021   0.263 0.792654    
fail02        -0.0232745  0.0204451  -1.138 0.255023    
fail03         0.0388402  0.0208008   1.867 0.061936 .  
happy          0.0159924  0.0128845   1.241 0.214596    
lifeSat       -0.0331126  0.0222983  -1.485 0.137623    
MLQ            0.0388739  0.0145357   2.674 0.007515 ** 
c19NormShould  0.0111543  0.0176845   0.631 0.528246    
c19NormDo      0.0166202  0.0201941   0.823 0.410541    
c19IsStrict   -0.0018843  0.0178316  -0.106 0.915849    
c19IsPunish   -0.0006084  0.0134813  -0.045 0.964008    
c19IsOrg       0.0335818  0.0175303   1.916 0.055477 .  
trustGovCtry  -0.0115840  0.0174330  -0.664 0.506415    
trustGovState  0.0580254  0.0224296   2.587 0.009714 ** 
gender         0.0360926  0.0324033   1.114 0.265404    
age            0.0043685  0.0130940   0.334 0.738678    
edu            0.0049652  0.0124005   0.400 0.688881    
c19ProSo02     0.2625339  0.0133904  19.606  < 2e-16 ***
c19ProSo03     0.2284124  0.0134187  17.022  < 2e-16 ***
c19ProSo04     0.1121634  0.0132481   8.466  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.024 on 4269 degrees of freedom
Multiple R-squared:  0.3632,	Adjusted R-squared:  0.3568 
F-statistic: 56.62 on 43 and 4269 DF,  p-value: < 2.2e-16
```

plot(o1_fit)



# make a predict using the test data
o1pred <- o1_fit %>% predict(o1test.data)
# RMSE: 1.058412
RMSE(o1pred, o1test.data$c19ProSo01)

```
> RMSE(o1pred, o1test.data$c19ProSo01)
[1] 1.058412
```

# R-square: 0.3671841
R2(o1pred, o1test.data$c19ProSo01)

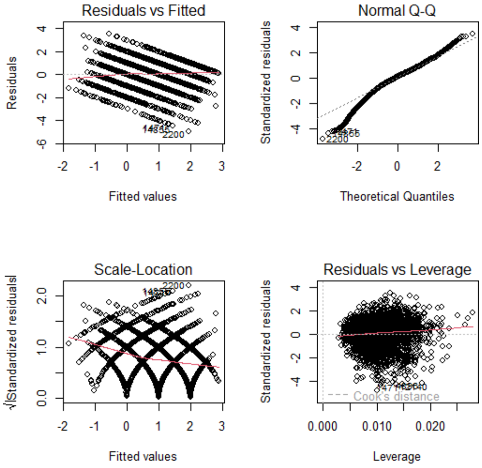
```
> R2(o1pred, o1test.data$c19ProSo01)
[1] 0.3671841
```

compared to the focus country model, they only have 1 same indicator which is
the 'c19ProSo02', and for 2 of the models 'c19ProSo02' is the strongest predictor.
For focus country have another 3 indicators which is PLRAC19, affBor and affBor.
Whereas for All other countries, have another 2 indicators which is c19ProSo03 and c19ProSo04.

# predict --> c19ProSo02
# partition 80% for training data
#   20% for testing data
o2training.samples <- others$c19ProSo02 %>% createDataPartition(p = 0.8, list = FALSE)
o2train.data <- others[o2training.samples,]
o2test.data <- others[-o2training.samples,]
# strongest predictors: c19ProSo03, c19ProSo01
# p-value: < 2.2e-16
# Multiple R-squared: 0.3997
o2_fit = lm(c19ProSo02 ~ ., data = o2train.data)
summary(o2_fit)

```
Residuals:
    Min      1Q  Median      3Q     Max 
-5.1092 -0.5725  0.1263  0.6734  4.7347 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -0.948533   0.219952  -4.312 1.65e-05 ***
affAnx         0.049089   0.021536   2.279 0.022694 *  
affBor         0.063127   0.015785   3.999 6.47e-05 ***
affCalm       -0.060832   0.023597  -2.578 0.009972 ** 
affContent    -0.071618   0.023696  -3.022 0.002522 ** 
affDepr       -0.031933   0.022050  -1.448 0.147622    
affEnerg      -0.011407   0.023414  -0.487 0.626152    
affExc         0.084732   0.021257   3.986 6.83e-05 ***
affNerv       -0.031256   0.022803  -1.371 0.170540    
affExh         0.015848   0.018579   0.853 0.393687    
affInsp        0.038161   0.022147   1.723 0.084945 .  
affRel         0.018701   0.024643   0.759 0.447956    
PLRAC19       -0.042407   0.019388  -2.187 0.028774 *  
PLRAEco       -0.010783   0.014192  -0.760 0.447406    
disc01        -0.018251   0.027823  -0.656 0.511864    
disc02         0.140356   0.029087   4.825 1.45e-06 ***
disc03         0.001848   0.023624   0.078 0.937658    
jbInsec01      0.014494   0.029259   0.495 0.620370    
jbInsec02     -0.014071   0.028196  -0.499 0.617769    
jbInsec03      0.034479   0.021437   1.608 0.107823    
jbInsec04      0.022083   0.030069   0.734 0.462735    
employstatus   0.040779   0.020749   1.965 0.049437 *  
PFS01         -0.079516   0.026410  -3.011 0.002621 ** 
PFS02         -0.006242   0.024277  -0.257 0.797089    
PFS03         -0.012403   0.025319  -0.490 0.624240    
fail01        -0.040480   0.020747  -1.951 0.051107 .  
fail02        -0.046915   0.022485  -2.087 0.036987 *  
fail03        -0.008140   0.022733  -0.358 0.720314    
happy          0.034889   0.014163   2.463 0.013805 *  
lifeSat        0.040541   0.024814   1.634 0.102377    
MLQ            0.029290   0.016027   1.828 0.067684 .  
c19NormShould  0.071237   0.019475   3.658 0.000257 ***
c19NormDo      0.041604   0.022068   1.885 0.059462 .  
c19IsStrict    0.010476   0.019718   0.531 0.595246    
c19IsPunish    0.012313   0.014824   0.831 0.406239    
c19IsOrg      -0.031912   0.019282  -1.655 0.098003 .  
trustGovCtry   0.052254   0.018997   2.751 0.005974 ** 
trustGovState  0.041032   0.024651   1.664 0.096087 .  
gender        -0.057251   0.035634  -1.607 0.108205    
age            0.002613   0.014422   0.181 0.856213    
edu            0.055556   0.013653   4.069 4.80e-05 ***
c19ProSo01     0.316760   0.015930  19.885  < 2e-16 ***
c19ProSo03     0.323415   0.014445  22.390  < 2e-16 ***
c19ProSo04     0.019453   0.014737   1.320 0.186909    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.126 on 4268 degrees of freedom
Multiple R-squared:  0.3997,	Adjusted R-squared:  0.3937 
F-statistic: 66.09 on 43 and 4268 DF,  p-value: < 2.2e-16
```

plot(o2_fit)



# make a predict using the test data
o2pred <- o2_fit %>% predict(o2test.data)
# RMSE: 1.105373
RMSE(o2pred, o2test.data$c19ProSo02)

```
> RMSE(o2pred, o2test.data$c19ProSo02)
[1] 1.105373
```

# R-square: 0.4252648
R2(o2pred, o2test.data$c19ProSo02)

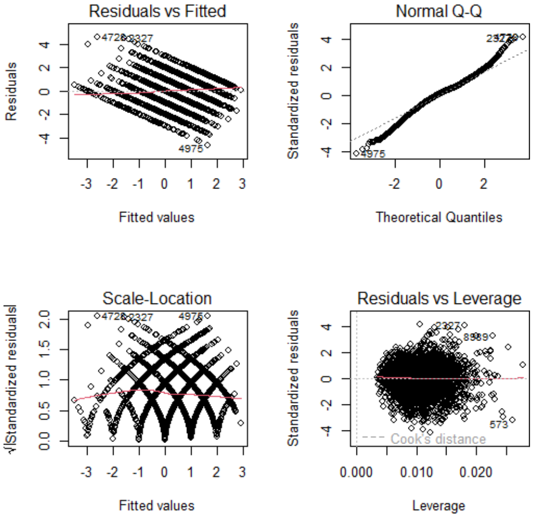
```
> R2(o2pred, o2test.data$c19ProSo02)
[1] 0.4252648
```

compared to the focus country model, they have 2 same significant indicator which is
'c19ProSo01' and 'c19ProSo03' and for 2 of the models 'c19ProSo01' and 'c19ProSo03'
is the strongest predictor. # For focus country have another 3 indicators
which is c19IsStrict, c19NormDo, c19IsPunish.

# predict --> c19ProSo03
# partition 80% for training data, 20% for testing data
o3training.samples <- others$c19ProSo03 %>% createDataPartition(p = 0.8, list = FALSE)
o3train.data <- others[o3training.samples,]
o3test.data <- others[-o3training.samples,]
# strongest predictors: c19ProSo01, c19ProSo02, c19ProSo04
# p-value: < 2.2e-16
# Multiple R-squared: 0.4306
o3_fit = lm(c19ProSo03 ~ ., data = o3train.data)
summary(o3_fit)

```
Residuals:
    Min      1Q  Median      3Q     Max 
-4.6222 -0.5951  0.1238  0.6497  4.6164 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -0.136207   0.217838  -0.625 0.531829    
affAnx         0.005983   0.021236   0.282 0.778167    
affBor        -0.055222   0.015746  -3.507 0.000458 ***
affCalm       -0.005951   0.023402  -0.254 0.799274    
affContent     0.011061   0.023620   0.468 0.639606    
affDepr        0.053974   0.021761   2.480 0.013164 *  
affEnerg      -0.013383   0.022914  -0.584 0.559226    
affExc         0.026196   0.020882   1.254 0.209731    
affNerv       -0.016688   0.022821  -0.731 0.464652    
affExh         0.013855   0.018422   0.752 0.452052    
affInsp       -0.018708   0.021728  -0.861 0.389281    
affRel        -0.001081   0.024466  -0.044 0.964759    
PLRAC19        0.040099   0.019153   2.094 0.036358 *  
PLRAEco       -0.030094   0.014039  -2.144 0.032127 *  
disc01         0.034955   0.027481   1.272 0.203463    
disc02        -0.029437   0.028425  -1.036 0.300443    
disc03         0.037116   0.023207   1.599 0.109823    
jbInsec01     -0.012122   0.029251  -0.414 0.678596    
jbInsec02     -0.017633   0.027780  -0.635 0.525644    
jbInsec03     -0.010605   0.021143  -0.502 0.615964    
jbInsec04      0.030258   0.030019   1.008 0.313520    
employstatus  -0.012175   0.020713  -0.588 0.556695    
PFS01          0.014636   0.026249   0.558 0.577160    
PFS02         -0.024349   0.023884  -1.019 0.308038    
PFS03         -0.029864   0.025057  -1.192 0.233396    
fail01        -0.028283   0.020625  -1.371 0.170360    
fail02         0.051998   0.022177   2.345 0.019089 *  
fail03         0.001641   0.022708   0.072 0.942413    
happy         -0.015059   0.014221  -1.059 0.289688    
lifeSat        0.019774   0.024062   0.822 0.411241    
MLQ           -0.009162   0.015969  -0.574 0.566176    
c19NormShould -0.084681   0.019392  -4.367 1.29e-05 ***
c19NormDo      0.042072   0.021931   1.918 0.055124 .  
c19IsStrict   -0.038131   0.019290  -1.977 0.048136 *  
c19IsPunish    0.016317   0.014825   1.101 0.271108    
c19IsOrg       0.000750   0.019056   0.039 0.968609    
trustGovCtry  -0.011032   0.019068  -0.579 0.562911    
trustGovState  0.062255   0.024758   2.515 0.011955 *  
gender         0.053901   0.035504   1.518 0.129049    
age           -0.042370   0.014333  -2.956 0.003133 ** 
edu            0.018184   0.013579   1.339 0.180580    
c19ProSo01     0.277446   0.016083  17.251  < 2e-16 ***
c19ProSo02     0.328495   0.014538  22.595  < 2e-16 ***
c19ProSo04     0.290431   0.013867  20.944  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.117 on 4269 degrees of freedom
Multiple R-squared:  0.4306,	Adjusted R-squared:  0.4249 
F-statistic: 75.09 on 43 and 4269 DF,  p-value: < 2.2e-16
```

plot(o3_fit)



# make a predict using the test data
o3pred <- o3_fit %>% predict(o3test.data)
# RMSE: 1.15123
RMSE(o3pred, o3test.data$c19ProSo03)

```
> RMSE(o3pred, o3test.data$c19ProSo03)
[1] 1.15123
```

# R-square: 0.4141071
R2(o3pred, o3test.data$c19ProSo03)

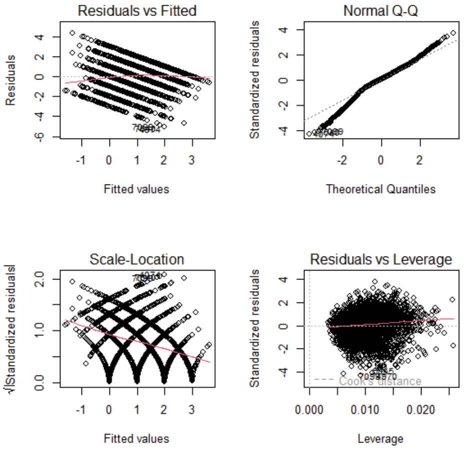
```
> R2(o3pred, o3test.data$c19ProSo03)
[1] 0.4141071
```

compared to the focus country model, they have only 1 same significant indicator which is
'c19ProSo04'and for 2 of the models 'c19ProSo04' is the strongest predictor.
For focus country have another 3 indicators which is c19NormDo, edu, lifeSat.
Whereas for All other countries, have another 2 indicators which is c19ProSo01 and c19ProSo02.

# predict --> c19ProSo04
# partition 80% for training data, 20% for testing data
o4training.samples <- others$c19ProSo04 %>% createDataPartition(p = 0.8, list = FALSE)
o4train.data <- others[o4training.samples,]
o4test.data <- others[-o4training.samples,]
# strongest predictors: c19ProSo01, c19ProSo03, c19NormShould
# p-value: < 2.2e-16
# Multiple R-squared: 0.3293
o4_fit = lm(c19ProSo04 ~ ., data = o4train.data)
summary(o4_fit)

```
lm(formula = c19ProSo04 ~ ., data = o4train.data)

Residuals:
    Min      1Q  Median      3Q     Max 
-5.0048 -0.5647  0.0721  0.7246  4.3306 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)    0.2599667  0.2290983   1.135 0.256549    
affAnx        -0.0023600  0.0224836  -0.105 0.916407    
affBor         0.0147184  0.0164959   0.892 0.372312    
affCalm        0.0448518  0.0243536   1.842 0.065590 .  
affContent     0.0036933  0.0247769   0.149 0.881512    
affDepr       -0.0217377  0.0228829  -0.950 0.342191    
affEnerg      -0.0707298  0.0242425  -2.918 0.003546 ** 
affExc        -0.0264779  0.0218877  -1.210 0.226455    
affNerv       -0.0110352  0.0239139  -0.461 0.644495    
affExh        -0.0208185  0.0192466  -1.082 0.279459    
affInsp       -0.0271113  0.0226744  -1.196 0.231888    
affRel         0.0193324  0.0254478   0.760 0.447484    
PLRAC19        0.0255316  0.0202175   1.263 0.206713    
PLRAEco        0.0128776  0.0146301   0.880 0.378792    
disc01        -0.0323785  0.0285438  -1.134 0.256713    
disc02         0.1187438  0.0293592   4.045 5.34e-05 ***
disc03         0.0508703  0.0243532   2.089 0.036780 *  
jbInsec01      0.0332875  0.0308055   1.081 0.279949    
jbInsec02      0.0072912  0.0295878   0.246 0.805363    
jbInsec03     -0.0090173  0.0223701  -0.403 0.686899    
jbInsec04     -0.0187545  0.0314795  -0.596 0.551361    
employstatus   0.0005043  0.0214753   0.023 0.981265    
PFS01         -0.0702588  0.0274521  -2.559 0.010522 *  
PFS02          0.0993162  0.0252014   3.941 8.25e-05 ***
PFS03          0.0131874  0.0263295   0.501 0.616494    
fail01        -0.0719809  0.0216367  -3.327 0.000886 ***
fail02        -0.0514948  0.0230327  -2.236 0.025421 *  
fail03         0.0639821  0.0236374   2.707 0.006820 ** 
happy         -0.0080179  0.0147508  -0.544 0.586776    
lifeSat        0.0404076  0.0254004   1.591 0.111723    
MLQ            0.0074531  0.0165801   0.450 0.653077    
c19NormShould  0.2123543  0.0200063  10.614  < 2e-16 ***
c19NormDo      0.0021643  0.0230540   0.094 0.925208    
c19IsStrict    0.0683922  0.0204079   3.351 0.000811 ***
c19IsPunish   -0.0700168  0.0153380  -4.565 5.14e-06 ***
c19IsOrg       0.0310906  0.0199505   1.558 0.119215    
trustGovCtry  -0.0903438  0.0198878  -4.543 5.71e-06 ***
trustGovState  0.0658261  0.0260583   2.526 0.011569 *  
gender        -0.0510472  0.0370468  -1.378 0.168303    
age            0.0314205  0.0150012   2.095 0.036272 *  
edu           -0.0509845  0.0140706  -3.623 0.000294 ***
c19ProSo01     0.1668681  0.0171827   9.711  < 2e-16 ***
c19ProSo02     0.0365894  0.0159309   2.297 0.021681 *  
c19ProSo03     0.3289226  0.0151100  21.769  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.168 on 4268 degrees of freedom
Multiple R-squared:  0.3293,	Adjusted R-squared:  0.3225 
F-statistic: 48.73 on 43 and 4268 DF,  p-value: < 2.2e-16
```

plot(o4_fit)



# make a predict using the test data
o4pred <- o4_fit %>% predict(o4test.data)
# RMSE: 1.158719
RMSE(o4pred, o4test.data$c19ProSo04)

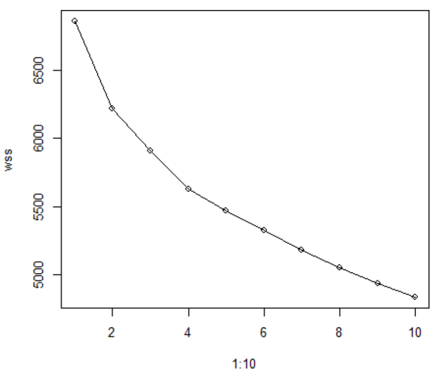
```
> RMSE(o4pred, o4test.data$c19ProSo04)
[1] 1.183097
```

# R-square: 0.3216867
R2(o4pred, o4test.data$c19ProSo04)

```
> R2(o4pred, o4test.data$c19ProSo04)
[1] 0.2851207
```

strongest predictors:
compared to the focus country model, they don't have same significant indicator.
for focus country model have 'fail03', 'PLRAEco' as the strongest predictor.
For All other countries, have c19ProSo01, c19ProSo03, c19NormShould as the
strongest predictor.

Question 3.a
##
# to reproduce the results
set.seed(32637888)
# to find the best number of cluster
# best: 3
wss<- NULL
for (i in 1:10){
  fit = kmeans(focuss,centers = i)
  wss = c(wss, fit$tot.withinss)
}
plot(1:10, wss, type = "o")

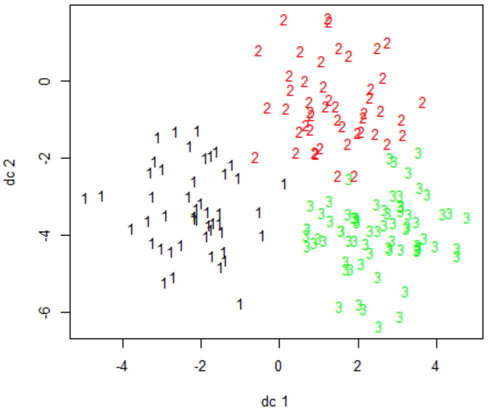


Attributes used: "affAnx","affBor","affCalm","affContent","affDepr","affEnerg" ,
"affExc","affNerv","affExh","affInsp","affRel","PLRAC19","PLRAEco","disc01",
"disc03","jbInsec01" ,"jbInsec02","jbInsec03","jbInsec04","employstatus","PFS01",
"PFS02","PFS03","fail01","fail02","fail03","happy",lifeSat","MLQ","c19NormShould",
"c19NormDo","c19IsStrict","c19IsPunish","c19IsOrg","trustGovCtry","trustGovState",
"gender","age","edu","c19ProSo01","c19ProSo02","c19ProSo03","c19ProSo04"

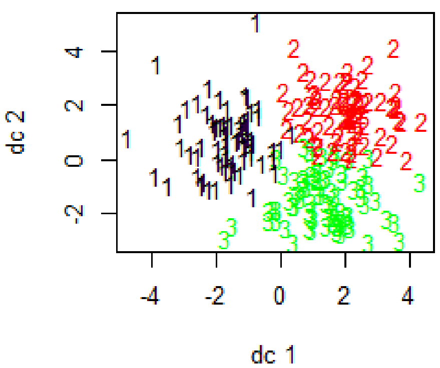
attributes(focuss)

```
> attributes(focuss)
$names
 [1] "affAnx"         "affBor"         "affCalm"        "affContent"     "affDepr"        "affEnerg"       "affExc"         "affNerv"        "affExh"
[10] "affInsp"        "affRel"         "PLRAC19"        "PLRAEco"        "disc01"         "disc03"         "jbInsec01"      "jbInsec02"      "jbInsec03"
[19] "jbInsec04"      "employstatus"   "PFS01"          "PFS02"          "PFS03"          "fail01"         "fail02"         "fail03"         "happy"
[28] "lifeSat"        "MLQ"            "c19NormShould"  "c19NormDo"      "c19IsStrict"    "c19IsPunish"    "c19IsOrg"       "trustGovCtry"   "trustGovState"
[37] "gender"         "age"            "edu"            "c19ProSo01"     "c19ProSo02"     "c19ProSo03"     "c19ProSo04"

$class
[1] "data.frame"

$row.names
  [1]   13   17   18   24   26   29   44   49   56   67   78   90  101  102  108  130  131  136  138  145  148  162  163  168  169  173  174  184  187  190  202
 [32]  203  210  212  215  217  223  224  226  234  236  244  245  252  253  260  263  270  274  290  296  302  306  307  308  313  314  322  324  342  345  348
 [63]  349  350  356  360  363  381  382  390  409  411  413  414  421  425  426  427  451  452  477  479  491  496  508  510  520  542  546  548  555  558  566
 [94]  571  573  575  577  594  601  602  606  619  623  625  626  628  632  637  654  655  658  660  674  694  698  703  706  709  710  722  724  727  734  736
[125]  744  747  763  764  769  777  779  786  793  805  812  814  815  817  833  835  844  872  881  882  889  898  904  913  917  925  927  928  930  932  951
[156]  952  959  967  971  972  975  978  991  996  999 1003
```

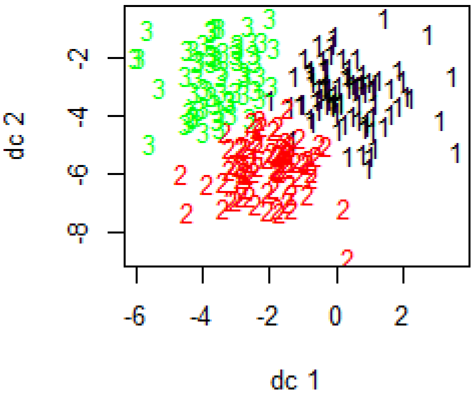
# Clustering to fit --> c19ProSo01
# focus country --> Greece
par(mfrow=c(2,2))
# k-means clustering
f1fit = kmeans(focuss,3,nstart=30)
# table to show the fitness
T1 <- table(actual = focuss$c19ProSo01, fitted = f1fit$cluster)
T1 = as.data.frame.matrix(T1)
# show the column names
colnames(T1)
T1
# plot the cluster chart
plotcluster(focuss,f1fit$cluster,pointsbyclvecd = TRUE)



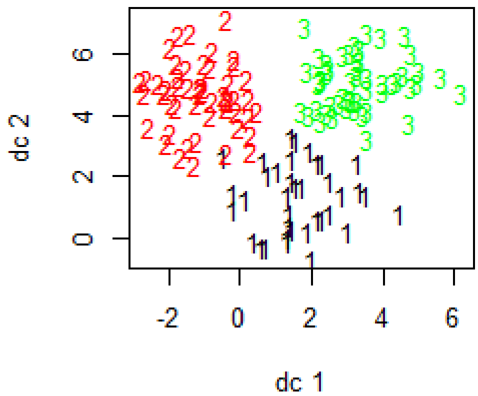
# Country --> Republic of Serbia
any1 = modi1 %>% filter(coded_country == "Republic of Serbia")
any1s <- any1[,-41]
any1s <- any1s[,-15]
# remove outliers in data frame 'any1s'
any1s <- remove_outliers(any1s)
f2fit = kmeans(any1s,3,nstart=30)
T2 <- table(actual = any1s$c19ProSo01, fitted = f2fit$cluster)
T2 = as.data.frame.matrix(T1)
colnames(T2)
T2
plotcluster(any1s,f2fit$cluster,pointsbyclvecd = TRUE)



```
# Country --> Ukraine
any2 = modi1 %>% filter(coded_country == "Ukraine")
any2s <- any2[,-41]
any2s <- any2s[,-15]
# remove outliers in data frame 'any1s'
any2s = na.omit(any2s)
any2s <- remove_outliers(any2s)
f3fit = kmeans(any2s,3,nstart=30)
T3 <- table(actual = any2s$c19ProSo01, fitted = f3fit$cluster)
T3 = as.data.frame.matrix(T3)
colnames(T3)
T3
plotcluster(any2s,f3fit$cluster,pointsbyclvecd = TRUE)
```

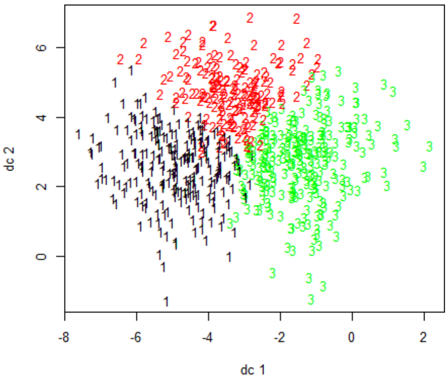



```
# Country --> United Kingdom
any3 = modi1 %>% filter(coded_country == "United Kingdom")
any3s <- any3[,-41]
any3s <- any3s[,-15]
# remove outliers in data frame 'any3s'
any3s = na.omit(any3s)
any3s <- remove_outliers(any3s)
f4fit = kmeans(any3s,3,nstart=30)
T4 <- table(actual = any3s$c19ProSo01, fitted = f4fit$cluster)
T4 = as.data.frame.matrix(T4)
colnames(T4)
T4
plotcluster(any3s,f4fit$cluster,pointsbyclvecd = TRUE)
```




```
# Country --> United Kingdom
any4 = modi1 %>% filter(coded_country == "United States of America")
any4s <- any4[,-41]
any4s <- any4s[,-15]
# remove outliers in data frame 'any3s'
any4s = na.omit(any4s)
any4s <- remove_outliers(any4s)
```

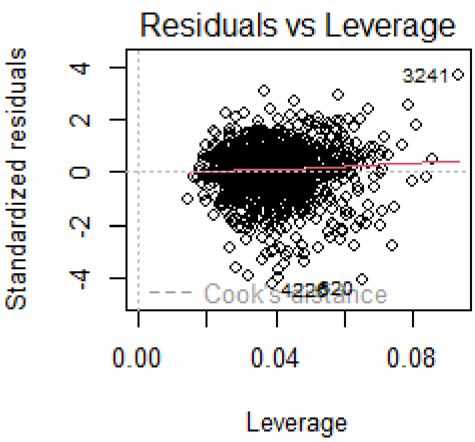
```
f5fit = kmeans(any4s,3,nstart=30)
T5 <- table(actual = any4s$c19ProSo01, fitted = f5fit$cluster)
T5 = as.data.frame.matrix(T5)
colnames(T5)
T5
plotcluster(any4s,f5fit$cluster,pointsbyclvecd = TRUE)
```




```
# Similar countries: Republic of Serbia, Ukraine, United Kingdom, United States of America

# Question 3.b ####################################
Smlar = modi1 %>% filter(coded_country == "United States of America" | coded_country == "United Kingdom" |
coded_country == "Ukraine" | coded_country == "Republic of Serbia")
Smlar <- Smlar[,-41]
Smlar <- Smlar[,-15]
Smlar = na.omit(Smlar)
Smlar <- remove_outliers(Smlar)

# predict --> c19ProSo01
# partition the data for training and testing
Smtraining.samples <- Smlar$c19ProSo01 %>% createDataPartition(p = 0.8, list = FALSE)
Smtrain.data  <- Smlar[Smtraining.samples, ]
Smtest.data <- Smlar[-Smtraining.samples, ]
# fit into linear regression model
# strongest predictors: c19ProSo02, c19ProSo03, trustGovState
# p-value: < 2.2e-16
# Multiple R-squared:  0.3502
Sm1fit = lm(c19ProSo01 ~ ., data = Smtrain.data)
summary(Sm1fit)
```

```
lm(formula = c19ProSo01 ~ ., data = Smtrain.data)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.2245 -0.5174  0.1147  0.6338  3.6000 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -0.749374   0.432075  -1.734  0.08315 .  
affAnx        -0.043118   0.047954  -0.899  0.36878    
affBor        -0.001738   0.029019  -0.060  0.95225    
affCalm        0.102556   0.049578   2.069  0.03883 *  
affContent    -0.045609   0.049102  -0.929  0.35318    
affDepr       -0.021073   0.041180  -0.512  0.60894    
affEnerg       0.059396   0.043984   1.350  0.17719    
affExc        -0.043239   0.042394  -1.020  0.30800    
affNerv        0.016141   0.046938   0.344  0.73100    
affExh         0.025390   0.035402   0.717  0.47342    
affInsp        0.050469   0.041619   1.213  0.22554    
affRel        -0.004505   0.049552  -0.091  0.92758    
PLRAC19        0.038485   0.037980   1.013  0.31116    
PLRAEco        0.074629   0.027508   2.713  0.00678 ** 
disc01        -0.070123   0.047190  -1.486  0.13758    
disc03        -0.025821   0.044461  -0.581  0.56153    
jbInsec01     -0.016497   0.057220  -0.288  0.77317    
jbInsec02      0.006680   0.053131   0.126  0.89997    
jbInsec03     -0.011403   0.039855  -0.286  0.77486    
jbInsec04     -0.074518   0.058272  -1.279  0.20125    
employstatus   0.035381   0.039781   0.889  0.37400    
PFS01          0.006553   0.051622   0.127  0.89901    
PFS02         -0.027371   0.046076  -0.594  0.55261    
PFS03          0.077613   0.048335   1.606  0.10864    
fail01        -0.018819   0.039700  -0.474  0.63558    
fail02        -0.024370   0.043074  -0.566  0.57168    
fail03        -0.013039   0.043743  -0.298  0.76570    
happy          0.048659   0.029618   1.643  0.10071    
lifeSat       -0.109407   0.048766  -2.244  0.02507 *  
MLQ            0.013966   0.031553   0.443  0.65814    
c19NormShould -0.052948   0.054242  -0.976  0.32922    
c19NormDo      0.027505   0.042665   0.645  0.51928    
c19IsStrict    0.036063   0.034844   1.035  0.30092    
c19IsPunish    0.012825   0.027171   0.472  0.63702    
c19IsOrg       0.017996   0.034835   0.517  0.60553    
trustGovCtry  -0.034537   0.037166  -0.929  0.35297    
trustGovState  0.126440   0.044247   2.858  0.00435 ** 
gender         0.053682   0.068609   0.782  0.43413    
age            0.009453   0.027187   0.348  0.72815    
edu           -0.015577   0.024878  -0.626  0.53137    
c19ProSo02     0.292488   0.027126  10.782  < 2e-16 ***
c19ProSo03     0.226015   0.025619   8.822  < 2e-16 ***
c19ProSo04     0.081404   0.040623   2.004  0.04534 *  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.031 on 1040 degrees of freedom
Multiple R-squared:  0.3502,	Adjusted R-squared:  0.3239 
F-statistic: 13.34 on 42 and 1040 DF,  p-value: < 2.2e-16
```

plot(Sm1fit)



# make a predict using the test data
Sm1pred <- Sm1fit %>% predict(Smtest.data)
# RMSE: 1.063085
RMSE(Sm1pred, Smtest.data$c19ProSo01)

```
> RMSE(Sm1pred, Smtest.data$c19ProSo01)
[1] 1.063085
```

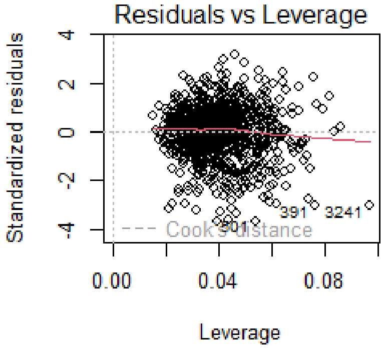
# R-square: 0.2585837
R2(Sm1pred, Smtest.data$c19ProSo01)

```
> R2(Sm1pred, Smtest.data$c19ProSo01)
[1] 0.2585837
```

# predict --> c19ProSo02
# fit into linear regression model
# strongest predictors: c19ProSo01, c19ProSo03, c19ProSo04
# p-value: < 2.2e-16
# Multiple R-squared: 0.3872
Sm2fit = lm(c19ProSo02 ~ ., data = Smtrain.data)
summary(Sm2fit)

```
Call:
lm(formula = c19ProSo02 ~ ., data = Smtrain.data)

Residuals:
    Min      1Q  Median      3Q     Max 
-4.0166 -0.5723  0.1281  0.6686  3.4685 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)  -0.5189700  0.4688261  -1.107  0.26857    
affAnx       -0.0015640  0.0520090  -0.030  0.97602    
affBor        0.0664117  0.0313927   2.116  0.03462 *  
affCalm      -0.0821880  0.0537990  -1.528  0.12689    
affContent    0.0215175  0.0532509   0.404  0.68624    
affDepr      -0.0005622  0.0446505  -0.013  0.98996    
affEnerg      0.0279468  0.0477186   0.586  0.55823    
affExc        0.0612554  0.0459446   1.333  0.18274    
affNerv       0.0159524  0.0508872   0.313  0.75397    
affExh       -0.0228058  0.0383831  -0.594  0.55253    
affInsp       0.0599100  0.0451137   1.328  0.18448    
affRel       -0.1291314  0.0535719  -2.410  0.01611 *  
PLRAC19      -0.0225503  0.0411899  -0.547  0.58417    
PLRAEco      -0.0489480  0.0298888  -1.638  0.10179    
disc01        0.1152270  0.0510891   2.255  0.02431 *  
disc03       -0.0239495  0.0482032  -0.497  0.61940    
jbInsec01     0.0678468  0.0620004   1.094  0.27408    
jbInsec02     0.0347915  0.0575907   0.604  0.54590    
jbInsec03     0.0260006  0.0432026   0.602  0.54742    
jbInsec04    -0.0312194  0.0632166  -0.494  0.62152    
employstatus  0.0110381  0.0431433   0.256  0.79812    
PFS01        -0.0454098  0.0559480  -0.812  0.41718    
PFS02        -0.0565995  0.0499305  -1.134  0.25724    
PFS03        -0.0208682  0.0524622  -0.398  0.69088    
fail01       -0.0590547  0.0430053  -1.373  0.16999    
fail02       -0.0079776  0.0467046  -0.171  0.86441    
fail03       -0.0195610  0.0474208  -0.412  0.68006    
happy        -0.0222781  0.0321441  -0.693  0.48842    
lifeSat       0.1304133  0.0528423   2.468  0.01375 *  
MLQ           0.0370366  0.0341920   1.083  0.27897    
c19NormShould -0.0229967 0.0588283  -0.391  0.69594    
c19NormDo     0.0479028  0.0462399   1.036  0.30046    
c19IsStrict   0.0077944  0.0377945   0.206  0.83665    
c19IsPunish  -0.0034670  0.0294596  -0.118  0.90634    
c19IsOrg     -0.0598896  0.0377247  -1.588  0.11269    
trustGovCtry  0.0242113  0.0403024   0.601  0.54814    
trustGovState 0.0866565  0.0480826   1.802  0.07180 .  
gender       -0.0454027  0.0743892  -0.610  0.54177    
age           0.0738446  0.0293870   2.513  0.01213 *  
edu           0.0002468  0.0269762   0.009  0.99270    
c19ProSo01    0.3437723  0.0318826  10.782  < 2e-16 ***
c19ProSo03    0.2832476  0.0274230  10.329  < 2e-16 ***
c19ProSo04    0.1299306  0.0439413   2.957  0.00318 ** 
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.118 on 1040 degrees of freedom
Multiple R-squared:  0.3872,	Adjusted R-squared:  0.3624 
F-statistic: 15.65 on 42 and 1040 DF,  p-value: < 2.2e-16
```

plot(Sm2fit)



# make a predict using the test data
Sm2pred <- Sm2fit %>% predict(Smtest.data)
# RMSE: 1.136873
RMSE(Sm2pred, Smtest.data$c19ProSo02)

```
> RMSE(Sm2pred, Smtest.data$c19ProSo02)
[1] 1.117186
```

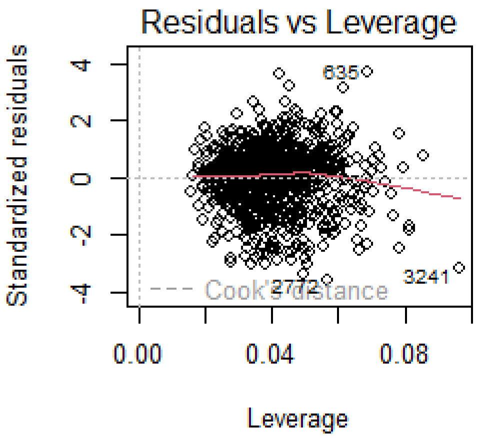
# R-square: 0.3489879
R2(Sm2pred, Smtest.data$c19ProSo02)

```
> R2(Sm2pred, Smtest.data$c19ProSo02)
[1] 0.3416506
```

# predict --> c19ProSo03
# fit into linear regression model
# strongest predictors: c19ProSo01, c19ProSo02, c19ProSo04
# p-value: < 2.2e-16
# Multiple R-squared: 0.3524
Sm3fit = lm(c19ProSo03 ~ ., data = Smtrain.data)
summary(Sm3fit)

```
Residuals:
    Min      1Q  Median      3Q     Max 
-4.1621 -0.6203  0.1707  0.7091  4.2970 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -0.298070   0.505077  -0.590  0.55522    
affAnx         0.030535   0.055999   0.545  0.58567    
affBor        -0.043592   0.033852  -1.288  0.19812    
affCalm        0.035563   0.057989   0.613  0.53983    
affContent    -0.019429   0.057346  -0.339  0.73482    
affDepr        0.063201   0.048043   1.316  0.18863    
affEnerg      -0.006724   0.051395  -0.131  0.89594    
affExc         0.028789   0.049511   0.581  0.56105    
affNerv       -0.074194   0.054753  -1.355  0.17569    
affExh        -0.001911   0.041341  -0.046  0.96315    
affInsp        0.011237   0.048621   0.231  0.81727    
affRel         0.012039   0.057850   0.208  0.83519    
PLRAC19        0.141284   0.044146   3.200  0.00141 ** 
PLRAEco        0.008485   0.032227   0.263  0.79238    
disc01        -0.063304   0.055116  -1.149  0.25100    
disc03        -0.017871   0.051912  -0.344  0.73073    
jbInsec01     -0.092426   0.066743  -1.385  0.16641    
jbInsec02     -0.054024   0.062006  -0.871  0.38381    
jbInsec03     -0.001389   0.046532  -0.030  0.97619    
jbInsec04      0.044496   0.068070   0.654  0.51346    
employstatus  -0.062087   0.046421  -1.337  0.18136    
PFS01         -0.102258   0.060184  -1.699  0.08960 .  
PFS02         -0.013767   0.053800  -0.256  0.79808    
PFS03         -0.010837   0.056498  -0.192  0.84793    
fail01        -0.046141   0.046331  -0.996  0.31953    
fail02         0.065078   0.050255   1.295  0.19563    
fail03         0.014930   0.051068   0.292  0.77008    
happy         -0.020173   0.034617  -0.583  0.56019    
lifeSat       -0.071160   0.057028  -1.248  0.21238    
MLQ            0.051523   0.036806   1.400  0.16186    
c19NormShould  0.011554   0.063354   0.182  0.85532    
c19NormDo     -0.039614   0.049805  -0.795  0.42657    
c19IsStrict   -0.046651   0.040675  -1.147  0.25167    
c19IsPunish    0.021753   0.031717   0.686  0.49297    
c19IsOrg       0.052245   0.040641   1.286  0.19890    
trustGovCtry  -0.005666   0.043408  -0.131  0.89617    
trustGovState -0.018894   0.051856  -0.364  0.71567    
gender         0.054150   0.080104   0.676  0.49920    
age           -0.045566   0.031710  -1.437  0.15104    
edu            0.037310   0.029027   1.285  0.19895    
c19ProSo01     0.308053   0.034919   8.822  < 2e-16 ***
c19ProSo02     0.328467   0.031801  10.329  < 2e-16 ***
c19ProSo04     0.241514   0.046924   5.147 3.16e-07 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 1.203 on 1040 degrees of freedom
Multiple R-squared:  0.3524,	Adjusted R-squared:  0.3262 
F-statistic: 13.47 on 42 and 1040 DF,  p-value: < 2.2e-16
```

plot(Sm3fit)



# make a predict using the test data
Sm3pred <- Sm3fit %>% predict(Smtest.data)
# RMSE: 1.292243
RMSE(Sm3pred, Smtest.data$c19ProSo03)

```
> RMSE(Sm3pred, Smtest.data$c19ProSo03)
[1] 1.096615
```

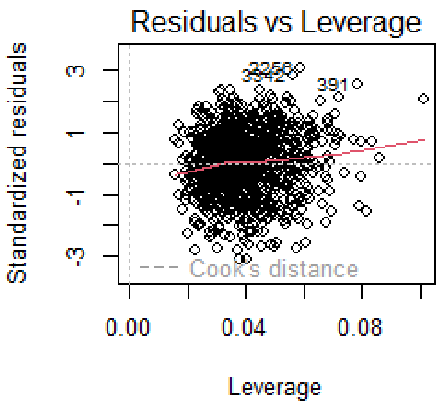
# R-square: 0.3246896
R2(Sm3pred, Smtest.data$c19ProSo03)

```
> R2(Sm3pred, Smtest.data$c19ProSo03)
[1] 0.3709666
```

# predict --> c19ProSo04
# fit into linear regression model
# strongest predictors: c19NormShould, c19ProSo03, c19IsPunish
# p-value: < 2.2e-16
# Multiple R-squared: 0.3424
Sm4fit = lm(c19ProSo04 ~ ., data = Smtrain.data)
summary(Sm4fit)

```
Residuals:
     Min       1Q   Median       3Q      Max 
-2.36832 -0.46450  0.02424  0.54204  2.35628 

Coefficients:
               Estimate Std. Error t value Pr(>|t|)    
(Intercept)   -0.334934   0.329492  -1.017  0.30962    
affAnx         0.051884   0.036513   1.421  0.15563    
affBor         0.014074   0.022104   0.637  0.52445    
affCalm        0.054401   0.037811   1.439  0.15052    
affContent     0.022488   0.037418   0.601  0.54797    
affDepr       -0.062512   0.031318  -1.996  0.04619 *  
affEnerg      -0.064070   0.033480  -1.914  0.05594 .  
affExc        -0.058925   0.032263  -1.826  0.06808 .  
affNerv        0.005974   0.035762   0.167  0.86735    
affExh         0.010132   0.026976   0.376  0.70731    
affInsp       -0.032897   0.031714  -1.037  0.29983    
affRel         0.015544   0.037749   0.412  0.68059    
PLRAC19        0.010302   0.028948   0.356  0.72201    
PLRAEco        0.038151   0.020998   1.817  0.06952 .  
disc01         0.082957   0.035898   2.311  0.02103 *  
disc03        -0.007682   0.033877  -0.227  0.82066    
jbInsec01     -0.001496   0.043595  -0.034  0.97262    
jbInsec02      0.066059   0.040426   1.634  0.10255    
jbInsec03      0.038021   0.030342   1.253  0.21046    
jbInsec04     -0.065314   0.044384  -1.472  0.14144    
employstatus   0.056944   0.030268   1.881  0.06020 .  
PFS01         -0.077390   0.039256  -1.971  0.04894 *  
PFS02          0.100437   0.034971   2.872  0.00416 ** 
PFS03          0.006586   0.036869   0.179  0.85826    
fail01        -0.069408   0.030172  -2.300  0.02162 *  
fail02        -0.056173   0.032775  -1.714  0.08685 .  
fail03         0.071517   0.033253   2.151  0.03173 *  
happy         -0.010897   0.022591  -0.482  0.62965    
lifeSat        0.041493   0.037221   1.115  0.26519    
MLQ           -0.023369   0.024031  -0.972  0.33104    
c19NormShould  0.375458   0.039671   9.464  < 2e-16 ***
c19NormDo     -0.027281   0.032500  -0.839  0.40142    
c19IsStrict    0.061312   0.026492   2.314  0.02084 *  
c19IsPunish   -0.088225   0.020521  -4.299 1.87e-05 ***
c19IsOrg       0.047691   0.026501   1.800  0.07222 .  
trustGovCtry  -0.045437   0.028292  -1.606  0.10858    
trustGovState  0.109509   0.033671   3.252  0.00118 ** 
gender        -0.004485   0.052285  -0.086  0.93166    
age            0.017014   0.020707   0.822  0.41146    
edu           -0.042487   0.018911  -2.247  0.02487 *  
c19ProSo01     0.047249   0.023579   2.004  0.04534 *  
c19ProSo02     0.064165   0.021700   2.957  0.00318 ** 
c19ProSo03     0.102849   0.019983   5.147 3.16e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.7854 on 1040 degrees of freedom
Multiple R-squared:  0.3424,	Adjusted R-squared:  0.3159 
F-statistic: 12.89 on 42 and 1040 DF,  p-value: < 2.2e-16
```

plot(Sm4fit)



# make a predict using the test data
Sm4pred <- Sm4fit %>% predict(Smtest.data)
# RMSE: 0.7941543
RMSE(Sm4pred, Smtest.data$c19ProSo04)

```
> RMSE(Sm4pred, Smtest.data$c19ProSo04)
[1] 0.7941543
```

# R-square: 0.2800452
R2(Sm4pred, Smtest.data$c19ProSo04)

```
> R2(Sm4pred, Smtest.data$c19ProSo04)
[1] 0.2800452
```

# For c19ProSo01,
# Focus country (Greece):
# strongest predictors: c19ProSo02, PLRAC19, affBor, affBor
# Similar COuntries:
# strongest predictors: c19ProSo02, c19ProSo03, trustGovState
summary(Sm1fit)
# All other countries:
# strongest predictors: c19ProSo02, c19ProSo03, c19ProSo04
# In my opinion for attribute 'c19ProSo01', 'c19ProSo02' is the most important indicator for prediction
# because it appeared as the most significant in all models.
# For both group of All other countries 2(c) and Similar COuntries 3(b) give a similar match
# to the important attributes for predicting pro-social attitudes in my focus country
# because they 2 only have 1 same indicator with the my country model.

For c19ProSo02,
Focus country (Greece):
strongest predictors: c19ProSo01, c19ProSo03, c19IsStrict, c19NormDo, c19IsPunish
Similar COuntries:
strongest predictors: c19ProSo01, c19ProSo03, c19ProSo04
All other countries:
strongest predictors: c19ProSo03, c19ProSo01
In my opinion for attribute 'c19ProSo02', 'c19ProSo01' and 'c19ProSo03'is the
most important indicator for prediction because it appeared as the most
significant in all models.
All other countries 2(c) give a similar match to the important attributes for
predicting pro-social attitudes in my focus country.
This is because although they have the same similar very significant indicators,
but for model of All Countries have 'c19NormDo' as significant indicator and
for similar countries have it as not significant indicator.

```
# For c19ProSo03,
# Focus country (Greece):
# strongest predictors: c19NormDo, edu, lifeSat, c19ProSo04
# Similar COuntries:
# strongest predictors: c19ProSo01, c19ProSo02, c19ProSo04
# All other countries:
# strongest predictors: c19ProSo01, c19ProSo02, c19ProSo04
# In my opinion for attribute 'c19ProSo03', 'c19ProSo04' is the
# most important indicator for prediction because it appeared as the most
# significant in all models.
# All other countries 2(c) give a similar match to the important attributes for
# predicting pro-social attitudes in my focus country.
# This is because although they have the same similar very significant indicators,
# but for model of All Countries have 'c19NormDo' as significant indicator and
# for similar countries have it as not significant indicator.
```

```
# For c19ProSo04,
# Focus country (Greece):
# strongest predictors: c19ProSo03, c19NormShould
# Similar COuntries:
# strongest predictors: c19NormShould, c19ProSo03, c19IsPunish
# All other countries:
# strongest predictors: c19ProSo01, c19ProSo03, c19NormShould
# In my opinion for attribute 'c19ProSo04', 'c19ProSo03' and 'c19NormShould' is
# the most important indicator for prediction because it appeared as the most
# significant in all models.
# Similar COuntries 3(b) give a similar match to the important attributes for
# predicting pro-social attitudes in my focus country.
# This is because although they have the same similar very significant indicators,
# but for model of Similar COuntries have focus on the 2 indicators as significant
# indicator and for All other countries 2(c) have other significant indicator for prediction.
```