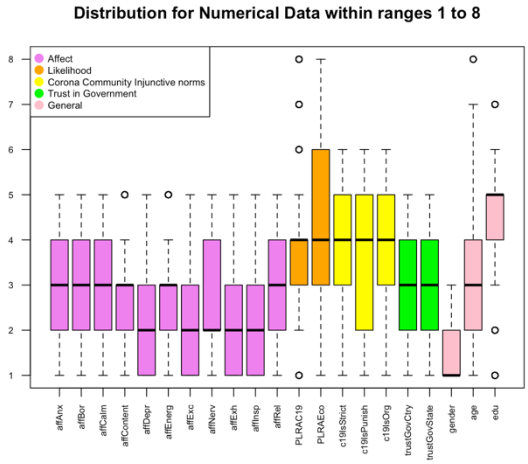
Section 1: Descriptive analysis and pre-processing

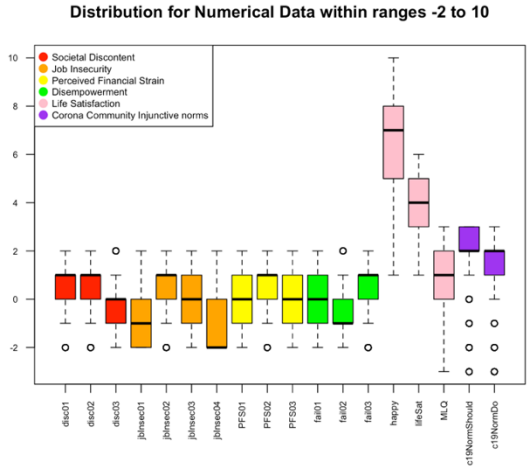
Question 1 (a) – Describe data

PsyCoronaBaselineExtract excel file has a dimension of 40,000 rows and 54 columns. Out of the 54 columns, there is only "coded_country" that's of character data type while the rest are integer data type.

There are 53 columns of numerical attributes, they are being split into 4 separate plots for a better view. The first graph is a box plot for attributes ranging between 1 to 8 where they are colour coded by concept for better viewing.

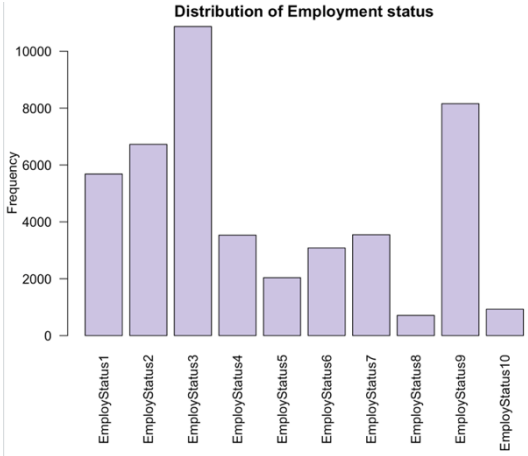


Most of the affect responses have a median of 2 and 3 meaning they feel "a little" or "moderately" about the listed feelings. For Corona Community Injunctive norms, the all have a median of where they feel neutral about the situation. With ages having median of 3, this indicates that most of the participants fall under the age of 35 – 44.



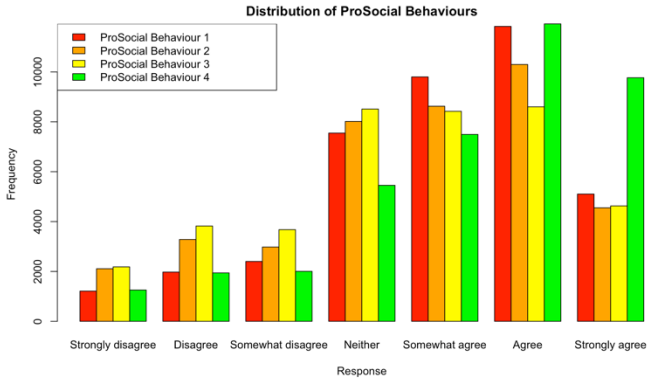
The same is done for numerical data within the range of -2 to 10.

According to medians obtained from the box plots, the participants are quite discontent with society. They are neutral when it comes to the future of their jobs, most of them are keeping their jobs despite the pandemic and are happy with their current lives. For disempowerment, the participants are neutral for the help they are getting and they agree that the people in their areas are social distancing.



For frequency of Employment Status in figure 1.3 in appendix, most of the participants worked 40 hours or more in the last week and a large fraction of them are also students.

For the distribution of pro-social behaviours, majority of the participants agree with the fourth pro-social behaviour statements the most. For prosocial behaviour 3, most participants fall between neutral, somewhat agree and agree when it comes to helping others in their own expense. But for the three other statements, it can be seen that participants mostly agree with them.



The percentage of how much NA takes up in each column can be calculated by getting the total number of NA in each column, dividing it by the number of rows and *100 to get the percentage. From figure 1.0, we can see that employment status columns, especially employstatus_8 has the most NA counts with 98.2% of the data being NA. But this is normal since it's a tick only if applicable questions, in this case, the NA are just equivalent to 0.

Aside from that, from a numerical point of view, there are 36 rows where all of the attributes, except coded_country is left empty meaning NA.

There are 110 unique values in the column coded_country. This indicates that there are 109 different countries taken part in this survey. It is 109 instead of 110 because one distinct value belongs to "" empty string counted in due to unfilled responses.

For the column coded_country, there are some participants who did no fill in the field resulting in "" (empty string) for some of the rows. There are 141 rows where the country field is left empty.

Is.na counts the number of NA's in the row, while ncol is the total number of columns, but since we are excluding coded_country because even if participants did not fill, it is "" which is not NA so it doesn't affect the result, which explains the -1.

Question 1 (b) – Manipulating data

Firstly, we remove the rows where all data except coded_country is NA. The number of rows have now been reduced from 40,000 to 39,964.

Since this report is about predicting pro-social behaviours, responses where all pro-social behaviour questions are left empty should be omitted.
Now we are left with 39889 rows.

Since Employment status is a yes or no question, Yes = 1 and No are NA because participants did not select a value, we can replace the NAs with 0 instead.

The same goes for jbInsec columns but here we can only use -3 for not applicable since 0 is already used to represent "Neither agree nor disagree"

Section 2: Focus country vs all other countries as a group

Question 2 (a) – Difference in responses

The focus country for this report is Malaysia. The cvbase data frame is separated into two data frames namely MsiaCvbase and OthersCvbase. MsiaCvbase contains only rows where coded_country is Malaysia while the rest of the rows goes to OthersCvbase.

Coded_country column can now be removed since they are already separated by country in different data frames. There are 569 records in MsiaCvbase and 39320 records for OthersCvbase.

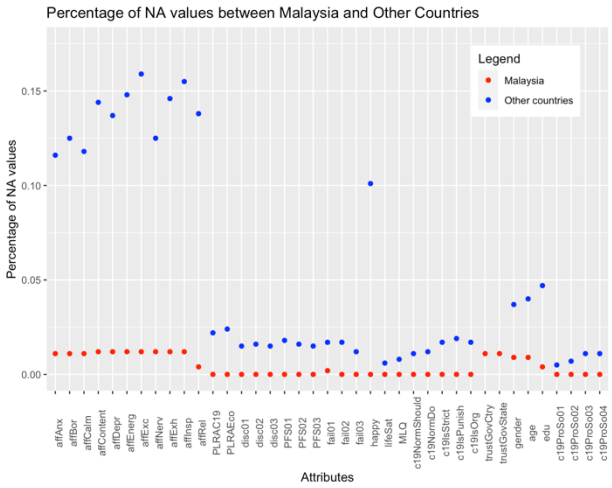
Difference in responses (NA)

For the percentage of NA's in each column for Malaysia, it is calculated by counting the number of NA's and dividing it with the total number of records. It is then rounded to 3 decimal places for a neater view. The same goes for other countries.

Both data frames are then bound for comparison. Employment Statuses and Job Insecurity are excluded since there are no more NA's in those columns after data manipulation.

A summary of the NAcomparision table shows that there are 39 out of 39 columns where the percentage of NA in Malaysia is lower than other countries. Malaysia participants have a better response rate compared to other countries.

It can be clearly seen in the figure, participants who answered Malaysia, the dots in red, have a lower NA percentage than other countries

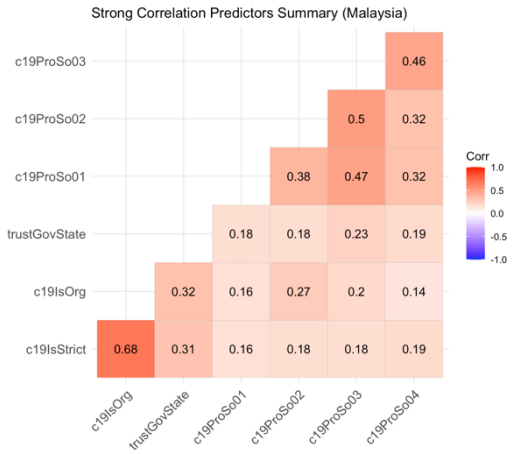


Question 2 (b) – Predictors for Malaysia

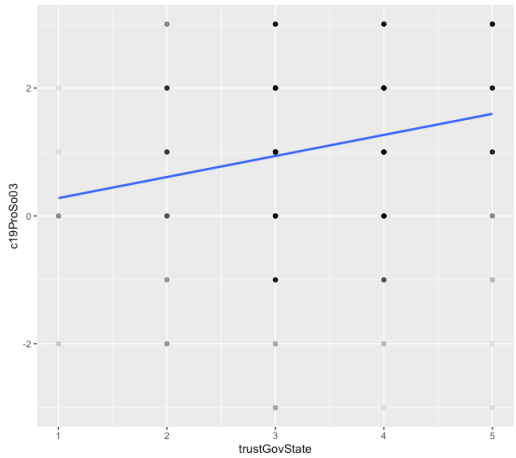
Pearson correlation coefficient is a way to measure linear correlation. It's calculates from a range of -1 to 1 where it measures the strength and direction of the relationship between two variables which in this case is pro-social behaviours against other survey questions. The more extreme to the end of the ranges, the stronger the correlation. There is one limitation of coefficient is that it doesn't work for NA values therefore we have to remove them.

From doing correlation for all pro-socials individually and finding extremes, it can be concluded that trustGovState, c19IsOrg and c19IsStrict have the strongest correlation.

It can be visualised using a correlation matrix where the darker the colour the stronger the correlation. Since we're only interested in the correlation of other attributes against behaviours, we only need to look at the 3 rows from the bottom.



From the correlation matrix, we can see that c19ProSo03 has the highest correlation with trustGovState and it can be represented in linear plot. The darker the points, the more dense it is.



Aside from correlations, we can use linear regression to get p-values and r squared values to indicate strong predictors.

<u>Linear regression to identify strong predictors for c19ProSo01</u>

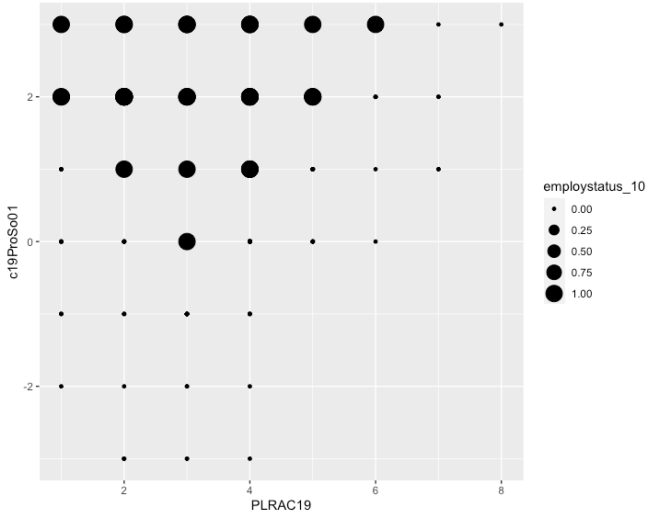
In general, an alpha of 0.05 is used for significance so these are considered strong predictors because their p-value is under 0.05 which means we reject the null hypothesis that there's no difference between the means and conclude that a significant difference does exist between that predictor and that pro-social behaviour.

R squared value is the coefficient of determination, it's also the strength of the relationship, here it's approximately 0.14 which means 14% of the variability in the target variable is explained by predictors of the model.

Having an R squared value of 0.1439 is really low, there were multiple failed attempts to increase it such as removing predictors with highest correlation but the R squared value reduced to 0.1375 instead.

The second way was to remove predictors with low p-value, 5 of the lowest p-value predictors were removed but the R squared value still stayed the same and the 6th lowest predictor would make the R squared value drop to 0.1438.

The strongest predictors for pro-social behaviour 1 are employment status 10 and PLRAC19

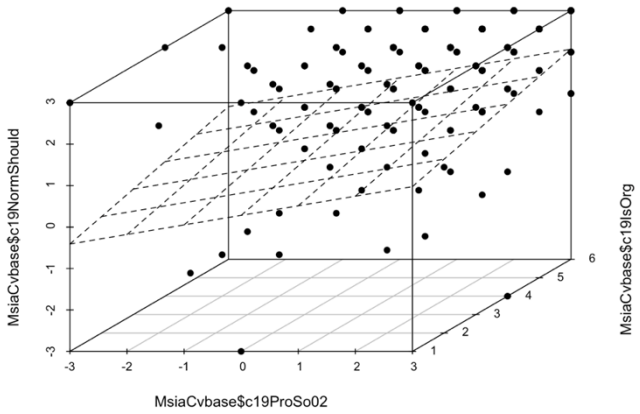


Their relationship can be visualised using a plot. A slight pattern can be seen from the qplot. If it's a big point, the participant is a volunteer, and they are more saturated at the higher scale of c19ProSo01, this means volunteers are more willing help others from coronavirus. They also think it's less likely to get coronavirus which makes them more willing to help

Linear regression is used to identify strong predictors for c19ProSo02

Strong predictors for ProSocial02 are c19IsOrg and c19NormShould. The R squared value is approximately 0.22 which means 22% of the variability in the target variable is explained by predictors of the model.

The two strong predictors were selected to build a scatter 3d plot. We can see that they are very saturated at the top right of the cube. This means participants are more likely to agree with the statement of making donations, if their community is organized in response to covid and participants agree the people around them should social distance.



Linear regression is used to identify strong predictors for c19ProSo03

Strong predictors for ProSocial03 are trustGovState 0.0033 and PFS01 0.0501

There aren't many strong predictors for c19ProSo03 and the second strongest one is PFS01 which is 0.0501, any predictor more than 0.05 is not considered such a strong predictor.

The R squared value is approximately 0.14 which means 14% of the variability in the target variable is explained by predictors of the model.

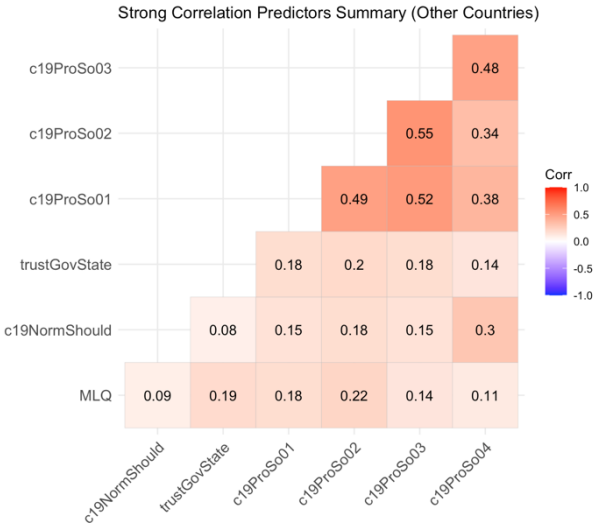
Linear regression is used to identify strong predictors for c19ProSo04

Strong predictors for ProSocial04 are c19IsStrict and disc01.

The R squared value is approximately 0.14 which means 14% of the variability in the target variable is explained by predictors of the model which is same as previous.

Question 2 (c) – Predictors for Other countries

We can see that from figure 2.6's correlation matrix for Other countries, participants clear sense of purpose in life, Corona Community Injunctive norms and amount of trust in government is frequently highly correlated with pro-social responses.



Comparison

For correlation, both Malaysia and other countries strongest correlation for ProSocial01 and ProSocial03 are the same. But for ProSocial02, Malaysia is more correlated with how organised is community responding. For ProSocial04, Other countries have a higher correlation with c19Normshould

Linear regression is used to identify strong predictors for c19ProSo01

Strong predictors for ProSocial01: trustGovState and c19NormShould

The R squared value is approximately 0.11 which means 11% of the variability in the target variable is explained by predictors of the model.

COMPARISON : The R squared value is a lot lower than the one compared to Malaysia which is 0.14. Making Malaysia responses more accurate at predicting c19ProSo01

Linear regression is used to identify strong predictors for c19ProSo02

Strong predictors for ProSocial02: c19NormShould and MLQ

The R squared value is approximately 0.15 which means 15% of the variability in the target variable is explained by predictors of the model.

COMPARISON : The R squared value is lower than the one compared to Malaysia which is 0.22. Making Malaysia responses more accurate at predicting c19ProSo02

Linear regression is used to identify strong predictors for c19ProSo03

Strong predictors for ProSocial03: trustGovState and c19NormShould

The R squared value is approximately 0.11 which means 11% of the variability in the target variable is explained by predictors of the model.

COMPARISON : The R squared value is lower than the one compared to Malaysia which is 0.14. Making Malaysia responses more accurate at predicting c19ProSo03

Linear regression is used to identify strong predictors for c19ProSo04

Strong predictors for ProSocial04: c19NormShould and disc02

The R squared value is approximately 0.15 which means 15% of the variability in the target variable is explained by predictors of the model.

COMPARISON : The R squared value is higher than the one compared to Malaysia which is 0.14. Making Other countries responses more accurate at predicting c19ProSo04

In summary, all the R squared values recorded for other countries predicting the pro-social behaviours are lower than Malaysia except pro-social behaviour 4, this makes Malaysia's participant responses more accurate in predicting their own pro-social behaviours than other countries predicting theirs.

Using other countries responses to predict pro-social behaviours in Malaysia

Aside from comparing their accuracy of prediction with their own pro-socials, we can use Other countries as the training dataset and Malaysia for testing. This way we can see how well attributes from Other countries can predict pro-social behaviours in Malaysia by checking the number of correct predictions using confusion matrix.

For pro-social 1, there are 177 correct matches, For pro-social 2, there are 175 correct matches.

For pro-social 3, there are 157 correct matches, For pro-social 4, there are 190 correct matches.

	-3	-2	-1	0	1	2	3
-3	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	2	3	9	12	8	5
1	0	1	17	53	72	118	28
2	6	2	1	19	41	96	57
3	0	0	0	0	0	0	0

	-3	-2	-1	0	1	2	3
-3	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	3	4	9	9	10	4
1	3	4	6	37	69	118	52
2	2	2	2	11	36	97	72
3	0	0	0	0	0	0	0

	-3	-2	-1	0	1	2	3
-3	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	3	3	16	14	10	7
1	0	0	0	0	0	0	0
2	6	12	36	96	133	141	73
3	0	0	0	0	0	0	0

	-3	-2	-1	0	1	2	3
-3	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0
-1	0	0	0	0	0	0	0
0	0	3	3	16	14	10	7
1	0	0	0	0	0	0	0
2	6	12	36	96	133	141	73
3	0	0	0	0	0	0	0

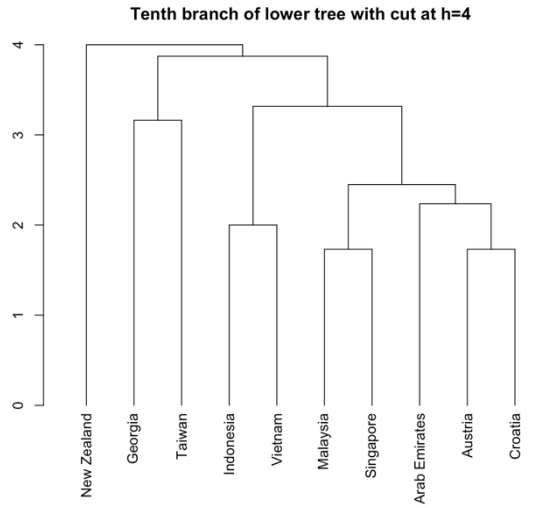
Section 3: Clustering

Question 3 (a) – Cluster similar countries

For the table of values used for clustering, 14 indicators are chosen due to their relation to society and government because this two aspect played the largest part by frequently being a strong predictor in the previous linear regressions. The indicators are PLRAC19, Societal Discontent, Disempowerment, Corona Community Injunctive Norms and Trust in Government.

The table of values is generated by the dataset country_indicator. When creating country_indicator, grouping them by country is necessary as it is to let each country be represented individually using their mode which allows us to cluster them later on.

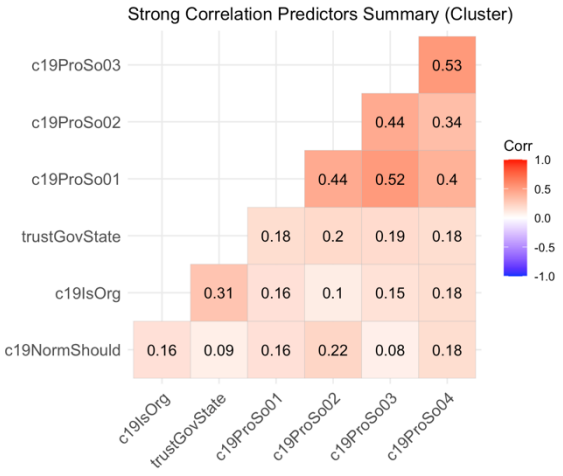
To identify similar countries, hierarchical clustering, more specifically complete linkage, is used where it joins clusters based on the maximum distance between any pair of results in the two clusters. The maximum distance can be computed using the dist(). Here, the distance matrix is computed using the Euclidean matrix.



Since there are a lot of countries, to get a better view, we can choose to plot a specific branch. The branch with Malaysia is selected which is the tenth branch and countries under the same section are Indonesia, Vietnam, Singapore, United Arab Emirates, Austria and Croatia.

Question 3 (b) – Predictors for cluster

In correlation matrix for Cluster countries, participants trustGovState, c19IsOrg and c19NormShould is frequently highly correlated with pro-social responses.



Comparison

For correlation, both Malaysia and other countries strongest correlation for ProSocial01 and ProSocial03 are the same. Overall, the values here are higher meaning attributes here are more strongly correlated.

By using linear regression, it would result the followings:

For c19ProSo01 : R squared value = 0.1218
    Strong Predictors are disc02 and c19NormShould
For c19ProSo02 : R squared value = 0.1943
    Strong Predictors are c19NormShould and trustGovState
For c19ProSo03 : R squared value = 0.1092
    Strong Predictor is trustGovState and PLRAC19
For c19ProSo04 : R squared value = 0.1179
    Strong Predictors are c19NormShould and trustGovState

With an R squared value of range 0.11 to 0.19, this means roughly only 11% to 19% of the variance found in pro-socials can be explained but the rest of the attributes which is not that good.

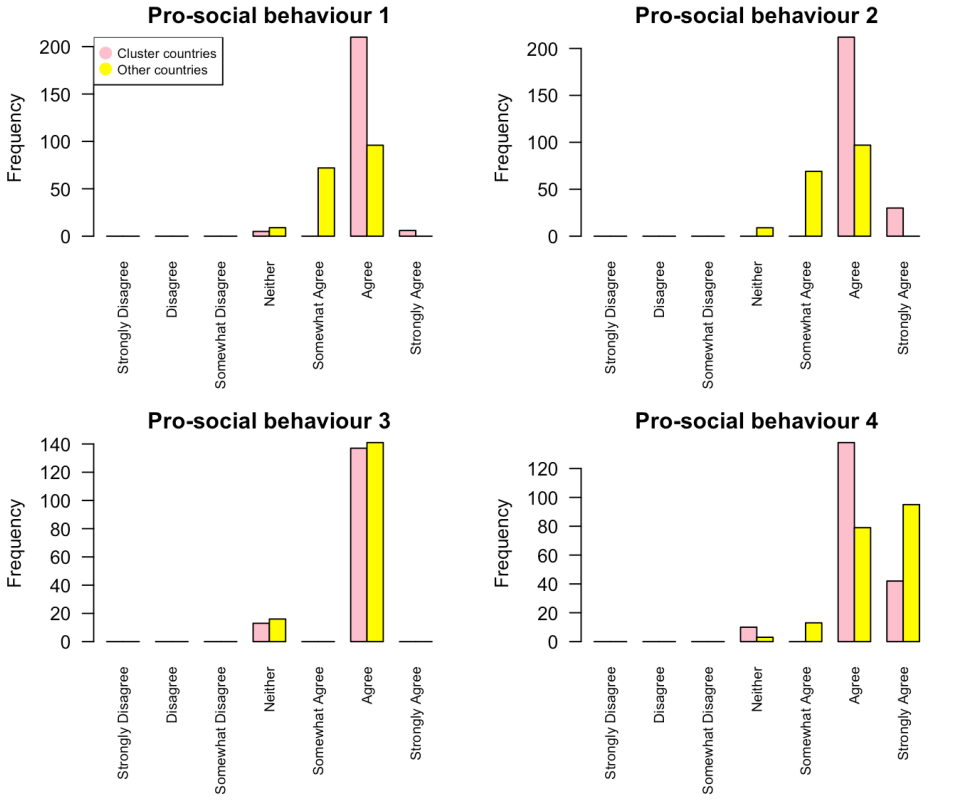
Although each pro-social regression has their own strong predictors, c19NormShould seems to play a big role as it has the lowest p-values among pro-social behaviours 1,2 and 4.

On the other hand, all of the R squared values for predicting the 4 pro-social behaviours in cluster is lower when compared to both both Malaysia and Other countries, this means this cluster of countries are worse than Malaysia and all the other countries at predicting their own pro-social behaviours linearly.

The better dataset to be used to predict Malaysia's pro-social behaviours can be concluded by using rpart function which means recursive partitioning as it works best for predicting factors which in this case, our pro-social behaviours. This is preferred instead of linear regression because linear regression returns a numerical value recursive partitioning returns predictions as value from the scale. The predicted values are then compared with the actual values from Malaysia's Cvbase.

We can see that clustered countries are better at predicting Malaysia's pro-social behaviours because they have higher correct matches which is taken by calculating the sum of values along the diagonal line. I can be clearly seen for behaviours 1 and 2 as there is a drastic difference. For behaviour 3 is slightly lower by 7 matches and for behaviour 4 they are the same. In total Cluster countries predicted 798 correct values while other countries only 699. Making clustered countries a more reliable dataset to predict the pro-social behaviours in Malaysia.

For c19ProSo01 : using cluster countries = 216 matches using other countries = 177 matches
For c19ProSo02 : using cluster countries = 242 matches using other countries = 175 matches
For c19ProSo03 : using cluster countries = 150 matches using other countries = 157 matches
For c19ProSo04 : using cluster countries = 190 matches using other countries = 190 matches



Appendix

coded_country	PLRAC19	disc01	disc02	disc03	fail01	fail02	fail03	c19NormShould	c19NormDo	c19IsStrict	c19IsPunish	c19IsOrg	trustGovCtry	trustGovState
Albania	4	2	2	-2	2	-1	2	2	-3	1	1	1	3	1
Algeria	4	1	1	-1	1	0	1	3	1	4	4	2	3	2
Argentina	4	1	1	0	0	0	1	3	3	6	5	4	4	3
Australia	4	1	1	0	-1	0	1	3	2	5	4	4	3	3
Austria	3	1	1	-1	-1	-2	1	3	2	5	4	5	4	4
Azerbaijan	8	0	0	0	1	-1	2	0	-2	1	6	1	3	3
Bahrain	3	1	1	0	-2	-2	-1	2	-2	4	4	3	3	3
Bangladesh	4	1	2	-2	1	1	1	3	3	3	2	3	2	2
Belarus	4	1	1	0	0	-1	-2	2	1	1	1	2	3	4
Belgium	4	1	1	-1	-1	-1	-1	3	2	5	3	5	3	3
Benin	2	0	0	0	-2	-2	1	3	-3	4	2	3	3	3
Bosnia and Herzegovina	3	1	1	-1	0	-2	1	3	1	5	4	3	3	3
Brazil	4	1	1	-1	0	-1	1	3	2	4	1	4	3	3
Brunei	5	1	1	0	-2	-1	0	3	3	6	6	5	5	4
Bulgaria	2	-1	1	1	-2	-2	-2	2	2	3	2	6	3	3
Cambodia	2	1	1	-2	1	0	0	0	-1	4	5	3	2	3
Canada	4	1	1	0	0	0	1	3	2	5	4	4	3	3
Chile	4	1	1	-1	1	0	1	3	3	4	3	4	1	3
Colombia	4	1	1	0	0	0	1	3	2	6	4	3	3	3
Costa Rica	2	-1	-1	0	0	0	0	3	2	5	4	3	5	4
Croatia	4	1	1	-1	0	-1	1	3	2	5	4	5	4	4
Cyprus	4	0	1	0	-1	0	1	3	2	5	5	4	4	3
Czech Republic	5	1	2	1	-1	0	-1	3	2	5	2	3	3	3
Denmark	1	1	1	2	-1	-1	-1	3	1	3	2	4	2	2
Dominican Republic	2	-1	-1	-2	0	1	1	3	3	3	2	2	3	3
Ecuador	3	1	2	0	0	-2	1	3	3	6	3	3	2	2
Egypt	4	1	1	-1	0	0	1	3	-3	3	2	1	3	2
El Salvador	4	1	1	0	-1	0	1	3	3	6	6	1	3	2
Estonia	4	1	1	-1	0	1	-1	2	0	4	3	3	3	4
Finland	4	1	1	0	-1	-2	1	3	-1	5	1	4	3	3
France	4	1	1	0	0	0	0	3	1	4	4	4	3	3
Georgia	2	0	0	1	0	-1	1	2	1	5	5	5	5	3
Germany	4	1	1	0	0	-1	1	2	2	4	4	4	4	3
Greece	4	1	1	0	0	-1	1	3	2	5	5	4	3	3
Guatemala	7	2	2	-1	-1	-1	0	3	-1	5	5	4	4	3
Hong Kong S.A.R.	3	1	1	-1	0	-1	1	2	1	4	4	4	1	3
Hungary	4	1	1	-1	0	-1	1	3	2	4	3	5	3	4
Iceland	3	-1	-1	1	-1	-1	-1	1	2	3	2	5	5	4
India	3	1	1	0	-1	-2	1	3	3	6	4	4	3	3
Indonesia	4	1	1	0	0	-1	1	3	2	6	4	6	4	3
Iran	5	1	1	-2	2	0	1	3	1	4	2	3	1	3
Iraq	2	1	1	1	-1	1	-2	2	-1	5	3	3	3	3
Ireland	3	1	1	0	0	-1	1	3	2	4	3	5	4	3
Israel	4	0	2	-1	-1	-1	1	3	2	5	5	5	4	3
Italy	4	1	1	0	0	0	1	3	2	5	4	4	3	3
Jamaica	4	1	0	0	-2	-1	1	3	1	6	6	5	3	2
Japan	4	1	1	0	0	0	0	1	1	3	3	3	3	3
Jordan	4	1	2	-1	0	-2	0	3	2	6	4	4	4	5
Kazakhstan	3	1	1	-1	-1	-1	1	3	1	5	3	4	3	3
Kenya	4	1	1	-1	0	-2	1	1	1	5	5	5	3	2
Kosovo	4	1	1	1	0	-1	1	2	2	4	4	4	3	3
Kuwait	5	2	1	-1	0	2	0	2	1	4	5	4	4	3
Lebanon	3	1	1	-1	1	-1	1	3	3	5	5	4	5	5
Lithuania	4	1	1	-1	-1	-2	-2	3	1	3	2	4	3	3
Luxembourg	4	-1	2	0	-2	-1	0	3	3	5	4	3	4	4
Malaysia	4	1	1	0	-1	-1	1	3	2	5	5	5	5	3
Mali	5	1	1	0	-1	-1	1	2	1	3	4	3	3	3
Malta	4	0	1	-2	-1	-1	1	2	-1	5	2	5	4	3
Mexico	4	1	1	-1	1	-1	2	3	2	1	1	4	1	3
Moldova	2	1	1	-1	1	-1	1	3	1	5	4	3	3	3
Montenegro	4	1	1	0	0	-1	1	3	3	6	5	4	4	3
Morocco	4	1	1	-1	1	1	0	2	2	4	4	5	3	3
Netherlands	4	1	1	0	-1	-1	1	2	2	4	4	5	4	3
New Zealand	3	0	2	0	1	0	1	3	3	6	5	6	4	3
Nigeria	2	1	2	0	2	1	1	3	2	6	6	2	2	2
Norway	3	1	-1	-1	-1	-1	1	2	1	3	4	5	3	3
Pakistan	1	1	1	0	0	-1	1	2	2	4	4	4	3	3
Panama	3	2	2	-1	0	2	1	3	3	4	3	4	1	1
Peru	4	1	1	-1	0	-1	1	3	3	4	2	3	3	3
Philippines	1	1	1	0	0	-1	1	3	3	6	4	4	4	3
Poland	4	1	1	-1	1	0	1	2	1	4	3	4	1	3
Portugal	5	1	1	-1	-1	-1	1	3	2	4	3	4	4	3
Qatar	4	2	2	0	0	-2	2	3	1	6	4	6	5	1
Republic of Serbia	4	1	1	-1	1	0	1	3	2	5	3	4	4	3
Romania	4	1	1	-1	0	0	1	3	1	5	5	4	3	3
Russia	4	1	1	-1	1	-1	1	2	1	4	4	4	3	3
Saudi Arabia	4	1	1	1	-2	0	1	3	3	6	6	6	5	4
Singapore	3	1	1	-1	-1	-1	1	3	1	5	5	5	5	3
Slovakia	3	0	1	-1	0	-1	1	1	2	4	3	4	4	3
Slovenia	4	1	1	0	-2	-2	-1	2	-3	4	1	5	4	2
South Africa	4	1	1	-1	0	0	1	3	3	6	4	4	3	3
South Korea	1	1	1	0	0	-1	1	2	2	4	4	5	4	4
Spain	4	1	1	0	0	0	1	3	3	6	5	4	1	3
Sweden	4	0	1	0	-1	-2	-1	3	2	1	1	5	4	3
Switzerland	4	1	1	-1	0	0	1	3	2	4	4	5	3	4
Taiwan	3	1	1	0	-1	-1	1	2	2	4	6	6	4	3
Thailand	4	1	1	-1	1	-1	1	3	2	4	3	4	3	3
Trinidad and Tobago	4	1	1	-1	-1	-1	0	3	3	5	5	4	3	3
Tunisia	4	1	1	-1	1	0	1	3	1	4	4	3	5	3
Turkey	4	1	1	-1	0	-1	1	3	2	4	3	4	3	3
Ukraine	4	1	1	-1	1	-1	1	2	1	4	3	3	3	3
United Arab Emirates	3	1	1	-1	-1	-1	0	3	2	5	4	6	5	4
United Kingdom	4	1	1	0	0	0	1	3	2	4	4	4	3	3
United States of America	4	1	1	-1	0	-1	1	3	1	4	1	4	3	3
Uzbekistan	1	1	1	0	-1	0	-2	3	2	6	5	4	4	3
Vietnam	4	1	1	1	-1	-1	1	2	2	6	5	6	4	3

Appendix

```
setwd("/Users/aliciang/Downloads/FIT3152_Assignment_1")

install.packages("ggplot2")
install.packages("ggcorrplot")
install.packages("scatterplot3d")
install.packages("rpart")
install.packages("dplyr")
install.packages("tibble") # for column_to_rownames function in 3 (a)
install.packages("tidyverse") # for summarise in 3 (b)

library(ggplot2)
library(ggcorrplot)
library(scatterplot3d)
library(rpart)
library(dplyr)
library(tibble)
library(tidyverse)

rm(list = ls())
set.seed(31861148)
cvbase = read.csv("PsyCoronaBaselineExtract.csv")
cvbase <- cvbase[sample(nrow(cvbase), 40000), ] # 40000 rows

# Question 1 (a) - Describing data  ------------------------------------------

dim(cvbase) # To get dimension
str(cvbase) # To show data types of all attributes

# NUMERICAL DISTRIBUTION #

numerical_data <- cvbase[,-50] # removed coded-country in col 50
summary(numerical_data)
par(mar=c(5,4,2,2)) # Adjusting margin size

# Distribution for Numeric Data within range 1 to 8
cols_to_exclude <- c(14:41) # we can exclude [14 to 41]
cols <- setdiff(1:49, cols_to_exclude)    # to get col [1 to 13] and [42 to 29]
# set of colors where each number are the number of attributes in the concept
my_colors <- c(rep("violet",11), rep("orange", 2), rep("yellow", 3), rep("green", 2)
               , rep("pink", 3))
boxplot(numerical_data[, cols], col = my_colors, las=2,
        main = "Distribution for Numerical Data within ranges 1 to 8",
        cex.axis = 0.65)
legend("topleft", legend = c("Affect","Likelihood","Corona Community Injunctive norms",
                             "Trust in Government","General") ,
       col = c("violet", "orange", "yellow", "green", "pink") , pch=20 ,
       pt.cex = 2, cex = 0.7)

# Distribution for Numeric Data within range -2 to 10
cols_to_exclude <- c(21:30)
cols <- setdiff(14:41, cols_to_exclude)
my_colors <- c(rep("red", 3), rep("orange", 4), rep("yellow", 3), rep("green", 3),
               rep("pink", 3), rep("purple", 2))
boxplot(numerical_data[, cols], col = my_colors, las=2,
        main = "Distribution for Numerical Data within ranges -2 to 10", cex.axis = 0.65)
legend("topleft", legend = c("Societal Discontent","Job Insecurity",
                             "Perceived Financial Strain","Disempowerment",
                             "Life Satisfaction","Corona Community Injunctive norms") ,
       col = c("red", "orange", "yellow", "green", "pink", "purple") , pch=20 , pt.cex = 2,
       cex = 0.7)

# Distribution for Employment Status
# counts the number of 1s in each employment status
EmployStatus1 = table(cvbase$employstatus_1)
EmployStatus2 = table(cvbase$employstatus_2)
EmployStatus3 = table(cvbase$employstatus_3)
EmployStatus4 = table(cvbase$employstatus_4)
EmployStatus5 = table(cvbase$employstatus_5)
EmployStatus6 = table(cvbase$employstatus_6)
EmployStatus7 = table(cvbase$employstatus_7)
EmployStatus8 = table(cvbase$employstatus_8)
EmployStatus9 = table(cvbase$employstatus_9)
```

```
EmployStatus10 = table(cvbase$employstatus_10)
# create a frequency table by binding all previous values
EmployStatusFreqTable =cbind(EmployStatus1, EmployStatus2, EmployStatus3, EmployStatus4,
                             EmployStatus5, EmployStatus6, EmployStatus7, EmployStatus8,
                             EmployStatus9, EmployStatus10)
rownames(EmployStatusFreqTable) = c("Frequency")
par(mar=c(8,4,2,2))
barplot(EmployStatusFreqTable, main = "Distribution of Employment status",
        col="#CBC3E3", ylab = "Frequency", las = 2)

# Distribution for ProSocial Behaviours
# Generating frequency table
ProSocial01Freq = table(cvbase$c19ProSo01)
ProSocial02Freq = table(cvbase$c19ProSo02)
ProSocial03Freq = table(cvbase$c19ProSo03)
ProSocial04Freq = table(cvbase$c19ProSo04)
ProSoFreqTable = rbind(ProSocial01Freq, ProSocial02Freq, ProSocial03Freq, ProSocial04Freq)
par(mar=c(5,4,2,2))
colnames(ProSoFreqTable) <- c("Strongly disagree","Disagree","Somewhat disagree",
                              "Neither","Somewhat agree","Agree","Strongly agree")
barplot(ProSoFreqTable, main = "Distribution of ProSocial Behaviours", ylab = "Frequency",
        xlab = "Response", col = c("red","orange","yellow","green"),beside = T)
legend("topleft",c("ProSocial Behaviour 1", "ProSocial Behaviour 2", "ProSocial Behaviour 3",
                   "ProSocial Behaviour 4"), fill = c("red","orange","yellow","green"))

# NA ANALYSIS #

NumofNa = colSums(is.na(cvbase)) # counts num of NA in each col
NumofNa/400                      # show in percentage
# Number of rows where every field is NA except coded_country
nrow(cvbase[rowSums(is.na(cvbase)) == ncol(cvbase)-1, ])

# TEXT ATTRIBUTES #

# Outputs the number of unique values in coded_country
dim(table(unique(unlist(cvbase$coded_country))))
# Outputs the number of empty responses for coded_country
nrow(cvbase[cvbase$coded_country == "", ])


# Question 1 (b) - Data Manipulation  -------------------------------------------

# extract all rows except the ones where all attributes are NA except coded_country
cvbase = cvbase[rowSums(is.na(cvbase)) != ncol(cvbase)-1, ]
nrow(cvbase)
# eliminating rows where ALL pro-social behaviours are NA
cvbase <- cvbase[!apply(is.na(cvbase[, c(51:54)]), 1, all), ]
nrow(cvbase)
# Change all NA in employment status to O
cvbase[is.na(cvbase$employstatus_1),21]= 0
cvbase[is.na(cvbase$employstatus_2),22]= 0
cvbase[is.na(cvbase$employstatus_3),23]= 0
cvbase[is.na(cvbase$employstatus_4),24]= 0
cvbase[is.na(cvbase$employstatus_5),25]= 0
cvbase[is.na(cvbase$employstatus_6),26]= 0
cvbase[is.na(cvbase$employstatus_7),27]= 0
cvbase[is.na(cvbase$employstatus_8),28]= 0
cvbase[is.na(cvbase$employstatus_9),29]= 0
cvbase[is.na(cvbase$employstatus_10),30]= 0
# Change all NA in job insecurity to O
cvbase[is.na(cvbase$jbInsec01),17]= -3
cvbase[is.na(cvbase$jbInsec02),18]= -3
cvbase[is.na(cvbase$jbInsec03),19]= -3
cvbase[is.na(cvbase$jbInsec04),20]= -3

# Question 2 (a) - Difference in responses for Malaysia and other countries -----

# Splitting into two datas et, Malaysia and not in Malaysia
MsiaCvbase = cvbase[cvbase$coded_country == 'Malaysia', ]
OthersCvbase = cvbase[cvbase$coded_country != 'Malaysia', ]

# Removing coded_country since it's redundant now
MsiaCvbase$coded_country <- NULL
OthersCvbase$coded_country <- NULL

# Get Dimensions
dim(MsiaCvbase)
```

```
dim(OthersCvbase)

# DIFFERENCE IN RESPONSES #

# Get number of NA in each column, divide by number of rows to get percentage of NA
# comparison between Malaysia and Other countries
NApercentForMalaysia = as.data.frame(as.table(colSums(is.na(MsiaCvbase))/570))
NApercentForOthers = as.data.frame(as.table(colSums(is.na(OthersCvbase))/3943))
# Rounding
NApercentForMalaysia$Freq <- round(NApercentForMalaysia$Freq ,digit=3)
NApercentForOthers$Freq <- round(NApercentForOthers$Freq ,digit=3)
# Bind to create a table
NAcomparison = cbind(NApercentForMalaysia, NApercentForOthers[,2])
# Removing Job Insecurity and Employment Status as NA are replaced previously
NAcomparison <- subset(NAcomparison, !(row.names(NAcomparison) %in%
                                       c(17,18,19,20,21,22,23,24,25,26,27,28,29,30)))
# Renaming columns
colnames(NAcomparison) = c("Attributes", "NAinMalaysia","NAinOtherCountries" )
# Sum of columns where % of NA in Malaysia is less than Other Countries
sum(NAcomparison$NAinMalaysia < NAcomparison$NAinOtherCountries)         # Output : 39

# Scatter plot for % of NA values in Malaysia Vs Other Countries
colors <- c("Malaysia" = "red", "Other countries" = "blue")
ggplot(NAcomparison, aes(x = Attributes)) +
  geom_point(aes(y = NAinMalaysia, color = "Malaysia"), size = 1.5) +
  geom_point(aes(y = NAinOtherCountries, color = "Other countries"), size = 1.5) +
  labs(x = "Attributes", y = "Percentage of NA values", color = "Legend") +
  scale_color_manual(values = colors) +
  theme(axis.text.x = element_text(angle = 90)) +
  theme(legend.position = c(.95, .95),legend.justification = c("right", "top"),
        legend.box.just = "right",legend.margin = margin(6, 6, 6, 6)  ) +
  ggtitle("Percentage of NA values between Malaysia and Other Countries") +
  coord_cartesian( ylim = c(0, 0.175))

# Question 2 (b) - Predictors for Malaysia -----------------------------------

MsiaCvbase = na.omit(MsiaCvbase) # Remove rows with NA for correlation

# Correlation list for all Pro-socials
MsiaCorr01 = round(cor(MsiaCvbase[,1:49],MsiaCvbase[,50]), digits= 3)
MsiaCorr02 = round(cor(MsiaCvbase[,1:49],MsiaCvbase[,51]), digits= 3)
MsiaCorr03 = round(cor(MsiaCvbase[,1:49],MsiaCvbase[,52]), digits= 3)
MsiaCorr04 = round(cor(MsiaCvbase[,1:49],MsiaCvbase[,53]), digits= 3)

# CORRELATION MATRIX #

corr_matrix = cor(MsiaCvbase[,c(42,44,46,50:53)])
ggcorrplot(corr_matrix, hc.order =FALSE, type = "lower", lab = TRUE,
           title = "Strong Correlation Predictors Summary (Malaysia)")

# Strongest correlation with trustGovState for c19ProSo03
Msiacorr = round(cor(MsiaCvbase[,1:49],MsiaCvbase[,52]), digits= 3)
colnames(Msiacorr) = c("Correlation")
fit_MsiaCvbase = lm(c19ProSo03 ~ trustGovState, data = MsiaCvbase)
ggplot(MsiaCvbase, aes(x=trustGovState, y=c19ProSo03) ) +  geom_point(alpha = 0.1) +
  geom_smooth(method = "lm", se = FALSE)

# LINEAR REGRESSION #

# Linear regression with all predictors for c19ProSo01
fit_MsiaCvbase = lm(c19ProSo01 ~ affAnx + affCalm + affContent + affBor + affEnerg + affDepr +
                      affExc + affNerv + affExh + affInsp + affRel + PLRAC19 + PLRAEco +
                      disc01 + disc02 + disc03 + jbInsec01 + jbInsec02 + jbInsec03 +
                      jbInsec04 + employstatus_1 + employstatus_2 + employstatus_3 +
                      employstatus_4 + employstatus_5 + employstatus_6 + employstatus_7 +
                      employstatus_8 + employstatus_9 + employstatus_10 + PFS01 + PFS02 +
                      PFS03 + fail01 + fail02 + fail03 + happy + lifeSat + MLQ +
                      c19NormShould + c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                      trustGovCtry + trustGovState +
                      gender + age + edu , data = MsiaCvbase)
summary(fit_MsiaCvbase)
# Strong predictors : employstatus_10, PLRAC19, MLQ and employstatus_7
# orders p-values from lowest to highest
modcoef <- summary(fit_MsiaCvbase)[["coefficients"]]
modcoef[order(modcoef[ , 4]), ]

# Attempts to reduce R^2 value
```

```
# Attempt 1 - removing predictors with highest correlation ()
fit_MsiaCvbase = lm(c19ProSo01 ~ affAnx + affCalm + affContent + affBor + affEnerg + affDepr +
                      affExc + affNerv + affExh + affInsp + affRel + PLRAC19 + PLRAEco +
                      disc01 + disc02 + disc03 + jbInsec01 + jbInsec02 + jbInsec03 +
                      jbInsec04 + employstatus_1 + employstatus_2 + employstatus_3 +
                      employstatus_4 + employstatus_5 + employstatus_6 + employstatus_7 +
                      employstatus_8 + employstatus_9 + employstatus_10 + PFS01 + PFS02 +
                      PFS03 + fail01 + fail02 + fail03 + happy + lifeSat + MLQ +
                      c19NormShould + c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                      trustGovCtry + gender + age + edu , data = MsiaCvbase)
summary(fit_MsiaCvbase)                  # R^2 value : 0.1375

# Attempt 2 - removing predictors with high p-value
# if up until employstatus_9 was removed (the 6th predictor with largest p-value)
fit_MsiaCvbase = lm(c19ProSo01 ~ affBor + affEnerg + affDepr + affNerv + affExh + affInsp +
                      affRel + PLRAC19 + PLRAEco + disc01 + disc02 + disc03 + jbInsec01 +
                      jbInsec02 + jbInsec03 + jbInsec04 + employstatus_1 + employstatus_3 +
                      employstatus_4 + employstatus_5 + employstatus_6 + employstatus_7 +
                      employstatus_8 + employstatus_10 + PFS01 + PFS02 +
                      PFS03 + fail01 + fail02 + fail03 + happy + lifeSat + MLQ +
                      c19NormShould + c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                      trustGovCtry + trustGovState + gender + age + edu , data = MsiaCvbase)
summary(fit_MsiaCvbase)     # R^2 value : 0.1438 , which is lesser than original


# the best set of predictors to use is to eliminate the top 5 largest p-values
# (affAnx, affExc, affCalm, affContent and employstatus_2)
fit_MsiaCvbase = lm(c19ProSo01 ~ affBor + affEnerg + affDepr + affNerv + affExh + affInsp +
                      affRel + PLRAC19 + PLRAEco + disc01 + disc02 + disc03 + jbInsec01 +
                      jbInsec02 + jbInsec03 + jbInsec04 + employstatus_1 + employstatus_3 +
                      employstatus_4 + employstatus_5 + employstatus_6 + employstatus_7 +
                      employstatus_8  + employstatus_9 + employstatus_10 + PFS01 + PFS02 +
                      PFS03 + fail01 + fail02 + fail03 + happy + lifeSat + MLQ +
                      c19NormShould + c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                      trustGovCtry + trustGovState + gender + age + edu , data = MsiaCvbase)
summary(fit_MsiaCvbase)
# R^2 value : 0.1439 but with reduced RSE and overall p-value compared to original
# Strong predictors : employstatus_10, PLRAC19, MLQ and employstatus_7 which is still same

# Relationship between the 2 strong predictors
qplot(PLRAC19,c19ProSo01, data = MsiaCvbase, size = employstatus_10)

# Linear regression with all predictors for c19ProSo02
fit_MsiaCvbase = lm(c19ProSo02 ~ affAnx + affCalm + affContent + affBor + affEnerg + affDepr +
                      affExc + affNerv + affExh + affInsp + affRel + PLRAC19 + PLRAEco +
                      disc01 + disc02 + disc03 + jbInsec01 + jbInsec02 + jbInsec03 +
                      jbInsec04 + employstatus_1 + employstatus_2 + employstatus_3 +
                      employstatus_4 + employstatus_5 + employstatus_6 + employstatus_7 +
                      employstatus_8 + employstatus_9 + employstatus_10 + PFS01 + PFS02 +
                      PFS03 + fail01 + fail02 + fail03 + happy + lifeSat + MLQ +
                      c19NormShould + c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                      trustGovCtry + trustGovState +
                      gender + age + edu , data = MsiaCvbase)
summary(fit_MsiaCvbase)     # R^2 value : 0.2185

# the best set of predictors to use is to eliminate the top 4 largest p-values
# (employstatus_7, disc_01, fail03 and fail02)
fit_MsiaCvbase = lm(c19ProSo02 ~ affAnx + affCalm + affContent + affBor + affEnerg + affDepr +
                      affExc + affNerv + affExh + affInsp + affRel + PLRAC19 + PLRAEco +
                      disc02 + disc03 + jbInsec01 + jbInsec02 + jbInsec03 +
                      jbInsec04 + employstatus_1 + employstatus_2 + employstatus_3 +
                      employstatus_4 + employstatus_5 + employstatus_6 +
                      employstatus_8 + employstatus_9 + employstatus_10 + PFS01 + PFS02 +
                      PFS03 + fail01 + happy + lifeSat + MLQ +
                      c19NormShould + c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                      trustGovCtry + trustGovState +
                      gender + age + edu , data = MsiaCvbase)
summary(fit_MsiaCvbase)     # R^2 value : 0.2185

# 3D scatterplot of the 2 strongest predictors
sur <- scatterplot3d(MsiaCvbase$c19ProSo02,MsiaCvbase$c19IsOrg,MsiaCvbase$c19NormShould,
pch=16)
fit_MsiaCvbase = lm(c19ProSo02 ~ c19IsOrg + c19NormShould, data = MsiaCvbase)
sur$plane3d(fit_MsiaCvbase)

# Linear regression with all predictors for c19ProSo03
fit_MsiaCvbase = lm(c19ProSo03 ~ affAnx + affCalm + affContent + affBor + affEnerg + affDepr +
```

```
                      affExc + affNerv + affExh + affInsp + affRel + PLRAC19 + PLRAEco +
                      disc01 + disc02 + disc03 + jbInsec01 + jbInsec02 + jbInsec03 +
                      jbInsec04 + employstatus_1 + employstatus_2 + employstatus_3 +
                      employstatus_4 + employstatus_5 + employstatus_6 + employstatus_7 +
                      employstatus_8 + employstatus_9 + employstatus_10 + PFS01 + PFS02 +
                      PFS03 + fail01 + fail02 + fail03 + happy + lifeSat + MLQ +
                      c19NormShould + c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                      trustGovCtry + trustGovState +
                      gender + age + edu , data = MsiaCvbase)
summary(fit_MsiaCvbase) # R^2 : 0.1445

modcoef <- summary(fit_MsiaCvbase)[["coefficients"]]
modcoef[order(modcoef[ , 4]), ]

# the best set of predictors to use is to eliminate the top 4 largest p-values
#   (affExh, affContent, disc02 and affBor)
fit_MsiaCvbase = lm(c19ProSo03 ~ affAnx + affCalm + affEnerg + affDepr +
                      affExc + affNerv + affInsp + affRel + PLRAC19 + PLRAEco +
                      disc01 + disc03 + jbInsec01 + jbInsec02 + jbInsec03 +
                      jbInsec04 + employstatus_1 + employstatus_2 + employstatus_3 +
                      employstatus_4 + employstatus_5 + employstatus_6 + employstatus_7 +
                      employstatus_8 + employstatus_9 + employstatus_10 + PFS01 + PFS02 +
                      PFS03 + fail01 + fail02 + fail03 + happy + lifeSat + MLQ +
                      c19NormShould + c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                      trustGovCtry + trustGovState +
                      gender + age + edu , data = MsiaCvbase)
summary(fit_MsiaCvbase)      # R^2 : 0.1445

# Linear regression with all predictors for c19ProSo04
fit_MsiaCvbase = lm(c19ProSo04 ~ affAnx + affCalm + affContent + affBor + affEnerg + affDepr +
                      affExc + affNerv + affExh + affInsp + affRel + PLRAC19 + PLRAEco +
                      disc01 + disc02 + disc03 + jbInsec01 + jbInsec02 + jbInsec03 +
                      jbInsec04 + employstatus_1 + employstatus_2 + employstatus_3 +
                      employstatus_4 + employstatus_5 + employstatus_6 + employstatus_7 +
                      employstatus_8 + employstatus_9 + employstatus_10 + PFS01 + PFS02 +
                      PFS03 + fail01 + fail02 + fail03 + happy + lifeSat + MLQ +
                      c19NormShould + c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                      trustGovCtry + trustGovState +
                      gender + age + edu , data = MsiaCvbase)
summary(fit_MsiaCvbase)      # R^2 : 0.1417

# the best set of predictors to use is to eliminate the top 4 largest p-values
#   (affRel, fail01, affEnerg, employstatus_9 and c19IsPunish)
fit_MsiaCvbase = lm(c19ProSo04 ~ affAnx + affCalm + affContent + affBor + affDepr +
                      affExc + affNerv + affExh + affInsp + PLRAC19 + PLRAEco +
                      disc01 + disc02 + disc03 + jbInsec01 + jbInsec02 + jbInsec03 +
                      jbInsec04 + employstatus_1 + employstatus_2 + employstatus_3 +
                      employstatus_4 + employstatus_5 + employstatus_6 + employstatus_7 +
                      employstatus_8 + employstatus_10 + PFS01 + PFS02 +
                      PFS03 + fail02 + fail03 + happy + lifeSat + MLQ +
                      c19NormShould + c19NormDo + c19IsStrict  + c19IsOrg +
                      trustGovCtry + trustGovState +
                      gender + age + edu , data = MsiaCvbase)
summary(fit_MsiaCvbase)      # R^2 : 0.1417

# Question 2 (c) - Predictors for Other countries -----------------------------

# CORRELATION MATRIX #

# Remove rows with NA for correlation
OthersCvbase = na.omit(OthersCvbase)

# Correlation list for all Pro-socials
OthersCorr01 = round(cor(OthersCvbase[,c(1:27,29:49)],OthersCvbase[,50]), digits= 3)
# Strongest Correlation : MLQ & trustGovState
OthersCorr02 = round(cor(OthersCvbase[,c(1:27,29:49)],OthersCvbase[,51]), digits= 3)
# Strongest Correlation : MLQ
OthersCorr03 = round(cor(OthersCvbase[,c(1:27,29:49)],OthersCvbase[,52]), digits= 3)
# Strongest Correlation : trustGovState
OthersCorr04 = round(cor(OthersCvbase[,c(1:27,29:49)],OthersCvbase[,53]), digits= 3)
# Strongest Correlation : c19NormShould

# create subset for strong correlation predictors and ProSocial Behaviors
corr_matrix = round(cor(OthersCvbase[,c(39,40,46,50:53)]),2)
ggcorrplot(corr_matrix, hc.order =FALSE, type = "lower", lab = TRUE,
           title = "Strong Correlation Predictors Summary (Other Countries)")
```

```
# Linear regression with all predictors for c19ProSo01
fit_OthersCvbase = lm(c19ProSo01 ~ affAnx + affCalm + affContent + affBor + affEnerg +
                        affDepr + affExc + affNerv + affExh + affInsp + affRel + PLRAC19 +
                        PLRAEco +
                        disc01 + disc02 + disc03 + jbInsec01 + jbInsec02 + jbInsec03 +
                        jbInsec04 + employstatus_1 + employstatus_2 + employstatus_3 +
                        employstatus_4 + employstatus_5 + employstatus_6 + employstatus_7 +
                        employstatus_8 + employstatus_9 + employstatus_10 + PFS01 + PFS02 +
                        PFS03 + fail01 + fail02 + fail03 + happy + lifeSat + MLQ +
                        c19NormShould + c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                        trustGovCtry + trustGovState +
                        gender + age + edu , data = OthersCvbase)
summary(fit_OthersCvbase)
# orders p-values from lowest to highest
modcoef <- summary(fit_OthersCvbase)[["coefficients"]]
modcoef[order(modcoef[ , 4]), ]


# Linear regression with all predictors for c19ProSo02
fit_OthersCvbase = lm(c19ProSo02 ~ affAnx + affCalm + affContent + affBor + affEnerg +
                        affExc + affNerv + affExh + affInsp + affRel + PLRAC19 + PLRAEco +
                        disc01 + disc02 + disc03 + jbInsec01 + jbInsec02 + jbInsec03 +
                        jbInsec04 + employstatus_1 + employstatus_2 + employstatus_3 +
                        employstatus_4 + employstatus_5 + employstatus_6 + employstatus_7 +
                        employstatus_8 + employstatus_9 + employstatus_10 + PFS01 + PFS02 +
                        PFS03 + fail01 + fail02 + fail03 + happy + lifeSat + MLQ +
                        c19NormShould + c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                        trustGovCtry + trustGovState + affDepr +
                        gender + age + edu , data = OthersCvbase)
summary(fit_OthersCvbase)

# Linear regression with all predictors for c19ProSo03
fit_OthersCvbase = lm(c19ProSo03 ~ affAnx + affCalm + affContent + affBor + affEnerg +
                        affExc + affNerv + affExh + affInsp + affRel + PLRAC19 + PLRAEco +
                        disc01 + disc02 + disc03 + jbInsec01 + jbInsec02 + jbInsec03 +
                        jbInsec04 + employstatus_1 + employstatus_2 + employstatus_3 +
                        employstatus_4 + employstatus_5 + employstatus_6 + employstatus_7 +
                        employstatus_8 + employstatus_9 + employstatus_10 + PFS01 + PFS02 +
                        PFS03 + fail01 + fail02 + fail03 + happy + lifeSat + MLQ +
                        c19NormShould + c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                        trustGovCtry + trustGovState +  affDepr +
                        gender + age + edu , data = OthersCvbase)
summary(fit_OthersCvbase)

# Linear regression with all predictors for c19ProSo04
fit_OthersCvbase = lm(c19ProSo04 ~ affAnx + affCalm + affContent + affBor + affEnerg +affDepr+
                        affExc + affNerv + affExh + affInsp + affRel + PLRAC19 + PLRAEco +
                        disc01 + disc02 + disc03 + jbInsec01 + jbInsec02 + jbInsec03 +
                        jbInsec04 + employstatus_1 + employstatus_2 + employstatus_3 +
                        employstatus_4 + employstatus_5 + employstatus_6 + employstatus_7 +
                        employstatus_8 + employstatus_9 + employstatus_10 + PFS01 + PFS02 +
                        PFS03 + fail01 + fail02 + fail03 + happy + lifeSat + MLQ +
                        c19NormShould + c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                        trustGovCtry + trustGovState +
                        gender + age + edu , data = OthersCvbase)
summary(fit_OthersCvbase)

# CONFUSION MATRIX #
set.seed(31861148)

# Changing Malaysia's pro-socials to factors for better prediction
MsiaCvbase$c19ProSo01 = as.factor(MsiaCvbase$c19ProSo01)
MsiaCvbase$c19ProSo02 = as.factor(MsiaCvbase$c19ProSo02)
MsiaCvbase$c19ProSo03 = as.factor(MsiaCvbase$c19ProSo03)
MsiaCvbase$c19ProSo04 = as.factor(MsiaCvbase$c19ProSo04)

# Recursive partitioning model for other countries predicting c19ProSo01 in Malaysia
fit_OthersCvbaseRpart = rpart(c19ProSo01 ~ affAnx + affCalm + affContent + affBor +
                                affEnerg + affDepr + affExc + affNerv + affExh + affInsp +
                                affRel + PLRAC19 + PLRAEco + disc01 + disc02 + disc03 +
                                jbInsec01 + jbInsec02 + jbInsec03 + jbInsec04 +
                                employstatus_1 + employstatus_2 + employstatus_3 +
                                employstatus_4 + employstatus_5 + employstatus_6 +
                                employstatus_7 + employstatus_8 + employstatus_9 +
                                employstatus_10 + PFS01 + PFS02 + PFS03 + fail01 +
                                fail02 + fail03 + happy + lifeSat + MLQ + c19NormShould +
                                c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
```

```
                              trustGovCtry + trustGovState + gender + age + edu ,
                            data = OthersCvbase
                            , method="class")

predicted_values = as.data.frame(predict(fit_OthersCvbaseRpart, MsiaCvbase[1:49], type
="class"))
colnames(predicted_values) = c("c19ProSo01")
CM_Others1 = table(predicted_values$c19ProSo01, MsiaCvbase$c19ProSo01)

# Recursive partitioning model for other countries predicting c19ProSo02 in Malaysia
fit_OthersCvbaseRpart = rpart(c19ProSo02 ~ affAnx + affCalm + affContent + affBor +
                              affEnerg + affDepr + affExc + affNerv + affExh + affInsp +
                              affRel + PLRAC19 + PLRAEco + disc01 + disc02 + disc03 +
                              jbInsec01 + jbInsec02 + jbInsec03 + jbInsec04 +
                              employstatus_1 + employstatus_2 + employstatus_3 +
                              employstatus_4 + employstatus_5 + employstatus_6 +
                              employstatus_7 + employstatus_8 + employstatus_9 +
                              employstatus_10 + PFS01 + PFS02 + PFS03 + fail01 +
                              fail02 + fail03 + happy + lifeSat + MLQ + c19NormShould +
                              c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                              trustGovCtry + trustGovState + gender + age + edu ,
                            data = OthersCvbase
                            , method="class")

predicted_values = as.data.frame(predict(fit_OthersCvbaseRpart, MsiaCvbase[1:49], type
="class"))
colnames(predicted_values) = c("c19ProSo02")
CM_Others2 = table(predicted_values$c19ProSo02, MsiaCvbase$c19ProSo02)

# Recursive partitioning model for other countries predicting c19ProSo03 in Malaysia
fit_OthersCvbaseRpart = rpart(c19ProSo03 ~ affAnx + affCalm + affContent + affBor +
                              affEnerg + affDepr + affExc + affNerv + affExh + affInsp +
                              affRel + PLRAC19 + PLRAEco + disc01 + disc02 + disc03 +
                              jbInsec01 + jbInsec02 + jbInsec03 + jbInsec04 +
                              employstatus_1 + employstatus_2 + employstatus_3 +
                              employstatus_4 + employstatus_5 + employstatus_6 +
                              employstatus_7 + employstatus_8 + employstatus_9 +
                              employstatus_10 + PFS01 + PFS02 + PFS03 + fail01 +
                              fail02 + fail03 + happy + lifeSat + MLQ + c19NormShould +
                              c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                              trustGovCtry + trustGovState + gender + age + edu ,
                            data = OthersCvbase
                            , method="class")

predicted_values = as.data.frame(predict(fit_OthersCvbaseRpart, MsiaCvbase[1:49], type
="class"))
colnames(predicted_values) = c("c19ProSo03")
CM_Others3 = table(predicted_values$c19ProSo03, MsiaCvbase$c19ProSo03)

# Recursive partitioning model for other countries predicting c19ProSo04 in Malaysia
fit_OthersCvbaseRpart = rpart(c19ProSo04 ~ affAnx + affCalm + affContent + affBor +
                              affEnerg + affDepr + affExc + affNerv + affExh + affInsp +
                              affRel + PLRAC19 + PLRAEco + disc01 + disc02 + disc03 +
                              jbInsec01 + jbInsec02 + jbInsec03 + jbInsec04 +
                              employstatus_1 + employstatus_2 + employstatus_3 +
                              employstatus_4 + employstatus_5 + employstatus_6 +
                              employstatus_7 + employstatus_8 + employstatus_9 +
                              employstatus_10 + PFS01 + PFS02 + PFS03 + fail01 +
                              fail02 + fail03 + happy + lifeSat + MLQ + c19NormShould +
                              c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                              trustGovCtry + trustGovState + gender + age + edu ,
                            data = OthersCvbase
                            , method="class")

predicted_values = as.data.frame(predict(fit_OthersCvbaseRpart, MsiaCvbase[1:49], type
="class"))
colnames(predicted_values) = c("c19ProSo04")
CM_Others4 = table(predicted_values$c19ProSo04, MsiaCvbase$c19ProSo04)



# Question 3 (a) - Cluster similar countries ----------------------------------

cvbase = na.omit(cvbase)
# create subset of cvbase with only indicator attributes
cvbaseIndicators = cvbase[,c(12,14:16,34:36,40:46,50)]
```

```
# This function finds the most common value (the mode) for the attribute
mode_fun <- function(x) {
  ux <- unique(x)
  ux[which.max(tabulate(match(x, ux)))]
}

# Group the indicator dataset by coded_country and calculate the mode of each
#   variable using the function above
# column_to_rownames turns the coded_country into the index for each row (row name)
country_indicator <- cvbaseIndicators %>%
  group_by(coded_country) %>%
  summarise(across(PLRAC19:trustGovState, mode_fun)) %>%
  column_to_rownames("coded_country")

write_csv(country_indicator,
          "/Users/aliciang/Downloads/FIT3152_Assignment_1/Indicator_dataset")


# perform hierarchical clustering on country_indicator by first computing the distance
#   matrix between responses using the Euclidean distance metric
# Complete linkage clustering is then used on the result
Indicator_cluster <- hclust(dist(country_indicator, method = 'euclidean'),
                            method = 'complete')

plot(Indicator_cluster, main = 'Country Clusters', xlab = "Country", sub = "",  cex = 0.8)

# Make a dendrogram object
Indicator_dendogram = as.dendrogram(Indicator_cluster)

# Plot the branch for clearer view of cluster
plot(cut(Indicator_dendogram, h =4)$lower[[10]],
     main="Tenth branch of lower tree with cut at h=4")

# Question 3 (b) - Predictors for Cluster -------------------------------------
ClusteredCvbase <- cvbase[cvbase$coded_country %in% c("Indonesia","Vietnam", "Singapore",
                                                      "United Arab Emirates", "Austria",
                                                      "Croatia"), ]

ClusteredCvbase$coded_country <- NULL


MsiaCvbase = na.omit(MsiaCvbase) # Remove rows with NA for correlation

# Correlation list for all Pro-socials
ClusCorr01 = sort(round(cor(ClusteredCvbase[,1:49],ClusteredCvbase[,50]), digits= 3))
ClusCorr02 = sort(round(cor(ClusteredCvbase[,1:49],ClusteredCvbase[,51]), digits= 3))
ClusCorr03 = sort(round(cor(ClusteredCvbase[,1:49],ClusteredCvbase[,52]), digits= 3))
ClusCorr04 = sort(round(cor(ClusteredCvbase[,1:49],ClusteredCvbase[,53]), digits= 3))

# CORRELATION MATRIX #

corr_matrix = cor(ClusteredCvbase[,c(40,44,46,50:53)])
ggcorrplot(corr_matrix, hc.order =FALSE, type = "lower", lab = TRUE,
           title = "Strong Correlation Predictors Summary (Cluster)")

# LINEAR REGRESSION #

# Linear regression model for cluster predicting c19ProSo01
fit_ClusteredCvbaseReg = lm(c19ProSo01 ~ affAnx + affCalm + affContent + affBor + affEnerg +
                              affDepr + affExc + affNerv + affExh + affInsp + affRel +
                              PLRAC19 + PLRAEco + disc01 + disc02 + disc03 + jbInsec01 +
                              jbInsec02 + jbInsec03 + jbInsec04 + employstatus_1 +
                              employstatus_2 + employstatus_3 + employstatus_4 +
                              employstatus_5 + employstatus_6 + employstatus_7 +
                              employstatus_8 + employstatus_9 + employstatus_10 +
                              PFS01 + PFS02 + PFS03 + fail01 + fail02 + fail03 +
                              happy + lifeSat + MLQ + c19NormShould + c19NormDo +
                              c19IsStrict + c19IsPunish + c19IsOrg + trustGovCtry +
                              trustGovState + gender + age + edu , data = ClusteredCvbase)
summary(fit_ClusteredCvbaseReg)
modcoef <- summary(fit_ClusteredCvbaseReg)[["coefficients"]]
modcoef[order(modcoef[ , 4]), ]
# Linear regression model for cluster predicting c19ProSo02
fit_ClusteredCvbaseReg = lm(c19ProSo02 ~ affAnx + affCalm + affContent + affBor + affEnerg +
                              affDepr + affExc + affNerv + affExh + affInsp + affRel +
                              PLRAC19 + PLRAEco + disc01 + disc02 + disc03 + jbInsec01 +
                              jbInsec02 + jbInsec03 + jbInsec04 + employstatus_1 +
```

```
                              employstatus_2 + employstatus_3 + employstatus_4 +
                              employstatus_5 + employstatus_6 + employstatus_7 +
                              employstatus_8 + employstatus_9 + employstatus_10 +
                              PFS01 + PFS02 + PFS03 + fail01 + fail02 + fail03 +
                              happy + lifeSat + MLQ + c19NormShould + c19NormDo +
                              c19IsStrict + c19IsPunish + c19IsOrg + trustGovCtry +
                              trustGovState + gender + age + edu , data = ClusteredCvbase)
summary(fit_ClusteredCvbaseReg)

# Linear regression model for cluster predicting c19ProSo03
fit_ClusteredCvbaseReg = lm(c19ProSo03 ~ affAnx + affCalm + affContent + affBor + affEnerg +
                              affDepr + affExc + affNerv + affExh + affInsp + affRel +
                              PLRAC19 + PLRAEco + disc01 + disc02 + disc03 + jbInsec01 +
                              jbInsec02 + jbInsec03 + jbInsec04 + employstatus_1 +
                              employstatus_2 + employstatus_3 + employstatus_4 +
                              employstatus_5 + employstatus_6 + employstatus_7 +
                              employstatus_8 + employstatus_9 + employstatus_10 +
                              PFS01 + PFS02 + PFS03 + fail01 + fail02 + fail03 +
                              happy + lifeSat + MLQ + c19NormShould + c19NormDo +
                              c19IsStrict + c19IsPunish + c19IsOrg + trustGovCtry +
                              trustGovState + gender + age + edu , data = ClusteredCvbase)
summary(fit_ClusteredCvbaseReg)

# Linear regression model for cluster predicting c19ProSo04
fit_ClusteredCvbaseReg = lm(c19ProSo04 ~ affAnx + affCalm + affContent + affBor + affEnerg +
                              affDepr + affExc + affNerv + affExh + affInsp + affRel +
                              PLRAC19 + PLRAEco + disc01 + disc02 + disc03 + jbInsec01 +
                              jbInsec02 + jbInsec03 + jbInsec04 + employstatus_1 +
                              employstatus_2 + employstatus_3 + employstatus_4 +
                              employstatus_5 + employstatus_6 + employstatus_7 +
                              employstatus_8 + employstatus_9 + employstatus_10 +
                              PFS01 + PFS02 + PFS03 + fail01 + fail02 + fail03 +
                              happy + lifeSat + MLQ + c19NormShould + c19NormDo +
                              c19IsStrict + c19IsPunish + c19IsOrg + trustGovCtry +
                              trustGovState + gender + age + edu , data = ClusteredCvbase)
summary(fit_ClusteredCvbaseReg)

# CONFUSION MATRIX #

# Repetitive partitioning model for cluster predicting c19ProSo01
fit_ClusteredCvbaseRpart = rpart(c19ProSo01 ~ affAnx + affCalm + affContent + affBor +
                                   affEnerg + affDepr + affExc + affNerv + affExh + affInsp +
                                   affRel + PLRAC19 + PLRAEco + disc01 + disc02 + disc03 +
                                   jbInsec01 + jbInsec02 + jbInsec03 + jbInsec04 +
                                   employstatus_1 + employstatus_2 + employstatus_3 +
                                   employstatus_4 + employstatus_5 + employstatus_6 +
                                   employstatus_7 + employstatus_8 + employstatus_9 +
                                   employstatus_10 + PFS01 + PFS02 + PFS03 + fail01 +
                                   fail02 + fail03 + happy + lifeSat + MLQ + c19NormShould +
                                   c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                                   trustGovCtry + trustGovState + gender + age + edu ,
                                 data = ClusteredCvbase
                                 , method="class")

predicted_values = as.data.frame(predict(fit_ClusteredCvbaseRpart, MsiaCvbase[1:49], type
="class"))
colnames(predicted_values) = c("c19ProSo01")
CM_Cluster1 = table(predicted_values$c19ProSo01, MsiaCvbase$c19ProSo01)

# Repetitive partitioning model for cluster predicting c19ProSo02
fit_ClusteredCvbaseRpart = rpart(c19ProSo02 ~ affAnx + affCalm + affContent + affBor +
                                   affEnerg + affDepr + affExc + affNerv + affExh + affInsp +
                                   affRel + PLRAC19 + PLRAEco + disc01 + disc02 + disc03 +
                                   jbInsec01 + jbInsec02 + jbInsec03 + jbInsec04 +
                                   employstatus_1 + employstatus_2 + employstatus_3 +
                                   employstatus_4 + employstatus_5 + employstatus_6 +
                                   employstatus_7 + employstatus_8 + employstatus_9 +
                                   employstatus_10 + PFS01 + PFS02 + PFS03 + fail01 +
                                   fail02 + fail03 + happy + lifeSat + MLQ + c19NormShould +
                                   c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                                   trustGovCtry + trustGovState + gender + age + edu ,
                                 data = ClusteredCvbase
                                 , method="class")

predicted_values = as.data.frame(predict(fit_ClusteredCvbaseRpart, MsiaCvbase[1:49], type
="class"))
colnames(predicted_values) = c("c19ProSo02")
```

```
CM_Cluster2 = table(predicted_values$c19ProSo02, MsiaCvbase$c19ProSo02)

# Repetitive partitioning model for cluster predicting c19ProSo03
fit_ClusteredCvbaseRpart = rpart(c19ProSo03 ~ affAnx + affCalm + affContent + affBor +
                                   affEnerg + affDepr + affExc + affNerv + affExh + affInsp +
                                   affRel + PLRAC19 + PLRAEco + disc01 + disc02 + disc03 +
                                   jbInsec01 + jbInsec02 + jbInsec03 + jbInsec04 +
                                   employstatus_1 + employstatus_2 + employstatus_3 +
                                   employstatus_4 + employstatus_5 + employstatus_6 +
                                   employstatus_7 + employstatus_8 + employstatus_9 +
                                   employstatus_10 + PFS01 + PFS02 + PFS03 + fail01 +
                                   fail02 + fail03 + happy + lifeSat + MLQ + c19NormShould +
                                   c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                                   trustGovCtry + trustGovState + gender + age + edu ,
                                 data = ClusteredCvbase
                                 , method="class")

predicted_values = as.data.frame(predict(fit_ClusteredCvbaseRpart, MsiaCvbase[1:49], type
="class"))
colnames(predicted_values) = c("c19ProSo03")
CM_Cluster3 = table(predicted_values$c19ProSo03, MsiaCvbase$c19ProSo03)


# Repetitive partitioning model for cluster predicting c19ProSo04
fit_ClusteredCvbaseRpart = rpart(c19ProSo04 ~ affAnx + affCalm + affContent + affBor +
                                   affEnerg + affDepr + affExc + affNerv + affExh + affInsp +
                                   affRel + PLRAC19 + PLRAEco + disc01 + disc02 + disc03 +
                                   jbInsec01 + jbInsec02 + jbInsec03 + jbInsec04 +
                                   employstatus_1 + employstatus_2 + employstatus_3 +
                                   employstatus_4 + employstatus_5 + employstatus_6 +
                                   employstatus_7 + employstatus_8 + employstatus_9 +
                                   employstatus_10 + PFS01 + PFS02 + PFS03 + fail01 +
                                   fail02 + fail03 + happy + lifeSat + MLQ + c19NormShould +
                                   c19NormDo + c19IsStrict + c19IsPunish + c19IsOrg +
                                   trustGovCtry + trustGovState + gender + age + edu ,
                                 data = ClusteredCvbase
                                 , method="class")

predicted_values = as.data.frame(predict(fit_ClusteredCvbaseRpart, MsiaCvbase[1:49], type
="class"))
colnames(predicted_values) = c("c19ProSo04")
CM_Cluster4 = table(predicted_values$c19ProSo04, MsiaCvbase$c19ProSo04)


# Summary Key Findings

CorrectClusterPred = cbind(CM_Cluster1[1,1],CM_Cluster1[2,2],CM_Cluster1[3,3],
                           CM_Cluster1[4,4],CM_Cluster1[5,5],CM_Cluster1[6,6],
                           CM_Cluster1[7,7])
CorrectOthersPred = cbind(CM_Others1[1,1],CM_Others1[2,2],CM_Others1[3,3],
                          CM_Others1[4,4],CM_Others1[5,5],CM_Others1[6,6],
                          CM_Others1[7,7])
CorrectPred1 = rbind(CorrectClusterPred,CorrectOthersPred)
par(mar=c(7,4,2,2))
colnames(CorrectPred1) <- c("Strongly Disagree","Disagree","Somewhat Disagree",
                            "Neither","Somewhat Agree","Agree","Strongly Agree")
# Fits 4 plots together
par(mfrow=c(2,2))
barplot(CorrectPred1, main = "Pro-social behaviour 1", ylab = "Frequency",
        col = c("pink","yellow"),beside = T, las =2, cex.names=0.75)
legend("topleft", legend = c("Cluster countries","Other countries") ,
       col = c("pink","yellow") , pch=20 ,
       pt.cex = 2, cex = 0.7)

CorrectClusterPred = cbind(CM_Cluster2[1,1],CM_Cluster2[2,2],CM_Cluster2[3,3],
                           CM_Cluster2[4,4],CM_Cluster2[5,5],CM_Cluster2[6,6],
                           CM_Cluster2[7,7])
CorrectOthersPred = cbind(CM_Others2[1,1],CM_Others2[2,2],CM_Others2[3,3],
                          CM_Others2[4,4],CM_Others2[5,5],CM_Others2[6,6],
                          CM_Others2[7,7])
CorrectPred2 = rbind(CorrectClusterPred,CorrectOthersPred)

colnames(CorrectPred2) <- c("Strongly Disagree","Disagree","Somewhat Disagree",
                            "Neither","Somewhat Agree","Agree","Strongly Agree")
barplot(CorrectPred2, main = "Pro-social behaviour 2", ylab = "Frequency",
         col = c("pink","yellow"),beside = T, las =2, cex.names=0.75)
```

```
CorrectClusterPred = cbind(CM_Cluster3[1,1],CM_Cluster3[2,2],CM_Cluster3[3,3],
                           CM_Cluster3[4,4],CM_Cluster3[5,5],CM_Cluster3[6,6],
                           CM_Cluster3[7,7])
CorrectOthersPred = cbind(CM_Others3[1,1],CM_Others3[2,2],CM_Others3[3,3],
                          CM_Others3[4,4],CM_Others3[5,5],CM_Others3[6,6],
                          CM_Others3[7,7])
CorrectPred3 = rbind(CorrectClusterPred,CorrectOthersPred)
colnames(CorrectPred3) <- c("Strongly Disagree","Disagree","Somewhat Disagree",
                            "Neither","Somewhat Agree","Agree","Strongly Agree")
barplot(CorrectPred3, main = "Pro-social behaviour 3", ylab = "Frequency",
         col = c("pink","yellow"),beside = T, las =2, cex.names=0.75)

CorrectClusterPred = cbind(CM_Cluster4[1,1],CM_Cluster4[2,2],CM_Cluster4[3,3],
                           CM_Cluster4[4,4],CM_Cluster4[5,5],CM_Cluster4[6,6],
                           CM_Cluster4[7,7])
CorrectOthersPred = cbind(CM_Others4[1,1],CM_Others4[2,2],CM_Others4[3,3],
                          CM_Others4[4,4],CM_Others4[5,5],CM_Others4[6,6],
                          CM_Others4[7,7])
CorrectPred4 = rbind(CorrectClusterPred,CorrectOthersPred)
colnames(CorrectPred4) <- c("Strongly Disagree","Disagree","Somewhat Disagree",
                            "Neither","Somewhat Agree","Agree","Strongly Agree")
barplot(CorrectPred4, main = "Pro-social behaviour 4", ylab = "Frequency",
        col = c("pink","yellow"),beside = T, las =2, cex.names=0.75)
```