Predicting Individual Depression Level by Analysing Personal Characteristics

Yutang Li

Contents

Abstract

Depression has become a common mental illness. In this paper, we report on the possible predictors of depression, which allow people to detect depression at its early stage quickly. We provide an analysis of the characteristics of people showing depression symptoms and develop models that explain the correlations between the level of depression and these characteristics with three iterations. Details about the data set, data mining methods and algorithms, and appropriate software tools we use to build effective models are outlined. Interpretations and implementation of the model are provided.

Code on Github: https://github.com/YutangLi-UoA/722-Iteration4.git

Business and/or Situation understanding

1.1 Identify the objectives of the business/situation

According to the World Health Organization (2020), nowadays, there are more than 264 million people of all ages suffer from depression globally. It is also a leading cause of disability worldwide (CDC, 2019) and is a significant contributor to diseases. The Health Navigator Editorial Team (2019) reports that about 1 in 6 people from all the age groups may suffer from depression at some time in their life. The team also discovered some key symptoms of depression. It includes continuously having negative feelings, losing interests in doing things one used to enjoy doing before and having sleep problems. Depression has become a severe mental health issue worldwide, and we must overcome this obstacle to achieve the Sustainable Development Goals of “Good Health and Wellbeing”.

To achieve this goal, we want to understand possible predictors that correlate to depression and stopping depression at its earlier stage is vital for our present world. It would reduce depression cases and improve people’s health and wellbeing in general. We will collect data from the mental health domain, and we will focus on these following objectives in this research:

- Explore the correlations between the level of depression and the characters of people who have depression problems
- Investigate and predict which kind of people are likely to have depression issues
- Generate a depression checklist which would increase the awareness of people towards depression so that they can take action to stop the development of depression at an early stage

The study will be successful if it satisfies the following criteria:

- Identify the primary characters of depression people
- Decrease the number of severe depression cases as people take actions to prevent the development in time
- People’s overall awareness of depression issues increases
- The study finishes on time

1.2 Assess the situation

Before we start looking for data, we assessed the following issues:

- Can we collect sufficient data to achieve our objectives?
- Do we have adequate software to process and analyse the data?
- What are the possible risks involved in this project?
- How can we reduce those possible risks?

1.2.1 Resource Inventory

There are three analytics solutions available for our study - IBM Software Analytics Solution, Open Source Analytics Solution and Big Data Analytics Solutions. We will select Pyspark as one of the Big Data Analytics Solutions for this project, and we will also use the other solutions to conduct iterations. We will select the best results by comparing and contrasting the iterations.

The primary data source we choose is **Kaggle** because it is an open platform with extensive free datasets from all areas, including mental health. Datasets from Kaggle is available in CSV format, and we can easily import them into the software we have chosen.

This project also needs to be assisted by a data manager and a data analysis involved in this project. Moreover, we reckon it is also good to redo this study every five years and make updates to the characteristics checklist, which correlated to depression.

1.2.2 Requirements, assumptions and constraints

In terms of the requirements, we believe there will not be security or legal restrictions on the data or the result. We will collect our data from an open-source, and all the records and results in this study will be anonymous.

Moreover, the assumptions we made for the data qualities are:

- The sample is randomly selected so that it is not biased because of the collector
- Records are independent of each other so that each line will not affect each other when inputting into the model
- The measurement system of the dataset is accurate so that the results of our model is not inaccurate

The primary constraint of our study is the limited timeframe.

1.2.3 Risks and Contingencies

To accomplish the study smoothly, we have identified risks associated with four aspects: scheduling, financial, data and output. Details of these risks and corresponding contingency plans are shown in *Table 1.2.3* below.

Aspects	Risks	Contingency Plan
Scheduling	Research time is not sufficient	We set deadlines and design a Gantt chart for this project, and communicate our plan with relevant parties before researching
Financial	Licences are needed to access specific data	We will collect data from open sources first. If the datasets from those sources are sufficient for our research, then no need to pay any licences. If paying a licence is crucial, we will make sure the fees

		are under our budget
Data	Data quality is inferior	We will address any null values and missing values, outliers, or data errors in the dataset before conducting any data mining procedures (shown in *Section 3*)
Output	We cannot find a strong correlation between the variables of our interest from the datasets (i.e. characters of depressed people and depression levels)	We plan to process the data with three iterations by the different data analysis software so that we will generate several models to analyse the correlations
Output	We cannot interpret the results	We will visualise the data output in several ways to facilitate our interpretation (shown in 8)

Table 1.2.3 Risks and Contingency

1.2.4 Cost/Benefit Analysis

We estimated that the possible costs for this study are:

- Licenses to assess the dataset
- Cost to install software
- Approximate 12 weeks of researching time

We consider that the benefits of this study are:

- The research objectives are met, a self-check list to alert depression will be available to the public
- Contributions to the Sustainable Development Goals
- Insights generated from data can be reused in future projects

1.3 Determine data mining objectives

1.3.1 Data Ming Goals

We set these data mining goals for this study:

- We use information about people's characters and their depression level to conduct undirected knowledge discovery to explore the correlations between characters of depressed people and their depression level. We plan to firstly explore any pattern in the datasets, then describe and explain how the participants' characters correlate to the participants' depression level.

- After the exploration, we then predict the likelihood of certain people getting depression issues by building a prediction model with Neural Network and Regression Analysis.

- Finally, summaries the patterns to a self-check list. This list would increase the awareness of people towards depression as well as alert people who are very likely to have depression.

1.3.2 Data Ming Success Criteria

Methods for Model Assessment

It is vital to ensure that our model is useful and effective. We will first clean our input data and make sure the input data are free from outliers, missing and null values, as well as data errors. Second, we would visualise and interpret the output of our model, also compare the output with the existing researches to check whether there are any faults in the results. Finally, we will evaluate how well our data mining output achieve our data mining goals and support our business objectives.

Benchmark for Evaluating Success

Success is evaluated base on whether the model illustrates an identifiable relationship between the factors and responses. In this case, we expect the model to describe a significant correlation between the characters of depressed people and their depression level.

Subjective Measurements and the Arbiter of Success

The purpose of this study is to increase the awareness of people to this growing mental issue of depression. We would consider the study as a success if we could identify and explain at least 5 personal characteristics that are highly associated with depression level. The list of the 5 characteristics can be a warning to people with those traits and remind them to pay attention whether they have any depression symptoms.

Successful deployment of model results

The model result can be deployed in the mental health domain or the health system. Even those this research is not supported with medical reasoning. However, it may provide a direction for medical researchers to conduct further study on depression. Also, hospital and insurance companies can refer to this study and identify people who may have a high level of depression using our findings. Hospitals and companies can send out regular emails with mental health advice to prevent these people from catching depression. Governments can also refer to our findings so that they would take actions to support the groups vulnerable to depression to improve the wellbeing of their cities and countries. Ultimately, the motives behind this are to contribute to the sustainable development goal of “Good Health and Wellbeing”.

1.4 Produce a project plan

We plan to collect data about the information of people with depression problems and their level of depression. We will implement three iterations of data mining and build models that simulate and predict the likelihood of a person with particular traits to have depression. We will take characters which have a significant correlation with depression symptoms as the predictors of depression.

There are 9 phases/steps carried out in this research shown by *Table 1.4.1* below.

Phases	Weighting
Business and/or situation understanding	10

Data understanding	10
Data preparation	15
Data transformation	5
Data-mining method(s) selection	10
Data-mining algorithm(s) selection	15
Data mining	15
Interpretation	20
Action	5

Table 1.4.1 Weighting

We assign weightings to each step with a scale from 0 to 20 by their importance to the research. This assignment is essential because we will allocate our research time according to the weightings.

Phase	Time	Resources	Risks
Business and/or situation understanding	5 days	All open source	Causes and correlations changes or new variables come out
Data understanding	6 days	All open source	No appropriate data to serve the research objective
First Iteration	27 days		
Data preparation	6 days	The chosen dataset	Low data quality, hard to clean and transfer to the desirable datatype
Data transformation	4 days	Data after preparation	Valuable data being eliminated, or the processed data is not useful in serving the research objective
Data-mining method(s) selection	4 days	Data after transformation	The method chosen may not be effective to discover a pattern
Data-mining algorithm(s) selection	4 days	Data after transformation	Inability to find an adequate model or building a false simulation of the reality

Data mining	8 days	Data after transformation	Technology problems or no significant relationship found between variables
Interpretation	1 day	Models built from data mining	Inability to find an adequate way to visualise the data given from the models
Second Iteration	34 days		
Third Iteration	28 days		
Action	7 days	Models built from data mining	Situation change, inability to implement the result

Table 1.4.2 Detailed Research Plan

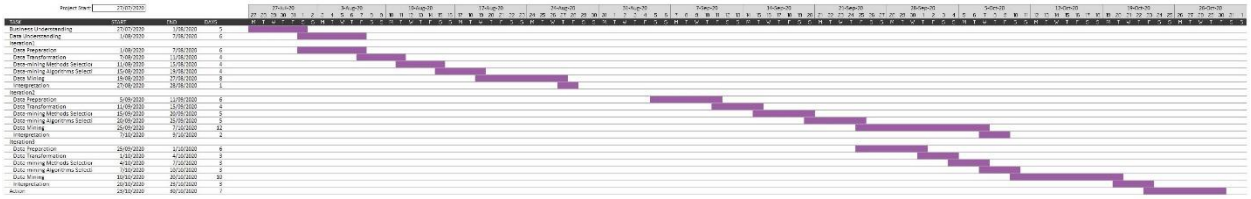


Figure 1.4 Research Gantt Chart

We plan to finish the research in 12 weeks. Within this timeframe, we will execute three iterations for **step 3** to **step 9** and complete each iteration in approximate four weeks. *Table 1.4.2* and *Figure 1.4* above illustrate our research plan in detail. The three segments clearly show the three iterations on the Gantt chart, and we plan to overlap the **Iteration 2** and **3** by two weeks.

Data Understanding

2.1 Collect initial data

During the data collection process, we encounter the following issues:

- There are a larget number of datasets relating to our topic, and it is hard to decide which is the most appropriate set
- Some datasets contain a small number of rows and columns
- The datasets are in a different format and file types, we need to decide the format of our data file
- The sources of some of the datasets are not clearly explained so that we can not rely on those data

To choose the most appropriate dataset, we set criteria of the dataset as follows:

- The dataset must have at least 10,000 rows of records and 20 rows
- We prefer dataset in csv format which can be processed by Pyspark easily
- The source of the data must be clearly stated

The best dataset which satisfies all the criteria is found in Kaggle, titled “Depression Anxiety Stress Scales Responses”. It is available through this link: https://www.kaggle.com/lucasgreenwell/depression-anxiety-stress-scales-responses. The data was initially posted on OpenPsychometrics.org and was collected between 2017 and 2019 from an online depression level test called Depression Anxiety Stress Scales (DASS). DASS is a 42-item self-report questionnaire, designed to measure the three related negative emotional states of depression, anxiety and stress.

The searching process of this dataset is explained below:

First, we use the search function at the home page of Kaggle.com and enter the keyword “Depression”.

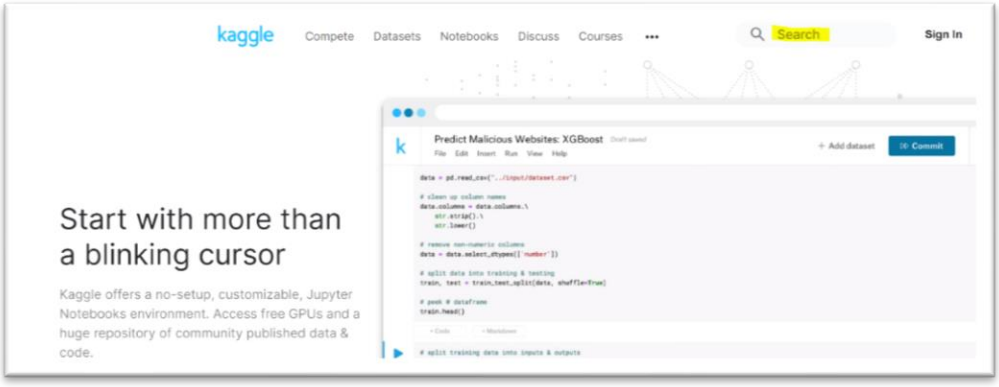


Figure 2.1.1 Kaggle Homepage

We prefer a CSV file for our dataset because we can easily read and process by our chosen software – SPSS Modeller, Python and Pyspark. Thus, we select “csv” in the Filter By column, under Dataset File Types. We obtained 33 results.

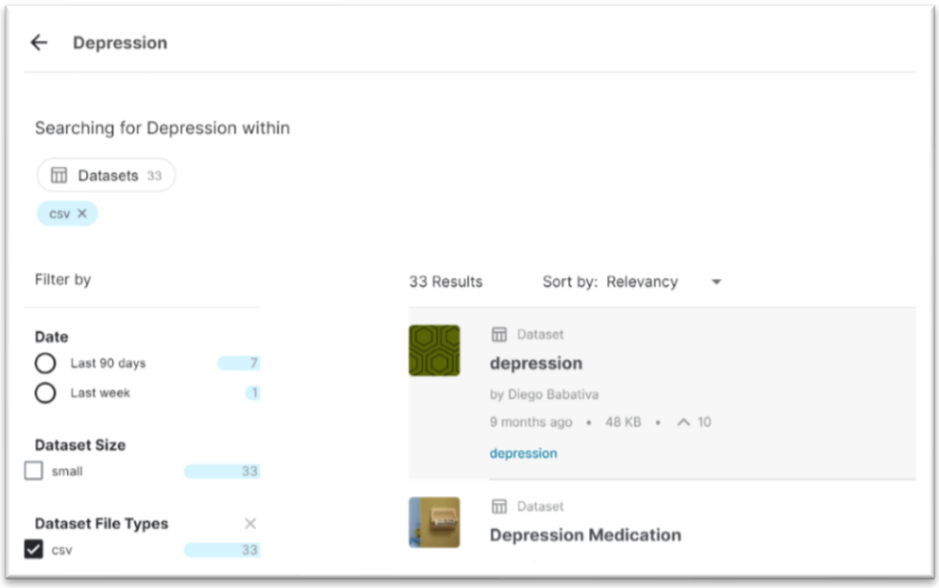


Figure 2.1.2 Searching Results

To select the best dataset to build our model, we would like a dataset which contains at least 5,000 records and 10 columns. So, the model would contain enough data to explain the correlations between the depression level and people's traits and achieve our objective to generate a self-check list for depression. We browse through the results and decide that the "Depression Anxiety Stress Scales Responses" dataset satisfy our purpose best. We will describe this dataset in detail in the next section.

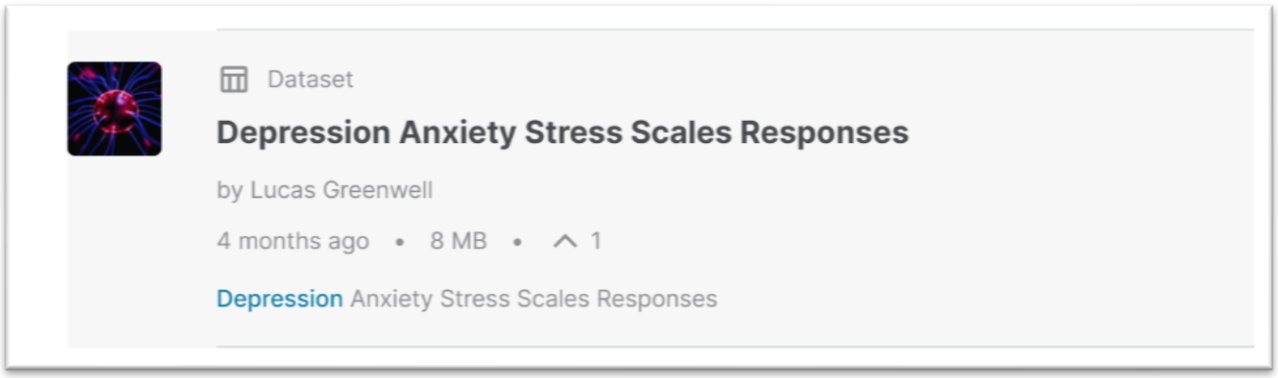


Figure 2.1.3 Selected Dataset – Depression Anxiety Stress Scales Responses

A problem we encountered is that the data contains 172 columns, and it is crucial to understand their meaning before start building our model fully. To interpret those columns, we must understand the data collection process of the DASS test and the meaning of each column, using the codebook provided alongside the dataset.

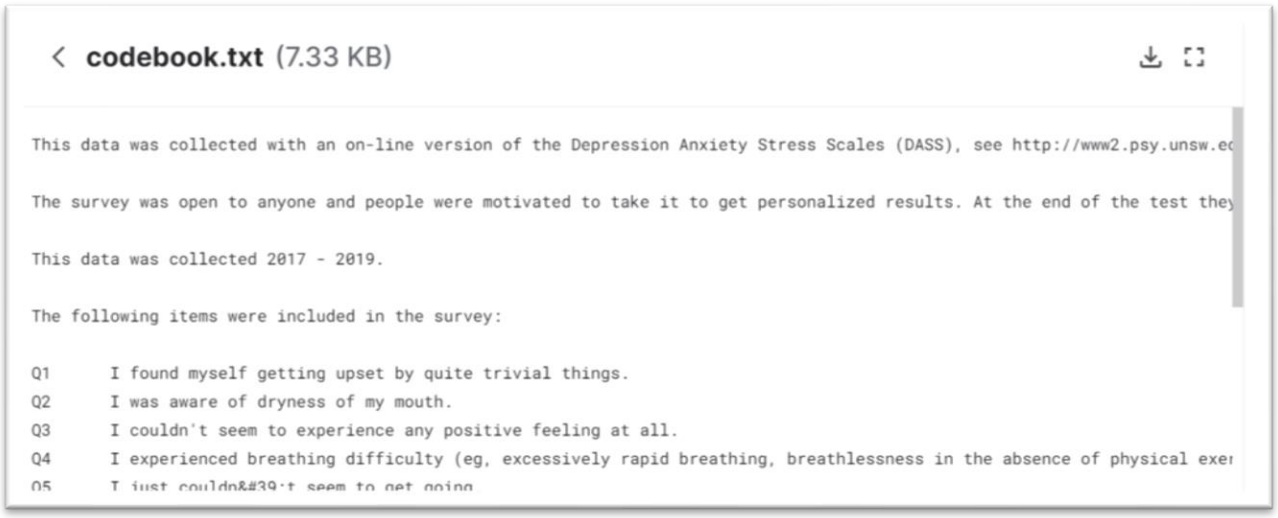


Figure 2.1.4 Codebook

2.2 Describe the data

This data has a sample size of 15000 people. In the raw data package, there are also two csv files and a file of metadata explaining the filed names in the dataset. The first csv file **Depression Scale** records the responses to the 42 questions in the DSAA test, and each question is a symptom of depression. The other csv file **Personal Information** stores the personal information of the participants collected from a follow-up survey. These two sessions are linked by the same attribute **PersonID**.

In this report, we are using **Pyspark** on **Jupyter Notebook** for data mining to understand and explore the patterns of this data. We import the two csv files to Jupytor Notebook by creating two data frames **DS** and **PI,** as shown in *Figure 2.2*.

```
In [1]: import findspark
        findspark.init('/home/ubuntu/spark-2.1.1-bin-hadoop2.7')
        import pyspark
        from pyspark.sql import SparkSession
        spark = SparkSession.builder.appName('basics').getOrCreate()

In [2]: DS = spark.read.load("./DepressionScale.csv",format="csv",header="true")

In [3]: PI = spark.read.load("./PersonInformation.csv",format="csv",header="true")
```

Figure 2.2 Import Datasets

2.2.1 Depression Scale

The first dataset **Depression Scale** records the responses to the DASS test, which asks people to score their relevancy to each depression symptom from 1 to 4, from hardly applicable to highly applicable.

We used **Pyspark** to explore that there are 129 columns and 15000 rows this dataset. From the codebook, we understand that there are three kinds of variables in the first set – Variable A, E and I. **Variable A** represents the answers to each question. At the same time, **Variable E** is the time taken for each person to answer the questions. The questions are also randomly allocated

in the questionnaires, **Variable I** show the position of the question appeared in the test. The sum of Variable A and the sum of Variable E are recorded as **TotalA** and **TotalE**. The shared key **PersonID** is given in the dataset as well.

```
In [10]: print((DS.count(), len(DS.columns)))

(15000, 129)
```

Figure 2.2.1.1 Overview Depression Scale

```
In [4]: DS.columns

Out[4]: ['PersonID',
         'TotalA',
         'TotalE',
         'Q1A',
         'Q1I',
         'Q1E',
         'Q2A',
         'Q2I',
         'Q2E',
         'Q3A',
         'Q3I',
         'Q3E',
         'Q4A',
         'Q4I',
         'Q4E',
```

```
In [5]: DS.show()

+--------+------+--------+---+---+----+---+---+----+---+---+----+---+---+----+---+---+----+---+---+-----+---+---+-----+---+-
--+----+---+---+-----+----+----+-----+----+----+-----+----+----+-----+----+----+-----+----+----+-----+----+----+-----+----+----
+-----+----+----+-----+----+--------+----+----+-----+----+----+-----+----+----+-----+----+----+-----+----+----+-----+----+-----
+----+----+----+-----+----+----+-----+----+----+-----+----+----+-----+----+----+-----+----+----+-----+----+----+-----+----+---
-+----+-----+----+----+-----+----+----+-----+----+----+-----+----+----+-----+----+----+-----+----+----+-----+----+----+-----+--
--+----+----+----+----+-----+----+-----+
|PersonID|TotalA|  TotalE|Q1A|Q1I| Q1E|Q2A|Q2I| Q2E|Q3A|Q3I|  Q3E|Q4A|Q4I| Q4E|Q5A|Q5I| Q5E|Q6A|Q6I|  Q6E|Q7A|Q7I|  Q7E|Q8A|Q
8I| Q8E|Q9A|Q9I|  Q9E|Q10A|Q10I|Q10E|Q11A|Q11I|Q11E|Q12A|Q12I| Q12E|Q13A|Q13I|Q13E|Q14A|Q14I| Q14E|Q15A|Q15I| Q15E|Q16A|Q16I|
Q16E|Q17A|Q17I|Q17E|Q18A|Q18I|    Q18E|Q19A|Q19I| Q19E|Q20A|Q20I| Q20E|Q21A|Q21I|Q21E|Q22A|Q22I|  Q22E|Q23A|Q23I| Q23E|Q24A|Q
24I|Q24E|Q25A|Q25I| Q25E|Q26A|Q26I|Q26E|Q27A|Q27I| Q27E|Q28A|Q28I|Q28E|Q29A|Q29I|   Q29E|Q30A|Q30I| Q30E|Q31A|Q31I|Q31E|Q32A|
Q32I| Q32E|Q33A|Q33I| Q33E|Q34A|Q34I|Q34E|Q35A|Q35I| Q35E|Q36A|Q36I|Q36E|Q37A|Q37I| Q37E|Q38A|Q38I| Q38E|Q39A|Q39I| Q39E|Q40
A|Q40I|Q40E|Q41A|Q41I| Q41E|Q42A|Q42I| Q42E|
```

Figure 2.2.1.2 Columns Depression Scale

We then use **describe()** function to explore more details about the dataset and obtain more information about the dataset, such as the count of the values in each row and the mean of each field. However, the display is in a format that's very hard to read. We will also use **Tableau** to visualise the data so that we can gain more insights from the visualisations.

```
In [6]: DS.describe().show()

---------------+-----------------+----------------+
|  count|          15000|          15000|          15000|          15000|          15000|          15000|
15000|          15000|          15000|          15000|          15000|          15000|          15000|
15000|         15000|          15000|          15000|         15000|          15000|         15000|
15000|          15000|          15000|          15000|          15000|          15000|          15000|
15000|          15000|          15000|          15000|          15000|          15000|          15000|
15000|         15000|          15000|          15000|         15000|          15000|         15000|
15000|         15000|          15000|         15000|          15000|          15000|         15000|
15000|          15000|          15000|          15000|          15000|          15000|          15000|
15000|         15000|         15000|         15000|          15000|          15000|         15000|
15000|          15000|          15000|          15000|          15000|          15000|          15000|
15000|          15000|          15000|          15000|          15000|          15000|          15000|
15000|         15000|          15000|          15000|         15000|          15000|         15000|
15000|          15000|          15000|         15000|          15000|          15000|         15000|
15000|          15000|          15000|          15000|          15000|          15000|          15000|
15000|       15000|       15000|       15000|       15000|       15000|       15000|
15000|          15000|          15000|          15000|          15000|          15000|          15000|
15000|          15000|          15000|          15000|          15000|          15000|          15000|
15000|         15000|          15000|          15000|          15000|          15000|          15000|
```

Figure 2.2.1.2 Row Count Depression Scale

Here is an illustration of the first row, which represents the response of participant with **PersonID 1** from the dataset.

```
In [3]: DS.head(1)

Out[3]: [Row(PersonID='1', TotalA='143', TotalE='157622', Q1A='4', Q1I='28', Q1E='3890', Q2A='4', Q2I='25', Q2E='2122', Q3A='2', Q3I='16', Q3E='1944', Q4A='4', Q4I='8', Q4E='2044', Q5A='4', Q5I='34', Q5E='2153', Q6A='4', Q6I='33', Q6E='2416', Q7A='4', Q7I='10', Q7E='2818', Q8A='4', Q8I='13', Q8E='2259', Q9A='2', Q9I='21', Q9E='5541', Q10A='1', Q10I='38', Q10E='4441', Q11A='4', Q11I='31', Q11E='2451', Q12A='4', Q12I='24', Q12E='3325', Q13A='4', Q13I='14', Q13E='1416', Q14A='4', Q14I='37', Q14E='5021', Q15A='4', Q15I='27', Q15E='2342', Q16A='4', Q16I='39', Q16E='2480', Q17A='3', Q17I='6', Q17E='2476', Q18A='4', Q18I='35', Q18E='1627', Q19A='3', Q19I='17', Q19E='9050', Q20A='3', Q20I='30', Q20E='7001', Q21A='1', Q21I='11', Q21E='4719', Q22A='4', Q22I='20', Q22E='2984', Q23A='4', Q23I='36', Q23E='1313', Q24A='4', Q24I='42', Q24E='2444', Q25A='4', Q25I='1', Q25E='9880', Q26A='4', Q26I='2', Q26E='4695', Q27A='4', Q27I='5', Q27E='1677', Q28A='3', Q28I='4', Q28E='6723', Q29A='4', Q29I='3', Q29E='5953', Q30A='2', Q30I='26', Q30E='8062', Q31A='4', Q31I='12', Q31E='5560', Q32A='4', Q32I='7', Q32E='3032', Q33A='2', Q33I='29', Q33E='3316', Q34A='3', Q34I='40', Q34E='3563', Q35A='4', Q35I='23', Q35E='5594', Q36A='4', Q36I='41', Q36E='1477', Q37A='1', Q37I='18', Q37E='3885', Q38A='2', Q38I='9', Q38E='5265', Q39A='4', Q39I='19', Q39E='1892', Q40A='3', Q40I='22', Q40E='4228', Q41A='4', Q41I='32', Q41E='1574', Q42A='4', Q42I='15', Q42E='2969')]
```

Figure 2.2.1.3 First Participant Depression Scale

Our data mining objective is to understand the correlations between the depression level and the characteristics of people. Therefore, Total A will be the variable we focus in this dataset, for it represents the overall level of depression of a participant. We visualised Total A in **Tableau** and discovered that it is normally distributed. The normal distribution would contribute to the accuracy of our model.

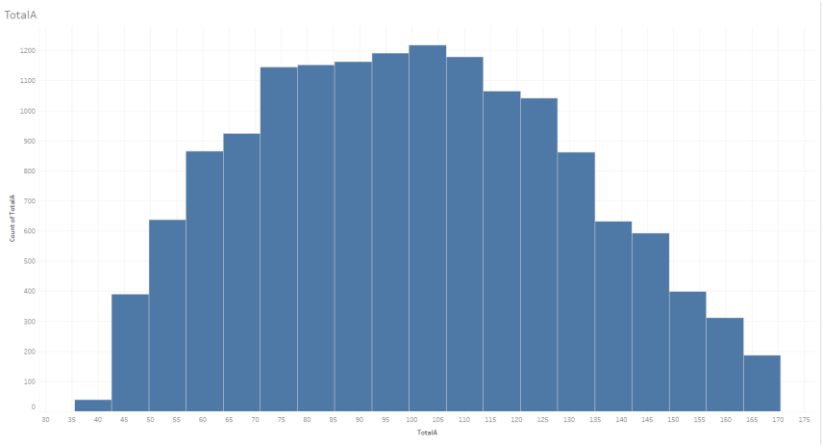


Figure 2.2.1.3 Histogram Total A

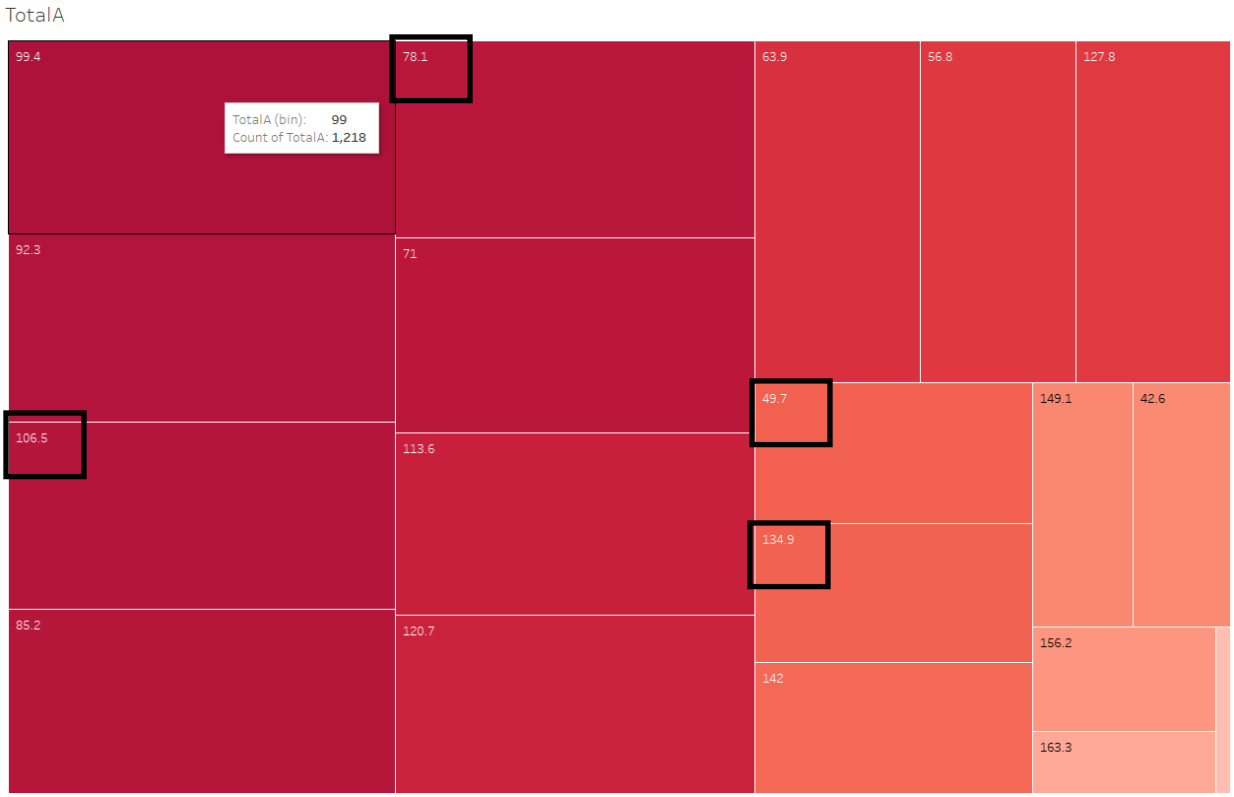


Figure 2.2.1.3 TreeMap Total A

From the treemap above, we discovered that the largest group (with 1,218 people) is having a depression level of 99. We also identify the asymmetric pattern of distribution from the treemap of Total A where the groups with similar sizes have both high and low depression levels.

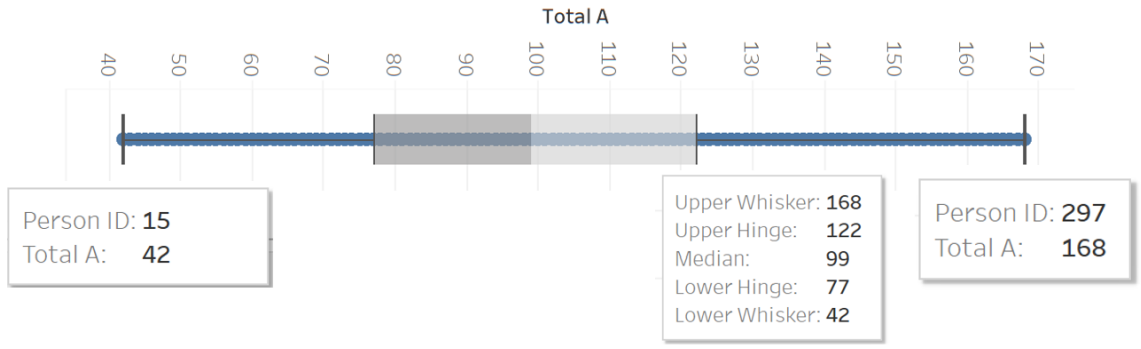


Figure 2.2.1.4 Box plot Total A

We also find out from the bar chart that the lowest level of depression is 42, the participant ID is 15; the highest level of depression is 168, the ID is 297. On average, the participants have a depression level of 99.

The codebook explained that a greater **Variable A** indicates a higher depression level. We believe it is essential to understand the meaning of each column in the beginning so that we can provide an adequate interpretation of our model output later.

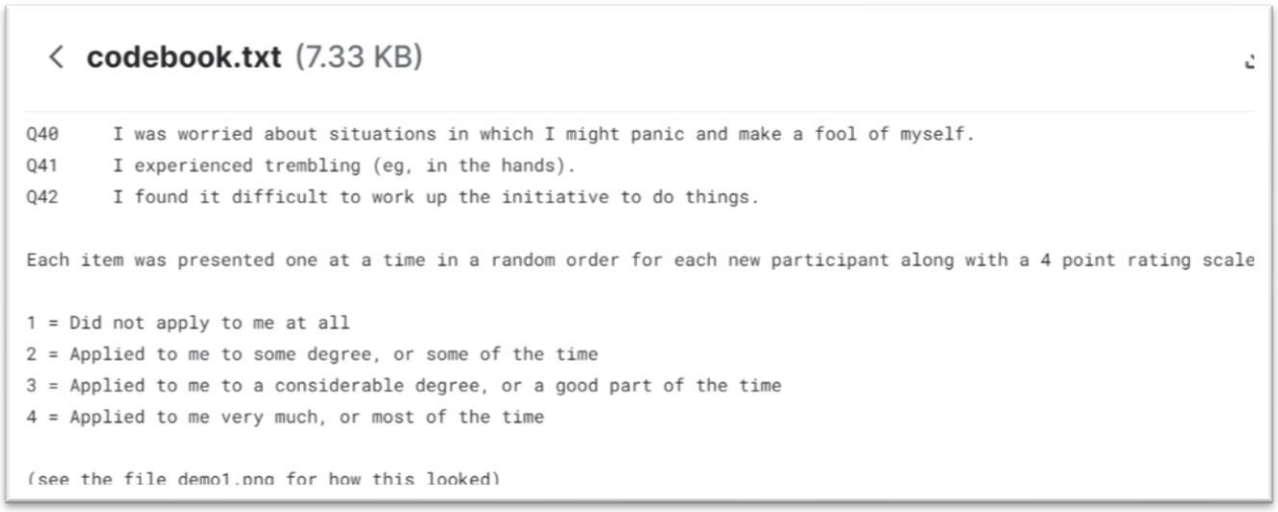


Figure 2.2.1.5 Codebook Metadata Variable A

We further explore the datatype of each column. It is important to ensure the input data is the appropriate type to fit into the data mining algorithm.

```
In [8]: DS.printSchema()
root
 |-- PersonID: string (nullable = true)
 |-- TotalA: string (nullable = true)
 |-- TotalE: string (nullable = true)
 |-- Q1A: string (nullable = true)
 |-- Q1I: string (nullable = true)
 |-- Q1E: string (nullable = true)
 |-- Q2A: string (nullable = true)
 |-- Q2I: string (nullable = true)
 |-- Q2E: string (nullable = true)
 |-- Q3A: string (nullable = true)
 |-- Q3I: string (nullable = true)
 |-- Q3E: string (nullable = true)
 |-- Q4A: string (nullable = true)
 |-- Q4I: string (nullable = true)
 |-- Q4E: string (nullable = true)
 |-- Q5A: string (nullable = true)
 |-- Q5I: string (nullable = true)
 |-- Q5E: string (nullable = true)
```

Figure 2.2.1.6 Datatype

All the columns contain textual data. We need to transform our data into numeric type because the depression scale (Variable A), answering time (Variable E) and position (Variable I) ought to be recorded in numbers. Besides, numeric data is easy for us to process and transform.

2.2.2 Personal Information

We apply a similar approach to the second dataset, **Personal Information**. There are 47 columns about the personal information of the same 15,000 participants in the first set. The data is about their characteristic, such as geographic location, personality, gender, education level, religion, race, family size.

```
In [3]: print((PI.count(), len(PI.columns)))
(15000, 47)
```

Figure 2.2.2.1 Overview Personal Information

Here is an example of the first row in the dataset shown in *Figure 2.2.2.2* below.

```
In [9]: PI.head(1)

Out[9]: [Row(PersonID='1', country='IN', source='2', introelapse='19', testelapse='167', surveyelapse='166', TIPI1='1', TIPI2='5', TIPI
3='7', TIPI4='7', TIPI5='7', TIPI6='7', TIPI7='7', TIPI8='5', TIPI9='1', TIPI10='1', VCL1='1', VCL2='0', VCL3='0', VCL4='1', VC
L5='1', VCL6='0', VCL7='1', VCL8='0', VCL9='0', VCL10='1', VCL11='0', VCL12='0', VCL13='0', VCL14='1', VCL15='1', VCL16='1', ed
ucation='2', urban='3', gender='2', engnat='2', age='16', screensize='1', uniquenetworklocation='1', hand='1', religion='12', o
rientation='1', race='10', voted='2', married='1', familysize='2', major=None)]
```

Figure 2.2.2.2 First Row Personal Information

We discovered that the participants are from 132 distinct countries around the world by aggregating the **country** column.

```
In [6]: from pyspark.sql.functions import countDistinct, avg, stddev

In [7]: PI.select(countDistinct("country").alias("Distinct country")).show()

+----------------+
|Distinct country|
+----------------+
|             132|
+----------------+
```

Figure 2.2.2.3 Country - PySpark

We map the number of participants on the map, as shown in *Figure 2.2.2.4*. The size of the dot and colour indicates the number of participants. The larger and darker the dots are, the more participants are from that country.

Figure 2.2.2.4 Country Map - Tableau

Then we created a treemap and a highlighted table in Tableau. We further discovered that that most of the participants are from Malaysia (MY 5,332) and United States (US 4,701).

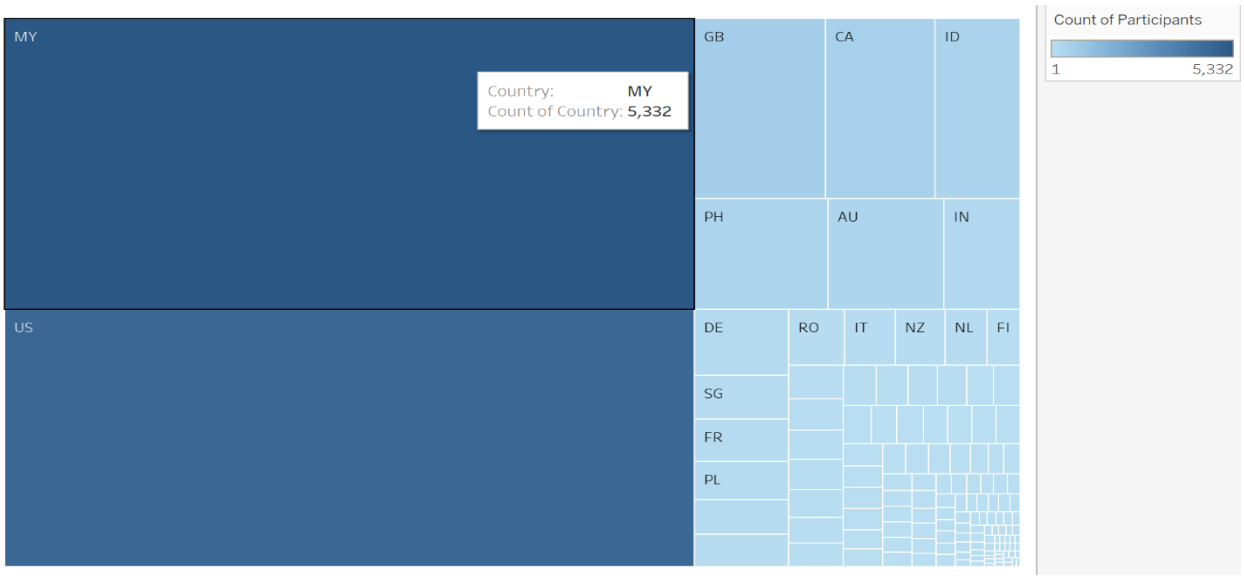


Figure 2.2.2.5 Country TreeMap– Tableau

Count..	
MY	5,332
US	4,701
GB	622
CA	524
ID	410
PH	390
AU	341
IN	222
DE	165
SG	109
FR	105
PL	95
MX	86
BR	82

Figure 2.2.2.6 Country Highlighted Table– Tableau

Then we exam the data type of each column.

```
In [10]: PI.printSchema()

root
 |-- PersonID: string (nullable = true)
 |-- country: string (nullable = true)
 |-- source: string (nullable = true)
 |-- introelapse: string (nullable = true)
 |-- testelapse: string (nullable = true)
 |-- surveyelapse: string (nullable = true)
 |-- TIPI1: string (nullable = true)
 |-- TIPI2: string (nullable = true)
 |-- TIPI3: string (nullable = true)
 |-- TIPI4: string (nullable = true)
 |-- TIPI5: string (nullable = true)
 |-- TIPI6: string (nullable = true)
 |-- TIPI7: string (nullable = true)
 |-- TIPI8: string (nullable = true)
 |-- TIPI9: string (nullable = true)
 |-- TIPI10: string (nullable = true)
 |-- VCL1: string (nullable = true)
 |-- VCL2: string (nullable = true)
 |-- VCL3: string (nullable = true)
 |-- VCL4: string (nullable = true)
 |-- VCL5: string (nullable = true)
 |-- VCL6: string (nullable = true)
 |-- VCL7: string (nullable = true)
 |-- VCL8: string (nullable = true)
 |-- VCL9: string (nullable = true)
 |-- VCL10: string (nullable = true)
 |-- VCL11: string (nullable = true)
 |-- VCL12: string (nullable = true)
 |-- VCL13: string (nullable = true)
 |-- VCL14: string (nullable = true)
 |-- VCL15: string (nullable = true)
 |-- VCL16: string (nullable = true)
 |-- education: string (nullable = true)
 |-- urban: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- engnat: string (nullable = true)
 |-- age: string (nullable = true)
 |-- screensize: string (nullable = true)
 |-- uniquenetworklocation: string (nullable = true)
 |-- hand: string (nullable = true)
 |-- religion: string (nullable = true)
 |-- orientation: string (nullable = true)
 |-- race: string (nullable = true)
 |-- voted: string (nullable = true)
 |-- married: string (nullable = true)
 |-- familysize: string (nullable = true)
 |-- major: string (nullable = true)
```

Figure 2.2.2.7 Datatype Personal Information

We discovered that similar to the **Depression Scale** dataset, all the columns contain text. By reading the metadata, we understand that the meaning and the data type each column ought to be as shown in the table below.

Column Name	Meaning	Datatype Ought to Be
PersonID	The ID number assigned to each participant	Continous
introelapse	The time spent on the introduction/landing page (in seconds)	Continous
testelapse	The time spent on all the DASS questions (should be equivalent to the time elapsed on all the individual questions combined)	Continous
surveyelapse	The time spent answering the rest of the demographic and survey questions	Continous
Personality (from 1 = Disagree strongly to 7 = Agree strongly)		
TIPI1	Extraverted, enthusiastic.	Categorical
TIPI2	Critical, quarrelsome.	Categorical
TIPI3	Dependable, self-disciplined.	Categorical
TIPI4	Anxious, easily upset.	Categorical
TIPI5	Open to new experiences, complex.	Categorical
TIPI6	Reserved, quiet.	Categorical
TIPI7	Sympathetic, warm.	Categorical
TIPI8	Disorganized, careless.	Categorical
TIPI9	Calm, emotionally stable.	Categorical
TIPI10	Conventional, uncreative.	Categorical
“In the grid below, check all the words whose definitions you are sure you know” (1 is checked, 0 means unchecked)		
VCL1	boat	Categorical
VCL2	incoherent	Categorical
VCL3	pallid	Categorical
VCL4	robot	Categorical
VCL5	audible	Categorical
VCL6	cuivocal	Categorical
VCL7	paucity	Categorical
VCL8	epistemology	Categorical
VCL9	florted	Categorical
VCL10	decide	Categorical
VCL11	pastiche	Categorical

VCL12	verdid	Categorical
VCL13	abysmal	Categorical
VCL14	lucid	Categorical
VCL15	betray	Categorical
VCL16	funny	Categorical
Characteristics		
education	"How much education have you completed?" 1=Less than high school, 2=High school, 3=University degree, 4=Graduate degree	Categorical
urban	What type of area did you live when you were a child? 1=Rural (countryside), 2=Suburban, 3=Urban (town, city)	Categorical
gender	What is your gender? 1=Male, 2=Female, 3=Other	Categorical
engnat	Is English your native language? 1=Yes, 2=No	Categorical
age	How many years old are you?	Numeric
hand	What hand do you use to write with? 1=Right, 2=Left, 3=Both	Categorical
religion	What is your religion?, 1=Agnostic, 2=Atheist, 3=Buddhist, 4=Christian (Catholic), 5=Christian (Mormon), 6=Christian (Protestant), 7=Christian (Other), 8=Hindu, 9=Jewish, 10=Muslim, 11=Sikh, 12=Other	Categorical
orientation	What is your sexual orientation? 1=Heterosexual, 2=Bisexual, 3=Homosexual, 4=Asexual, 5=Other	Categorical
race	What is your race? 10=Asian, 20=Arab, 30=Black, 40=Indigenous Australian, 50=Native American, 60=White, 70=Other	Categorical
voted	Have you voted in a national election in the past year? 1=Yes, 2=No	Categorical
married	What is your marital status? 1=Never married, 2=Currently married, 3=Previously married	Categorical
familysize	Including you, how many children did your mother have?	Numeric or Categorical
major	If you attended a university, what was your major (e.g. psychology", "English", "civil engineering")?"	Categorical
Environment		
country	ISO country code of where the user connected from	Categorical
screen size	1=device with a small screen (phone, etc), 2=device with big screen (laptop, desktop, etc.)	Categorical

uniquenetworklocation	1=only one survey from user's specific network in the dataset, 2=multiple surveys submitted from the network of this user (2 does not necessarily imply duplicate records for an individual, as it could be different students at a single school or different members of the same household; and even if 1 there still could be duplicate records from a single individual, e.g. if they took it once on their wifi and once on their phone)	Categorical
source	how the user found the test, 1=from the front page of the site hosting the survey, 2=from google, 0=other or unknown	Categorical

Table 2.2.2.8 Datatype Personal Information

We consider we should transfer the categorical data into numeric type because numeric data is easy to process and transform. We discovered that the categorical columns, except for **country** and **major**, have been recorded in numbers which allow us to transform easily. We also discovered that the major column is recorded in a format of plain text. The records in this column are unstructured and hard to process by Pyspark. Therefore, we will exclude the **major** column from our input data.

```
In [11]: PI.select('major').show()
```

```
+--------------------+
|               major|
+--------------------+
|                null|
|                null|
|                null|
|             biology|
|           Psychology|
|                null|
|Mechatronics enge...|
|               Music|
|         Psychology |
|computer programming|
|                null|
|                null|
|            Sociology|
|                 Art|
|Biology and Philo...|
|                null|
|criminal justice;...|
|       Hairdressing |
|    Computer Science|
|            Business|
+--------------------+
only showing top 20 rows
```

Figure 2.2.2.9 Major Column - Personal Information

Since we are interested in the interaction between those factors in the Personal Information dataset and the variable Total A, we need to integrate the two sets before we can obtain further insights which contribute to our data mining goals.

2.3 Explore the data

In this section, we are going to explore our dataset further, to examine the data quality and internal relationships in the dataset. Then, we will formulate our hypotheses and plan for data

preparation and transformation with the guidance of our data mining goals. Since we have not cleaned or transformed our data yet, the exploration will be bias and limited. We take the outputs from this phase mainly as a reference to form our hypothesis rather than our theory.

2.3.1 Depression Scale Exploration

As we explained in *Section 2.2* that **Variable A** represents the depression scale, **Variable E** is the answering time. We suspect that the responses with an extraordinary long answering time are inaccurate. Therefore, we examine whether there are extreme values in **TotalE**.

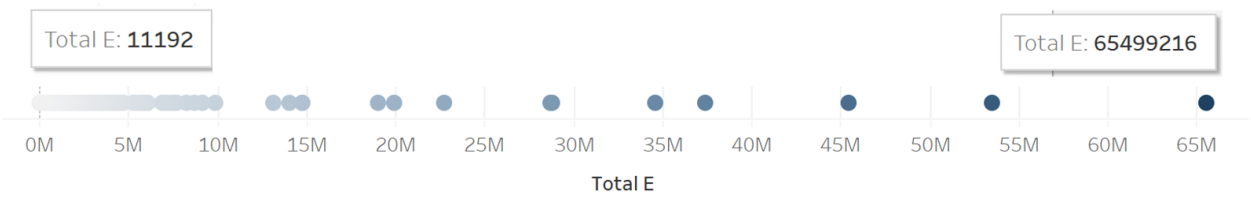


Figure 2.3.1.1 Total E Depression Scale

We identify that there are a few outliers in **TotalE**, after visualising this column in Tableau.

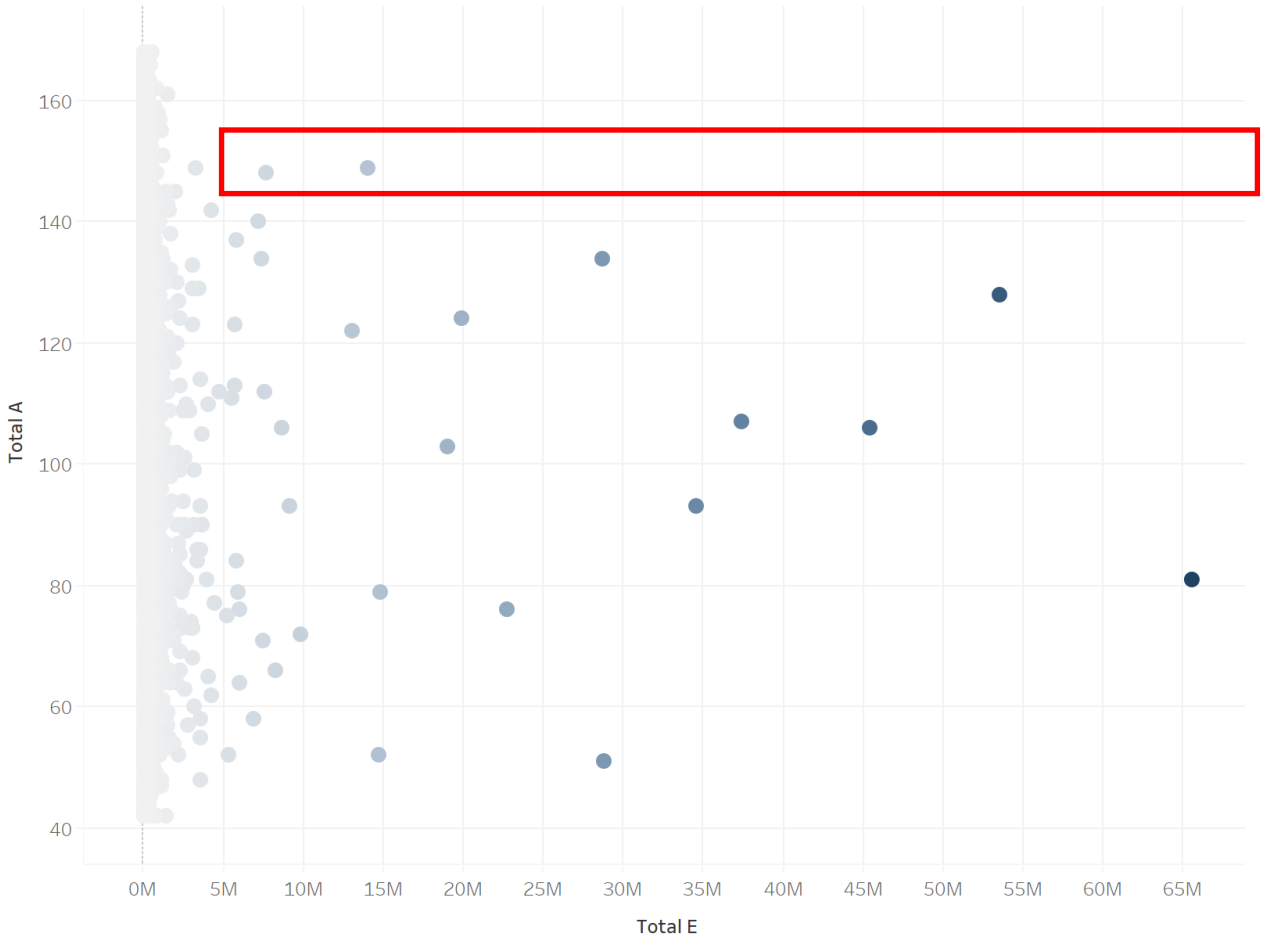


Figure 2.3.1.2 Total E vs Total A Scatter Plot Depression Scale

We also created a scatter plot of Total E vs Total A. We discover that the outliers of Total E do not correspond to the extreme values of TotalA. Nevertheless, we prefer to eliminate those extreme values, for we are suspicious about their validity.

2.3.2 Personal Information Exploration

Our data mining objective is to predict individual depression level by analysing personal characteristics. Therefore, we decide to focus on exploring the 13 characteristic columns – **education**, **urban**, **gender**, **engnat**, **age**, **hand**, **religion**, **orientation**, **race**, **voted**, **married**, **familysize**, **major**. However, the **major** column contains unstructured data so that we will focus on the other 12 columns. We created line charts for the numeric column **age** and **familysize**, and bar charts for the other categorical variables.

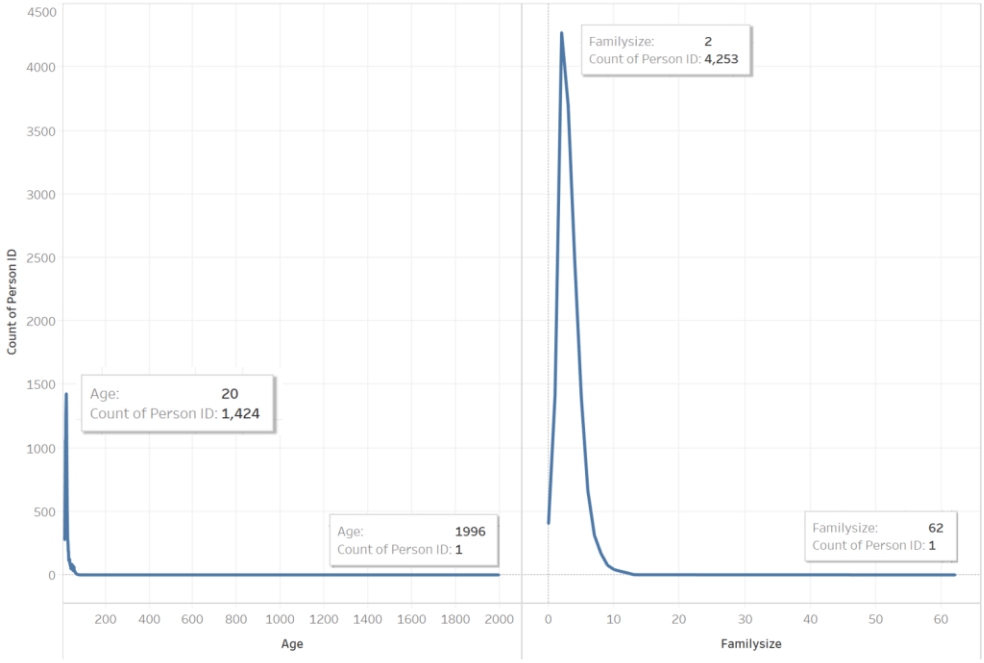


Figure 2.3.2.1 Age and Family Size

We discovered that there are outliers in **age** (age = 1996) and **familysize** (familysize = 62) that need to be removed from the data. We also discovered that the mode (the value that appears most frequently) of **age** is 20 and the mode of **familysize** is 2.

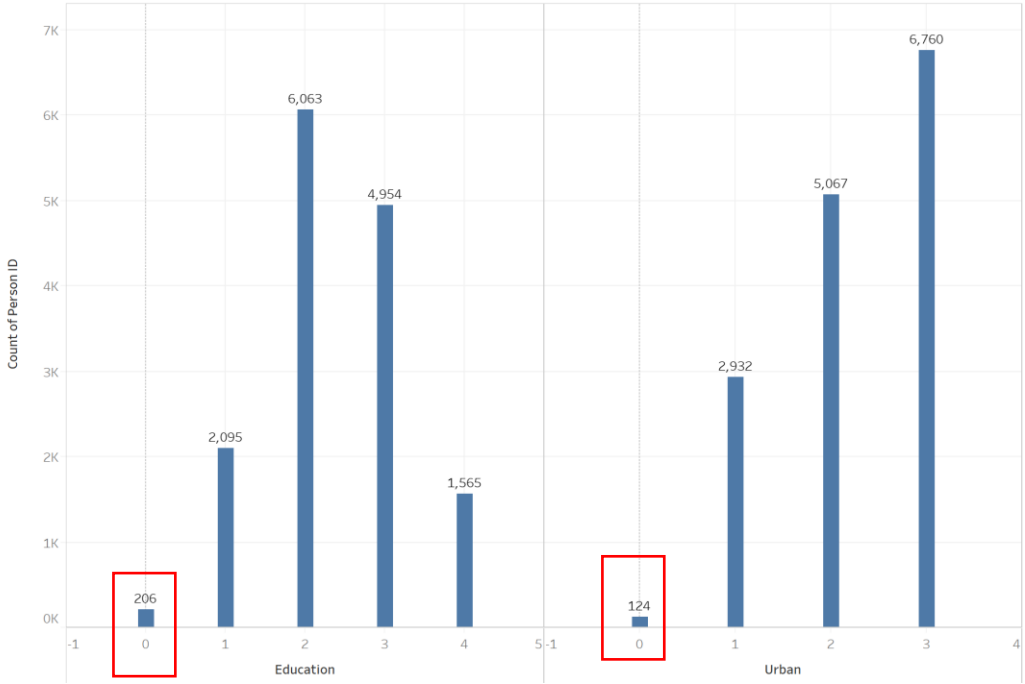


Figure 2.3.2.2 Education and Urban

We discovered that most of our participants have a high school level of education (education =2, 6,063 participants). Also, most people live in a town or city (Urban = 3, 6,760 participant). Moreover, we identify a measurement error in the dataset. According to the metadata, 0 is not assigned to any category of **education** or **urban**. Those 0 values should be removed from the dataset.

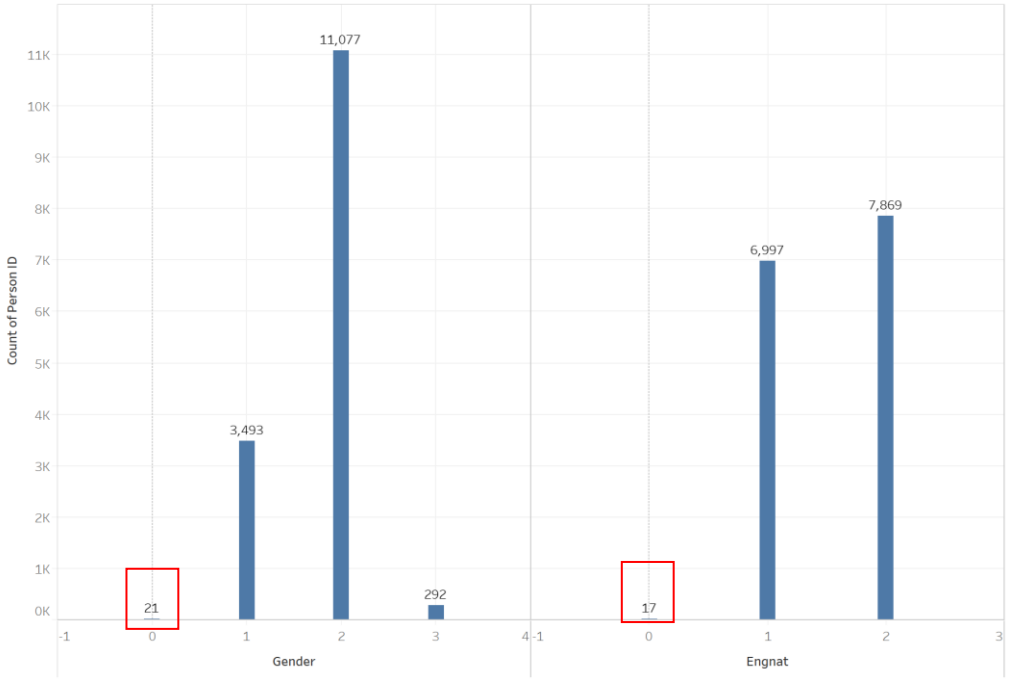


Figure 2.3.2.3 Gender and Engat

We discovered that most of our participants are female (**gender** = 2, 11,077 participants). For most people, English is not their native language (engat = 2, 7,869 participants). The same measurement errors occur in **gender** and **engant**, as well. There is value 0 in the record, which have not been assigned to any categories.

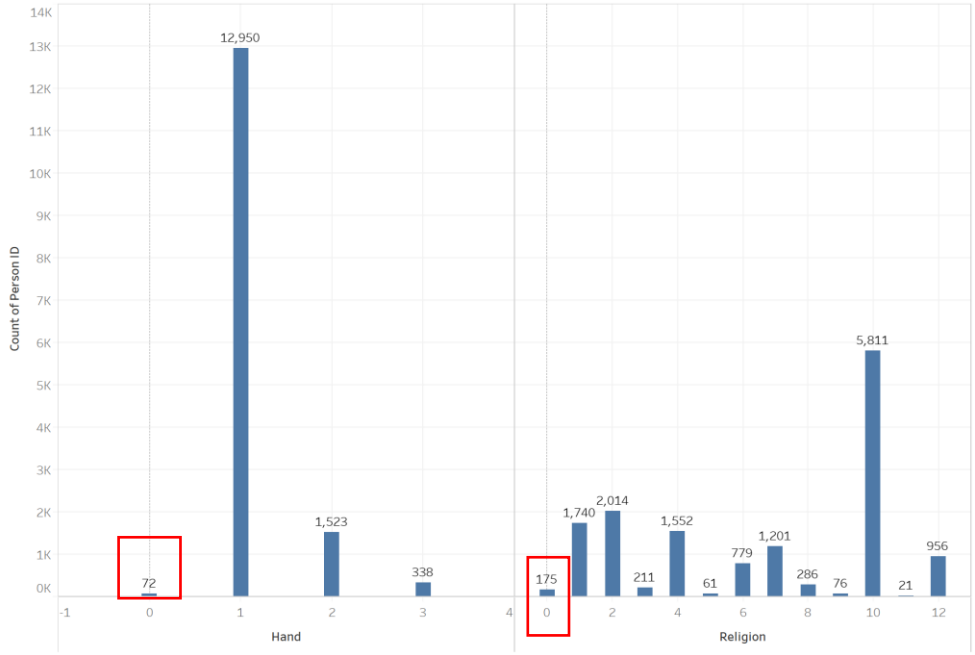


Figure 2.3.2.4 Hand and Religion

We discovered that most of the participants use the right hand to write (hand = 1, 12,950 participants), and the majority of the participants are Muslim (religion = 10, 5,811 participants).

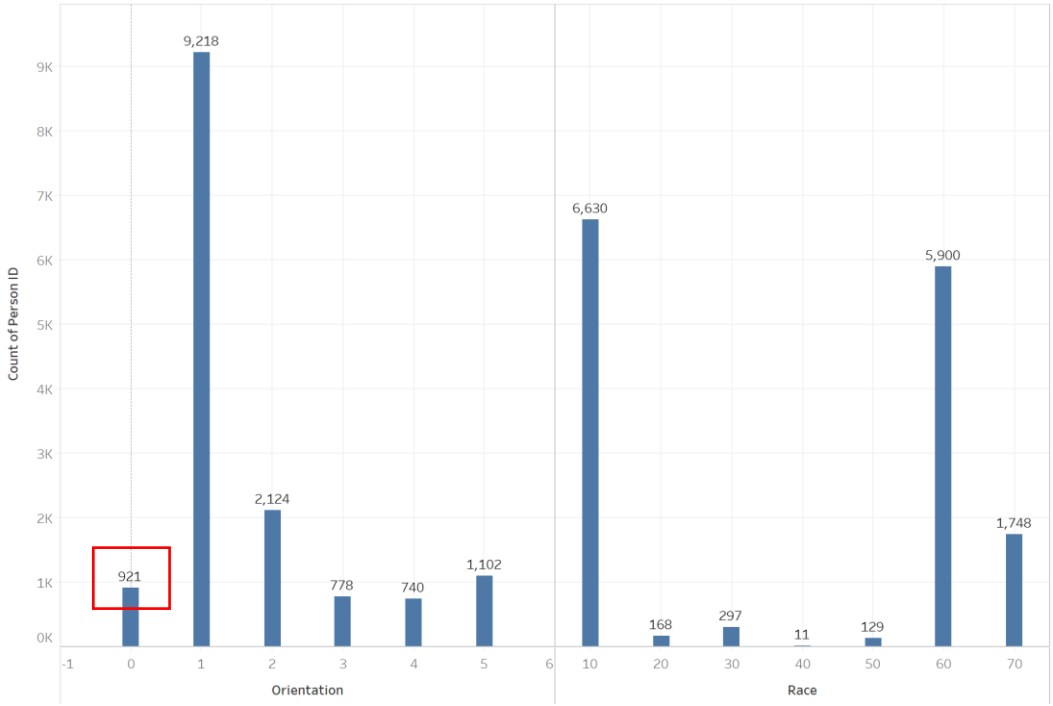


Figure 2.3.2.5 Orientation and Race

We explore that the mode of orientation is Heterosexual (orientation = 1, 9,218 participants), that of race is Asian (Race=10, 6,630 participants). There are measurement errors in the **orientation** column which need to be removed later.

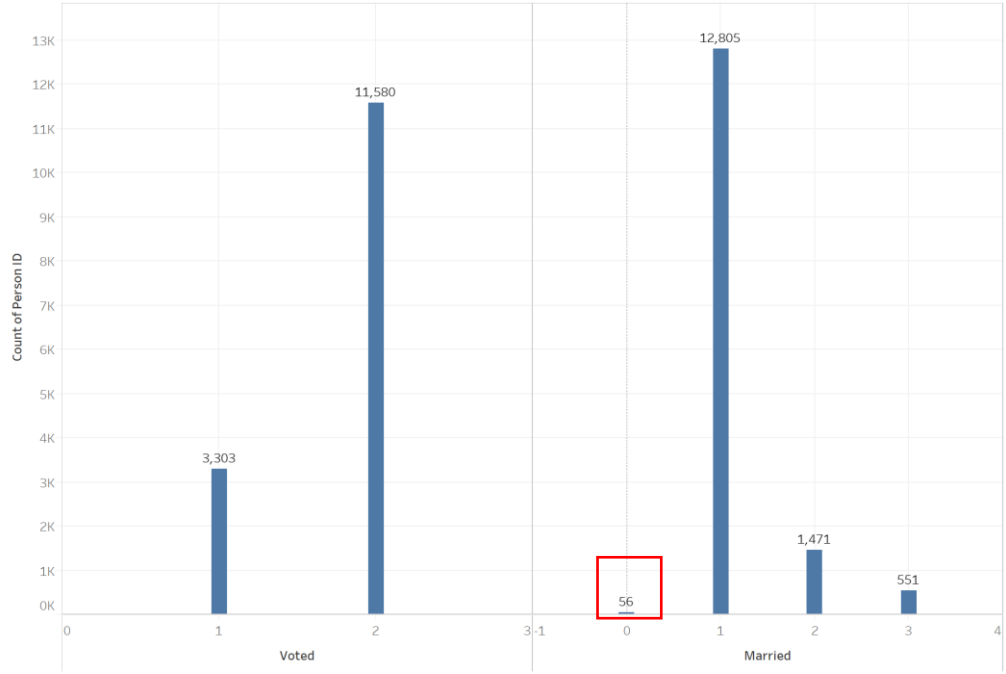


Figure 2.3.2.5 Voted and Married

We find out that the majority did not vote in a national election in the past year (voted = 2, 11,580 participants). Also, most of the participants never married before (married = 1, 12,805 participants). There is the same measurement error of value 0 in the **married** attribute.

In summary, there are outliers in the **age** and **familysize** columns, and the category settings restrict the values of the other categorical. Meaning of each column and category are summarised in *Section 2.2*. However, several columns contain the value 0, which we reckon is a measurement error. The metadata does not assign any meaning to category 0. Moreover, we notice that the count of participant varies from categories. We consider that this variation may cause bias in our output data.

At this stage, the data from the Personal Information along is not useful for achieving our goals and objectives. We can only draw insights from this part of the data after we merge the two datasets. We will then be able to explore the correlations between people's characters and their depression level. The primary data type of this set is categorical/nominal, so we reckon a clustering and classification approach will be appropriate when building our models.

Nevertheless, we can identify from the data exploration that this dataset needs to be cleaned and transform before inputting into our model.

Assumptions and Hypotheses

We assume that people with some specific characters are highly likely to have depression issues. Therefore, we first formulated a null hypothesis that it is no different in the depression level of people with different characters. We will apply an undirected knowledge discovery approach because we cannot identify any correlation between the factors and the depression level. We plan to use Clustering and classification to explore which factors can reject this null hypothesis. We will also consider the factors that rejected the null hypothesis to be the characters that correlated to depression issues, to achieve our data mining objectives. At the same time, we will include them in the depression checklist to achieve our business objectives.

2.4 Verify the data quality

The quality of the input data will determine the quality output. We must verify the data quality before the modelling process.

2.4.1 Missing Values and Extreme Values

Missing Values

First, we examined the missing values contained in the two datasets.

We import the SQL functions so that we can use the **IsNull()** and aggregation to count any missing values in our datasets. Coding is shown in *Figure 2.4.1.1* below.

```
In [27]: from pyspark import SparkContext
         from pyspark.sql import SparkSession
         from pyspark.sql.functions import *
```

Figure 2.4.1.1 Create count_not_null Function

Next, we will fit our two datasets into the function.

```
In [30]: DS.select([count(when(isnull(c), c)).alias(c) for c in DS.columns]).show()

+--------+------+------+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---
+---+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
--+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----
+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
---+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
|PersonID|TotalA|TotalE|Q1A|Q1I|Q1E|Q2A|Q2I|Q2E|Q3A|Q3I|Q3E|Q4A|Q4I|Q4E|Q5A|Q5I|Q5E|Q6A|Q6I|Q6E|Q7A|Q7I|Q7E|Q8A|Q8I|Q8E|Q9A|Q9I
|Q9E|Q10A|Q10I|Q10E|Q11A|Q11I|Q11E|Q12A|Q12I|Q12E|Q13A|Q13I|Q13E|Q14A|Q14I|Q14E|Q15A|Q15I|Q15E|Q16A|Q16I|Q16E|Q17A|Q17I|Q17E|Q1
8A|Q18I|Q18E|Q19A|Q19I|Q19E|Q20A|Q20I|Q20E|Q21A|Q21I|Q21E|Q22A|Q22I|Q22E|Q23A|Q23I|Q23E|Q24A|Q24I|Q24E|Q25A|Q25I|Q25E|Q26A|Q26I
|Q26E|Q27A|Q27I|Q27E|Q28A|Q28I|Q28E|Q29A|Q29I|Q29E|Q30A|Q30I|Q30E|Q31A|Q31I|Q31E|Q32A|Q32I|Q32E|Q33A|Q33I|Q33E|Q34A|Q34I|Q34E|Q
35A|Q35I|Q35E|Q36A|Q36I|Q36E|Q37A|Q37I|Q37E|Q38A|Q38I|Q38E|Q39A|Q39I|Q39E|Q40A|Q40I|Q40E|Q41A|Q41I|Q41E|Q42A|Q42I|Q42E|
+--------+------+------+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---
+---+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+--
--+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----
+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+-
---+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+----+
|       0|     0|     0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0|  0
|  0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|
0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|
0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|
0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|   0|
+--------+------+------+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---
```

Figure 2.4.1.2 Missing Values Depression Scale

```
In [31]: PI.select([count(when(isnull(c), c)).alias(c) for c in PI.columns]).show()

+--------+-------+------+-----------+----------+-----------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+----+
----+----+----+----+----+----+----+----+-----+-----+-----+-----+-----+-----+-----+---------+-----+------+------+---+----------+
------------------+----+--------+-----------+----+-----+-------+----------+-----+
|PersonID|country|source|introelapse|testelapse|surveyelapse|TIPI1|TIPI2|TIPI3|TIPI4|TIPI5|TIPI6|TIPI7|TIPI8|TIPI9|TIPI10|VCL1|
VCL2|VCL3|VCL4|VCL5|VCL6|VCL7|VCL8|VCL9|VCL10|VCL11|VCL12|VCL13|VCL14|VCL15|VCL16|education|urban|gender|engnat|age|screensize|
uniquenetworklocation|hand|religion|orientation|race|voted|married|familysize|major|
+--------+-------+------+-----------+----------+-----------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+----+
----+----+----+----+----+----+----+----+-----+-----+-----+-----+-----+-----+-----+---------+-----+------+------+---+----------+
------------------+----+--------+-----------+----+-----+-------+----------+-----+
|       0|    236|     0|          0|         0|           0|    0|    0|    0|    0|    0|    0|    0|    0|    0|     0|   0|
   0|   0|   0|   0|   0|   0|   0|   0|    0|    0|    0|    0|    0|    0|    0|        0|    0|     0|     0|  0|         0|
                    0|   0|       0|          0| 117|    0|      0|         0| 5342|
+--------+-------+------+-----------+----------+-----------+-----+-----+-----+-----+-----+-----+-----+-----+-----+------+----+
```

Figure 2.4.1.3 Missing Values Personal Information

After examining the two datasets using the **IsNull()** function, we obtain the outputs shown in *Figure 2.4.1.2* and *2.4.1.3*.

First, we count the is null values contained in the **DS** (Depression Scale) data frame. The output shows that all the columns have 0 rows of missing value, so we do not eliminate any row from the data frame. Then we explore the **PI** (Personal Information) data frame. The dataset contains null values in 236 rows, 117 rows and 5342 rows in the columns **country**, **race** and **major** respectively. We can drop the null values in the **country** and **race** columns only, for we plan to

eliminate the entire unstructured column **major**. In this way, we will retain more rows from the dataset and have clean data at the same time.

Extreme Values

We can not examine the extreme values with **Pyspark** before we convert the data type from string to numeric types. Therefore, we choose to explore the outliers by visualising the data on **Tableau.**

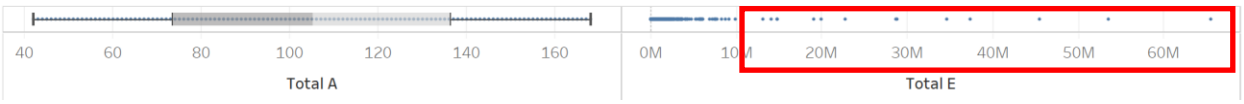


Figure 2.4.1.4 Outliers TotalA and TotalE in Depression Scale

As we mentioned before, we decided to set **TotalA** as our target variable, for it indicates the depression level of each individual. At the same time, we argue that the outliers in **TotalE** may affect the accuracy of the **TotalA**, so we should eliminate the rows which contain extreme values of **TotalE**. As shown in *Figure 2.4.1.4*, we discover that there are a few outliers in **TotalE**. We will exclude them for our input data in the data cleaning phase.

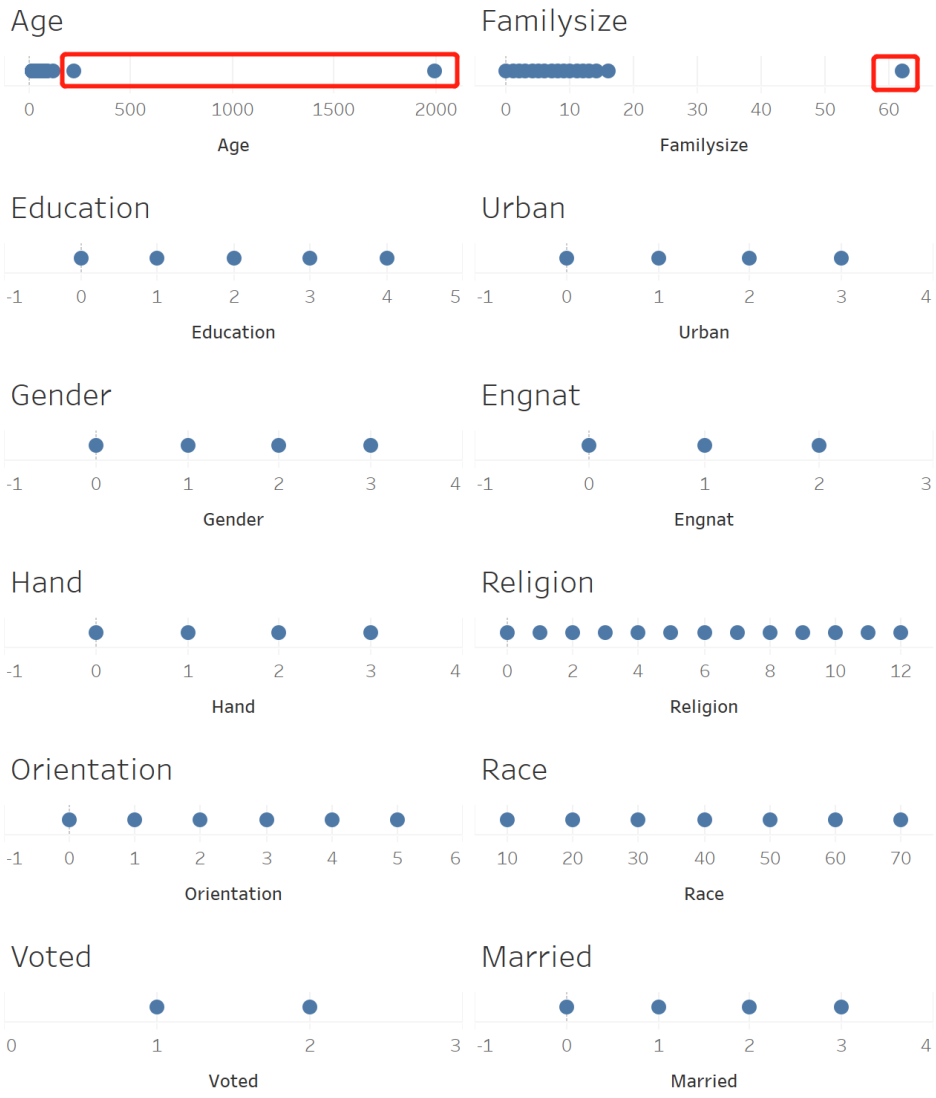


Figure 2.4.1.5 Outliers 12 characteristics in Personal Information

We discussed that according to our data mining objective, we would focus on the 13 columns that explain the characteristics of the participants. We also demonstrated that we would eliminate the **major** column. Hence, we focused on the other 12 columns and visualised their values in *Figure 2.4.1.5*. The figure illustrates that the **age** column and the **familysize** column contain a few outliers. We should eliminate those values from the dataset.

Both missing values and extreme values will affect the accuracy of our model. To resolve the problems of missing values, we can fill them with random numbers or eliminate the rows which contain any missing value. We should also eliminate the rows which contain extreme values, for they will create bias in the final output of the model. Since we have a large sample size of 15,000, we plan to eliminate the rows that contain missing or extremes. We will still have sufficient data after taking out those rows and trim our data to a more manageable size with higher intelligent density. We describe this process in detail in *Section 3.2 Clean the data* in the data preparation phase.

2.4.2 Data Errors

The **Depression Scale** dataset was automatically generated, so this is not a great worry. We assume there are no typographical errors made in entering the data.

One this other hand, the data from the **Personal Information** dataset was collected from a survey carried out after the participants have taken the test. We identify the column **major** was recorded in an unstructured text format, shown in *Figure 2.4.2.1* below. It is hard for us to process data from this field using Pyspark. Therefore, we will exclude this column from the input data in the data preparation stage.

```
+--------------------+
|               major|
+--------------------+
|                null|
|                null|
|                null|
|             biology|
|          Psychology|
|                null|
|Mechatronics enge...|
|               Music|
|         Psychology |
|computer programming|
|                null|
|                null|
|           Sociology|
|                 Art|
|Biology and Philo...|
|                null|
|criminal justice;...|
|       Hairdressing |
|    Computer Science|
|            Business|
+--------------------+
only showing top 20 rows
```

Figure 2.4.2.1 Unstructured Text - Major

2.4.3 Measurement Errors

We reckon there is not any significant measurement error in **Depression Scale**, since the computer automatically generated these records. We also visualised all the **Variable A** values in Tableau. As shown in *Figure 2.4.3.1* illustrates that the minimum value of A for each question is 1, and the maximum value is 4. This result is expected as the questionnaire constraint the

responded score in the range of 1 to 4. We reckon there is not any significant measurement error in **Variable A.**

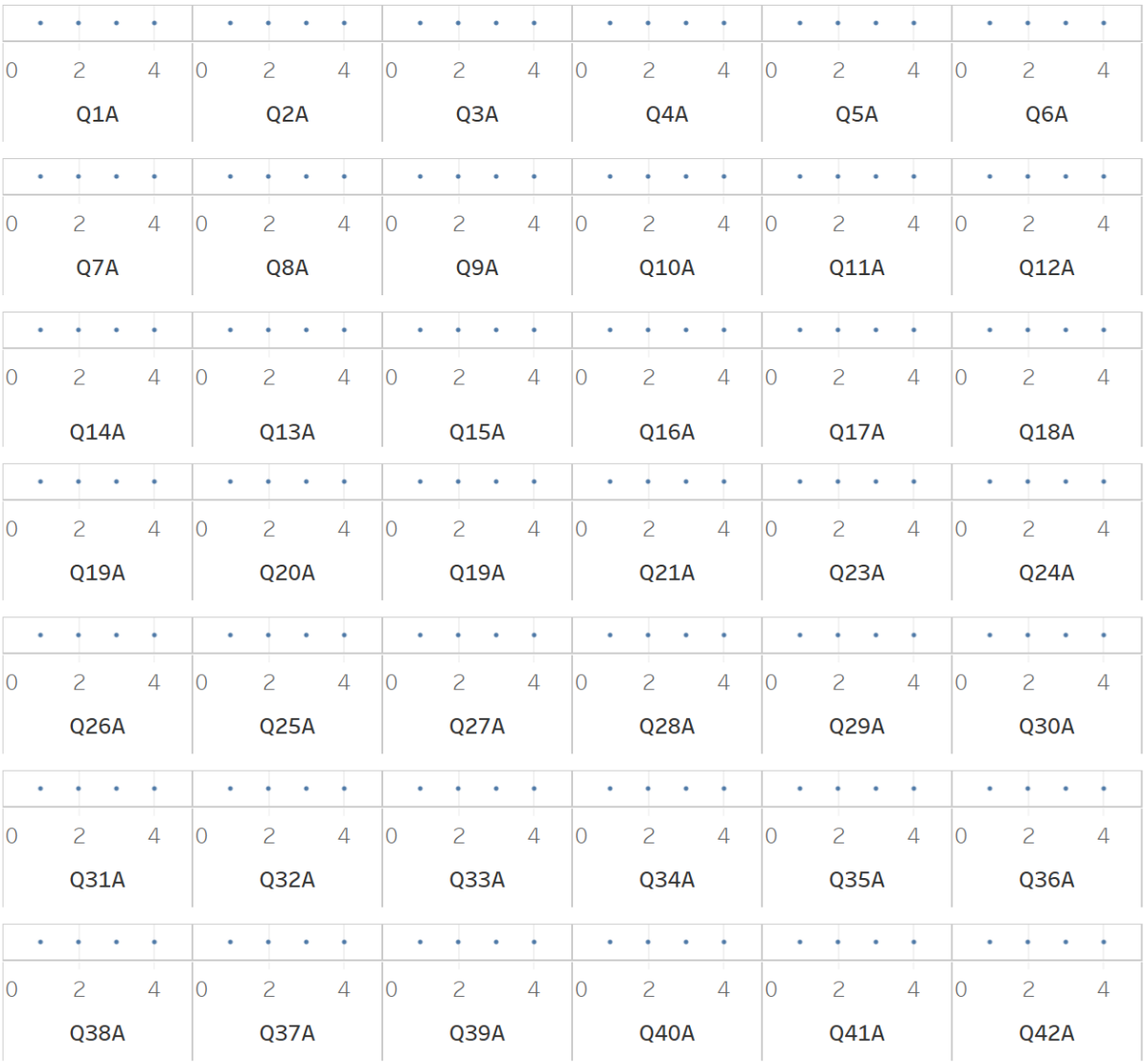


Figure 2.4.3.1 Variable A Depression Scale

However, we have identified several incorrect schemes in the **Personal Information** dataset. As discussed in *Section 2.3.2,* there are incorrect records of 0 in the dataset. From the visualisation below (*Figure 2.4.3.2*), we can also discover the same problem in the column **education, urban, gender, engnat, hand, religion, orientation.** The value zero marked in red do not represent any category according to the metadata, as explained by *Figure 2.4.3.3*.

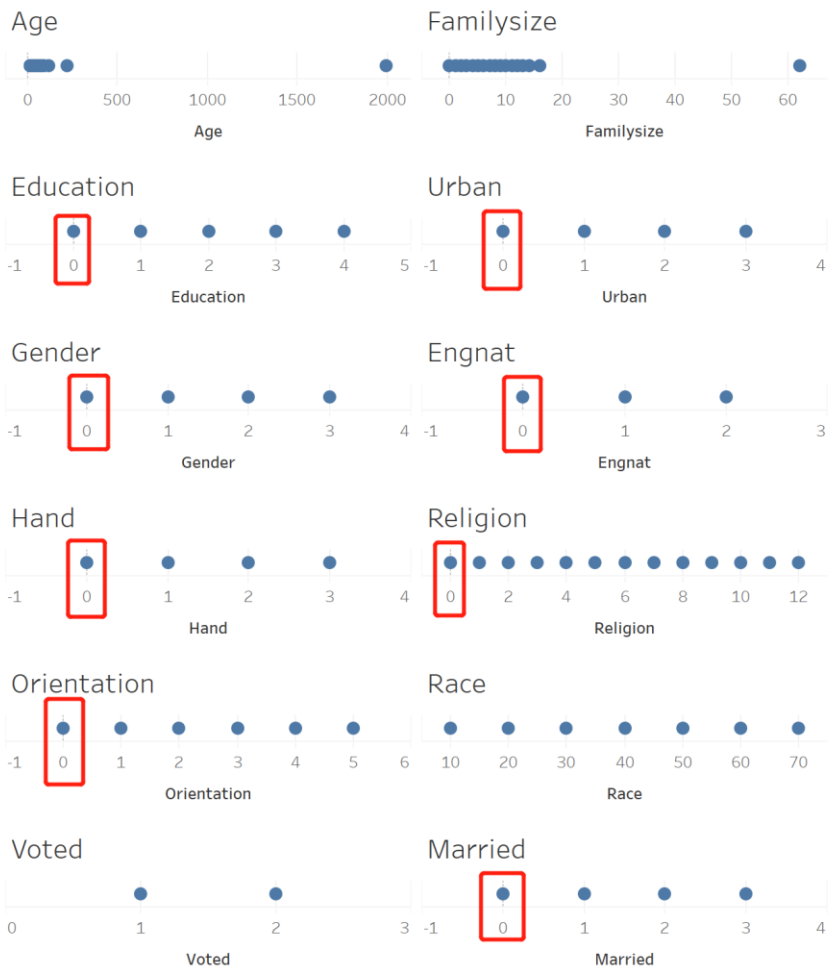


Figure 2.4.3.2 Data Measurement Errors

education	"How much education have you completed?", 1=Less than high school, 2=Hig
urban	"What type of area did you live when you were a child?", 1=Rural (country sid
gender	"What is your gender?", 1=Male, 2=Female, 3=Other
engnat	"Is English your native language?", 1=Yes, 2=No
age	"How many years old are you?"
hand	"What hand do you use to write with?", 1=Right, 2=Left, 3=Both
religion	"What is your religion?", 1=Agnostic, 2=Atheist, 3=Buddhist, 4=Christian (Catholic), 5=
orientation	"What is your sexual orientation?", 1=Heterosexual, 2=Bisexual, 3=Homosexu
race	"What is your race?", 10=Asian, 20=Arab, 30=Black, 40=Indigenous Australia
voted	"Have you voted in a national election in the past year?", 1=Yes, 2=No
married	"What is your marital status?", 1=Never married, 2=Currently married, 3=Prev
familysize	"Including you, how many children did your mother have?"

Figure 2.4.3.3 Data Measurement Errors

All the scheme of the fields marked in *Figure 2.4.3.3*, should start from 1 rather than 0. We assume the 0 should be null values, but incorrectly recorded as 0 in the dataset. We will take out those rows to prevent our models from producing inaccurate outputs.

Data Preparation

Data preparation is crucial to building a reliable model in the later steps. As it says in the concept GIGO (garbage in, garbage out), the quality of input data determines the quality of output data. Therefore, to achieve both our data mining objective and business objective, we will prepare the data with the following tasks:

- Selecting relevant data as input data for our model
- Removing or replacing blank or missing values
- Removing outliers
- Construct new data that facilitate the construction of model
- Merging the records from the data set **Depression Scale** and **Personal Information**
- Format the data into appropriate datatype

Overall, the data preparation process will increase the intelligence density of our data, as it will be more useful after it has been scrubbed and integrated. We will further transform the data, carry out data mining methods and algorithms and interpret the data. With these processes, we can bring the initial massive data to be condensed and useful knowledge.

3.1 Select the data

The first data mining goal of our study is to discover and describe the correlations between the characters of people and depression level. To achieve this, we will select data about depression level from the **Depression Scale** data set and attributes of people from the **Personal Information** dataset.

Depression Scale

The final depression scale of each person is recorded as **TotalA** in the **Depression Scale** data set. We choose the column of **TotalA** as the independent variable of our model. We want to create a new data frame named **Target** that only includes **TotalA** and the linking key **PersonID**. However, as we discussed in *Section 2.4*, we suspect the outliers of **TotalE** may affect the corresponding values of **TotalA**. Therefore, we would include **PersonalID**, **TotalA** and **TotalE** in the **Target** data frame, as shown in *Figure 3.1.1* below.

```
In [29]: Target = DS.select('PersonID','TotalA','TotalE')

In [30]: Target.show()
```

```
+--------+------+--------+
|PersonID|TotalA|  TotalE|
+--------+------+--------+
|       1|   143|  157622|
|       2|   110|  168927|
|       3|   110|  270194|
|       4|    91|  253531|
|       5|   143|  163402|
|       6|    73|  348041|
|       7|   106|45458075|
|       8|    56|  231704|
|       9|   149|  192088|
|      10|   146|  116857|
|      11|    82|  156695|
|      12|   108|  111826|
|      13|    89|  112454|
|      14|    87|  174640|
|      15|    42|  137197|
|      16|    78|  154031|
|      17|    97|  142700|
|      18|    71|   98718|
|      19|    72|  127739|
|      20|    87|  265310|
+--------+------+--------+
only showing top 20 rows
```

Figure 3.1.1 Target

Personal Information

There are 46 columns in the **Personal Information** dataset, which record the personal details of each participant. Since those columns are not as highly correlated as **Variable A** and **Variable E**, we assume the extreme values in a column would not affect the accuracy of the other columns. Hence, we decide to extract the useful columns as our independent variables, before the data cleaning to reduce the amount of dirty data. Among the 46 columns, three describes the time durations spent on different sections of the DASS test (introelapse, testelapse, surveyelapse). Another group of fields starting with “TIPT” is relevant to a personality test. The attributes staring with “VCL” are the participants’ response to identifying a list of words. Four other columns record the technical information of the participants’ devices (country, screensize, uniquenetworklocation, source). Finally, there are 12 fields related to the characteristics of the participants. We believe these 12 fields are most relevant to this study; they are:

1. education
2. urban
3. gender
4. engnat
5. age
6. hand
7. religion
8. orientation
9. race
10. voted
11. married
12. familysize

The meaning and measuring of these factors are explained in ***Table 2.2.2.8*** in the data exploration phase. Moreover, as we explored in *Section 2.4.2*, we need to clean and transform all these columns. All the12 columns are recorded as strings but should be transferred into numeric data. As shown in the discussed in *Section 2.4*, there are a few outliers, and measurement errors need to be cleaned before the data mining process. Moreover, the “major” column contains unstructured text that cannot be easily processed. Therefore, we will select these 12 attributes listed above as potential predictors of our model and examine their correlations with the depression level. We expect some attributes will reject our null hypothesis, which claims that there is no significant relationship between people’s characters and their depression level.

```
In [32]: Predictors = PI.select('PersonID','education','urban','gender','engnat','age','hand',
                'religion','orientation','race','voted','married','familysize')

In [33]: Predictors.show()
```

```
+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
|PersonID|education|urban|gender|engnat|age|hand|religion|orientation|race|voted|married|familysize|
+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
|       1|        2|    3|     2|     2| 16|   1|      12|          1|  10|    2|      1|         2|
|       2|        2|    3|     2|     1| 16|   2|       7|          0|  70|    2|      1|         4|
|       3|        2|    3|     2|     2| 17|   1|       4|          3|  60|    1|      1|         3|
|       4|        1|    3|     2|     1| 13|   2|       4|          5|  70|    2|      1|         5|
|       5|        3|    2|     2|     2| 19|   3|      10|          1|  10|    2|      1|         4|
|       6|        2|    3|     2|     2| 20|   1|       4|          1|  70|    2|      1|         4|
|       7|        2|    3|     2|     2| 17|   1|       7|          2|  60|    2|      1|         4|
|       8|        4|    2|     2|     2| 29|   1|       2|          2|  60|    1|      1|         2|
|       9|        2|    3|     1|     2| 16|   1|      12|          2|  70|    2|      1|         4|
|      10|        1|    1|     2|     2| 18|   1|       2|          2|  60|    2|      1|         3|
|      11|        1|    2|     1|     1| 15|   1|       6|          1|  60|    2|      1|         1|
|      12|        2|    1|     2|     1| 18|   1|       6|          1|  60|    2|      1|         2|
|      13|        3|    0|     2|     1| 20|   1|       1|          1|  60|    2|      1|         2|
|      14|        4|    2|     1|     2| 31|   1|      12|          1|  60|    2|      1|         5|
|      15|        3|    2|     1|     1| 34|   1|       1|          1|  60|    1|      3|         2|
|      16|        2|    2|     2|     1| 17|   1|       1|          2|  60|    2|      1|         1|
|      17|        2|    2|     2|     2| 19|   1|       7|          1|  60|    2|      1|         2|
|      18|        3|    3|     2|     1| 18|   1|      12|          1|  60|    2|      1|         5|
|      19|        2|    1|     1|     1| 19|   3|       2|          1|  60|    2|      1|         1|
|      20|        2|    3|     2|     1| 19|   1|       4|          2|  50|    2|      1|         2|
+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
only showing top 20 rows
```

Figure 3.1.2 Predictors

As illustrated in *Figure 3.1.2* above, we created a new data frame named **Predictors** which include the 12 columns selected and the linking key **PersonID**.

We have greatly reduced the amount of data we need to clean and process after the data selection. Now we can work on the new data frames **Target** and **Predictors** instead of the original datasets.

```
print((DS.count(), len(DS.columns)))
(15000, 129)

print((PI.count(), len(PI.columns)))
(15000, 47)

print((Target.count(), len(Target.columns)))
(15000, 3)

print((Predictors.count(), len(Predictors.columns)))
(15000, 13)
```

Figure 3.1.3 Comparison of Data Frames

There are 15,000 rows and 176 columns from the two original datasets. After creating the new datasets, we only need to process 15,000 rows and 16 columns of data.

3.2 Clean the data

3.2.1 Missing Values and Extreme Values

In *Section 2.4*, we examine the **Depression Scale** dataset with the **isnull()** function and boxplot (*Figure 2.4.1.1* and *2.4.1.3*). We discovered that there is not any null value in this dataset but a few outliers. Next, we examined the **Personal Information** dataset that there are 237, 117 and 5393 null values in the columns **country**, **voted** and **major** respectively (*Figure 2.4.1.2*). Also, there are outliers in **age** and **familysize** columns in this dataset as illustrated by *Figure 2.4.1.5*.

Since we have obtained a large sample size of 15,000 participants, we decide to eliminate those columns that contain null values and outliers.

Missing Values

We created two new data frames named **Target_2** and **Predictors_2** to contain the datasets after eliminating the null values from the original data frames.

```
In [39]: Target_2 = Target.na.drop()
         Predictors_2 = Predictors.na.drop()

In [40]: print((Target_2.count(), len(Target_2.columns)))
         (15000, 3)

In [41]: print((Predictors_2.count(), len(Predictors_2.columns)))
         (14883, 13)
```

Figure 3.2.1.1 Eliminating Missing Values

As illustrated by *Figure 3.2.1*, the **Target_2** still have 15,000 rows after eliminating the null values. At the same time, 117 rows have eliminated from the **Personal Information** ad 14,883 rows remains in the dataset.

Extreme Values

As we explored in *Section 2.*4, all the data in the two datasets, **Depression Scale** and **Personal Information**, are recorded in a text format (string). To take out the outliers, we must transform the data into numeric forms.

We will first transfer the columns in **Target_2** (**PersonID**, **TotalA** and **TotalE)**. **TotalA** should be integers because this column is the sum of 42 responses that range for 1 to 4. We will transform **TotalE** into integer as well because this column do not have decimals and it has already been recorded in millisecond according to the metadata. **PersonID** should be integers as well since it is the ID number assigned to each participant.

```
+--------+------+--------+
|PersonID|TotalA|  TotalE|
+--------+------+--------+
|       1|   143|  157622|
|       2|   110|  168927|
|       3|   110|  270194|
|       4|    91|  253531|
|       5|   143|  163402|
|       6|    73|  348041|
|       7|   106|45458075|
|       8|    56|  231704|
|       9|   149|  192088|
|      10|   146|  116857|
|      11|    82|  156695|
|      12|   108|  111826|
|      13|    89|  112454|
|      14|    87|  174640|
|      15|    42|  137197|
|      16|    78|  154031|
|      17|    97|  142700|
|      18|    71|   98718|
|      19|    72|  127739|
|      20|    87|  265310|
+--------+------+--------+
only showing top 20 rows
```

Figure 3.2.1.2 PersonalID, Variable A, Variable E

We examined the data in **Target_2** were recorded as text (or string). Next, we imported the **cast()** function SQL functions. Then, we change all the columns in **Target_2** into integers as illustrated in *Figure 3.2.3* below.

```
Target_2.printSchema()
```

```
root
 |-- PersonID: string (nullable = true)
 |-- TotalA: string (nullable = true)
 |-- TotalE: string (nullable = true)
```

```
from pyspark.sql.functions import col
Target_2 = Target_2.select(*(col(c).cast("integer").alias(c) for c in Target_2.columns))
```

Figure 3.2.1.3 Change Datatype – Target_2

We would conduct a similar approach to the **Predictors_2** data frame. Reading the metadata, we understand that all the columns should be integers.

education	"How much education have you completed?", 1=Less than high school, 2=High school, 3=University degree, 4=Gradu
urban	"What type of area did you live when you were a child?", 1=Rural (country side), 2=Suburban, 3=Urban (town, city)
gender	"What is your gender?", 1=Male, 2=Female, 3=Other
engnat	"Is English your native language?", 1=Yes, 2=No
age	"How many years old are you?"
hand	"What hand do you use to write with?", 1=Right, 2=Left, 3=Both
religion	"What is your religion?", 1=Agnostic, 2=Atheist, 3=Buddhist, 4=Christian (Catholic), 5=Christian (Mormon), 6=Christian (Protes
orientation	"What is your sexual orientation?", 1=Heterosexual, 2=Bisexual, 3=Homosexual, 4=Asexual, 5=Other
race	"What is your race?", 10=Asian, 20=Arab, 30=Black, 40=Indigenous Australian, 50=Native American, 60=White, 70=(
voted	"Have you voted in a national election in the past year?", 1=Yes, 2=No
married	"What is your marital status?", 1=Never married, 2=Currently married, 3=Previously married
familysize	"Including you, how many children did your mother have?"

Figure 3.2.1.4 Meta Data Predictors_2

We explored all the categorical columns have already been recorded in numbers so that we do not need to assign numbers to each category ourselves. We examined the columns in

Predictors_2 are recorded as text (or string) and transform them into integers, as shown in *Figure 3.2.5*.

```
Predictors_2.printSchema()
```

```
root
 |-- PersonID: string (nullable = true)
 |-- education: string (nullable = true)
 |-- urban: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- engnat: string (nullable = true)
 |-- age: string (nullable = true)
 |-- hand: string (nullable = true)
 |-- religion: string (nullable = true)
 |-- orientation: string (nullable = true)
 |-- race: string (nullable = true)
 |-- voted: string (nullable = true)
 |-- married: string (nullable = true)
 |-- familysize: string (nullable = true)
```

```
Predictors_2 = Predictors_2.select(*(col(c).cast("integer").alias(c) for c in Predictors_2.columns))
```

Figure 3.2.1.5 PersonalID, Variable A, Variable E – Change Datatype Predictors_2

We examine the two frames and find that all the columns are recorded as integers now.

```
Target_2.printSchema()
Predictors_2.printSchema()
```

```
root
 |-- PersonID: integer (nullable = true)
 |-- TotalA: integer (nullable = true)
 |-- TotalE: integer (nullable = true)

root
 |-- PersonID: integer (nullable = true)
 |-- education: integer (nullable = true)
 |-- urban: integer (nullable = true)
 |-- gender: integer (nullable = true)
 |-- engnat: integer (nullable = true)
 |-- age: integer (nullable = true)
 |-- hand: integer (nullable = true)
 |-- religion: integer (nullable = true)
 |-- orientation: integer (nullable = true)
 |-- race: integer (nullable = true)
 |-- voted: integer (nullable = true)
 |-- married: integer (nullable = true)
 |-- familysize: integer (nullable = true)
```

Figure 3.2.1.6 Target_2 and Predictors_2 Data Type

Now we can detect and remove the outliers from the datasets.

We choose the Interquartile Range method (IQR) developed by the American mathematician John Widler Turk. This method divide dataset in four quartiles and each quartile contain 25% of the data from the smallest to the greatest. The range of IQR equal to the difference between the 75% percentile(Q3) and the 25% percentile (Q1). We define the lower bound of acceptable data as of Q1 – 1.5*IQR, and the upper bound as Q3 + 1.5*IQR. We consider the data points outside this range to be outliers and will eliminate them in the latter steps.

First, we create an IQR range and bounds of the data frame **Target_2**. We define the columns to be examined as **TotalA** and **TotalE,** as shown in *Figure 3.2.7*. Then we calculate the IQR as

and the bounds.

```
cols = ['TotalA', 'TotalE']
bounds = {}
for col in cols:
    quantiles = Target_2.approxQuantile(col, [0.25, 0.75], 0.05)
    IQR = quantiles[1] - quantiles[0]
    bounds[col] = [quantiles[0]-1.5*IQR, quantiles[1]+1.5*IQR]
bounds
```

```
{'TotalA': [16.5, 180.5], 'TotalE': [8131.0, 390595.0]}
```

Figure 3.2.1.7 IQR and Bounds

Then we compare the data in **TotalA** and **TotalE** with the bounds. We created a data frame, Target_2_outliers, as shown in the table shown by *Table 3.2.9* using the codes illustrated in *Figure 3.2.7*. We created the "_o" columns for each field, and if the value is "true" then the corresponding column is an outlier. For example, the **TotalE** of 45,458,075 (with **PersonID 7**) is an outlier.

```
Target_2_outliers = Target_2.select( "*",
    *[
    (
        (Target_2[c] < bounds[c][0])|
        (Target_2[c] > bounds[c][1])
    ).alias(c+'_o') for c in cols
])
Target_2_outliers.show()
```

Figure 3.2.1.8 Target_2_outliers

PersonID	TotalA	TotalE	TotalA_o	TotalE_o
1	143	157622	false	false
2	110	168927	false	false
3	110	270194	false	false
4	91	253531	false	false
5	143	163402	false	false
6	73	348041	false	false
7	106	45458075	false	true
8	56	231704	false	false
9	149	192088	false	false
10	146	116857	false	false
11	82	156695	false	false
12	108	111826	false	false
13	89	112454	false	false
14	87	174640	false	false
15	42	137197	false	false
16	78	154031	false	false
17	97	142700	false	false
18	71	98718	false	false
19	72	127739	false	false
20	87	265310	false	false

only showing top 20 rows

Table 3.2.1.9 Target_2_outliers Table

Next, we will use the **filter()** function to create data frame **Target_3** which exclude the outliers. However, the **filter()** function can only process numeric and string while the "_o" columns are in boolean type. To filter out the outliers, we need to transform the "_o" columns into strings.

```
Target_2_outliers.printSchema()
```

```
root
 |-- PersonID: integer (nullable = true)
 |-- TotalA: integer (nullable = true)
 |-- TotalE: integer (nullable = true)
 |-- TotalA_o: boolean (nullable = true)
 |-- TotalE_o: boolean (nullable = true)
```

Figure 3.2.1.10 Target_2_outliers Datatype - 1

```
from pyspark.sql.functions import col
Target_2_outliers = Target_2_outliers.select(*(col(c).cast("string")
```

```
Target_2_outliers.printSchema()
```

```
root
 |-- PersonID: string (nullable = true)
 |-- TotalA: string (nullable = true)
 |-- TotalE: string (nullable = true)
 |-- TotalA_o: string (nullable = true)
 |-- TotalE_o: string (nullable = true)
```

Figure 3.2.1.11 Target_2_outliers Datatype – 2

Now we can filter out any rows that contain outliers, i.e. the "_o" values are "true". The new dataset **Target_3** is now free from the outliers.

```
Target_3 = Target_2_outliers.filter((Target_2_outliers['TotalA_o']
Target_3.show()
```

```
+--------+------+------+
|PersonID|TotalA|TotalE|
+--------+------+------+
|       1|   143|157622|
|       2|   110|168927|
|       3|   110|270194|
|       4|    91|253531|
|       5|   143|163402|
|       6|    73|348041|
|       8|    56|231704|
|       9|   149|192088|
|      10|   146|116857|
|      11|    82|156695|
|      12|   108|111826|
|      13|    89|112454|
|      14|    87|174640|
|      15|    42|137197|
|      16|    78|154031|
|      17|    97|142700|
|      18|    71| 98718|
|      19|    72|127739|
|      20|    87|265310|
|      21|    72|181710|
+--------+------+------+
only showing top 20 rows
```

Figure 3.2.1.12 Target_3

We will apply the same method to the **Predictors_2** data frame and examine the 12 predictors. As we explained in the data exploration phase, only the columns **age** and **familysize** are numeric data, and the other columns are categorical. Therefore, we only need to eliminate the

outliers in these two columns.

```
cols = ['age','familysize']
bounds = {}
for col in cols:
    quantiles = Predictors_2.approxQuantile(col, [0.25, 0.75], 0.05)
    IQR = quantiles[1] - quantiles[0]
    bounds[col] = [quantiles[0]-1.5*IQR, quantiles[1]+1.5*IQR]
bounds
Predictors_2_outliers = Predictors_2.select( "*",
    *[
    (
        (Predictors_2[c] < bounds[c][0])|
        (Predictors_2[c] > bounds[c][1])
    ).alias(c+'_o') for c in cols
])
Predictors_2_outliers.show()
```

```
+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+-----+------------+
|PersonID|education|urban|gender|engnat|age|hand|religion|orientation|race|voted|married|familysize|age_o|familysize_o|
+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+-----+------------+
|       1|        2|    3|     2|     2| 16|   1|      12|          1|  10|    2|      1|         2|false|       false|
|       2|        2|    3|     2|     1| 16|   2|       7|          0|  70|    2|      1|         4|false|       false|
|       3|        2|    3|     2|     2| 17|   1|       4|          3|  60|    1|      1|         3|false|       false|
|       4|        1|    3|     2|     1| 13|   2|       4|          5|  70|    2|      1|         5|false|       false|
|       5|        3|    2|     2|     2| 19|   3|      10|          1|  10|    2|      1|         4|false|       false|
|       6|        2|    3|     2|     2| 20|   1|       4|          1|  70|    2|      1|         4|false|       false|
|       7|        2|    3|     2|     2| 17|   1|       7|          2|  60|    2|      1|         4|false|       false|
|       8|        4|    2|     2|     2| 29|   1|       2|          2|  60|    1|      1|         2|false|       false|
|       9|        2|    3|     1|     2| 16|   1|      12|          2|  70|    2|      1|         4|false|       false|
|      10|        1|    1|     2|     2| 18|   1|       2|          2|  60|    2|      1|         3|false|       false|
|      11|        1|    2|     1|     1| 15|   1|       6|          1|  60|    2|      1|         1|false|       false|
|      12|        2|    1|     2|     1| 18|   1|       6|          1|  60|    2|      1|         2|false|       false|
|      13|        3|    0|     2|     1| 20|   1|       1|          1|  60|    2|      1|         2|false|       false|
|      14|        4|    2|     1|     2| 31|   1|      12|          1|  60|    2|      1|         5| true|       false|
|      15|        3|    2|     1|     1| 34|   1|       1|          1|  60|    1|      3|         2| true|       false|
|      16|        2|    2|     2|     1| 17|   1|       1|          2|  60|    2|      1|         1|false|       false|
|      17|        2|    2|     2|     2| 19|   1|       7|          1|  60|    2|      1|         2|false|       false|
|      18|        3|    3|     2|     1| 18|   1|      12|          1|  60|    2|      1|         5|false|       false|
|      19|        2|    1|     1|     1| 19|   3|       2|          1|  60|    2|      1|         1|false|       false|
|      20|        2|    3|     2|     1| 19|   1|       4|          2|  50|    2|      1|         2|false|       false|
+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+-----+------------+
only showing top 20 rows
```

Figure 3.2.1.13 Outliers Age and Familysize

We use the **age_o** and **familysize_o** columns to indicate whether the value of age and family size are outliers. If it is “true”, then the value is an outlier.

```
Predictors_2_outliers.printSchema()
```

```
root
 |-- PersonID: integer (nullable = true)
 |-- education: integer (nullable = true)
 |-- urban: integer (nullable = true)
 |-- gender: integer (nullable = true)
 |-- engnat: integer (nullable = true)
 |-- age: integer (nullable = true)
 |-- hand: integer (nullable = true)
 |-- religion: integer (nullable = true)
 |-- orientation: integer (nullable = true)
 |-- race: integer (nullable = true)
 |-- voted: integer (nullable = true)
 |-- married: integer (nullable = true)
 |-- familysize: integer (nullable = true)
 |-- age_o: boolean (nullable = true)
 |-- familysize_o: boolean (nullable = true)
```

Figure 3.2.1.14 Perdictors_2_outliers Datatype – 1

We change the data type of the **age_o** and **familysize_o** columns from Boolean to string (shown in *Figure 3.2.1.15*).

```
from pyspark.sql.functions import col
Predictors_2_outliers = Predictors_2_outliers.select(*(col(c).cast("string").alias(c) for c in Predictors_2_outliers.columns))
```

```
Predictors_2_outliers.printSchema()
```

```
root
 |-- PersonID: string (nullable = true)
 |-- education: string (nullable = true)
 |-- urban: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- engnat: string (nullable = true)
 |-- age: string (nullable = true)
 |-- hand: string (nullable = true)
 |-- religion: string (nullable = true)
 |-- orientation: string (nullable = true)
 |-- race: string (nullable = true)
 |-- voted: string (nullable = true)
 |-- married: string (nullable = true)
 |-- familysize: string (nullable = true)
 |-- age_o: string (nullable = true)
 |-- familysize_o: string (nullable = true)
```

Figure 3.2.1.15 Perdictors_2_outliers Datatype – 2

Now we can eliminate the outliers and select the 12 predictors with PersonID column as illustrated in *Figure 3.2.1.16*. Outliers are excluded in the new data frame **Predictors_3**.

```
Predictors_3 = Predictors_2_outliers.filter((Predictors_2_outliers['age_o'] == 'false')
                                            & (Predictors_2_outliers['familysize_o'] == 'false')
       ).select('PersonID','education','urban','gender','engnat','age','hand',
       'religion','orientation','race','voted','married','familysize')
Predictors_3.show()
```

```
+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
|PersonID|education|urban|gender|engnat|age|hand|religion|orientation|race|voted|married|familysize|
+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
|       1|        2|    3|     2|     2| 16|   1|      12|          1|  10|    2|      1|         2|
|       2|        2|    3|     2|     1| 16|   2|       7|          0|  70|    2|      1|         4|
|       3|        2|    3|     2|     2| 17|   1|       4|          3|  60|    1|      1|         3|
|       4|        1|    3|     2|     1| 13|   2|       4|          5|  70|    2|      1|         5|
|       5|        3|    2|     2|     2| 19|   3|      10|          1|  10|    2|      1|         4|
|       6|        2|    3|     2|     2| 20|   1|       4|          1|  70|    2|      1|         4|
|       7|        2|    3|     2|     2| 17|   1|       7|          2|  60|    2|      1|         4|
|       8|        4|    2|     2|     2| 29|   1|       2|          2|  60|    1|      1|         2|
|       9|        2|    3|     1|     2| 16|   1|      12|          2|  70|    2|      1|         4|
|      10|        1|    1|     2|     2| 18|   1|       2|          2|  60|    2|      1|         3|
|      11|        1|    2|     1|     1| 15|   1|       6|          1|  60|    2|      1|         1|
|      12|        2|    1|     2|     1| 18|   1|       6|          1|  60|    2|      1|         2|
|      13|        3|    0|     2|     1| 20|   1|       1|          1|  60|    2|      1|         2|
|      16|        2|    2|     2|     1| 17|   1|       1|          2|  60|    2|      1|         1|
|      17|        2|    2|     2|     2| 19|   1|       7|          1|  60|    2|      1|         2|
|      18|        3|    3|     2|     1| 18|   1|      12|          1|  60|    2|      1|         5|
|      19|        2|    1|     1|     1| 19|   3|       2|          1|  60|    2|      1|         1|
|      20|        2|    3|     2|     1| 19|   1|       4|          2|  50|    2|      1|         2|
|      21|        1|    1|     2|     1| 15|   1|       4|          1|  60|    2|      1|         5|
|      22|        1|    2|     1|     1| 15|   1|       6|          1|  60|    2|      1|         1|
+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
only showing top 20 rows
```

Figure 3.2.1.16 Predictors_3

After removing the extreme values and outliers, there are 13,629 rows remains in the **Target_3** data frame, and 12,449 rows stay in the **Predictors_3** data frame.

```
print((Target_3 .count(), len(Target_3 .columns)))
print((Predictors_3.count(), len(Predictors_3.columns)))
```

```
(13629, 3)
(12449, 13)
```

Figure 3.2.1.17

3.2.2 Data Errors

As we discussed in *Section 2.4.2*, we assume there are no typographical errors made in the **Depression Scale** data set as the records are automatically generated. For the **Personal Information** dataset, we identified that the “major” column in the was recorded in an unstructured text format, and it is very difficult to be processed by Python. Therefore, we decide to filter out this column from the input data. We excluded the “major” field by creating a new data frame which only contains the selected 12 columns (education, urban, gender, engraft, age, hand, religion, orientation, race, voted, married, familysize). The detailed process is explained in *Section 3.1* above.

3.2.3 Measurement Errors

In *Section 2.4.3,* we found that there is no measurement error in the **Depression Scale** dataset. However, there are errors in the **Personal Information** dataset, where values in the categorical data falsely recorded to be 0. We assume they are null values and should be eliminated from the data.

```
from pyspark.sql.functions import col
Predictors_2_outliers = Predictors_2_outliers.select(*(col(c).cast("string").alias(c) for c in Predictors_2_outliers.columns))
Predictors_3 = Predictors_2_outliers.filter((Predictors_2_outliers['age_o'] == 'false')
                                            & (Predictors_2_outliers['familysize_o'] == 'false')
        ).select('PersonID','education','urban','gender','engnat','age','hand',
        'religion','orientation','race','voted','married','familysize')

Predictors_4 = Predictors_3.filter((Predictors_3['education'] != 0)
                                   & (Predictors_3['urban'] != 0)
                                   & (Predictors_3['gender'] != 0)
                                   & (Predictors_3['engnat'] != 0)
                                   & (Predictors_3['hand'] != 0)
                                   & (Predictors_3['religion'] != 0)
                                   & (Predictors_3['orientation'] != 0)
                                   & (Predictors_3['race'] != 0)
                                   & (Predictors_3['voted'] != 0)
                                   & (Predictors_3['married'] != 0)
                                  & (Predictors_3['familysize'] != 0)
        ).select('PersonID','education','urban','gender','engnat','age','hand',
        'religion','orientation','race','voted','married','familysize')

InputData = Target_3.join(Predictors_4, Target_3.PersonID == Predictors_4.PersonID)
```

```
print((Predictors_4.count(), len(Predictors_4.columns)))
```

```
(10883, 13)
```

Figure 3.2.3.1 Eliminating The Measurement Errors

There are 10,883 rows, and 13 columns remain after we removed the measurement errors.

3.3 Construct the data

To detect the outliers, we transform the data type of the data frames from strings to integers. To eliminate those outliers, we change the data back to strings again. We will need to transform our dataset to integer in *Section 3.5* format the data. We also constructed the outlier columns in *Section 3.2*, indicated by the suffix “_o”. We have already excluded them using the **select()** function. The outlier columns and the excluding process is illustrated in *Figure 3.2.1* below.

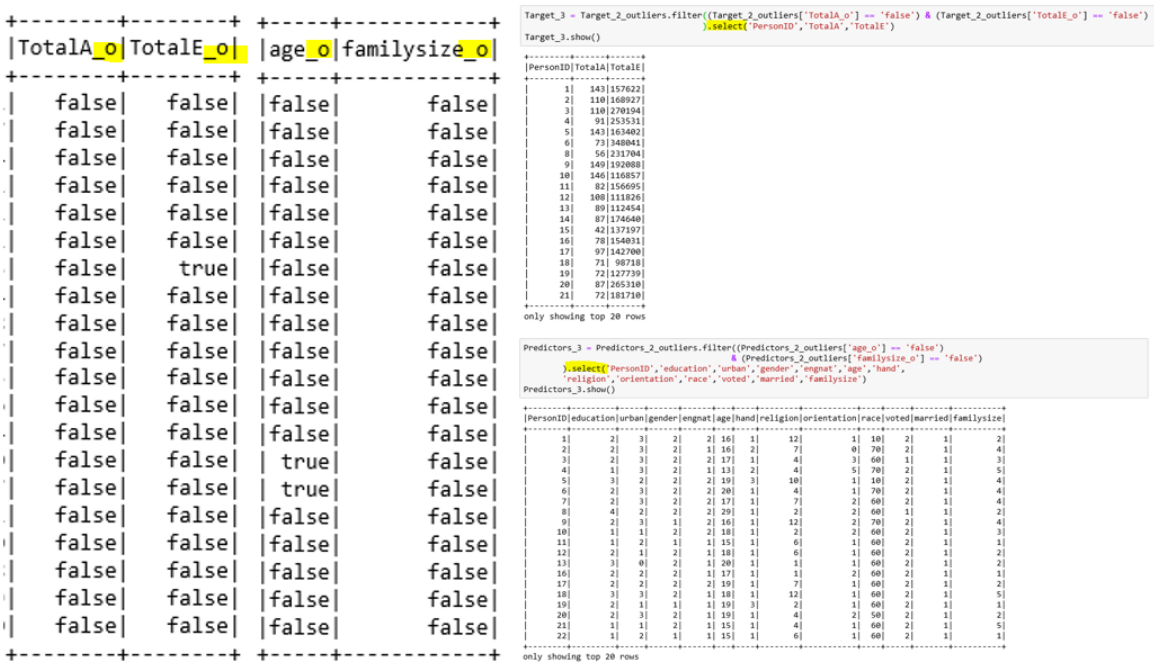


Figure 3.2.1 Construct and Exclude Outlier Detection Columns

We have included the **TotalE** column in the data frame **Target_3** for eliminating extreme values. Now we need to exclude it from the data frame because it is not the independent variable nor the dependent variable that contribute to our data mining objective.

```
Target_4 = Target_3.drop("TotalE")
```

```
Target_4.show()
```

```
+--------+------+
|PersonID|TotalA|
+--------+------+
|       1|   143|
|       2|   110|
|       3|   110|
|       4|    91|
|       5|   143|
|       6|    73|
|       8|    56|
|       9|   149|
|      10|   146|
|      11|    82|
|      12|   108|
|      13|    89|
|      14|    87|
|      15|    42|
|      16|    78|
|      17|    97|
|      18|    71|
|      19|    72|
|      20|    87|
|      21|    72|
+--------+------+
only showing top 20 rows
```

Figure 3.2.2 Show Target_4

As shown in Figure 3.2.2, we created the **Target_4** data frame that only contains the **PersonID** column for integration and our target variable **TotalA**.

To achieve the data mining objective, we also need the 12 predictors selected. The data frame **Predictors_4** is free from missing values, outliers and measurement errors; it is ready to be used. We can integrate this data frame with **Target_4** directly without constructing or eliminating any data.

```
Predictors_4.show()
```

```
+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
|PersonID|education|urban|gender|engnat|age|hand|religion|orientation|race|voted|married|familysize|
+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
|       1|        2|    3|     2|     2| 16|   1|      12|          1|  10|    2|      1|         2|
|       3|        2|    3|     2|     2| 17|   1|       4|          3|  60|    1|      1|         3|
|       4|        1|    3|     2|     1| 13|   2|       4|          5|  70|    2|      1|         5|
|       5|        3|    2|     2|     2| 19|   3|      10|          1|  10|    2|      1|         4|
|       6|        2|    3|     2|     2| 20|   1|       4|          1|  70|    2|      1|         4|
|       7|        2|    3|     2|     2| 17|   1|       7|          2|  60|    2|      1|         4|
|       8|        4|    2|     2|     2| 29|   1|       2|          2|  60|    1|      1|         2|
|       9|        2|    3|     1|     2| 16|   1|      12|          2|  70|    2|      1|         4|
|      10|        1|    1|     2|     2| 18|   1|       2|          2|  60|    2|      1|         3|
|      11|        1|    2|     1|     1| 15|   1|       6|          1|  60|    2|      1|         1|
|      12|        2|    1|     2|     1| 18|   1|       6|          1|  60|    2|      1|         2|
|      16|        2|    2|     2|     1| 17|   1|       1|          2|  60|    2|      1|         1|
|      17|        2|    2|     2|     2| 19|   1|       7|          1|  60|    2|      1|         2|
|      18|        3|    3|     2|     1| 18|   1|      12|          1|  60|    2|      1|         5|
|      19|        2|    1|     1|     1| 19|   3|       2|          1|  60|    2|      1|         1|
|      20|        2|    3|     2|     1| 19|   1|       4|          2|  50|    2|      1|         2|
|      21|        1|    1|     2|     1| 15|   1|       4|          1|  60|    2|      1|         5|
|      22|        1|    2|     1|     1| 15|   1|       6|          1|  60|    2|      1|         1|
|      23|        2|    3|     2|     1| 22|   1|       1|          2|  60|    2|      1|         1|
|      24|        2|    3|     2|     2| 19|   1|       4|          1|  70|    2|      1|         3|
+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
only showing top 20 rows
```

Figure 3.2.3 Show Predictors_4

3.4 Integrate various data sources

Now we want to merge the two datasets after we have cleaned them separately. This aggregation is crucial as after which we can examine the correlations between the participants' traits and their depression level, which is our business and data mining goal.

The data frame **Target**_4 contains the independent variable TotalA, and the **Predictors**_4 has the 12 predictors of depression level. We identify that these two data frames share the same key **PersonID**, so we are going to merge the data frames we created and clean base on this key.

```
InputData = Target_4.join(Predictors_4, Target_4.PersonID == Predictors_4.PersonID)
InputData.show()
```

```
+--------+------+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
|PersonID|TotalA|PersonID|education|urban|gender|engnat|age|hand|religion|orientation|race|voted|married|familysize|
+--------+------+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
|       1|   143|       1|        2|    3|     2|     2| 16|   1|      12|          1|  10|    2|      1|         2|
|       3|   110|       3|        2|    3|     2|     2| 17|   1|       4|          3|  60|    1|      1|         3|
|       4|    91|       4|        1|    3|     2|     1| 13|   2|       4|          5|  70|    2|      1|         5|
|       5|   143|       5|        3|    2|     2|     2| 19|   3|      10|          1|  10|    2|      1|         4|
|       6|    73|       6|        2|    3|     2|     2| 20|   1|       4|          1|  70|    2|      1|         4|
|       8|    56|       8|        4|    2|     2|     2| 29|   1|       2|          2|  60|    1|      1|         2|
|       9|   149|       9|        2|    3|     1|     2| 16|   1|      12|          2|  70|    2|      1|         4|
|      10|   146|      10|        1|    1|     2|     2| 18|   1|       2|          2|  60|    2|      1|         3|
|      11|    82|      11|        1|    2|     1|     1| 15|   1|       6|          1|  60|    2|      1|         1|
|      12|   108|      12|        2|    1|     2|     1| 18|   1|       6|          1|  60|    2|      1|         2|
|      16|    78|      16|        2|    2|     2|     1| 17|   1|       1|          2|  60|    2|      1|         1|
|      17|    97|      17|        2|    2|     2|     2| 19|   1|       7|          1|  60|    2|      1|         2|
|      18|    71|      18|        3|    3|     2|     1| 18|   1|      12|          1|  60|    2|      1|         5|
|      19|    72|      19|        2|    1|     1|     1| 19|   3|       2|          1|  60|    2|      1|         1|
|      20|    87|      20|        2|    3|     2|     1| 19|   1|       4|          2|  50|    2|      1|         2|
|      21|    72|      21|        1|    1|     2|     1| 15|   1|       4|          1|  60|    2|      1|         5|
|      22|    93|      22|        1|    2|     1|     1| 15|   1|       6|          1|  60|    2|      1|         1|
|      23|   136|      23|        2|    3|     2|     1| 22|   1|       1|          2|  60|    2|      1|         1|
|      24|    97|      24|        2|    3|     2|     2| 19|   1|       4|          1|  70|    2|      1|         3|
|      25|    89|      25|        2|    1|     1|     1| 20|   1|       7|          1|  60|    1|      1|         3|
+--------+------+--------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
only showing top 20 rows
```

Figure 3.4.1 Integrate the Data Frames

We used the join function to conduct an inner join of the dataset of **Target_4** and **Predictor_4** and obtain the new data frame **InputData**.

```
print((InputData.count(), len(InputData.columns)))
InputData.printSchema()
```

```
(10192, 15)
root
 |-- PersonID: string (nullable = true)
 |-- TotalA: string (nullable = true)
 |-- PersonID: string (nullable = true)
 |-- education: string (nullable = true)
 |-- urban: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- engnat: string (nullable = true)
 |-- age: string (nullable = true)
 |-- hand: string (nullable = true)
 |-- religion: string (nullable = true)
 |-- orientation: string (nullable = true)
 |-- race: string (nullable = true)
 |-- voted: string (nullable = true)
 |-- married: string (nullable = true)
 |-- familysize: string (nullable = true)
```

Figure 3.4.2 InputData Description - 1

As shown in *Figure 3.4.2*, we explore the **InputData** data frame contains 10,192 rows and 15 columns. The 15 columns include two **PersonID** columns, and all the data is in “string” data type. Therefore, we change the coding of the integration illustrated in *Figure 3.4.3* so that we will have one **PersonID** column.

```
InputData = Target_4.join(Predictors_4, ['PersonID'])
InputData.show()
```

```
+--------+------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
|PersonID|TotalA|education|urban|gender|engnat|age|hand|religion|orientation|race|voted|married|familysize|
+--------+------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
|       1|   143|        2|    3|     2|     2| 16|   1|      12|          1|  10|    2|      1|         2|
|       3|   110|        2|    3|     2|     2| 17|   1|       4|          3|  60|    1|      1|         3|
|       4|    91|        1|    3|     2|     1| 13|   2|       4|          5|  70|    2|      1|         5|
|       5|   143|        3|    2|     2|     2| 19|   3|      10|          1|  10|    2|      1|         4|
|       6|    73|        2|    3|     2|     2| 20|   1|       4|          1|  70|    2|      1|         4|
|       8|    56|        4|    2|     2|     2| 29|   1|       2|          2|  60|    1|      1|         2|
|       9|   149|        2|    3|     1|     2| 16|   1|      12|          2|  70|    2|      1|         4|
|      10|   146|        1|    1|     2|     2| 18|   1|       2|          2|  60|    2|      1|         3|
|      11|    82|        1|    2|     1|     1| 15|   1|       6|          1|  60|    2|      1|         1|
|      12|   108|        2|    1|     2|     1| 18|   1|       6|          1|  60|    2|      1|         2|
|      16|    78|        2|    2|     2|     1| 17|   1|       1|          2|  60|    2|      1|         1|
|      17|    97|        2|    2|     2|     2| 19|   1|       7|          1|  60|    2|      1|         2|
|      18|    71|        3|    3|     2|     1| 18|   1|      12|          1|  60|    2|      1|         5|
|      19|    72|        2|    1|     1|     1| 19|   3|       2|          1|  60|    2|      1|         1|
|      20|    87|        2|    3|     2|     1| 19|   1|       4|          2|  50|    2|      1|         2|
|      21|    72|        1|    1|     2|     1| 15|   1|       4|          1|  60|    2|      1|         5|
|      22|    93|        1|    2|     1|     1| 15|   1|       6|          1|  60|    2|      1|         1|
|      23|   136|        2|    3|     2|     1| 22|   1|       1|          2|  60|    2|      1|         1|
|      24|    97|        2|    3|     2|     2| 19|   1|       4|          1|  70|    2|      1|         3|
|      25|    89|        2|    1|     1|     1| 20|   1|       7|          1|  60|    1|      1|         3|
+--------+------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
only showing top 20 rows
```

Figure 3.4.3 Integration Iteration

```
print((InputData.count(), len(InputData.columns)))
InputData.printSchema()
```

```
(9887, 14)
root
 |-- PersonID: string (nullable = true)
 |-- TotalA: string (nullable = true)
 |-- education: string (nullable = true)
 |-- urban: string (nullable = true)
 |-- gender: string (nullable = true)
 |-- engnat: string (nullable = true)
 |-- age: string (nullable = true)
 |-- hand: string (nullable = true)
 |-- religion: string (nullable = true)
 |-- orientation: string (nullable = true)
 |-- race: string (nullable = true)
 |-- voted: string (nullable = true)
 |-- married: string (nullable = true)
 |-- familysize: string (nullable = true)
```

Figure 3.4.4 InputData Description - 2

There is no duplicate column in the data frame and contains our target and predictors now. Nevertheless, we still need to change the datatypes from string to integer so that we can process the data with data mining algorisms. There are 9,887 rows, and 14 columns remain in the dataset.s

3.5 Format the data as required

The desirable data types for our input data is the integer type. However, as we discussed in *Section 3.4*, our input data is in the string datatype. We can achieve that with the **col** and **cast()** functions easily as shown by *Figure 3.5.1* below.

```
InputData = InputData.select(*(col(c).cast("integer").alias(c) for c in InputData.columns))
InputData.printSchema()
```

```
root
 |-- PersonID: integer (nullable = true)
 |-- TotalA: integer (nullable = true)
 |-- education: integer (nullable = true)
 |-- urban: integer (nullable = true)
 |-- gender: integer (nullable = true)
 |-- engnat: integer (nullable = true)
 |-- age: integer (nullable = true)
 |-- hand: integer (nullable = true)
 |-- religion: integer (nullable = true)
 |-- orientation: integer (nullable = true)
 |-- race: integer (nullable = true)
 |-- voted: integer (nullable = true)
 |-- married: integer (nullable = true)
 |-- familysize: integer (nullable = true)
```

Figure 3.5.1 Format InputData

Now all columns of **InputData** is in the correct format of integer type.

Data Transformation

4.1 Reduce the data

The input data has already significantly reduced in phase 3 - **Data Preparation**. The initial data contains **15,000 records**, **129 fields** from Depression Scale and **47 columns** from Personal Information dataset, and we have reduced it to a set with **9,887 rows** and **14 columns**. We believe we have sufficient data to build our model after data cleaning. In the beginning, we have two datasets with irrelevant columns, outliers, missing values and errors. Now it has become one clean, integrated dataset. In this way, we have also increased the intelligence density, which means we can gain more insights from less amount of data in a shorter amount of time.

4.2 Project the data

The Target

Our data objective is to explore the correlations between characters of depressed people and their depression level. To achieve this goal, we first check the distribution of our target **TotalA**.

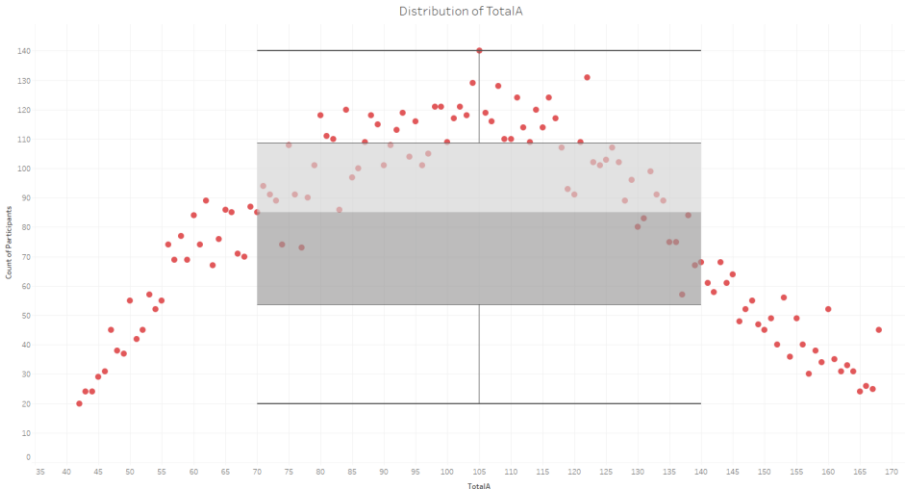


Figure 4.2.1 Distribution of TotalA

In the visualisation of **TotalA** above (*Figure 4.2.1*), we can see a normal distribution of the data points. It is ideal that the independent variable that the targeted variable is normally distributed for building our model. This is because many models require the targeted variable to satisfy a normality assumption. Therefore, we do not need to further project **TotalA**. Otherwise, we need to import the **Normalizer** to normalise the **TotalA** column. We can normalise the data using L1 norm or L-infinity norm. L1 norm is the sum of the magnitudes of the vectors in space, and the L-infinity norm gives the largest magnitude among each element of a vector.

12 Predictors

As we explained in the data understanding phase, there are 10 categorical columns (highlighted in *Table 4.2.2*) in the 12 predictors we selected.

Characteristics		
education	"How much education have you completed?"	Categorical

	1=Less than high school, 2=High school, 3=University degree, 4=Graduate degree	
urban	What type of area did you live when you were a child? 1=Rural (countryside), 2=Suburban, 3=Urban (town, city)	Categorical
gender	What is your gender? 1=Male, 2=Female, 3=Other	Categorical
engnat	Is English your native language? 1=Yes, 2=No	Categorical
age	How many years old are you?	Continous
hand	What hand do you use to write with? 1=Right, 2=Left, 3=Both	Categorical
religion	What is your religion?, 1=Agnostic, 2=Atheist, 3=Buddhist, 4=Christian (Catholic), 5=Christian (Mormon), 6=Christian (Protestant), 7=Christian (Other), 8=Hindu, 9=Jewish, 10=Muslim, 11=Sikh, 12=Other	Categorical
orientation	What is your sexual orientation? 1=Heterosexual, 2=Bisexual, 3=Homosexual, 4=Asexual, 5=Other	Categorical
race	What is your race? 10=Asian, 20=Arab, 30=Black, 40=Indigenous Australian, 50=Native American, 60=White, 70=Other	Categorical
voted	Have you voted in a national election in the past year? 1=Yes, 2=No	Categorical
married	What is your marital status? 1=Never married, 2=Currently married, 3=Previously married	Categorical
familysize	Including you, how many children did your mother have?	Continous

Table 4.2.1 Distribution of TotalA

As shown in the metadata in *Table 4.2.1* and *Figure 4.2.2*, the categorical columns are already recorded with index.

```
+--------+------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
|PersonID|TotalA|education|urban|gender|engnat|age|hand|religion|orientation|race|voted|married|familysize|
+--------+------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
|       1|   143|        2|    3|     2|     2| 16|   1|      12|          1|  10|    2|      1|         2|
|       3|   110|        2|    3|     2|     2| 17|   1|       4|          3|  60|    1|      1|         3|
|       4|    91|        1|    3|     2|     1| 13|   2|       4|          5|  70|    2|      1|         5|
|       5|   143|        3|    2|     2|     2| 19|   3|      10|          1|  10|    2|      1|         4|
|       6|    73|        2|    3|     2|     2| 20|   1|       4|          1|  70|    2|      1|         4|
|       8|    56|        4|    2|     2|     2| 29|   1|       2|          2|  60|    1|      1|         2|
|       9|   149|        2|    3|     1|     2| 16|   1|      12|          2|  70|    2|      1|         4|
|      10|   146|        1|    1|     2|     2| 18|   1|       2|          2|  60|    2|      1|         3|
|      11|    82|        1|    2|     1|     1| 15|   1|       6|          1|  60|    2|      1|         1|
|      12|   108|        2|    1|     2|     1| 18|   1|       6|          1|  60|    2|      1|         2|
|      16|    78|        2|    2|     2|     1| 17|   1|       1|          2|  60|    2|      1|         1|
|      17|    97|        2|    2|     2|     2| 19|   1|       7|          1|  60|    2|      1|         2|
|      18|    71|        3|    3|     2|     1| 18|   1|      12|          1|  60|    2|      1|         5|
|      19|    72|        2|    1|     1|     1| 19|   3|       2|          1|  60|    2|      1|         1|
|      20|    87|        2|    3|     2|     1| 19|   1|       4|          2|  50|    2|      1|         2|
|      21|    72|        1|    1|     2|     1| 15|   1|       4|          1|  60|    2|      1|         5|
|      22|    93|        1|    2|     1|     1| 15|   1|       6|          1|  60|    2|      1|         1|
|      23|   136|        2|    3|     2|     1| 22|   1|       1|          2|  60|    2|      1|         1|
|      24|    97|        2|    3|     2|     2| 19|   1|       4|          1|  70|    2|      1|         3|
|      25|    89|        2|    1|     1|     1| 20|   1|       7|          1|  60|    1|      1|         3|
+--------+------+---------+-----+------+------+---+----+--------+-----------+----+-----+-------+----------+
only showing top 20 rows
```

Figure 4.2.2 Categorical Columns of InputData

Therefore, we do not need to assign indexes to these categorical columns. Otherwise, we will use the **StringIndexer** to encode the category name and transform the data from string to numeric type.

Since we do not understand the pattern of the data yet, we decide to conduct an undirected knowledge discovery as we explained in the data mining goal. We are going to use the method of Principal component analysis (PCA) to facilitate us for the discovery. PCA is a dimensionality-reduction method that reduces the dimensionality of large data sets. It transforms the set of variables in the data into a smaller one. Still, it contains most of the information in the large set. We will lose some accuracy at this point (Jaadi, 2019). However, we only need to have an overall understanding of our dataset, so it is still worthy of visualising the PCA value of the 12 predictors against **TotalA**. Therefore, we will project our predictors to a **PCAfeature** column.

In addition, we need to transform the 12 predictors into a vector with 12 elements. Most of the data mining algorithms in Pyspark can only process vectors.

Transformation

We first import the **VectorAssember**. For the input columns, we selected the 12 predictors and combined them into a single vector column **features** (shown in *Figure 4.2.3*).

```
from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import VectorAssembler
assembler = VectorAssembler(
    inputCols=['education','urban','gender','engnat','age','hand',
        'religion','orientation','race','voted','married','familysize'],
    outputCol="features")
output = assembler.transform(InputData)
```

Figure 4.2.3 Vector Transform

Next, we created a new data frame named **InputData_2**, as shown in *Figure 4.2.4* below.

```
InputData_2 = output.select('PersonID','TotalA','features')
InputData_2.show()
print((InputData_2.count(), len(InputData_2.columns)))
InputData_2.printSchema()
```

```
+--------+------+--------------------+
|PersonID|TotalA|            features|
+--------+------+--------------------+
|       1|   143|[2.0,3.0,2.0,2.0,...|
|       3|   110|[2.0,3.0,2.0,2.0,...|
|       4|    91|[1.0,3.0,2.0,1.0,...|
|       5|   143|[3.0,2.0,2.0,2.0,...|
|       6|    73|[2.0,3.0,2.0,2.0,...|
|       8|    56|[4.0,2.0,2.0,2.0,...|
|       9|   149|[2.0,3.0,1.0,2.0,...|
|      10|   146|[1.0,1.0,2.0,2.0,...|
|      11|    82|[1.0,2.0,1.0,1.0,...|
|      12|   108|[2.0,1.0,2.0,1.0,...|
|      16|    78|[2.0,2.0,2.0,1.0,...|
|      17|    97|[2.0,2.0,2.0,2.0,...|
|      18|    71|[3.0,3.0,2.0,1.0,...|
|      19|    72|[2.0,1.0,1.0,1.0,...|
|      20|    87|[2.0,3.0,2.0,1.0,...|
|      21|    72|[1.0,1.0,2.0,1.0,...|
|      22|    93|[1.0,2.0,1.0,1.0,...|
|      23|   136|[2.0,3.0,2.0,1.0,...|
|      24|    97|[2.0,3.0,2.0,2.0,...|
|      25|    89|[2.0,1.0,1.0,1.0,...|
+--------+------+--------------------+
only showing top 20 rows

(10192, 3)
root
 |-- PersonID: integer (nullable = true)
 |-- TotalA: integer (nullable = true)
 |-- features: vector (nullable = true)
```

Figure 4.2.4 InputData_2

The **InputData_2** data frame contains 9,887 rows and 3 columns – **PersonID**, **TotalA** and **features**. **PersonID** is the primary key of the data frame and is recorded as integers. The **TotalA** is our targeted column, which is also recorded as integers. It indicates the depression level of the participants. The **features** column contains vectors which consist of the 12 predictors, which are also characteristics of the participants.

Next, we created a PCA column for each participant.

```
from pyspark.ml.feature import PCA
from pyspark.ml.linalg import Vectors
pca = PCA(k=1, inputCol='features', outputCol='PCA_Value')
model = pca.fit(InputData_2)
PCA = model.transform(InputData_2).select('PersonID','TotalA','PCA_Value')
```

Figure 4.2.5 PCA Settings

We want to combine the 12 elements of in each vector into one dimension. Hence, we set the k value for PCA function to be 1, and input column as the **features** column from the **InputData_2** data frame. The new data frame **PCA,** which contains the PCA values of each vector, is shown in *Figure 4.2.6* below.

```
PCA.show()
```

```
+--------+------+--------------------+
|PersonID|TotalA|           PCA_Value|
+--------+------+--------------------+
|       1|   143| [8.803203608640292]|
|       3|   110| [59.20135603563315]|
|       4|    91| [69.22401397741447]|
|       5|   143| [8.862151099072122]|
|       6|    73| [69.10647546315172]|
|       8|    56| [59.15380881650373]|
|       9|   149| [68.60866090721585]|
|      10|   146|[59.335818838374315]|
|      11|    82| [59.13716154825016]|
|      12|   108| [59.06724983573774]|
|      16|    78|[59.449011119901606]|
|      17|    97|[58.970921040356394]|
|      18|    71|  [58.5868992526471]|
|      19|    72| [59.35133049973317]|
|      20|    87| [49.21768841258286]|
|      21|    72| [59.21843763156579]|
|      22|    93| [59.13716154825016]|
|      23|   136| [59.36931646357907]|
|      24|    97| [69.13698183732366]|
|      25|    89| [58.95487646262147]|
+--------+------+--------------------+
only showing top 20 rows
```

Figure 4.2.5 PCA Data frame

Even though the **PCA_Value** is in one dimension, it is still a vector. We need to change the format of this column to the float datatype.

```
import pyspark.sql.functions as f
from pyspark.sql.types import FloatType
changetype=f.udf(lambda v:float(v[0]),FloatType())
PCA=PCA.withColumn("PCA_Value", changetype("PCA_Value"))
PCA.show()
PCA.printSchema()
```

```
+--------+------+---------+
|PersonID|TotalA|PCA_Value|
+--------+------+---------+
|       1|   143| 8.803204|
|       3|   110|59.201355|
|       4|    91|69.224014|
|       5|   143| 8.862151|
|       6|    73|69.106476|
|       8|    56| 59.15381|
|       9|   149| 68.60866|
|      10|   146| 59.33582|
|      11|    82| 59.13716|
|      12|   108| 59.06725|
|      16|    78|59.449013|
|      17|    97| 58.97092|
|      18|    71|  58.5869|
|      19|    72| 59.35133|
|      20|    87| 49.21769|
|      21|    72|59.218437|
|      22|    93| 59.13716|
|      23|   136|59.369316|
|      24|    97| 69.13698|
|      25|    89|58.954876|
+--------+------+---------+
only showing top 20 rows

root
 |-- PersonID: integer (nullable = true)
 |-- TotalA: integer (nullable = true)
 |-- PCA_Value: float (nullable = true)
```

Figure 4.2.6 PCA Data frame Change Datatype

We also export the **PCA** data frame into a csv file so that we can import the file into Tableau. We will visualize the relationship between the PCA values of the 12 predictors and TotalA in the data mining stage to conduct an undirected knowledge discovery.

```
PCA.toPandas().to_csv('PCA.csv')
```

```
1  ,PersonID,TotalA,PCA_Value
2  0,1,143,8.803203582763672
3  1,3,110,59.20135498046875
4  2,4,91,69.22401428222656
5  3,5,143,8.862151145935059
6  4,6,73,69.10647583007812
7  5,8,56,59.15380859375
8  6,9,149,68.60865783691406
9  7,10,146,59.335819244384766
10 8,11,82,59.13716125488281
```

Figure 4.2.7 Export PCA Data Frame

Data-mining Methods Selection

5.1 Match and discuss the objectives of data mining to data mining methods

The two primary goals of data mining in most practice are prediction and description (Fayyad et al., 1996). To achieve our data mining objectives, we need to describe our data, but furthermore, to generate a checklist of important predictors which are highly correlated to depression. This list will also support our business objective of increase the awareness of depression issues so that decrease severe depression cases. However, we do not know which characters have a strong correlation with depression level, so we must conduct an undirected knowledge discovery to first. The results from the discovery will help us establish an expectation of the final output.

Berry and Linoff (1997) introduce six basic data mining methods: Classification, Estimation, Affinity Grouping, Clustering, Description, Prediction.

Classification is a method that assigns objects into groups according to their feature. It is not applicable in this case because we are not assigning people into classes but examine the relationship between their characteristics and depression level.

We do not think **Affinity Grouping** or **Market Basket Analysis** is appropriate for our study neither. It aims to determine which items go together rather than a correlation towards one specific variable.

The data mining method of **Clustering** is often used to conduct an undirected knowledge discovery of the data. We reckon this method will help to discover some patterns and correlations in our data.

Estimation aims to estimate some unknown variables base on the existing data we have. It is often used in the area of accounting and finance, which allows the manager to predict income, budget costs. The unknown variables are usually continuous, but the prediction we want to make is the depression level of individual persons. Therefore, we reckon. Estimation is not suitable for our modelling.

The **Description** method is to describe the reality; it can be a statistic summary or rules of the existing data. Some model can be both descriptive and predictive, and the boundary between Description and Prediction is blurry (Fayyad et al., 1996). However, the overall data mining behind them is different. We consider this method useful in terms of understanding the existing data. Nevertheless, we want to build a model which eventually predict a person's likelihood to catch depression. Hence, we must go further than just describing the data.

We believe **Prediction** would be the methods that produce the ultimate results for our study. It will allow us to discover the significant factors that correlate with depression level. Although **Prediction** contains aspects of classification and estimation, it can grant us insights about future behaviours and future values of the target variable. In our case, the depression level of a person with specific traits.

5.2 Select the appropriate data-mining methods based on discussion

Base on the discussion in *Section 5.1* above, we decide to use **Clustering** to explore any patterns in data firstly. Then we will use **Description** to describe the behaviour of our existing data. Finally, we will use **Prediction** to generate our final outputs. We believe this progression

will not only increase the accuracy of our model but also increase the intelligence density of our output. It will transform the data into knowledge.

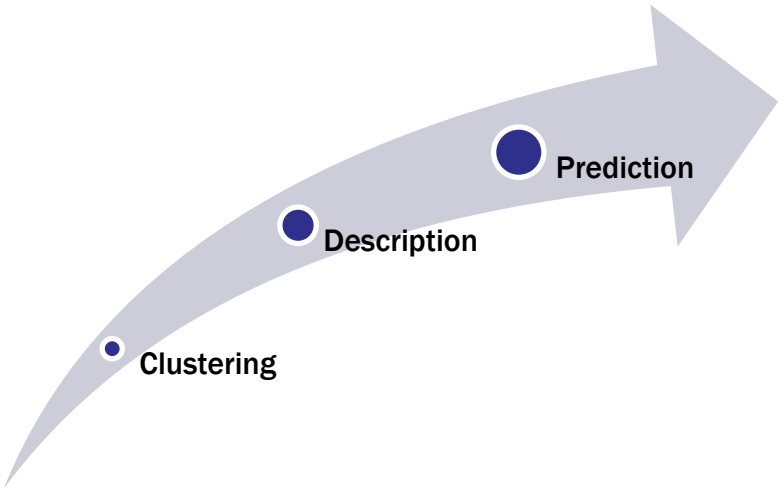


Figure 5.2 Data Mining Progression

Clustering

We select Clustering to conduct an undirected knowledge discovery. This method will segment the population into several relatively homogeneous subgroups or cluster. The factors that distinguish each group will be the expected predictors which we will expect to see in the output of Estimation and Prediction. We will further discuss our selection of data-mining algorithms in the next section.

Description

We believe it is essential to describe the existing reality first before we conduct a prediction about the future. We select the Description method to investigate the patterns that occur in the historical data. This method will help us achieve the data mining goal of describing and explaining how the participants' characteristics correlate to the participants' depression level.

Prediction

Although there may be some overlaps between Description and Prediction, our data mining objective with Prediction method is at a higher level than Description. We are forming a prediction of future behaviour unknown values base on our current finding, which is further than merely describing reality. The purpose of this study is not only understanding people who have depression issues. We also aim to alert those with a high potential to catch this mental illness so that we will prevent from developing depression.

Data-mining Algorithms Selection

6.1 Conduct exploratory analysis and discuss

We select **Clustering, Description,** and **Prediction** us our primary data mining methods, as discussed in phrase 5 Data-mining methods selection. Now we will explore and discuss what data-mining algorithms are available to facilitate our data-mining algorithms.

Clustering and available algorithms

We select Clustering to conduct an undirected knowledge discovery, which segments the population into several finite subgroups or clusters. With Pyspark, we choose the method of K-mean clustering as it is a quick, easy method to help us understand our data. However, this method does not decide the number of clusters automatically, so we must conduct a few iterations to find the best number for clustering the data.

Along with Clustering, we are also going to use Principal Component Analysis (PCA) to support our analysis. PCA help us to create new data which can represent multiple columns of our dataset. Since the PCA columns are not the original columns of our datasets, we will compromise the accuracy in this discovery stage. However, the visualisation of **PCA** values against **TotalA** can still give us an idea of our data and support the achievement of our goals.

Description and available algorithms

Our goal to use the Description method is to accomplish our data mining objective of describing and explaining the relationship between people's characters and their level of depression. We consider implementing a statistics approach to analyse and describe our data. A statistics approach can be summarising the means and standard deviations for all fields (Fayyad et al., 1996), calculating the percentages and discovering rules in the data.

With Python, we plan to conduct a Regression Analysis to understand and describe our data. Regression Analysis is a powerful statistical approach that examines the relationship between one or multiple variables. We believe we will achieve our data mining goal to explore the correlations between the depression level of participants and their characteristics.

Prediction and available algorithms

Prediction is critical in terms of reducing latency. If we can take action to stop the development in its early stage and prevent depression from happening acting proactively. An example can be Walmart predict an increase in sales of strawberry Pop-Tarts when hurricanes happen (Marr, 2016) so that they can proactively stock up the tarts in advance. To construct a predictive model, we can use the algorism, such as Decision Trees, Neural Networks, Regression Analysis. Those models allow us to predict future events, in this study, the occurrence of depression, with factors correlated to the event.

Decision Trees is often used to answer a True and False question, such as whether to grant a loan to a person, or whether to sign a business contract. We do not think it is suitable in our case because our objective is to identify a list of factors that are correlated to depression issues.

Neural Networks algorithms are powerful as it processed information like a human. It recognises patterns in the data using machine learning (Singh Gill, 2014). We identify that the Neural Networks algorithms are a black box model which we do not understand the entire reasoning behind its output. Hence, we will also apply a white box algorithm to complement the black-box model.

Regression Analysis is a white box model that is both powerful and easier to understand. It will express the correlation with a formula, as well as present the correlations to us with multiple statistics data.

6.2 Select data-mining algorithms based on discussion

K-mean

Our data mining goal of using Clustering is to establish an expectation of what patterns exist among the participants. The Clustering method will be the first step of our data mining, so we reckon a concise but insightful overview of the data is enough at this stage. Einstein states, “Everything should be made as simple as possible, but not simpler.” We believe K-means is a simple but powerful approach to help us to have an overall idea of our data and build a foundation for our further exploration. Its output would meet our data mining goals to generate a preview of the data correlations, but simple at the same time (Fayyad et al., 1996).

Regression Analysis

We spot that the Regression Analysis is appropriate algorithms to support both Description method and Prediction method. These two algorithms will assist us in achieving two objectives:

- Explaining the correlations and patterns in current participants’ behaviours
- Discovering the factors which predict the depression level.

Therefore, we believe we should implement these two algorithms as they will generate output with the highest intelligence density to meet our objectives.

6.3 Build/Select appropriate model(s) and choose relevant parameter(s)

We selected two algorithms for three data mining method in the discussion in *Section 6.2*. The corresponding modelling is K-mean clustering and Linear Regression Model.

Sufficient Records

In *Section 4.1 Reduce the data*, we explain that we have reduced the two datasets into on aggregated set with 10,192 rows and 14 columns. In *Section 4.2 Project the data*, we transform our input data to 10,192 rows and 2 columns – one target column, one vector column that contains the 12 predictors. We set the benchmark for successful modelling to be at least 5,000 of good records and ten predictor fields. The processed dataset fulfils our requirements.

Data quality and format

The quality of our data is good. We have eliminated all the missing values and extremes from the data. Then we remove the rows with measurement errors from the 12 predictors column. Moreover, we also reduce the data to contain only the primary key **PersonID**, our target and 12 predictors. In order to implement the data mining algorithms, we further transform the data into a data frame with only 2 columns – one integer column (**TotalA**) and one vector column (**12 predictors**).

6.3.1 K-mean

For the K-mean algorithm, we must decide how many clusters we want our data to have. We will first try to cluster our data into 3 and 6 groups in the test design to examine we need to make further adjustments.

```
from pyspark.ml.clustering import KMeans
kmeans = KMeans().setK(3).setSeed(1)
model = kmeans.fit(InputData_2)
wssse = model.computeCost(InputData_2)
print("Within Set Sum of Squared Errors = " + str(wssse))
```

```
Within Set Sum of Squared Errors = 414780.07720809925
```

Figure 6.3.1.1 K-mean set up – 3 clusters

```
from pyspark.ml.clustering import KMeans
kmeans = KMeans().setK(6).setSeed(1)
model = kmeans.fit(InputData_2)
wssse = model.computeCost(InputData_2)
print("Within Set Sum of Squared Errors = " + str(wssse))
```

```
Within Set Sum of Squared Errors = 250694.93366073133
```

Figure 6.3.1.2 K-mean set up – 6 clusters

By comparing the sum of squared errors of the two clusterings, we found the model with 6 clusters is more accurate than the model with 3 clusters.

```
from pyspark.ml.clustering import KMeans
kmeans = KMeans().setK(7).setSeed(1)
model = kmeans.fit(InputData_2)
wssse = model.computeCost(InputData_2)
print("Within Set Sum of Squared Errors = " + str(wssse))
```

```
Within Set Sum of Squared Errors = 266325.9829985107
```

Figure 6.3.1.3 K-mean set up – 7 clusters

However, the sum of squared errors increased when we change the model to 7 clusters. Therefore, we will fit our data into 6 clusters in the data mining phase.

6.3.2 Regression Model

Now we will set up the regression model. There are three parts in a linear regression model – independent variables, dependent variables and predictions. We will also conduct participation by randomly split our input data into a training dataset and a testing dataset. In this way, we can compare the predictions of our model from the training data with the testing data and evaluate our model. The data for independent variables is the **feactures** column, which contains the 12 characteristics of the participants, and each record is a vector. We will fit the **TotalA** column for the values of the dependent variable.

We will obtain a regression model that explain the relationship of the 12 predictors with our target variable by 12 coefficients and an intercept. We can also compare the predictions with the actual values in the test data so that we can evaluate the accuracy of our model.

In the next section, we will also create test designs to find out the most appropriate ratio of splitting our data.

Data Mining

7.1 Create and justify test designs

The purpose of creating a test design is to adjust the criteria for successful modelling. We have not compared the criteria we set before with the actual models yet. We will create the first version of our models and adjust to improve the quality of our model comparing to our success criteria and data mining objective.

K-mean

As we discussed in *Section 6.3*, we decide to set the number of clusters as 6 in the K-mean model.

```
from pyspark.ml.clustering import KMeans
kmeans = KMeans().setK(6).setSeed(1)
model = kmeans.fit(InputData_2)
wssse = model.computeCost(InputData_2)
print("Within Set Sum of Squared Errors = " + str(wssse))

centers = model.clusterCenters()
print("Cluster Centers: ")
for center in centers:
    print(center)
```

```
Within Set Sum of Squared Errors = 230630.57603615624
Cluster Centers:
[ 2.65092074  2.30050707  1.86068855  1.82839605 20.6581265   1.13344009
  9.88630905  1.80731252 10.          1.92500667  1.0376301   3.7576728 ]
[ 2.69961089  2.11439689  1.68015564  1.2848249  24.65836576  1.16809339
  4.2381323   1.58287938 59.85992218  1.49338521  1.17743191  2.44124514]
[ 2.29899497  2.37269682  1.80234506  1.56951424 19.76716918  1.1440536
  6.53852596  2.00586265 70.          1.85175879  1.04522613  3.20435511]
[ 2.14098361  2.43278689  1.77704918  1.35081967 19.50491803  1.15409836
  6.81311475  1.83606557 27.1147541   1.79016393  1.05901639  2.9442623 ]
[ 2.21313364  2.46889401  1.77880184  1.63248848 19.15898618  1.10253456
  3.27880184  1.62788018 10.          1.80299539  1.01497696  2.5437788 ]
[ 1.63223473  2.17282958  1.77130225  1.21623794 16.76889068  1.18770096
  4.18207395  1.9176045  59.71061093  1.86575563  1.0068328   2.5494373 ]
```

Figure 7.1.1 K-mean Cluster Centers

The model is established, and we can obtain the details of the 6 clusters as desired. Therefore, we do not need to make further adjustments.

Regression Analysis

Before building the linear regression model, we need to examine both 20/80 and 30/70 partition and choose the one gives a higher accuracy of our model. We first tried out the 20/80 partitions by creating a training dataset that contains 80% randomly selected data from our dataset, and a testing dataset that contains the other 20% of our data.

We first drop the primary key **PersonID** from **InputData_2** and only reserve the **TotalA** and the **features** column.

```
InputData_3 = InputData_2.drop("PersonID")
InputData_3.show()
```

```
+------+--------------------+
|TotalA|            features|
+------+--------------------+
|   143|[2.0,3.0,2.0,2.0,...|
|   110|[2.0,3.0,2.0,2.0,...|
|    91|[1.0,3.0,2.0,1.0,...|
|   143|[3.0,2.0,2.0,2.0,...|
|    73|[2.0,3.0,2.0,2.0,...|
|    56|[4.0,2.0,2.0,2.0,...|
|   149|[2.0,3.0,1.0,2.0,...|
|   146|[1.0,1.0,2.0,2.0,...|
|    82|[1.0,2.0,1.0,1.0,...|
|   108|[2.0,1.0,2.0,1.0,...|
|    78|[2.0,2.0,2.0,1.0,...|
|    97|[2.0,2.0,2.0,2.0,...|
|    71|[3.0,3.0,2.0,1.0,...|
|    72|[2.0,1.0,1.0,1.0,...|
|    87|[2.0,3.0,2.0,1.0,...|
|    72|[1.0,1.0,2.0,1.0,...|
|    93|[1.0,2.0,1.0,1.0,...|
|   136|[2.0,3.0,2.0,1.0,...|
|    97|[2.0,3.0,2.0,2.0,...|
|    89|[2.0,1.0,1.0,1.0,...|
+------+--------------------+
only showing top 20 rows
```

Figure 7.1.1 K-mean Cluster Centers

Then we split the data by 30/70 and 20/80.

```
train_data_1,test_data_1 = InputData_3.randomSplit([0.7,0.3])
```

```
train_data_1.describe().show()
test_data_1.describe().show()
```

```
+-------+------------------+
|summary|            TotalA|
+-------+------------------+
|  count|              6859|
|   mean|102.60212859017349|
| stddev|30.045297795904705|
|    min|                42|
|    max|               168|
+-------+------------------+

+-------+------------------+
|summary|            TotalA|
+-------+------------------+
|  count|              3028|
|   mean|102.68031704095112|
| stddev|29.956716638941014|
|    min|                42|
|    max|               168|
+-------+------------------+
```

```
train_data_2,test_data_2 = InputData_3.randomSplit([0.8,0.2])
train_data_1.describe().show()
test_data_1.describe().show()
```

```
+-------+------------------+
|summary|            TotalA|
+-------+------------------+
|  count|              6859|
|   mean|102.60212859017349|
| stddev|30.045297795904705|
|    min|                42|
|    max|               168|
+-------+------------------+

+-------+------------------+
|summary|            TotalA|
+-------+------------------+
|  count|              3028|
|   mean|102.68031704095112|
| stddev|29.956716638941014|
|    min|                42|
|    max|               168|
+-------+------------------+
```

Figure 7.1.2 Partition

We will then fit the 2 sets of training data into the regression model, and compare the root mean squared error of the two partitions.

```
from pyspark.ml.regression import LinearRegression
lr = LinearRegression(labelCol='TotalA')
```

```
lrModel_1 = lr.fit(train_data_1)
test_results = lrModel_1.evaluate(test_data_1)
```

```
print("RSME: {}".format(test_results.rootMeanSquaredError))
```

```
RSME: 28.71393480395135
```

Figure 7.1.3 Root Mean Squared Error of train_data_1

```
lrModel_2 = lr.fit(train_data_2)
test_results = lrModel_1.evaluate(test_data_2)
print("RSME: {}".format(test_results.rootMeanSquaredError))
```

```
RSME: 29.48734757984158
```

Figure 7.1.4 Root Mean Squared Error of train_data_2

We find out that the root mean squared error of the 30/70 partition is 28.71, which is less than that of the 20/80 partitions (29.49). Hence, we will adopt the 30/70 partition for splitting our input data which will be fit into the linear regression model.

7.2 Conduct data mining

Learning from our test design, we conclude that we should cluster our data into 6 clusters and implement the 30/70 splits for building a linear regression model.

7.2.1 Undirected knowledge discovery

We firstly conducted an undirected knowledge discovery using K-mean model.

```
from pyspark.ml.clustering import KMeans
kmeans = KMeans().setK(6).setSeed(1)
model = kmeans.fit(InputData_2)
wssse = model.computeCost(InputData_2)
print("Within Set Sum of Squared Errors = " + str(wssse))

centers = model.clusterCenters()
print("Cluster Centers: ")
for center in centers:
    print(center)
```

```
Within Set Sum of Squared Errors = 230630.57603615624
Cluster Centers:
[ 2.65092074  2.30050707  1.86068855  1.82839605 20.6581265   1.13344009
  9.88630905  1.80731252 10.          1.92500667  1.0376301   3.7576728 ]
[ 2.69961089  2.11439689  1.68015564  1.2848249  24.65836576  1.16809339
  4.2381323   1.58287938 59.85992218  1.49338521  1.17743191  2.44124514]
[ 2.29899497  2.37269682  1.80234506  1.56951424 19.76716918  1.1440536
  6.53852596  2.00586265 70.          1.85175879  1.04522613  3.20435511]
[ 2.14098361  2.43278689  1.77704918  1.35081967 19.50491803  1.15409836
  6.81311475  1.83606557 27.1147541   1.79016393  1.05901639  2.9442623 ]
[ 2.21313364  2.46889401  1.77880184  1.63248848 19.15898618  1.10253456
  3.27880184  1.62788018 10.          1.80299539  1.01497696  2.5437788 ]
[ 1.63223473  2.17282958  1.77130225  1.21623794 16.76889068  1.18770096
  4.18207395  1.9176045  59.71061093  1.86575563  1.0068328   2.5494373 ]
```

Figure 7.2.1.1 K-mean Result

From the K-mean clustering, we obtain 6 clusters and the cluster centres. The cluster centres are the arithmetic means of all the points belonging to each cluster. Since there is a difference between participants, we expect there is also a difference in the value of **TotalA** corresponding with different participants. We will further discover the patterns of the data by visualising the **PCA** values of the predictors against **TotalA**.

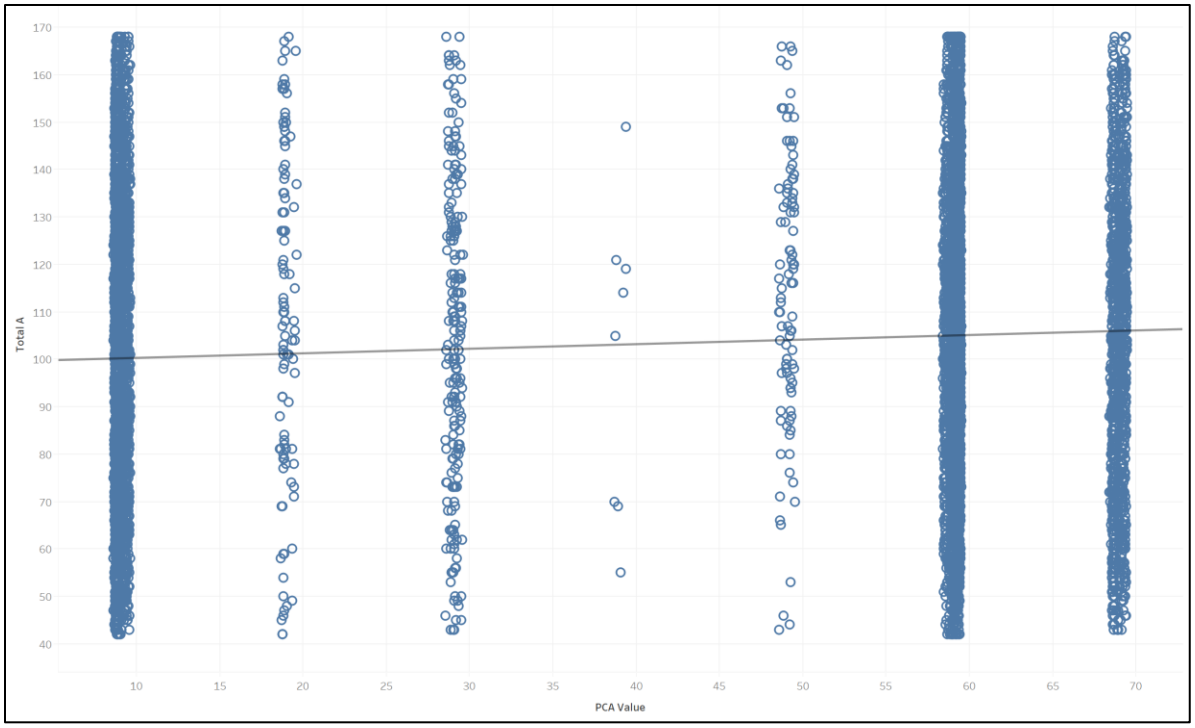


Figure 7.2.1.2 PCA vs TotalA

From *Figure 7.2.1.2* above, we discovered that there are 6 clusters. The clustering shown in this visualisation conforms with the result of K-mean clustering. We found a significant positive linear relationship between **PCA** and **TotalA,** but there are also large residuals in this relationship.

From the undirected knowledge discovery, we believe it is appropriate to build a Linear Regression model to analyse the data further.

7.2.2 Regression Analysis

lrModelV1

Next, we input the train data from the 30/70 splits into our linear regression model. The code successfully ran, and we obtained a linear regression model, **lrModelV1**.

```
from pyspark.ml.regression import LinearRegression
lr = LinearRegression(labelCol='TotalA')
lrModelV1 = lr.fit(train_data_1)
```

```
print("Coefficients: {} Intercept: {}".format(lrModelV1.coefficients,lrModelV1.intercept))
```

```
Coefficients: [-2.933259513057644,-0.5572549268899366,6.4513511927410585,-0.9788370233327931,-0.7003113176596841,2.8095091926904736,-0.4517955299413307,1.9564486330919426,0.015183570323670412,1.866546203965819,2.167981046354314,-0.30109669172135545] Intercept: 105.37065982541041
```

Figure 7.2.2.1 lrModelV1

We can also obtain the coefficients and intercept of this linear regression model. The coefficient of each predictor is illustrated in *Table 7.2.2.2*.

Predictor	Coefficient
education	-2.933259513057644
urban	-0.5572549268899366

gender	6.4513511927410585
engnat	-0.9788370233327931
age	-0.7003113176596841
hand	2.8095091926904736
religion	-0.4517955299413307
orientation	1.9564486330919426
race	0.015183570323670412
voted	1.866546203965819
married	2.167981046354314
familysize	-0.30109669172135545

Table 7.2.2.2 ***lrModelV1 Coefficents***

From the coefficient array, we identified that the coefficients of **urban, age, religion, race, familysize** are less than one. They are 0.1446, -0.2186, -0.3614, 0.0160, -0.4611, respectively. We consider all the predictors with a coefficient less than one as insignificant predictors. The intercept is 105.3707.

Then we further evaluate the model by its residuals, root means squared error and R-squared value.

```
test_results = lrModelV1.evaluate(test_data_1)
test_results.residuals.show()
print("RSME: {}".format(test_results.rootMeanSquaredError))
print("R2: {}".format(test_results.r2))
```

```
+-------------------+
|          residuals|
+-------------------+
|-56.949339398784275|
|-54.954762811821595|
| -47.01549842889095|
|-55.614317693795655|
| -62.49230633908479|
| -52.77900120560129|
| -46.62344696390517|
| -48.07744385471423|
|-54.212969113869974|
|  -45.5055431610124|
| -57.92111496920717|
| -57.26832555657387|
| -54.62857565937121|
| -45.13265679527768|
| -52.93444340819687|
| -54.04843638512148|
| -49.42888443273968|
| -37.34514877767798|
| -65.51013341678815|
|  -65.9256327342539|
+-------------------+
only showing top 20 rows

RSME: 28.71393480395135
R2: 0.08094723299847995
```

Figure 7.2.2.3 ***lrModelV1 Evaluation***

To improve the accuracy of our model, we will eliminate the insignificant predictors from the model.

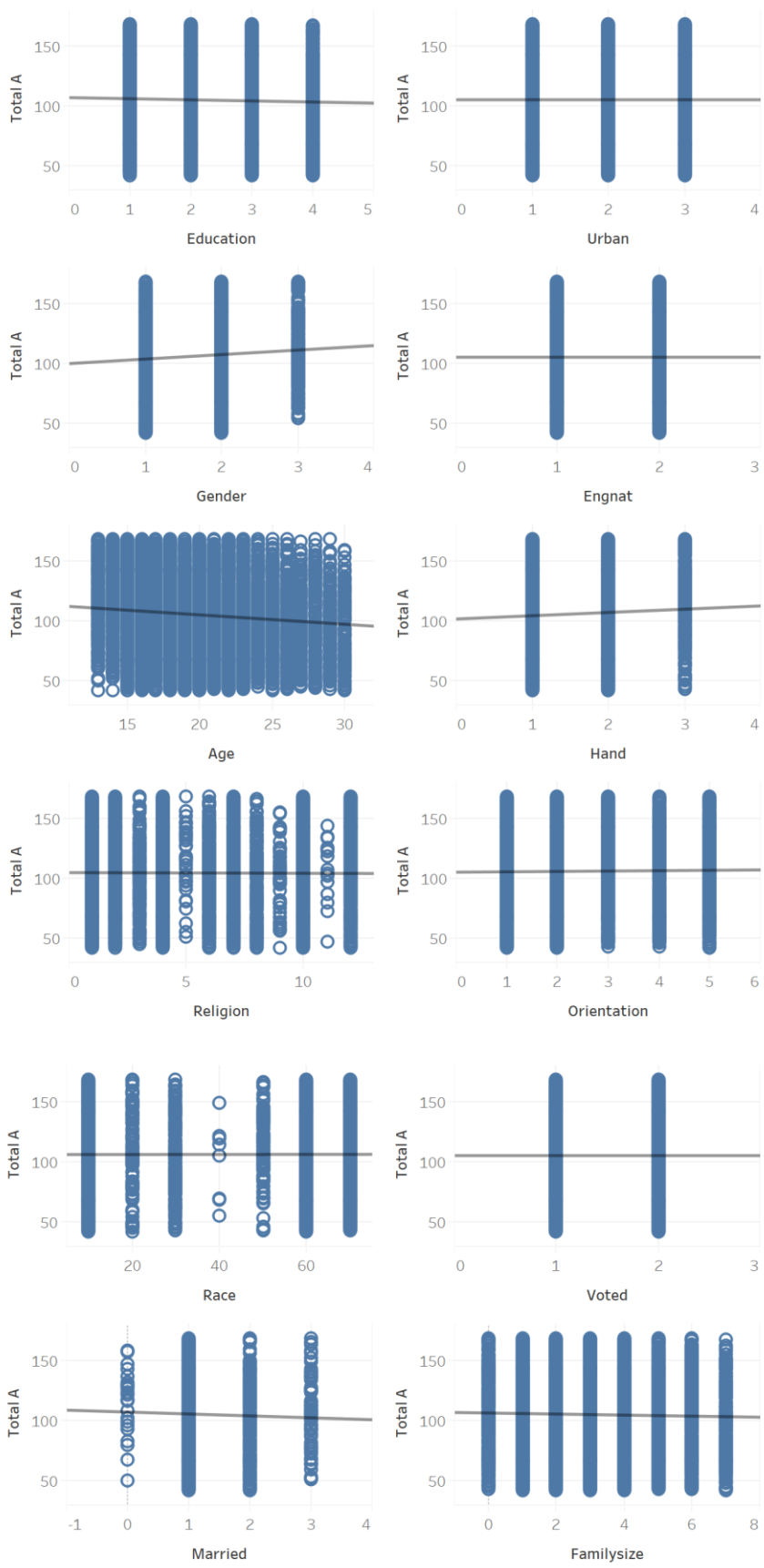


Figure 7.2.2.4 12 Predictor Scatter Plots

We visualised the predictors separately against **TotalA**, and record the p-values for each of the regression lines in the 12 scatter graphs.

Predictor	P-value	R-squared
education	0.532517	0.0007736
urban	1	0.0000
gender	0.118794	0.0070379
engnat	1	0
age	0.0001	0.121387
hand	0.254775	0.0036415
religion	0.823916	0.0000
orientation	0.760255	0.0001505
race	0.965993	0.0000
voted	0.965993	0
married	1	0.0015019
familysize	0.402662	0.0007251

*Table 7.2.2.5 **12 Predictor P-values***

A p-value is the probability of an observed value, assuming that the null hypothesis is true. According to our success criteria, we want to include the predictors that have a p-value less than 0.5. Having a P-value less than 0.5 means the predictor have more 50% of change that it is correlated to the target **TotalA**.

We identified four predictors highlighted above have a p-value smaller than 0.5. Next, we will construct another linear regression model which only contain these 4 predictors.

lrModelV2

We will create another linear regression model **lrModelV2** which exclude the predictors have p-values greater than 0.5.

```
from pyspark.ml.linalg import Vectors
from pyspark.ml.feature import VectorAssembler
assembler = VectorAssembler(
    inputCols=['gender','age','hand','familysize'],
    outputCol="features")
output = assembler.transform(InputData)
dfIrModelV2 = output.select('TotalA','features')
```

Table 7.2.2.6 Selecting 4 Predictors

```
dfIrModelV2.show()
print((dfIrModelV2.count(), len(dfIrModelV2.columns)))
dfIrModelV2.printSchema()
```

```
+------+-----------------+
|TotalA|         features|
+------+-----------------+
|   143|[2.0,16.0,1.0,2.0]|
|   110|[2.0,17.0,1.0,3.0]|
|    91|[2.0,13.0,2.0,5.0]|
|   143|[2.0,19.0,3.0,4.0]|
|    73|[2.0,20.0,1.0,4.0]|
|    56|[2.0,29.0,1.0,2.0]|
|   149|[1.0,16.0,1.0,4.0]|
|   146|[2.0,18.0,1.0,3.0]|
|    82|[1.0,15.0,1.0,1.0]|
|   108|[2.0,18.0,1.0,2.0]|
|    78|[2.0,17.0,1.0,1.0]|
|    97|[2.0,19.0,1.0,2.0]|
|    71|[2.0,18.0,1.0,5.0]|
|    72|[1.0,19.0,3.0,1.0]|
|    87|[2.0,19.0,1.0,2.0]|
|    72|[2.0,15.0,1.0,5.0]|
|    93|[1.0,15.0,1.0,1.0]|
|   136|[2.0,22.0,1.0,1.0]|
|    97|[2.0,19.0,1.0,3.0]|
|    89|[1.0,20.0,1.0,3.0]|
+------+-----------------+
only showing top 20 rows

(9887, 2)
root
 |-- TotalA: integer (nullable = true)
 |-- features: vector (nullable = true)
```

*Table 7.2.2.7 Inputdata - **4 Predictor***

We selected the **gender**, **age**, **hand**, **familysize** columns from the **InputData** data frame, which is then cleaned, integrated dataset we created in *Section 3.4*. The coefficients of this model is shown in *Table 7.2.2.9*.

```
from pyspark.ml.regression import LinearRegression
lr = LinearRegression(labelCol='TotalA')
lrModelV2 = lr.fit(train_dataV2)
print("Coefficients: {} Intercept: {}".format(lrModelV2.coefficients,lrModelV2.intercept))
```

```
Coefficients: [6.957149952114612,-1.3311742824796455,1.8406065517253503,-0.9279598016275632] Intercept: 117.3235359507061
```

*Figure 7.2.2.8 **lrModelV2***

Predictor	Coefficient
gender	6.957149952114612
age	-1.3311742824796455
hand	1.8406065517253503
familysize	-0.9279598016275632

*Table 7.2.2.9 **lrModelV2 Coefficients***

We then evaluate the root mean squared error and R-squared of **lrModelV2**.

```
test_results = lrModelV2.evaluate(test_dataV2)
test_results.residuals.show()
print("RSME: {}".format(test_results.rootMeanSquaredError))
print("R2: {}".format(test_results.r2))
```

```
+-------------------+
|          residuals|
+-------------------+
| -62.29775861409625|
| -60.03862452998905|
| -56.97306148417768|
| -63.00225163466473|
|-57.304235766657314|
| -53.71392740007046|
| -53.82014518846408|
| -53.40161765608224|
| -46.13009618604468|
| -47.58280130844771|
| -63.21189507722097|
| -65.59256000125157|
|  -65.1893455203995|
| -62.00225163466473|
|-57.080768985598226|
| -59.36979881246869|
|-58.966584331616616|
|-53.641887201698026|
| -52.41693070761201|
|  -61.4054661155168|
+-------------------+
only showing top 20 rows

RSME: 29.727558036308036
R2: 0.05424802290488351
```

Figure 7.2.2.9 lrModelV2 Evaluation

We will summarise the patterns of these two models in *Section 7.3* and interpret the outputs in phase 8 Interpretation.

7.3 Search for patterns

From the undirected knowledge discovery, we discovered that there are 6 typical groups of participant and **TotalA** seems to have a positive linear relationship with these 6 groups. Therefore, we reckon it is appropriate to construct a linear regression model to explore the correlations between the characteristics of the participants and their depression level.

We conducted 2 iterations of the linear model.

First, we fit all the 12 predictors into the model and built **lrModelV1**. We can summarise the model by the equation:

TotalA = -2.933* education - 0.5572*urban + 6.4513* gender - 0.9788*engnat -0.7003*age
+ 2.810* hand - 0.4517*religion + 1.956* orientation + 0.0152* race + 1.8665* voted
+ 2.1680* married - 0.3011* familysize + 105.3707

To prepare for the evaluation of the model, we also examine the p-values of the model.

```
from pyspark.ml.regression import GeneralizedLinearRegression
```

```
summary = lrModelV1.summary
print("P Values lrModelV1: " + str(summary.pValues))
```

```
P Values lrModelV1: [1.0752536194758022e-08, 0.3457422640913066, 0.0, 0.0210354
88019047532, 3.2345097222830077e-07, 0.01324119387152467, 0.000145198308538985
2, 2.0159873770353443e-11, 0.29090362673251047, 0.041087696063455637, 0.4778620
0147117874, 0.5786401868354665, 0.0]
```

Figure 7.3.1 lrModelV1 P-values

We sorted the predictors in **lrModelV1** according to the absolute values of their coefficients, as well as recorded negative coefficients in blue (Table 7.3.2). We also documented the p-values of each predictor.

Predictors	lrModelV1	P-values
gender	6.451351193	0.000
education	-2.933259513	0.000
hand	2.809509193	0.0132
married	2.167981046	0.4779
orientation	1.956448633	2.0160
voted	1.866546204	0.0410
engnat	-0.978837023	0.0210
age	-0.700311318	0.000
urban	-0.557254927	0.3457
religion	-0.45179553	0.000
familysize	-0.301096692	0.5786
race	0.01518357	0.2909

Table 7.3.2 lrModelV1 Coefficients In Order

Then we visualised each predictor against the target and found out 4 predictors which have a significant level greater than 50% in relation to **TotalA**. The predictors are **age, gender, hand** and **familysize**. Therefore, we selected these 4 predictors and constructed **lrModelV2**. The pattern we found in this model is:

TotalA = 6.9571*gender -1.3312*age + 1.8406*hand - 0.9280*familysize + 117.3235

We also examine the p-values of the model, as shown in Figure 7.3.3.

```
summary = lrModelV2.summary
print("P Values lrModelV1: " + str(summary.pValues))
```

```
P Values lrModelV1: [0.0, 0.0, 1.842970123622223e-06, 0.009467622154439193, 0.
0]
```

Table 7.3.3 lrModelV2 P-values

We also sorted the predictors in **lrModelV2** with the absolute values their coefficients and showed the negative coefficients in blue (Table 7.3.4).

Predictors	lrModelV2	P-values
gender	6.95715	0.000
hand	1.840607	0.000
age	-1.33117	0.000
familysize	-0.92796	0.0095

Table 7.3.4 lrModelV2 Coefficients In Order

We also notice there are large residual values in both models, as the R-squared of both models are very small (0.08095 and 0.05425 respectively). These residuals are expected, looking at the visualisations in *Figure 7.2.1.2 PCA VS TotalA* and *Figure 7.2.2.4 12 Predictor Scatter Plots.* We will further discuss and interpret these patterns in the next phase - interpretation.

Interpretation

8.1 Study and discuss the mined patterns

In this section, we will be focusing on the outputs of two linear regression models since they are achieving our ultimate data mining goals.

8.1.1 lrModelV1

We will first study the coefficients of **lrModelV1**.

lrModelV1 Summary

TotalA = -2.933* education - 0.5572*urban + 6.4513* gender - 0.9788*engnat -0.7003*age
+ 2.810* hand - 0.4517*religion + 1.956* orientation + 0.0152* race + 1.8665* voted
+ 2.1680* married - 0.3011* familysize + 105.3707

Predictors	lrModelV1	P-values
gender	6.451351193	0.000
education	-2.933259513	0.000

hand	2.809509193	0.0132
married	2.167981046	0.4779
orientation	1.956448633	2.0160
voted	1.866546204	0.0410
engnat	-0.978837023	0.0210
age	-0.700311318	0.000
urban	-0.557254927	0.3457
religion	-0.45179553	0.000
familysize	-0.301096692	0.5786
race	0.01518357	0.2909

From the equation and the table summary of the coefficients above, we notice that among the 12 predictors, 6 of them have positive coefficients, and the other 6 have negative coefficients. The predictors with positive coefficients are **gender**, **hand**, **orientation**, **race**, **voted**, **married**; they have a positive relationship with the target. The other predictors, **education**, **hand**, **married**, **orientation**, **voted**, **engnat**, **age**, **urban**, **religion**, have a negative relationship with the dependent variable.

We order the predictors by the absolute value of their coefficient because the greater the absolute value is, the more **TotalA** change corresponding to a unit change in the predict. We identify the predictor **gender** (Coefficient= 6.4514) will have the greatest influence on **TotalA,** and **race** (Coefficient= 0.01518) will have the least impact on **TotalA**.

In the 12 predictors, **gender, education, hand, voted, engnat, age and religion** have a p-value less than 0.05, which means they have a statistically significant relationship with the dependent variable. Those predictors can reject the null hypothesis that no predictor is correlated to **TotalA**.

We will not be focusing on the intercept of the model. The intercept value is the level of depression, while the other predictors equal to 0. However, as we explained in the previous sections, the categorical predictors cannot be 0 as there is not a category assigned to this index, and age and family size cannot be 0 either.

8.1.2 lrModelV2

Then we will study the coefficients of **lrModelV2**.

lrModelV2 Summary

TotalA = 6.9571*gender -1.3312*age + 1.8406*hand - 0.9280*familysize + 117.3235

Predictors	lrModelV2	P-value
gender	6.95715	0.000
hand	1.840607	0.000
age	-1.33117	0.000
familysize	-0.92796	0.0095

We first identify that **gender** and **hand** have a positive relationship with the target **TotalA**, age, and **familysize** have a negative relationship with **TotalA**.

There are only four predictors in this model, and each has p-values less than 0.5 their relationship with **TotalA**. After fitting them into the model, the p-value of these predictors all reduces below 0.01. The predictors in this model can reject the null hypothesis at the 0.01 level of significance. The intercept is not important in this model, as well.

We reckon **gender** (Coefficient= 6.95715) he largest impact of **Total** as it has the largest coefficient absolute value, and the **familysize** (Coefficient= -0.92796) have the least impact as it has the smallest coefficient absolute value.

8.2 Visualise the data, results, models, and patterns

Coefficients

We first visualise the size of the coefficients of the two models separately.

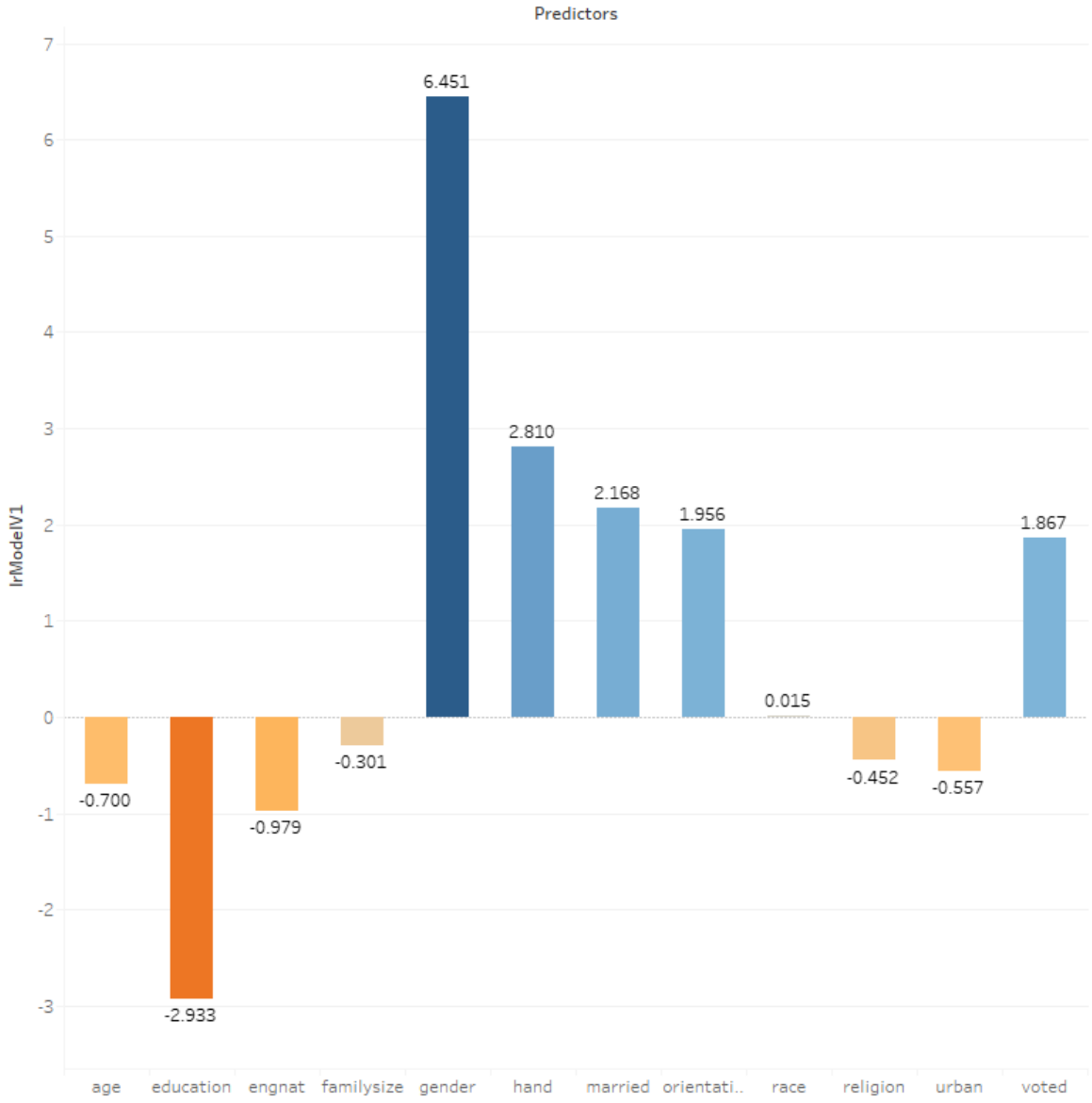


Figure 8.2.1 lrModelV1 Coefficients Visualisation

We fist visualize the coefficient of **lrModelV1**. It is more straight forward for us to understand each coefficient. The blue coefficients indicate a positive relationship and the orange group indicate a negative relationship. The impact of **gender** is the largest on **TotalA** as the magnitude of its coefficient is the largest, and the **race** will have the least impact as it has the smallest bar in *Figure 8.2.1*.

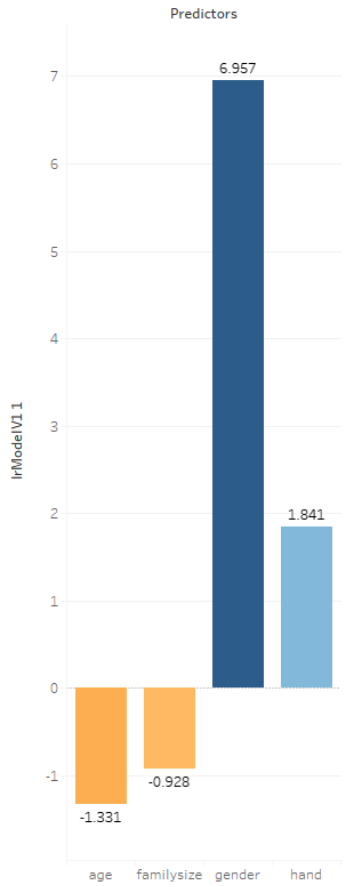


Figure 8.2.2 IrModelV2 Coefficients Visualisation

There are only 4 predictors in **IrModelV2,** and the predictor **gender** still have the greatest coefficient. Moreover, the magnitude of the gender coefficient increases from 6.451 to 6.957. In addition, the absolute values of **age** and **familysize** also increase from 0.700 to 1.331 and 0.301 to 0.928, respectively. The absolute value of hand coefficient decreases from 2.810 to 1.841.

P-values

We also visualize the p-value for each predictor.

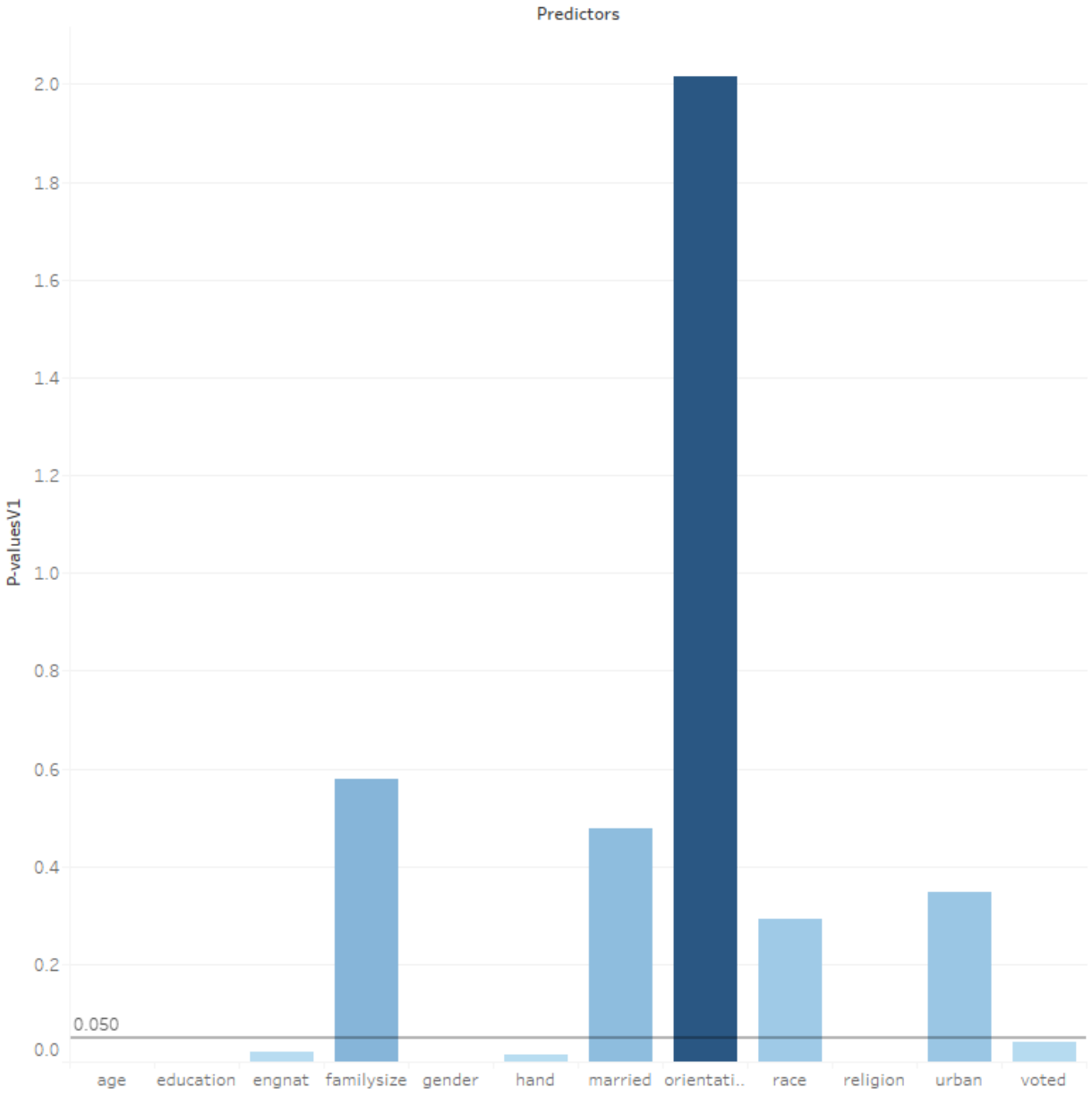
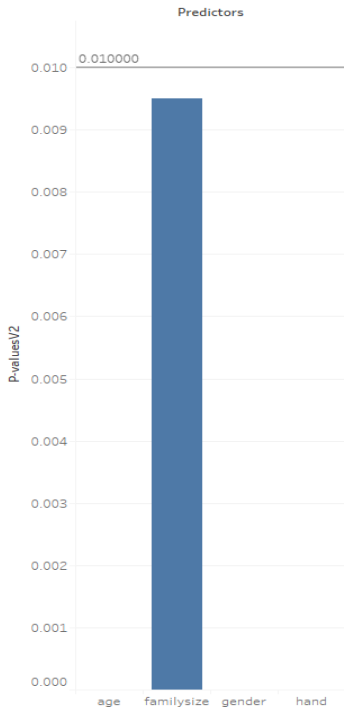
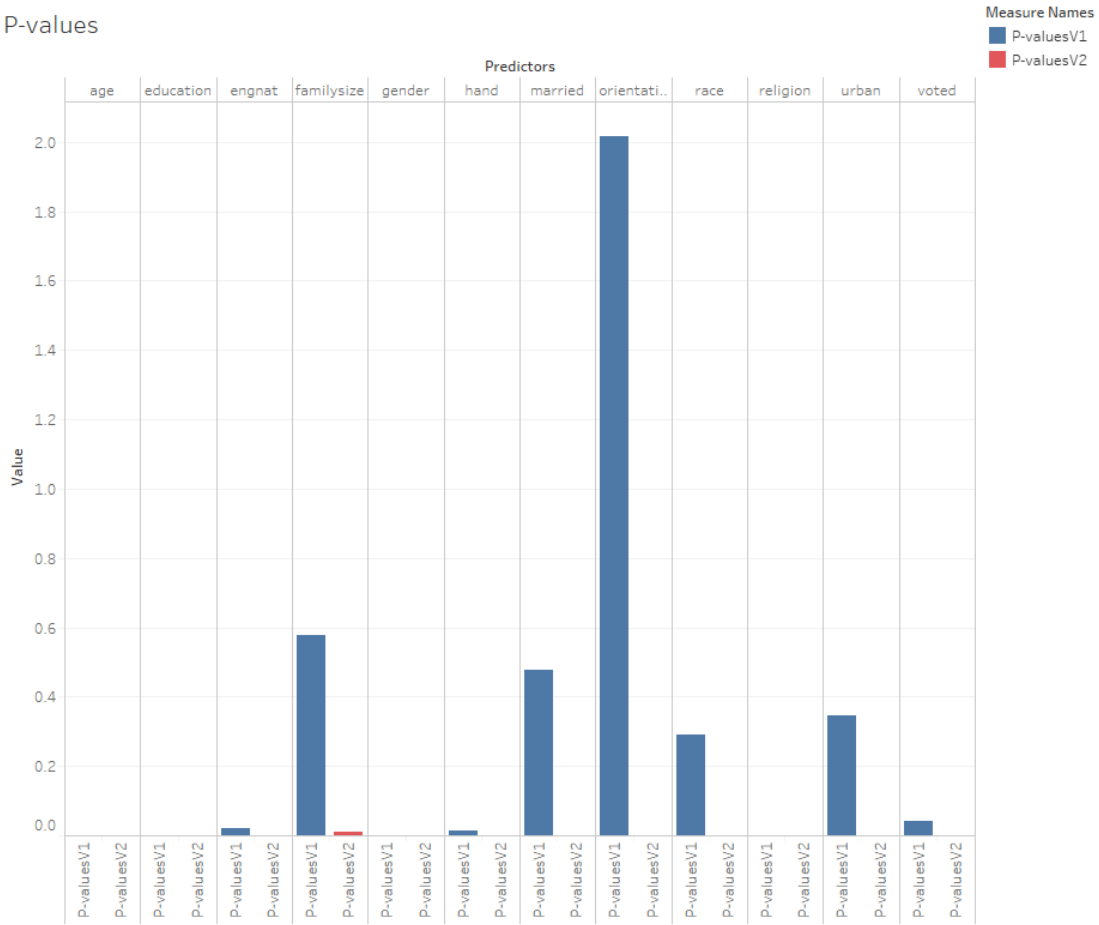


Figure 8.2.1 lrModelV1 P-values Visualisation

In **lrModelV1**, we identified that **age, education, engnat, gender, hand, religion** and **voted** can have a p-value less than 0.05. Hence, these predictors have a significant relationship with **TotalA** and can reject the null hypothesis. When we generate the final checklist of personal characteristics that predict depression level, we should only include these factors that have a significant relationship with **TotalA**.



All the predictors in **IrModelV2** have a p-value less than 0.01. They can reject the null hypothesis and have a strong relationship with the target **TotalA**. We can include all the 4 factors in the checklist if we choose to accept the result of **IrModelV2**.We will compare the two models in the later stage.



It is difficult to visualise the p-values of **IrModelV1** and **IrModelV2** in the same axis because

there is a large difference in the p-values of the two models. The p-values in **lrModelV2** is much lower than in **lrModelV1**.

We will interpret each variable and the meaning of its coefficient in the next section.

8.3 Interpret the results, models, and patterns

As we discussed at the beginning of this study, we believe understanding the meaning of each column of our dataset is important. We will interpret the equation we obtained from **lrModelV1** and **lrModelV2** with the metadata:

```
education       "How much education have you completed?", 1=Less than high school, 2=High school, 3=University degree, 4=Graduate degree
urban           "What type of area did you live when you were a child?", 1=Rural (country side), 2=Suburban, 3=Urban (town, city)
gender          "What is your gender?", 1=Male, 2=Female, 3=Other
engnat          "Is English your native language?", 1=Yes, 2=No
age                "How many years old are you?"
hand            "What hand do you use to write with?", 1=Right, 2=Left, 3=Both
religion        "What is your religion?", 1=Agnostic, 2=Atheist, 3=Buddhist, 4=Christian (Catholic), 5=Christian (Mormon), 6=Christian (P
orientation     "What is your sexual orientation?", 1=Heterosexual, 2=Bisexual, 3=Homosexual, 4=Asexual, 5=Other
race            "What is your race?", 10=Asian, 20=Arab, 30=Black, 40=Indigenous Australian, 50=Native American, 60=White, 70=Other
voted           "Have you voted in a national election in the past year?", 1=Yes, 2=No
married         "What is your marital status?", 1=Never married, 2=Currently married, 3=Previously married
familysize      "Including you, how many children did your mother have?"
major           "If you attended a university, what was your major (e.g. "psychology", "English", "civil engineering")?"
```

Figure 8.3.1 Meta Data

We summarise the interpretations of the coefficients of **lrModelV1** in the table below:

Predictors	lrModelV1	Interpretation
gender	6.451351193	Holding the other variables constant, a female person has a depression level 6 points higher than a male participant.
education	-2.933259513	Holding the other variables constant, a person with relatively higher education would have a lower depression level. The depression level would decrease by 3 points, if the education level increases by 1 stage from Less than high school to High school, High school to University degree, or University degree to Graduate degree.
hand	2.809509193	Holding the other variables constant, the depression level would 3 points higher, if a person who write with the left hand than the right hand, or a person write with
married	2.167981046	Holding the other variables constant, a married person has a depression level 2 points lower than a never-married participant.
orientation	1.956448633	Holding the other variables constant, the depression level is 2 points higher is we compare a bisexual with a heterosexual person, a homosexual with a bisexual person, or an asexual with a homosexual
voted	1.866546204	Holding the other variables constant, a person who did not vote in a national election in the past years has a depression level 2 point higher than a person who voted.
engnat	-0.978837023	Holding the other variables constant, a person whose native language is English would have a depression level 2 points

		higher than a person with other native languages.
age	-0.700311318	Holding the other variables constant, the depression level would decrease by 0.7 points if the age of the person increases by one.
urban	-0.557254927	Holding the other variables constant, the depression level will decrease by 0.5 points if we compare a person who lives in suburban than in rural or live in urban than suburban.
religion	-0.45179553	The indexes of religions are: 1=Agnostic, 2=Atheist, 3=Buddhist, 4=Christian (Catholic), 5=Christian (Mormon), 6=Christian (Protestant), 7=Christian (Other), 8=Hindu, 9=Jewish, 10=Muslim, 11=Sikh, 12=Other Holding the other variables constant, the depression level will decrease by 0.5 points if we compare the religion with a higher index than the lower index.
familysize	-0.301096692	Holding the other variables constant, if the one's mother has 1 more child than another person, the person will 0.3 points less in depression.
race	0.01518357	The race indexes are: 10=Asian, 20=Arab, 30=Black, 40=Indigenous Australian, 50=Native American, 60=White, 70=Other If the race index increases by 10, the person will be 0.1 points higher in depression level.

Figure 8.3.2 Interpretation of Coefficients lrModelV1

We summarise the interpretations of the coefficients of **lrModelV2** in the table below:

Predictors	lrModelV2	Interpretation
gender	6.95715	Holding the other variables constant, a female person has a depression level 7 points higher than a male participant.
hand	1.840607	Holding the other variables constant, the depression level would 2 points higher, if a person who write with the left hand than the right hand, or a person write with
age	-1.33117	Holding the other variables constant, the depression level would decrease by 1 point if the age of the person increases by one.
familysize	-0.92796	Holding the other variables constant, if the one's mother has 1 more child than another person, the person will 1 point less in depression.

Figure 8.3.3 Interpretation of Coefficients lrModelV2

The intercept can be explained as the value of **TotalA** if the values of all the predictors equal to zero. However, as we discussed before, zero is not assigned to any meaning for those six variables. Therefore, we believe it is not useful to interpret the intercept.

The **lrModelV1** model examines 12 characteristics of the participants, and there are 7 of them

(**age, education, engnat, gender, hand, religion** and **voted**) show a significant relationship with the depression level at 0.05 significant level. The **lrModelV2** include 4 characteristics but can reject the null hypothesis and show a relationship with depression level at 0.01 significance level. **lrModelV1** has weaker evidence to reject the null hypothesis but include more predictors.

8.4 Assess and evaluate results, models, and patterns

8.4.1 Comparing the two models

Model	Number of significant predictors	Level of significance	Root mean squared error	R-squared
lrModelV1	7	0.05	28.7139	0.0809
lrModelV2	4	0.01	29.0552	0.0488

Table 8.4.1.1 Comparing The Two Models

We compare the two models with a different number of significant predictors, level of significance, root means squared error and r-squared, as shown in *Table 8.4.1.1*. From the comparison, we reckon **lrModelV1** is a better model to achieve our business and data mining objectives.

Even though **lrModelV2** has stronger evidence at 0.01 significant level, a significant level at 0.05 is acceptable comparing to our success criteria of that “we expect the model to describe a significant correlation between the characters of depressed people and their depression level”. Besides, there are more predictors included in lrModelV1, which will include more aspects of a person’s characteristics and will achieve our goal of increase the awareness of the public towards depression issues. Besides, the root means squared error of **lrModelV1** is less than **lrModelV2**; the r-squared of **lrModelV1** is higher than **lrModelV2**.

These two comparisons explain that **lrModelV1** has higher accuracy than **lrModelV2.** From the discussion above, we will adopt **lrModelV1** as the model for predicting the level of depression level. The seven significant predictors, **age, education, engnat, gender, hand, religion** and **voted,** will be the characteristics we include in our checklist. The final checklist and the interpretation of each characteristic are illustrated below in *Table 8.4.1.2.*

Checklist		
education	“How much education have you completed?” 1=Less than high school, 2=High school, 3=University degree, 4=Graduate degree	Holding the other variables constant, a person with relatively higher education would have a lower depression level. The depression level would decrease by 3 points, if the education level increases by 1 stage from Less than high school to High school, High school to University degree, or University degree to Graduate degree.
gender	What is your gender? 1=Male, 2=Female, 3=Other	Holding the other variables constant, a female person has a depression level 6 points higher

		than a male participant.
engnat	Is English your native language? 1=Yes, 2=No	Holding the other variables constant, a person whose native language is English would have a depression level 2 points higher than a person with other native languages.
age	How many years old are you?	Holding the other variables constant, the depression level would decrease by 0.7 point if the age of the person increases by one.
hand	What hand do you use to write with? 1=Right, 2=Left, 3=Both	Holding the other variables constant, the depression level would 3 points higher, if a person who write with left hand than right hand, or a person write with
religion	What is your religion?, 1=Agnostic, 2=Atheist, 3=Buddhist, 4=Christian (Catholic), 5=Christian (Mormon), 6=Christian (Protestant), 7=Christian (Other), 8=Hindu, 9=Jewish, 10=Muslim, 11=Sikh, 12=Other	Holding the other variables constant, the depression level will decrease by 0.5 points if we compare the religion with higher index than lower index.
race	What is your race? 10=Asian, 20=Arab, 30=Black, 40=Indigenous Australian, 50=Native American, 60=White, 70=Other	If the race index increases by 10, the person will be 0.1 point higher in depression level.
voted	Have you voted in a national election in the past year? 1=Yes, 2=No	Holding the other variables constant, a person who did not vote in a national election in the past years has a depression level 2 point higher than a person who voted.

Table 8.4.1.2 Checklist

8.4.2 Evaluating the Project

We will further evaluate our project and the **IrModelV1** model, as shown in the table below.

Evaluation Question	Response
Are the results stated clearly and in a form that can be easily presented?	Yes. The results from the data mining are presented by plots, tables and a summarised equation.
Are there particularly novel or unique findings that should be highlighted?	Yes. We identified 7 personal characteristics that is significantly correlated with depression level, which are **age, education, engnat, gender, hand, religion,** and **voted.**

	The detailed explanation of the relationship between each characteristic and depression, are stated in *Section 8.3*.
Can we rank the models and findings in order of their applicability to the business goals?	Yes. Our business goals are: • Explore the correlations between the level of depression and the characters of people who have depression problems • Investigate and predict which kind of people are likely to have depression issues • Generate a depression checklist which would increase the awareness of people towards depression so that they can take action to stop the development of depression at an early stage We have explored the correlations with two linear regression model **lrModelV1** and **lrModelV2.** We also discussed that the **lrModelV1** model is the best model to achieve our objectives. We have discussed the relationship between people's characteristics and their depression level in details in *Section 8.3*.
In general, how well do these results answer the business goals?	In general, the linear regression model - **lrModelV1**, has answered the business goal very well, as it clearly explained the correlations between the level of depression and the characters of people who have depression problems. We also state that we consider the model as a success if we can identify at least 5 characteristics that are highly correlated with depression level, while our model generated 7 predictors. However, we must identify the limitation of our model. As we discussed before, the original dataset does not have the same number of records for each class of participants. The model may be biased towards the group with less information. We conclude that our result cannot be used as medical evidence. This study would give a direction for medical researchers to conduct further researches in this area of mental health.
Any additional question raised?	After we identified the 7 characteristics that correlated to depression level, we would further question that: • What are the reasons behind the correlations we identified? • What can we decrease the depression level of a person with specific unchangeable characteristics? • How can we improve people's mental health in general?
Contributions	The seven characteristics we identified can be used as a self-

	checklist. People with characteristics associated with high depression level will be more aware of their mental health. Moreover, this study provides a direction for medical researchers to conduct further investigation of depression.

8.4.2 Reviewing the Process

Reviewing Questions	Response
Did the stages contribute to the value of the final results?	All stages contributed to the value of the results. Stage 1 Business Understanding - allow us to understand the background of our study and set up the business and defaming goals. Stage 2 Data Understanding – allow us to explore the content of our dataset, as well as any patterns, errors hiding in the dataset. Stage 3 Data Preparation – allow us to identify the relevant data we need, clean our data, integrate and format the data in the appropriate form to conduct data mining. Stage 4 Data Preparation – allow us to reduce the amount of input data and discover any patterns hiding in the dataset preliminarily. Stage 5 and 6 Data Mining Method and Algorism Selection – allow us to select the adequate data mining method and algorism, to conduct a data mining which will achieve our business and data mining objectives. Stage 7 Data Mining – allow us to identify and discover patterns from the data and build models that describe and predict reality. Stage 8 Interpretation – allow us to have a clear understanding of our output from the model and evaluate our models and the study in general. In conclusion, Stage 1 determine the goals and directions of this study, and the other stages gradually transform the raw data to insights that accomplish the goals we set. The intelligence density increases as the process progress.
Are there ways to streamline or improve the operation?	We identify that there are repetitive steps in Stage 2, 3 and 4. We can conduct these three stages simultaneously so that we can reduce repetitive work. In addition, we integrated the datasets after we cleaned the two datasets (**Depression Scale** and **Personal Information**) separately. It would be much efficient if we integrate the datasets first before the data cleaning process. In this way, we can clean all the data at the same time.

Any failure or mistake?	Our data mining objective is to identify at least 5 critical predictors. After we build the **IrModelV1**, we realised that it is very likely that we cannot obtain enough significant predictors by choosing only 12 predictors. We may need to identify more predictors that may be correlated to the depression level next time.
Any surprises?	We are a bit surprised by the results we obtained from the Regression Analyses. We did not realise that **engnat** (whether English is one's native language), **hand** (the hand one use to write) and **voted** (whether one voted in a national election) would have a significant correlation with depression level.
Alternative decisions	If we have enough computer power and memory, we would like to integrate our data first before any cleaning. In addition, we will select more 12 predictors that are likely to be correlated to **TotalA.**

8.5 Iterate prior steps (1 – 7) as required

We have conducted a few iterations in prior steps.

Step	Iteration	Purpose
Step 6.3 Build/Select appropriate model(s) and choose relevant parameter(s)	3 iteration of k-mean clustering design	Finding the appropriate number of clusters that generate the most accurate k-means model
Step 7.1 Create and justify test designs	2 iteration of partition	Finding the appropriate data partition for the linear regression model to generate the most accurate prediction
Step 7.2 Conduct data mining	2 iteration of modelling	Construct models with different predictors and select the best model that achieve our business and data mining objectives.

References

Berry, M. J. A., & Linoff, G. S. (1997). Data Mining Techniques: For Marketing, Sales, and Customer Support: Wiley

CDC. (2019). Disability and Health Related Conditions. Retrieved from https://www.cdc.gov/ncbddd/disabilityandhealth/relatedconditions.html

Kaggle. (2020). Depression Anxiety Stress Scales Responses [csv file, collected from Depression Anxiety Stress Scales 2017 to 2019]. Retrieved from https://www.kaggle.com/lucasgreenwell/depression-anxiety-stress-scales-responses

Marr, B. (2016). Big Data in Practice: How 45 Successful Companies Used Big Data Analytics to Deliver Extraordinary Results. ISBN: 978-1-119-23138-7

Singh Gill, R. (2014). Neural Networks in Data Mining. IOSR Journal of Engineering, 4(3), 1-6. doi:10.9790/3021-04360106

World Health Organization (2020). Depression. Retrieved from https://www.who.int/news-room/fact-sheets/detail/depression

Disclaimer

I acknowledge that the submitted work is my own original work in accordance with the University of Auckland guidelines and policies on academic integrity and copyright.