CCT College

Programme Title:	BSc (Hons) in Computing in IT (4th Yr)
Module Title(s):	Data Exploration & Preparation
Assignment Title:	CA1 Project
Lecturer(s):	Dr. Muhammad Iqbal
Submission Deadline Date:	3rd December 2023 11:59pm
Student Name:	Pedro Henrique Simoes Marcal
Student Email:	2020300@student.cct.ie
Student Number:	2020300

Contents

Introduction

This is a pair-based project (Max 2 students) using R programming language or any other language of your choice. Analyse a specific problem only in the one of following areas:

- Crime
- Covid 19
- Dublin Transport

The dataset should have at least 7000 rows and 10 columns after cleaning and there is not any upper bound. The type of question(s) that you should formulate for the project will depend on the chosen domain of the dataset that your pair is considering for the Data Exploration and Preparation (DEP) project. The objectives of the DEP project are based on the domain knowledge of data. The pair would need to complete the following tasks during the development of this pair project.

Report

a) Identify which variables are categorical, discrete and continuous in the chosen data set and show using some visualization or plot. Explore whether there are missing values for any of the variables.

First, we will have to prepare the data to identify missing and wrong values in our dataset:

```
# This dataset contains information about vaccination rates over Ireland

# Display data
head(data)

# Check structure
str(data)
```

```
4         2704          16    %     2
5          247 5 - 11 years    %
6         2704          34    %   2.2
> # Check structure
> str(data)
'data.frame':	35856 obs. of  10 variables:
 $ STATISTIC          : chr  "CDC45C01" "CDC45C01" "CDC45C01" "CDC45C01" ...
 $ Statistic.Label    : chr  "Fully vaccinated" "Fully vaccinated" "Fully vaccinated" "Fully vaccinated" ...
 $ TLIST.M1.          : int  202101 202101 202101 202101 202101 202101 202101 202101 202101 202101 ...
 $ Month              : chr  "2021 January" "2021 January" "2021 January" "2021 January" ...
 $ C03898V04649       : chr  "2ae19629-3eff-13a3-e055-000000000001" "2ae19629-3eff-13a3-e055-000000000001" "2ae19629-3f00-13a3-e055-000000000001" "2ae19629-3f00-13a3-e055-000000000001" ...
 $ Local.Electoral.Area: chr  "Borris-In-Ossory-Mountmellick, Laois" "Borris-In-Ossory-Mountmellick, Laois" "Portlaoise, Laois" "Portlaoise, Laois" ...
 $ C02076V03371       : int  247 2704 247 2704 247 2704 247 2704 247 2704 ...
 $ Age.Group          : chr  "5 - 11 years" "25" "5 - 11 years" "16" ...
 $ UNIT               : chr  "%" "%" "%" "%" ...
 $ VALUE              : chr  "" "2.8" "" "2" ...
> []
```

Here we can analyse some wrongness on our dataset, such as ‘Age.Group’ as a char when it should be a numeric value and the range of age is ‘5 – 11 years’ so I will replace that with different ages between 5 and 11 years to make it easier and the ‘VALUE’ also as a char when should be numeric value.

Using skimr, we can observe that the dataset has 0 ‘n_missing’ values, which means no values are missing.

```
install.packages("devtools")
devtools::install_github("ropensci/skimr")

library(ggplot2)
library(skimr)
skimr::skim(data)
```

```
  numeric                2
________________________
Group variables          None

── Variable type: character ──────────────────────────────────────────────
  skim_variable        n_missing complete_rate min max empty n_unique whitespace
1 STATISTIC                    0             1   8   8     0        6          0
2 Statistic.Label              0             1  16  76     0        6          0
3 Month                        0             1   8  14     0       18          0
4 C03898V04649                 0             1  36  36     0      166          0
5 Local.Electoral.Area         0             1  11  44     0      166          0
6 Age.Group                    0             1   1   2     0       56          0
7 UNIT                         0             1   1   1     0        1          0
8 VALUE                        0             1   0   4 21913      951          0

── Variable type: numeric ────────────────────────────────────────────────
  skim_variable n_missing complete_rate    mean     sd     p0    p25     p50    p75   p100
1 TLIST.M1.             0             1 202139.   45.8 202101 202105 202110. 202202 202206
2 C02076V03371          0             1   1476. 1229.     247    247   1476.   2704   2704
  hist
1 ▇▁▁▁▂
2 ▇▁▁▁▇
> []
```

Before transforming into numeric value, I will assign the random ages between 5 and 11 years old where it says '5 – 11 years':

```r
# Replace '5 - 11 years' with random ages between 5 and 11
data$Age.Group <- ifelse(data$`Age.Group` == '5 - 11 years',
                  sample(5:11, sum(data$`Age.Group` == '5 - 11 years'), replace = TRUE),
                  data$Age.Group)

# Viewing the updated 'Age' column
head(data$Age.Group)
```

```
 $ STATISTIC           : chr  "CDC45C01" "CDC45C01" "CDC45C01" "CDC45C01" ...
 $ Statistic.Label     : chr  "Fully vaccinated" "Fully vaccinated" "Fully vaccinated" "Fully vaccinated" ...
 $ TLIST.M1.           : int  202101 202101 202101 202101 202101 202101 202101 202101 202101 202101 ...
 $ Month               : chr  "2021 January" "2021 January" "2021 January" "2021 January" ...
 $ C03898V04649        : chr  "2ae19629-3eff-13a3-e055-000000000001" "2ae19629-3eff-13a3-e055-000000000001" "2ae19629-3f00-13a3-e055-000000000001" "2ae19629-3f00-13a3-e055-000000000001" ...
 $ Local.Electoral.Area: chr  "Borris-In-Ossory-Mountmellick, Laois" "Borris-In-Ossory-Mountmellick, Laois" "Portlaoise, Laois" "Portlaoise, Laois" ...
 $ C02076V03371        : int  247 2704 247 2704 247 2704 247 2704 247 2704 ...
 $ Age.Group           : chr  "5 - 11 years" "25" "5 - 11 years" "16" ...
 $ UNIT                : chr  "%" "%" "%" "%" ...
 $ VALUE               : chr  "" "2.8" "" "2" ...
> # Replace '5 - 11 years' with random ages between 5 and 11
> data$Age.Group <- ifelse(data$`Age.Group` == '5 - 11 years',
+                  sample(5:11, sum(data$`Age.Group` == '5 - 11 years'), replace = TRUE),
+                  data$Age.Group)
> # Viewing the updated 'Age' column
> head(data$Age.Group)
[1] "8"  "25" "8"  "16" "9"  "34"
>
```

Structure of the cleaned and updated dataset:

```
> str(data)
'data.frame':	35856 obs. of  11 variables:
 $ STATISTIC           : chr  "CDC45C01" "CDC45C01" "CDC45C01" "CDC45C01" ...
 $ Statistic.Label     : chr  "Fully vaccinated" "Fully vaccinated" "Fully vaccinated" "Fully vaccinated" ...
 $ TLIST.M1.           : int  202101 202101 202101 202101 202101 202101 202101 202101 202101 202101 ...
 $ Year                : chr  "2021" "2021" "2021" "2021" ...
 $ Month               : chr  "January" "January" "January" "January" ...
 $ Local.Electoral.Area: chr  "Borris-In-Ossory-Mountmellick, Laois" "Borris-In-Ossory-Mountmellick, Laois" "Portlaoise, Laois" "Portlaoise, Laois" ...
 $ C02076V03371        : int  247 2704 247 2704 247 2704 247 2704 247 2704 ...
 $ Age.Group           : num  9 46 9 15 5 19 6 56 5 23 ...
 $ UNIT                : chr  "%" "%" "%" "%" ...
 $ VALUE               : chr  "" "2.8" "" "2" ...
 $ Vaccination_Code    : num  1 1 1 1 1 1 1 1 1 1 ...
>
```

Continuous variables are those variables in our dataset that can be measured, it can be any value. In our dataset, 'Age.Group' and 'Vaccination_Code' can be considered continuous variables since we can measure.

Categorical variables are the variables in our dataset that are not values but rather anyway to describe something. In our dataset, 'Local.Electoral.Area' and 'Month' can be considered as categorical variables since they giving description of our data, which month the vaccination occurred and where is based the Electoral Area of given person.

Discrete variables are the variables that contain a specified set of values and it can also be counted, only values specified can be considered allowed. In our dataset we can consider 'Age.Group' as a discrete variable since the range of age goes from 5-11 and 12 and over years old.

b) Calculate the statistical parameters (mean, median, minimum, maximum, and standard deviation) for each of the numerical variables.

Age:

```r

# Calculate mean, median, minimum, maximum, and standard deviation for Age
mean_Age <- mean(data$Age.Group)
median_Age <- median(data$Age.Group)
min_Age <- min(data$Age.Group)
max_Age <- max(data$Age.Group)
sd_Age <- sd(data$Age.Group)

# Print the results
cat("\nMean of Age:", mean_Age, "\n")
cat("Median of Age:", median_Age, "\n")
cat("Minimum of Age:", min_Age, "\n")
cat("Maximum of Age:", max_Age, "\n")
cat("Standard Deviation of Age:", sd_Age, "\n")


```

89:1 (Top Level) R Script

Console Terminal Background Jobs

R 4.3.1 · ~/

```
> cat("\nMean of Age:", mean_Age, "\n")

Mean of Age: 22.18827
> cat("Median of Age:", median_Age, "\n")
Median of Age: 11.5
> cat("Minimum of Age:", min_Age, "\n")
Minimum of Age: 5
> cat("Maximum of Age:", max_Age, "\n")
Maximum of Age: 60
> cat("Standard Deviation of Age:", sd_Age, "\n")
Standard Deviation of Age: 17.41683
```

Value:

```r
# Calculate mean, median, minimum, maximum, and standard deviation for Value
mean_value <- mean(data$VALUE)
median_value <- median(data$VALUE)
min_value <- min(data$VALUE)
max_value <- max(data$VALUE)
sd_value <- sd(data$VALUE)

# Print the results
cat("\nMean of VALUE:", mean_value, "\n")
cat("Median of VALUE:", median_value, "\n")
cat("Minimum of VALUE:", min_value, "\n")
cat("Maximum of VALUE:", max_value, "\n")
cat("Standard Deviation of VALUE:", sd_value, "\n")
```

```
> cat("\nMean of VALUE:", mean_value, "\n")

Mean of VALUE: 8.496528
> cat("Median of VALUE:", median_value, "\n")
Median of VALUE: 0
> cat("Minimum of VALUE:", min_value, "\n")
Minimum of VALUE: 0
> cat("Maximum of VALUE:", max_value, "\n")
Maximum of VALUE: 99.4
> cat("Standard Deviation of VALUE:", sd_value, "\n")
Standard Deviation of VALUE: 21.89832
```

c) Apply Min-Max Normalization, Z-score Standardization and Robust scalar on the numerical data variables.

To calculate the Min-Max Normalization, Z-Score Standardization and Robust Scalar I have just used R standard library, no extra libraries were needed since RStudio already has the tools needed. Here is the piece of code used for this task:

```r
# Numerical variables
numerical_variables <- c("Age.Group")

# Min-Max Normalization
minmax_values <- (data[, column_name] - min(data[, column_name], na.rm = TRUE)) /
  (max(data[, column_name], na.rm = TRUE) - min(data[, column_name], na.rm = TRUE))

# Z-score Standardization
zscore_values <- scale(data[, column_name])

# Robust Scaling
robust_values <- (data[, column_name] - median(data[, column_name], na.rm = TRUE)) / IQR(data[, column_name], na.rm = TRUE)
```

“numerical_variables” is used to store the column we want to perform the tasks.

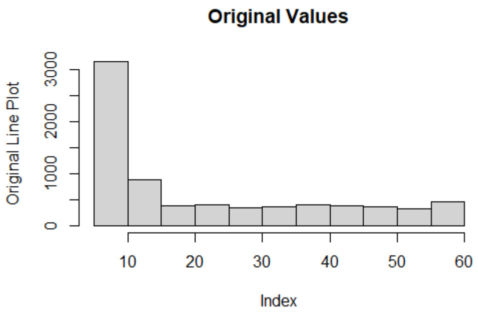
“minmax_values” is used to store the outcome of the Min-Max Normalization. To calculate our minimum value we use “min (data [, column_name])”, min will calculate the minimum value within our chosen column and to calculate the maximum value we use “max(data[, column_name]”.

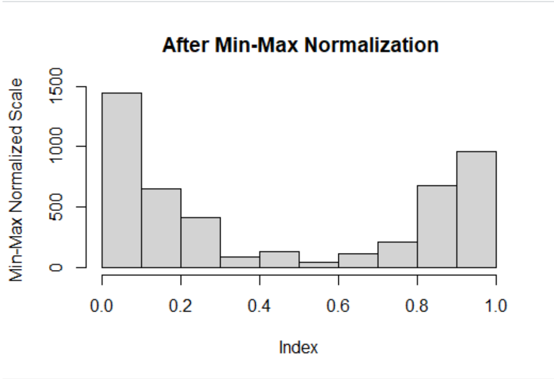
“zscore_values” is used to store the outcome of our Z-Score Standardization where only the “scale” function is needed to calculate the Z-Score Standardization.

“robust_values” is used to store the outcome of your Robust Scalar where “median (data [, column_name]” is used to calculate the median of given data and “IQR (data [, column_name]” is used to calculate the interquartile of given data. “na.rm = TRUE” is used to exclude any missing values in our data.

Min-Max normalization:

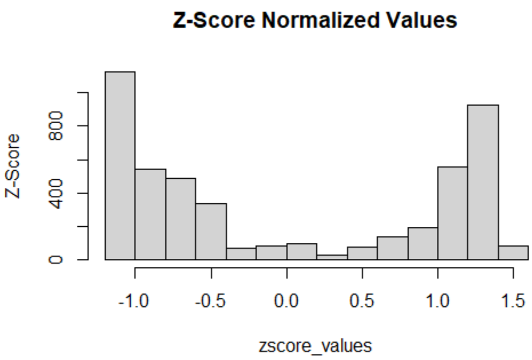
Min-Max normalization is the process of transforming the variables in range values between 0 and 1.





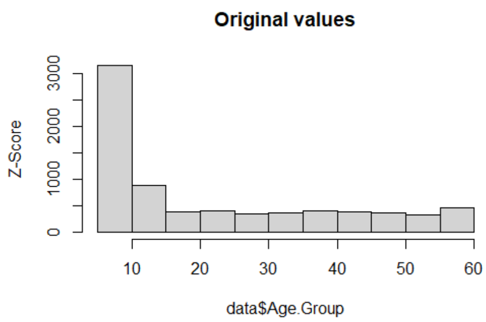
Z Score Standardization:

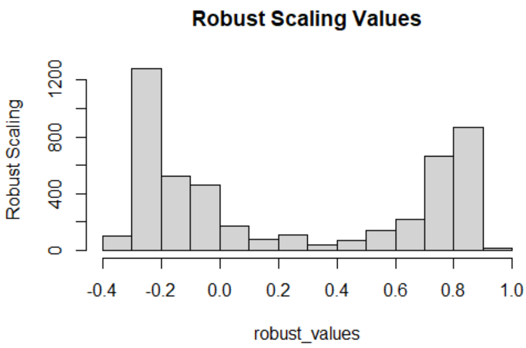
Z Score Standardization is the process of transforming the values in a way that the mean becomes 0 and the standard deviation becomes 1. Z Score basically says how far away the standard deviation is from the mean.



Robust Scaling:

Robust Scaling is the process of transforming the values in a way that center the data around the median and normalize it by interquartile range making the data more robust to outliners.



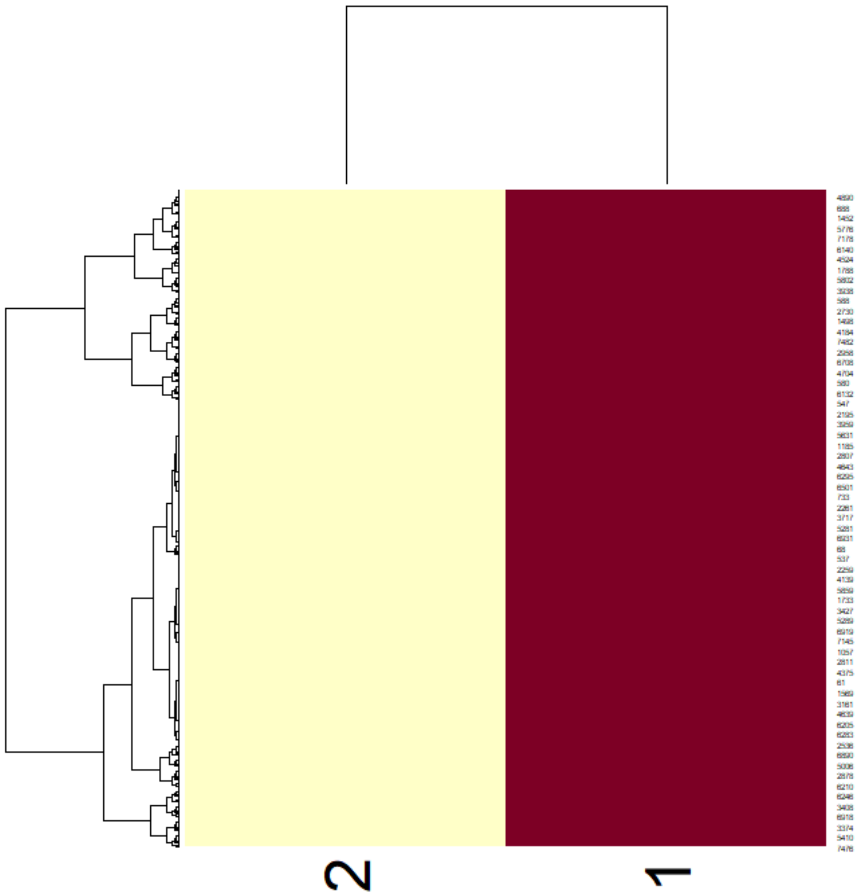


d) Line, Scatter and Heatmaps can be used to show the correlation between the features of the dataset.

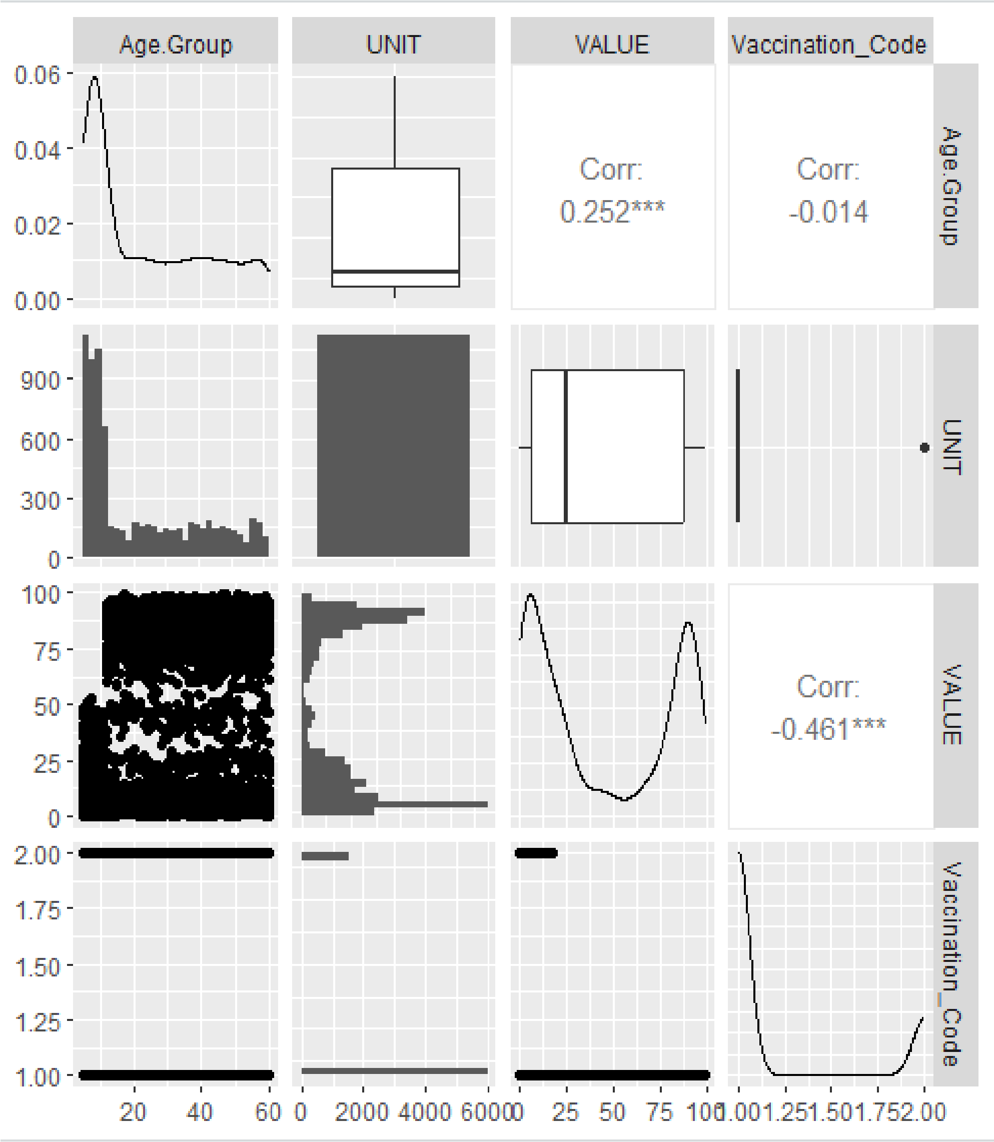
To accomplish this task, the library “corrplot” was needed, since it is very efficient and easy to use when creating heatmaps. First, we bind the data by Age and Vaccination Code and next we just assign it into a variable and use the “corrplot” to create our correlation heatmap.

```
data_2 <- cbind(data$Age.Group, data$Vaccination_Code)
ggpairs(data_2)

corr_coefficients = data_2
corr_coefficients

heatmap(corr_coefficients)

# Calculate the correlation matrix
cor_matrix <- cor(data_2, use = "complete.obs")

# Create a correlation heatmap
corrplot(cor_matrix, method = "color", title = "Correlation Heatmap")
```

Heatmap between the columns “Age.Group” and “Vaccination_Code”:

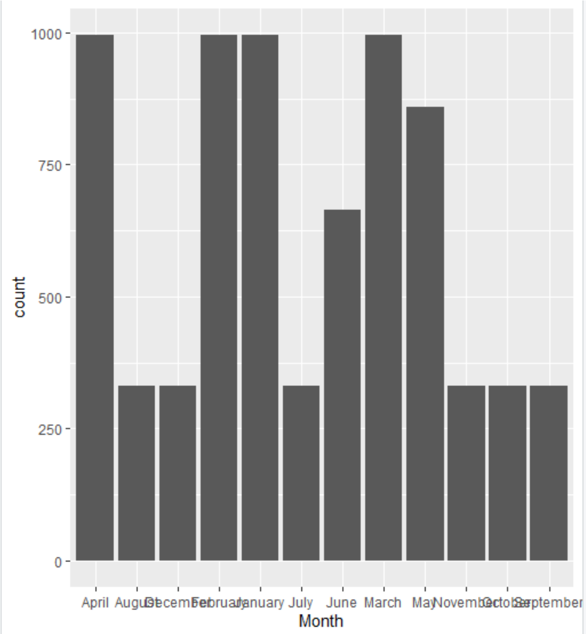


Scatter Matrix to show the correlation between Age.Group, Unit, Value and Vaccination_Code:



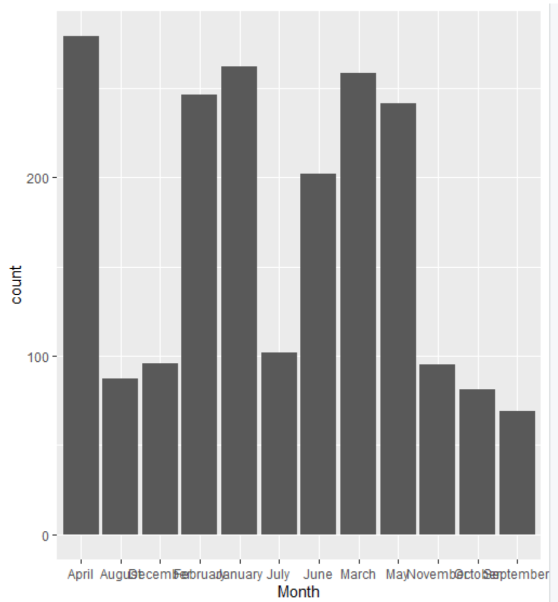
e) Graphics and descriptive understanding should be provided along with Data Exploratory analysis (EDA). Identify subgroups of features that can explore some interesting facts.

Vaccinations per month:



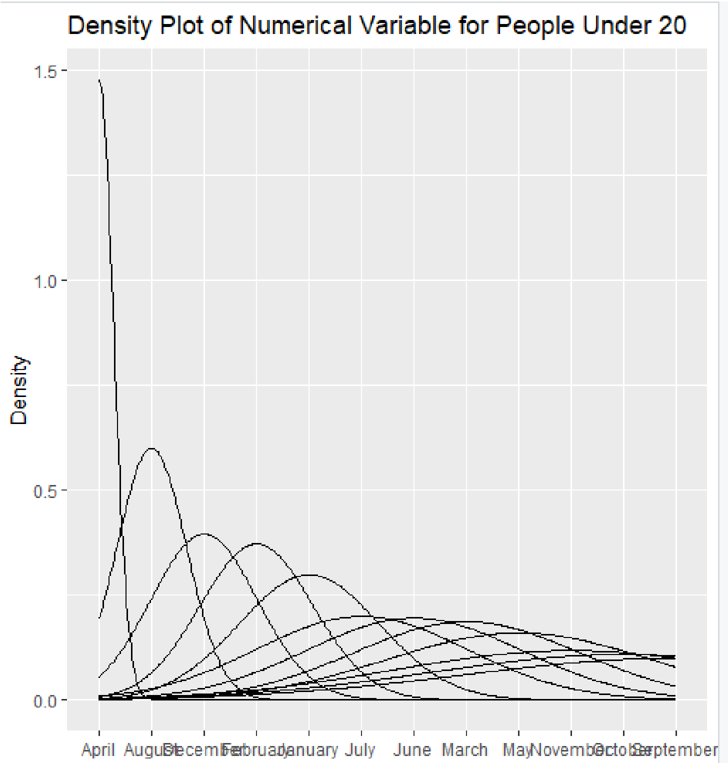
After analysing the graph, we can observe that the months of January, February, April and March were the ones with the most people being vaccinated. After a quickly research to see what happened in those months, I could see that the vaccination in Ireland start at late December, the 29th. With this info we can conclude that because those were the first months of the vaccination, everyone tried to get it as soon as possible to be protected against Covid-19.

Vaccinations over 35:



By analysing the graph, we can observe that April was the month with most vaccines given during 2021. After quick research, I could see that Ireland had a plan to reopen the country starting in April 2021 which might be the reason why people got the vaccine.

Vaccinations Under 20:



By analysing the density plot we can conclude that August has the highest value and February has the second. After some research I could see that the reason behind it was, In February vaccine was released for babies and kids and that in August there were no priority in taking the vaccine.

f) Apply dummy encoding to categorical variables (at least one variable used from the data set and discuss the benefits of dummy encoding to understand the categorical data.

Dummy encoding is the process of taking a categorical variable and transforming the chr value into a binary, where 1 means the existence of the value and 0 means the nonexistence of the value, that is the most common one but also there are dummy encoding practices where the column has a lot a different values, such as age and we can assign it to a key value instead, such as “Age5”, “Age10” and so on, where we splitting the ages in gaps of 5 years between them. In our case we are assigning 1 where the “Statistic.Label” has “Fully Vaccinated” value and 0 where it says “Partially Vaccinated”. To do it, there are different ways and formats of Dummy Encoding, in my case I have decided to use a simple ifelse in a new column in our dataset and assign the binary values 0 or 1 depending on the value in our row, as you can see below.

```r
# Assign "Fully Vaccinated" as 1 and not Fully Vaccinated 2, to make it easier to manipulate the data
data$Vaccination_Code <- ifelse(data$Statistic.Label == "Fully Vaccinated", 1, 2)
```

```
> print(data$Vaccination_Code)
   [1] 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
  [51] 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
 [101] 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
 [151] 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
 [201] 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
 [251] 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
 [301] 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
 [351] 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
 [401] 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
 [451] 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
 [501] 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
 [551] 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
 [601] 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
 [651] 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
 [701] 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
 [751] 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
 [801] 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
 [851] 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
 [901] 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
 [951] 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
 [ reached getOption("max.print") -- omitted 6500 entries ]
> tail(data$Vaccination_Code)
[1] 1 1 1 1 1 1
```

Benefits: The benefits of putting in practice Dummy Encoding are, if you have one very long sentence or value that can be easily reduced into 0 or 1 to make it readable and easier to understand making it very useful for both human and machine learning understanding and it will improve the model performance since binary numbers is the main language that machine uses and need so if you are reducing the value in your categorical data into 0's and 1's will improve the understanding and readability of the data from the computer side making it faster and with a better accuracy.

g) Apply PCA with your chosen number of components. Write up a short profile of the first few components extracted based on your understanding.

PCA, Principal Component Analysis, is the process of taking a column with many different values and transform it into a smaller set of values making it easier to understand and read the data and to do that there are some steps involved. Below I will show how I did it:

First, with "numeric_columns" I am assigning the columns that I will use, Age and Vaccination Code. After, scale the data chosen using "scale" function.

```r
# Scale the data
numeric_columns <- data[, c(8, 11)]
scaled_data <- scale(numeric_columns)
print(scaled_data)
```

```
180:35  (Top Level) ÷

Console   Terminal ×   Background Jobs ×

R 4.3.1 · ~/
493  -0.981600396        -0.5049615
494   0.737690349        -0.5049615
495  -0.695051939        -0.5049615
496   0.909619424        -0.5049615
497  -0.866981013        -0.5049615
498   1.024238807        -0.5049615
499  -0.695051939        -0.5049615
500   0.852309732        -0.5049615
 [ reached getOption("max.print") -- omitted 7000 rows ]
attr(,"scaled:center")
       Age.Group Vaccination_Code
         22.1280           0.2032
attr(,"scaled:scale")
       Age.Group Vaccination_Code
      17.4490557        0.4024069
```

After, “pca_result” is used to perform the PCA, where “scaled_data” is the data scaled in the step before, “center=true” means that the data must be centralized and “scale=true” says if the data should be scaled.

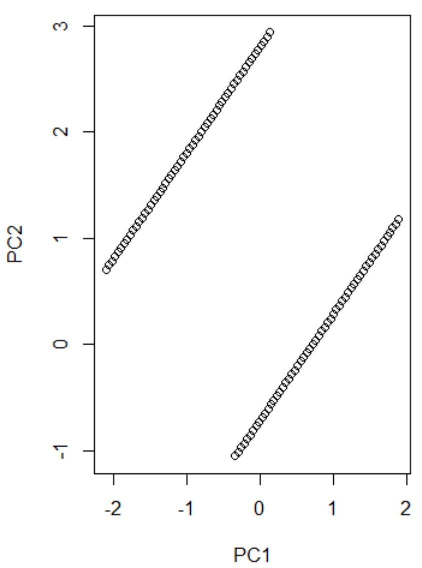
```
> pca_result <- prcomp(scaled_data, center = TRUE, scale. = TRUE)
> print(pca_result)
Standard deviations (1, .., p=2):
[1] 1.0070030 0.9929476

Rotation (n x k) = (2 x 2):
                        PC1       PC2
Age.Group         0.7071068 0.7071068
Vaccination_Code -0.7071068 0.7071068
> 
```

Now we use summary function to check the results.

```
summary(pca_result)

184:18   (Top Level) ÷

Console   Terminal ×   Background Jobs ×
R 4.3.1 · ~/
> summary(pca_result)
Importance of components:
                         PC1    PC2
Standard deviation     1.007 0.9929
Proportion of Variance 0.507 0.4930
Cumulative Proportion  0.507 1.0000
> 
```

By checking the “Cumulative Proportion” we can conclude that PC1 explain 50% of the data and PC2 explains 100% of the data.

Now we use plot to explore the main components.



h) What is the purpose of dimensionality reduction? Explore the situations where you can gain the benefit of dimensionality reduction for data analysis.

Dimensionality reduction is the process of checking a dataset and getting the conclusion that the features or values are of very high dimensionality, a column where the value inside is too much for a human to read so it means that it will take more time to process per example and reduce it into a smaller dimension of set, like has been done on the previous tasks in this assignment with the PCA. There are two ways to do the dimensionality reduction: Features selection is where subsets that are very important and relevant to the data are selected, the main goal is to reduce the number of columns while any relevant data is left out and Features extraction where data is transformed or combined thus creating a new column, the main goal is to capture the essence of the data in a lower-dimensional space.

One of the main benefits when using dimensionality reduction is, just like I mentioned before, a human would take more time reading and processing the value thus increasing the thinking time, the same would happen with machine learning, it would increase the time taken for the machine to process all the value so by reducing the dimensions, we are improving the readability and performance of the machine learning.

Conclusion

After the end of the research and the assignment, I could see and do in practice from the basics steps of how to prepare and clean the data do start manipulating to learn more about it to steps more complicated but that will give a better and a more accurate results for the research and understanding, such as the Principal Component Analysis. With this project I could see and learn more about datasets, how to clean and prepare it to use, by handling duplicates and dropping missing values per example, how to create graphs to have a better understanding of the data and how transforming categorical data into binary code value will have to improve the performance of the machine learning.

Word count: 1,515

References

Johnson, R. (2021) *Discrete, continuous & categorical variables definition*, *Discrete, Continuous & Categorical Variables Definition*. Available at: https://study.com/academy/lesson/continuous-discrete-variables-definition-examples.html (Accessed: 03 December 2023).

How to Normalize Data in R for my Data: Methods and Examples (2020) *RPubs*. Available at: https://rpubs.com/zubairishaq9/how-to-normalize-data-r-my-data#:~:text=%2DZ%2Dscore%20normalization%20transforms%20each,range%20(maximum%2Dminimum) (Accessed: 03 December 2023).

Balde, B. (2023) *Visualizing correlations: Scatter matrix and heat map*, *Medium*. Available at: https://medium.com/@becaye-balde/visualizing-correlations-scatter-matrix-and-heat-map-d597436b7d23 (Accessed: 03 December 2023).

Zhu, Y.F. and J. (2022) *R programming: Zero to pro*, *7.2 Separate and Combine Columns via separate() and unite()*. Available at: https://r02pro.github.io/separate-unite-columns.html (Accessed: 03 December 2023).

Data visualization with GGPLOT2 (2023) *Data Analysis and Visualisation in R for Ecologists: Data visualization with ggplot2*. Available at: https://datacarpentry.org/R-ecology-lesson/04-visualization-ggplot2.html (Accessed: 03 December 2023).

Jadi, Z. (2023) *A step-by-step explanation of principal component analysis (PCA)*, *Built In*. Available at: https://builtin.com/data-science/step-step-explanation-principal-component-analysis (Accessed: 03 December 2023).

Anushruthika (2023) *From raw to rescaled: A guide to Z-score, normalization, and standardization in data preprocessing*, *Medium*. Available at: https://medium.com/@anushruthikae/from-raw-to-rescaled-a-guide-to-z-score-normalization-and-standardization-in-data-preprocessing-173874df077d#:~:text=This%20difference%20highlights%20the%20robustness,range%20of%200%20to%200.25. (Accessed: 03 December 2023).

Uberoi, A. (2023) *Introduction to dimensionality reduction*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/dimensionality-reduction/ (Accessed: 03 December 2023).