Section A

Part 1 Exploratory Data Analysis (EDA)

Undertake an exploratory analysis of the variables in the survey dataset. Illustrate your analysis with suitable plots. Communicate what you have learnt from the EDA about the data to be analysed

1.1 Introduction

Within part 1 of this report survey data relating to different aspects of subjective wellbeing and demographic questions will be analysed. The data used within this section can be found below in **Table 1**.

Table 1. Description of Variables Used Within Study

Variable	Description	Coding
Case	Case number	-
MCZ_1	Overall, how satisfied are you with your life nowadays?	Likert scale (1-10)
MCZ_2	Overall, to what extent do you feel things you do in your life are worthwhile?	Likert scale (1-10)
MCZ_3	Overall, how happy did you feel yesterday?	Likert scale (1-10)
MCZ_4	Overall, how anxious did you feel yesterday	Likert scale (1-10)
Qhealthr8	How is your health in general	(1 = very good, 2 = good, 3 = fair, 4 = bad, 5 = very bad)
rsex	Sex of respondent	(1 = male, 2 = female)
agex	Grouped age	(1 = 16 to 24, 2 = 25 to 44, 3 = 45 to 54, 4 = 55 to 64, 5 = 65 to 74, 6 = 75 and over)
Marstat3r9	Marital status	(1 = married/cohabiting, 2 = single, 3 = widowed/divorced/separated)
Highed4	Highest level of qualification	(1 = degree of equivalent, 2 = below degree level, 3 = other, 4 = no qualifications)
ethnicity	Ethnicity	(1 = white, 2 = other)
Dvilo3a	Employment status according to international Labour Organisation (ILO) definition	(1 = employed, 2 = unemployed, 3 = economically inactive)
ftptwk	Full-time or part-time work	(1 = full-time, 2 = part-time)
Nsecac3	NS-SEC	(1 = managerial and professional occupations, 2 = intermediate occupations, 3 = routine and manual occupations, 4 = not classified)
gora	Government Office Region	(1 = North East, 2 = North West, 3 = Yorkshire and the Humber, 4 = East Midlands, 5 = West Midlands, 6 = Eastern, 7 = London, 8 = South East, 9 = South West, 10 = Wales, 11 = Scotland)

1.2 Data Preparation

```
# loading required libraries
library(tidyverse)
library(moments)
library(patchwork)
```

```
library(corrplot)
library(PerformanceAnalytics)
# Importing Survey
survey_data <- read.delim("survey.txt", header = TRUE)
```

1.2.1 Missing Values

Before the EDA was carried out the data was prepared. This first included the identification and removal of missing values. The data was found to contain 215 missing values.

```
# Checking for missing values
is.na(survey_data)
sum(is.na(survey_data))
# Removing rows with missing
survey_data <- drop_na(survey_data)
```

1.2.2 Removal of -2 and -1

Some of the variables were identified as containing observations coded as '-1 = Do Not Know' and '-2 = Refused to Answer' this coding may cause errors in the analyses. These problem observations were identified and then removed from the data.

```
# checking which columns have
# redundant -2 or -1 coding
# Looking at min value
summary(survey_data)

# Removal of -1 and -2 observations
# MCZ_1
survey_data <- subset(survey_data, MCZ_1!=-1 & MCZ_1!=-2)

# MCZ_2
survey_data <- subset(survey_data, MCZ_2!=-1 & MCZ_2!=-2)

# MCZ_3
survey_data <- subset(survey_data, MCZ_3!=-1 & MCZ_3!=-2)

# MCZ_4
survey_data <- subset(survey_data, MCZ_4!=-1 & MCZ_4!=-2)
```

1.3 EDA

1.3.1 Numerical Summaries and Visualisations

Within this study the variables MCZ 1 - 4 will be treated as numeric (not ordinal). These summary statistics can be found below in **Table 2**, the variables MCZ 1-3 all share similar distributions with means between 7.42-7.71. This can be better observed in **Figure 1** and **Figure 2**, displaying the histogram and overlaid density plot.

In the context of the data these variables relate to questions regarding current happiness and satisfaction, this may indicate the surveyed popular are quite satisfied and happy with their lives. This

is further supported by the mean for MCZ 4, which relates to how anxious the participant felt the day before. With a relatively low mean of 3.43 indicating low levels of anxiety on average.

```
# Creating function to calculate stats
summary_stats = function(x){
  summary = summary(x)
  std = c('STD',sd(x))
  var = c('VAR',var(x))
  skew = c('SKEW',skewness(x))
  kurt = c('KURT',kurtosis(x))
  return(c(summary, std, var, skew, kurt))
}

# calculating summary stats for MCZ variables
# MCZ_1
summary_stats(survey_data$MCZ_1)
# MCZ_2
summary_stats(survey_data$MCZ_2)
# MCZ_3
summary_stats(survey_data$MCZ_3)
# MCZ_4
summary_stats(survey_data$MCZ_4)

# Distribution visualisations
# Numerical
# MCZ_1
mcz_1 <- ggplot(survey_data, aes(x = MCZ_1)) +
  geom_histogram()

# MCZ_2
mcz_2 <- ggplot(survey_data, aes(x = MCZ_2)) +
  geom_histogram()

# MCZ_3
mcz_3 <- ggplot(survey_data, aes(x = MCZ_3)) +
  geom_histogram()

# MCZ_4
mcz_4 <- ggplot(survey_data, aes(x = MCZ_4)) +
  geom_histogram()

mcz_1 + mcz_2 + mcz_3 + mcz_4


# density plot
density <- ggplot()
density <- density + geom_density(data=survey_data, aes(MCZ_1,
colour = "MCZ_1"))
density <- density + geom_density(data=survey_data, aes(MCZ_2,
colour = "MCZ_2"),  size=1.2)
density <- density + geom_density(data=survey_data, aes(MCZ_3,
colour = "MCZ_3"),  size = 1)
```

```
density <- density + geom_density(data=survey_data, aes(MCZ_4,
colour = "MCZ_4"))

density + scale_color_brewer(palette = "Set1") + theme_minimal() +
  theme(legend.spacing.y = unit(4.0, 'cm'),
        legend.text = element_text(size=10),
        legend.key.size = unit(3,"line")) +
  xlab("MCZ Score") +
  guides(colour=guide_legend(title="MCZ Variable", title.position =
"left"))
```

Table 2. Summary Statistics for the MCZ Variables (1-4)

Variable	Mean	Median	1st quartile	3rd quartile	Min	Max	STD	VAR	Skew	Kurtosis
MCZ 1	7.42	8	7	9	0	10	1.83	3.33	-0.92	1.02
MCZ 2	7.71	8	7	9	0	10	1.73	2.99	-1.02	1.53
MCZ 3	7.55	8	7	9	0	10	2.16	4.67	-1.17	1.25
MCZ 4	3.43	3	1	6	0	10	3.03	9.20	0.50	-0.97

Figure 1. Histograms for the MCZ Variables (MCZ 1-4)

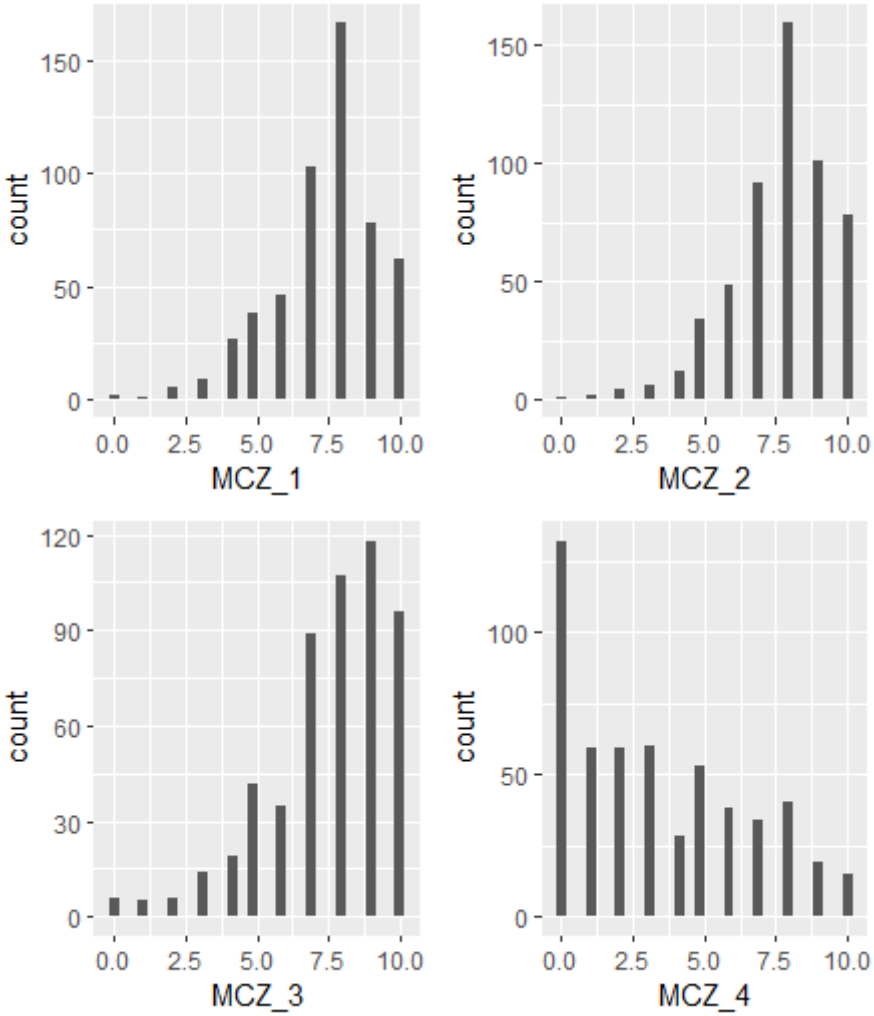
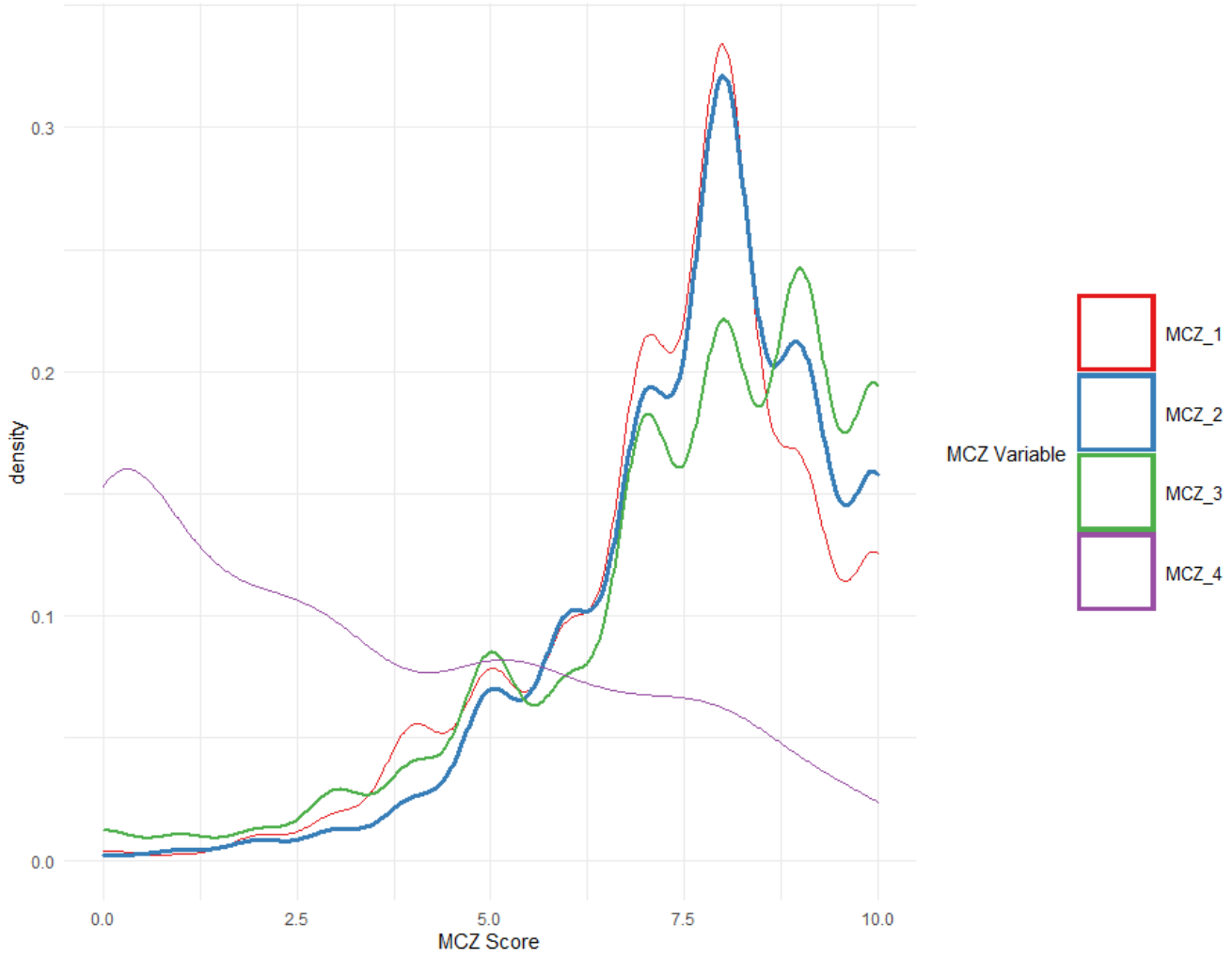


Figure 2. Overlaid Density Plots for the MCZ Variables (MCZ 1- 4)



1.3.2 Categorical Summary and Visualisation

For the remaining categorical variables, the summaries are presented in the form of the counts and proportions per category observed. These can be found below in **Table's 3-4**. A visualisation of the distribution per category can be found below in **Figure 3**.

From these summaries some conclusions regarding the surveyed population can be made:

- The proportion of male and female is almost equal
- 89% of those surveyed rated their health in general as either very good or good
- The largest proportion of participants were in the 25 to 44 age group
- 89% of the participants were white

```
# for categorical (frequencies)
# Counts
for(i in 6:15){
  print(table(survey_data[i]))
}

# proportions
for(i in 6:15){
  print(signif((prop.table(table(survey_data[i]))), digits = 2) *
100)
}
survey_data$QHealthr <- as.factor(survey_data$QHealthr)
survey_data$RSEX <- as.factor(survey_data$RSEX)
survey_data$AGEX <- as.factor(survey_data$AGEX)
survey_data$marstat3r <- as.factor(survey_data$marstat3r)
survey_data$highed4 <- as.factor(survey_data$highed4)
survey_data$Ethnicity <- as.factor(survey_data$Ethnicity)
survey_data$DVILO3a <- as.factor(survey_data$DVILO3a)
survey_data$FtPtWk <- as.factor(survey_data$FtPtWk)
survey_data$NSECAC3 <- as.factor(survey_data$NSECAC3)
survey_data$GorA <- as.factor(survey_data$GorA)
# Categorical Visualizations
# Bar Charts
# qhealthr8
qhealthr8 <- ggplot(survey_data, aes(x=QHealthr)) +
  geom_bar()
# rsex
rsex <- ggplot(survey_data, aes(x = RSEX)) +
  geom_bar()
# agex
agex <- ggplot(survey_data, aes(x = AGEX)) +
  geom_bar()
# marstat3r9
marstat3r9 <- ggplot(survey_data, aes(x = marstat3r)) +
  geom_bar()
# highed4
highed4 <- ggplot(survey_data, aes(x = highed4)) +
  geom_bar()
# ethnicity
ethnicity <- ggplot(survey_data, aes(x = Ethnicity)) +
```

```
  geom_bar()
# dvilo3a
dvilo3a <- ggplot(survey_data, aes(x = DVILO3a)) +
  geom_bar()
# ftptwk
ftptwk <- ggplot(survey_data, aes(x = FtPtWk)) +
  geom_bar()

# nsecac3
nsecac3 <- ggplot(survey_data, aes(x = NSECAC3)) +
  geom_bar()

# gora
gora <- ggplot(survey_data, aes(x = GorA)) +
  geom_bar()

qhealthr8 + rsex + agex + marstat3r9 +
  highed4 + ethnicity + dvilo3a +
  ftptwk + nsecac3 + gora
```

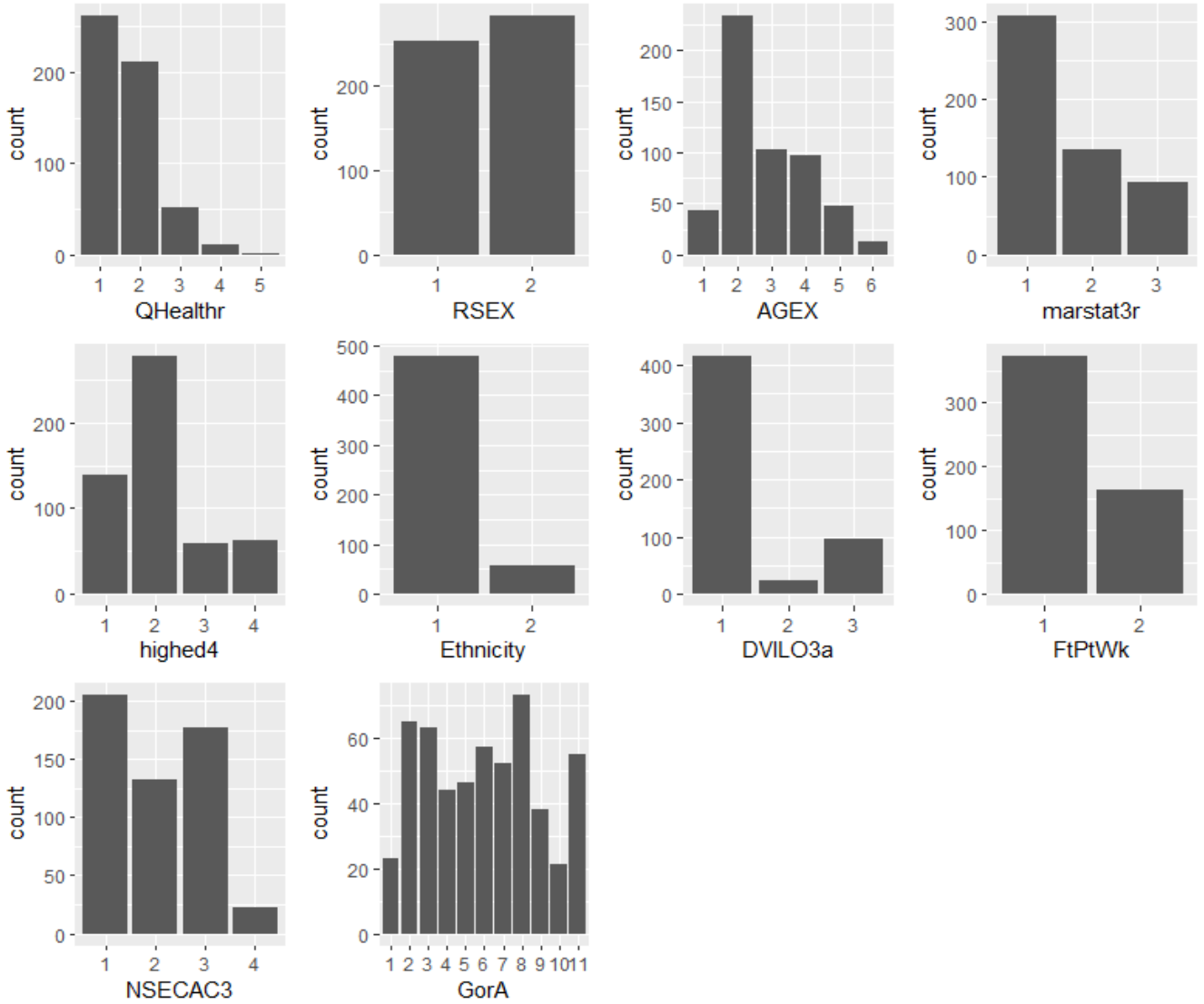
Table 3. Categorical Summary (Counts)

	Category										
Variable	1	2	3	4	5	6	7	8	9	10	11
qhealthr8	262	213	51	11	1	-	-	-	-	-	-
rsex	255	283	-	-	-	-	-	-	-	-	-
agex	43	234	103	98	48	12	-	-	-	-	-
marstat3r9	308	136	94	-	-	-	-	-	-		-
highed4	138	278	60	62	-	-	-	-	-	-	-
ethnicity	481	57	-	-	-	-	-	-	-	-	-
dvilo3a	418	24	96	-	-	-	-	-	-	-	-
ftptwk	375	163	-	-	-	-	-	-	-	-	-
nsecac3	206	132	177	23	-	-	-	-	-	-	-
gora	23	65	63	44	46	58	52	73	38	21	55

Table 4 Categorical Summary (Proportions)

	Category (%)										
Variable	1	2	3	4	5	6	7	8	9	10	11
qhealthr8	49	40	9.50	2	0.19	-	-	-	-	-	-
rsex	47	53	-	-	-	-	-	-	-	-	-
agex	8	43	19	18	8.9	2.2	-	-	-	-	-
marstat3r9	57	25	17	-	-	-	-	-	-		-
Highed4	26	52	11	12	-	-	-	-	-	-	-
ethnicity	89	11	-	-	-	-	-	-	-	-	-
dvilo3a	78	4.5	18	-	-	-	-	-	-	-	-
ftptwk	70	30	-	-	-	-	-	-	-	-	-
nsecac3	38	25	33	4.3	-	-	-	-	-	-	-
gora	4.3	12	12	8.2	8.6	11	9.7	14	7.1	3.9	10

Figure 3. Numerical Variable Visualisation of Counts



1.3.2 Exploration of MCZ Scores and Ethnicity and Highest Level of Qualification

Through conducting the EDA an interesting pattern was identified between the respondent's ethnicity, highest level of qualification, and the MCZ scores. To explore this the means for each MCZ score for each level of qualification and the two ethnicities "White" and "Other" were calculated. **Table 5** and **Figure 4** display these patterns. First, the ethnicities coded as "other" show lower mean scores for MCZ 1-3 and a higher mean score for MCZ 4 for almost all qualification levels. This may suggest that "other" ethnicities have lower life satisfaction and higher anxiety compared to those of white ethnicity.

Secondly, it can be seen the greatest difference in means in general is seen in the "no qualification" category between white and other ethnicities for MCZ 1- 3. With white ethnicities score on average 2 whole points higher. This suggests that when both ethnicities have no qualifications, those who are white report significantly higher happiness and highlights an area for further exploration.

Although it should be noted that the "other" ethnicities have significantly less representation in this study, making up just 11% of the population.

```
# MCZ and qualification and ethnicity
plotdata.EQ <- survey_data %>%
  group_by(highed4, Ethnicity) %>%
  summarise(mean_mcz1 = round(mean(MCZ_1),2),
            mean_mcz2 = round(mean(MCZ_2),2),
            mean_mcz3 = round(mean(MCZ_3),2),
            mean_mcz4 = round(mean(MCZ_4),2))
print.data.frame(plotdata.EQ)

# visualising

p1 <- ggplot(plotdata.EQ, aes(x=factor(highed4,
                                       labels = c("degree",
                                                  "below degree",
"other", "no qualification")),
                              y=mean_mcz1, fill=Ethnicity)) +
  geom_bar(stat="identity", position = position_dodge(width = 0.5),
width=0.5) +
  geom_text(aes(label=round(mean_mcz1, 2)),
            vjust=-0.25,
            position = position_dodge(width=0.9)) +
  scale_y_continuous(breaks = 0) +
  labs(title = "MCZ 1",
       x="",
       y="") +
  scale_fill_discrete(name = "Ethnicity", labels = c("White",
"Other"))


p2 <- ggplot(plotdata.EQ, aes(x=factor(highed4,
                                       labels = c("degree",
                                                  "below degree",
"other", "no qualification")),
                              y=mean_mcz2, fill = Ethnicity)) +
  geom_bar(stat="identity", position = position_dodge(width = 0.5),
```

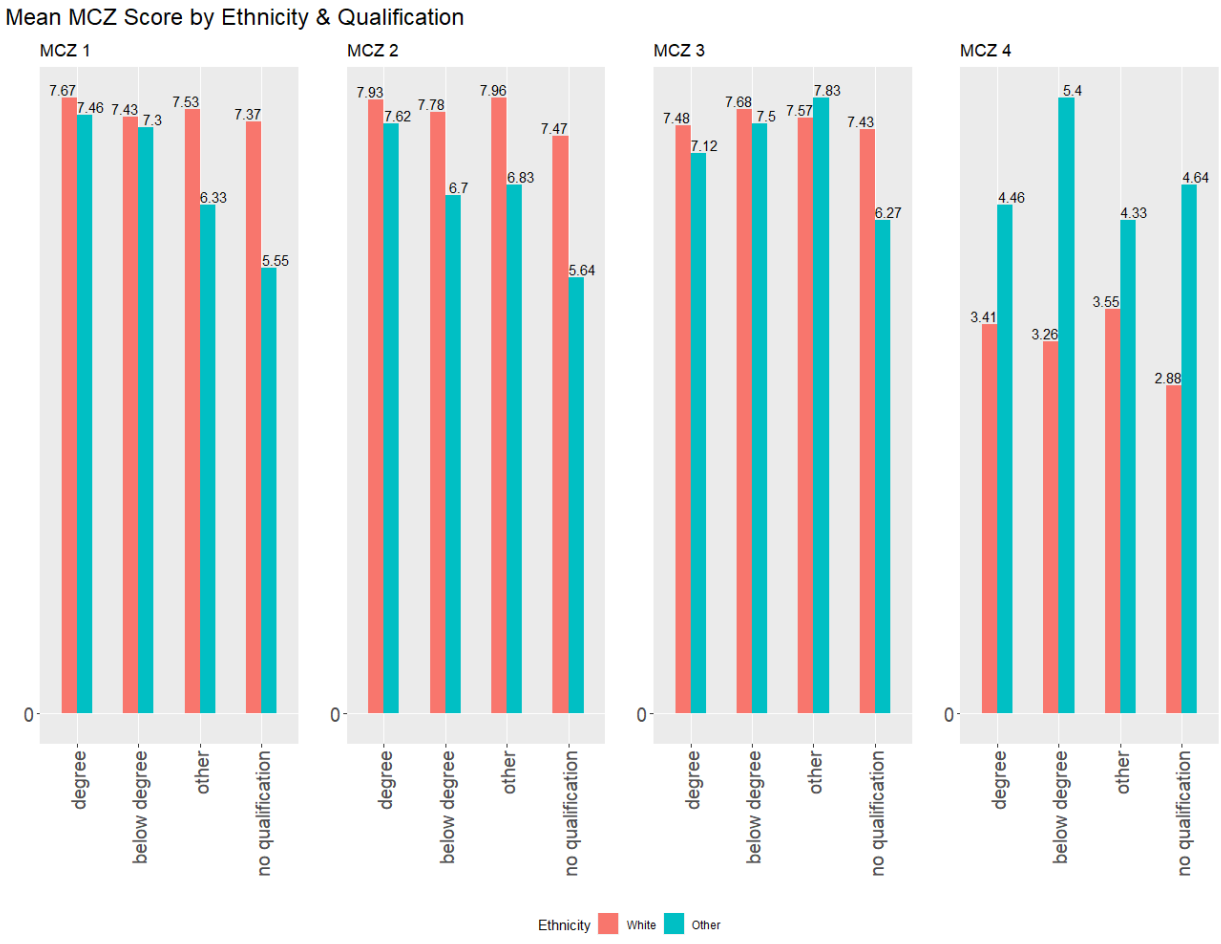
```
width=0.5) +
  geom_text(aes(label=round(mean_mcz2, 2)),
            vjust=-0.25,
            position = position_dodge(width=0.9)) +
  scale_y_continuous(breaks = 0) +
  labs(title = "MCZ 2",
       x="",
       y="") +
  scale_fill_discrete(name = "Ethnicity", labels = c("White",
"Other"))

p3 <- ggplot(plotdata.EQ, aes(x=factor(highed4,
                                       labels = c("degree",
                                                  "below degree",
"other", "no qualification")),
                              y=mean_mcz3, fill = Ethnicity)) +
  geom_bar(stat="identity", position = position_dodge(width = 0.5),
width=0.5) +
  geom_text(aes(label=round(mean_mcz3, 2)),
            vjust=-0.25,
            position = position_dodge(width=0.9)) +
  scale_y_continuous(breaks = 0) +
  labs(title = "MCZ 3",
       x="",
       y="") +
  scale_fill_discrete(name = "Ethnicity", labels = c("White",
"Other"))

p4 <- ggplot(plotdata.EQ, aes(x=factor(highed4,
                                         labels = c("degree",
                                                    "below degree",
"other", "no qualification")),
                                y=mean_mcz4, fill = Ethnicity)) +
  geom_bar(stat="identity", position = position_dodge(width = 0.5),
width=0.5) +
  geom_text(aes(label=round(mean_mcz4, 2)),
            vjust=-0.25,
            position = position_dodge(width=0.9)) +
  scale_y_continuous(breaks = 0) +
  labs(title = "MCZ 4",
       x="",
       y="") +
  scale_fill_discrete(name = "Ethnicity", labels = c("White",
"Other"))


patchwork <- p1  + p2 + p3 + p4 +
  plot_layout(ncol = 4, guides = 'collect') & theme(legend.position
= "bottom") +
  theme(axis.text = element_text(size=15),
        axis.text.x = element_text(angle = 90, vjust = 0.5, hjust =
```

```
1))

patchwork + plot_annotation(
  title = "Mean MCZ Score by Ethnicity & Qualification",
  theme = theme(plot.title = element_text(size=18))
)
```

Table 5. MCZ 1-4 Mean Scores for Each Level of Qualification and Both Ethnicities

		Mean			
Highest level of qualification	Ethnicity	MCZ 1	MCZ 2	MCZ 3	MCZ 4
Degree	White	7.67	7.93	7.48	3.41
	Other	7.46	7.62	7.12	4.46
Below Degree	White	7.43	7.78	7.68	3.26
	Other	7.30	6.70	7.50	5.40
Other	White	7.53	7.96	7.57	3.55
	Other	6.33	6.83	7.83	4.33
No Qualification	White	7.37	7.47	7.43	2.88
	Other	5.55	5.64	6.27	4.64

Figure 4. Bar Plot of MCZ 1-4 Mean Scores for Both Ethnicities



Part 2 Linear regression

Use the variables from qhealthr8 to gora as features (potential predictor variables) in a model for mcz_1 using a suitable supervised learning approach. Describe what your model shows about the influence of the different predictors included within it on the outcome variable

2.1 Introduction

As the MCZ variables are treated as numeric the supervised learning approach chosen was linear regression. The aim of this section is to fit a linear model using MCZ 1, a measure of life satisfaction, as the dependent variable and the variables from qhealthr8 to gora as the potential predictor variables.

2.2 Feature Selection Via Best Subset Selection

```
# loading libraries
library(olsrr)
library(tidyverse)
library(caret)
library(car)
```

The variables MCZ 2-4 were removed from the data frame as these variables will not be used within this analysis.

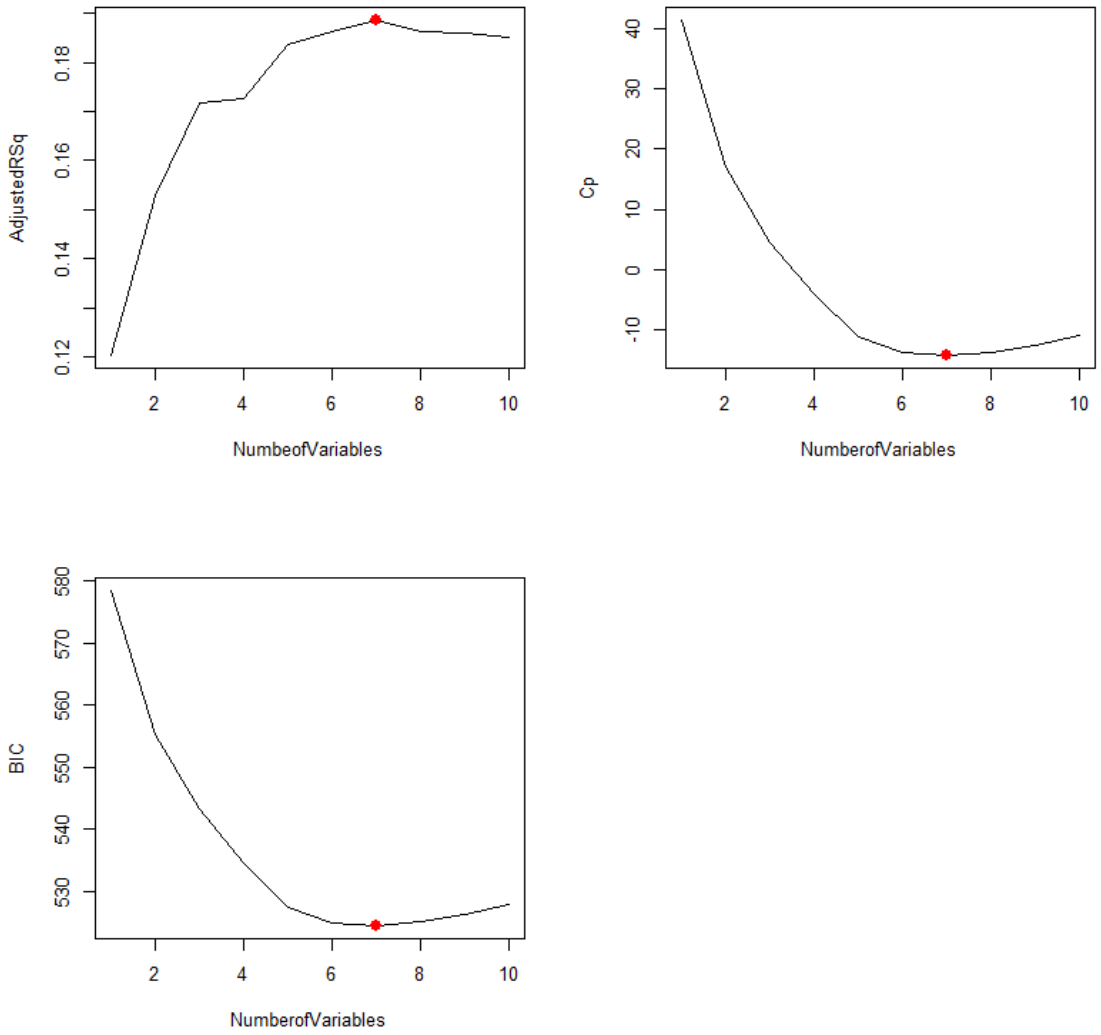
```
# dropping mcz2-4 (3-5)
survey_df <- survey_data[-c(1,3:5)]
str(survey_df)
```

Best subset was used as a method of feature selection. However, as regsubset() from the leaps library does not work on categorical data for feature selection, in this analysis ols_step_best_subset() from the olsrr library will be used. From the outputs of best subset selection, the Adjusted R-squared, Mallows Cp (Cp), and Bayesian Information Criterion (BIC) were plotted. These are metrics used in model selection, the aim is to maximise the Adjusted R-squared and minimise the prediction error (Cp and BIC). These can be found below in **Figure 5**. Best subset selection identified the 7-variable model as the potential best subset.

```
# fitting model with all predictors
model <- lm(MCZ_1 ~., data=survey_df)
# using ols_step_best_subset() to conduct best subset
subset <- ols_step_best_subset(model)

# plotting the best subset outputs
par(mfrow=c(2,2))

# adjusted R2
plot(subset$adjr, xlab = "NumbeofVariable", ylab = "AdjustedRSq",
     type = "l")
which.max(subset$adjr)
#7
points(7, subset$adjr[7], col="red", cex=2, pch=20)
```

```r
#cp
plot(subset$cp, xlab = "NumberofVariables", ylab = "Cp",
     type = "l")
which.min(subset$cp)
# 7
points(7, subset$cp[7], col="red", cex = 2, pch=20)

# BIC
plot(subset$sbic, xlab = "NumberofVariables", ylab = "BIC",
     type = "l")
which.min(subset$sbic)
#7
points(7, subset$sbic[7], col="red", cex=2, pch=20)
```

Figure 5. Best Subset Selection Output



2.3 Model Selection Via k-fold cross-validation

Best subset selection offers a viable method for feature selection, however, the metrics for model selections (R2, BIC Cp) are calculated on the training that data, which has been used to fit the model and may lead to overfitting.

k-fold cross-validation was used on each of the models created by the best subset selection. The k-fold cross-validation measures the model's ability to predict 'out of sample' by partitioning the data into training and test data. This method produces an Averaged Root Mean Squared Error (RMSE) for each model which describes the prediction error within each model, the lower the RMSE the better the model. The output of this can be found below in **Table 6**. This method suggests that the model containing 3 variables is the most optimal model. This model was chosen despite best subset selections output, as it is clear the addition of extra variables increases the RMSE.

The variables of the 3 variable model were QHealthR, AGE, and DVILo3a.

```
# using k-fold cross-validation
# helper function 1
get_model_formula <- function(id, object, outcome) {
  predictors <- object$predictors[id]
  predictors <- gsub(" ", "+", predictors)
  as.formula(paste0(outcome, "~", predictors))
}

# Helper function 2
get_cv_error <- function(model.formula, data){
  set.seed(1)
  train.control <- trainControl(method = "repeatedcv", number = 10,
repeats = 5)
  cv <- train(model.formula, data=data, method="lm",
              trControl=train.control)
  round(cv$results$RMSE, 2)
}

# generating errors for each best subset model
model.ids <- 1:10
cv.errors <- map(model.ids, get_model_formula, subset, "MCZ_1") %>%
  map(get_cv_error, data=survey_df) %>%
  unlist()
cv.errors

# identifying lowest errror
which.min(cv.errors)
```

Table 6. RMSE Output for The k-fold cross-validation Method

Model	RMSE
1	1.72
2	1.70
3	1.69
4	1.71
5	1.70
6	1.70
7	1.70
8	1.71
9	1.72
10	1.72

2.4 Model Interpretation

2.4.1 Model Summary

The output for this linear model can be found below in **Table 7**. These outputs suggest an estimated 17% of the variation in the dependent variable 'MCZ 1', is explained by the selected explanatory variables using adjusted R-square. Additionally, the significance of the reported F value of <2.2e-16 suggests this model provides a better fit to the data compared to the null model and at least one variable is a significant explainer.

```
# fitting the model
final.model <- lm(MCZ_1 ~ QHealthr + AGEX + DVILO3a, data=survey_df)

# analysing model
summary(final.model)
```

Table 7. Linear Regression Model Summary Output

Variable	Unstandardized coefficient	Significance
Constant (intercept)	7.74	<2e-16
QHealthr2	-0.48	0.002
QHealthr3	-1.20	7.05e-06
QHealthr4	-3.95	5.73e-13
QHealthr5	-3.67	0.030
AGEX2	-0.0030	0.99
AGEX3	-0.24	0.44
AGEX4	0.47	0.12
AGEX5	0.71	0.053
AGEX6	1.67	0.0030
DVILo3a2	-1.32	0.00020
DVILo3a3	-0.070	0.75
R-Square	Adjusted R-square	Sig.
0.19	0.17	<2.2e-16

2.4.2 Coefficient Interpretation

As the explanatory variables are categorical the interpretation of the coefficients must be made in reference to the 'reference' category of each variable. This is the category that is not included in the model.

QHealthr – How is your health in general?

Reference category: '1 = very good'

Interpretation of the QHealthr variable suggest, controlling for the other variables in the model respectively, respondents who scored their health in general as '2', '3', '4', and '5'all have respectively 0.48, 1.20, 3.95, and 3.67 decrease in expected MCZ 1 score compared to respondents who scored their health as '1'. This may suggest that as general health decreases (i.e. a number closer to 5), respondents satisfaction with life also decreases (i.e. an MCZ 1 number closer to 1). All observed differences were significant.

AGEX – Grouped Age

Reference category: '1 = 16 to 24'

Interpretation of the AGEX variable suggest, controlling for the other variables in the model respectively, respondents in the '2' and '3' age group have respectively 0.0030 and 0.24 decrease in expected MCZ 1 score compared to respondents in the '1' age group. This suggests respondents aged 25-44 and 45-54 have an expected life satisfaction score lower than respondents aged 16-24. However, these observed differences were not significant.

Whereas respondents in the '4', '5', and '6' age group have respectively 0.47, 0.71, and 1.67 increase in expected MCZ 1 score compared to respondents in the '1' age group. This suggest respondents aged 55-64, 65-76, and 75+ have an expected life satisfaction score higher than respondents aged 16-24. However, only the different between age group 6 and 1 were significant.

Dvilo3a – Employment status according to international labour organisation

Reference category: '1 = employed'

Interpretation of the Dvilo3a variable suggest, controlling for the other variables in the model respectively, respondents in the '2' and '3' employment status group both have respectively 1.32 and 0.070 decrease in expected MCZ 1 score compared to respondents in the '1' employment status group. This suggests respondents who work full time have an expected life satisfaction score higher than both respondents who are unemployed and respondents who are economically inactive. However only the observed difference between group '2', unemployed, and group '1' was significant.

Part 2 Prediction

Use your model from part 2 to predict mcz_1 for the cases in eval.data. Use this information to assess the quality of your model

3.1 Introduction

Within this section the model produced in part 2 will be used to predict values for MCZ 1 using a new dataset of unseen observations. The accuracy of these predictions will be assessed using the R2, Root Mean Square Error (RMSE), Mean Absolute Error (MAE), Mean Squared Error (MSE), and a quantification of the error rate.

3.2 Model Predictions

The new dataset first had to be processed and cleaned in order to match the original training data.

```
#loading in predict set
eval_data <- read.delim("eval.txt", header = TRUE)

# Checking for missing values
is.na(eval_data)
sum(is.na(eval_data))

# removing missing
eval_data <- drop_na(eval_data)


# removing mcz's
str(eval_data)
eval_df <- eval_data[-c(1,3:5)]
str(eval_df)

# checking which variables have -1 and -2 coding
summary(eval_df)
#Qhealth
# removing rows with this coding
remove2 <- which(eval_df$QHealthr == -2)

eval_df <- eval_df[-remove2,]
summary(eval_df)

# Recoding -1 and -2 as 11 in MCZ_1

# MCZ_1
eval_df$MCZ_1[eval_df$MCZ_1<0] <- 11
eval_df$MCZ_1

# creating factors in predict set
eval_df$QHealthr <- as.factor(eval_df$QHealthr)
eval_df$RSEX <- as.factor(eval_df$RSEX)
eval_df$AGEX <- as.factor(eval_df$AGEX)
eval_df$marstat3r <- as.factor(eval_df$marstat3r)
eval_df$highed4 <- as.factor(eval_df$highed4)
eval_df$Ethnicity <- as.factor(eval_df$Ethnicity)
```

```
eval_df$DVILO3a <- as.factor(eval_df$DVILO3a)
eval_df$FtPtWk <- as.factor(eval_df$FtPtWk)
eval_df$NSECAC3 <- as.factor(eval_df$NSECAC3)
eval_df$GorA <- as.factor(eval_df$GorA)
```

The model is then used to predict values for MCZ 1 based on the new observations for the independent variables. The accuracy results of these predictions can be found below in **Table 8**. As can be seen the RMSE, representing the average difference between the observed and predicted outcome values, is 1.64, representing an error rate of 21.8% which can be considered relatively low. The MAE, and MSE are both relatively low which further suggests the model performed well in prediction of MCZ 1. As the MAE are in the units of the outcome variable, we can conclude that the model's prediction on average is 1.25 away from the actual value.

However, the R-square (R2), representing the correlation between the observed outcome values and the predicted values of 0.14 is quite low. To further assess why this low value occurred model diagnostics should be performed on the linear model to identify assumption violations.

```
# making predictions
predicted <- predict(final.model, newdata = eval_df)

# computing accruacy metrics of the models predictions
data.frame(R2 = R2(predicted, eval_df$MCZ_1),
           RMSE = RMSE(predicted, eval_df$MCZ_1),
           MAE = MAE(predicted, eval_df$MCZ_1),
           MSE = mean((eval_df$MCZ_1-predicted)^2),
           error.rate = (RMSE(predicted, eval_df$MCZ_1) /
mean(eval_df$MCZ_1)))
```

Table 8. Model Prediction Accuracy Metrics Output

R2	RMSE	MAE	MSE	Error Rate (%)
0.14	1.64	1.25	2.70	21.8

Section B

An extract from ONS's statement of administrative sources is provided. You will need to do some tidying up to get it in a suitable form for analysis, and should describe the procedures you follow to do that. Use the prepared dataset to construct clusters, making any necessary choices about how to treat different types of variables. Explain how you decide how many clusters are most appropriate for this dataset, and comment on the characteristics of each cluster

4.1 Introduction

Within this analysis data extracted from the ONS's statement of administrative sources will be analysed using cluster analysis to construct clusters on the observations within the data. Prior to this analysis this data must be cleaned and processed appropriately in order to be suitable for analysis. In this report clustering will be conducted using hierarchical clustering methods.

4.2 Data Preparation

The data was first checked for missing values, R was unable to detect any missing values within the data.

```
# Loading the data
# loading libraries
library(tidyverse)
library(readxl)
SOAS_extract <- read_excel("SOAS extract.xlsx")
View(SOAS_extract)
# checking missing
# (No missing)
sum(is.na(SOAS_extract))
```

4.2.1 Column 1 'Name of Admin Source'

As this column is simply an identifier of the observation this column was not altered, however, it will not be used in the clustering analysis.

4.2.2 Column 2 'Theme'

The observations with more than one theme were identified and grouped into a new category "multiple theme'. These groupings can be found below in **Table 9**.

```
# Column 2 'Theme'
# changing all multiple themes to "multiple theme"
multiples <- c(26,31,35,36,71)

SOAS_extract$Theme[multiples] <- "Multiple Theme"
SOAS_extract$Theme

# checking results
table(SOAS_extract$Theme)
```

Table 9. 'Theme' Variable Groups

Theme	Count
Agriculture and Environment	1
Economy	29
Government	10
Health and Social Care	7
Labour Market	14
Multiple Theme	5
People and Places	3
Population	23
Travel and Transport	8

4.2.3 Column 3 'Format of Admin Source'

The categories can be found below in **Table 10**.

Table 10. 'Format of Admin Source' Variable Groups

Admin Source	Count
Electronic	98
Paper	2

4.2.4 Column 4 'Data Type'

This column was identified as already being appropriately categorised, consisting of five categories; "Aggregate", "Management Information", "Management Information; Microdata", "Mircodata", "Microdata and Aggregte". These categories and counts can be found below in **Table 11**.

Table 11. 'Data Type' Variable Groups

Data Type	Count
Aggregate	55
Management Information	3
Management Information ; Microdata	2
Microdata	30
Microdata & Aggregate	10

4.2.5 Column 5 'Number of Records'

First the values within this column that contained numbers were prepared in order to be detected as numerical values. They were modified as follows:

- Any value measured ‘per month’ or ‘per day’ were multiplied appropriately to obtain the yearly value
- Any numeric value containing characters, for example “per year”, the characters were removed in order to obtain the number only.

A new column is added to the data frame called ‘is.numeric’, this is used to identify whether the observations is numeric and therefore contains a number. If yes the value is 1, if no the value is 0. This output can be found below in **Table 12**. A second column is added to the data frame. This column is called ‘no.records’ and contains the number of records for the observation if it is numeric. If it is not numeric then the number of records is equal to 0. The original column ‘Number of Records’ is removed.

```
# column 5 'number of records'
# first 'cleaning' the column so numbers
# can be identified as numeric
SOAS_extract$`Number of Records`[29] <- 1000000
SOAS_extract$`Number of Records`[30] <- 1200000
SOAS_extract$`Number of Records`[31] <- (100000 * 12)
SOAS_extract$`Number of Records`[32] <- 200
SOAS_extract$`Number of Records`[33] <- 120000
SOAS_extract$`Number of Records`[36] <- 207462982
SOAS_extract$`Number of Records`[37] <- 240000
SOAS_extract$`Number of Records`[38] <- (50000 * 12)
SOAS_extract$`Number of Records`[39] <- 400
SOAS_extract$`Number of Records`[41] <- 500000
SOAS_extract$`Number of Records`[42] <- (6500 * 365)
SOAS_extract$`Number of Records`[43] <- 800000
SOAS_extract$`Number of Records`[44] <- 800
SOAS_extract$`Number of Records`[45] <- 8000

# creating a second variable
# will provide the number of records
# or 0 if the observation is non-numeric

# iniate empty vector
no.records <- c()

#use for loop to iterate
for(i in 1:100){
  if(SOAS_extract$is.numeric[i] == 1){
    no.records <- c(no.records, SOAS_extract$`Number of Records`[i])
  } else {
    no.records <- c(no.records, 0)
  }
}
# checking results
no.records
# adding this variable to the data frame
SOAS_extract <- add_column(SOAS_extract, no.records, .after =
"is.numeric")
# Removing the original column
```

```
# as this is redudant now
SOAS_extract <- SOAS_extract[-5]
```

Table 12. 'Is the value numeric' Variable Groups

Is the value numeric?	Count
Yes	41
No	59

4.2.6 Column 6 'Population Coverage'

This column was used to generate a new column called 'pop.coverage'. The names stored in 'Population Coverage' are grouped into 4 groups of populations:

- World Populations
- National Populations – This encompasses any observations population identified as being England, Wales, Scotland, or the UK.
- Smaller Populations – This encompasses smaller populations identified. For example 'Private Households', 'Asylum Seekers', and 'Working Populations'
- Other – This encompasses the remainder of the observations such as "Divorces in England and Wales" which did not appear to fit any category

The original ungrouped column is removed. The counts of 'pop.coverage' for each category can be found below in **Table 13**.

```
# Column 6 'Population Coverage'
# Create a new variable grouped based on the orginal column
# iniate a new varible with the value 4 repeated 100 times
pop.coverage <- rep(4,100)

pop.coverage

# Add this column to data frame
SOAS_extract <- add_column(SOAS_extract, pop.coverage)

# replace these values with "other" as this will be the last
category
SOAS_extract$pop.coverage[SOAS_extract$pop.coverage == 4] <- "other"

# Category 1 World population coverage
# World Variants
World <- c("Worldwide residents", "World (excluding UK)", "worldwide
residents")

for(i in World){
  for(j in 1:100){
    if(SOAS_extract$`Population Coverage`[j] == i){
      SOAS_extract$pop.coverage[j] <- "world"
    }
```

```
  }
}

SOAS_extract$pop.coverage

# Category 2 National Populations (UK national)
# national variants
national <- c("England and Wales", "UK", "E&W", "Wales", "United
Kingdom - (England, Wales, Scotland, NI)", "Scotland")

for(i in national){
  for(j in 1:100){
    if(SOAS_extract$`Population Coverage`[j] == i){
      SOAS_extract$pop.coverage[j] <- "national"
    }
  }
}

SOAS_extract$pop.coverage


# Category 3 smaller populations within national
# Smaller Populations
smaller_pop <- c("Private households", "Private household
population", "Private household ", "private household",
                 "Asylum seekers", "All employees of the Office for
National Statistics", "All employees of the Office for National
Statistics.", "All HM Forces Workers",
                 "All NHS Workers in Scotland, including secondees",
"All Police workers, including secondees",
                 "Paid workers", "Paid workers and people on
government funded training schemes without contracts of employment",
                 "Public sector workers", "Workers, including
secondees",
                 "Workers, including secondees, in Scotland")

for(i in smaller_pop){
  for(j in 1:100){
    if(SOAS_extract$`Population Coverage`[j] == i){
      SOAS_extract$pop.coverage[j] <- "smaller_pop"
    }
  }
}

SOAS_extract$pop.coverage

# finally checking counts
table(SOAS_extract$pop.coverage)

# removing old un grouped column (5)
```

```
SOAS_extract <- SOAS_extract[-5]
str(SOAS_extract)
```

Table 13 'Population Coverage' Variable Groups

Population Coverage	Count
World	6
National	21
Smaller Populations	27
Other	46

4.2.7 Column 7 'Geographic Coverage'

Similar to column 6, this column was used to generate a new column 'geo.cover' which is grouped based on the values stored in 'Geographic Coverage'. This column was categorised into three columns for geographic coverage:

- UK – This category refers to any geographic coverage that includes the entire UK only.
- National – This category refers to any geographic coverage which refers to the UK nations separately i.e. Scotland, England, etc.
- Other – this category covers any other geographic coverage not encapsulated within the other two categories.

The original column was removed. The counts for 'geo.cover' can be found below in **Table 14**.

```
# Column 7 'Geographic Coverage'
# checking current categories
table(SOAS_extract$`Geographic Coverage`)

# Category 1
# creating whole UK
whole.uk <- c("United Kingdom - (England, Wales, Scotland, NI)", "UK
& Overseas",
              "UK")

# category 2
# creating national
national.new <- c("Scotland", "Northern Ireland", "Great Britain",
                  "England and Wales", "England & Wales", "England",
"Wales")

# Category 3
# "Other"
# creating new column
geo.cover <- rep(3,100)
geo.cover
SOAS_extract <- add_column(SOAS_extract, geo.cover)
# coverting these to other
SOAS_extract$geo.cover[SOAS_extract$geo.cover == 3] <- "other"
```

```
# appending new column to match categories of original
# whole uk
for(i in whole.uk){
  for(j in 1:100){
    if(SOAS_extract$`Geographic Coverage`[j] == i){
      SOAS_extract$geo.cover[j] <- "whole UK"
    }
  }
}
SOAS_extract$geo.cover
# national
for(i in national.new){
  for(j in 1:100){
    if(SOAS_extract$`Geographic Coverage`[j] == i){
      SOAS_extract$geo.cover[j] <- "national cover"
    }
  }
}
SOAS_extract$geo.cover
table(SOAS_extract$geo.cover)
# removing old column (5)
SOAS_extract <- SOAS_extract[-5]
```

Table 14. 'Geographic Coverage' Variable Groups

Geographic Coverage	Count
Whole UK	41
National	43
Other	16

4.2.8 Column 8 'Time Period Coverage'

This column was already appropriately categorised into five categories. However, the "Annual" category was separated into multiple subcategories. These were combined into a single "Annual" category. The counts for these categories can be found below in **Table 15**.

```
# Column 8 'time period coverage'
# converting any annual to 'annual' group
# Annual - Financial Year
afy <- which(SOAS_extract$`Time Period Coverage` == "Annual -
Financial Year")
SOAS_extract$`Time Period Coverage`[afy] <- "Annual"

# Annual - Calendar Year
acy <- which(SOAS_extract$`Time Period Coverage` == "Annual -
Calendar Year")
SOAS_extract$`Time Period Coverage`[acy] <- "Annual"
# Annual - Snapshot
as <- which(SOAS_extract$`Time Period Coverage` == "Annual -
```

```
Snapshot")
SOAS_extract$`Time Period Coverage`[as] <- "Annual"

# Annual - Mid Year
amy <- which(SOAS_extract$`Time Period Coverage` == "Annual - Mid
Year")
SOAS_extract$`Time Period Coverage`[amy] <- "Annual"

# Annual - Academic Year
aay <- which(SOAS_extract$`Time Period Coverage` == "Annual -
Academic Year")
SOAS_extract$`Time Period Coverage`[aay] <- "Annual"
table(SOAS_extract$`Time Period Coverage`)
```

Table 15. 'Time Period Coverage' Variable Groups

Time Period Coverage	Count
Annual	44
Quarterly	40
Monthly	11
Weekly	1
Daily	4

4.3 Clustering Analysis

```
# Loading libraries
library(cluster)
library(fpc)
library(tidyverse)
library(dendextend)
library(factoextra)
# Creating factors for the appropriate variables
SOAS_extract$Theme <- as.factor(SOAS_extract$Theme)
SOAS_extract$`Format of Admin Source` <-
as.factor(SOAS_extract$`Format of Admin Source`)
SOAS_extract$`Data Type` <- as.factor(SOAS_extract$`Data Type`)
SOAS_extract$`Time Period Coverage` <- as.factor(SOAS_extract$`Time
Period Coverage`)
SOAS_extract$is.numeric <- as.factor(SOAS_extract$is.numeric)
SOAS_extract$pop.coverage <- as.factor(SOAS_extract$pop.coverage)
SOAS_extract$geo.cover <- as.factor(SOAS_extract$geo.cover)
```

4.3.1 Dissimilarity matrix

As many of the variables are categorical the Euclidean distance as a measure of dissimilarity is not appropriate. As such the dissimilarity between observations will be calculated using Gower's distance.

```
# dissimlarity matrix
gower.dist <- daisy(SOAS_extract[-1], metric = c("gower"))
class(gower.dist)
```

4.3.2 Selecting the Agglomerative Clustering Method

As there are alternative methods to perform agglomerative clustering (average, single, complete, and ward) the agnes() function is used to obtain the agglomerative coefficient for each method on the data. The agglomerative coefficient measures the amount of clustering structure, values closer to 1 show strong clustering structure. This output can be found below in **Table 16**, this output shows Ward achieved the highest coefficient and as such will be selected as the method for agglomerative clustering.

```
# methods to assess
m <- c("average", "single", "complete","ward")
names(m) <- c("average", "single", "complete","ward")

# function to compute coefficient
ac <- function(x){
  agnes(gower.dist, method = x)$ac
}

map_dbl(m,ac)
```

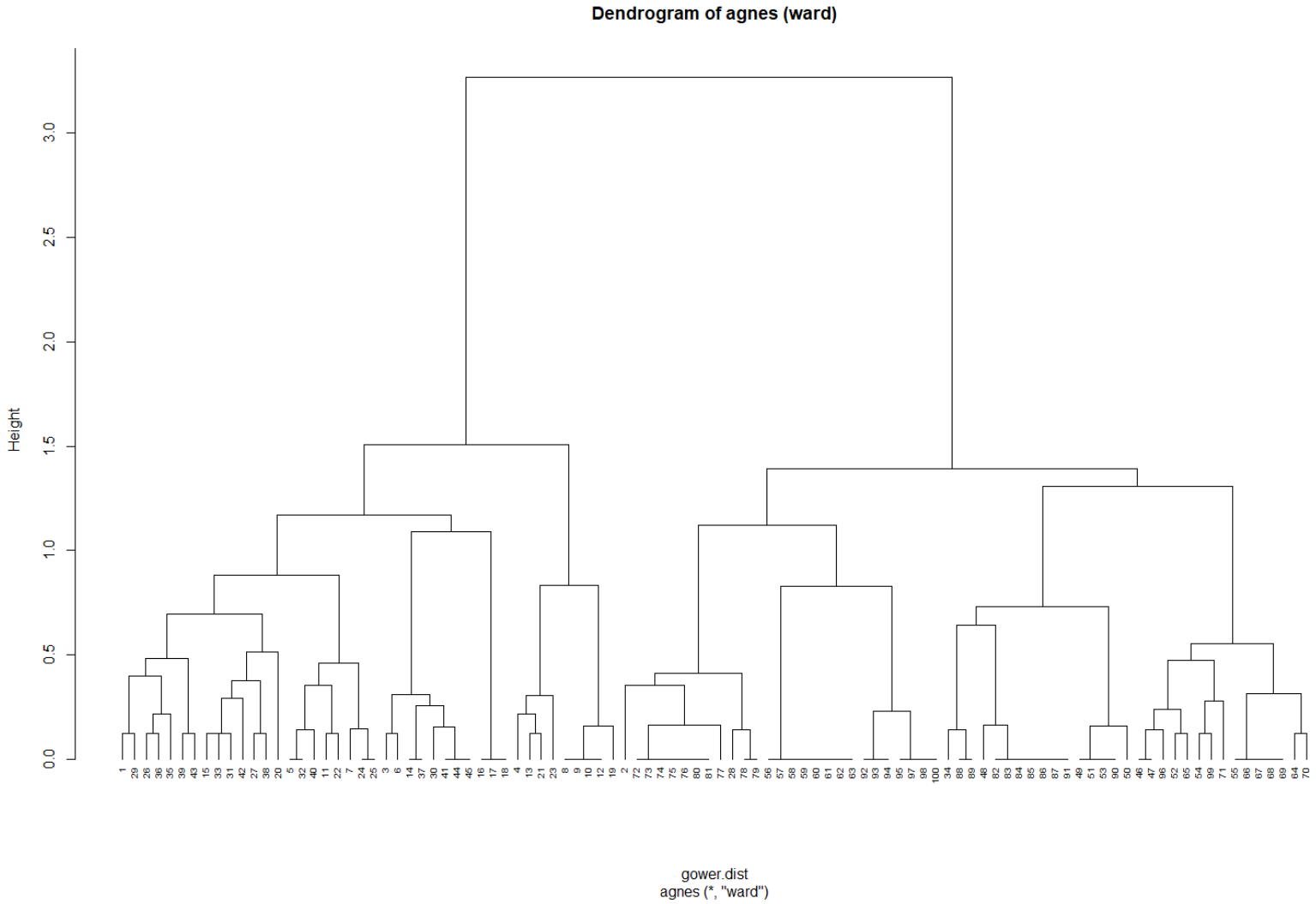
Table 16. Agglomerative Coefficients for Each Method

Method	Agglomerative Coefficient
Average	0.90
Single	0.85
Complete	0.93
Ward	0.98

Using Ward method, a dendrogram is produced to visualise the clustering. This can be found below in **Figures 6**.

```
# visualise dendrogram

# ward
aggl.clus.ward <- agnes(gower.dist, method = "ward")
pltree(aggl.clus.ward, cex=0.6, hang=-1,
       main="Dendrogram of agnes (ward)")
```

Figure 6. Dendrogram for Ward Agglomerative



4.3.3 Selecting the Number of Clusters

To identify the optimal number of clusters the ‘elbow’ method was chosen. The ‘elbow’ method involves plotting the within cluster sum of squares (WSS) as a function of the number clusters, with the to minimise this value. After plotting these WSS the location of a bend in the plot is considered as an indicator of the appropriate number of clusters. An alternative method is to use the “Calinski & Harabasz index”. Both plots can be found below in **Figures 7** and **8**. Although there is no clear elbow from **Figure 7**, it may be reasonable to suggest there is an elbow present at k=8. This was the number of clusters selected within this study.

```
# Ward Method
hicla1 <- hclust(d=gower.dist, method = "ward.D")

max.k <- 15
group <- cutree(hicla1,2:max.k)

## use a loop to construct continuous calls to cluster.stats
for (i in 1:(max.k-1)){
  ## we drop the silhouette criterion to save computation time
  call <- paste("clustat",i+1,"<- cluster.stats(gower.dist,
group[,",i,"], silhouette = FALSE)", sep = "")
  eval(parse(text = call)) #evaluates the text string containing an
R-command.
}
ls(pattern = "clustat")

## Extract the within cluster sum of squares
wss <- numeric(length = 0) #object (numeric vector) to hold WSS's
for (i in 1:(max.k-1)){
  wss <- c(wss,eval(parse(text =
paste("clustat",i+1,"$within.cluster.ss",
                                          sep = ""))))
}
## Extract the Calinski & Harabasz index
ch <- numeric(length = 0) #object to hold CH-indices
for (i in 1:(max.k-1)){
  ch <- c(ch,eval(parse(text = paste("clustat",i+1,"$ch", sep =
"")))))
}
plot(2:max.k, wss , type = "b", xlab = "Number of clusters", ylab =
"Within-c
luster sum of squares")

plot(2:max.k, ch , type = "b", xlab = "Number of clusters", ylab =
"Calinski
& Harabasz index")
```

Figure 7. Elbow Plot Using Within Cluster Sum of Square Errors

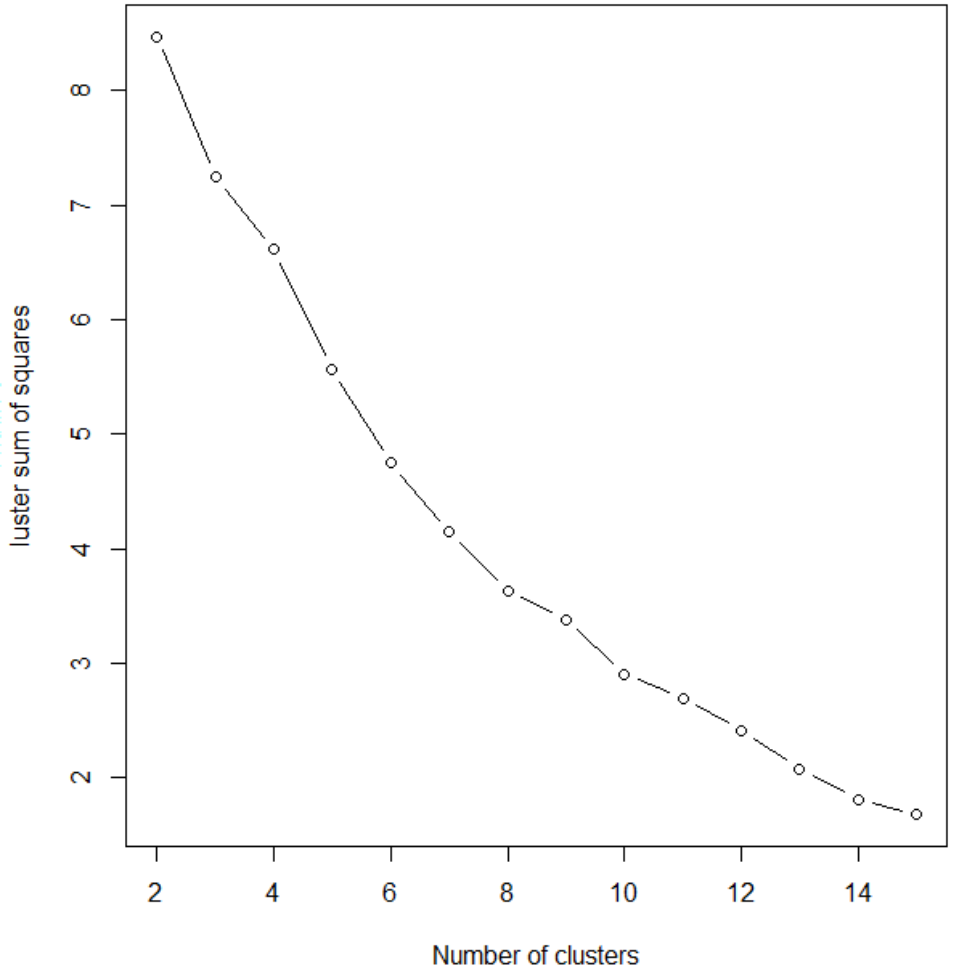
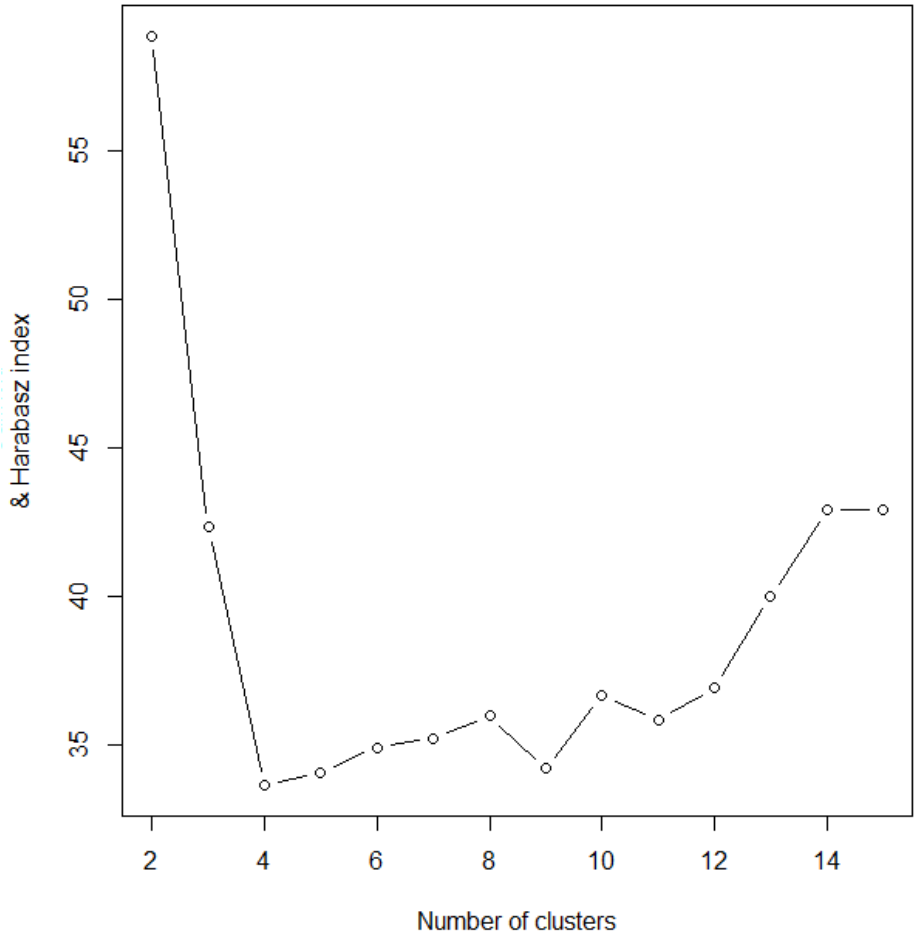


Figure 8. Elbow Plot Using Calinski & Harabasz index

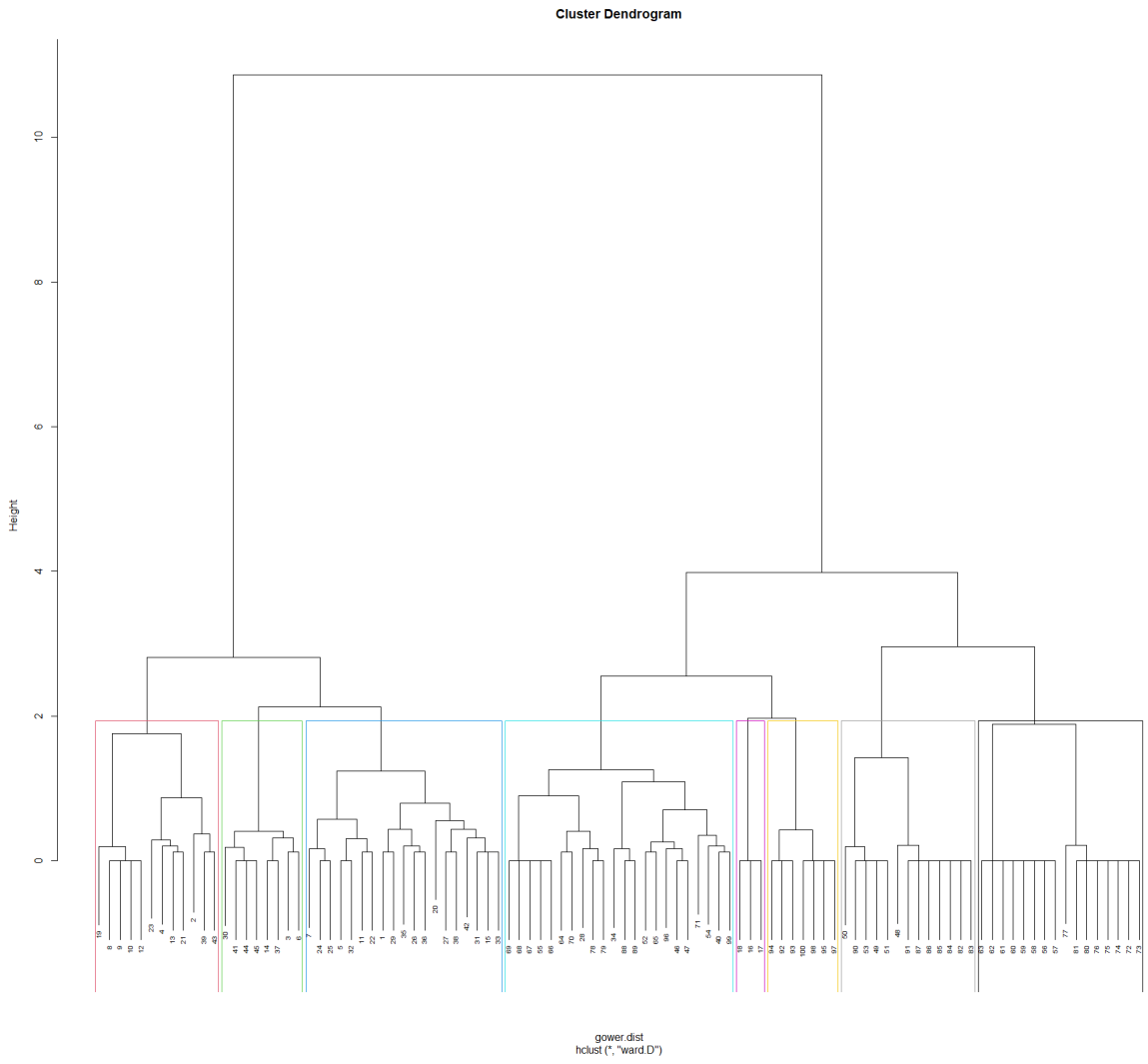


4.4 Visualisations of Final Clusters

Using the ward method with 8 clusters the final clustering is performed. This can be observed below in **Figure 9** which differentiates each cluster by a coloured rectangle.

```
# visualising final results
final.clus <- hclust(gower.dist, method="ward.D")
# cut tree into 8 groups
sub_grp <- cutree(final.clus, k=8)
# number of members in each cluster
table(sub_grp)
```

Figure 9. Final Cluster Dendogram using Ward Method



4.5 Cluster Interpretation

```
# adding clusters to orginal data
SOAS_extract <- SOAS_extract %>%
  mutate(cluster=sub_grp)


clus.sum <- SOAS_extract %>%
  group_by(cluster) %>%
  do(the_summary = summary(.))

clus.sum$the_summary
```

4.5.1 Cluster 1

The summary for this cluster can be found below in **Table 17**. This table displays the variables within the study and the most occurring category of said variable within the cluster. It appears cluster 1 relates to economic observations that are covered on an annual basis which are of microdata data type. This cluster contains 19 observations.

Table 17. Cluster Summary for Cluster 1

Variable	Dominant Category
Theme	Economy
Format of Admin Source	Electronic
Data Type	Microdata
Time Period Coverage	Annual
Is.numeric	Yes
Population Coverage	Other
Geographic Coverage	National
Total Observations	19

4.5.2 Cluster 2

Similar to cluster 1 this cluster contains numerical observations however these observations are of the microdata and aggregate data type covered on an monthly and quarterly time period coverage (**Table 18**).

4.5.3 Cluster 3

Again, similar to clusters 1 and 2 this cluster is also numerical observations. However, this cluster contains only national population and national geographic cover and relates to observations from the Health and Social Care theme (**Table 19**).

Table 18. Cluster Summary for Cluster 2

Variable	Dominant Category
Theme	Travel and Transport
Format of Admin Source	Electronic
Data Type	Microdata & Aggregate
Time Period Coverage	Quarterly, Monthly
Is.numeric	Yes
Population Coverage	World
Geographic Coverage	Whole UK
Total Observations	12

Table 19. Cluster Summary for Cluster 3

Variable	Dominant Category
Theme	Health and Social Care
Format of Admin Source	Electronic
Data Type	Microdata
Time Period Coverage	Annual
Is.numeric	Yes
Population Coverage	National
Geographic Coverage	National
Total Observations	8

4.5.4 Cluster 4

Cluster 4 are the final numeric observations of ‘Management Information’ data type, of the ‘smaller populations’ population coverage and from the ‘people and places’ theme. This cluster is the smallest cluster with only 3 total observations (**Table 20**).

Table 20. Cluster Summary for Cluster 4

Variable	Dominant Category
Theme	People and Places
Format of Admin Source	Electronic
Data Type	Management Information
Time Period Coverage	Annual
Is.numeric	Yes
Population Coverage	Smaller Populations
Geographic Coverage	Other
Total Observations	3

4.5.5 Cluster 5

Cluster 5 is the first cluster of the non-numeric variables. The time-period coverage for these observations is Annual and the geographic coverage is National. Although 'Population' was the dominant category many other themes were also included suggesting this cluster was not grouped by theme but by geographic coverage of the observations. (**Table 21**).

Table 21. Cluster Summary for Cluster 5

Variable	Dominant Category
Theme	Population
Format of Admin Source	Electronic
Data Type	Aggregate
Time Period Coverage	Annual
Is.numeric	No
Population Coverage	Other
Geographic Coverage	National
Total Observations	22

4.5.6 Cluster 6

Cluster 6, like cluster 5 are non-numerical observations, however these observations are from the quarterly time period coverage (**Table 22**) and the Whole UK geographic coverage.

Table 22. Cluster Summary for Cluster 6

Variable	Dominant Category
Theme	Economy
Format of Admin Source	Electronic
Data Type	Aggregate
Time Period Coverage	Quarterly
Is.numeric	No
Population Coverage	National
Geographic Coverage	Whole UK
Total Observations	13

4.5.7 Cluster 7

Cluster 7 appears to be very similar to cluster 6, both are non-numeric observations from the quarterly time coverage. However, the population coverage for cluster 7 is 'smaller populations' whereas cluster 6 was 'national' (**Table 23**).

4.5.8 Cluster 8

The final cluster, cluster 8 is also non-numerical however these observations are from the 'Government' theme and a population coverage of 'smaller populations' (**Table 24**).

Table 23. Cluster Summary for Cluster 7

Variable	Dominant Category
Theme	Labour Market Population
Format of Admin Source	Electronic
Data Type	Aggregate
Time Period Coverage	Quarterly
Is.numeric	No
Population Coverage	Smaller Populations
Geographic Coverage	Whole UK
Total Observations	16

Table 24. Cluster Summary for Cluster 8

Variable	Dominant Category
Theme	Government
Format of Admin Source	Electronic
Data Type	Aggregate
Time Period Coverage	Annual
Is.numeric	No
Population Coverage	Smaller Populations
Geographic Coverage	Whole UK
Total Observations	7

4.6 Conclusion

In general, it appears that clustering was done based on first whether the data for the number of records was numeric or not. Another recurring way the clusters were grouped was based on the 'Data Type', for numeric data whether the type was microdata or management. However, for non-numerical clustering the data type was always aggregate. For these non-numeric observations, the "theme" of the observation was more important in determining the clusters.

https://medium.com/@vieille.francois/compare-clusters-with-comparegroups-package-in-r-4cac20a0c00e

Part C

Write a web scraping program to extract the titles from an official statistics related blog site other than National Statistical (that is, do not use https://blog.ons.gov.uk/ as your site). Find a suitable R package (or similar) to produce a word cloud from these titles. Comment on the words which are most frequent, in relation to the organisation/person that wrote the blogs

5.1 Introduction

Within this analysis a web scraping program is used to extract the titles from The Government Statistical Service (GSS) blogs (available at: https://gss.civilservice.gov.uk/blog/). Using these titles a word cloud is produced, the most frequent words are then analysed in relation to the organisation/person that authored the blog.

5.2 Web Scraping Program

The web scraping program below scrapes the first page of the blog for the titles. Next a for loop is used to access the pages two – five, obtaining all the titles of each blog on the site. This produces fifty blog titles obtained from the site.

```
# loading libraries for the scraping and cleaning
library(rvest)
library(tidyverse)
library(stringr)
# conducting the first scraping on the page 1

# assigning the url
url <- "https://gss.civilservice.gov.uk/blog/"

webpage <- read_html(url)

# Reading the Titles

Title_html <- html_nodes(webpage, 'h2 a')

blog.title <- html_text(Title_html)


# writing a for loop to scrape the titles for the
# rest of the pages.
for (i in 2:5){
  urltemp <- paste(url,"page/",i,"/", sep="")
  webpage <- read_html(urltemp)
  next_title_html <- html_nodes(webpage, 'h2 a')
```

```
  next_title <- html_text(next_title_html)
  blog.title <- append(blog.title,next_title)
}

# checking results
blog.title
```

However, in their raw from these titles are not suitable for any word cloud libraries. The titles are first 'cleaned' by remove all punctation and separating the titles into separate words.

```
# creating a for loop to replace all punctuation in titles with ""
title.clean <- c()
for(string in blog.title){
  current_title <- string
  replace_title <- str_replace_all(current_title, "[[:punct:]]", "")
  title.clean <- c(title.clean, replace_title)
}
title.clean

# seperating each title into seperate words
title.words <- c()
for (title in title.clean){
  current_title <- title
  current_words <- strsplit(current_title, " ")[[1]]
  title.words <- c(title.words, current_words)
}
title.words
```

In order to comment on the most common words in relation to the author of such titles containing these words, the author names for each blog was obtained. To achieve this the URL web address of each blog is used to access the bog as this is where the author information is available.

```
# scraping all authors for each blog
# Getting links for all blogs

# for first page
gss <- read_html('https://gss.civilservice.gov.uk/blog/')

gss_link <- 'https://gss.civilservice.gov.uk/blog/'

link <- gss %>%
  html_nodes("h2 a") %>%
  html_attr("href")

link

# for pages 2-5
for (i in 2:5){
  urltemp <- paste(gss_link,"page/",i,"/", sep="")
  webpage <- read_html(urltemp)
  next_link <- webpage %>%
    html_nodes("h2 a") %>%
    html_attr("href")
```

```
  link <- append(link, next_link)
}
link


# Use these links to scrape all authors

# Getting all authors for all blogs

author <- c()
for(l in link){
  web_temp <- read_html(l)
  author_html <- html_nodes(web_temp, '.font-bold')
  author_text <- html_text(author_html)
  author <- c(author, author_text)
}

author
```

The author information is also cleaned.

```
# Cleaning all authors
author.clean <- c()
for(string in author){
  current_author <- string
  replace_author_remove1 <- str_replace_all(current_author, "\n\n",
"")
  replace_author_remove2 <- str_replace_all(replace_author_remove1,
"\n", "")
  replace_author_remove3 <- str_replace_all(replace_author_remove2,
" By ", "")
  final_author_trim <- str_trim(replace_author_remove3)
  author.clean <- c(author.clean, final_author_trim)
}
author.clean
```

5.3 Word Cloud

To generate a word cloud from the blog title data the ‘tm’ library was used for data mining purposes, this was essential for preparing the data for the word clouds. Additionally, the ‘wordcloud’ packaged was used to generate the word cloud.

```
# loading the required libraries
library(wordcloud)
library(tm)
```

First the data was converted to a text document

```
# converting titles to text
write.table(title.words, file = "title.txt", sep = "",
            row.names = FALSE)
```

This text document is read into R and converted to a Corpus, a data structure required in order to produce the word cloud.

```
# Loading the text
text <- readLines("title.txt")
text

# Load the data as a corpus
docs <- Corpus(VectorSource(text))
inspect(docs)
```

The words within the corpus are further cleaned and prepared for the word cloud. Tm_map() is used to remove English common stopwords, these include words such as ‘the’ and ‘is’, these need to be removed to ensure meaningful results.

```
# Cleaning the text
# removequotes
removeSpecialChars <- function(x) gsub("[^a-zA-Z0-9 ]","",x)
docs <- tm_map(docs, content_transformer(removeSpecialChars))
inspect(docs)

# Remove english common stopwords
docs <- tm_map(docs, removeWords, stopwords("english"))
inspect(docs)

# Convert the text to lower case
docs <- tm_map(docs, content_transformer(tolower))
inspect(docs)

# Eliminate extra white spaces
docs <- tm_map(docs, stripWhitespace)
inspect(docs)

# removing "i", "the", and "how"
docs <- tm_map(docs, removeWords, c("i", "the", "how"))
```

A term document matrix is created to store the count of each word as a matrix.

```
# Build a term document matrix
dtm <- TermDocumentMatrix(docs)
matrix <- as.matrix(dtm)
sorted.matrix <- sort(rowSums(matrix), decreasing = TRUE)
word.data.frame <- data.frame(word = names(sorted.matrix), freq =
sorted.matrix)
```

Using this matrix wordcloud() is used to generate the word cloud of the words within the titles from the GSS blogs. The max words were set to 60 to enable a better visualisation. This word cloud can be

found below in **Figure 10.** Additionally, **Table 25** displays the 10 most frequent words and their counts.

```
# Generate the word cloud
set.seed(1234)
wordcloud(words=word.data.frame$word, freq = word.data.frame$freq,
min.freq = 1,
          max.words = 60, random.order = FALSE, rot.per = 0.35,
          color = brewer.pal(8, "Dark2"))
```

Figure 10. Word Clouds

Table 25. GSS Blogs Titles Top 10 Words

Word	Count
Statistic	8
Data	6
Gss	6
Practice	4
User	4
Engagement	4

Strategy	4
Quality	4
Secondment	3
Best	3

5.4 Word Cloud Interpretation

To assess word occurrence by author, the two variables are combined into a data frame. The titles are then split into separate words. A custom function is then used to find the authors corresponding to the occurrence of a specific word. For the most occurring word, statistic, the authors identified can be found below in **Table 26**. In terms of interpretation, the reason why this word may have been used specifically by these authors may be due to these authors being statisticians of some kind. Additionally, as the GSS blogs are statistics blogs it is expected for the most common words to be "Statistic" and "Data". However, the relatively high occurrence of "User" and "Engagement" may emphasis a key goal of the GSS, to increase the inclusivity of statistics, making it accessible to all users. The high occurrence of 'Quality' and 'Strategy' may reflect the GSS's "quality strategy" wherein they emphasises the importance of high quality statistics to build trust.

```
# combining title and author
blogs <- data.frame(title.clean, author.clean)
# Accessing first row
blogs[1,]
words <- c()
for (title in title.clean){
  current_title <- title
  current_words <- strsplit(current_title, " ")
  words <- c(words, current_words)
}

for(i in 1:50){
  blogs$word[i] <- wordst[i]
}
length(blogs$word[[1]])


# make new variable
yes <- rep(0,50)
blogs <- add_column(blogs, yes)
find_authors <- function(word){
  blogs[grep(word, blogs$word, value = F), "yes"] <- 1
  rows <- which(blogs$yes == 1)
  authors <- blogs$author.clean[rows]
  return(authors)
  blogs$yes <- yes
}


find_authors("statistic")
```

Table 26. Authors of Blogs With "Statistic" in the Title

Author
Tegwen Green
Helen Miller-Bakewell
George Pickering
David Mais and David Foster
Darren Barnes
Andy Schofield
Angela Potter
Esther Sutherland & Ian Boreham
Paul Niblett