CS 699 Assignment 1

Katherine Rein

```
# Import libraries
library(glue)
```

```
## Warning: package 'glue' was built under R version 4.3.3
```

```
library(modeest)
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(fastDummies)
```

```
## Warning: package 'fastDummies' was built under R version 4.3.3
```

```
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

Problem 1

Question 1

Calculate the mean, median, and standard deviation (sample) of the age feature

```
# Read in data
autism_data = read.csv('autism-adult.csv')

# Drop fully NA rows
autism_data <- autism_data[!apply(is.na(autism_data), 1, all), ]

# Remove 383 year old
autism_data = autism_data[autism_data$age != 383, ]

# Convert age to integer
autism_data$age = as.integer(autism_data$age)
```

```
## Warning: NAs introduced by coercion
```

```r
# Calculate the mean median and standard deviation for the age feature
mean =  mean(autism_data$age, na.rm = TRUE)
median = median(autism_data$age, na.rm = TRUE)
stdev = sd(autism_data$age, na.rm = TRUE)

# Print answers
glue('Mean: {mean}')
```

```
## Mean: 29.1940085592011
```

```r
glue('Median: {median}')
```

```
## Median: 27
```

```r
glue('Standard Deviation (sample): {stdev}')
```

```
## Standard Deviation (sample): 9.71152590893556
```

After some preliminary data analysis, I noticed there was someone who was 383 years old which seems impossible. I removed this entry before continuing on.
The mean of the data set is 29.20. The median is 27. The sample standard deviation is 9.71.

Question 2

Determine Q1, Q2, and Q3 of age

```r
# Calculate Q1, Q2, Q3
quantile_vector = quantile(autism_data$age, probs = c(0.25, 0.5, 0.75), na.rm = TRUE)

# Store individually
Q1 = quantile_vector[1]
Q2 = quantile_vector[2]
Q3 = quantile_vector[3]

# Print
glue('Q1: {Q1}, Q2: {Q2}, Q3: {Q3}')
```

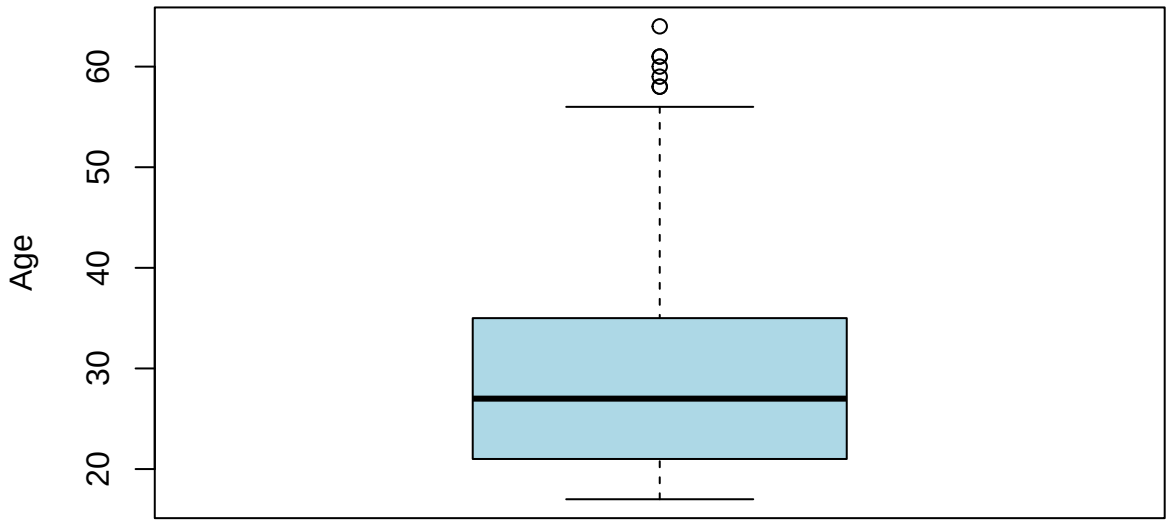
```
## Q1: 21, Q2: 27, Q3: 35
```

Q2 is the same as the median of the data set but I also recalculated it. Q1 is 21, Q2 is 27, and Q3 is 35.

Question 3

Plot the boxplot of the age feature

```r
# Plot Boxplot
boxplot(autism_data$age,
        main = "Distribution of Age", ylab = "Age",
        col = "lightblue", notch = FALSE, na.rm = TRUE)
```



Question 4

Implement min-max rescaling on the age feature. Replace the original age feature with the rescaled result. Provide the rescaled age for the seventh observation in the data.

```
# Find the minimum and maximum age
min_age = min(autism_data$age, na.rm = TRUE)
max_age = max(autism_data$age, na.rm = TRUE)

# Implement min-max rescaling to the [0, 1] interval
# Formula: (x - min) / (max - min)
autism_data$age = (autism_data$age - min_age) / (max_age - min_age)

# Print seventh observation
seventh_obs = autism_data$age[7]
print(seventh_obs)
```

```
## [1] 0
```

For min-max rescaling I used a minimum value of 0 and max of 1. The seventh observation is now 0. This means that before the data was rescaled it was the minimum age.

Question 5

Determine the mode of the country_of_res feature

```
# Handle NAs
autism_data$country_of_res <- dplyr::na_if(autism_data$country_of_res, '')

# Create a frequency table of unique values
freq_loc <- table(autism_data$country_of_res, useNA = "no")

# Find the most frequent values of a vector
mode_loc <- modeest::mfv(autism_data$country_of_res, na_rm = TRUE)
```

```r
# Concatenate frequencies and modes
list(frequencies = freq_loc, mode = mode_loc)
```

```
## $frequencies
##
##           'Costa Rica'       'Czech Republic'            'Hong Kong'
##                      1                      1                      1
##         'New Zealand'         'Saudi Arabia'         'Sierra Leone'
##                     80                      4                      1
##        'South Africa'           'Sri Lanka' 'United Arab Emirates'
##                      2                     14                     82
##      'United Kingdom'       'United States'             'Viet Nam'
##                     77                    113                      5
##           Afghanistan          AmericanSamoa                 Angola
##                     13                      2                      1
##             Argentina               Armenia                  Aruba
##                      2                      2                      1
##             Australia               Austria             Azerbaijan
##                     27                      4                      1
##               Bahamas            Bangladesh                Belgium
##                      2                      3                      3
##               Bolivia                Brazil                Burundi
##                      1                      9                      1
##                Canada                 Chile                 China
##                     15                      1                      1
##                Cyprus               Ecuador                 Egypt
##                      1                      1                      3
##              Ethiopia               Finland                France
##                      2                      1                     11
##               Germany               Iceland                 India
##                      4                      2                     81
##             Indonesia                  Iran                  Iraq
##                      1                      7                      1
##               Ireland                 Italy                 Japan
##                      5                      5                      1
##                Jordan            Kazakhstan                Lebanon
##                     47                      3                      1
##              Malaysia                Mexico                 Nepal
##                      5                      8                      1
##           Netherlands             Nicaragua                 Niger
##                     10                      1                      1
##                  Oman              Pakistan           Philippines
##                      1                      3                      4
##              Portugal               Romania                Russia
##                      1                      3                      7
##                Serbia                 Spain                Sweden
##                      1                      3                      2
##                 Tonga                Turkey                Ukraine
##                      1                      1                      2
##               Uruguay
##                      1
##
## $mode
## [1] "'United States'"
```

The mode of the country of residence column is United States with 113 entries.

Question 6

Review the ethnicity feature. You will notice several missing values in this feature. Determine a reasonable imputation for this feature. Explain what you are going to do and why. Then replace the original ethnicity feature with the imputed result.

```
# Identify all categories used in the ethnicity feature
unique(autism_data$ethnicity)
```

```
##  [1] "White-European"    "Latino"            "?"
##  [4] "Others"            "Black"             "Asian"
##  [7] "'Middle Eastern '" "Pasifika"          "'South Asian'"
## [10] "Hispanic"          "Turkish"           "others"
```

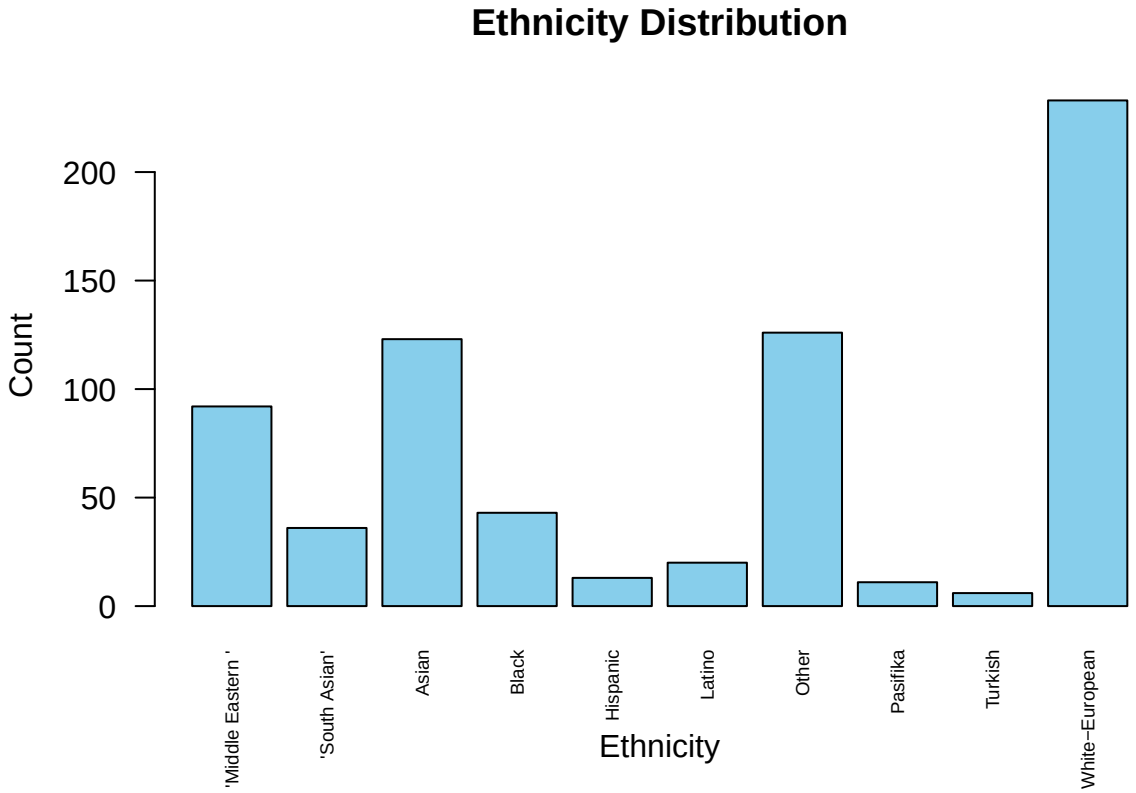
```
# Turn missing values into NA
autism_data$ethnicity <- dplyr::na_if(autism_data$ethnicity, '')

# Change all other values into one other value
autism_data <- autism_data %>%
  mutate(ethnicity = case_when(
    ethnicity == '?' ~ 'Other',
    ethnicity == 'others' ~ 'Other',
    ethnicity == 'Others' ~ 'Other',
    ethnicity == NA ~ 'Other',
    TRUE ~ ethnicity
  ))
```

After looking at all of the unique values for the ethnicity feature, I noticed there were many entries that meant other. I then changed all of them to other. This seemed like the best way to not overinflate one category (as we don't know where these individuals came from).

Question 7

Create a bar graph of your imputed ethnicity feature.

```
barplot(table(autism_data$ethnicity),
        main = "Ethnicity Distribution",
        xlab = "Ethnicity",
        ylab = "Count",
        col = "skyblue",
        cex.names = 0.6,
        las = 2)
```



Question 8

Implement dummy coding for the gender feature. Replace the original gender feature with the coded result. Provide the coded gender for the last ten observations in the data.

```
# Ensure gender is treated as a categorical factor
autism_data$gender <- as.factor(autism_data$gender)

# One encoding of gender column
d <- fastDummies::dummy_cols(autism_data,
                             select_columns = "gender",
                             remove_selected_columns = TRUE,
                             remove_first_dummy = TRUE
                             )

# Show the last 10
coded_gender_last10 = tail(d[ , grep("^gender_", names(d)) ], 10)
coded_gender_last10
```

```
##  [1] 1 1 1 0 0 0 1 0 1 0
```

Question 9

Identify which features in your data set are discrete and which are continuous.

From visual investigation, it seems that the following features are discrete: A1_score - A10_score, gender, ethnicity, jaundice, autism, country of residence, used app before, relation, and class ASD. The only continuous feature is age.

Question 10

Identify which features in your data set are numeric and which are non-numeric. Compare with the discrete/continuous classification you just made and discuss the similarities and/or differences you see.

```
# Convert blank strings to NA
autism_data <- mutate(autism_data, across(where(is.character), ~ dplyr::na_if(.x, '')))

# Identify column classes of the data
col_classes <- sapply(autism_data, class)

# Link classes and names
col_classes_df = data.frame(Column = names(autism_data), Class = col_classes)

print(col_classes_df)
```

```
##                          Column     Class
## A1_Score               A1_Score   integer
## A2_Score               A2_Score   integer
## A3_Score               A3_Score   integer
## A4_Score               A4_Score   integer
## A5_Score               A5_Score   integer
## A6_Score               A6_Score   integer
## A7_Score               A7_Score   integer
## A8_Score               A8_Score   integer
## A9_Score               A9_Score   integer
## A10_Score             A10_Score   integer
## age                         age   numeric
## gender                   gender    factor
## ethnicity             ethnicity character
## jaundice               jaundice character
## austim                   austim character
## country_of_res   country_of_res character
## used_app_before used_app_before character
## relation               relation character
## Class.ASD             Class.ASD character
```

The following columns are numeric: A1_Score - A10_Score and age. The non numeric columns are: gender, ethnicity, jaundice, autism, country of residence, used app before, relation, and Class ASD. The only difference between numeric and continuous is that the A score columns are discrete data with numeric classes. This makes them a numeric discrete feature which is unlike any other column.

Question 11

After completing all requested tasks above, print the first 4 observations of the data.

```
head(autism_data, n = 4)
```

```
##   A1_Score A2_Score A3_Score A4_Score A5_Score A6_Score A7_Score A8_Score
## 1        1        1        1        1        0        0        1        1
## 2        1        1        0        1        0        0        0        1
## 3        1        1        0        1        1        0        1        1
## 4        1        1        0        1        0        0        1        1
##   A9_Score A10_Score       age gender      ethnicity jaundice austim
## 1        0         0 0.1914894      f White-European       no     no
## 2        0         1 0.1489362      m         Latino       no    yes
```

```
## 3        1         1 0.2127660      m         Latino      yes    yes
## 4        0         1 0.3829787      f White-European       no    yes
##    country_of_res used_app_before relation Class.ASD
## 1 'United States'              no     Self        NO
## 2          Brazil              no     Self        NO
## 3           Spain              no   Parent       YES
## 4 'United States'              no     Self        NO
```

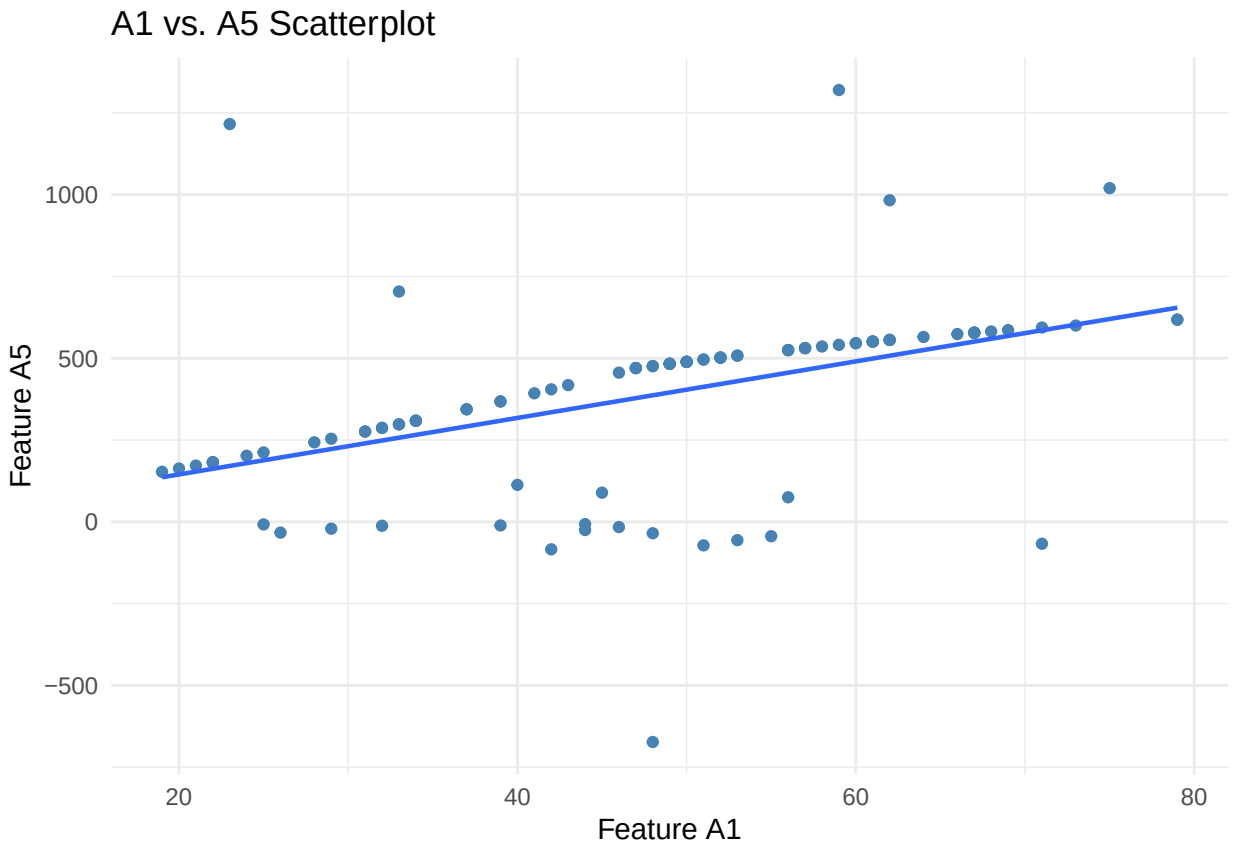
Problem 2

Question 1

Create a scatterplot of feature A1 vs. feature A5.

```
# Read in data
corr_data = read.csv('correlation.csv')

# Drop NAs in both columns
corr_clean <- tidyr::drop_na(corr_data, A1, A5)

# Make all values numeric
corr_clean$A1 <- suppressWarnings(as.numeric(corr_clean$A1))
corr_clean$A5 <- suppressWarnings(as.numeric(corr_clean$A5))

# Create scatterplot
ggplot(corr_clean, aes(x = A1, y = A5)) +
  geom_point(color = "steelblue") +
  geom_smooth(method = "lm", se = FALSE, linewidth = 0.8) + labs(title = "A1 vs. A5 Scatterplot",
                                                                 x = "Feature A1",
                                                                 y = "Feature A5") + theme_minimal()
```

```
## `geom_smooth()` using formula = 'y ~ x'
```



Question 2

Compute the correlation matrix for all five features in the data set.

```
# Create a vector of column names
cols = paste0("A", 1:5)

# Ensure all columns are numeric
for (v in cols) corr_data[[v]] <- suppressWarnings(as.numeric(corr_data[[v]]))

# Compute the Pearson correlation matrix
cor_mat <- cor(corr_data[cols], use = "pairwise.complete.obs", method = "pearson")

# Print correlation matrix
cor_mat
```

```
##           A1         A2        A3         A4         A5
## A1 1.0000000 0.17880899 0.2896419 0.17717981 0.45241040
## A2 0.1788090 1.00000000 0.4645427 0.34013541 0.04235286
## A3 0.2896419 0.46454268 1.0000000 0.29379683 0.19451370
## A4 0.1771798 0.34013541 0.2937968 1.00000000 0.06186254
## A5 0.4524104 0.04235286 0.1945137 0.06186254 1.00000000
```

Question 3

Identify the strongest correlation in the data set. Which factors are involved? Is it a positive correlation or a negative correlation?

```r
# Switch diagonal to NA
diag(cor_mat) <- NA

# Find the max correlation value
max_val = max(cor_mat, na.rm = TRUE)

# Find the position of that max value
which(cor_mat == max_val, arr.ind = TRUE)
```

```
##    row col
## A3   3   2
## A2   2   3
```

The strongest correlation is 0.465 which correlates A3 and A2. This is a positive correlation because the number is positive.

Question 4

Implement z-score normalization on all features in the data set

```r
corr_data[cols] <- as.data.frame(scale(corr_data[cols], center = TRUE, scale = TRUE))
```

Question 5

Compute the correlation matrix for all five normalized features in the data set. Compare this correlation matrix with the matrix you obtained earlier and discuss the similarities and/or differences you see.

```r
# Create a vector of column names
cols = paste0("A", 1:5)

# Ensure all columns are numeric
for (v in cols) corr_data[[v]] <- suppressWarnings(as.numeric(corr_data[[v]]))

# Compute the Pearson correlation matrix
cor_mat <- cor(corr_data[cols], use = "pairwise.complete.obs", method = "pearson")

# Print correlation matrix
cor_mat
```

```
##           A1         A2        A3         A4         A5
## A1 1.0000000 0.17880899 0.2896419 0.17717981 0.45241040
## A2 0.1788090 1.00000000 0.4645427 0.34013541 0.04235286
## A3 0.2896419 0.46454268 1.0000000 0.29379683 0.19451370
## A4 0.1771798 0.34013541 0.2937968 1.00000000 0.06186254
## A5 0.4524104 0.04235286 0.1945137 0.06186254 1.00000000
```

The correlation matricies are identical. This makes sense because when we normalize features we are already removing the affects of mean and standard deviation.