1. Data Cleaning

Contents

Data Cleaning: dataset 1

```
# Load tidyverse package
library(tidyverse)

# Import our data and save it as datafile
datafile <- read_csv("survey.csv")
```

head()

```
# View the first five rows with head()
# head(filename, n) shows the first n lines of the data file
head(datafile, 6)
```

```
## # A tibble: 6 x 7
##    ...1 Gender Handedness Pulse Exercise Smoke Height
##   <dbl> <chr>  <chr>      <dbl> <chr>    <chr>  <dbl>
## 1     1 Female Right         92 Some     Never   173
## 2     2 Male   Left         104 None     Regul   178.
## 3     3 Male   Right         87 None     Occas    NA
## 4     4 Male   Right         NA None     Never   160
## 5     5 Male   Right         35 Some     Never   165
## 6     6 Female Right         64 Some     Never   173.
```

str()

```
# Check the structure of the variables in the dataset by using str() function
str(datafile)
```

the structure of the variables

```
## spec_tbl_df [237 x 7] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ ...1      : num [1:237] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Gender    : chr [1:237] "Female" "Male" "Male" "Male" ...
##  $ Handedness: chr [1:237] "Right" "Left" "Right" "Right" ...
##  $ Pulse     : num [1:237] 92 104 87 NA 35 64 83 74 72 90 ...
##  $ Exercise  : chr [1:237] "Some" "None" "None" "None" ...
##  $ Smoke     : chr [1:237] "Never" "Regul" "Occas" "Never" ...
##  $ Height    : num [1:237] 173 178 NA 160 165 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   ...1 = col_double(),
##   ..   Gender = col_character(),
##   ..   Handedness = col_character(),
##   ..   Pulse = col_double(),
##   ..   Exercise = col_character(),
##   ..   Smoke = col_character(),
##   ..   Height = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

as.factor()

```
# When we check the structure of the data, we see that some of the variables are stored as characters i
datafile$Gender <- as.factor(datafile$Gender)
```

categorical variables

lappy()

```
# In order to change the data types of multiple columns, we will use lapply() function together with as

# lapply(vector, function)

# Set the correct measurement levels for Gender, Handedness, Exercise and Smoke using lapply() and as.f

# First generate a vector to keep the column names
columns <- c("Gender", "Handedness", "Exercise", "Smoke")

# Set the correct measurement levels or data types
datafile[columns] <- lapply(datafile[columns], as.factor)
```

levels()

```
# Check levels of the Gender column
levels(datafile$Gender)
```

```
## [1] "female" "Female" "male"   "Male"
```

which()

```


# Replace female with Female
# (1) Find the indices of rows with "female" value
( index_female <- which(datafile$Gender == "female") )
```

find indices

```
## [1]  31  37  57 237
```

```
# (2) Replace those entries with "Female"
datafile$Gender[index_female] = "Female"
```

```
# Replace male with Male
# (1) Find the indices of rows with "male" value
( index_male <- which(datafile$Gender == "male") )
```

```
## [1] 22 39 58
```

```
# (2) Replace those entries with "Male"
datafile$Gender[index_male] = "Male"
```

```r
# Check levels of the Gender column again
levels(datafile$Gender)
```

```
## [1] "female" "Female" "male"   "Male"
```

```r
# Although we corrected the entry errors in the Gender column, the levels did not change. Hence, we sho

# Update the levels
datafile$Gender <- factor(datafile$Gender)
```

missing value

summary()

```r
# Check the summary of the data file
summary(datafile)
```

statistical information about the data -> check missing values

```
##       ...1        Gender    Handedness      Pulse        Exercise     Smoke
##  Min.   :  1   Female:118   Left : 18   Min.   : 35.00   Freq:115   Heavy: 11
##  1st Qu.: 60   Male  :118   Right:218   1st Qu.: 66.00   None: 24   Never:189
##  Median :119   NA's  :  1   NA's :  1   Median : 72.50   Some: 98   Occas: 19
##  Mean   :119                           Mean   : 74.15              Regul: 17
##  3rd Qu.:178                           3rd Qu.: 80.00              NA's :  1
##  Max.   :237                           Max.   :104.00
##                                        NA's   :45
##      Height
##  Min.   :150.0
##  1st Qu.:165.0
##  Median :171.0
##  Mean   :172.4
##  3rd Qu.:180.0
##  Max.   :200.0
##  NA's   :28
```

mean()

```r
# Find the average pulse
mean(datafile$Pulse)
```

```
## [1] NA
```

na.omit()

```r
# Remove records with missing values and assign it to datafile_new
datafile_new <- na.omit(datafile)
```

```r
# Calculate the number of records removed from the data
nrow(datafile) - nrow(datafile_new)
```

remove records with missing values and assign it to `datafile_new`.

```
## [1] 68
```

```r
# Find the average of Pulse by excluding the missing values
avg_pulse <- mean(datafile$Pulse, na.rm = T)
            # na.rm: whether or not to remove NA values from the calculation
print(avg_pulse)
```

```
## [1] 74.15104
```

```r
# Find the average of Height by excluding the missing values
avg_height <- mean(datafile$Height, na.rm = T)

print(avg_height)
```

```
## [1] 172.3809
```

replace_na()

```r
# Replace records with missing values
datafile_replace <- replace_na(datafile, list(Pulse = avg_pulse, Height = avg_height))

#  Remove records with missing values and assign the redacted dataset to datafile_removed
datafile_removed <- na.omit(datafile_replace)

# Calculate the number of records removed from the data
nrow(datafile) - nrow(datafile_removed)
```

```
## [1] 3
```

Data Cleaning: dataset 2

new dataset

```r
# Load tidyverse package
library(tidyverse)

# Import the data and save it as crdata
crdata <- read_csv("Credit.csv")
```

```
# View the first four rows with head()
head(crdata, 4)
```

Credit.csv

```
## # A tibble: 4 x 11
##      No Income Limit Rating Cards   Age Education Gender Student Married Balance
##   <dbl>  <dbl> <dbl>  <dbl> <dbl> <dbl>     <dbl> <chr>  <chr>   <chr>     <dbl>
## 1     1   14.9  3606    283     2    34        11 Male   No      Yes         333
## 2     2  106.   6645    483     3    82        15 Female Yes     Yes         903
## 3     3  105.   7075    514     4    71        11 Male   No      No          580
## 4     4  149.   9504    681     3    36        11 Female No      No          964
```

```
# Check the structure of the variables in the dataset by using str() function
str(crdata)
```

```
## spec_tbl_df [400 x 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ No       : num [1:400] 1 2 3 4 5 6 7 8 9 10 ...
##  $ Income   : num [1:400] 14.9 106 104.6 148.9 55.9 ...
##  $ Limit    : num [1:400] 3606 6645 7075 9504 4897 ...
##  $ Rating   : num [1:400] 283 483 514 681 357 569 259 512 266 491 ...
##  $ Cards    : num [1:400] 2 3 4 3 2 4 2 2 5 3 ...
##  $ Age      : num [1:400] 34 82 71 36 68 77 37 87 66 41 ...
##  $ Education: num [1:400] 11 15 11 11 16 10 12 9 13 19 ...
##  $ Gender   : chr [1:400] "Male" "Female" "Male" "Female" ...
##  $ Student  : chr [1:400] "No" "Yes" "No" "No" ...
##  $ Married  : chr [1:400] "Yes" "Yes" "No" "No" ...
##  $ Balance  : num [1:400] 333 903 580 964 331 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   No = col_double(),
##   ..   Income = col_double(),
##   ..   Limit = col_double(),
##   ..   Rating = col_double(),
##   ..   Cards = col_double(),
##   ..   Age = col_double(),
##   ..   Education = col_double(),
##   ..   Gender = col_character(),
##   ..   Student = col_character(),
##   ..   Married = col_character(),
##   ..   Balance = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
# Check the summary of the dataset
summary(crdata)
```

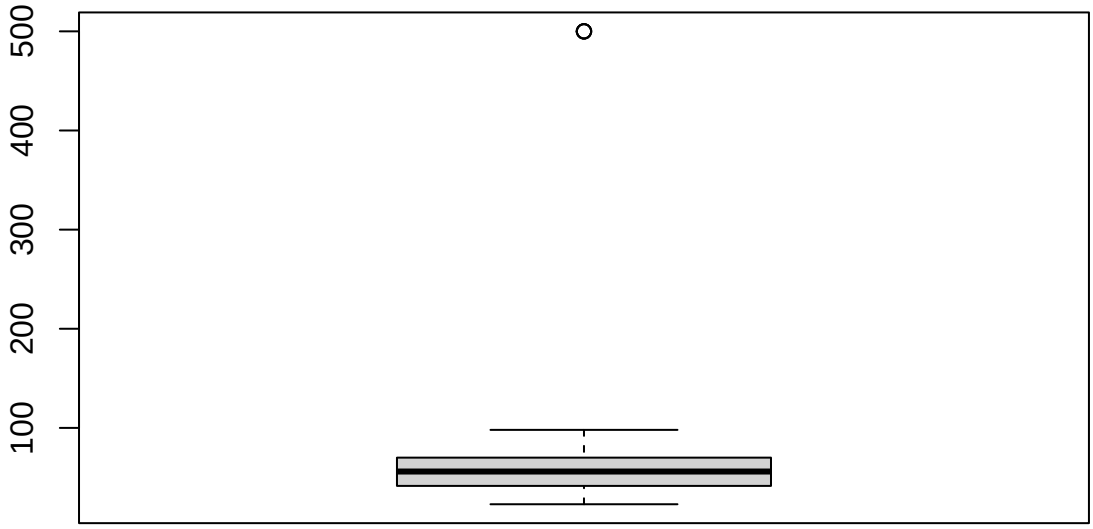
```
##        No             Income           Limit           Rating
##  Min.   :  1.0   Min.   : 10.35   Min.   :  855   Min.   : 93.0
##  1st Qu.:100.8   1st Qu.: 21.02   1st Qu.: 3091   1st Qu.:246.5
```

```
##  Median :200.5   Median : 33.21   Median : 4636   Median :344.0
##  Mean   :200.5   Mean   : 45.28   Mean   : 4745   Mean   :354.7
##  3rd Qu.:300.2   3rd Qu.: 57.60   3rd Qu.: 5880   3rd Qu.:436.0
##  Max.   :400.0   Max.   :186.63   Max.   :13913   Max.   :982.0
##                  NA's   :1        NA's   :2       NA's   :1
##      Cards            Age            Education        Gender
##  Min.   :1.000   Min.   : 23.00   Min.   : 5.00   Length:400
##  1st Qu.:2.000   1st Qu.: 41.50   1st Qu.:11.00   Class :character
##  Median :3.000   Median : 56.00   Median :14.00   Mode  :character
##  Mean   :2.958   Mean   : 58.79   Mean   :13.46
##  3rd Qu.:4.000   3rd Qu.: 70.00   3rd Qu.:16.00
##  Max.   :9.000   Max.   :500.00   Max.   :20.00
##                  NA's   :1        NA's   :1
##    Student             Married             Balance
##  Length:400         Length:400         Min.   :   0.00
##  Class :character   Class :character   1st Qu.:  68.75
##  Mode  :character   Mode  :character   Median : 459.50
##                                        Mean   : 520.01
##                                        3rd Qu.: 863.00
##                                        Max.   :1999.00
##
```

```
# Update the data types if necessary
```

boxplot()

```
# outliner in the Age column of crdata
boxplot(crdata$Age)
```



check outliner

which()

```
# Find the indices of records with age >= 100 and assign these indices to outliers
outliers <- which(crdata$Age>=100)

# Print data records with outliers
print(crdata[outliers,])
```

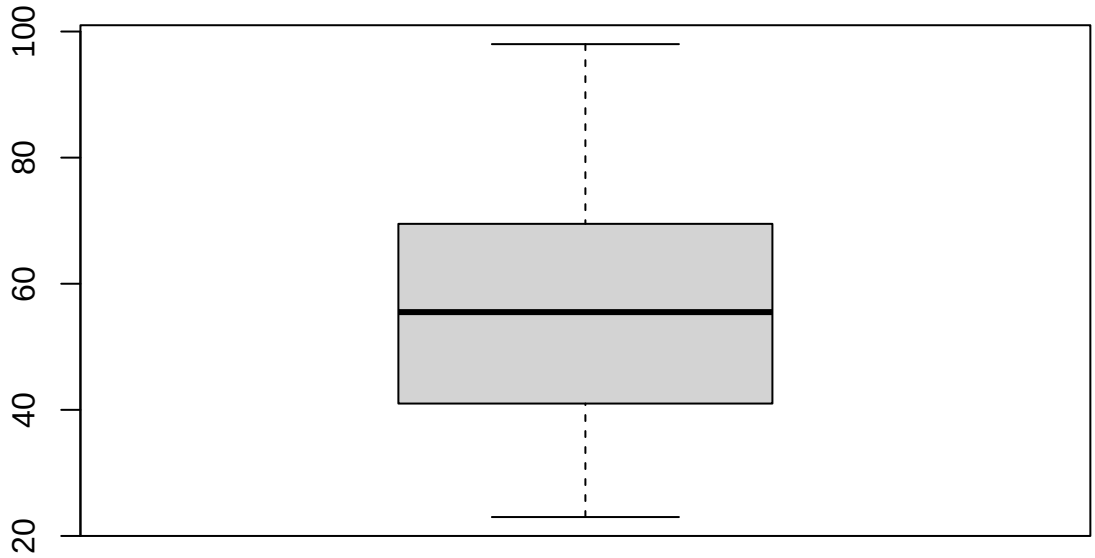
```
## # A tibble: 3 x 11
##      No Income Limit Rating Cards   Age Education Gender Student Married Balance
##   <dbl>  <dbl> <dbl>  <dbl> <dbl> <dbl>     <dbl> <chr>  <chr>   <chr>     <dbl>
## 1    56   32.9  1786    154     2   500         8 Female No      Yes           0
## 2    98   26.1  3388    266     4   500        17 Female No      Yes         155
## 3   130   18.1  3461    279     3   500        15 Male   No      Yes         255
```

remove rows/columns

```
# dataframe[-rowindex,]
```

```
# Remove data records with age >= 100
crdata <- crdata[-outliers,]
```

```r
# check the boxplot for Age column again
boxplot(crdata$Age)
```




```r
# Find the average income by excluding the missing values
income_avg <- mean(crdata$Income, na.rm=TRUE)

#print average income
print(income_avg)
```

```
## [1] 45.43253
```

```r
# Replace missing values in Income column with average income
crdata_replace <- replace_na(crdata, list(Income = income_avg))
```

```r
# Remove records with missing values and assign it to crdata_removed
crdata_removed <- na.omit(crdata_replace)

# Check the summary of the dataset again
summary(crdata_removed)
```

```
##        No             Income           Limit           Rating
##  Min.   :  1.0   Min.   : 10.35   Min.   :  855   Min.   : 93.0
##  1st Qu.:106.8   1st Qu.: 21.22   1st Qu.: 3098   1st Qu.:250.5
```

```
##  Median :206.5   Median : 33.68   Median : 4654   Median :344.0
##  Mean   :203.9   Mean   : 45.55   Mean   : 4760   Mean   :356.4
##  3rd Qu.:303.2   3rd Qu.: 58.04   3rd Qu.: 5886   3rd Qu.:438.2
##  Max.   :400.0   Max.   :186.63   Max.   :13913   Max.   :982.0
##      Cards            Age          Education        Gender
##  Min.   :1.000   Min.   :23.00   Min.   : 5.00   Length:388
##  1st Qu.:2.000   1st Qu.:41.00   1st Qu.:11.00   Class :character
##  Median :3.000   Median :56.00   Median :14.00   Mode  :character
##  Mean   :2.946   Mean   :55.66   Mean   :13.44
##  3rd Qu.:4.000   3rd Qu.:70.00   3rd Qu.:16.00
##  Max.   :9.000   Max.   :98.00   Max.   :20.00
##    Student            Married             Balance
##  Length:388         Length:388         Min.   :   0.0
##  Class :character   Class :character   1st Qu.:  79.5
##  Mode  :character   Mode  :character   Median : 466.0
##                                        Mean   : 525.2
##                                        3rd Qu.: 863.0
##                                        Max.   :1999.0
```

```
# Calculate the number of removed records
nrow(crdata) - nrow(crdata_removed)
```

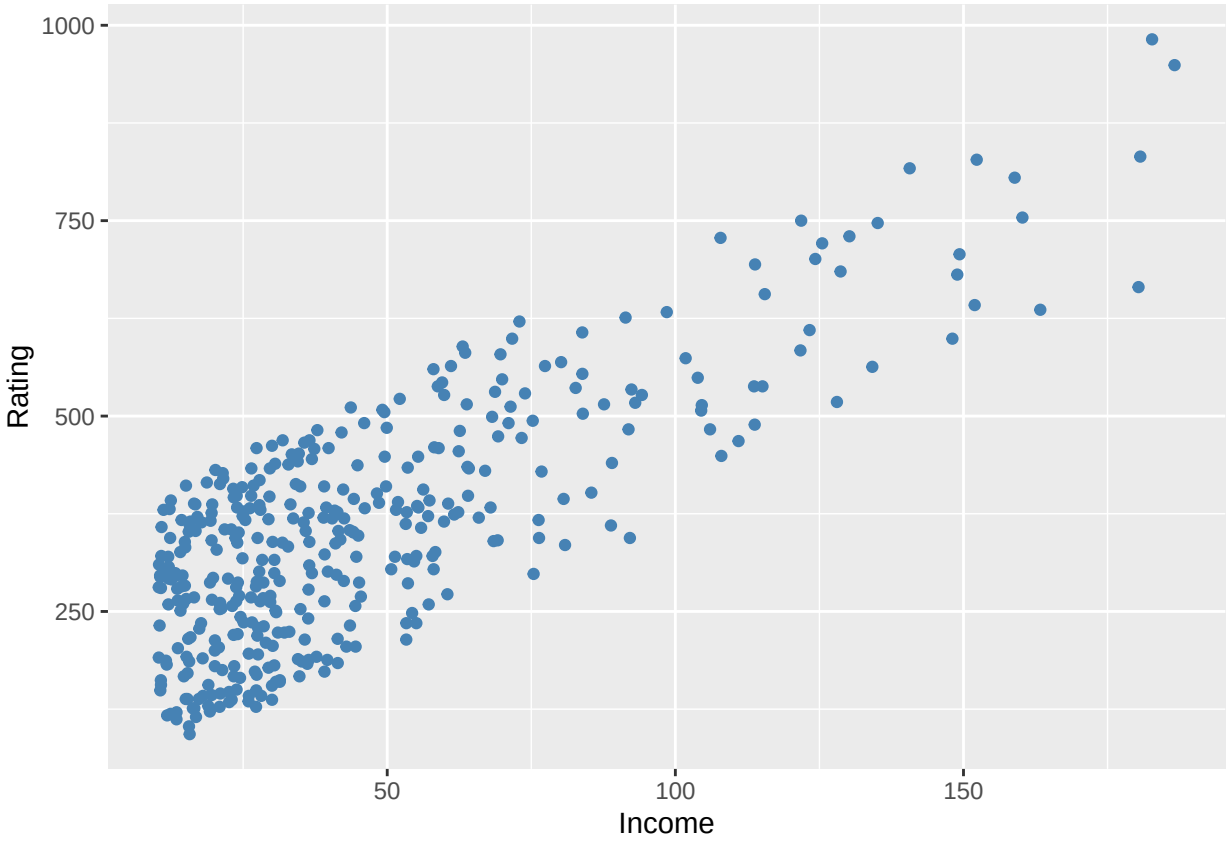
```
## [1] 9
```

Illustrate the relation between two features; "Income" and "Rating".

ggplot

```
# ggplot(data, aes(x, y)) + <geom_function>()
```

```
# Use geom_point(color = "steelblue") to add a scatter plot with blue colour
ggplot(crdata_removed, aes(x = Income, y = Rating)) + geom_point(color= "steelblue")
```


1000
750
500
250
Rating
50
100
150
Income