MATH2349 Data Wrangling

Assignment 2

Anirudhda Pardhi - s3807109

Required packages

Importing required libraries.

```
library(readr)
library(Hmisc)
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(plyr)
```

```
##
## Attaching package: 'plyr'
```

```
## The following objects are masked from 'package:Hmisc':
##
##     is.discrete, summarize
```

```
library(tidyr)
library(magrittr)
```

```
##
## Attaching package: 'magrittr'
```

```
## The following object is masked from 'package:tidyr':
##
##     extract
```

```
library(dplyr)
```

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

```
## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize
```

```
## The following objects are masked from 'package:Hmisc':
##
##     src, summarize
```

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

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

```
library(ggplot2)
library(forecast)
```

```
## Registered S3 method overwritten by 'quantmod':
##   method            from
##   as.zoo.data.frame zoo
```

Executive Summary

In this assignment, 2 datasets were used and imported in R. Dataset 1 name is “Cardio_train” which contain details cardiovascular diseases related to heart . Attributes of this dataset are id, age,gender,height, weight,ap_hi,ap_lo,cholesterol,gluc,smoke,alco,active,cardio. This dataset is highly untidy in which all the column names are present in single column separated by comma(,) and attributes respective values are also present in single cell separated by comma(,). Dataset 2 name is “heart” and is related to details about a person’s details related to health of heart.

These 2 datasets are merged using full join method using Primary key as variable “Age” which was common in both the dataset. For better readability purpose few of the columns were renamed. After merging 2 dataframes appropriate conversions were applied to confirm if all the variables or attributes are in correct datatype. There were few character variables which consists of numerical values were converted into numerical datatype. All the

categorical variables which were of character datatype were converted into factors. In order to Tidy the dataset “cardio_train” separate function was used to separate the comma seperated attribute values. Because of full join NA values were introduced which was handled by using Impute() function. All the numerical variables were replaced by mean and categorical values were replaced by mode method.

Mutation was performed on height and weight variables and “bmi_index” was calculated using formula (weight)/(height*height). Outliers were detected on numerical variables using boxplot and were handheld by Capping methods in which outliers are replaced with nearest neighbor.

Transformation was applied on numerical variable “weight” and “bmi_index” which earlier had skewness. Right skewness was corrected using logarithmic and boxcox transformation method.

Data

Source of heart dataset (1) - https://www.kaggle.com/ronitf/heart-disease-uci (https://www.kaggle.com/ronitf/heart-disease-uci) Source of Cardio_train dataset (2)- https://www.kaggle.com/sulianova/cardiovascular-disease-dataset (https://www.kaggle.com/sulianova/cardiovascular-disease-dataset)

Dataset 1- Heart dataset attribute information : 1. age 2. sex 3. chest pain type (4 values) 4. resting blood pressure 5. serum cholestoral in mg/dl 6. fasting blood sugar > 120 mg/dl 7. resting electrocardiographic results (values 0,1,2) 8. maximum heart rate achieved 9. exercise induced angina 10. oldpeak = ST depression induced by exercise relative to rest 11. the slope of the peak exercise ST segment 12. number of major vessels (0-3) colored by flourosopy 13. thal: 3 = normal; 6 = fixed defect; 7 = reversable defect

Dataset 2- Cardiovasular disease dataset attribute information : Age | Objective Feature | age | int (days) Height | Objective Feature | height | int (cm) | Weight | Objective Feature | weight | float (kg) | Gender | Objective Feature | gender | categorical code | Systolic blood pressure | Examination Feature | ap_hi | int | Diastolic blood pressure | Examination Feature | ap_lo | int | Cholesterol | Examination Feature | cholesterol | 1: normal, 2: above normal, 3: well above normal | Glucose | Examination Feature | gluc | 1: normal, 2: above normal, 3: well above normal | Smoking | Subjective Feature | smoke | binary | Alcohol intake | Subjective Feature | alco | binary | Physical activity | Subjective Feature | active | binary | Presence or absence of cardiovascular disease | Target Variable | cardio | binary |

These 2 datasets are merged using full join method using Primary key as variable “Age” which was common in both the dataset.

```
heart <- read_csv("/Users/ADMIN/Desktop/Sem 2/Data wrangling/Asg 2/Datasets/heart.csv")
```

```
## Parsed with column specification:
## cols(
##   age = col_double(),
##   sex = col_double(),
##   cp = col_double(),
##   trestbps = col_double(),
##   chol = col_double(),
##   fbs = col_double(),
##   restecg = col_double(),
##   thalach = col_double(),
##   exang = col_double(),
##   oldpeak = col_double(),
##   slope = col_double(),
##   ca = col_double(),
##   thal = col_double(),
##   target = col_double()
## )
```

```
cardio_train <- read_csv("/Users/ADMIN/Desktop/Sem 2/Data wrangling/Asg 2/Datasets/cardio_train.csv")
```

```
## Parsed with column specification:
## cols(
##   `id,age,gender,height,weight,ap_hi,ap_lo,cholesterol,gluc,smoke,alco,active,cardio` = col_character()
## )
```

```
head(heart)
```

age <dbl>	sex <dbl>	cp <dbl>	trestbps <dbl>	chol <dbl>	fbs <dbl>	restecg <dbl>	thalach <dbl>	exang <dbl>	oldpeak <dbl>
63	1	3	145	233	1	0	150	0	2.3
37	1	2	130	250	0	1	187	0	3.5
41	0	1	130	204	0	0	172	0	1.4
56	1	1	120	236	0	1	178	0	0.8
57	0	0	120	354	0	1	163	1	0.6
57	1	0	140	192	0	1	148	0	0.4

6 rows | 1-10 of 14 columns

```
str (heart)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 303 obs. of  14 variables:
##  $ age     : num  63 37 41 56 57 57 56 44 52 57 ...
##  $ sex     : num  1 1 0 1 0 1 0 1 1 1 ...
##  $ cp      : num  3 2 1 1 0 0 1 1 2 2 ...
##  $ trestbps: num  145 130 130 120 120 140 140 120 172 150 ...
##  $ chol    : num  233 250 204 236 354 192 294 263 199 168 ...
##  $ fbs     : num  1 0 0 0 0 0 0 0 1 0 ...
##  $ restecg : num  0 1 0 1 1 1 0 1 1 1 ...
##  $ thalach : num  150 187 172 178 163 148 153 173 162 174 ...
##  $ exang   : num  0 0 0 0 1 0 0 0 0 0 ...
##  $ oldpeak : num  2.3 3.5 1.4 0.8 0.6 0.4 1.3 0 0.5 1.6 ...
##  $ slope   : num  0 0 2 2 2 1 1 2 2 2 ...
##  $ ca      : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ thal    : num  1 2 2 2 2 1 2 3 3 2 ...
##  $ target  : num  1 1 1 1 1 1 1 1 1 1 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   age = col_double(),
##   ..   sex = col_double(),
##   ..   cp = col_double(),
##   ..   trestbps = col_double(),
##   ..   chol = col_double(),
##   ..   fbs = col_double(),
##   ..   restecg = col_double(),
##   ..   thalach = col_double(),
##   ..   exang = col_double(),
##   ..   oldpeak = col_double(),
##   ..   slope = col_double(),
##   ..   ca = col_double(),
##   ..   thal = col_double(),
##   ..   target = col_double()
##   .. )
```

```
head(cardio_train)
```

id,age,gender,height,weight,ap_hi,ap_lo,cholesterol,gluc,smoke,alco,active,cardio <chr>
0,18393,2,168,62.0,110,80,1,1,0,0,1,0
1,20228,1,156,85.0,140,90,3,1,0,0,1,1
2,18857,1,165,64.0,130,70,3,1,0,0,0,1
3,17623,2,169,82.0,150,100,1,1,0,0,1,1
4,17474,1,156,56.0,100,60,1,1,0,0,0,0
8,21914,1,151,67.0,120,80,2,2,0,0,0,0

6 rows

```
str(cardio_train)
```

```
## Classes 'spec_tbl_df', 'tbl_df', 'tbl' and 'data.frame': 70000 obs. of  1 variable:
##  $ id,age,gender,height,weight,ap_hi,ap_lo,cholesterol,gluc,smoke,alco,active,cardio: chr
"0,18393,2,168,62.0,110,80,1,1,0,0,1,0" "1,20228,1,156,85.0,140,90,3,1,0,0,1,1" "2,18857,1,165,6
4.0,130,70,3,1,0,0,0,1" "3,17623,2,169,82.0,150,100,1,1,0,0,1,1" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   `id,age,gender,height,weight,ap_hi,ap_lo,cholesterol,gluc,smoke,alco,active,cardio` =
col_character()
##   .. )
```

Merging and understanding

In cardio_train dataset since age was in number of days so we had to convert age in years by dividing age in days with 365.(age in years= age in days/365). Full join is applied to merge the heart and cardio_train dataset using “Age” variable as primary key

```
colnames(cardio_train) <-c("multiplenames")

testc <- cardio_train %>% separate(multiplenames,
                                   into = c("id", "age","gender","height","weight",
                                            "ap_hi","ap_lo","cholesterol","gluc","smoke","alco",
"active","cardio"), sep = ",")
head(testc)
```

id <chr>	age <chr>	gender <chr>	height <chr>	weight <chr>	ap_hi <chr>	ap_lo <chr>	cholesterol <chr>	gluc <chr>	smoke <chr>
0	18393	2	168	62.0	110	80	1	1	0
1	20228	1	156	85.0	140	90	3	1	0
2	18857	1	165	64.0	130	70	3	1	0
3	17623	2	169	82.0	150	100	1	1	0
4	17474	1	156	56.0	100	60	1	1	0
8	21914	1	151	67.0	120	80	2	2	0

6 rows | 1-10 of 13 columns

```
#Converting age of cardio dataset from days to years
testc$age <- as.numeric(as.character(testc$age))
testd <- mutate(testc, age = age/365)
head(testd)
```

id <chr>	age <dbl>	gender <chr>	height <chr>	weight <chr>	ap_hi <chr>	ap_lo <chr>	cholesterol <chr>	gluc <chr>	smoke <chr>

id <chr>	age <dbl>	gender <chr>	height <chr>	weight <chr>	ap_hi <chr>	ap_lo <chr>	cholesterol <chr>	gluc <chr>	smoke <chr>
0	50.39178	2	168	62.0	110	80	1	1	0
1	55.41918	1	156	85.0	140	90	3	1	0
2	51.66301	1	165	64.0	130	70	3	1	0
3	48.28219	2	169	82.0	150	100	1	1	0
4	47.87397	1	156	56.0	100	60	1	1	0
8	60.03836	1	151	67.0	120	80	2	2	0

6 rows | 1-10 of 13 columns

```
testd_subset<-testd[1:300,]
test1 <- full_join(testd_subset,heart,by = "age")
head(test1)
```

id <chr>	age <dbl>	gender <chr>	height <chr>	weight <chr>	ap_hi <chr>	ap_lo <chr>	cholesterol <chr>	gluc <chr>	smoke <chr>
0	50.39178	2	168	62.0	110	80	1	1	0
1	55.41918	1	156	85.0	140	90	3	1	0
2	51.66301	1	165	64.0	130	70	3	1	0
3	48.28219	2	169	82.0	150	100	1	1	0
4	47.87397	1	156	56.0	100	60	1	1	0
8	60.03836	1	151	67.0	120	80	2	2	0

6 rows | 1-10 of 26 columns

Tidy & Manipulate Data I (Heart dataset)

In Heart dataset there were lots of character variables which had numeric values but from readility point of view data was not clear so using Mapvalue function we had assigned to numbers to some meaningful names. Catagorical variables were converted into factors.

```
test1$gender<-mapvalues(test1$gender, from = c("1","2"), to = c("male","female"))
test1$gender <- as.factor(test1$gender)
levels(test1$gender)
```

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

```
test1$exang<-mapvalues(test1$exang, from = c("0", "1"), to = c("No", "Yes"))
test1$exang <- as.factor(test1$exang)
levels(test1$exang)
```

```
## [1] "No"  "Yes"
```

```
test1$fbs<-mapvalues(test1$fbs, from = c("0", "1"), to = c("False", "True"))
test1$fbs <- as.factor(test1$fbs)
levels(test1$fbs)
```

```
## [1] "False" "True"
```

```
test1$cp<-mapvalues(test1$cp, from = c("0", "1","2","3"), to = c("Typical Angina",
                                                                 "Atypical Angina",
                                                                 "Non-anginal Pain",
                                                                 "Asymptomatic"))
test1$cp <- as.factor(test1$cp)
levels(test1$cp)
```

```
## [1] "Asymptomatic"     "Atypical Angina"  "Non-anginal Pain" "Typical Angina"
```

```
test1$restecg<-mapvalues(test1$restecg, from = c("0", "1","2"), to = c("normal",
                                                                       "ST-T wave abnormality( d
epression of > 0.05 mV)",
                                                                       "definite left ventricula
r hypertrophy by Estes' criteria"))
test1$restecg <- as.factor(test1$restecg)
levels(test1$restecg)
```

```
## [1] "definite left ventricular hypertrophy by Estes' criteria"
## [2] "normal"
## [3] "ST-T wave abnormality( depression of > 0.05 mV)"
```

```
test1$slope<-mapvalues(test1$slope, from = c("0", "1","2"), to = c("upwordslope",
                                                                   "noslope",
                                                                   "downwordslope"))
test1$slope <- as.factor(test1$slope)
levels(test1$slope)
```

```
## [1] "downwordslope" "noslope"       "upwordslope"
```

Tidy & Manipulate Data II (Cardio_train dataset)

Cardio_train dataset is highly untidy in which all the column names are present in single column separated by comma(,) and attributes respective values are also present in single cell separated by comma(,). In order to Tidy this dataset SEPERATE() function was used to separate the comma seperated attribute values which is done just before merging of datasets in above section.Character variables which consists of numerical values were converted into numerical datatype. All the categorical variables which were of character datatype were converted into factors.For datatype conversion and factorization below is code :-

```
test1$cholesterol <-mapvalues(test1$cholesterol , from = c("1", "2","3"), to = c("normal","above
normal",
                                                                                  "well above nor
mal"))
test1$cholesterol <- as.factor(test1$cholesterol)
levels(test1$cholesterol)
```

```
## [1] "above normal"      "normal"            "well above normal"
```

```
test1$gluc <-mapvalues(test1$gluc, from = c("1", "2","3"), to = c("normal","above normal",
                                                                  "well above normal"))
test1$gluc <- as.factor(test1$gluc)
levels(test1$gluc)
```

```
## [1] "above normal"      "normal"            "well above normal"
```

```
test1$smoke <-mapvalues(test1$smoke, from = c("0", "1"), to = c("no","yes"))
test1$smoke <- as.factor(test1$smoke)
levels(test1$smoke)
```

```
## [1] "no"  "yes"
```

```
test1$alco <-mapvalues(test1$alco, from = c("0", "1"), to = c("no","yes"))
test1$alco <- as.factor(test1$alco)
levels(test1$alco)
```

```
## [1] "no"  "yes"
```

```
test1$active <-mapvalues(test1$active, from = c("0", "1"), to = c("no","yes"))
test1$active <- as.factor(test1$active)
levels(test1$active)
```

```
## [1] "no"  "yes"
```

```
test1$cardio <-mapvalues(test1$cardio, from = c("0", "1"), to = c("no","yes"))
test1$cardio <- as.factor(test1$cardio)
levels(test1$cardio)
```

```
## [1] "no"  "yes"
```

```
test1$thal <-mapvalues(test1$thal, from = c("0", "1","2","3"), to = c("normal","fixed defect",
                                                                       "reversable defect","sever
e defect"))
test1$thal <- as.factor(test1$thal)
levels(test1$thal)
```

```
## [1] "fixed defect"      "normal"            "reversable defect"
## [4] "severe defect"
```

```
test1$target <-mapvalues(test1$target, from = c("0", "1"), to = c("less chance of heart attack",
                                                                  "more chance of heart attack"
))
test1$target <- as.factor(test1$target)
levels(test1$target)
```

```
## [1] "less chance of heart attack" "more chance of heart attack"
```

```
test1$ca <-mapvalues(test1$ca, from = c("0", "1","2","3","4"), to = c("low","medium","large",
                                                                      "XL","XXL"))
test1$ca <- as.factor(test1$ca)
levels(test1$ca)
```

```
## [1] "large"  "low"    "medium" "XL"     "XXL"
```

```
head(test1)
```

id <chr>	age <dbl>	gen… <fctr>	height <chr>	weight <chr>	ap… <chr>	ap… <chr>	cholesterol <fctr>	gluc <fctr>	sm… <fctr>
0	50.39178	female	168	62.0	110	80	normal	normal	no
1	55.41918	male	156	85.0	140	90	well above normal	normal	no
2	51.66301	male	165	64.0	130	70	well above normal	normal	no
3	48.28219	female	169	82.0	150	100	normal	normal	no
4	47.87397	male	156	56.0	100	60	normal	normal	no
8	60.03836	male	151	67.0	120	80	above normal	above normal	no

6 rows | 1-10 of 26 columns

```
#conversion into numeric datatype
test1$height <- as.numeric(test1$height)
test1$weight <- as.numeric(test1$weight)
test1$ap_lo <- as.numeric(test1$ap_lo)
test1$ap_hi <- as.numeric(test1$ap_hi)

#Selecting columns from dataset
bmi_sub<- test1 %>% select(age,gender,height, weight,ap_hi,smoke,alco,cholesterol,cp,trestbps,ta
rget)
head(bmi_sub)
```

age <dbl>	gender <fctr>	height <dbl>	weight <dbl>	ap_hi <dbl>	sm… <fctr>	alco <fctr>	cholesterol <fctr>	cp <fctr>	trestbps <dbl>
50.39178	female	168	62	110	no	no	normal	*NA*	*NA*
55.41918	male	156	85	140	no	no	well above normal	*NA*	*NA*
51.66301	male	165	64	130	no	no	well above normal	*NA*	*NA*
48.28219	female	169	82	150	no	no	normal	*NA*	*NA*
47.87397	male	156	56	100	no	no	normal	*NA*	*NA*
60.03836	male	151	67	120	no	no	above normal	*NA*	*NA*

6 rows | 1-10 of 11 columns

Scan I (Handling Missing values)

There was a significant number of NA values in the merged dataset so variables with numerical datatype were replaced with Mean method and catagorical variables were replaced with Mode variables by using Impute() fn.

```
#checking and handling missing values
sapply(bmi_sub, function(x) sum( is.na(x) ))
```

```
##         age      gender      height      weight       ap_hi       smoke
##           0         281         281         281         281         281
##        alco cholesterol          cp    trestbps      target
##         281         281         298         298         298
```

```
bmi_sub$height <- impute(bmi_sub$height, fun = mean)
bmi_sub$gender <- impute(bmi_sub$gender, fun = mode)
bmi_sub$weight <- impute(bmi_sub$weight, fun = mean)
bmi_sub$ap_hi <- impute(bmi_sub$ap_hi, fun = mean)
bmi_sub$ap_lo <- impute(bmi_sub$ap_lo, fun = mean)
```

```
## Warning: Unknown or uninitialised column: 'ap_lo'.
```

```
bmi_sub$cholesterol <- impute(bmi_sub$cholesterol, fun = mode)
bmi_sub$smoke <- impute(bmi_sub$smoke, fun = mode)
bmi_sub$alco <- impute(bmi_sub$alco, fun = mode)
bmi_sub$cp <- impute(bmi_sub$cp, fun = mode)
bmi_sub$trestbps <- impute(bmi_sub$trestbps, fun = mean)
bmi_sub$target <- impute(bmi_sub$target, fun = mode)
sapply(bmi_sub, function(x) sum( is.na(x) ))
```

```
##         age      gender      height      weight       ap_hi       smoke
##           0           0           0           0           0           0
##        alco cholesterol          cp    trestbps      target
##           0           0           0           0           0
```

```
head(bmi_sub)
```

age <dbl>	gen… <fctr>	height <dbl>	wei… <dbl>	ap… <dbl>	sm… <fctr>	al… <fctr>	cholesterol <fctr>	cp <S3: impute>	trest < impu
50.39178	female	168	62	110	no	no	normal	Typical Angina	131.6
55.41918	male	156	85	140	no	no	well above normal	Typical Angina	131.6
51.66301	male	165	64	130	no	no	well above normal	Typical Angina	131.6
48.28219	female	169	82	150	no	no	normal	Typical Angina	131.6
47.87397	male	156	56	100	no	no	normal	Typical Angina	131.6
60.03836	male	151	67	120	no	no	above normal	Typical Angina	131.6

6 rows | 1-10 of 11 columns

Mutation

Mutation was performed on height and weight variables and “bmi_index” was created by using formula (weight)/(height*height).

Unit of weight- Kilograms Unit of height- Meter

```
bmi_sub1 <- mutate(bmi_sub ,hgt_mtr = height / 100)
bmi_sub2 <- mutate(bmi_sub1 ,bmi_index = weight / (hgt_mtr * hgt_mtr))
bmi_sub3 <- bmi_sub2[1:300,]
head(bmi_sub3)
```

age <dbl>	gen… <fctr>	height <dbl>	wei… <dbl>	ap… <dbl>	sm… <fctr>	al… <fctr>	cholesterol <fctr>	cp <S3: impute>	trest < impu
50.39178	female	168	62	110	no	no	normal	Typical Angina	131.6

age	gen…	height	wei…	ap…	sm…	al…	cholesterol	cp	trest
<dbl>	<fctr>	<dbl>	<dbl>	<dbl>	<fctr>	<fctr>	<fctr>	<S3: impute>	<impu
55.41918	male	156	85	140	no	no	well above normal	Typical Angina	131.6
51.66301	male	165	64	130	no	no	well above normal	Typical Angina	131.6
48.28219	female	169	82	150	no	no	normal	Typical Angina	131.6
47.87397	male	156	56	100	no	no	normal	Typical Angina	131.6
60.03836	male	151	67	120	no	no	above normal	Typical Angina	131.6

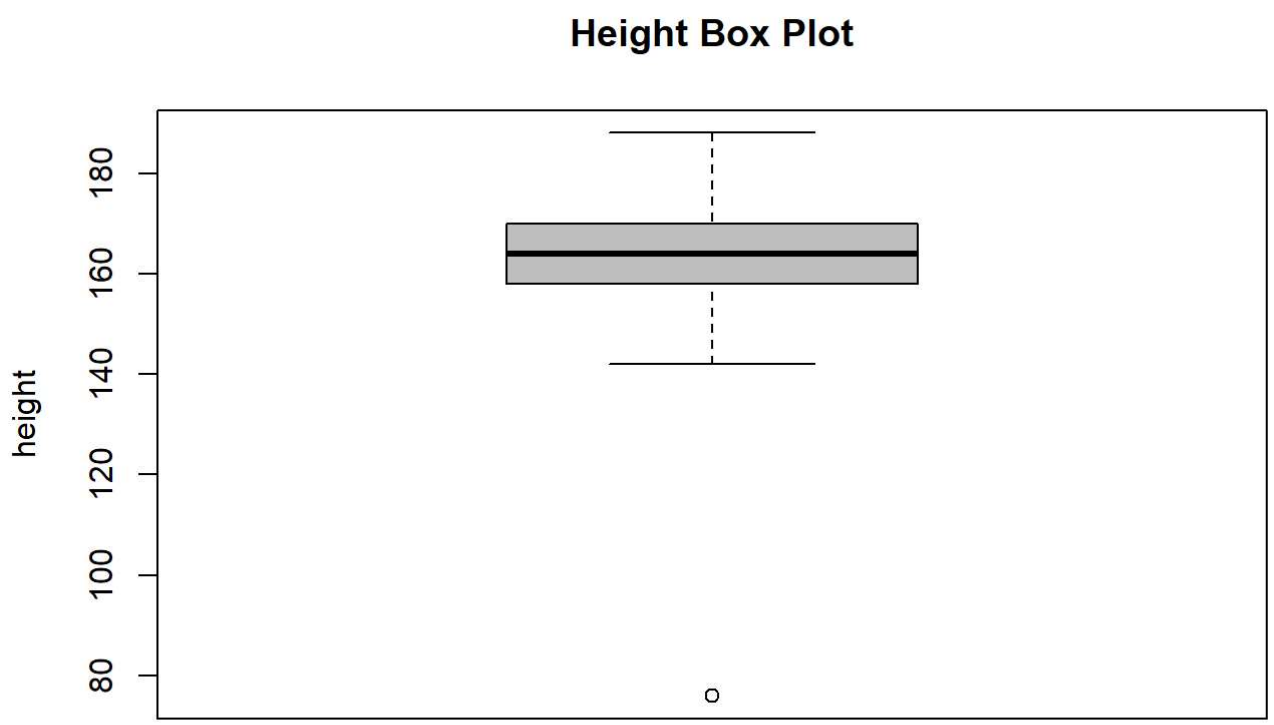
6 rows | 1-10 of 13 columns

Scan II (Outlier detection and handling)

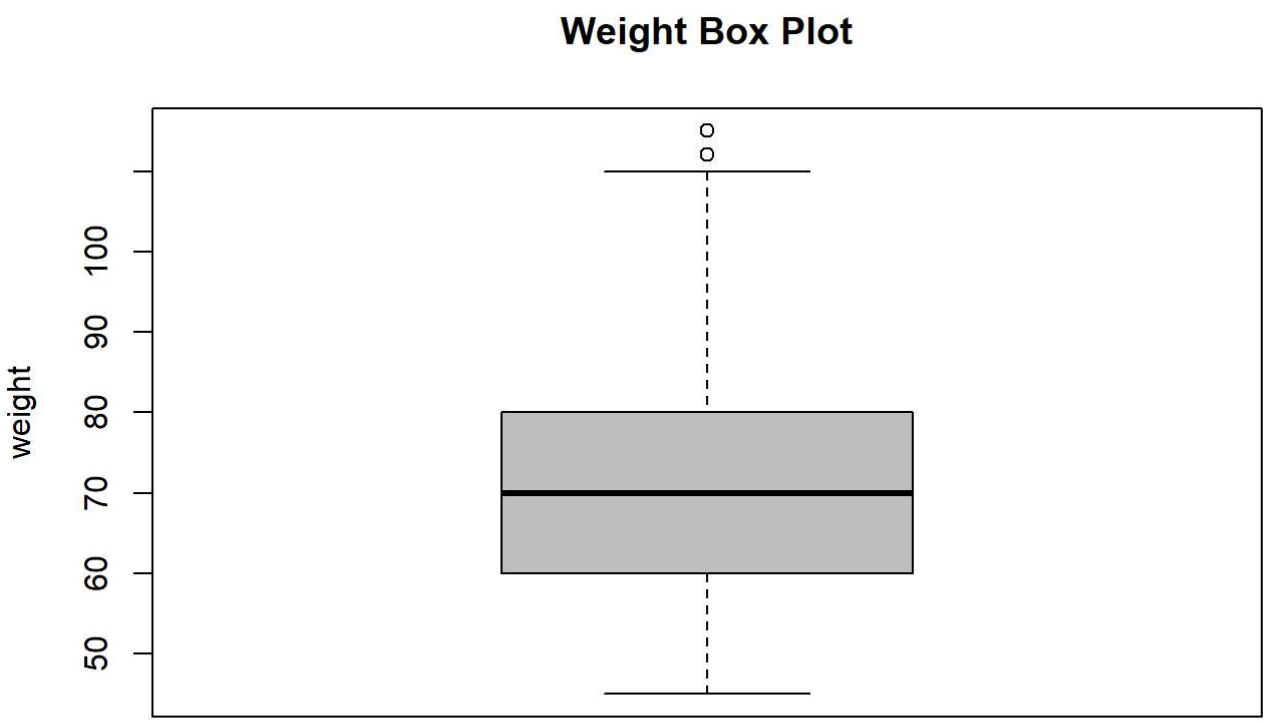
Scan the numeric data for outliers. In this step, you should fulfil the minimum requirement #8. In addition to the R codes and outputs, explain your methodology (i.e. explain why you have chosen that methodology and the actions that you have taken to handle these values) and communicate your results clearly.

Outliers for numeric variables was calculated using one of the method of detecting univariate detection outlier which is boxplot.There were few outliers detected in each of the numerical variables used below but since the value of outliers were non-negative so i preferred it corrected by using Capping method in which we assign the outliers to the closest fences of boxplot within range of 5 to 95 percentile.

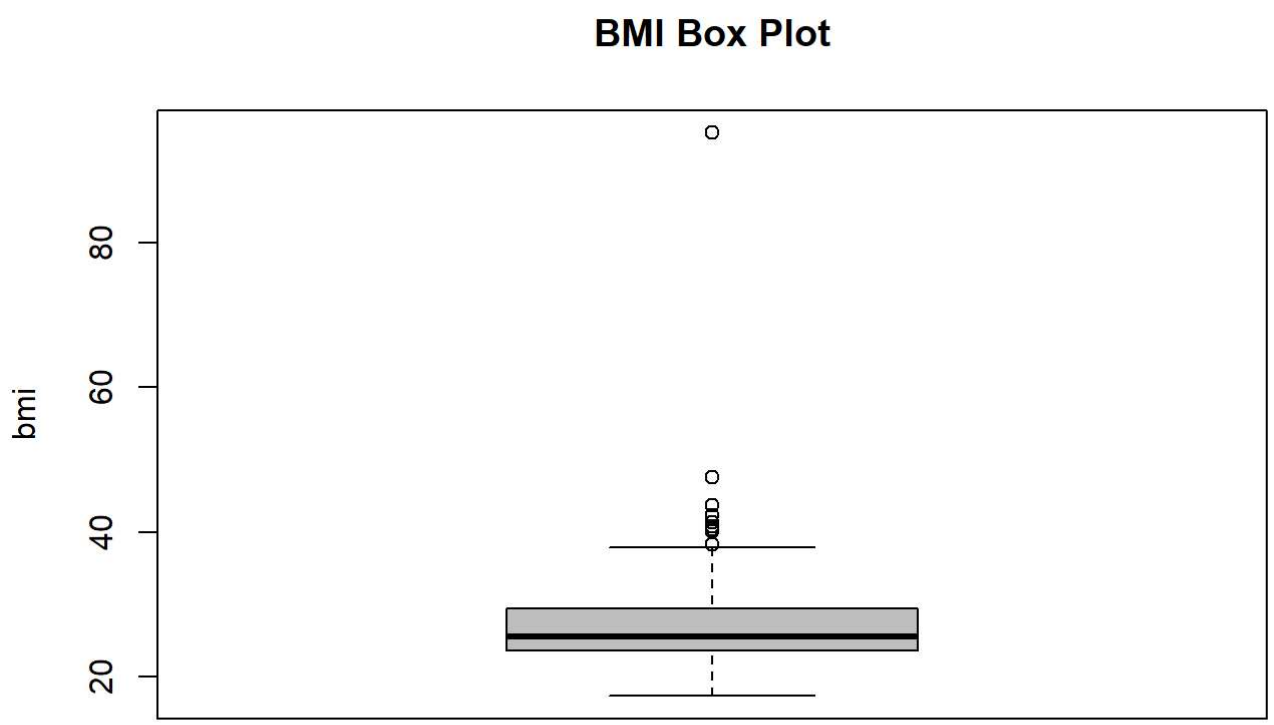
```
bmi_sub3$height %>%  boxplot(main="Height Box Plot", ylab="height", col = "grey")
```



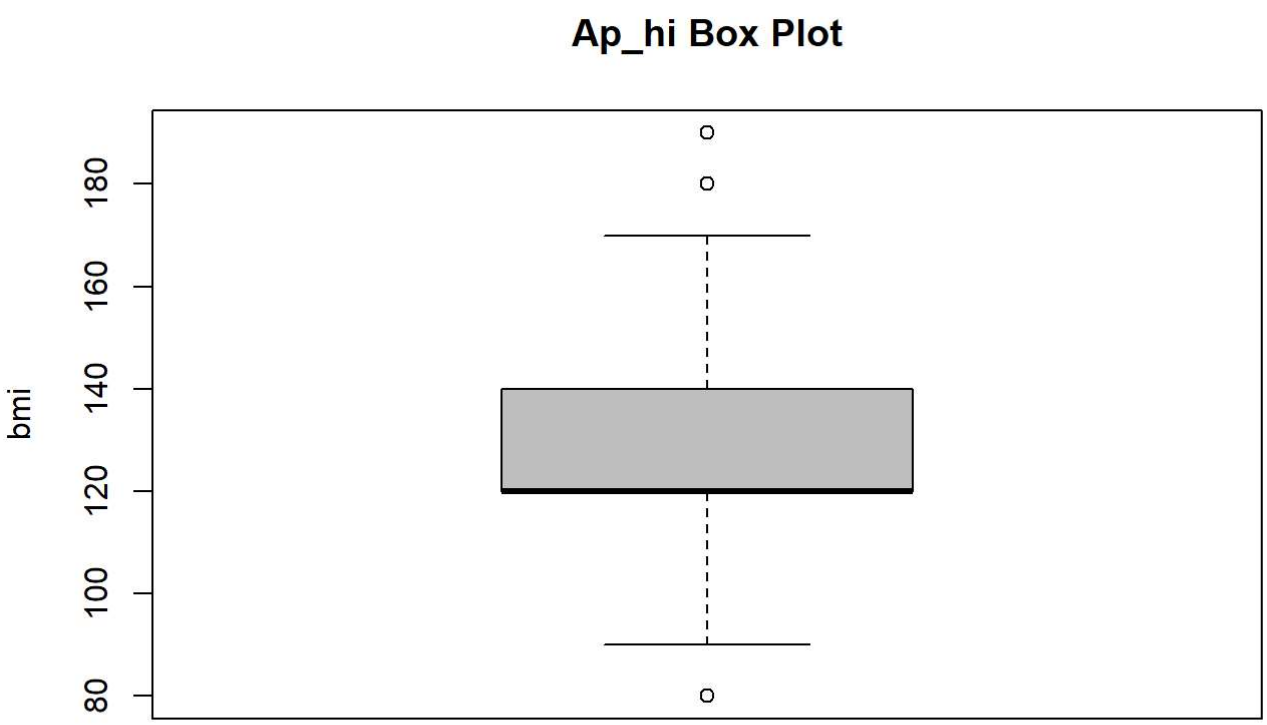

```
bmi_sub3$weight %>%  boxplot(main="Weight Box Plot", ylab="weight", col = "grey")
```



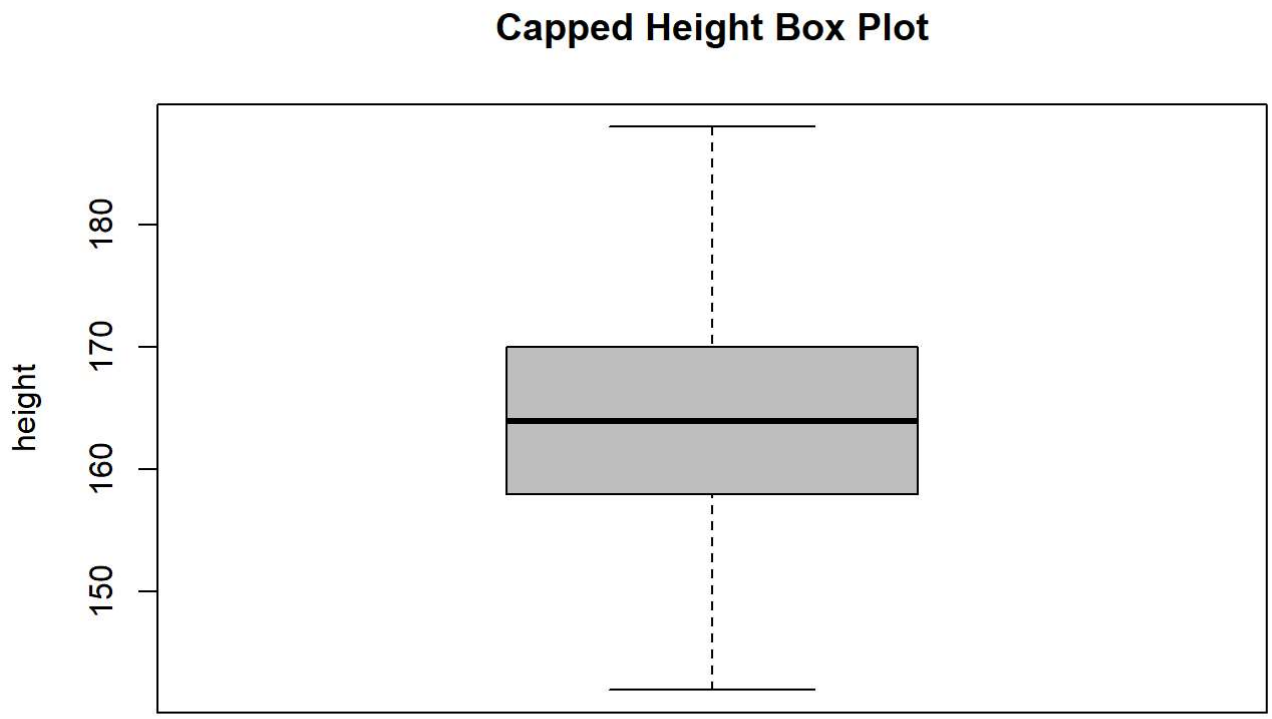

```
bmi_sub3$bmi_index %>%  boxplot(main="BMI Box Plot", ylab="bmi", col = "grey")
```



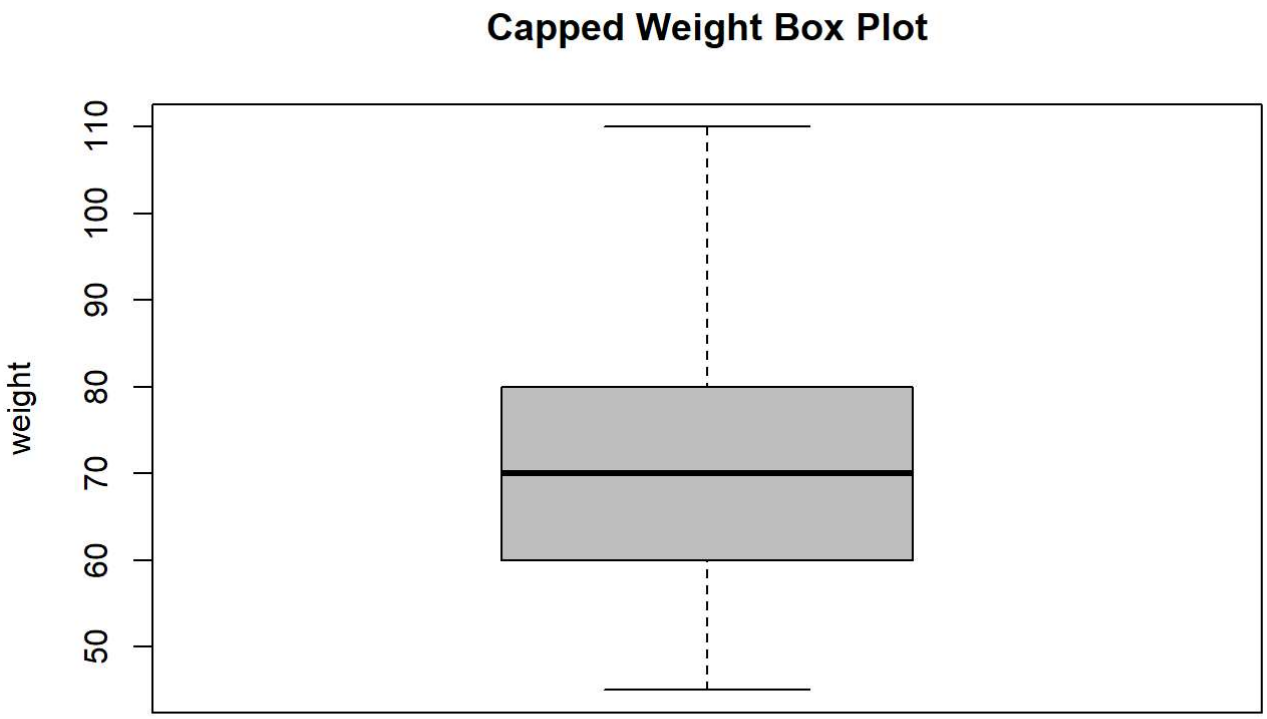

```
bmi_sub3$ap_hi %>%  boxplot(main="Ap_hi Box Plot", ylab="bmi", col = "grey")
```



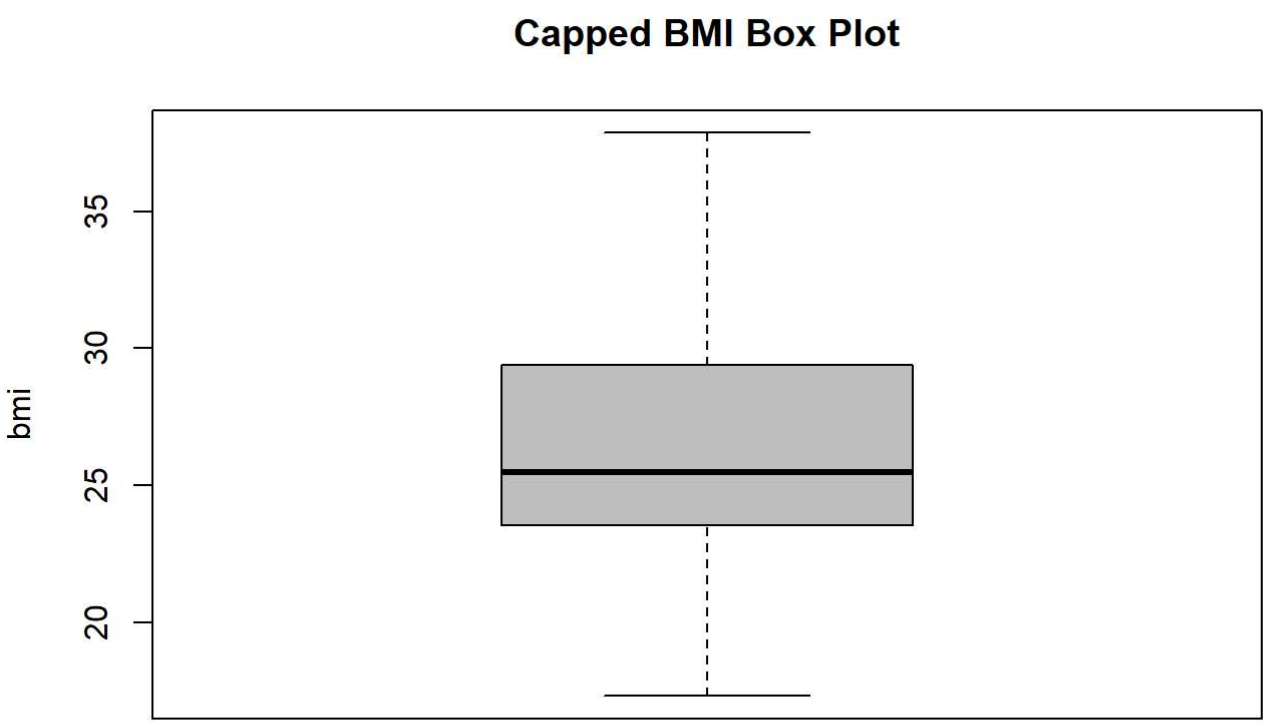

```
#Capping of numerical variables

cap <- function(x){
  quantiles <- quantile( x, c(.05, 0.25, 0.75, .95 ) )
  x[ x < quantiles[2] - 1.5*IQR(x) ] <- quantiles[1]
  x[ x > quantiles[3] + 1.5*IQR(x) ] <- quantiles[4]
  x }

height_capped <- bmi_sub3$height %>% cap()
height_capped %>%  boxplot(main="Capped Height Box Plot", ylab="height", col = "grey")
```



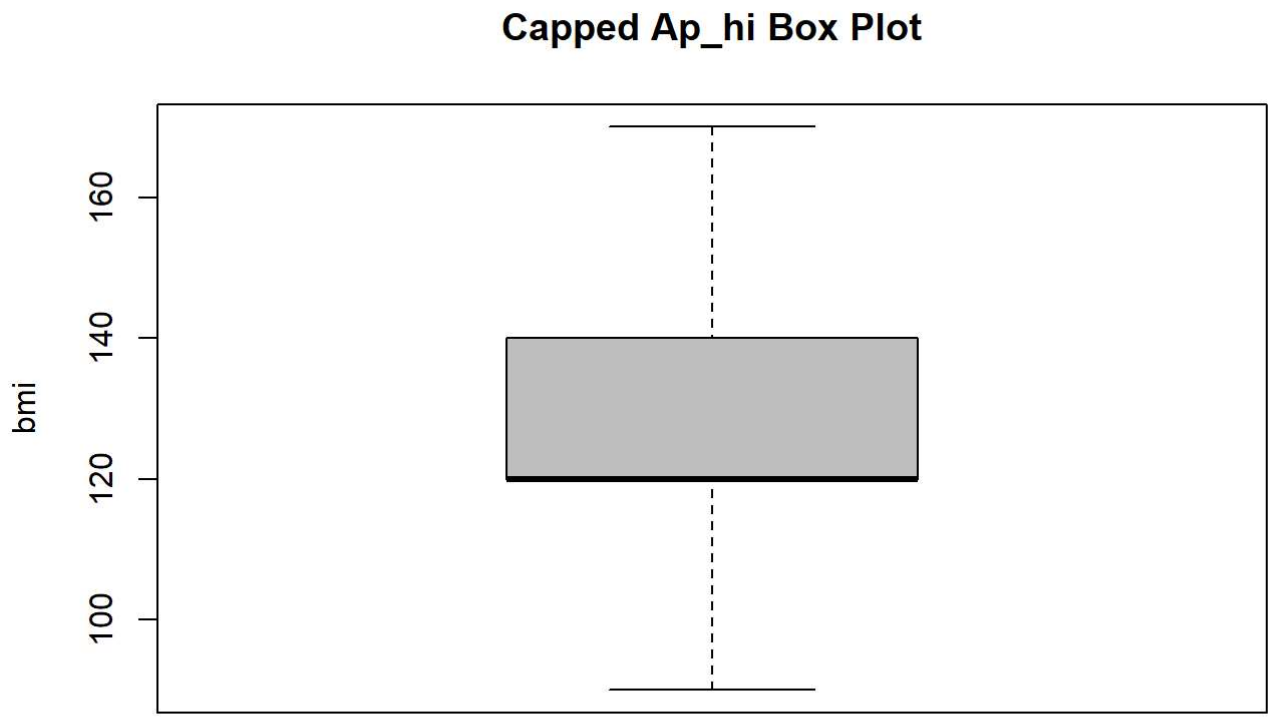

```
weight_capped <- bmi_sub3$weight %>% cap()
weight_capped %>%  boxplot(main="Capped Weight Box Plot", ylab="weight", col = "grey")
```




```
bmi_capped <- bmi_sub3$bmi_index %>% cap()
bmi_capped %>%  boxplot(main="Capped BMI Box Plot", ylab="bmi", col = "grey")
```



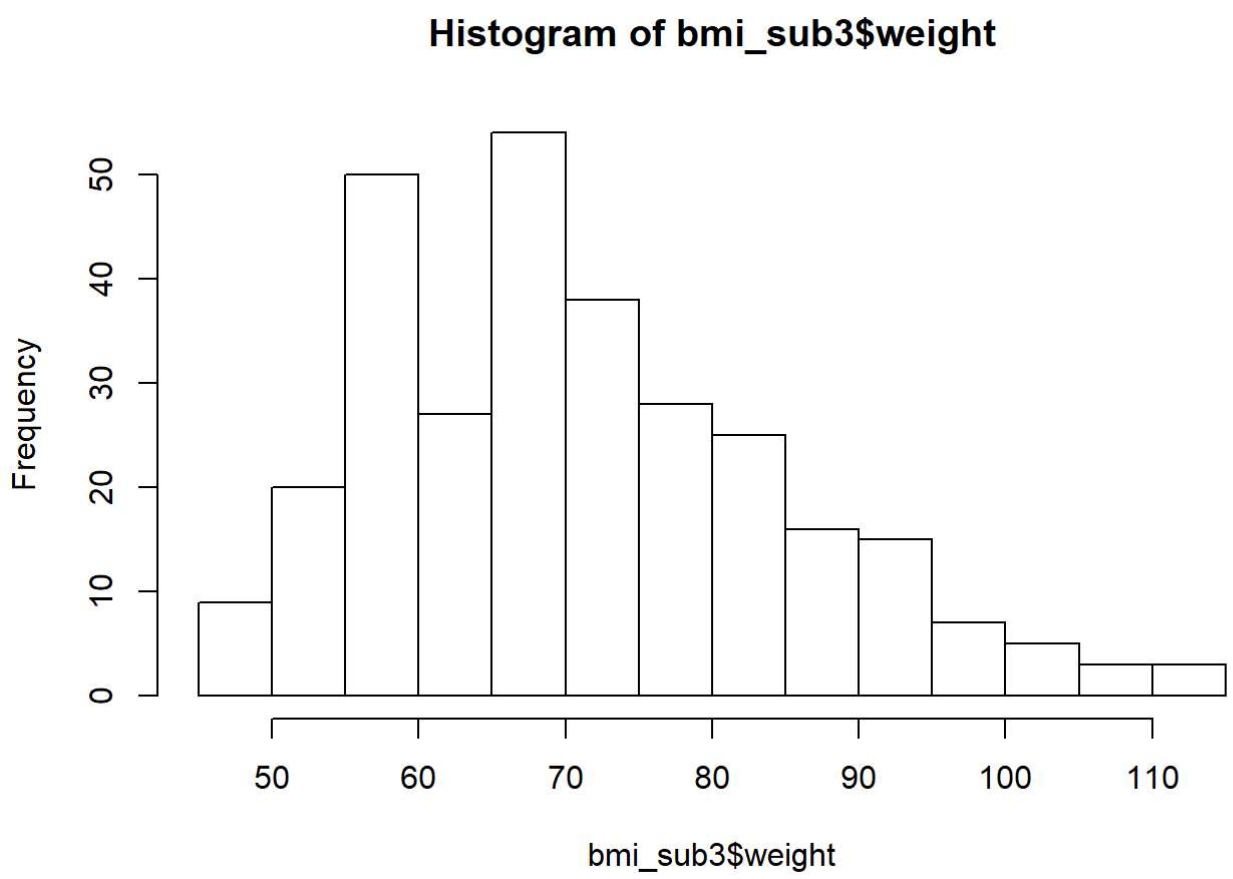

```
ap_hi_capped <- bmi_sub3$ap_hi %>% cap()
ap_hi_capped %>%  boxplot(main="Capped Ap_hi Box Plot", ylab="bmi", col = "grey")
```



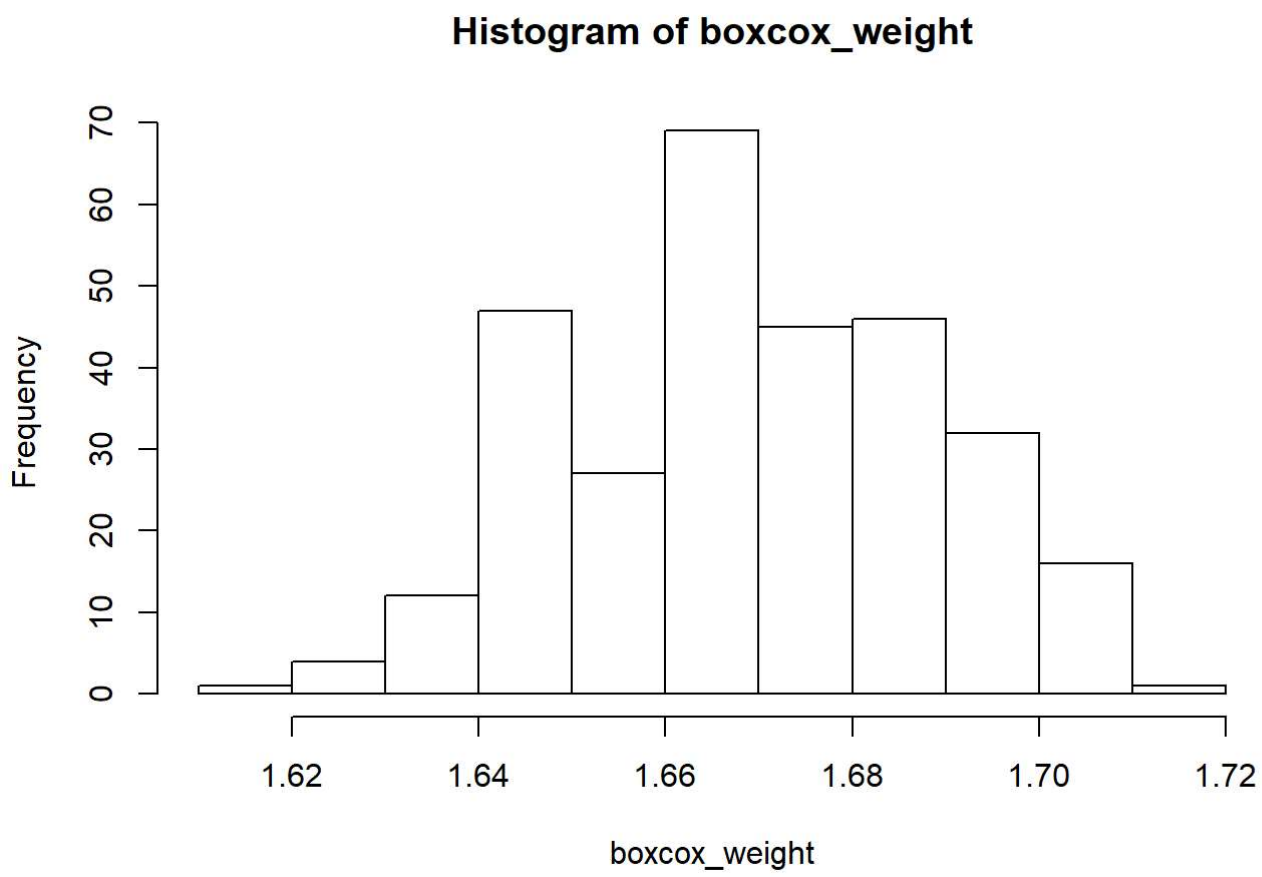
Transform

Transformation was applied on numerical variable “weight” and “bmi_index” which earlier had right skewness. Right skewness was corrected using logarithmic and boxcox transformation method.

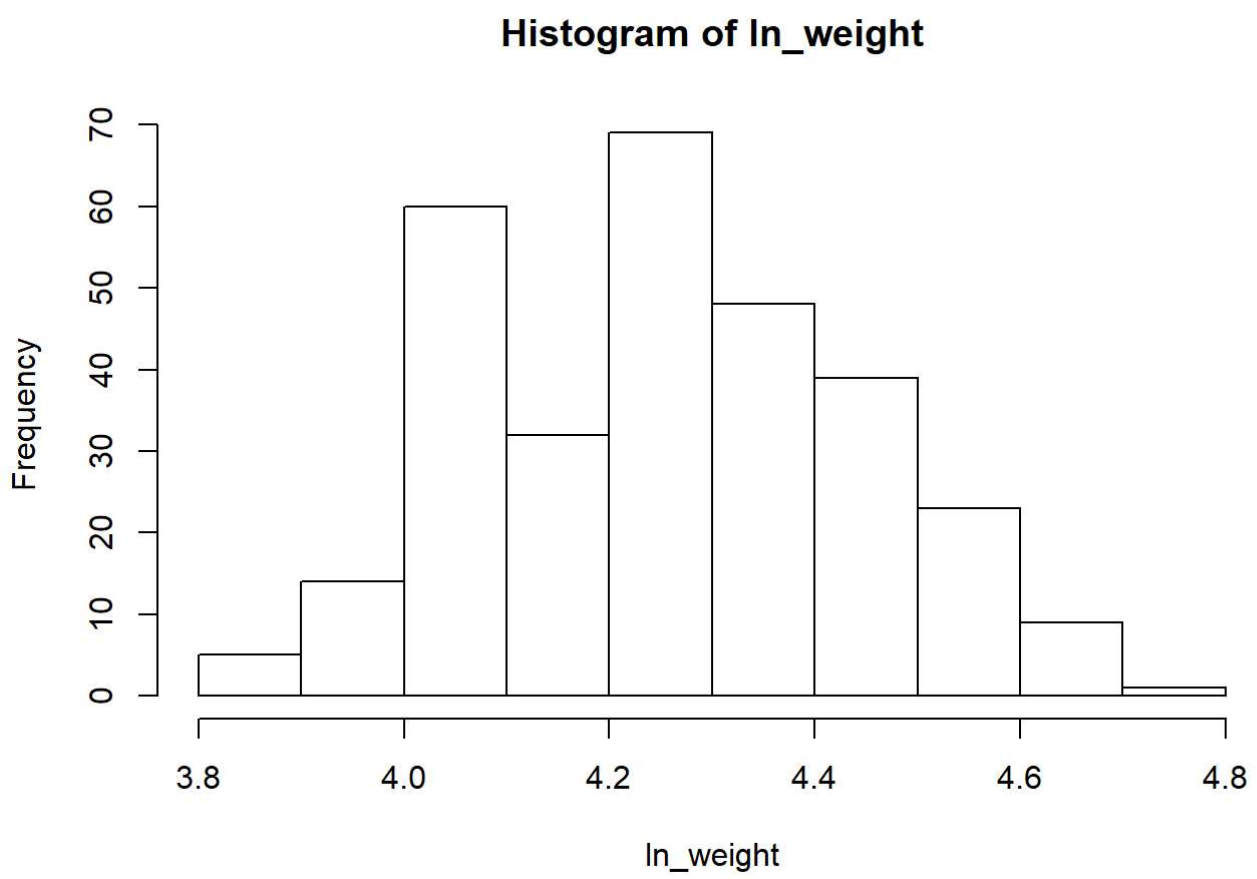
```
#Transformation of weight
hist(bmi_sub3$weight)
```



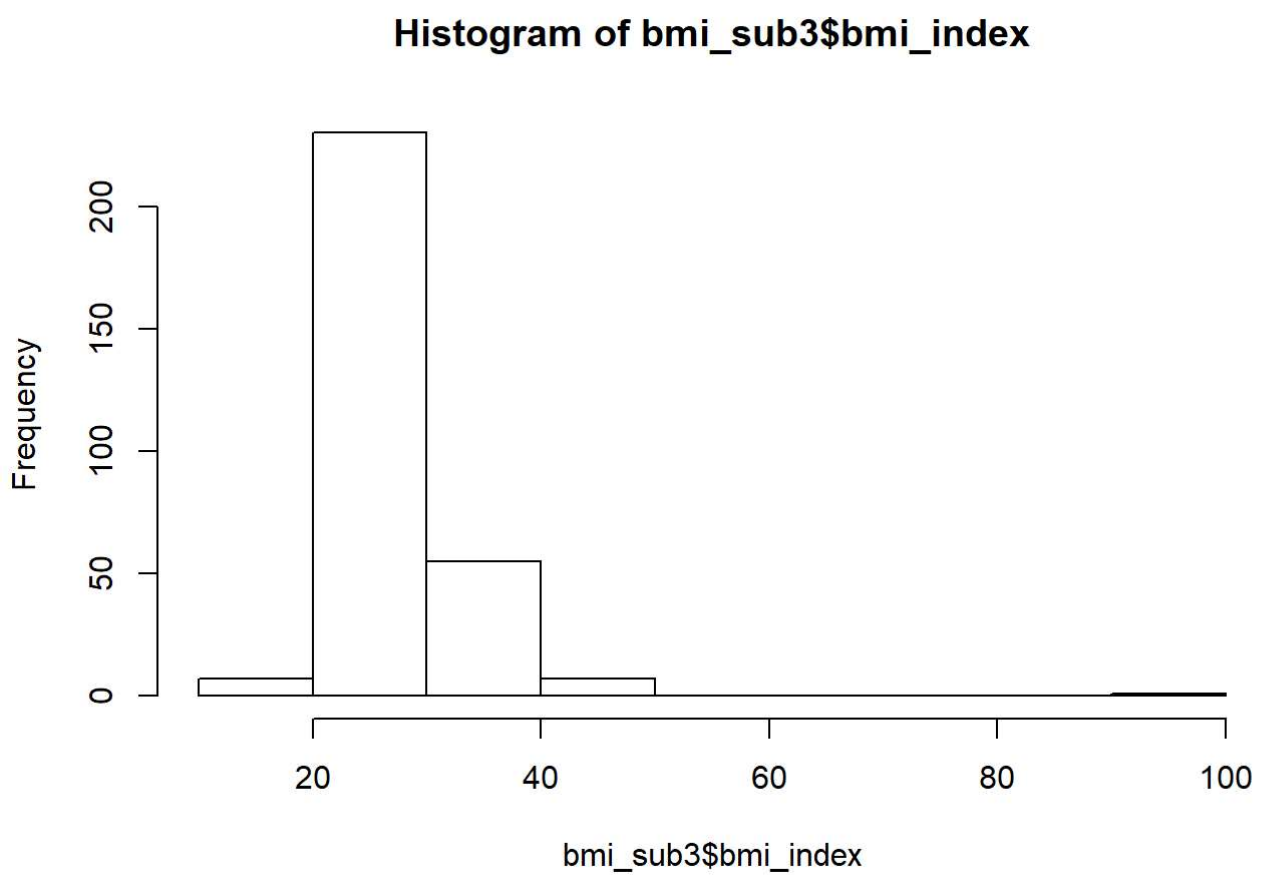

```
boxcox_weight <- BoxCox(weight_capped,lambda = "auto")
hist(boxcox_weight)
```



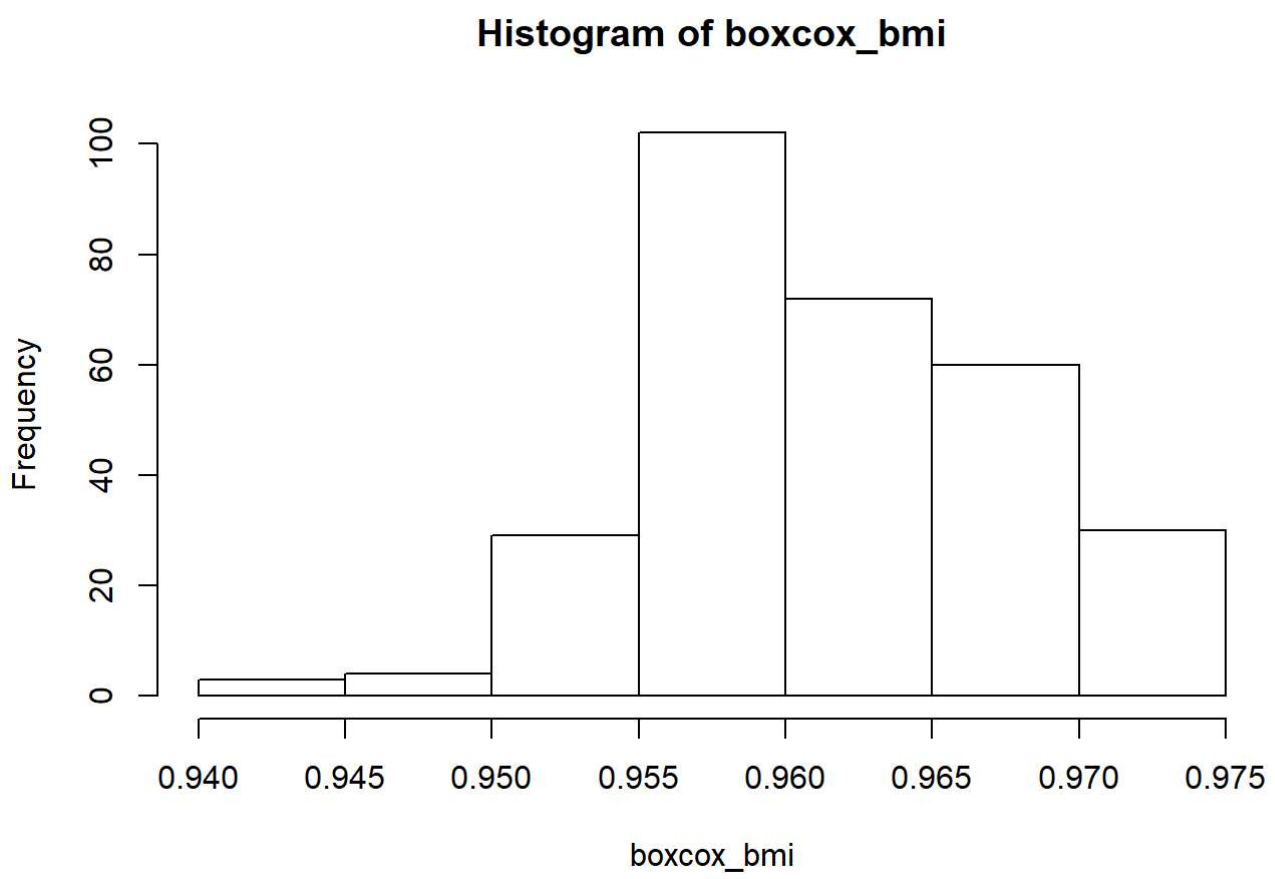

```
ln_weight <- log(weight_capped)
hist(ln_weight)
```



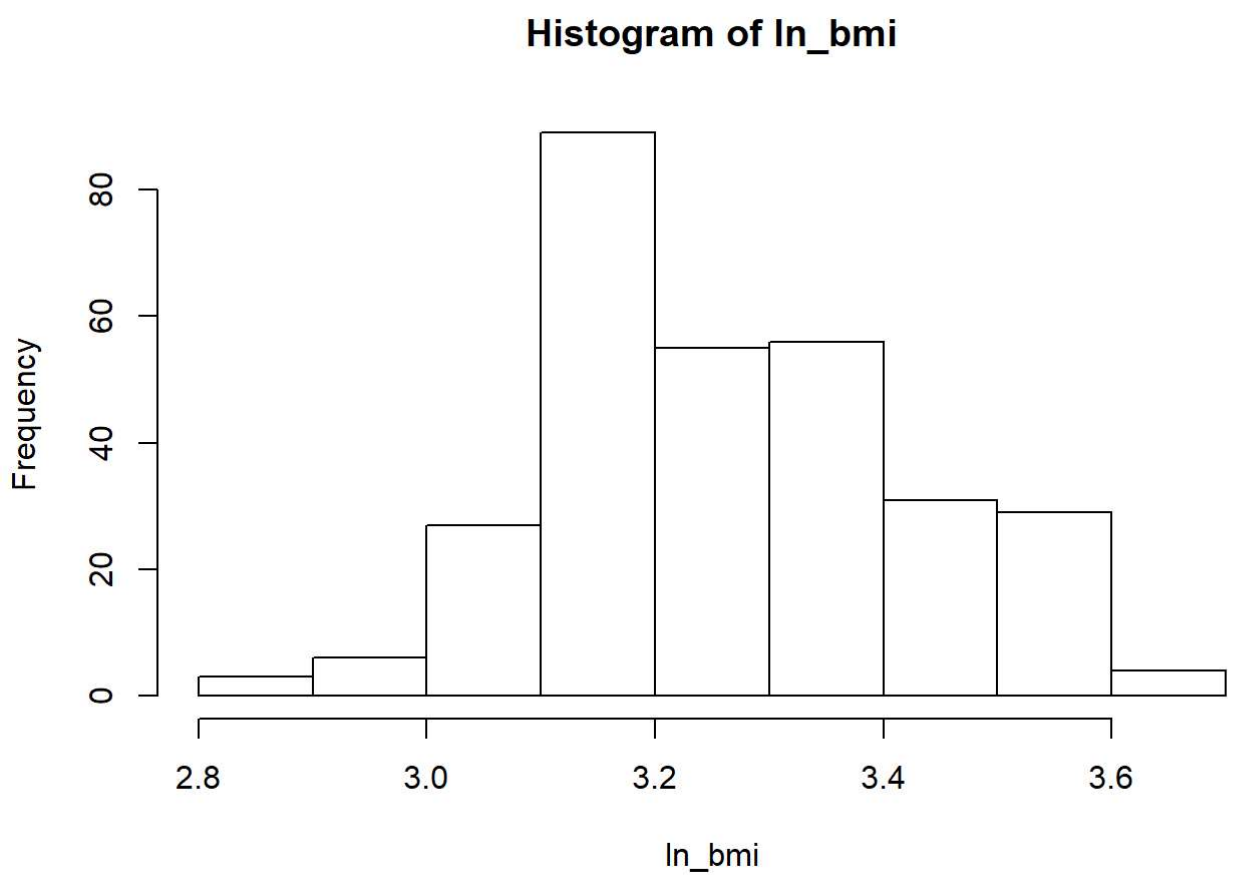

```
#Transformation of bmi index
hist(bmi_sub3$bmi_index)
```




```
boxcox_bmi <- BoxCox(bmi_capped,lambda = "auto")
hist(boxcox_bmi)
```




```
ln_bmi <- log(bmi_capped)
hist(ln_bmi)
```



NOTE: Note that sometimes the order of the tasks may be different than the order given here. For example, you may need to tidy the data sets first to be able to create the common key to merge. Therefore, for such cases you may have a different ordering of the sections.

Any further or optional pre-processing tasks can be added to the template using an additional section in the R Markdown file. Make sure your code is visible (within the margin of the page). Do not use View() to show your data, instead give headers (using head())