Assignment-3

**Abhishek Murthy**

**2024-08-09**

library(scales)  
library(tidyr)  
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(ggplot2)

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

# Import the dataset  
# 21BDS0064  
df= read.csv('C:\\Users\\91984\\OneDrive\\Desktop\\VIT\\Sem7\\Exploratory Data Analysis Lab\\Student\_bucketing.csv')  
head(df)

## Student\_id Age Grade Employed marks  
## 1 1 19 1st Class yes 29  
## 2 2 20 2nd Class no 41  
## 3 3 18 1st Class no 57  
## 4 4 21 2nd Class no 29  
## 5 5 19 1st Class no 57  
## 6 6 20 2nd Class yes 53

# PART 1.  
# Transformation functions  
  
# Finding outliers  
# 21BDS0064  
outliers <- boxplot.stats(df$Age)$out  
print(outliers)

## [1] 88 62 56

median\_df <- median(df$Age[!df$Age %in% outliers], na.rm = TRUE)  
mean\_df <- mean(df$marks[!df$marks %in% outliers], na.rm = TRUE)  
  
# Imputing outliers (e.g., replacing with median)  
# 21BDS0064  
df$Age[df$Age %in% outliers] <- median\_df  
  
# Finding variables with missing values  
# 21BDS0064  
missing\_data <- sapply(df, function(x) sum(is.na(x)))  
print(missing\_data)

## Student\_id Age Grade Employed marks   
## 0 17 0 0 23

# Imputing missing values (e.g., using mean, median)  
# 21BDS0064  
df$Age[is.na(df$Age)] <- median\_df  
df$marks[is.na(df$marks)] <- mean\_df  
  
# Checking for missing data again, it should print out 0  
# 21BDS0064  
missing\_data <- sapply(df, function(x) sum(is.na(x)))  
print(missing\_data)

## Student\_id Age Grade Employed marks   
## 0 0 0 0 0

# Verifying the absence of outliers  
# 21BDS0064  
outliers <- boxplot.stats(df$Age)$out  
print(outliers)

## numeric(0)

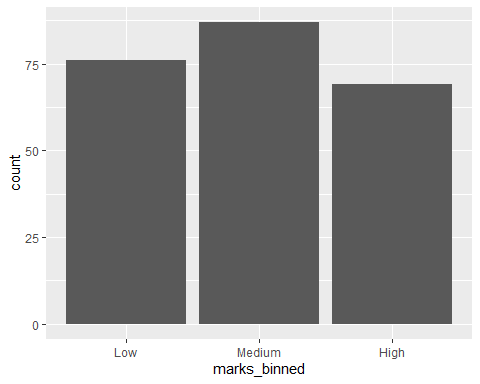
# Summary of the imputed variable  
# 21BDS0064  
summary(df$Age)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 18.0 19.0 20.0 19.9 21.0 22.0

# Binning marks into categories of low medium and high  
# 21BDS0064  
df$marks\_binned <- cut(df$marks, breaks = 3, labels = c("Low", "Medium", "High"))  
  
# Summary of the binning result  
# 21BDS0064  
summary(df$marks\_binned)

## Low Medium High   
## 76 87 69

# Plotting binned data  
# 21BDS0064  
ggplot(df, aes(x = marks\_binned)) + geom\_bar()



# PART 2.  
# Functions on the dataset  
  
#1. Arrange the dataframe in a particular order  
# 21BDS0064  
age\_df <- arrange(df, Age)  
print(head(age\_df))

## Student\_id Age Grade Employed marks marks\_binned  
## 1 3 18 1st Class no 57 Medium  
## 2 18 18 3rd Class no 27 Low  
## 3 32 18 1st Class no 88 High  
## 4 47 18 3rd Class no 72 Medium  
## 5 61 18 1st Class no 93 High  
## 6 76 18 3rd Class no 67 Medium

#2. Select only the age column of the dataframe  
# 21BDS0064  
age\_selected <- select(df, -Age)  
print(head(age\_selected))

## Student\_id Grade Employed marks marks\_binned  
## 1 1 1st Class yes 29 Low  
## 2 2 2nd Class no 41 Low  
## 3 3 1st Class no 57 Medium  
## 4 4 2nd Class no 29 Low  
## 5 5 1st Class no 57 Medium  
## 6 6 2nd Class yes 53 Medium

#3. Filter rows where Sepal.Length is greater than 5  
# 21BDS0064  
filtered\_df <- df[df$marks > 80, ]  
print(head(filtered\_df))

## Student\_id Age Grade Employed marks marks\_binned  
## 9 9 22 3rd Class yes 97 High  
## 11 11 20 3rd Class yes 83 High  
## 16 16 19 2nd Class no 98 High  
## 19 19 21 2nd Class yes 82 High  
## 24 24 22 3rd Class no 94 High  
## 25 25 21 1st Class no 84 High

#4. Gather example  
# 21BDS0064  
marks\_long <- gather(df, key = "marks", value = "Age", marks, Age)  
print(head(marks\_long))

## Student\_id Grade Employed marks\_binned marks Age  
## 1 1 1st Class yes Low marks 29  
## 2 2 2nd Class no Low marks 41  
## 3 3 1st Class no Medium marks 57  
## 4 4 2nd Class no Low marks 29  
## 5 5 1st Class no Medium marks 57  
## 6 6 2nd Class yes Medium marks 53

#5. Group by Grade and summarize  
# 21BDS0064  
df\_grouped <- df %>%  
 group\_by(Grade) %>%  
 summarize(average\_marks = mean(marks, na.rm = TRUE))  
print(head(df\_grouped))

## # A tibble: 3 × 2  
## Grade average\_marks  
## <chr> <dbl>  
## 1 1st Class 57.8  
## 2 2nd Class 59.0  
## 3 3rd Class 59.6

# PART 3.  
# Normalization examples  
  
#1. Normalize data between 0 and 1  
# 21BDS0064  
df\_normalized\_0\_1 <- as.data.frame(lapply(iris[, sapply(iris, is.numeric)], rescale))  
print(head(df\_normalized\_0\_1))

## Sepal.Length Sepal.Width Petal.Length Petal.Width  
## 1 0.22222222 0.6250000 0.06779661 0.04166667  
## 2 0.16666667 0.4166667 0.06779661 0.04166667  
## 3 0.11111111 0.5000000 0.05084746 0.04166667  
## 4 0.08333333 0.4583333 0.08474576 0.04166667  
## 5 0.19444444 0.6666667 0.06779661 0.04166667  
## 6 0.30555556 0.7916667 0.11864407 0.12500000

#2. Normalize data between -1 and 1  
# 21BDS0064  
normalize\_neg1\_1 <- function(x) {  
 return((2 \* (x - min(x)) / (max(x) - min(x))) - 1)  
}  
  
df\_normalized\_neg1\_1 <- as.data.frame(lapply(iris[, sapply(iris, is.numeric)], normalize\_neg1\_1))  
print(head(df\_normalized\_neg1\_1))

## Sepal.Length Sepal.Width Petal.Length Petal.Width  
## 1 -0.5555556 0.25000000 -0.8644068 -0.9166667  
## 2 -0.6666667 -0.16666667 -0.8644068 -0.9166667  
## 3 -0.7777778 0.00000000 -0.8983051 -0.9166667  
## 4 -0.8333333 -0.08333333 -0.8305085 -0.9166667  
## 5 -0.6111111 0.33333333 -0.8644068 -0.9166667  
## 6 -0.3888889 0.58333333 -0.7627119 -0.7500000

#3. Z-score normalization  
# 21BDS0064  
z\_score\_normalize <- function(x) {  
 return((x - mean(x)) / sd(x))  
}  
  
df\_normalized\_z\_score <- as.data.frame(lapply(iris[, sapply(iris, is.numeric)], z\_score\_normalize))  
print(head(df\_normalized\_z\_score))

## Sepal.Length Sepal.Width Petal.Length Petal.Width  
## 1 -0.8976739 1.01560199 -1.335752 -1.311052  
## 2 -1.1392005 -0.13153881 -1.335752 -1.311052  
## 3 -1.3807271 0.32731751 -1.392399 -1.311052  
## 4 -1.5014904 0.09788935 -1.279104 -1.311052  
## 5 -1.0184372 1.24503015 -1.335752 -1.311052  
## 6 -0.5353840 1.93331463 -1.165809 -1.048667

# Find range across all numeric columns in the dataframe  
# 21BDS0064  
numeric\_columns <- sapply(df, is.numeric)  
range\_df <- max(df[, numeric\_columns], na.rm=TRUE) - min(df[, numeric\_columns], na.rm=TRUE)  
print(range\_df)

## [1] 231