CSA1622 Data warehousing and Data Mining

Thanuja R
(192325004)

1.The intervals and corresponding frequencies are as follows. age frequency

1-5. 200

5-15 450

15-20 300

20-50 1500

50-80 700

80-110 44

Compute an approximate median value for the data

Code:

```
age_intervals <- c("1-5", "5-15", "15-20", "20-50", "50-80", "80-110")
frequencies <- c(200, 450, 300, 1500, 700, 44)
c <- cumsum(frequencies)
t <- sum(frequencies)
pos <- t / 2
i <- which(c >= pos)[1]
med <- age_intervals[i]
l <- as.numeric(strsplit(med, "-")[[1]][1])
u <- as.numeric(strsplit(med, "-")[[1]][2])
freq <- frequencies[i]
cumulative_frequency_before <- ifelse(i == 1, 0, c[i - 1])
median <- l + ((pos - cumulative_frequency_before) / freq) * (u - l)
cat("cumulative frequencies:",c)
cat("median:",median)
```

Output:

```
> cat("cumulative frequencies:",cumulative_frequencies)
cumulative frequencies: 200 650 950 2450 3150 3194> cat("median:",median)
median: 32.94
```

2.Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

(a) What is the mean of the data? What is the median?

(b) What is the mode of the data? Comment on the data's modality (i.e., bimodal, trimodal, etc.).

(c) What is the midrange of the data?

(d) Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data?

Code:

```
age <- c(13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25,30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70)
m <- mean(age)
med <- median(age)
mod <- as.numeric(names(sort(table(age), decreasing = TRUE)[1]))
mc <- max(table(age))
mfeq<- table(age)
mfreq <- max(mode_frequencies)
modes <- as.numeric(names(mode_frequencies[mfeq== mfreq]))
modality <- length(modes)
midrange_value <- (min(age) + max(age)) / 2
Q1 <- quantile(age, 0.25)
Q3 <- quantile(age, 0.75)
cat("Mean:", m, "\n")
cat("Median:", med, "\n")
cat("Mode(s):", paste(modes, collapse = ", "), "with frequency:", mc, "\n")
cat("Modality:", ifelse(modality == 1, "Unimodal", ifelse(modality == 2, "Bimodal", "Multimodal")), "\n")
cat("Midrange:", midrange_value, "\n")
cat("First Quartile (Q1):", Q1, "\n")
cat("Third Quartile (Q3):", Q3, "\n")
```

Output:

```
Mean: 29.96296
> cat("Median:", median_value, "\n")
Median: 25
> cat("Mode(s):", paste(modes, collapse = ", "), "with frequency:", mode_count, "\n")
Mode(s): 25, 35 with frequency: 4
> cat("Modality:", ifelse(modality == 1, "Unimodal",
+                         ifelse(modality == 2, "Bimodal", "Multimodal")), "\n")
Modality: Bimodal
> cat("Midrange:", midrange_value, "\n")
Midrange: 41.5
> cat("First Quartile (Q1):", Q1, "\n")
First Quartile (Q1): 20.5
> cat("Third Quartile (Q3):", Q3, "\n")
Third Quartile (Q3): 35
```

3. Data Preprocessing :Reduction and Transformation Use the two methods below to normalize the following group of data: 200, 300, 400, 600, 1000 (a) min-max normalization by setting min = 0 and max = 1 (b) z-score normalization

Code:

```
data <- c(200, 300, 400, 600, 1000)
min_val <- min(data)
max_val <- max(data)
min_max_norm <- (data - min_val) / (max_val - min_val)
mean_val <- mean(data)
sd_val <- sd(data)
z_score_norm <- (data - mean_val) / sd_val
cat("Min-Max Normalization:\n", min_max_norm, "\n\n")
cat("Z-Score Normalization:\n", z_score_norm, "\n")
```

Output :

```
Min-Max Normalization:
 0 0.125 0.25 0.5 1

> cat("Z-Score Normalization:\n", z_score_norm, "\n")
Z-Score Normalization:
 -0.9486833 -0.6324555 -0.3162278 0.3162278 1.581139
```

4.Data:11,13,13,15,15,16,19,20,20,20,21,21,22,23,24,30,40,45,45,45,71, 72,73,75
a) Smoothing by bin mean
b) Smoothing by bin median
c) Smoothing by bin boundaries.

Code:

```
data <- c(11,13,13,15,15,16,19,20,20,20,21,21,22,23,24,30,40,45,45,45,71,72,73,75)
num_bins <- 4
bin_size <- ceiling(length(data) / num_bins)
sorted_data <- sort(data)
bins <- split(sorted_data, ceiling(seq_along(sorted_data) / bin_size))
bin_means <- sapply(bins, mean)
smoothed_mean <- unlist(lapply(seq_along(bins), function(i) rep(bin_means[i], length(bins[[i]]))))
bin_medians <- sapply(bins, median)
smoothed_median <- unlist(lapply(seq_along(bins), function(i) rep(bin_medians[i], length(bins[[i]]))))
smoothed_boundaries <- unlist(lapply(bins, function(bin) {
  lower <- min(bin)
  upper <- max(bin)
  sapply(bin, function(x) ifelse(abs(x - lower) < abs(x - upper), lower, upper))
}))
cat("Original Data:\n", sorted_data, "\n\n")
cat("Smoothing by Bin Mean:\n", smoothed_mean, "\n\n")
cat("Smoothing by Bin Median:\n", smoothed_median, "\n\n")
cat("Smoothing by Bin Boundaries:\n", smoothed_boundaries, "\n")
```

Output:

```
Original Data:
 11 13 13 15 15 16 19 20 20 20 21 21 22 23 24 30 40 45 45 45 71 72 73 75

> cat("Smoothing by Bin Mean:\n", smoothed_mean, "\n\n")
Smoothing by Bin Mean:
 13.83333 13.83333 13.83333 13.83333 13.83333 13.83333 20.16667 20.16667 20.16667 20.16667 20.16667 20.16667 30.66667 30.66667 30.66667 30.66667
30.66667 30.66667 63.5 63.5 63.5 63.5 63.5 63.5

> cat("Smoothing by Bin Median:\n", smoothed_median, "\n\n")
Smoothing by Bin Median:
 14 14 14 14 14 14 20 20 20 20 20 20 27 27 27 27 27 27 71.5 71.5 71.5 71.5 71.5 71.5

> cat("Smoothing by Bin Boundaries:\n", smoothed_boundaries, "\n")
Smoothing by Bin Boundaries:
 11 11 11 16 16 16 19 21 21 21 21 21 22 22 22 22 45 45 45 45 75 75 75 75
```

5.Suppose that a hospital tested the age and body fat data for 18 randomly selected adults with the following results:

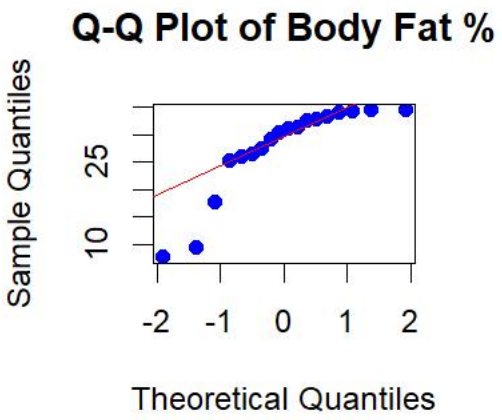
age	23	23	27	27	39	41	47	49	50
%fat	9.5	26.5	7.8	17.8	31.4	25.9	27.4	27.2	31.2
age	52	54	54	56	57	58	58	60	61
%fat	34.6	42.5	28.8	33.4	30.2	34.1	32.9	41.2	35.7

(a) Calculate the mean, median, and standard deviation of age and %fat.
(b) Draw the boxplots for age and %fat.
(c) Draw a scatter plot and a q-q plot based on these two variables.

Code:

```
age <- c(23, 23, 27, 27, 39, 41, 45, 49, 50, 52, 54, 54, 56, 57, 58, 58, 60, 61)
fat <- c(9.5, 26.5, 7.8, 17.8, 31.4, 25.9, 27.4, 25.2, 31.1, 34.4, 29.1, 32.5,
         30.3, 33.3, 34.2, 34.1, 32.9, 34.5)
age_mean <- mean(age)
age_median <- median(age)
age_sd <- sd(age)
fat_mean <- mean(fat)
fat_median <- median(fat)
fat_sd <- sd(fat)
cat("Age - Mean:", age_mean, "Median:", age_median, "Standard Deviation:", age_sd, "\n")
cat("Body Fat % - Mean:", fat_mean, "Median:", fat_median, "Standard Deviation:", fat_sd, "\n\n")
par(mfrow=c(1,2)) # Set layout for side-by-side plots
boxplot(age, main="Boxplot of Age", col="lightblue")
boxplot(fat, main="Boxplot of Body Fat %", col="lightcoral")
par(mfrow=c(1,2)) # Reset layout
plot(age, fat, main="Scatter Plot of Age vs. Body Fat %",xlab="Age", ylab="Body Fat %", col="blue", pch=19)
qqnorm(age, main="Q-Q Plot of Age", col="blue", pch=19)
qqline(age, col="red")
qqnorm(fat, main="Q-Q Plot of Body Fat %", col="blue", pch=19)
qqline(fat, col="red")
par(mfrow=c(1,1))
```

Output:



6.Suppose that a hospital tested the age and body fat data for 18 randomly selected adults with the following results:

(i) Use min-max normalization to transform the value 35 for age onto the range [0.0, 1.0].
(ii) Use z-score normalization to transform the value 35 for age, where the standard deviation of age is 12.94 years.
(iii) Use normalization by decimal scaling to transform the value 35 for age. Perform the above functions using R – tool

```
min_age <- 23
max_age <- 61
age_value <- 35
std_dev <- 12.94
min_max_norm <- (age_value - min_age) / (max_age - min_age)
mean_age <- (min_age + max_age) / 2
z_score_norm <- (age_value - mean_age) / std_dev
max_abs_age <- max(abs(min_age), abs(max_age))
scaling_factor <- 10^ceiling(log10(max_abs_age))
decimal_scaling_norm <- age_value / scaling_factor
print(paste("Min-Max Normalization:", min_max_norm))
print(paste("Z-Score Normalization:", z_score_norm))
print(paste("Decimal Scaling Normalization:", decimal_scaling_norm))
```

Output:

```
> scaling_factor <- 10^ceiling(log10(max_abs_age))
> decimal_scaling_norm <- age_value / scaling_factor
> print(paste("Min-Max Normalization:", min_max_norm))
[1] "Min-Max Normalization: 0.315789473684211"
> print(paste("Z-Score Normalization:", z_score_norm))
[1] "Z-Score Normalization: -0.540958268933539"
> print(paste("Decimal Scaling Normalization:", decimal_scaling_norm))
[1] "Decimal Scaling Normalization: 0.35"
>
```

7.The following values are the number of pencils available in the different boxes. Create a vector and find out the mean, median and mode values of set of pencils in the given data.

Box1 Box2 Box3 Box4 Box5 Box6 Box7 Box8 Box9 Box 10

9 25 23 12 11 6 7 8 9 10

Code:

```
pencils <- c(9, 25, 23, 12, 11, 6, 7, 8, 9, 10)
mean_value <- mean(pencils)
median_value <- median(pencils)
mode_value <- as.numeric(names(sort(table(pencils), decreasing=TRUE))[1])
print(paste("Mean:", mean_value))
print(paste("Median:", median_value))
print(paste("Mode:", mode_value))
```

Output :

```
[1] "Mean: 12"
> print(paste("Median:", median_value))
[1] "Median: 9.5"
> print(paste("Mode:", mode_value))
[1] "Mode: 9"
>
```

8.the following table would be plotted as (x,y) points, with the first column being the x values as number of mobile phones sold and the second column being the y values as money. To use the scatter plot for how many mobile phones sold.

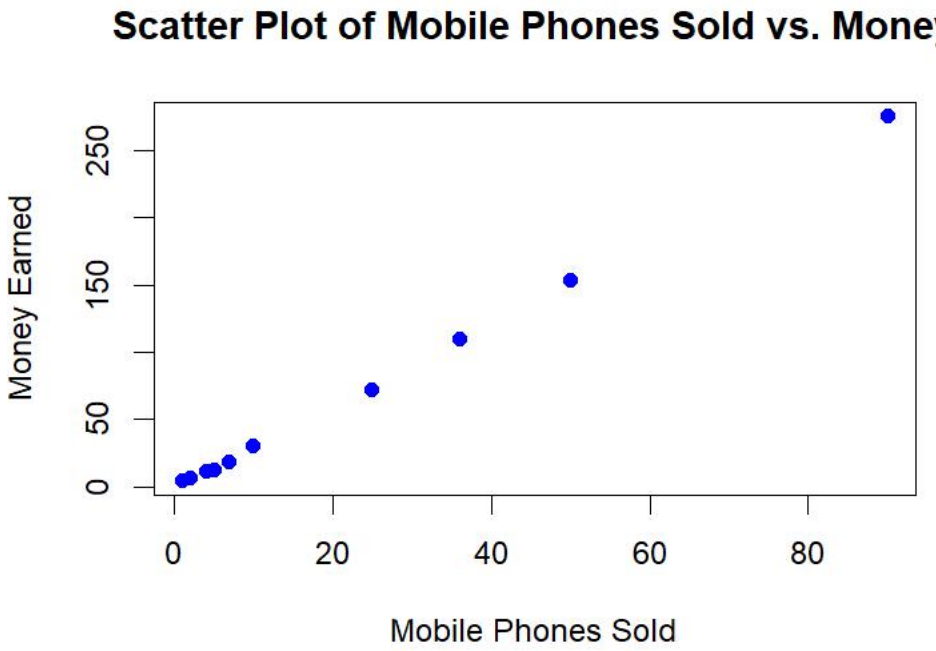
x :4 1 5 7 10 2 50 25 90 36

y :12 5 13 19 31 7 153 72 275 110

Code:

```
x <- c(4, 1, 5, 7, 10, 2, 50, 25, 90, 36)
y <- c(12, 5, 13, 19, 31, 7, 153, 72, 275, 110)
plot(x, y, main="Scatter Plot of Mobile Phones Sold vs. Money",
     xlab="Mobile Phones Sold", ylab="Money Earned", col="blue", pch=16)
```

Output :



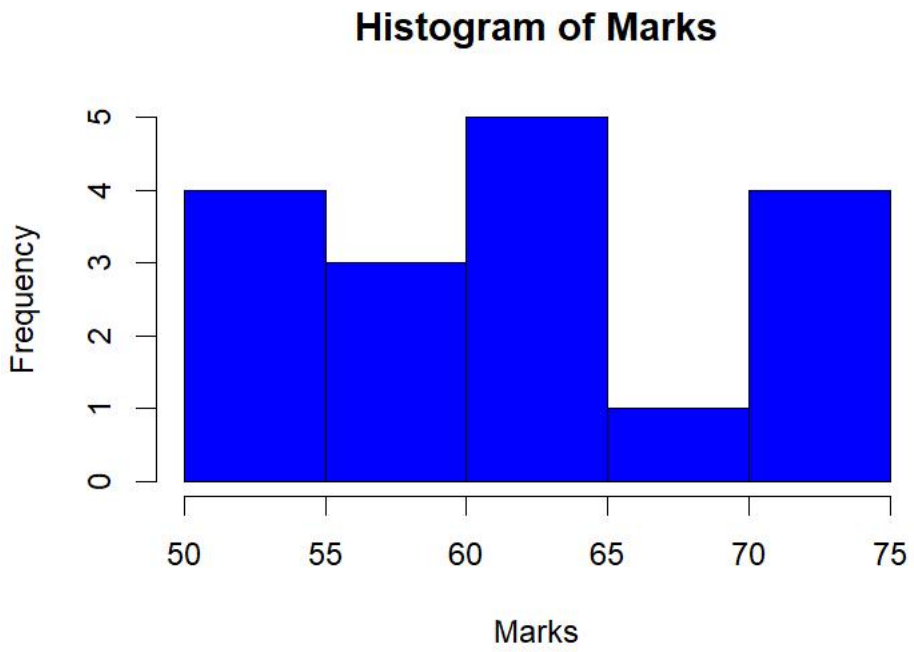
9.Implement of the R script using marks scored by a student in his model exam has been sorted as follows: 55, 60, 71, 63, 55, 65, 50, 55,58,59,61,63,65,67,71,72,75. Partition them into three bins by each of the following methods. Plot the data points using histogram.

(a) equal-frequency (equi-depth) partitioning (b) equal-width partitioning

Code:

```
marks <- c(55, 60, 71, 63, 55, 65, 50, 55, 58, 59, 61, 63, 65, 67, 71, 72, 75)
bins_eq_freq <- split(sort(marks), cut(seq_along(marks), breaks=3, labels=FALSE))
bins_eq_width <- split(marks, cut(marks, breaks=3, include.lowest=TRUE))
hist(marks, main="Histogram of Marks", xlab="Marks", col="blue", border="black")
```

Output :



10.Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order) 13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70.

Can you find (roughly) the first quartile (Q1) and the third quartile (Q3) of the data?

CODE:

```
age <- c(13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35,
Q1 <- quantile(age, 0.25)
Q3 <- quantile(age, 0.75)
print(paste("Q1:", Q1))
print(paste("Q3:", Q3))
```

Output :

```
> print(paste("Q1:", Q1))
[1] "Q1: 20.5"
> print(paste("Q3:", Q3))
[1] "Q3: 35"
```