

21000298.R

ASUS

2023-12-11

```
heart_data <- read.csv("dataset/heart.csv")

#1)iterate through data set to count the number of rows
row_count = 0
while(row_count < nrow(heart_data)){
  row_count <- row_count + 1
}
print(row_count)

## [1] 918

#2)find the summary of resting RestingBP

#get the Min. value
min_value <- min(heart_data$RestingBP, na.rm = TRUE)
max_value <- max(heart_data$RestingBP, na.rm = TRUE)

#calculate mean value
restingbp_row_count <- 0
restingbp_sum <- 0
total_non_missing <- sum(!is.na(heart_data$RestingBP))

while (restingbp_row_count < total_non_missing) {
  restingbp_sum <- restingbp_sum + heart_data$RestingBP[restingbp_row_count + 1]
  restingbp_row_count <- restingbp_row_count + 1
}
mean_value <- restingbp_sum / restingbp_row_count

# Calculate the median
sorted_data <- sort(heart_data$RestingBP[!is.na(heart_data$RestingBP)])
n <- length(sorted_data)
median_value <- ifelse(n %% 2 == 0, (sorted_data[n/2] + sorted_data[n/2 + 1]) / 2, sorted_data[(n + 1)/2])

# Print the custom summary
cat("Custom Summary for", "RestingBP", ":\n")

## Custom Summary for RestingBP :

cat("Mean:", mean_value, "\n")

## Mean: 132.3965

cat("Median:", median_value, "\n")

## Median: 130

cat("Minimum:", min_value, "\n")

## Minimum: 0

cat("Maximum:", max_value, "\n")

## Maximum: 200

#3)Find the number of patients with “higher than mean” RestingBP.

higher_than_mean_patient_count <- 0
restingbp_row_count <- 0

while (restingbp_row_count < total_non_missing) {
  if(heart_data$RestingBP[restingbp_row_count + 1] > mean_value){
    higher_than_mean_patient_count <- higher_than_mean_patient_count + 1
  }
  restingbp_row_count <- restingbp_row_count + 1
}
cat("Higher Than Mean Patient Count:", higher_than_mean_patient_count, "\n")

## Higher Than Mean Patient Count: 404

#4)Find unique chest pain types.

i <- 1
total_non_missing_chestpaintype <- sum(!is.na(heart_data$ChestPainType))
chestpaintype_row_count <- 1
unique_types <- list()

while (chestpaintype_row_count < total_non_missing_chestpaintype) {
  if(!is.na(heart_data$ChestPainType[chestpaintype_row_count + 1])){
    unique_types[[i]] <- heart_data$ChestPainType[chestpaintype_row_count + 1]
    i <- i + 1
  }
  chestpaintype_row_count <- chestpaintype_row_count + 1
}

print(unique_types)

## [[1]]
## [1] "NAP"
##
## [[2]]
## [1] "ATA"
##
## [[3]]
## [1] "ASY"
##
## [[4]]
## [1] "TA"

#5)Print the ages of the first 10 Females whose resting BP is greater than 140.

i <- 1
j <- 1
while (i <= 10) {
  if (heart_data$Sex[j] == "F" && heart_data$RestingBP[j] > 140) {
    print(heart_data$Age[j])
    i <- i + 1
  }
  j <- j + 1
}

## [1] 49
## [1] 43
## [1] 48
## [1] 54
## [1] 51
## [1] 43
## [1] 40
## [1] 58
## [1] 54
## [1] 62

#6)Write a function which takes a row of data as input and outputs the attribute
#information in a structured manner. If cholesterol level is less than 200 then print low
#else high.

patient_info <- function(row) {
  cat("Age of Patient : ", row$Age, "\n")

  if (row$Sex == "M"){
    cat("Gender of Patient : Male", "\n")
  }else{
    cat("Gender of Patient : Female", "\n")
  }

  if (row$Cholesterol >= 200) {
    cat("Cholesterol : High", "\n")
  } else {
    cat("Cholesterol : Low", "\n")
  }
}

patient_info(heart_data[1,])

## Age of Patient : 40
## Gender of Patient : Male
## Cholesterol : High

patient_info(heart_data[2,])

## Age of Patient : 49
## Gender of Patient : Female
## Cholesterol : Low
```