```r
mice <- read.csv("femaleMiceWeights.csv")
```

```r
#Lab 15_1A
class(mice)
```

```
## [1] "data.frame"
```

```r
#Lab 15_1C
dim(mice)
```

```
## [1] 24  2
```

```r
#Lab 15_2
colnames(mice)
```

```
## [1] "Diet"       "Bodyweight"
```

```r
#Lab 15_3
mice[12, 2]
```

```
## [1] 26.25
```

```r
#Lab 15_4
mice$Bodyweight[11]
```

```
## [1] 26.91
```

```r
#Lab 15_5
length(mice$Bodyweight)
```

```
## [1] 24
```

```r
#Lab 15_6
hf_weights <- mice$Bodyweight[mice$Diet == "hf"]
mean(hf_weights)
```

```
## [1] 26.83417
```

```r
#Lab 15_II 1
# First 20 elements: 1 to 20
part1 <- seq(1, 20)

# Next 10 elements: 10, 20, ..., 100
part2 <- seq(10, 100, by=10)

# Last 70 elements: 31 to 100
part3 <- seq(31, 100)

# Combine all parts into one vector
my_vector <- c(part1, part2, part3)
```

```r
#Lab 15_II 2
print(my_vector)
```

```
##   [1]   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##  [19]  19  20  10  20  30  40  50  60  70  80  90 100  31  32  33  34  35  36
##  [37]  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##  [55]  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##  [73]  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##  [91]  91  92  93  94  95  96  97  98  99 100
```

```r
# Lab 15_III
load("hersdata.Rdata")  # Ensure the file is in your working directory
```

```r
# Lab 15_III 2
class(hersdata$drinkany)  # Check current class
```

```
## [1] "character"
```

```r
hersdata$drinkany <- as.factor(hersdata$drinkany)  # Convert to factor
class(hersdata$drinkany)  # Verify conversion
```

```
## [1] "factor"
```

```r
# Lab 15_III 3
# Install package
library(summarytools)  # Load package

# Create a one-way frequency table
freq(hersdata$drinkany)
```

```
## Frequencies
## hersdata$drinkany
## Type: Factor
##
##               Freq   % Valid   % Valid Cum.   % Total   % Total Cum.
## ----------- ------ --------- -------------- --------- --------------
##          no   1680    60.848         60.848    60.803         60.803
##         yes   1081    39.152        100.000    39.124         99.928
##        <NA>      2                             0.072        100.000
##       Total   2763   100.000        100.000   100.000        100.000
```

```r
# Count missing values
sum(is.na(hersdata$drinkany))
```

```
## [1] 2
```

```r
# Lab 15_ III 4
hersdata$drinkany01 <- ifelse(hersdata$drinkany == "no", 0,
                              ifelse(hersdata$drinkany == "yes", 1, NA))

# Check the conversion
table(hersdata$drinkany, hersdata$drinkany01, useNA = "always")
```

```
##
##            0    1 <NA>
##   no    1680    0    0
##   yes      0 1081    0
##   <NA>     0    0    2
```

install.packages("tidyverse", dependencies = TRUE)

```r
# Lab 15_III 5
library(tidyverse)  # Load package
```

```
## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v ggplot2   3.5.1     v tibble    3.2.1
```

```
## v lubridate 1.9.4     v tidyr     1.3.1
## v purrr     1.0.4
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## x tibble::view()  masks summarytools::view()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```r
# Now try the corrected approach
mytiny <- hersdata %>%
  # First filter for observations where weight > 125
  filter(weight > 125) %>%
  # Then select only the specified variables
  select(HT, LDL, SBP)

# Lab 15_III 6
print(mytiny)
```

```
## # A tibble: 5 x 3
##   HT                LDL   SBP
##   <chr>           <dbl> <dbl>
## 1 hormone therapy  122.   129
## 2 placebo          205.   133
## 3 placebo          161.   112
## 4 placebo          137    130
## 5 placebo          148.   139
```

```r
# Lab 15_Gen AI
# Loading the dataset
# Assuming the file is named 'sample_transactions.csv' and is in your working directory
transactions <- read.csv("sample_transactions.csv", stringsAsFactors = FALSE)
```

```r
# Lab 15_Gen AI 1a

# Display the data frame structure
str(transactions)
```

```
## 'data.frame':    10 obs. of  6 variables:
##  $ Customer.ID       : int  101 102 103 104 105 106 107 108 109 110
##  $ Transaction.Date  : chr  "1/12/2025" "1/13/2025" "1/14/2025" "1/15/2025" ...
##  $ Product.Category  : chr  "Electronics" "Clothing" "Electronics" "Home & Kitchen" ...
##  $ Quantity.Purchased: int  1 3 2 1 4 2 5 3 1 2
##  $ Price.per.Unit    : num  500 40 300 80 30 ...
##  $ Payment.Method    : chr  "Credit Card" "PayPal" "Credit Card" "Debit Card" ...
```

```r
# Lab 15_Gen AI 1B
#  Write a function to find the top 3 most frequently purchased product categories
find_top_categories <- function(data) {
  # Calculate the frequency of each product category
  category_counts <- table(data$Product.Category)

  # Sort in descending order
  sorted_categories <- sort(category_counts, decreasing = TRUE)

  # Return the top 3 (or less if there are fewer categories)
  top_n <- min(3, length(sorted_categories))
```

```r
  return(head(sorted_categories, top_n))
}

# Call the function to find top 3 categories
top_categories <- find_top_categories(transactions)
print("Top 3 most frequently purchased product categories:")
```

```
## [1] "Top 3 most frequently purchased product categories:"
```

```r
print(top_categories)
```

```
##
##    Clothing Electronics      Beauty
##           3           3           2
```

```r
# Lab 15_Gen AI 2A
# 2. Control Structures
# Calculate the total revenue from each product category using a loop
calculate_revenue_by_category <- function(data) {
  # Get unique product categories
  categories <- unique(data$Product.Category)

  # Initialize an empty list to store results
  revenue_by_category <- list()

  # Loop through each category
  for (category in categories) {
    # Filter transactions for the current category
    category_data <- data[data$Product.Category == category, ]

    # Calculate revenue for each transaction (Quantity * Price)
    transaction_revenue <- category_data$Quantity.Purchased * category_data$Price.per.Unit

    # Sum the revenue for the category
    total_revenue <- sum(transaction_revenue)

    # Store in the list
    revenue_by_category[[category]] <- total_revenue
  }

  return(revenue_by_category)
}

# Call the function to calculate revenue by category
revenue_by_category <- calculate_revenue_by_category(transactions)
print("Total revenue generated from each product category:")
```

```
## [1] "Total revenue generated from each product category:"
```

```r
print(revenue_by_category)
```

```
## $Electronics
## [1] 2099.92
##
## $Clothing
## [1] 339.91
```

```
##
## $`Home & Kitchen`
## [1] 259.96
##
## $Beauty
## [1] 49.97
```

```
# Lab 15_Gen AI 2B
# Analyze the most preferred payment method using conditional statements
analyze_payment_methods <- function(data) {
  # Calculate the frequency of each payment method
  payment_counts <- table(data$Payment.Method)

  # Find the most common payment method
  most_preferred <- names(which.max(payment_counts))

  # Conditional analysis based on payment method popularity
  result <- list(
    counts = payment_counts,
    most_preferred = most_preferred,
    analysis = character(0)
  )

  if (most_preferred == "Credit Card") {
    result$analysis <- "Credit Card is the most popular payment method, suggesting customers prefer trad
  } else if (most_preferred == "PayPal") {
    result$analysis <- "PayPal is the most popular payment method, indicating a preference for digital w
  } else if (most_preferred == "Debit Card") {
    result$analysis <- "Debit Card is the most popular payment method, suggesting customers prefer direc
  } else if (most_preferred == "Crypto") {
    result$analysis <- "Cryptocurrency is the most popular payment method, indicating customers are embr
  } else {
    result$analysis <- paste(most_preferred, "is the most popular payment method.")
  }

  return(result)
}

# Call the function to analyze payment methods
payment_analysis <- analyze_payment_methods(transactions)
print("Payment method analysis:")
```

```
## [1] "Payment method analysis:"
```

```
print(payment_analysis$counts)
```

```
##
## Credit Card      Crypto  Debit Card      PayPal
##           4           1           2           3
```

```
print(paste("Most preferred payment method:", payment_analysis$most_preferred))
```

```
## [1] "Most preferred payment method: Credit Card"
```

```
print(payment_analysis$analysis)
```

```
## [1] "Credit Card is the most popular payment method, suggesting customers prefer traditional payment
```

```r
# Additional summary statistics for the dataset
print("Summary of transaction data:")
```

```
## [1] "Summary of transaction data:"
```

```r
summary_stats <- list(
  total_transactions = nrow(transactions),
  total_revenue = sum(transactions$Quantity.Purchased * transactions$Price.per.Unit),
  average_order_value = mean(transactions$Quantity.Purchased * transactions$Price.per.Unit),
  total_items_sold = sum(transactions$Quantity.Purchased)
)
print(summary_stats)
```

```
## $total_transactions
## [1] 10
##
## $total_revenue
## [1] 2749.76
##
## $average_order_value
## [1] 274.976
##
## $total_items_sold
## [1] 24
```