**Prerequisites**

Before the class, please make sure to:

* Review **Module 4**'s asynchronous lessons and assignments to prepare for the live session.
* Ensure **R** and **RStudio** are installed and **RStudio** is open.
* Download the **Live Lesson 04 Challenge files** from Google Classroom [here](https://classroom.google.com/c/NzM2ODYxODA3Mzgz/m/Njg4MTc2MjI3NTUx/details).

**Overview**

In this lesson, you will guide students through the core concepts from Module 4, focusing on key areas of data analysis using R programming. The goal is to ensure students are able to apply R’s functionalities to solve real-world data challenges. You will cover topics such as operators, data structures, data manipulation, visualization, and statistical analysis.

This class is structured to help students move from conceptual understanding to practical application, as well as to engage them in problem-solving exercises that enhance their learning.

**Instructor Tips:**

* **Encourage Interaction:** Throughout the class, foster a space for questions and discussions. If students encounter difficulties, guide them through troubleshooting in R.
* **Real-world Examples:** Whenever possible, explain how these skills apply to actual data analysis problems that students might encounter in professional settings.
* **Pacing:** Keep an eye on the clock to ensure all sections are covered, but be flexible in case students need more time for certain activities.
* **Prepare to Troubleshoot:** Familiarize yourself with common issues that may arise in R and how to troubleshoot them. Be ready to walk through debugging if students encounter errors in their code.

**Lesson Objectives**

* **Reinforce understanding of key R programming concepts**, including operators, data structures, and date/time handling, to ensure students can apply them in real-world data analysis tasks.
* **Demonstrate effective data manipulation** using dplyr to clean and transform datasets, preparing them for analysis and visualization.
* **Guide students through creating meaningful data visualizations** with ggplot2, focusing on interpreting results and choosing the appropriate plot types.
* **Introduce basic statistical and inferential analysis in R**, specifically performing summary statistics and conducting a t-test to compare group differences.
* **Foster practical problem-solving skills** by guiding students through real-world application challenges that involve reading, cleaning, analyzing, and visualizing data.

**Slide Deck**

The slides for this lesson can be viewed on Google Drive here: [M04\_R\_Programming\_Language](https://docs.google.com/presentation/d/1lXFherCpMeirJuUBKzKSCA2Oa6HCspYYiKIze95sfEw/edit?usp=sharing)

**Time Plan**

| **Time** | **Activity** | **Duration** |
| --- | --- | --- |
| 00:00 | **Problem Framing**: R Programming Language | 10 mins |
|  | **Guided Demo:** |  |
| 00:10 | **Setting Up the Environment** | 5 mins |
| 00:15 | **Initial Exploration** | 10 mins |
| 00:25 | **Data Cleaning and Transformation** | 10 mins |
| 00:35 | **Data Visualization** | 10 mins |
| 00:45 | **Statistical Analysis: t-Test** | 10 mins |
| 00:55 | **Correlation and Regression Analysis** | 10 mins |
|  | **Application Challenge:** |  |
| 01:05 | **Challenge:** R-evolutionize Your Data Analysis | 15 mins |
| 01:20 | **Debrief:** World Happiness Report 2023 | 5 mins |
| 01:25 | **Discussion** | 5 mins |
| 01:30 | **END** |  |

**Problem Framing (5 minutes)**

This segment ensures that students are comfortable with foundational R programming skills and ready to apply them to real-world data analysis tasks.

**Key Points:**

* **Recap Core Concepts:** Review the essential topics that were covered in Module 4, including operators, data structures, data manipulation, visualization, and statistics.
* **Engage Students:** Actively involve students by prompting them to reflect on what they found challenging or interesting in the module.
* **Create Context:** Help students see how these foundational concepts are directly relevant to the practical tasks they’ll work on today.

**Recap Key Concepts**

Provide a high-level recap of the core topics from the module. Remember that these concepts will already be familiar to students, so the goal is to jog their memory and prepare them to apply the knowledge in the hands-on section.

* **Operators and Calculations:** Briefly touch on the use of arithmetic operations and functions like sum(), mean(), and sd().
* **Data Structures:** Review the types of **data structures** in R (vectors, data frames, lists, etc.), emphasizing how they are used for different types of data and analysis.
* **Dates and Times:** Mention the importance of handling date and time data, and how functions like as.Date() help with conversion and formatting.
* **Reading Data:** Explain the importance of importing data into R using functions like read.csv(), and the importance of inspecting the data after import.
* **Data Manipulation with dplyr:** Highlight how dplyr is used to **filter**, **mutate**, and **summarize** data efficiently.
* **Visualization with ggplot2:** Briefly mention how ggplot2 helps create insightful **visualizations** like scatter plots, bar charts, and histograms.
* **Statistics in R:** Touch on how basic statistics (mean, median, etc.) and inferential statistics (e.g., t-tests) are used to analyze and draw conclusions from data.

**Guided Demo (35 minutes)**

Students will apply their knowledge of data manipulation, visualization, statistical analysis, and modeling to the WHO Life Expectancy dataset. This hands-on session will demonstrate the use of operators, data structures, missing data handling, data visualization, statistical analysis, and regression modeling in R.

**1. Setting up the Environment**

* Import necessary libraries and load the dataset:
* library(tidyverse)
* library(lubridate)
* library(skimr)
* life\_exp <- read.csv("Life\_Expectancy\_Data.csv")

**2. Initial Exploration of the Dataset**

* Explore the structure and summary of the dataset:
* str(life\_exp) # View the structure of the dataset
* head(life\_exp) # Preview the first few rows
* summary(life\_exp) # Summary statistics of the dataset
* skim(life\_exp) # Detailed skim of the dataset
* Check for missing data:
* missing\_data <- colSums(is.na(life\_exp))
* missing\_percent <- (missing\_data/nrow(life\_exp))\*100
* missing\_df <- data.frame(
* variable = names(missing\_data),
* missing\_percent = missing\_percent
* )
* Visualize missing data:
* ggplot(missing\_df, aes(x = reorder(variable, missing\_percent),
* y = missing\_percent)) +
* geom\_bar(stat = "identity") +
* coord\_flip() +
* theme\_minimal() +
* labs(title = "Percentage of Missing Values by Variable",
* x = "Variables", y = "Missing Percentage")
* **Discussion:** Explain the importance of understanding missing data and how it can impact analysis.

**3. Data Cleaning and Transformation**

* Convert Year column to Date format, create new variables:
* life\_clean <- life\_exp %>%
* mutate(
* Year = as.Date(paste0(Year, "-01-01")), # Convert Year to Date format
* Development\_Status = ifelse(Status == "Developing", "Developing", "Developed"), # Create a Development Status variable
* GDP\_per\_capita = GDP / Population # Create new meaningful variable for GDP per capita
* ) %>%
* filter(!is.na(Life.expectancy)) # Remove rows with missing life expectancy values
* Group by Country and Development Status to calculate summary statistics:
* country\_stats <- life\_clean %>%
* group\_by(Country, Development\_Status) %>%
* summarise(
* Avg\_Life\_Exp = mean(Life.expectancy),
* Avg\_GDP\_per\_capita = mean(GDP\_per\_capita),
* Avg\_Schooling = mean(Schooling, na.rm = TRUE),
* Avg\_BMI = mean(BMI, na.rm = TRUE)
* ) %>%
* arrange(desc(Avg\_Life\_Exp))
* **Discussion:** Demonstrate how we can clean the data and create meaningful new variables.

**4. Data Visualization**

* Time Trend Plot (Life Expectancy by Development Status):
* yearly\_trends <- life\_clean %>%
* group\_by(Year, Development\_Status) %>%
* summarise(
* Avg\_Life\_Exp = mean(Life.expectancy),
* Avg\_GDP\_per\_capita = mean(GDP\_per\_capita),
* Avg\_Alcohol = mean(Alcohol, na.rm = TRUE),
* Avg\_BMI = mean(BMI, na.rm = TRUE)
* ) %>%
* ungroup()
* time\_plot <- ggplot(yearly\_trends,
* aes(x = Year, y = Avg\_Life\_Exp, color = Development\_Status)) +
* geom\_line(linewidth = 1) +
* geom\_point() +
* theme\_minimal() +
* labs(title = "Life Expectancy Trends Over Time",
* subtitle = "Comparing Developed vs Developing Countries",
* x = "Year", y = "Average Life Expectancy", color = "Development Status") +
* theme(legend.position = "bottom")
* print(time\_plot)
* Scatter Plot (Life Expectancy vs GDP per Capita):
* scatter\_plot <- ggplot(life\_clean,
* aes(x = GDP\_per\_capita, y = Life.expectancy, color = Development\_Status)) +
* geom\_point(alpha = 0.6) +
* scale\_x\_log10() +
* geom\_smooth(method = "lm", se = FALSE) +
* theme\_minimal() +
* labs(title = "Life Expectancy vs GDP per Capita",
* x = "GDP per Capita (log scale)", y = "Life Expectancy", color = "Development Status")
* print(scatter\_plot)
* Box Plot (Life Expectancy by Development Status):
* box\_plot <- ggplot(life\_clean,
* aes(x = Development\_Status, y = Life.expectancy, fill = Development\_Status)) +
* geom\_boxplot() +
* geom\_jitter(alpha = 0.1) +
* theme\_minimal() +
* labs(title = "Life Expectancy Distribution by Development Status",
* x = "Development Status", y = "Life Expectancy")
* print(box\_plot)
* **Discussion:** Explain how these visualizations provide insights into life expectancy trends and the relationship with GDP.

**5. Statistical Analysis: t-Test**

* **Hypothesis:** Do developed countries have significantly higher life expectancy than developing countries?
* Perform a t-test:
* t\_test\_result <- t.test(Life.expectancy ~ Development\_Status, data = life\_clean)
* print(t\_test\_result)
* **Discussion:** Explain the results of the t-test and interpret the p-value.

**6. Correlation and Regression Analysis**

* Correlation Matrix:
* correlation\_matrix <- life\_clean %>%
* select(Life.expectancy, GDP\_per\_capita, Schooling, BMI) %>%
* cor(use = "complete.obs")
* print(correlation\_matrix)
* Linear Regression Model: Predict life expectancy using GDP per capita, schooling, and BMI:
* model <- lm(Life.expectancy ~ GDP\_per\_capita + Schooling + BMI, data = life\_clean)
* summary(model)
* **Discussion:** Explain how the regression model works and the importance of each variable.

**Challenge (20 minutes)**

**Students Activity (15 minutes)**

In this challenge, you will apply your data manipulation, visualization, and basic statistical analysis skills to the World Happiness Report 2023 dataset. You will clean and prepare the data, create visualizations, and perform a basic statistical analysis to compare happiness scores across high and low GDP countries.

**Database:**[**World Happiness Report 2023**](https://www.kaggle.com/datasets/ajaypalsinghlo/world-happiness-report-2023)

The World Happiness Report uses data from the Gallup World Poll to estimate happiness scores and rankings across 6 key factors: economic production, social support, life expectancy, freedom, absence of corruption, and generosity. These factors contribute to making life evaluations higher or lower in each country compared to Dystopia, a hypothetical baseline country with the world's lowest national averages for each factor.

**Challenge Tasks:**

1. **Data Exploration and Cleaning :**
   * **Explore the dataset:** Load the dataset and check its structure, summary statistics, and the first few rows.
   * **Create a GDP category:** Classify countries into **High GDP** or **Low GDP** based on whether their Logged.GDP.per.capita is above or below the median value.
     + **Hint:** Use the **median()** function to find the median GDP, and **ifelse()** to categorize the countries.
   * **Clean the data:** Remove any rows where the happiness score Ladder.score is missing
2. **Data Summarization**
   * **Calculate average happiness scores:** Group the dataset by **GDP category** (high vs. low GDP) and calculate the average happiness score Ladder.score for each group.
3. **Data Visualization**
   * **Create a box plot:** Create a box plot that compares the happiness scores Ladder.score between high and low GDP countries. Use ggplot2 to create the plot.
4. **Statistical Analysis**
   * **Perform a t-test:** Perform a t-test to compare the average happiness scores between high and low GDP countries. Interpret the result briefly (focus on the p-value).

**Debrief (5 minutes)**

1. **Data Exploration and Cleaning :**

* Check the structure of the data and look at the first few rows and summary statistics:
* str(happiness\_data)
* head(happiness\_data)
* summary(happiness\_data)
* skim(happiness\_data)
* **Create a GDP category:** We calculate the median of Logged.GDP.per.capita and use ifelse() to classify the countries into High GDP or Low GDP.
* median\_gdp <- median(happiness\_data$Logged.GDP.per.capita, na.rm = TRUE)
* happiness\_data$GDP\_category <- ifelse(happiness\_data$Logged.GDP.per.capita > median\_gdp, "High GDP", "Low GDP")
* **Clean the data** by removing rows with missing happiness scores:
* happiness\_data\_clean <- happiness\_data %>%
* filter(!is.na(Ladder.score))

1. **Data Summarization**

* **Calculate average happiness scores** for each GDP category:
* summary\_stats <- happiness\_data\_clean %>%
* group\_by(GDP\_category) %>%
* summarise(avg\_happiness = mean(Ladder.score, na.rm = TRUE))
* print(summary\_stats)
  + **Explanation:** This code calculates the average happiness score for both **High GDP** and **Low GDP** countries. The group\_by() function groups the data based on GDP\_category, and summarise() computes the average happiness score for each group.

1. **Data Visualization**

* **Create a box plot** to compare happiness scores across the two GDP categories:
* ggplot(happiness\_data\_clean, aes(x = GDP\_category, y = Ladder.score, fill = GDP\_category)) +
* geom\_boxplot() +
* theme\_minimal() +
* labs(title = "Happiness Score by GDP Category", x = "GDP Category", y = "Happiness Score")
  + **Explanation:** This box plot visualizes the distribution of happiness scores for each GDP category. The plot helps us identify whether there is a significant difference in happiness scores between high and low GDP countries. The fill aesthetic colors the boxes by GDP category.

1. **Statistical Analysis**

* **Perform a t-test** to compare the average happiness scores between the two GDP categories:
* t\_test\_result <- t.test(Ladder.score ~ GDP\_category, data = happiness\_data\_clean)
* print(t\_test\_result)
  + **Explanation:** The t-test compares the means of happiness scores between high and low GDP countries. The output will provide the t-statistic, degrees of freedom, and the p-value. If the p-value is less than 0.05, it indicates a statistically significant difference in happiness scores between the two groups.

**Discussion**

Let's dedicate these final minutes to address any lingering questions about today's concepts. This is an opportunity for students to seek clarification on topics that may need reinforcement. Encourage students to share any points of confusion or request additional examples if needed. As we close the class, ensure that key concepts are solidified and students feel confident in their understanding of the material covered today. This open discussion format helps identify any common areas of misunderstanding that might need to be revisited in future sessions and allows for a collaborative learning environment where students can benefit from their peers' questions and insights.