**Week 1: AI-Assisted Code Generation & Analysis**

**1. Overview:**

This session introduces students to AI-assisted programming and its applications in data science. Students will learn how to generate Python and R scripts using AI tools (e.g., ChatGPT), analyze AI-generated outputs, and critically evaluate AI’s effectiveness in data science workflows.

**2. Learning Goals & Key Principles:**

By the end of this session, students will be able to:

* Understand and apply **effective prompt engineering** to generate AI-assisted code.
* Evaluate AI-generated scripts for **accuracy, efficiency, and best practices**.
* Identify **common AI-generated mistakes** in data science workflows.
* Develop a **structured approach** to analyzing and improving AI-generated code.

Key principles include:

* **AI as an Assistant, Not a Replacement** – Understanding AI’s role in coding.
* **Structured Prompting** – How different prompts affect AI outputs.
* **Critical Evaluation** – Analyzing AI-generated scripts before debugging.

**3. Mini-Lecture: Introduction to AI-Assisted Coding (Step-by-Step)**

**(15 min total)**

1. **What is AI-Assisted Programming?** (5 min)
   * Overview of AI-generated code and its role in data science.
   * Demonstration of how AI tools generate code based on user prompts.
2. **Prompt Engineering Techniques** (5 min)
   * Example of well-structured vs. poorly structured prompts.
   * Live demonstration of prompting AI to generate a simple script.
3. **Assessing AI Code Outputs** (5 min)
   * How to analyze AI-generated scripts for correctness and efficiency.
   * Discussion of common pitfalls and oversights in AI-generated code.

**4. Hands-on Activities: AI Code Generation & Evaluation**

**(60 min total, divided into three core tasks)**

**Task 1: Descriptive Statistics & Data Cleaning (20 min)**

* Students **ask AI to generate a script** that loads a dataset, calculates summary statistics, and handles missing values.
* Students **run and test the AI-generated script** to check for errors.
* **Discussion Questions:**
  + Did AI handle missing values correctly?
  + Were all summary statistics relevant and accurate?
  + How could the prompt be improved to get better AI results?

**Task 2: Data Visualization (20 min)**

* Students **ask AI to generate a script** that visualizes trends in the dataset.
* Students **evaluate the AI-generated visualizations** for clarity and effectiveness.
* **Discussion Questions:**
  + Did AI choose the right type of plot for the data?
  + Are the axis labels, titles, and legends clear and meaningful?
  + How can the visualization be improved?

**Task 3: Feature Engineering & Transformation (20 min)**

* Students **ask AI to generate a script** that creates new features and normalizes data.
* Students **analyze the AI-generated transformations** for appropriateness.
* **Discussion Questions:**
  + Did AI apply feature transformations logically?
  + Were unnecessary steps included?
  + How could the prompt be refined for a better result?

**5. Wrap-Up & Reflection (15 min)**

* **Group Discussion:** What worked well? What didn’t?
* **Key Takeaways:** AI-generated code requires structured prompts and critical evaluation.
* **Looking Ahead to Week 2:** Next session, students will focus on **debugging and improving faulty AI-generated scripts**.

**Outcome:**

By the end of this session, students will:

* Understand how to **effectively prompt AI** to generate useful code.
* Critically evaluate **AI-generated scripts** for correctness and efficiency.
* Recognize common AI errors and **anticipate debugging needs**.
* Develop structured problem-solving habits using **iterative refinement**.
* Lay the groundwork for **Week 2’s debugging and improvement exercises**