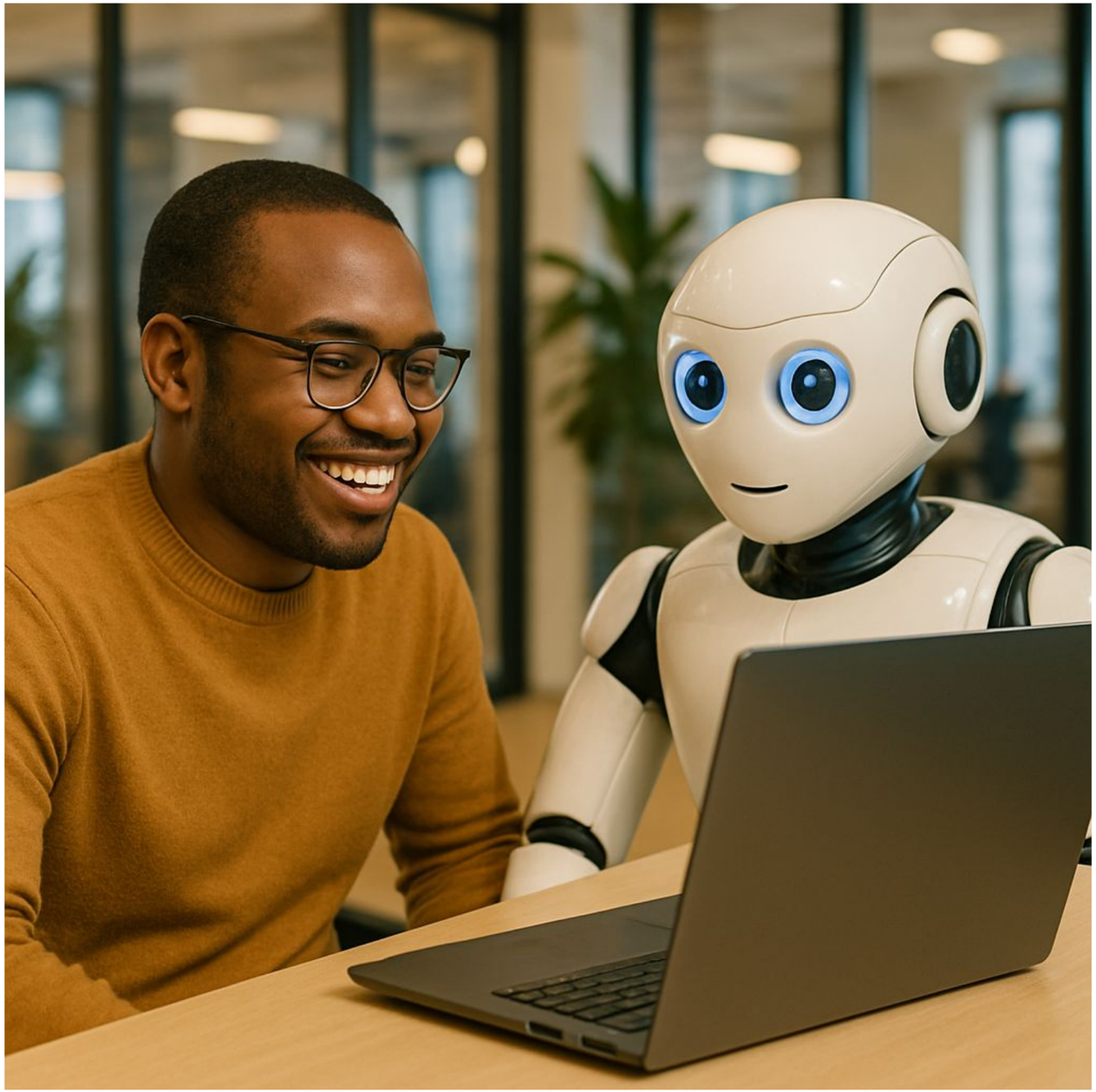
LLM-Driven Analysis and Reporting with R

A hands-on workshop exploring how Large Language Models can enhance statistical workflows.

Chester Ismay and Arturo Valdivia

WNAR 2025



UNIT A Connect & Primer

Welcome & Workshop Goals

Learn Prompting

Craft effective prompts for R code

GitHub Copilot

Integrate AI assistance into RStudio

Reporting

Build analyses and Quarto reports with LLM-generated elements

Ethics

Promote reproducibility and ethical use

Instructors’ Introduction

Chester Ismay

PORTLAND STATE UNIVERSITY

Arturo Valdivia

INDIANA UNIVERSITY

Participant Snapshot

Form Groups

Pairs or triplets

Round-Robin

Name, organization, desired ChatGPT task

Capture Goals

Note responses in Google Doc

Logistics & Tools

Room Essentials

- WiFi network & password
- Restrooms location
- Coffee station (Available at 3 PM)

Prerequisites

- Software setup (R & RStudio)
- Google Doc link: https://bit.ly/wnar-llm or use the QR code
  - Live updates
  - Post questions

Workshop Outcomes

Use Github Copilot

Scaffold analyses in RStudio

Craft Prompts

Effective ChatGPT prompts for R code

Build Reports

Quarto with LLM code and narrative

Best Practices

Reproducibility, auditing, ethics

Workshop Roadmap

LLM Theory

Fundamentals of language models

Copilot Setup

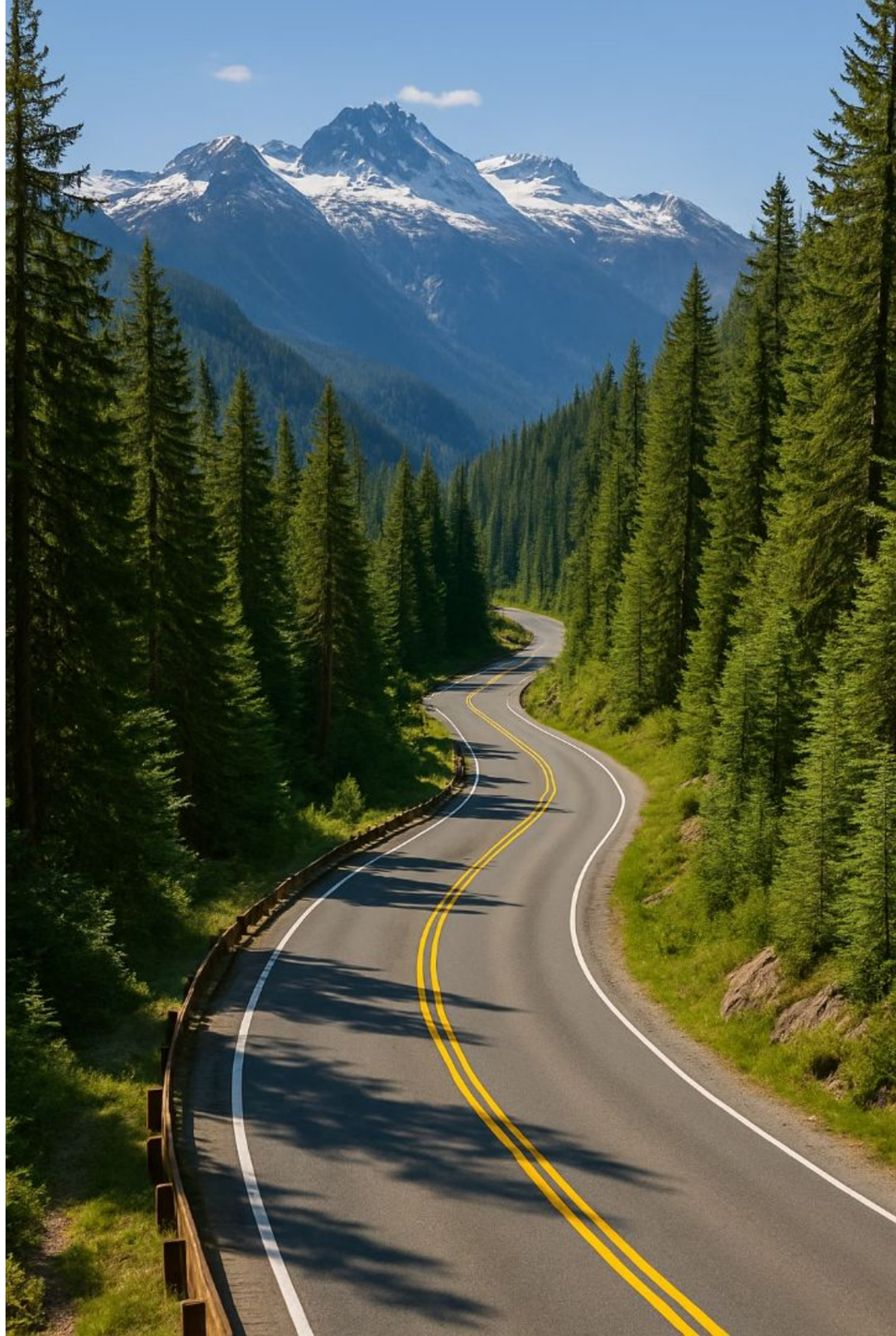
Integration with RStudio

Hands-on Prompts

Practice crafting effective prompts

Case Study & Pipeline

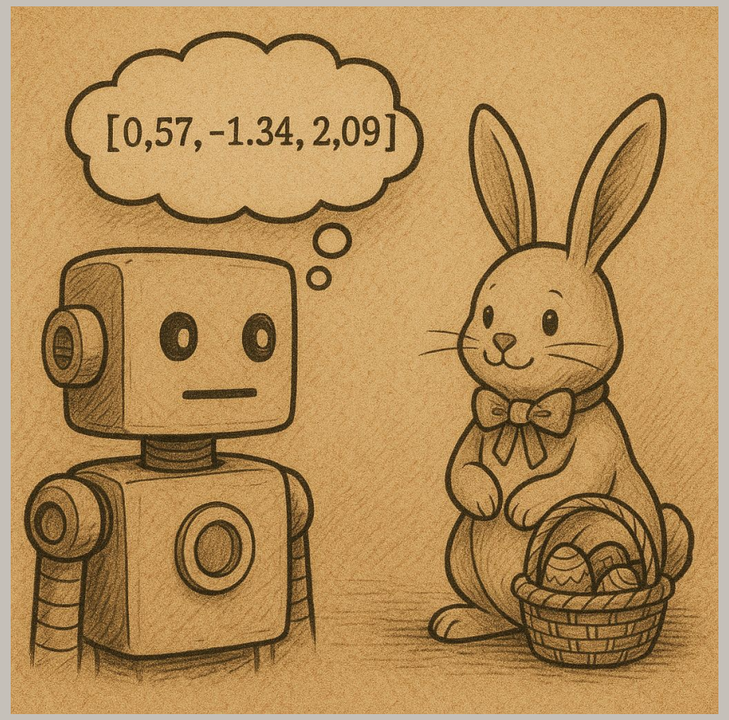
ModernDive example and automation



ONCE UPON A TEXT...

- Imagine teaching a child to speak using books alone
- No explanations, just reading patterns
- Transformers learn language the same way!

STEP 1: TURNING WORDS INTO MATH



Words become vectors

Similar meanings = closer vectors

Text becomes numerical data

STEP 2:
PAYING ATTENTION TO WHAT MATTERS

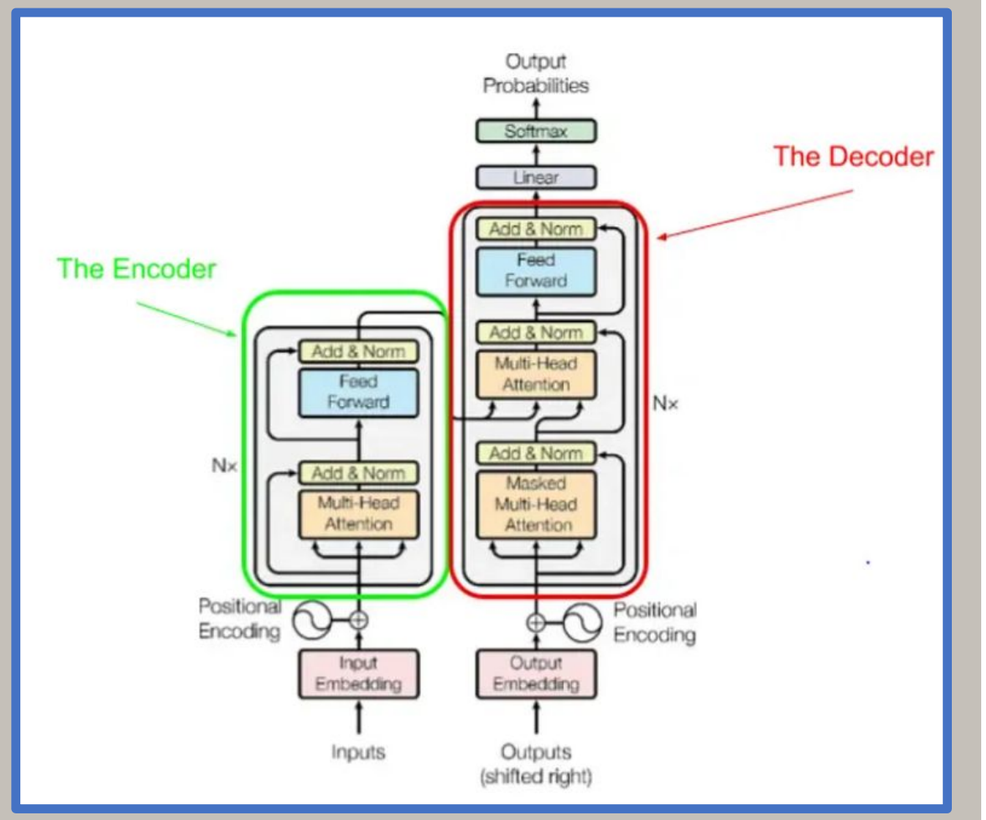
On a sunny spring morning, people walked through the beautiful city park.

STEP 2: PAYING ATTENTION TO WHAT MATTERS

On a sunny spring morning, people walked through the beautiful city park.

- Focus on important words
- Assign attention weights
- Handle long sentences

STEP 3: BUILDING A TRANSFORMER



Stacks of Attention + Feed-Forward layers

Modular design

Deeper layers = deeper understanding

STEP 4: TRAINING ON HUGE TEXT LIBRARIES

Billions of words

Predict next word repeatedly

Patterns emerge naturally

STEP 5: PREDICT, PREDICT, PREDICT!

Predict next word/token

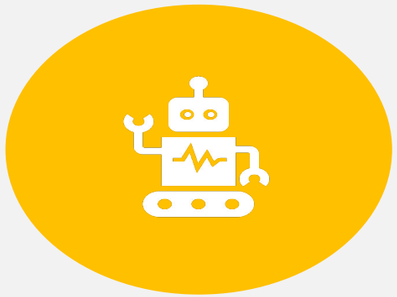
Assign probability scores

Sample or
choose highest probability

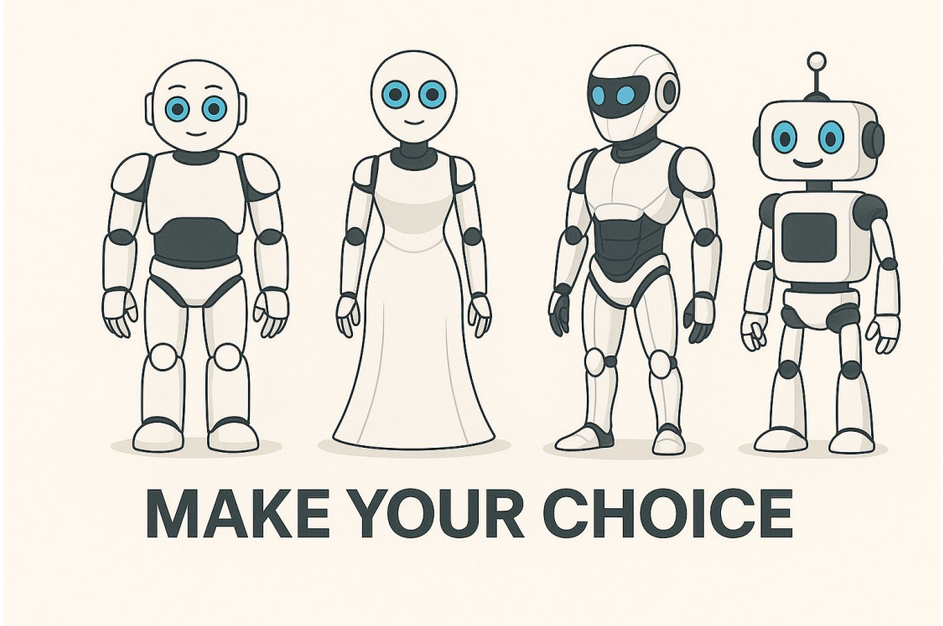
TALK LIKE A TRANSFORMER: KEY POINTS

WORDS BECOME MATH

ATTENTION DRIVES FOCUS



TRAINED ON PATTERNS, NOT LOGIC



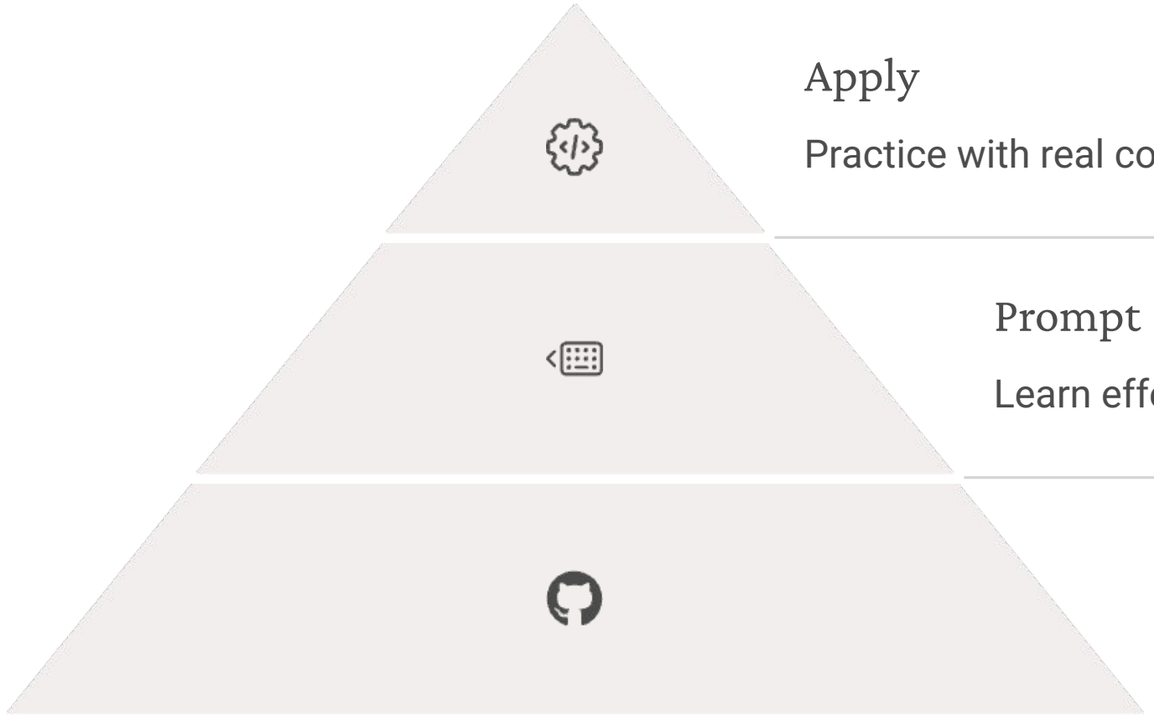
Hosted LLMs Overview (as of June 2025)

Model	Strengths	Limitations
ChatGPT	Wide adoption	Paywalled, not always real-time
Claude	Long context, reasoning	More cautious
Gemini	Google integration	Newer, less tested
DeepSeek	Efficiency	Privacy concerns

Break Time

UNIT B Configure & Prompt

Configure and Prompt Overview



Apply

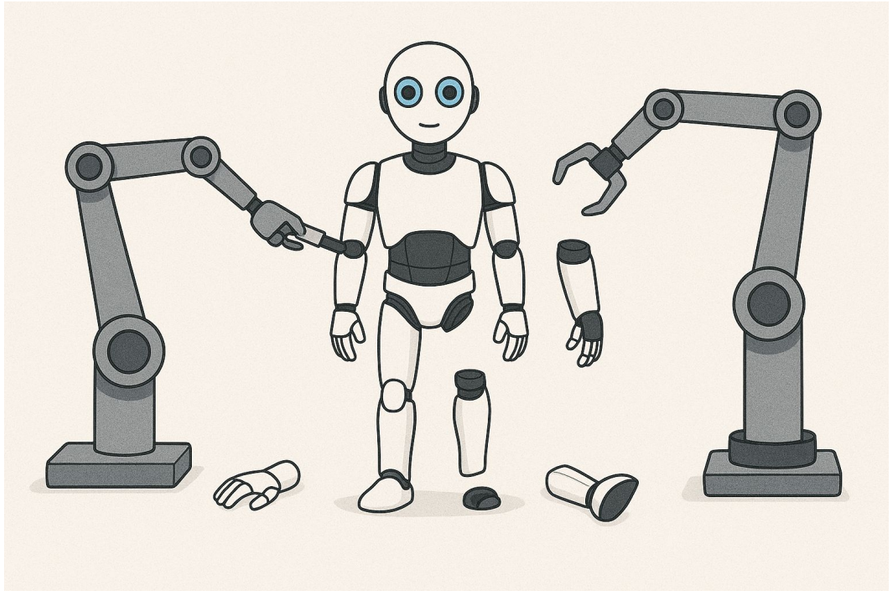
Practice with real code

Prompt

Learn effective techniques

Configure

Setup Copilot in RStudio



GitHub Copilot Setup

What is Copilot?

AI pair programmer for code completion

Benefits for R

Faster coding, fewer errors

How It Works

Suggests code based on comments

Copilot Account & Enrollment

GitHub Account

Sign up or verify existing

Copilot Enrollment

Visit Posit guide link (if needed)

Select Plan

Free trial for public users

Verify

Confirm license in settings

RStudio Integration

Update RStudio

Version 2024.10 or later (2025.05 is current version)

Enable Copilot

Tools → Global Options → GitHub Copilot

Authenticate

Connect via GitHub OAuth

Restart & Confirm

Verify Copilot text appears in new .R script in RStudio

Hands-On Code Practice

Live Prompt

Type comments in R script

Accept Suggestions

Use Tab to insert Copilot code

Iterate

Refine comments, observe changes

Share Experiences

Round-robin discussion

Code Comments to Try (One at a time)

```
# Install ggplot2 and moderndive packages


# Load the packages


# Plot waiting by duration for old_faithful_2024
```

ChatGPT for Code Generation

ChatGPT

Craft Prompt

Specific request for R code

Generate Code

ChatGPT creates solution

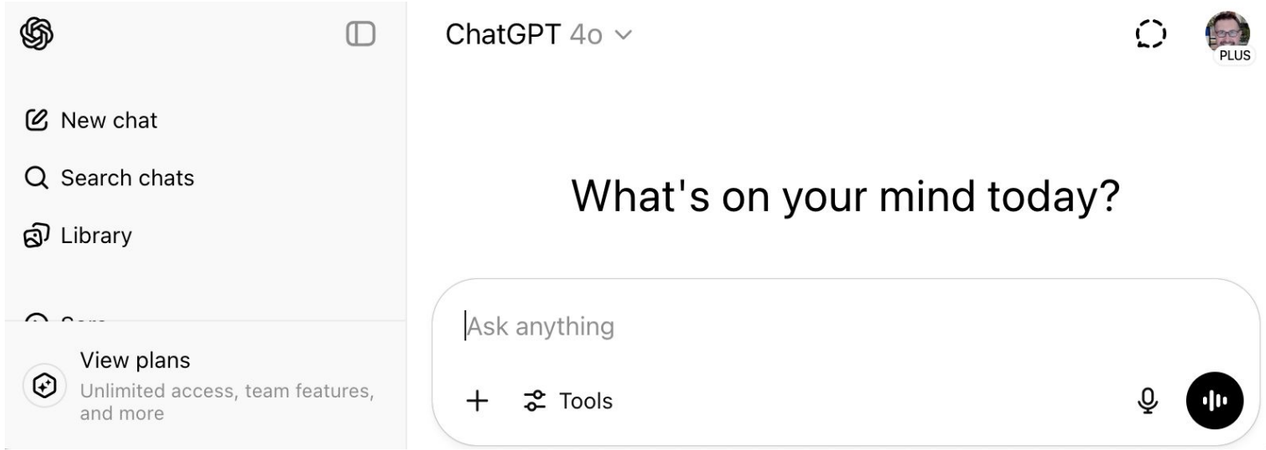
Test & Refine

Run code, iterate as needed

Copy Code

Transfer to RStudio

ChatGPT Interface Tour



Access

Create free account

Key UI Elements

Prompt box, message types

Code Formatting

Triple-backtick fences

Prompt Crafting Demo Review

Example Prompt

Violin plot of life expectancy by continent

Try also by region instead of continent

Using moderndive dataset

Student Exercise

Create top 10 population bar chart

Horizontal orientation

Descending order

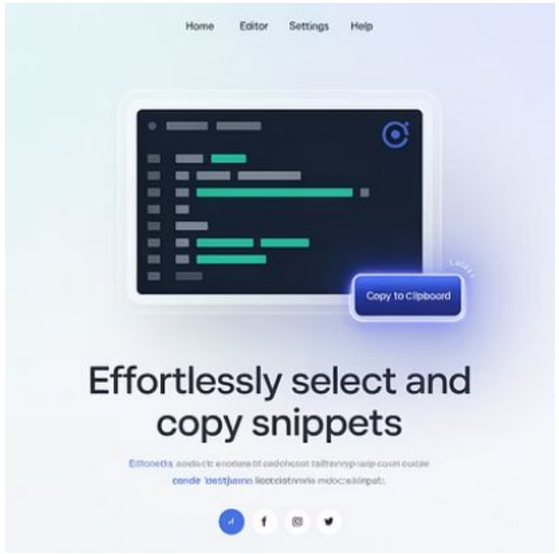
Best Practices

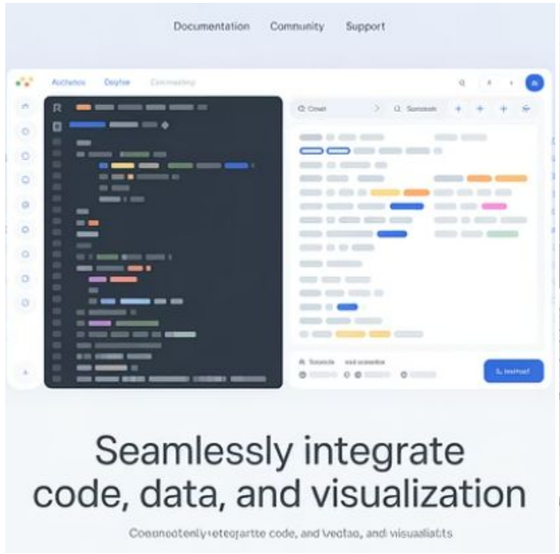
Specify language & dataset

Include column names

Request inline comments

Copy & Execute Workflow DEMO Review





You and Your AI + Human Collaborators

Form Teams

Work in small groups

Choose Dataset

`coffee_quality` or `spotify_by_genre`

Define Analysis

Set clear statistical goals


Teams & Dataset Selection

coffee_quality Dataset

- Bean characteristics
- Taste profiles
- Origin information

spotify_by_genre Dataset

- Musical attributes
- Popularity metrics
- Genre classifications

Analysis Goals

- Define clear objective
- Draft prompt together
- Plan initial exploration

AI-Assisted Analysis

ChatGPT Path

Paste prompt, copy results

Transfer to RStudio

Execute and verify

Copilot Path

Write comments in script

Accept code suggestions

Run incrementally

Human Oversight

Verify outputs

Check statistical validity

Guide the AI

Reflect & Share

What Worked

Successful AI assistance examples

Challenges

Where AI suggestions missed

Improvements

Prompt refinement ideas

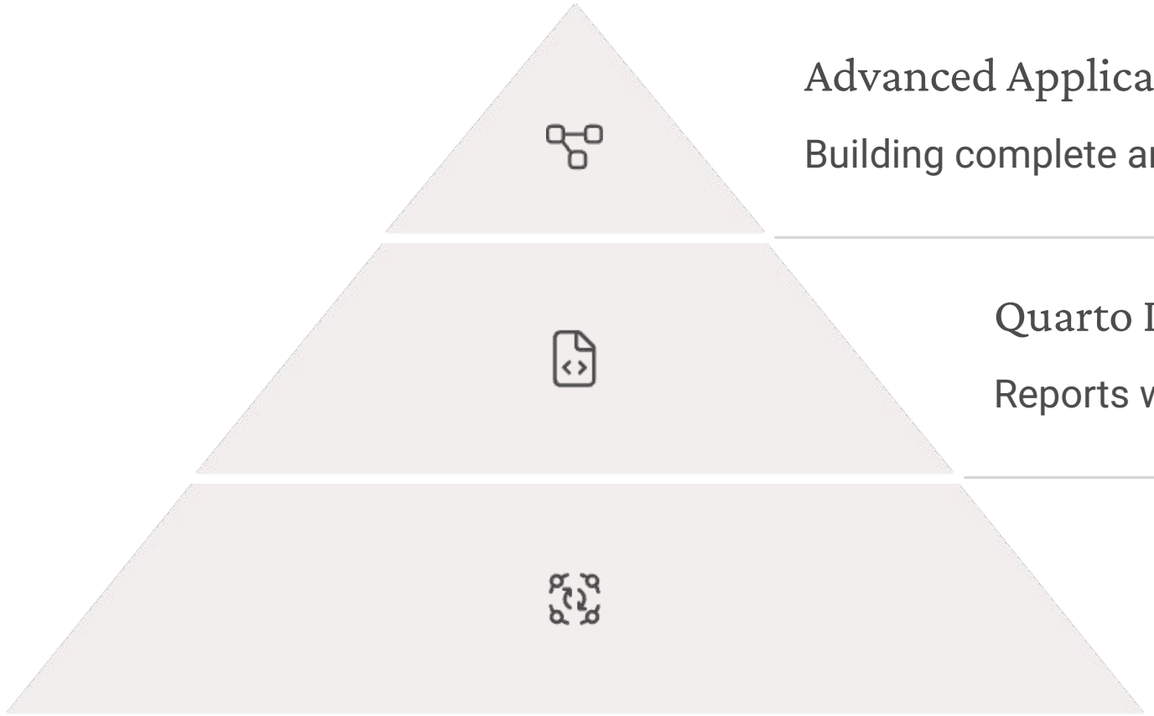
Documentation

Capture in Google Doc

Break Time

UNIT C More AI Collaboration

More AI Collaboration Overview



Advanced Applications

Building complete analyses

Quarto Integration

Reports with LLM elements

Workflow Refinement

Optimizing AI collaboration

Review of Units A and B

Workshop Goals

Effective prompts, Copilot use, Quarto reports

Practical Tools

Copilot setup, ChatGPT workflow

LLM Foundations

Transformers, attention, model types

Collaborative Work

Team exercises, dataset exploration

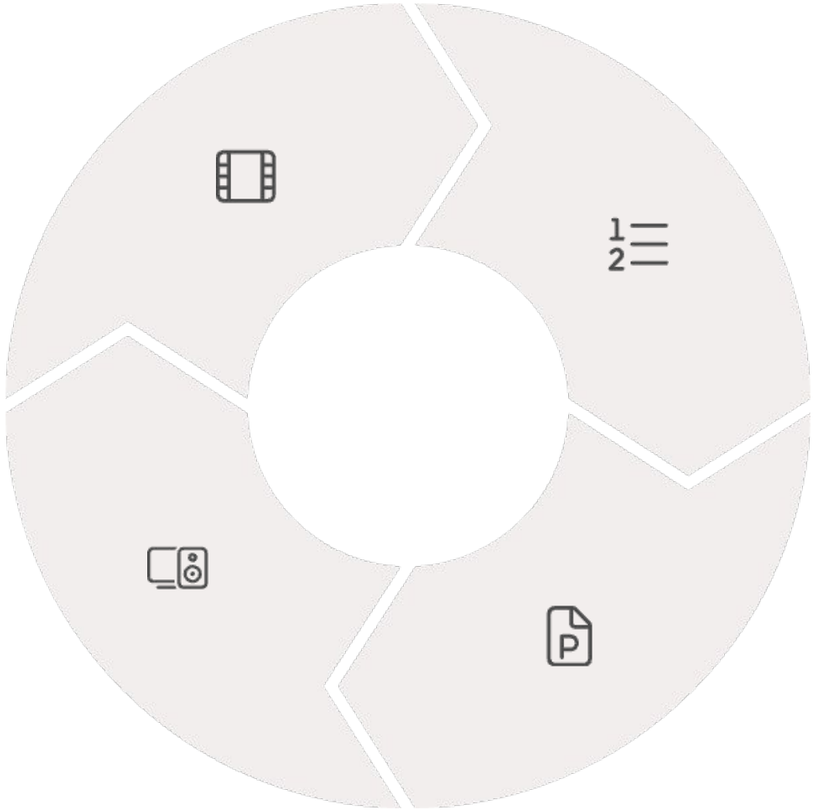
Exploring AI Limitations and Successes

Literate Programming

Quarto fundamentals

Document Assumptions

Statistical validity checks



Replication

Test AI reproducibility

Create Reports

Structure findings clearly

Literate Programming Refresher

Quarto Anatomy

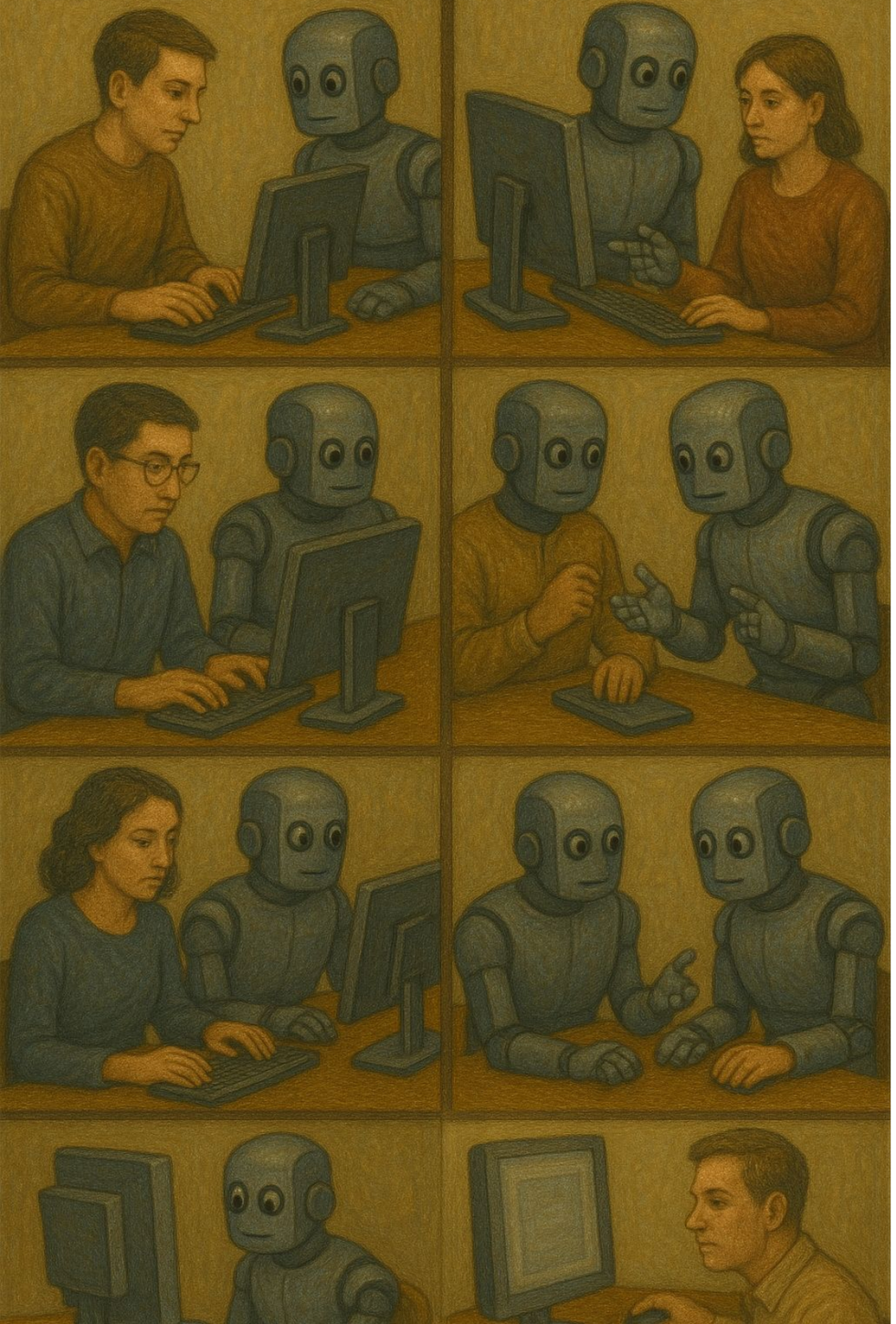
- Code chunks
- Narrative text
- YAML front matter

Chunk Options

- #| echo: true
- #| eval: true
- #| include: false

Live Demo

- Create mini .qmd
- Add code + text
- Render document



Automating with Copilot

Data Import

Load and clean datasets

EDA

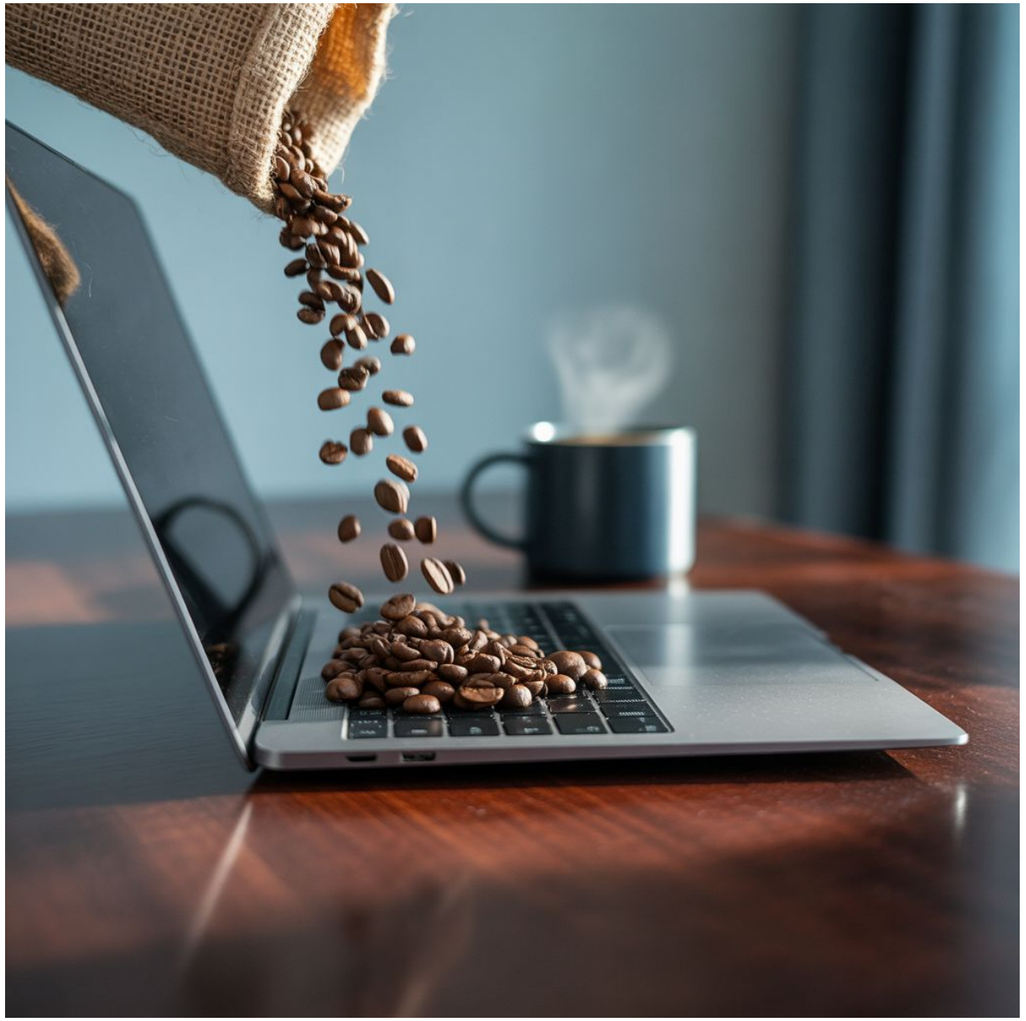
Create statistical overviews and plots

Modeling

Generate regression and testing code

Custom Functions

Build reusable components



Hands-On: Data Load

```
# Load coffee_quality data from moderndive package


# Preview first 10 rows
```

Hands-On: Exploratory Data Analysis

```
# Create a summary table of mean and SD for
# all numeric columns
```

```
# Generate side-by-side boxplot of acidity by continent
```

Hands-On: Modeling & Inference

```
# 1. Fit a linear regression: acidity ~ aroma


# 2. Get results using the three
# moderndive::get_regression functions
```

```
# Use the infer R package to perform a two-sample
# permutation test on total_cup_points for
# Asian vs North American countries
```

Hands-On: Custom Functions

```
# Write a function corr_matrix(df) that

# selects only numeric columns,

# computes pairwise Pearson correlations, and

# returns a tidy tibble with Var1, Var2, and correlation
```

Hands-On (Extension): Correlation Heatmap

```
# Create a heatmap from the correlation matrix call
```

Designing a Replication Prompt

Draft Prompt

Clear instructions to reproduce analysis

Workflow Plan

Systematic prompt-code-verify cycle

Version Control

Document prompts alongside code

Evaluating AI Alignment (Example)

Component	Your Code	AI Code	Alignment
Data Prep	Complete	Missing steps	Partial
Model Parameters	Optimized	Default	Fair
Visualization	Custom theme	Standard	Work Needed

Prompt Refinement

Identify Issues

Missing libraries, wrong filters

Add Details

Specify packages, column names

Regenerate

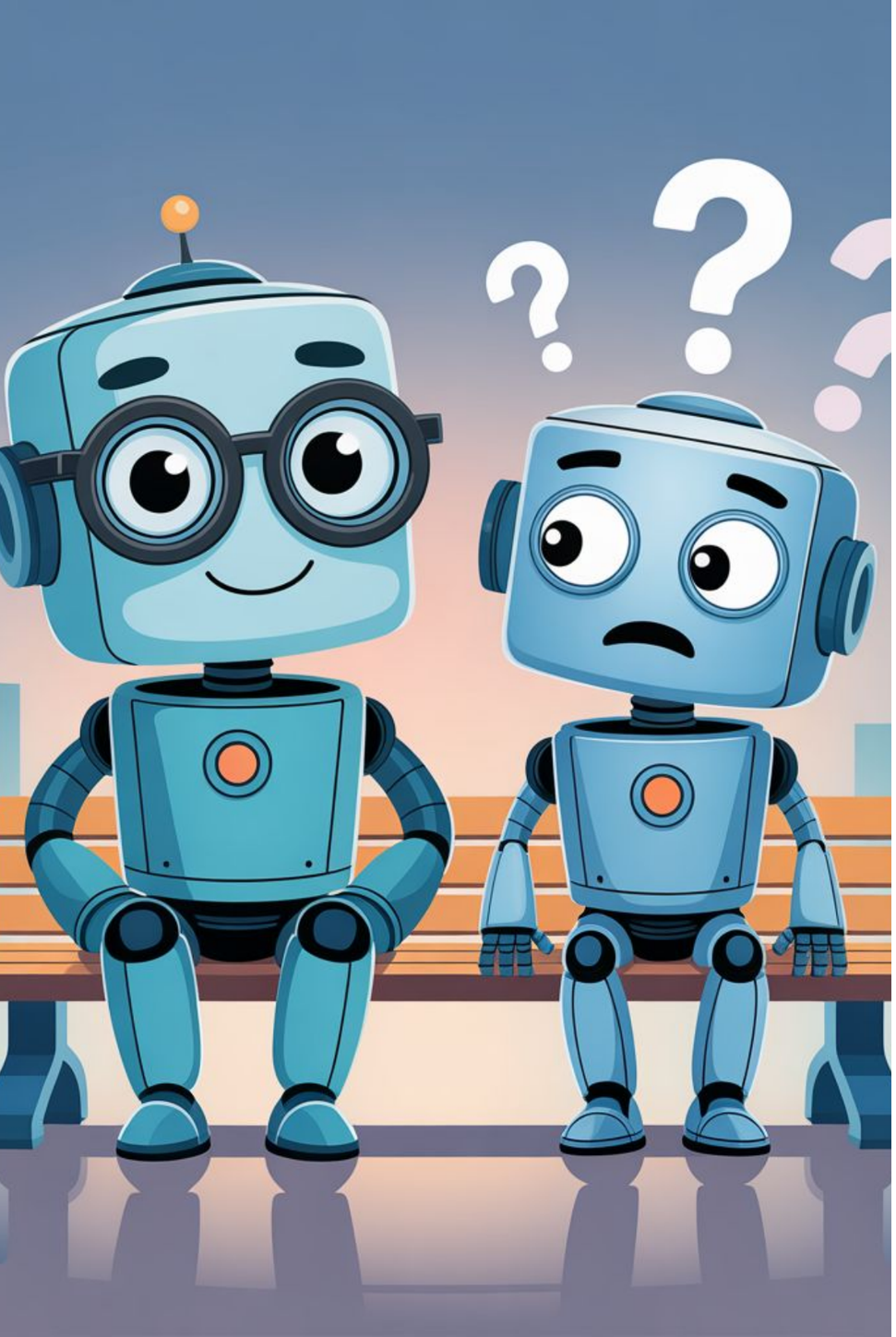
Get improved code output

Verify

Confirm matches expected result

Break Time

UNIT D Summary + Wrap-Up



Pitfalls, Best Practices, and Ethics

Identify Hallucinations

Systematic fact-checking

Verify Against Sources

CRAN docs, peer-reviewed articles

Ask for Citations

Request verifiable information

Maintain Fact-Check Section

Log corrections in QMD

Prompt-Logging Practices

Record Prompts

Comment before each LLM-driven chunk

Version Iterations

Track improvements with Git branches

Timestamp Entries

Enable traceability to model version

Document Changes

Note refinements between attempts

What About Reproducibility?

Declare Dependencies

Document package versions

Set Random Seeds

Fixed values before sampling

Archive Conversations

Save LLM logs with reports

Enable Replay

Allow auditors to recreate steps

Course Review & Next Steps

Continue Learning

Apply tools to your projects

Stay Connected

Share contact info (if you wish)

Thank you!

Additional resources

- YouTube links
  - Transformers explained visually
  - Attention in transformers
  - How might LLMs store facts
- Transformer Explainer

ModernDive v2 textbook

ModernDive