COURSE OVERVIEW DSC 205

ADVANCED INTRODUCTION TO DATA SCIENCE - SPRING 25

Marcus Birkenkrahe

January 13, 2025

What is an "Advanced intro to data science"?

Figure 1: Map of the Lonely Mountain (Tolkien, The Hobbit)

What's your interpretation of this map in the context of our course?

- The map is part of a greater map of Middle Earth ($< 10\%$).
- A map is one of many **metaphors** for learning (see below).

- Different people have different **external** maps (see below).
- To learn anything you need to develop an **internal** map.
- Your internal map should be **personal**, **professional** & **passionate**.
- Personal e.g. *"I really like high performance code."*
- Professional e.g. *"Better business decisions based on data."*
- Passionate: *"I'm fascinated by understanding human behavior through data."*

Data Science through Metaphor (A word from our sponsor)

Metaphors can be a useful way of approaching a journey when you don't really know yet where it is going or what you need to undertake it.

The Quarry: The Data Science Mountain

The journey to mastering Data Science can be seen as climbing a mountain, with different stages of learning represented by different levels on the mountain.

1. Basecamp: Foundational Skills
2. Mathematics (Probability, Statistics, Linear Algebra)
3. Programming (Python, R)
4. Data Management (SQL)
5. Mid-Level Trails: Data Wrangling and EDA
6. Data Cleaning and Preparation
7. Exploratory Data Analysis (EDA)
8. Data Visualization (ggplot2, Matplotlib, Seaborn)
9. Summit: Real-World Applications
10. Machine Learning (Supervised, Unsupervised Learning)
11. Deep Learning and Natural Language Processing
12. Ethical AI and Decision Science

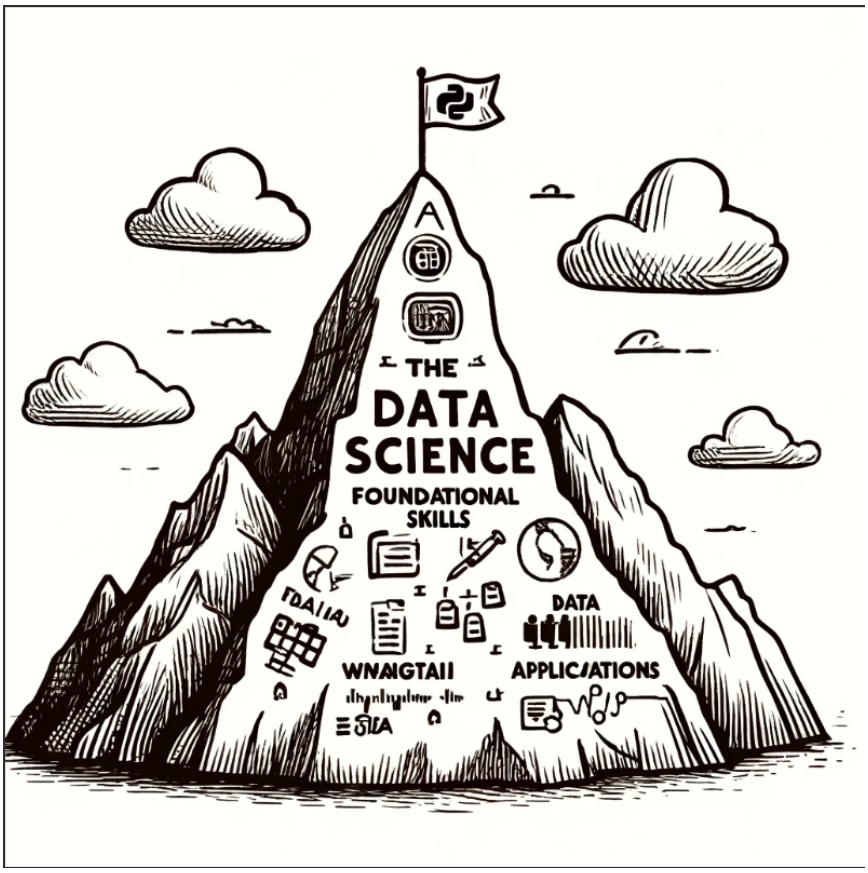

Figure 2: Data Science Mountain

Query: The Data Science Compass

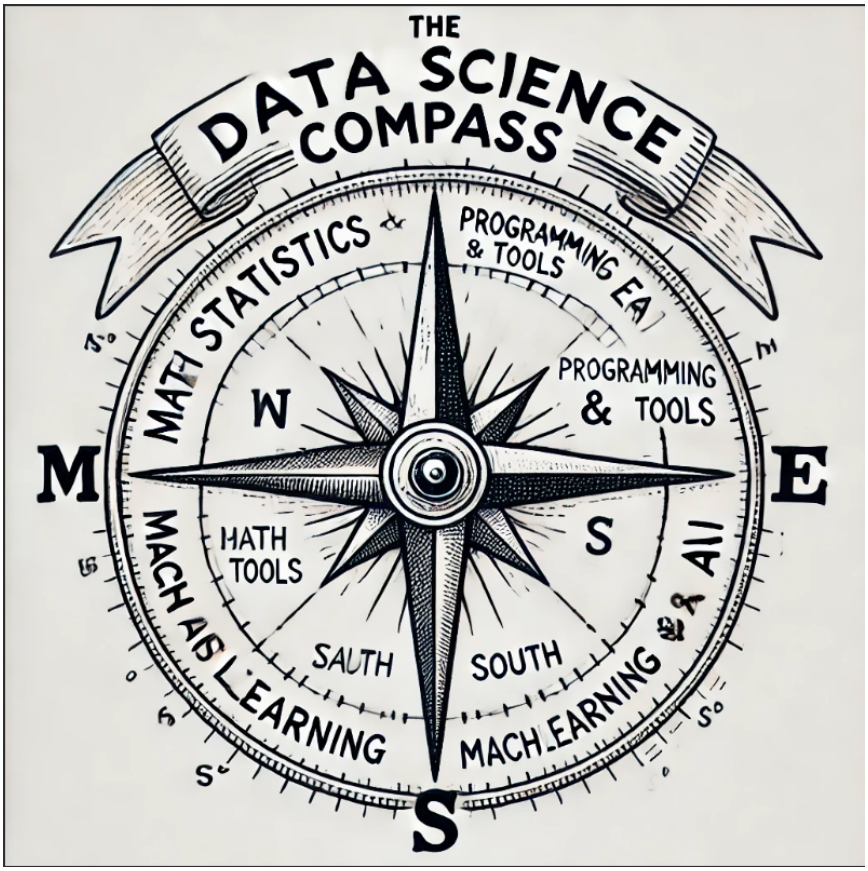

Figure 3: Data Science Compass

A compass can help guide learners through the core areas of Data Science, ensuring balanced growth in four key directions.

Cardinal Directions:

- **North:** Math & Statistics
- **East:** Programming & Tools (Python, R, SQL, Git)
- **South:** Data Analysis & Visualization (EDA, Reporting, Dashboards)
- **West:** Machine Learning & AI (Supervised Learning, Deep Learning, NLP)

The Quest: The Data Science Map of Realms

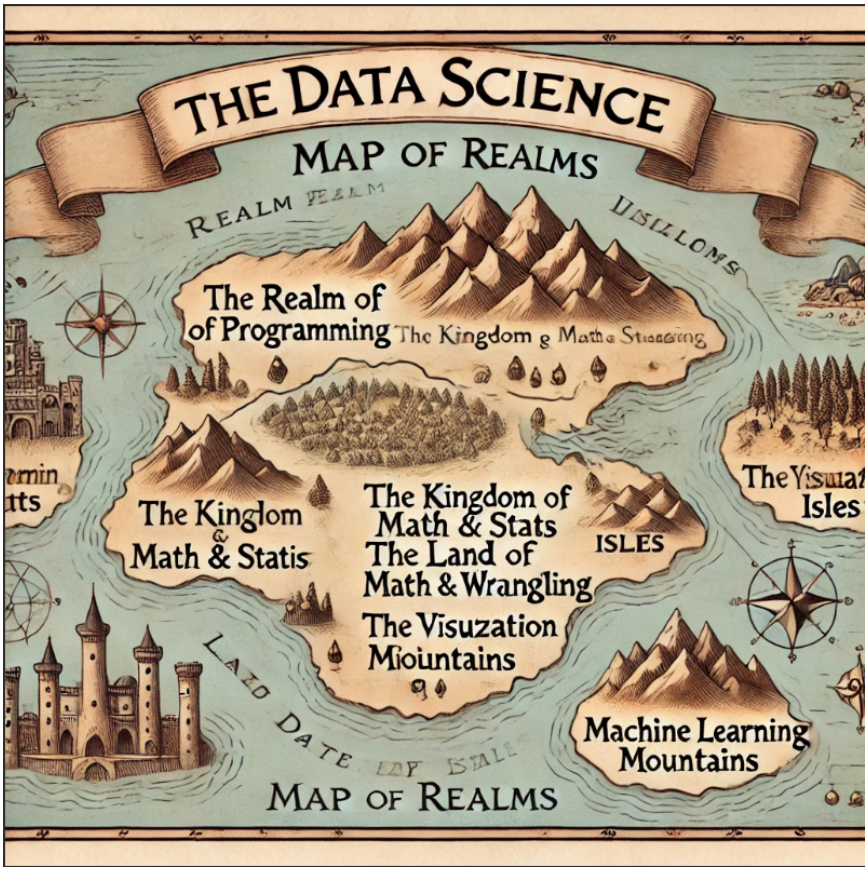

Figure 4: Data Science Map

Imagine Data Science as a fantasy world, divided into distinct regions, each representing a key area of learning.

Regions:

- The Realm of Programming: Python, R, SQL, Git
- The Kingdom of Math & Stats: Probability, Hypothesis Testing, Linear Algebra
- The Land of Data Wrangling: Pandas, Data Cleaning
- The Visualization Isles: Matplotlib, Seaborn, Plotly, ggplot2

- The Machine Learning Mountains: Supervised Learning, Deep Learning, Natural Language Processing
- The Ethics Forest: Data Privacy, Fairness, Bias

Data science as a Journey to the "Lonely Mountain"

Figure 5: Thorin Oakenshield's Company (The Hobbit)

Thorin Oakenshield and Company:

Table 1: The Hobbit vs. This course

"The Hobbit"	This course
Treasure hunters	Big Data
Skilled experts	Advanced Data science
Magic ring owners	R + Emacs + ESS + Org-mode
Fighters against evil	Statistics for good or evil
Band of brothers	Agile Scrum projects

What will you do in this course?

- Part 1: R: Control, iteration, functions, utilities (5 weeks)
- Part 2: Data processing: bash, Python, and SQL (4 weeks)

- Part 3: Writing functions like a champion: R scripting (3 weeks)
- Part 4: Optimizing R code with Rcpp (R and C++) (3 weeks)
- Part 5: Agile group projects: Application or package (15 weeks)

What will we not do in this course?

Table 2: Topics not part of this course

Topic	Course	See also
Advanced graphics and EDA	DSC 302	
Machine learning (classical)	DSC 305	
Statistics and probability	DSC 482.01	MTH/BUS 230
Testing and modeling	DSC 482.02	
Deep learning (neural nets)	DSC 482.03	CSC 482

- The 482 courses are special topics offered at our whim.
- Dr. Dall'Olio will teach two 482 classes in Fall 25
- I will teach Computer Architecture and Assembly

How will you be evaluated?

Table 3: Course evaluation (see Syllabus)

When	Description	Impact
Weekly	Assignments	25%
Weekly	Multiple choice tests	25%
Monthly	Project sprint reviews	25%
TBD	Final exam (optional)	25%

- The course carries 4 credits (lab overhead).
- All course requirements have deadlines.
- Late submissions will be penalized (point loss).

- Final exam (optional) will be sourced by term test questions.
- The project topic can come from any of the course sub-topics.
- The project deliverable is a working **literate** program.

How much work will you have to put in?

Workload (estimated):

- Time in class: 48 hrs.
- Time outside of class: 42 hrs.
- Time for tests [1 hrs/test]: 14 hrs.
- Time for home assignments [2 hrs/pgm]: 28 hrs.
- Total number of hrs in term: 90.
- Weekly workload (outside of class): 5.625 (2.625) hrs
- Daily workload outside of class in minutes: 25
- Grade expectations: A-B ($>$ 2 hrs/week), C-D (1-2 hr/week)

What are "sprint reviews"?

- Scrum is an important software engineering technique.
- IMRaD is an important framework to publish scientific papers.
- DevOps relate to the interface between software development and IT operations as data projects scale and become more complex.
- Modern ML workflows are highly layered and infrastructure-heavy (cp. the editorial by Andrew Ng shared in the Google Chat).

Scrum Project Structure (Monthly assignments)

Sprint 1: Introduction (Project Idea)

- Students present the problem they want to solve, dataset description, and potential impact.

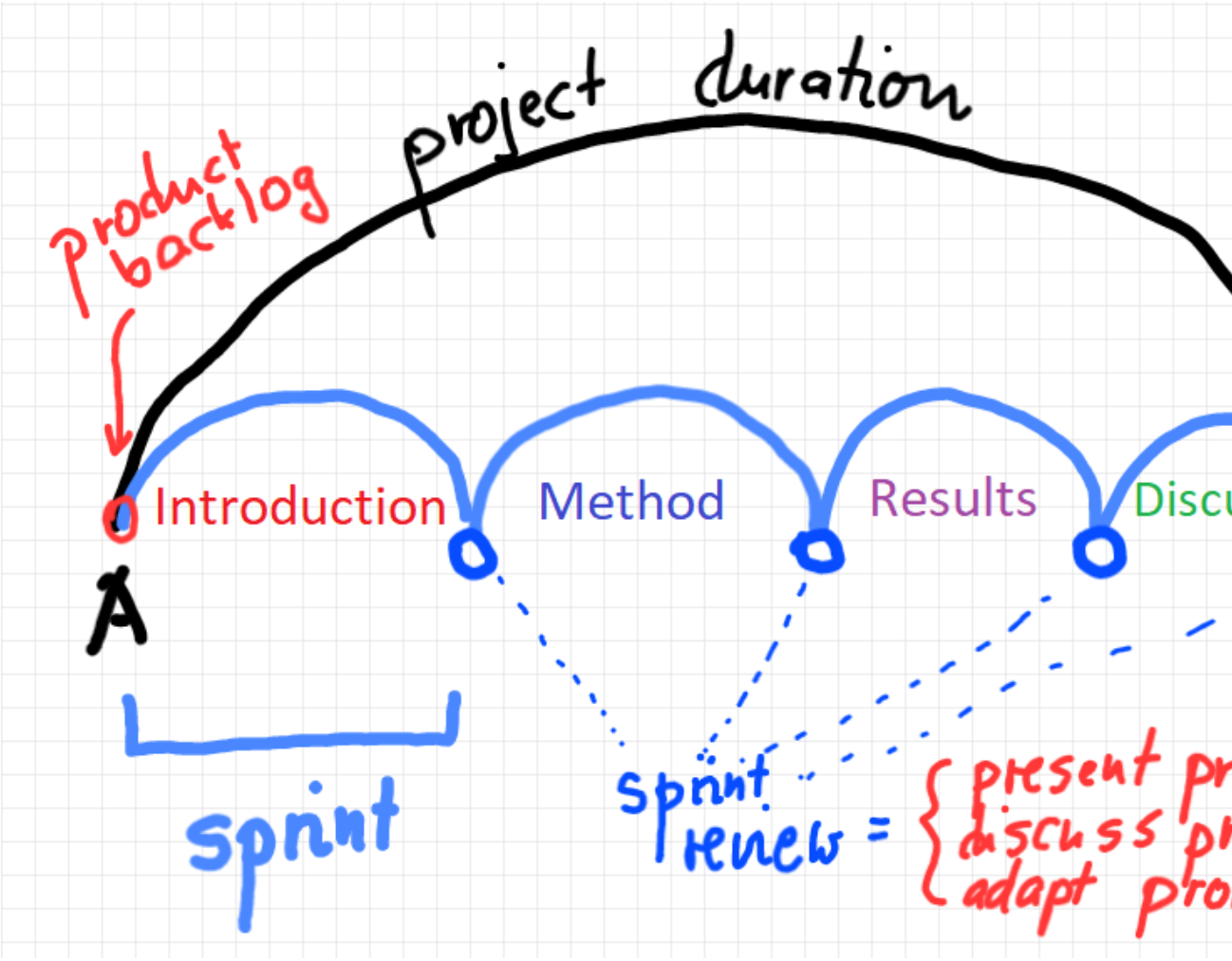

Figure 6: Scrum sprint review and IMRaD publishing framework

Sprint 2: Literature Review (Methods)

- Research appropriate methods, justify the choice, and plan implementation.

Sprint 3: Abstract (Results)

- Summarize progress, present preliminary results, and discuss challenges.

Sprint 4: Final Presentation

- Deliver a polished presentation, including project outcomes, lessons learned, and future directions.

What should we do as a project?

- For your term project, **pick a data science application or package**.
- The application exploration should include **examples and tests**
- Your final deliverable should be a **literate program** (notebook).
- The final presentation should include **exercises** for the audience
- All teams should consist of 2-3 members with clear responsibilities.
- Deliverables: proposal, literature review, abstract, presentation.

10 Student Group Project Ideas for Term Project

For your term project, pick a data science package or application to explore and explain. Below are suggested projects with packages or applications from different programming languages or language-independent tools.

1. Sentiment Analysis

- Topic: Natural Language Processing (NLP)
- Language: Python, R
- Packages: `vaderSentiment` (Python), `syuzhet` (R)
- Objective: Explore how to perform sentiment analysis using text data such as tweets or reviews. Compare the results from different tools.

Table 4: Summary of Project Ideas

Topic	Package/Tool	Language(s)
Sentiment Analysis	VADER, Syuzhet	Python, R
Data Wrangling	Pandas, DataFrames.jl	Python, Julia
Data Visualization	Plotly, Vega-Lite	Python, R, JavaScript
Data Validation/Testing	testthat, pytest	R, Python
Geospatial Analysis	GeoPandas, QGIS	Python, No-code
Web Scraping	BeautifulSoup, Web Scraper.io	Python, No-code
Statistical Analysis	SciPy, JASP	Python, No-code
Time Series Forecasting	Prophet, EViews	Python, R, No-code
Network Analysis	NetworkX, Gephi	Python, No-code
Numerical Computing	NumPy, Octave	Python, No-code
Using AI for advanced DS	ChatGPT, Grok, Claude, Gemini	R, Python, No-code

2. Data Cleaning and Transformation

- Topic: Data Wrangling
- Language: Python, Julia
- Packages: `pandas` (Python), `DataFrames.jl` (Julia)
- Objective: Demonstrate how to clean and manipulate messy datasets using modern data manipulation libraries.

3. Interactive Data Visualizations

- Topic: Data Visualization
- Language: Python, R, JavaScript
- Packages: `plotly` (Python/R), `vega-lite` (Language-Independent)
- Objective: Explore how to create interactive dashboards using Plotly or build declarative visualizations using Vega-Lite.

4. Data Validation and Testing

- Topic: Data Validation and Testing
- Language: R, Python
- Packages: `testthat` (R), `pytest` (Python)

- Objective: Explore how to validate data and test data transformations to ensure correctness. Create unit tests for data processing functions.

5. Geospatial Data Analysis

- Topic: Geospatial Analysis
- Language: Python, Language-Independent
- Packages: `geopandas` (Python), `QGIS` (Language-Independent)
- Objective: Demonstrate how to analyze and visualize geospatial data. Use datasets such as maps, population density, or earthquake data.

6. Web Scraping

- Topic: Data Collection
- Language: Python, Language-Independent
- Packages: `beautifulsoup4` (Python), `Web Scraper.io` (Language-Independent)
- Objective: Explore how to collect data from websites. Include a discussion on ethical considerations and dynamic web pages.

7. Statistical Analysis

- Topic: Statistical Analysis
- Language: Python, Language-Independent
- Packages: `scipy` (Python), `JASP` (Language-Independent)
- Objective: Perform statistical analysis and hypothesis testing using Python or the user-friendly JASP interface.

8. Time Series Analysis

- Topic: Time Series Forecasting
- Language: Python, R, Language-Independent
- Packages: `prophet` (Python/R), `EViews` (Language-Independent)
- Objective: Build time series forecasting models and visualize trends in data over time.

9. Network Analysis

- Topic: Graph Theory and Networks
- Language: Python, Language-Independent
- Packages: `networkx` (Python), `Gephi` (Language-Independent)
- Objective: Analyze and visualize networks such as social networks, transportation systems, or collaboration networks.

10. Numerical Computing

- Topic: Numerical Computing
- Language: Python, Language-Independent
- Packages: `numpy` (Python), `Octave` (Language-Independent)
- Objective: Explore numerical operations and matrix algebra. Include examples such as optimization or solving linear equations.

11. Using AI for Advanced Data Science Tasks

- Topic: AI-Powered Data Science
- Language: R, Python, No-code
- Tools: ChatGPT (OpenAI), Grok (xAI), GitHub Copilot, Claude (Anthropic), Gemini (Google)
- Objective: Explore how AI tools can assist with data science tasks such as cleaning, transformation, summarization, and code generation. Discuss ethical implications, limitations, and best practices.
- Suggested Activities:
  - Demonstrate how AI tools can automate repetitive tasks (e.g., cleaning messy datasets).
  - Compare code suggestions from different AI tools and evaluate their accuracy and usefulness.
  - Create a short "prompt engineering" guide for data science tasks.
  - Include a live demo or code-along session for using one of these AI tools effectively.

Which tools are you going to use?

- DataCamp courses (weekly home assignments)
- GitHub repository (all course materials except tests)
- GNU Emacs + ESS + R (literate programming environment)
- DataCamp's DataLab and Google Colaboratory notebooks (sometimes).
- Canvas (learning management system)

Linux, of course

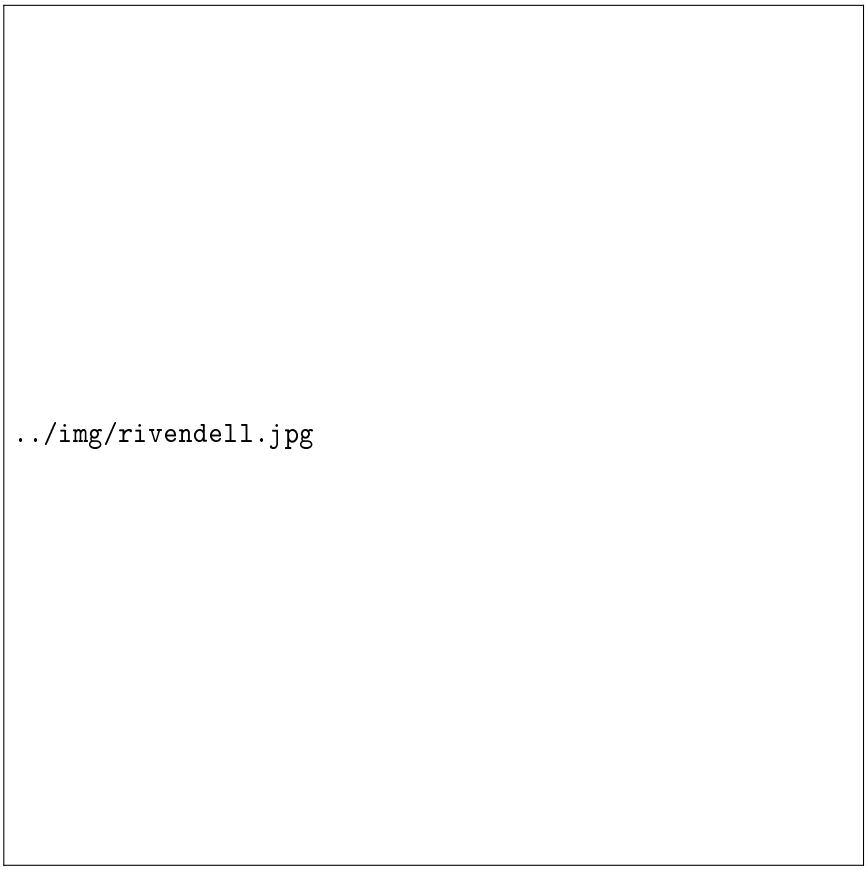

Figure 7: Rivendell

- The Linux VMs from last term should all work with the same access data as before. Remember to shut down Emacs when you're done.
- First thing: Run `update` and `upgrade`:

  ```
  sudo apt update -y && sudo apt upgrade -y
  ```

- Install WSL (Windows Subsystem for Linux) on your PC, then learn the command line with Shotts' book (5e, 2023).
- Recall: When you cannot install a package because of your version of R, you can look for a compatible version:
  1. check your R `version` and then pick an earlier package version using the CRAN archive.
  2. Example: Installing `MASS`. If you have R version 4.0.4 (2021-02-15), then version 7.3.54 from 2021-05-03 is a safe bet:

     ```
     install.packages("remotes")
     require(remotes)
     install_version("MASS", version="7.3.54")
     library(MASS)
     search()  # MASS appears in environment list
     ```

How can you register at DataCamp?

- You find the invitation link to the group for Spring 25 in Canvas.
- With this, you will automatically be subscribed to the DSC 205 team.
- If you are in more than one course, I will add you later manually.
- These accounts will be valid until June 23, 2025.
- There will be additional in-class and home assignments.

AI 2023 to 2025: Don't be fooled

How to handle AI for coding and data science

- 2024, I wrote "A note on using AI to write code for you" (tinyurl.com/Using-AI-to-code)

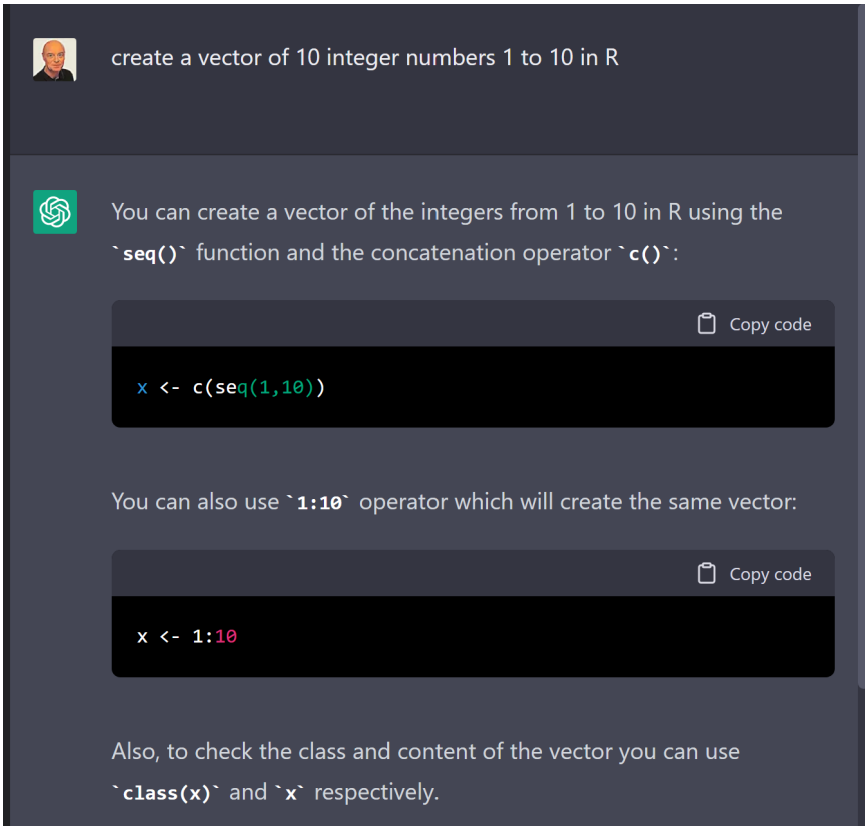

Figure 8: ChatGPT in Spring 2023

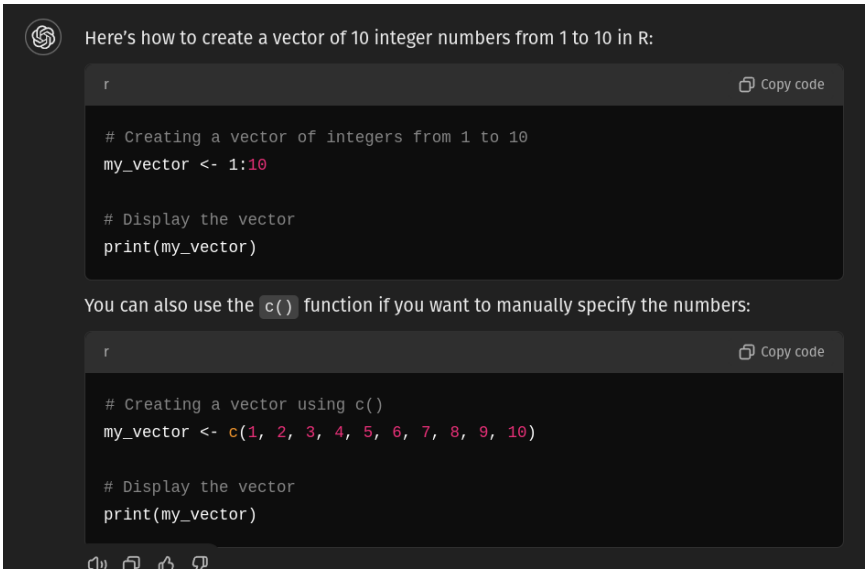

Figure 9: ChatGPT in Spring 2025

Figure 10: AI and the Garden of Eden

- My recommendation then was: Experiment with it but don't get dependent on it.
- But that's like saying: Start smoking but don't get addicted to it. It's not possible. Using AI will change your coding habits profoundly - (practically & psychologically, not chemically).
- **The bad news:** It'll make you dumber and perhaps not even faster, and you'll depend on it until you decide to wean yourself off it.
- **The good news:** Everybody else is doing it anyway, and if you both know how it works and maintain your independence, you may thrive.
- **What am I doing about it?** It doesn't matter because I'm too different from you to compare in too many ways.
- **I use AI for:** Multiple-choice test creation; 2nd coding opinion; last-resort debugging; project identification; literature search.

What am I looking forward to?

- Reconnecting with a language you know (R, C++, SQL) is fun.

- Reconnecting with a language (R) at a deeper level is fun.
- Looking forward to learn from **your projects**!
- Learning more about the **object-oriented** aspects of R.
- Dealing another blow to the **"Tidyverse"** (cp. "TidyverseSceptic").
- Picking up a little more **Python** along the way perhaps.

When is the first assignment due?

- The first DataCamp home assignment is due on Friday, January 21. For late submissions, you lose 1 point per day (out of 10).
- The first in-class assignment is due on Friday, January 21. For late submissions, you lose 1 point per day (out of 10 possible points)
- We'll write the first weekly multiple-choice test in class on January 23, covering the material that was covered until then.

Next topics

Figure 11: The One Ring To Rule Them All

- A Review of R: 10 Basic Problems
- Calling functions: Scoping (code along + practice)