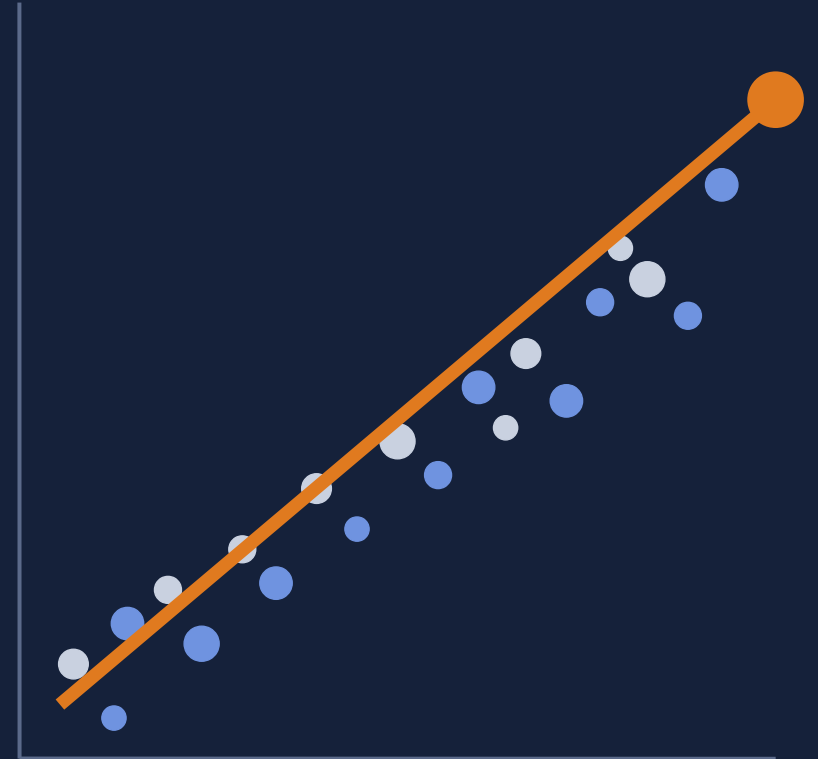


## LECTURE 1 · INTRODUCTION TO DATA SCIENCE

# Turning data into decisions

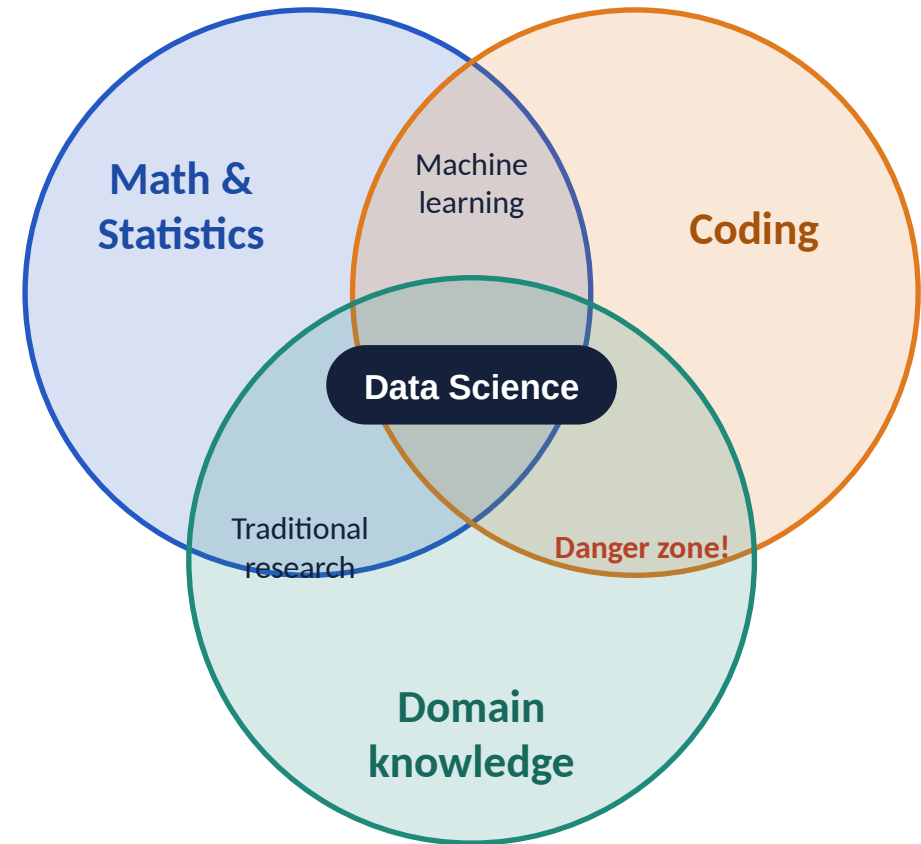
Welcome! No coding or statistics background needed, just curiosity.



# What is data science?

Using data to answer questions and make better decisions.

- Math & statistics: tell a real pattern from luck
- Coding: handle data too big for a spreadsheet
- Domain knowledge: ask questions that matter



Data science lives where all three overlap.

*Adapted from Drew Conway's data science Venn diagram (2010)*

# What is data?

Most data science starts with a table. Here are four text messages, described as data.

| message                       | length | contains<br>"free" | has a number | label |
|-------------------------------|--------|--------------------|--------------|-------|
| Claim your FREE gift now      | 24     | yes                | no           | spam  |
| Class moved to room 305       | 23     | no                 | yes          | ham   |
| Free entry: text WIN to 80086 | 29     | yes                | yes          | spam  |
| Ok, see you at the library    | 26     | no                 | no           | ham   |

**One row = one example**

a single message, patient or sale

**One column = one feature**

a measured property, used as input (X)

**Label column = the answer**

what we want to predict (y)

A model learns how the feature columns relate to the label, then predicts the label for new rows. In Lecture 5 this table has 5,572 rows; in Lecture 6 we call the features **X** and the label **y**.

# Why learn data science?

1

## Data is everywhere

Every tap, purchase and scan leaves a data trail.

2

## Better decisions

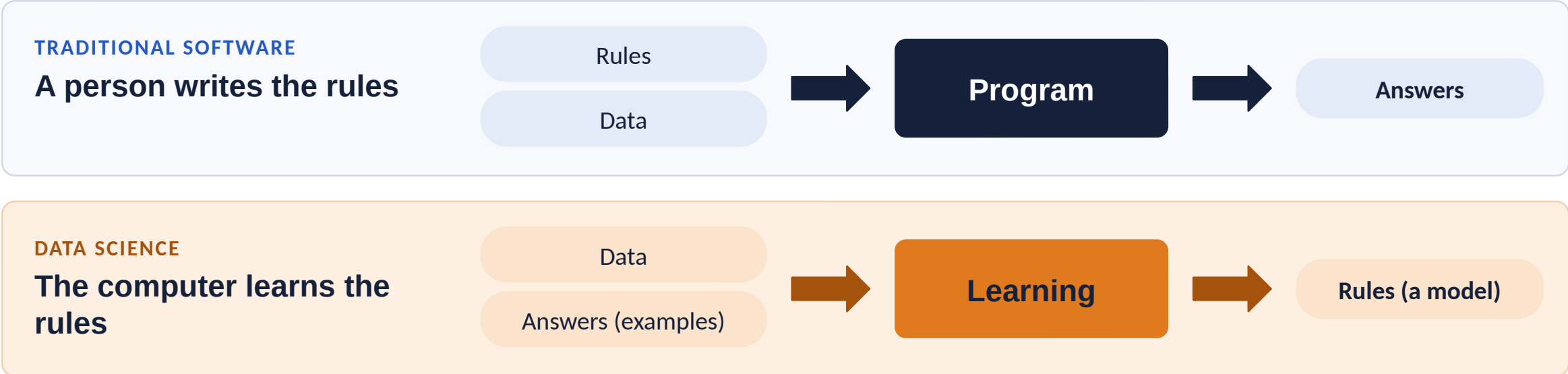
Swap guesses for evidence: what to stock, whom to treat first.

3

## Skills that travel

Health, finance, farming, research: all need people who read data.

# Traditional software vs. data science



|           | Traditional software        | Data science                               |
|-----------|-----------------------------|--------------------------------------------|
| Example   | Calculator, payroll system  | Spam filter, face unlock                   |
| Output    | Exact, the same every time  | A prediction, with some uncertainty        |
| Best when | The rules are known exactly | The rules are too complex to write by hand |

# Live demo: can you write rules that catch spam?

1

## Write rules

Keywords like "free" catch 4 of 8 messages. Every fix opens a new hole.

2

## Let it learn

A model trained on 30 labeled messages learns its own clues.

3

## Try your own

Type any message and see which words look spammy to the model.

# The data science lifecycle



A loop, not a line: exploring often sends you back to ask a better question. Modeling is one step of six.

# You already use data science every day

1

## Recommendations

What will you watch next? Learns from your history and what similar people enjoyed.

2

## Healthcare

Does this X-ray look healthy? Learns from past scans labeled by doctors.

3

## Finance

Is this payment fraud? Learns from normal spending and past fraud cases.

4

## Maps and travel

How long will this trip take? Learns from live speeds of phones on the road.

# The road ahead: from zero to a spam detector

| Lecture     | Topic                                     | You will be able to                               |
|-------------|-------------------------------------------|---------------------------------------------------|
| 1           | What is data science?                     | Explain what data science is and how models learn |
| 2.1 to 2.3  | Getting started, Python basics, debugging | Run code offline and online; read and fix errors  |
| 3.1 to 3.3  | Decisions, lists and tuples, loops        | Build a rule-based spam filter, and see it fail   |
| 4.1 to 4.2  | Functions, dictionaries                   | Count words and score messages from examples      |
| 5           | Real data with pandas                     | Load, clean and explore 5,572 real messages       |
| 6           | Build a spam detector                     | Train, test and honestly evaluate a real model    |
| [7 onwards] | [Remaining course topics]                 | [To be added]                                     |

# How this course works

| Item             | Details                                                                                 |
|------------------|-----------------------------------------------------------------------------------------|
| Classes and labs | [Days, times, rooms]. Labs use Jupyter Notebook on the lab PCs (offline)                |
| At home          | Google Colab (online), or Jupyter if you install Anaconda on a laptop                   |
| Assessment       | [Homework __%] · [Quizzes __%] · [Midterm __%] · [Project __%] · [Final __%]            |
| Homework         | After every lecture from Lecture 2. Submit your .ipynb file on [LMS / Google Classroom] |
| Late work        | [Late policy]                                                                           |
| Help             | Office hours: [day, time, room] · Contact: [email]                                      |

# Using AI tools in this course

1

## Allowed

[For example: asking an AI to explain an error message or a concept, like a tutor.]

2

## Not allowed

[For example: submitting code or answers generated by an AI as your own work.]

3

## Always

You must be able to explain every line you submit. [Say whether and how AI use must be disclosed.]

## BEFORE NEXT CLASS

# Your first mission

- 1 Lab: check that Jupyter Notebook opens on your lab PC.  
Home: open Google Colab if you have internet.
- 2 Spot one everyday decision that data could improve. Write it in one sentence.
- 3 Bring one question about data you are curious about.

Office hours: [day, time, room] · Contact: [email]



## Questions?

No question is too basic, today especially.

# Recap: Lecture 1 in 90 seconds

RECAP

## Lecture 1 in 90 seconds

A short recap of today's lecture. Pause any time.

*Click to play or pause. Watch it again at home to revise.*