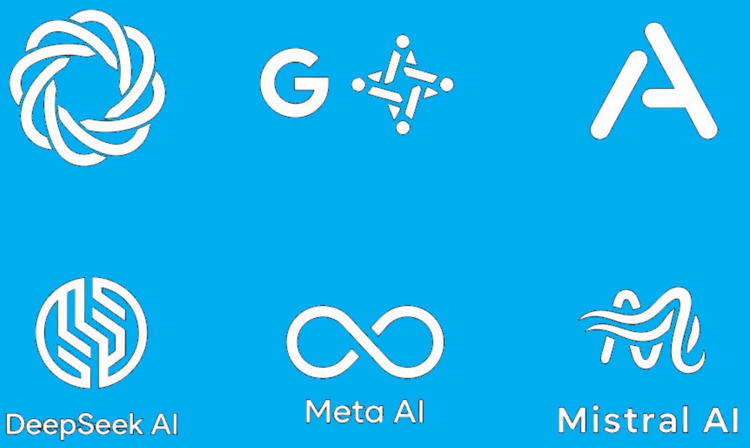
Warwick
Business
School

Data Science & Generative AI

Dr Michael Mortenson

Associate Professor (Reader)
michael.mortenson@wbs.ac.uk

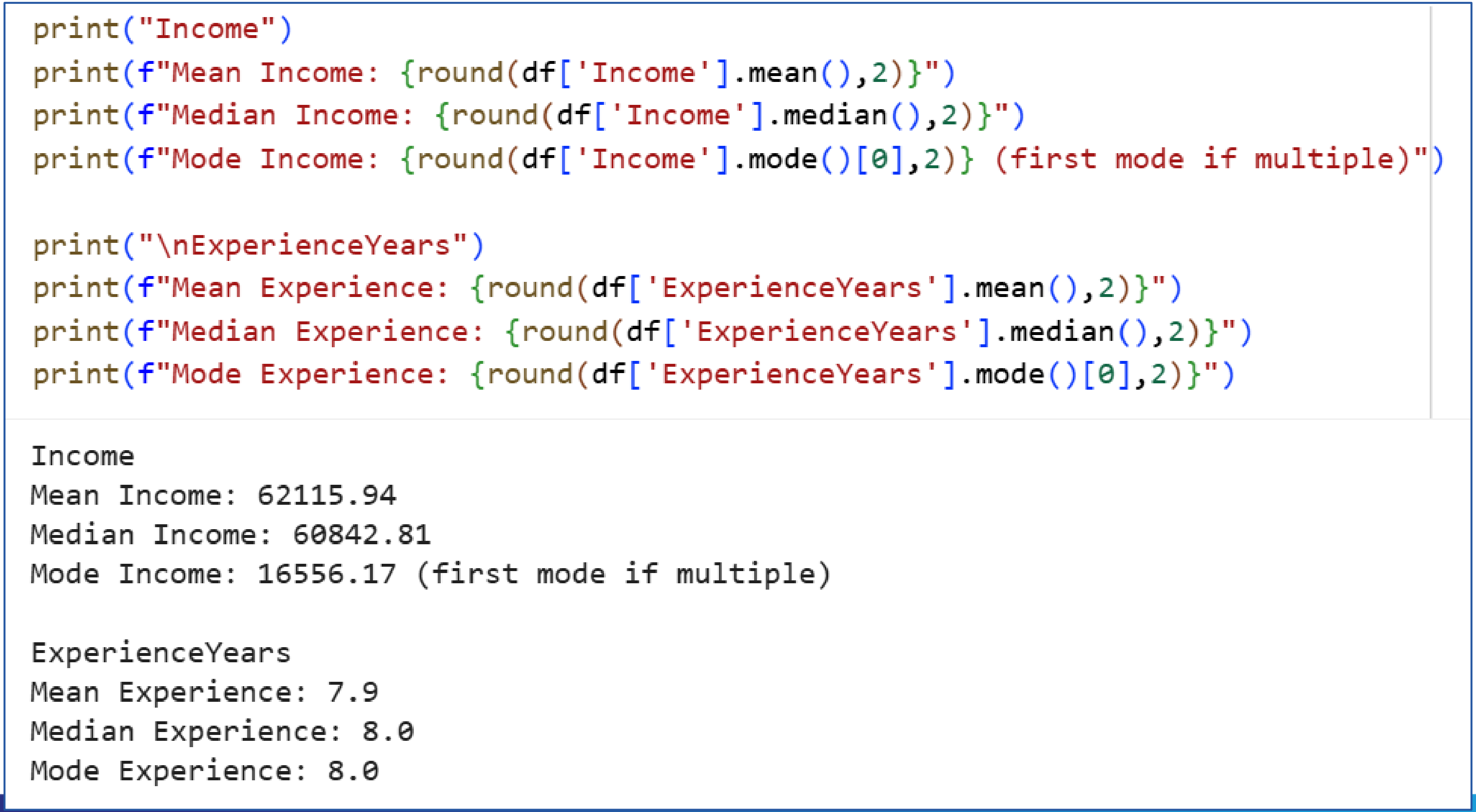


Session 3: Exploratory Data Analysis with Generative AI

1.1 From Crude to Refined Oil

Image source (creative commons): https://morningstaronline.co.uk/sites/default/files/styles/article_full/public/zbynek-burival-GrmwVnVSSdU-unsplash.jpg?itok=EObuKnRY&c=ef17667d17ca75f0f9b22106b473877d

1.2 Compression by Descriptive Statistics



```python
print("Income")
print(f"Mean Income: {round(df['Income'].mean(),2)}")
print(f"Median Income: {round(df['Income'].median(),2)}")
print(f"Mode Income: {round(df['Income'].mode()[0],2)} (first mode if multiple)")

print("\nExperienceYears")
print(f"Mean Experience: {round(df['ExperienceYears'].mean(),2)}")
print(f"Median Experience: {round(df['ExperienceYears'].median(),2)}")
print(f"Mode Experience: {round(df['ExperienceYears'].mode()[0],2)}")
```

```
Income
Mean Income: 62115.94
Median Income: 60842.81
Mode Income: 16556.17 (first mode if multiple)

ExperienceYears
Mean Experience: 7.9
Median Experience: 8.0
Mode Experience: 8.0
```

1.3 Compression by Machine Learning

x_1

x_2

x_3

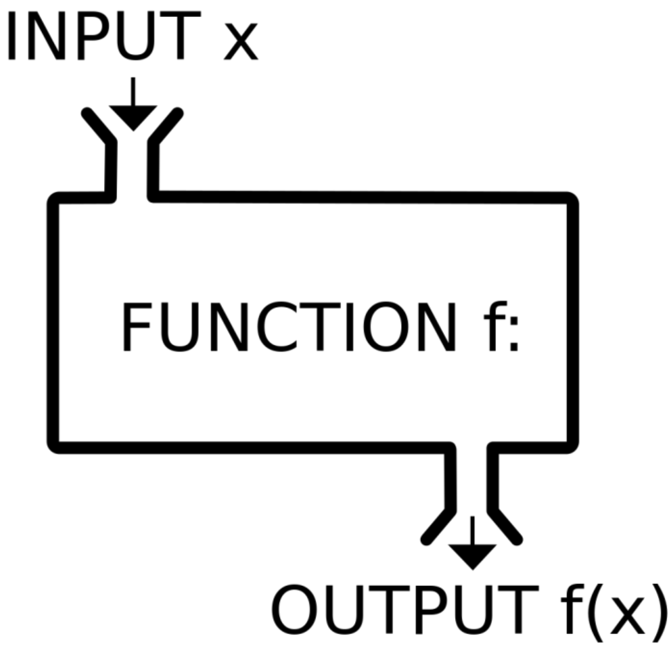
x_4

x_5

x_6

$\dots$

x_d



INPUT x

FUNCTION f:

OUTPUT f(x)

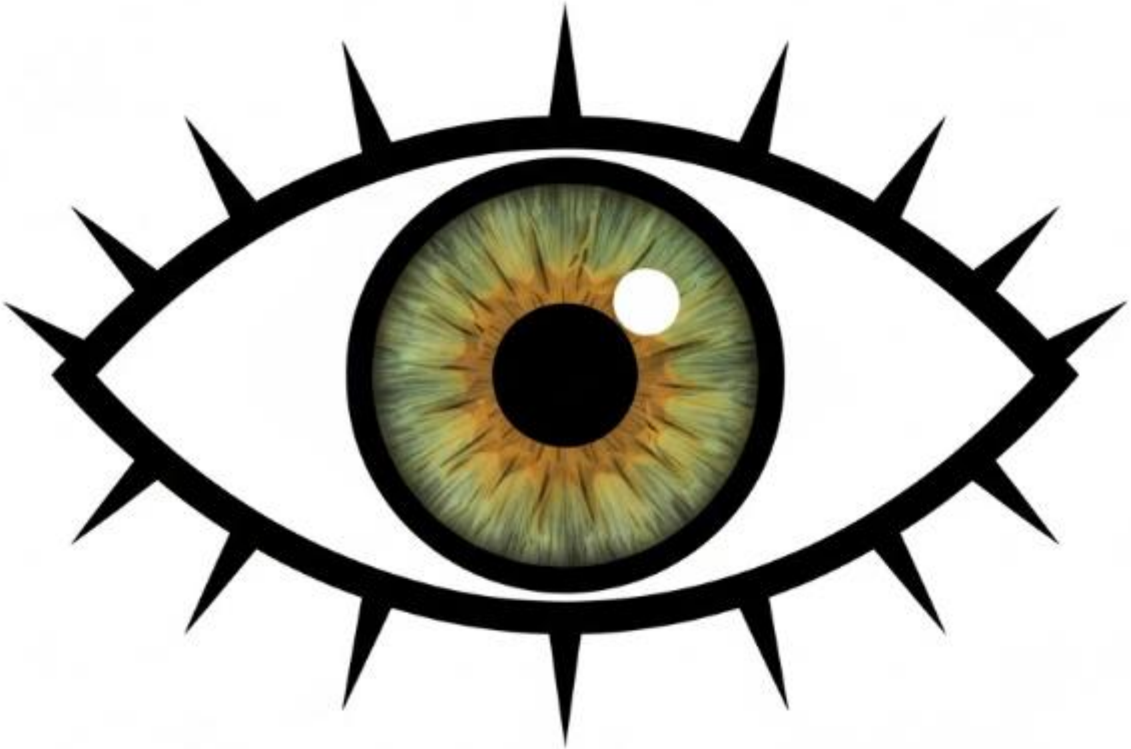
$\hat{Y}$

1.4 Compression by ChatGPT

570Gb

???

1.5 Compression by Eye?



Session Aims

Introduction

Visual Analysis 101

Generative AI for EDA (and coding)

EDA of the Titanic Dataset

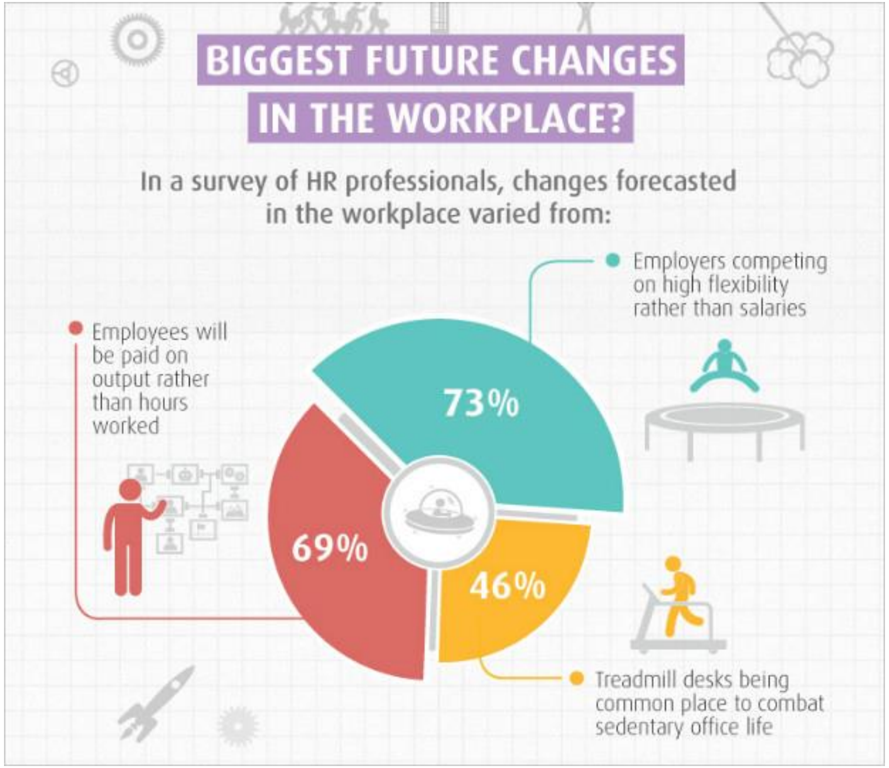
2.1 Shall We Play a Game?

2.2 Compression by Eye?

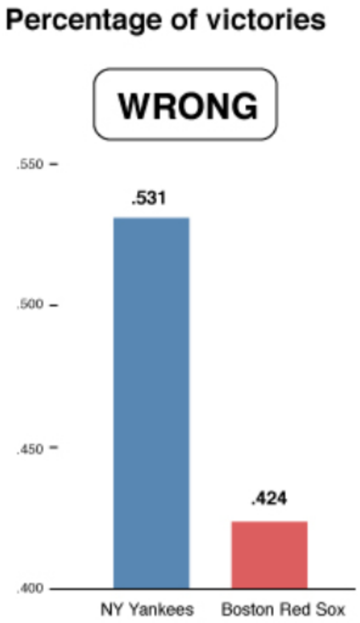
> "Visual analytics is the representation and presentation of data that exploits our visual perception abilities in order to amplify cognition."
>
> Andy Kirk

Andy Kirk. *Data Visualization: A Successful Design Process*
https://www.visualisingdata.com/

2.3 Bad Charts

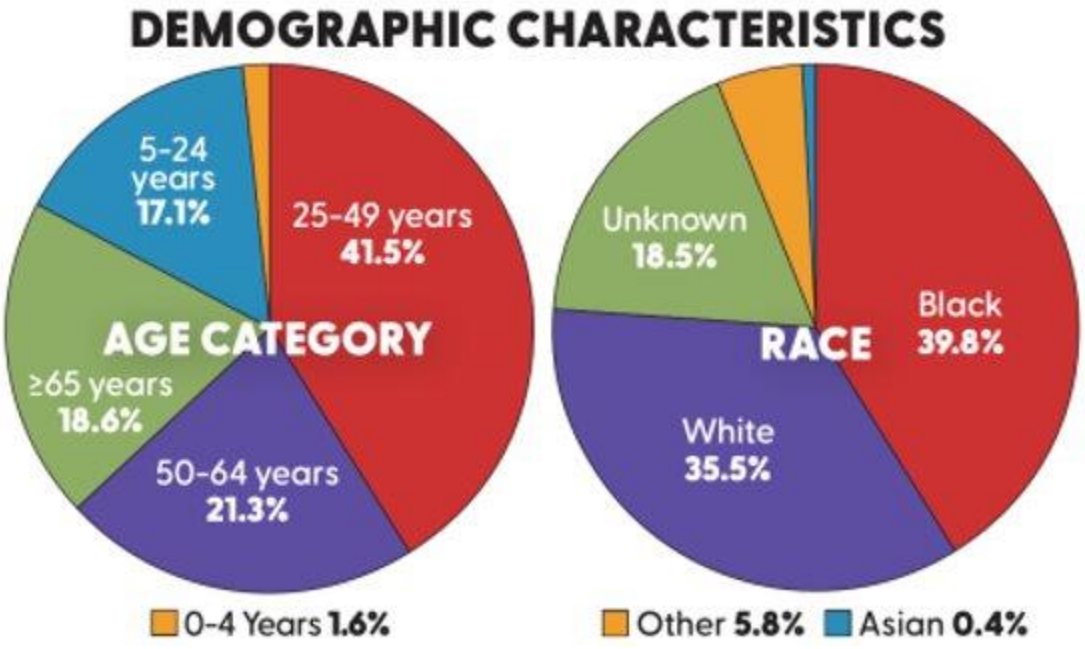


2.3 Bad Charts

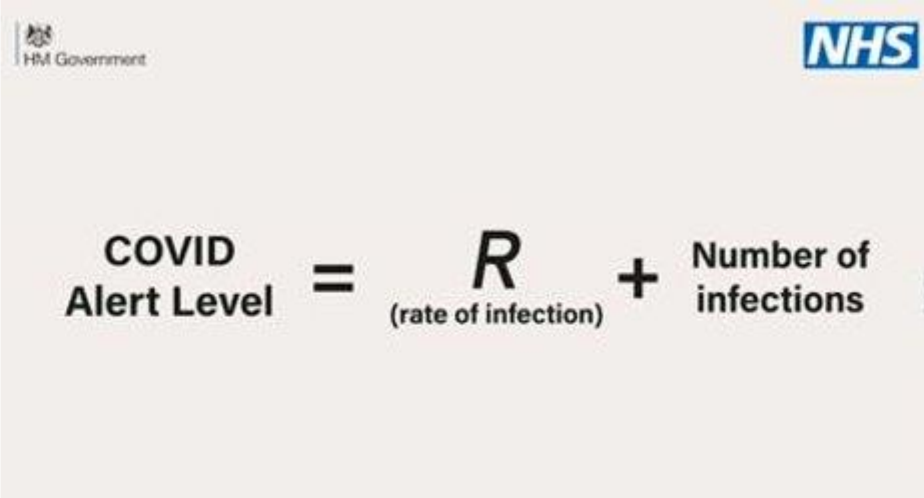


SOURCE: MAJOR LEAGUE BASEBALL

2.4 Bad Charts: COVID 19 special



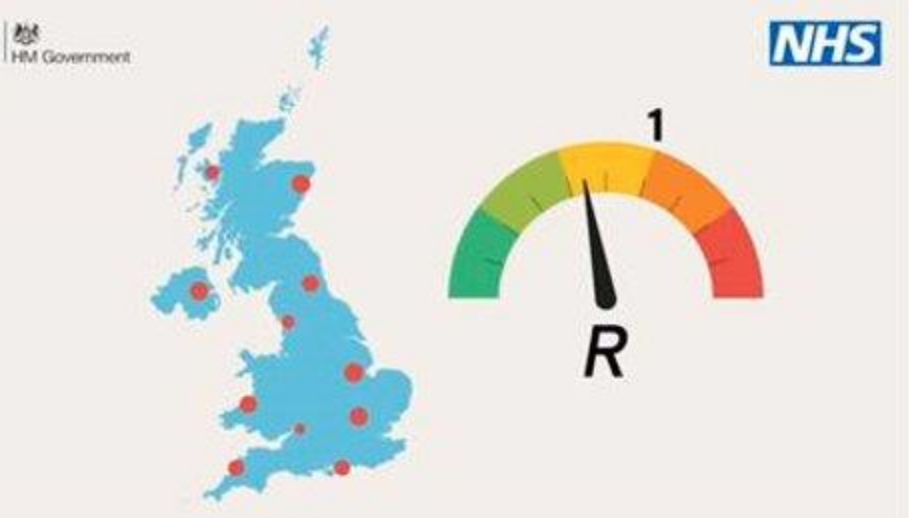
2.4 Bad Charts: COVID 19 special



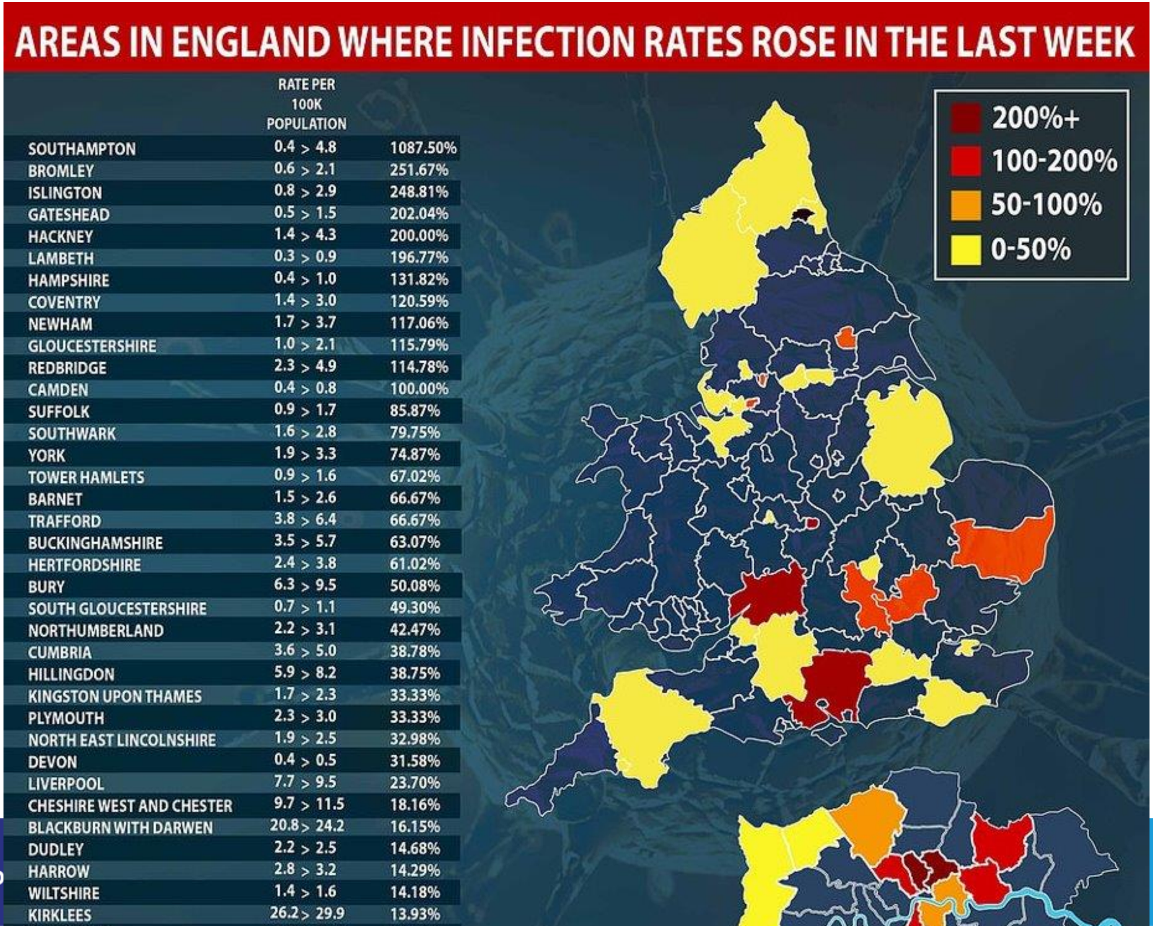
HM Government

NHS

COVID Alert Level = *R* (rate of infection) + Number of infections



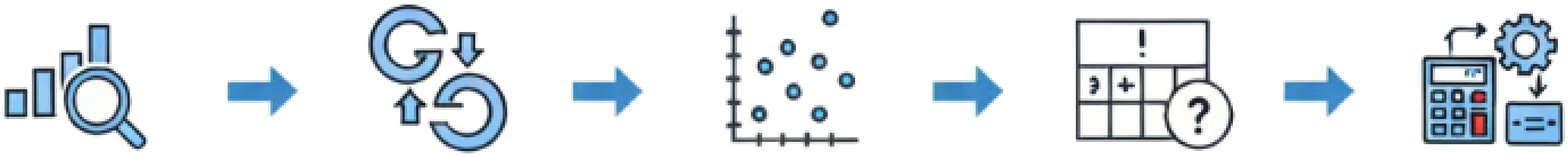
2.4 Bad Charts: COVID 19 special



AREAS IN ENGLAND WHERE INFECTION RATES ROSE IN THE LAST WEEK

	RATE PER 100K POPULATION	
SOUTHAMPTON	0.4 > 4.8	1087.50%
BROMLEY	0.6 > 2.1	251.67%
ISLINGTON	0.8 > 2.9	248.81%
GATESHEAD	0.5 > 1.5	202.04%
HACKNEY	1.4 > 4.3	200.00%
LAMBETH	0.3 > 0.9	196.77%
HAMPSHIRE	0.4 > 1.0	131.82%
COVENTRY	1.4 > 3.0	120.59%
NEWHAM	1.7 > 3.7	117.06%
GLOUCESTERSHIRE	1.0 > 2.1	115.79%
REDBRIDGE	2.3 > 4.9	114.78%
CAMDEN	0.4 > 0.8	100.00%
SUFFOLK	0.9 > 1.7	85.87%
SOUTHWARK	1.6 > 2.8	79.75%
YORK	1.9 > 3.3	74.87%
TOWER HAMLETS	0.9 > 1.6	67.02%
BARNET	1.5 > 2.6	66.67%
TRAFFORD	3.8 > 6.4	66.67%
BUCKINGHAMSHIRE	3.5 > 5.7	63.07%
HERTFORDSHIRE	2.4 > 3.8	61.02%
BURY	6.3 > 9.5	50.08%
SOUTH GLOUCESTERSHIRE	0.7 > 1.1	49.30%
NORTHUMBERLAND	2.2 > 3.1	42.47%
CUMBRIA	3.6 > 5.0	38.78%
HILLINGDON	5.9 > 8.2	38.75%
KINGSTON UPON THAMES	1.7 > 2.3	33.33%
PLYMOUTH	2.3 > 3.0	33.33%
NORTH EAST LINCOLNSHIRE	1.9 > 2.5	32.98%
DEVON	0.4 > 0.5	31.58%
LIVERPOOL	7.7 > 9.5	23.70%
CHESHIRE WEST AND CHESTER	9.7 > 11.5	18.16%
BLACKBURN WITH DARWEN	20.8 > 24.2	16.15%
DUDLEY	2.2 > 2.5	14.68%
HARROW	2.8 > 3.2	14.29%
WILTSHIRE	1.4 > 1.6	14.18%
KIRKLEES	26.2 > 29.9	13.93%

2.5 EDA Process



Analyse / visualise each variable (univariate analysis)

Compare variables for patterns (bivariate analysis)

Check for outliers and extreme values

Check for missing values and replace

Feature engineering (discussed previously)

Session Aims

Introduction

Visual Analysis 101

Generative AI for EDA (and coding)

EDA of the Titanic Dataset

3.1 GenAI Tools (IMO)

- **ChatGPT:** the original. Useful for most tasks and particularly good at image generation. Not aware of any current discounts for students on premium models.
- **Claude/Anthropic:** the writer's choice. Useful particularly for creative writing but good at most tasks. Not aware of any current discounts for students on premium models.
- **Gemini/Google:** the all-rounder. Maybe not the best at any single task, but better than most at most. (My current preferred tool). Free PRO version for students here: https://gemini.google/students/.

3.1 GenAI Tools (IMO)

- **DeepSeek:** the upstart. Useful for most tasks and, for obvious reasons, particularly good at both English and Mandarin. Also offers good open source versions you can run for free if you want to customise (and have those skills). No pro version.
- **Llama/Meta:** the other open source one. Also offers good open source versions you can run for free if you want to customise (and have those skills). Not currently as good as DeepSeek.
- **Mistral and Cursor:** the coding specialists. Mistral AI and Cursor are mostly specialised to coding tasks. While this may be attractive to us here, probably not necessary for the work we are doing compared to Gemini’s integration with Colab.

3.2 Generative AI for Coding

- ✓ Writing code is hard (as we have seen in some of the visualisation code in Python). Because we don’t speak in code (it is not our native tongue), we are always translating into code and having to remember that translation.
- ✓ Although code is hard, the vocabulary is relative small. I.e. we use certain specific words to perform specific actions, without lots of synonyms and artistic language! There is less to learn.
- ✓ Generative AI was trained on a large corpus of internet data. There’s lots of code on the internet (particularly Python).
- ✓ It’s quite good at it. Not great, but quite good.

3.3 AI Generation

- **IMPORTANT!** Every time the AI generates a word it is paying attention to the words you typed in (the *prompt*) and also **all** of the words it has already generated.
- If your prompt is not perfect then the output may not be perfect!
- If parts of the previous output (e.g. code) are not perfect, it may well pay attention to these too and continually produce imperfect output. This means AI tools can get stuck in a loop where they keep making the same mistakes.

3.4 AI Generation – Best Practices

- Before you ask the tool to generate anything, think carefully about what you expect the output to look like. If you don't know what to expect you cannot check the quality of the output.
- Ask the AI only to generate relatively small blocks of code to serve a specific purposes.
  - The longer the output, the harder it is to find the important context and to avoid repeating mistakes.
  - The longer the output, the harder it is for you to check the output.
- Be as specific as possible. Give the AI the name of the variables you want to use and the specific output you want. E.g. "*Use the 'rating' and 'location' data from 'mydf' to create a scatter plot*".

3.4 AI Generation – Best Practices

- Use an AI that has direct access to your code to save you copying and pasting. Google Colab has Gemini built in to do this (which is what we will use). Cursor is a popular tool in industry.
- If the code generates errors, try to get a simple explanation for the error. Python is a lot better today, but still can be unclear.
- If the AI you are using makes an error, and then can't fix it, copy out to a different tool (e.g. ChatGPT) to see if it can see it. You can also try changing the prompt.
- Don't let it get out of control! It should be helping you not building its own thing. Keep it small and measurable and read the output in full and do your best to understand it.

Session Aims

Introduction

Statistical Exploratory Data Analysis

Generative AI for EDA (and coding)

EDA of the Titanic Dataset

4.1 Exercise #2

- Download the “titanic.csv” dataset from my.wbs. If it opens in a browser window, right click and select “save as”.
- You are going to be performing EDA on this data and exploring for any issues and patterns within it.
- You can write the code yourself, copy from the Github file(s) shared, or (preferably) use AI to help you do this (following the tips shared on the previous slides!).
- Take your time to explore the results, not just generate something and move on without thought. The point is to have a conversation with the data and understand its secrets ☺

4.2 Data Description

- **PassengerId:** unique ID for each passenger.
- **Survived:** did the passenger survive (value = 1) or die (value – 0)?
- **Pclass:** what class of ticket did the passenger have? 1 = “Upper”, 2 = “Middle”, 3 = “Lower”.
- **Name:** passenger name.
- **Sex:** male or female.
- **Age:** age in years.

4.2 Data Description

- **SibSp:** the number of siblings (brothers, sisters or step-siblings) also on the boat.
- **Parch:** the number of parents and children also on the boat.
- **Ticket:** ticket number.
- **Fare:** amount paid.
- **Cabin:** cabin number.
- **Embarked:** where did the passenger get on the boat? C = Cherbourg, Q = Queenstown, S = Southampton.