DATA1002 Week 8 Tutorial

Monday 22/09/25

Tutorial Outline

- Content revision (Visualisation), Research Task
- Content revision (Matplotlib), Python Task
- Work on Assignment 1

THE UNIVERSITY OF SYDNEY

Tutor: *Tommy Lu*

Housekeeping

1st hour we'll be revising content.
2nd hour we'll be working on the Assignment.

Group Project Stage 1

Due: 17:00 pm on Sunday at the end of week 8 (Sep 28th)

Data analysis	**Project stage 2** in-class group presentation on predictive model and evaluation of its success.	8%	Week 12

Housekeeping

Assignment 1

- Aggregate summaries
- Charts and visual representations
- Machine learning predictions
- Presentations

Data analysis	**Project stage 2** in-class group presentation on predictive model and evaluation of its success.	8%	Week 12

Housekeeping

1st hour we'll be revising content.
2nd hour we'll be working on the Assignment.

This week's content

- Aggregate summaries
- Charts and visual representations
- Machine learning predictions
- Presentations

Group Project Stage 1

Due: 17:00 pm on Sunday at the end of week 8 (Sep 28th)

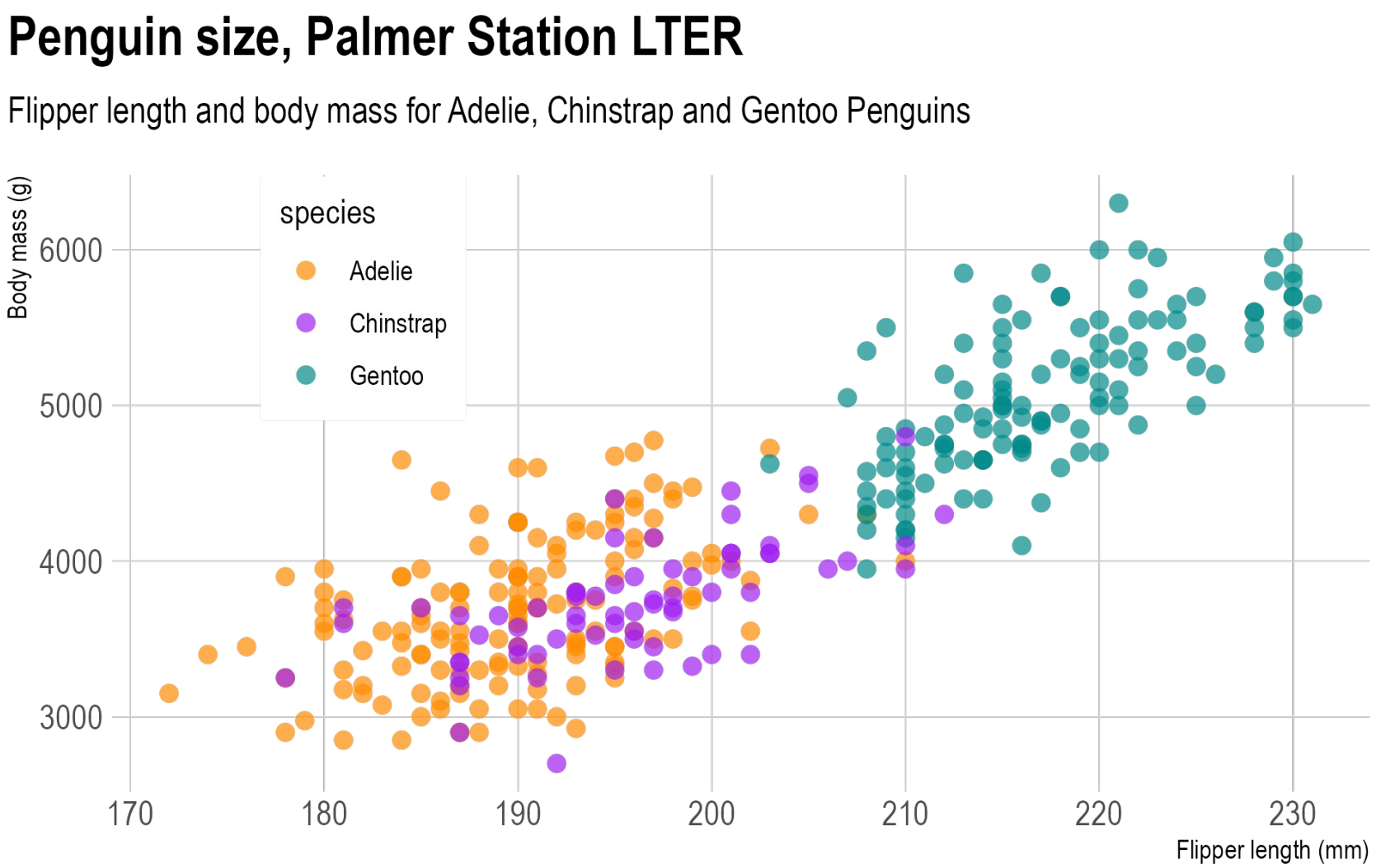
Data analysis	**Project stage 2** in-class group presentation on predictive model and evaluation of its success.	8%	Week 12

Content

Information Visualisation

What is a Chart?

What is a Chart?



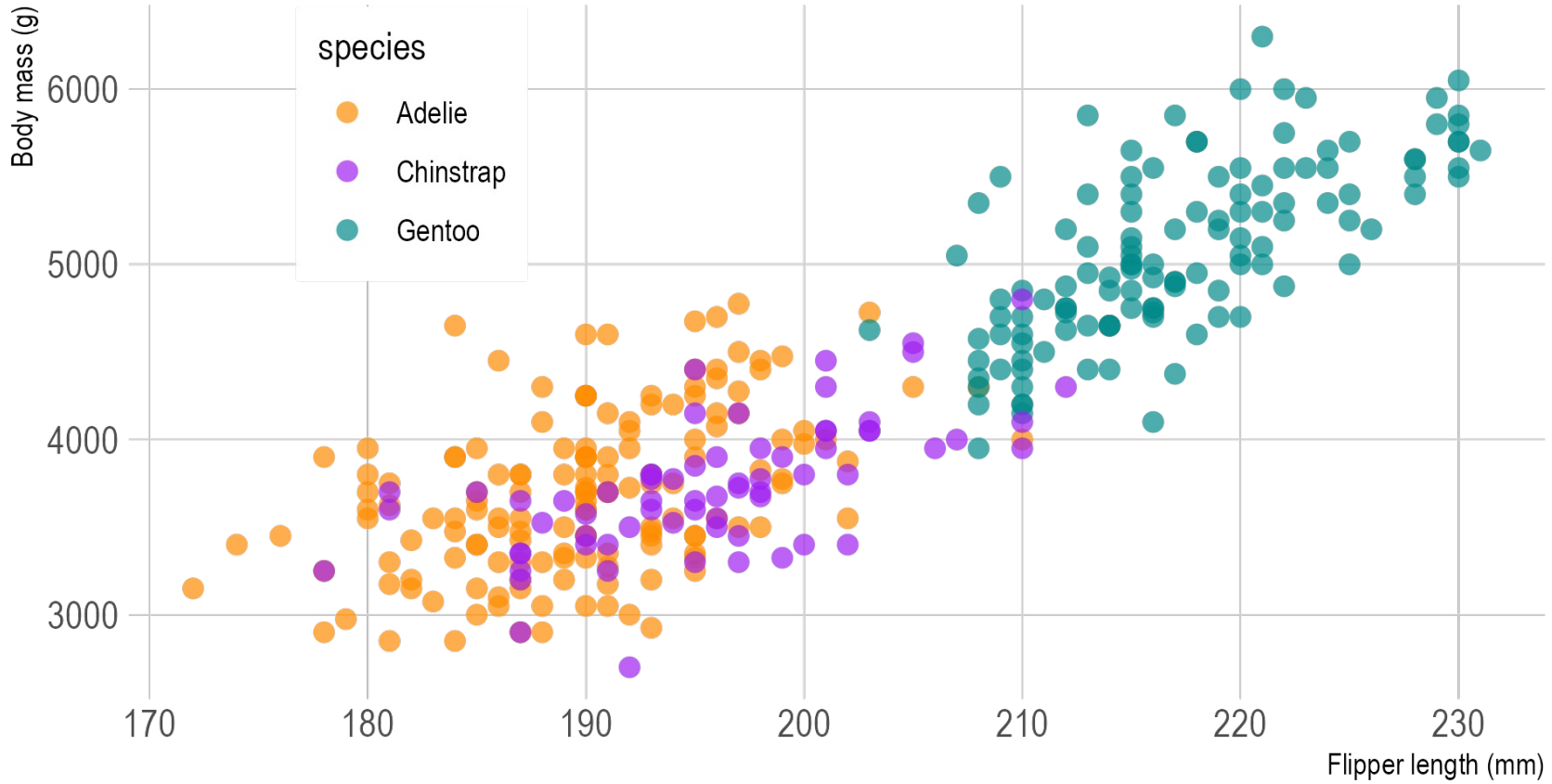
What is a Chart?

Title

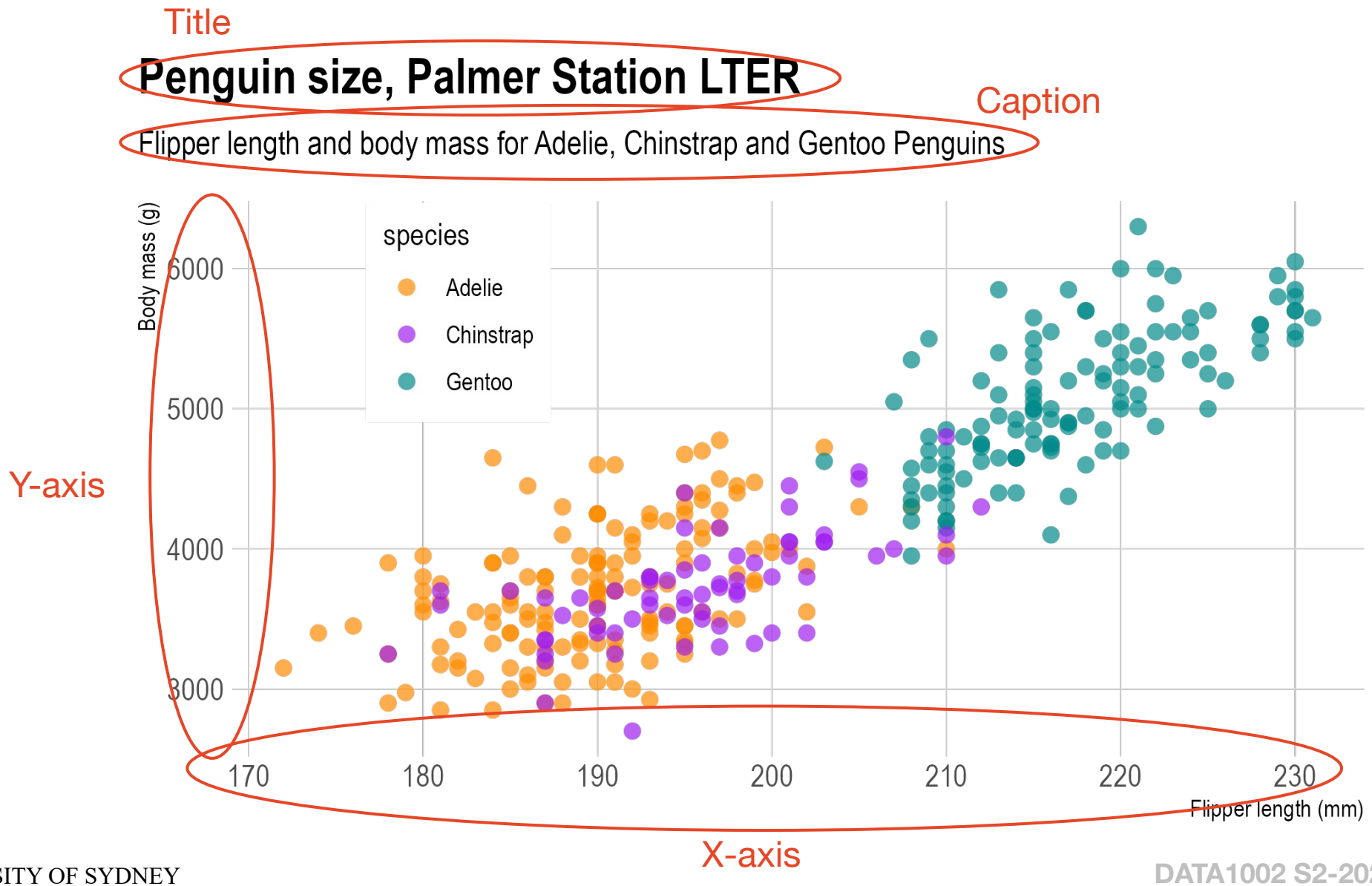
Penguin size, Palmer Station LTER

Caption

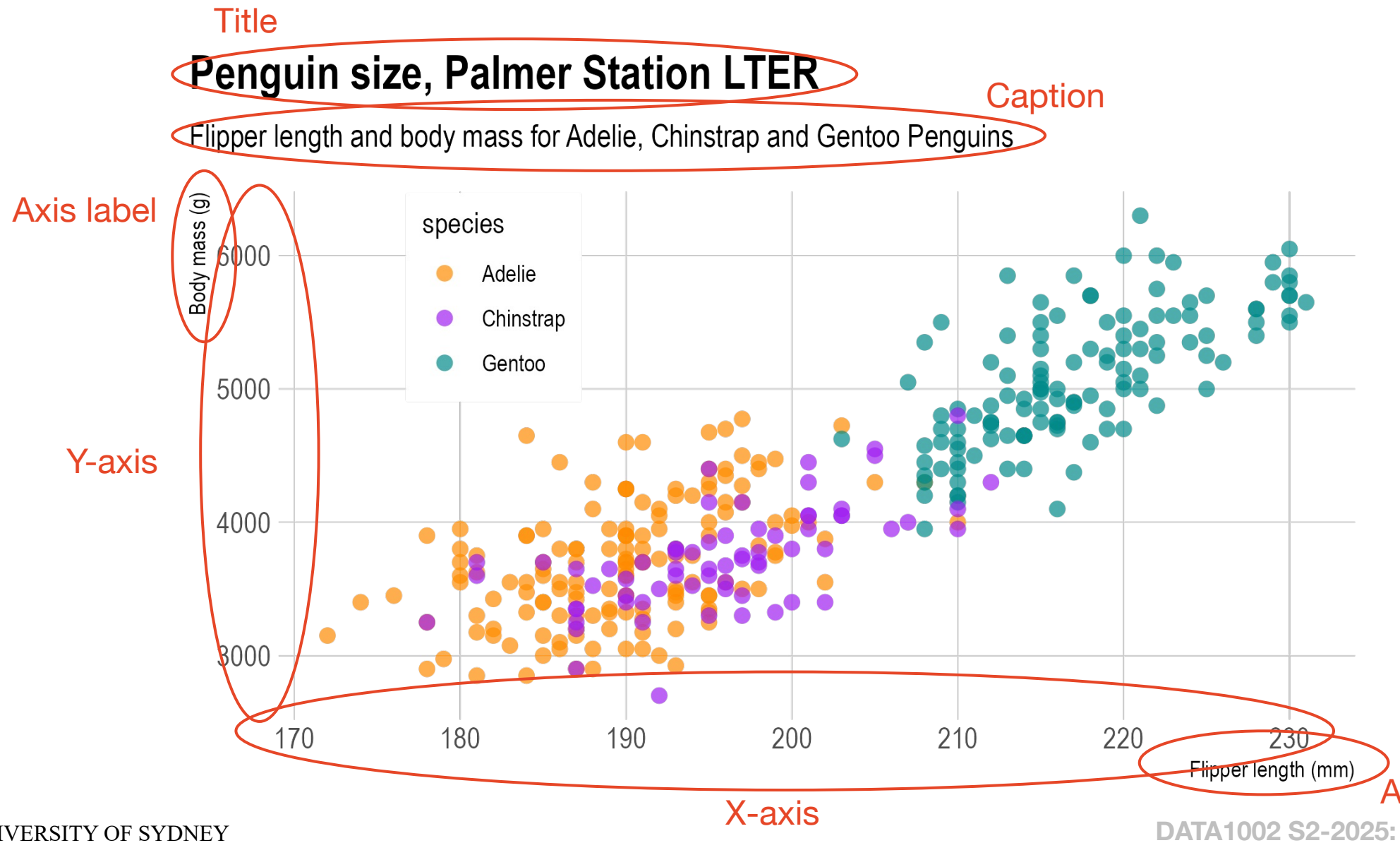
Flipper length and body mass for Adelie, Chinstrap and Gentoo Penguins



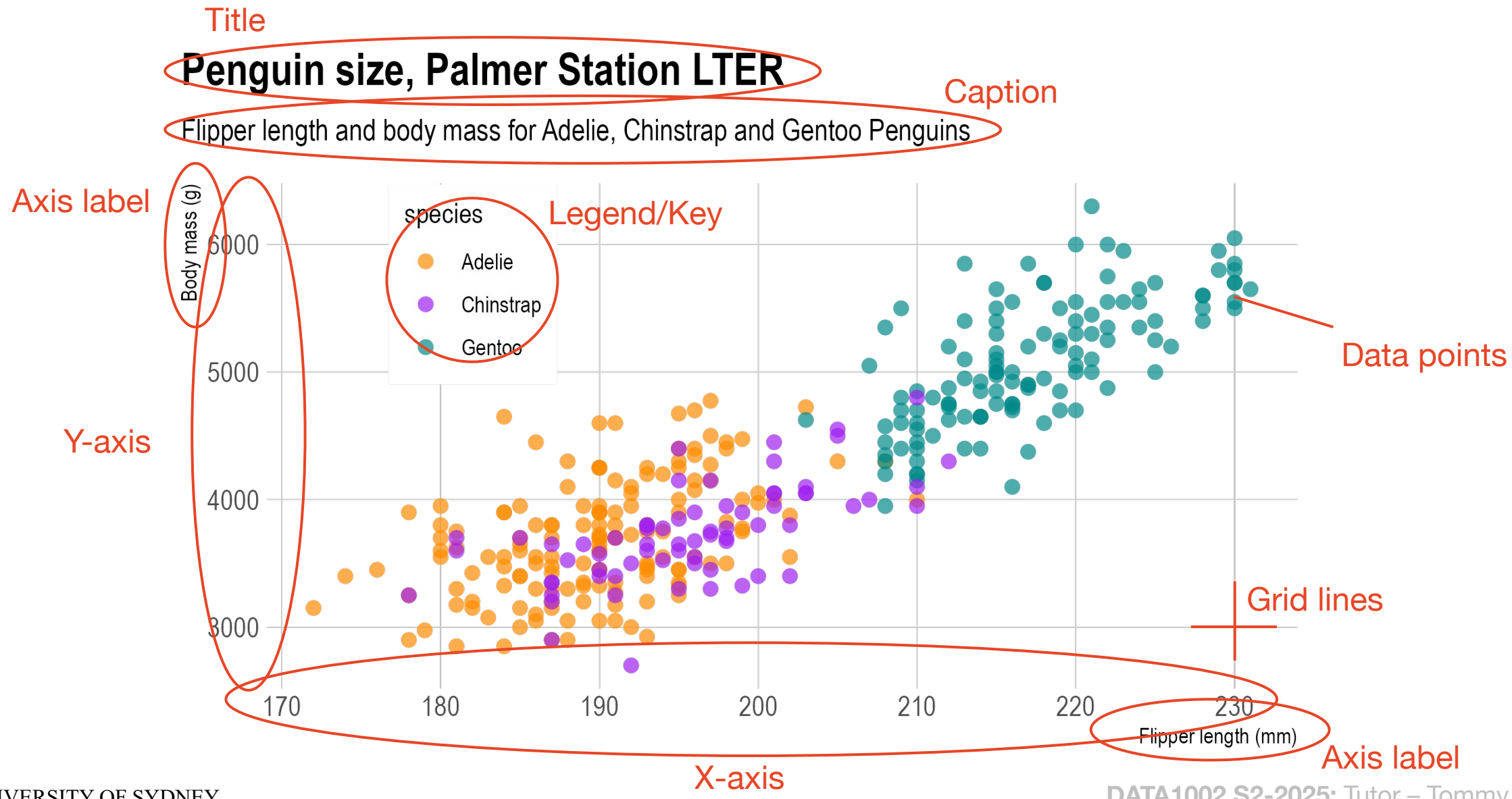
What is a Chart?



What is a Chart?

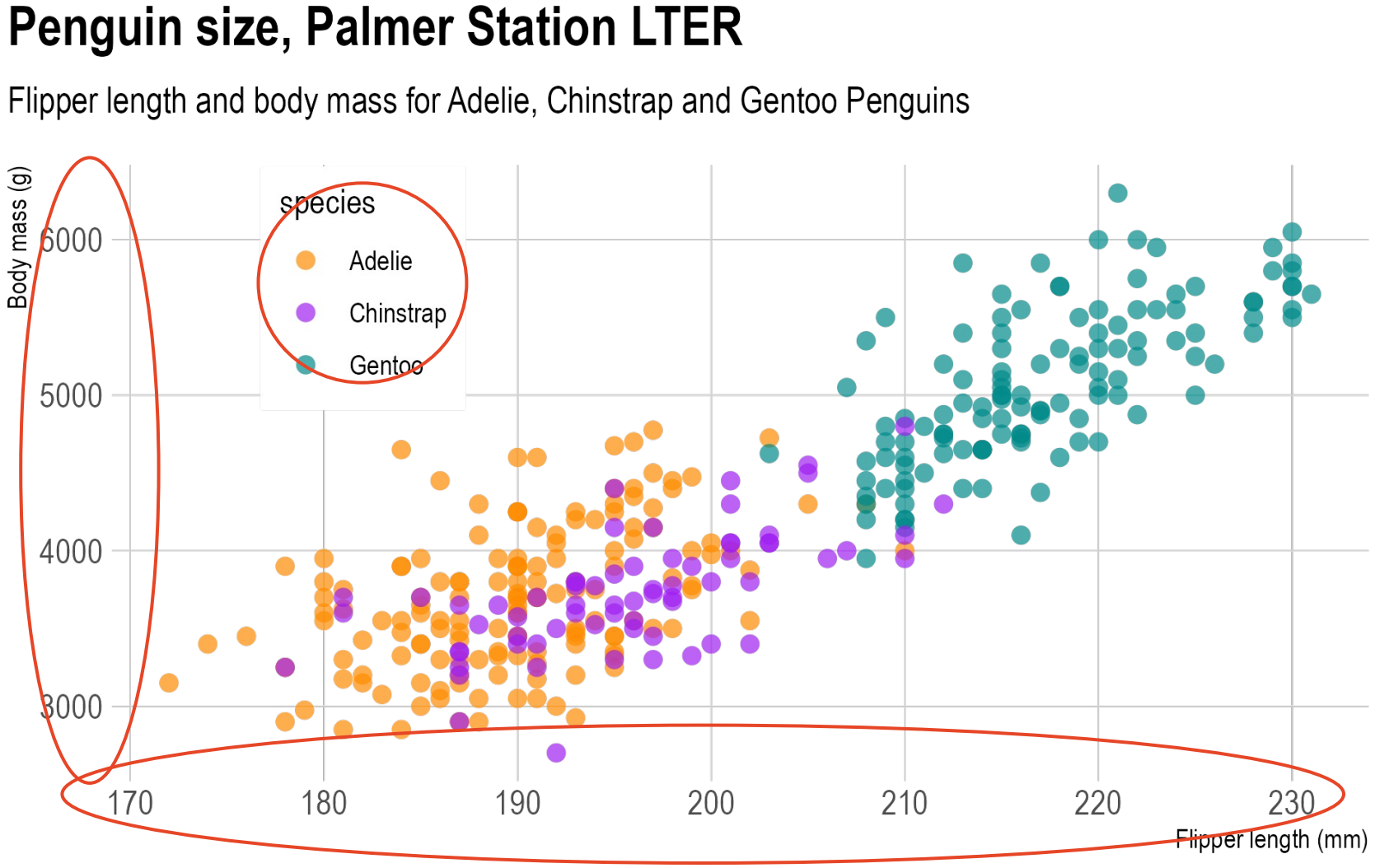


What is a Chart?



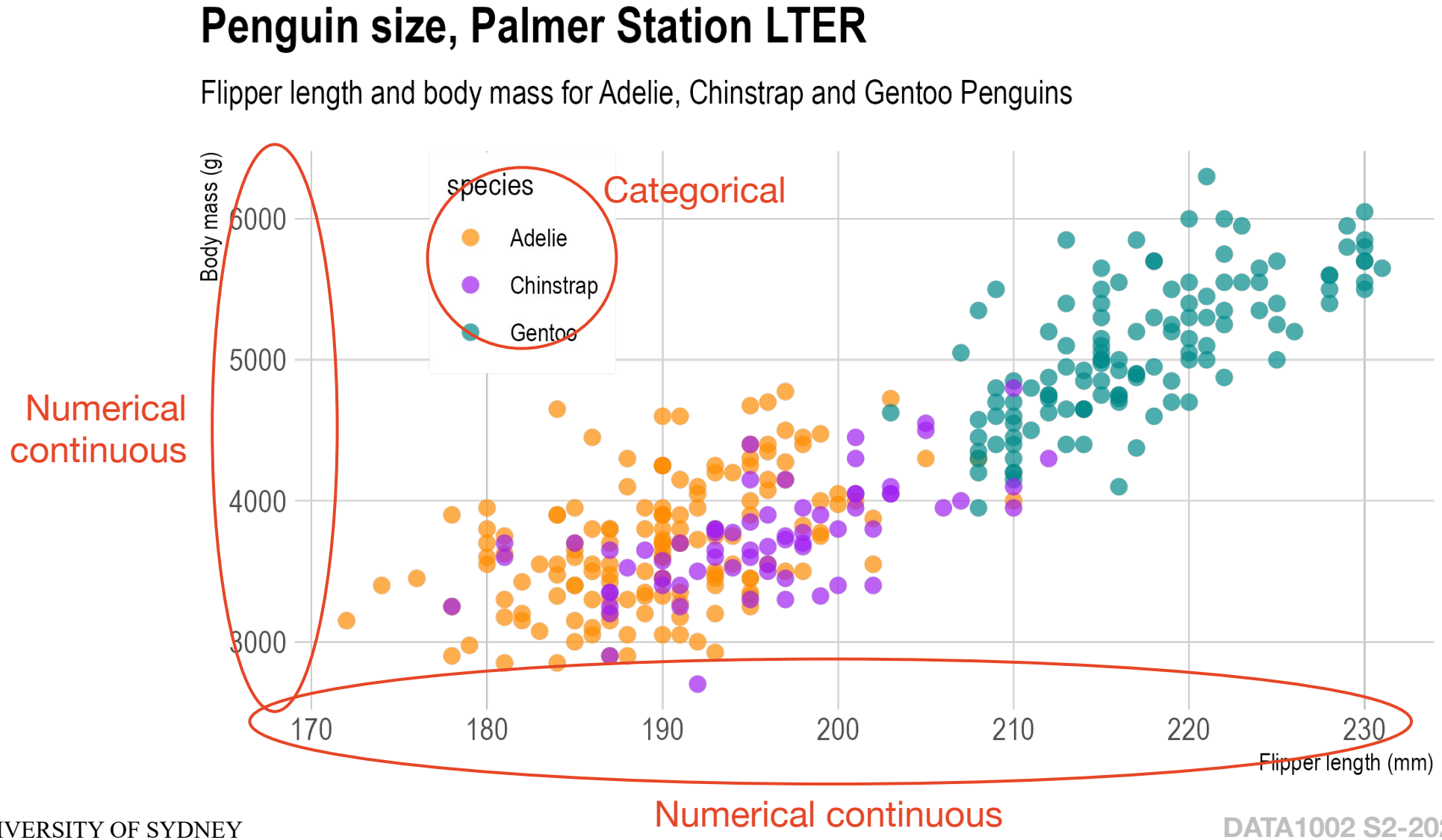
What is a Chart?

What is the type of data we have?
How can we best represent that data?



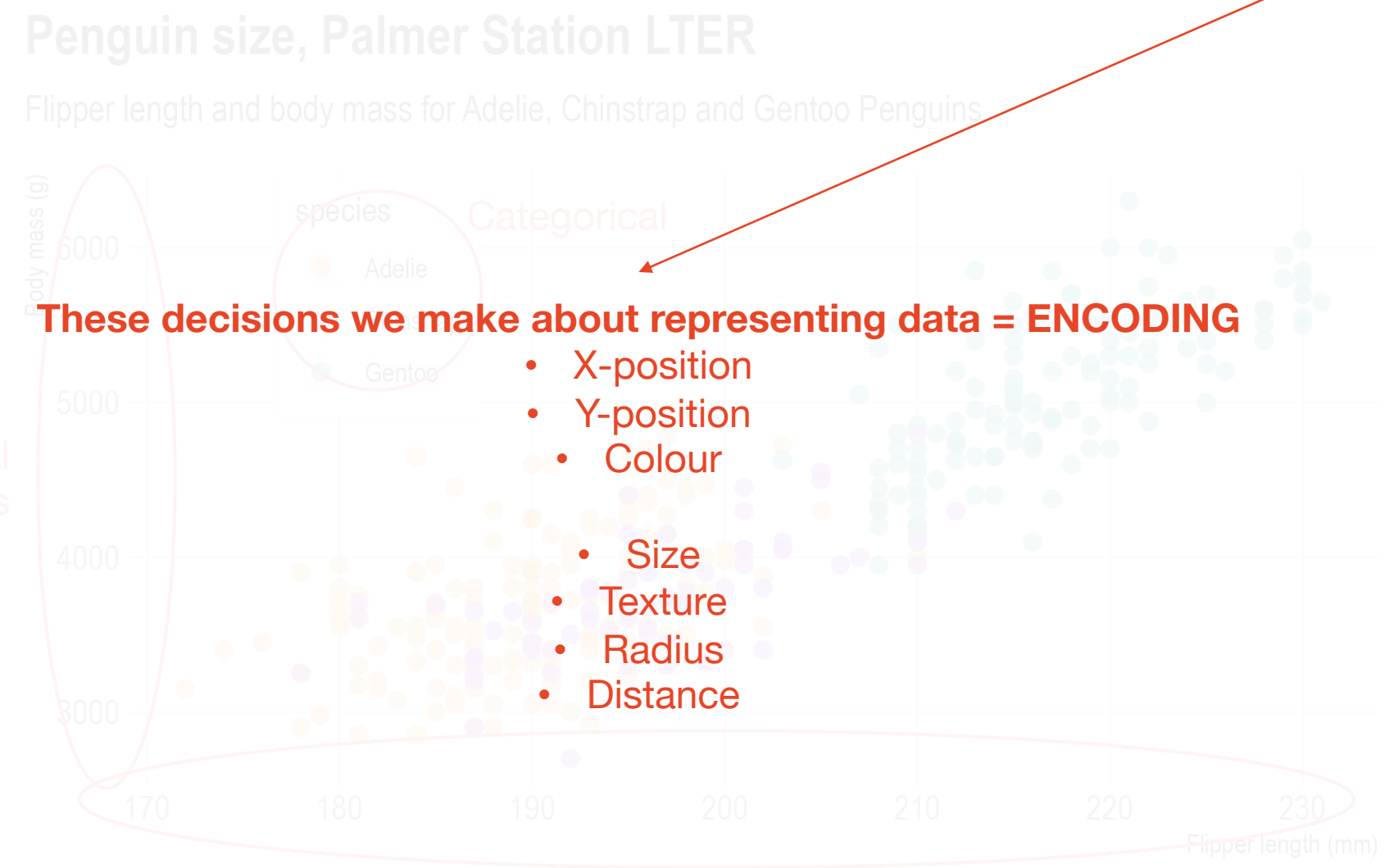
What is a Chart?

What is the type of data we have?
How can we best represent that data?



What is a Chart?

What is the type of data we have?
How can we best represent that data?



These decisions we make about representing data = ENCODING

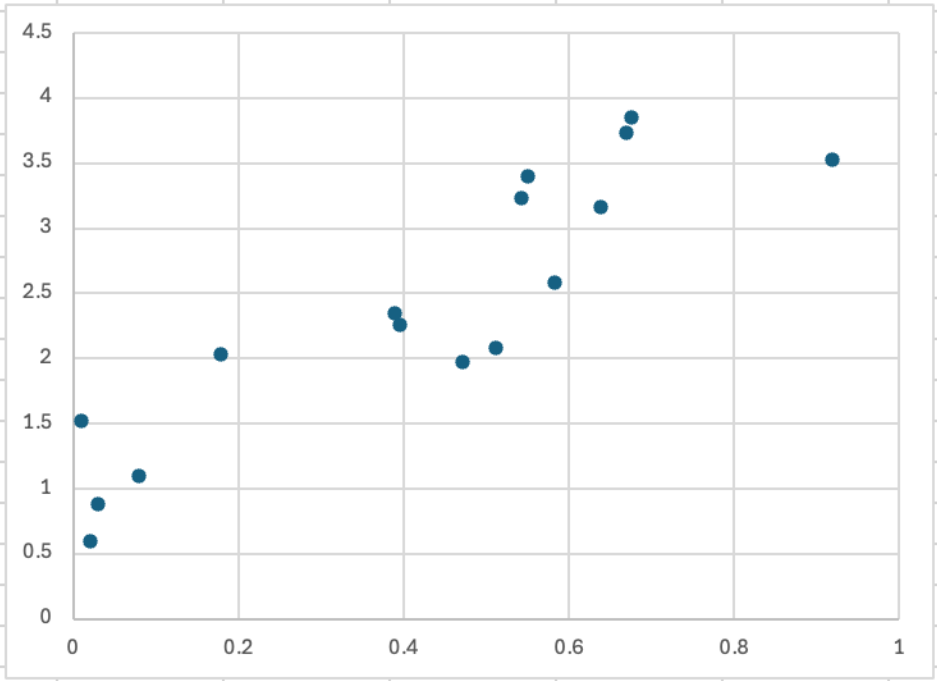
- X-position
- Y-position
- Colour

- Size
- Texture
- Radius
- Distance

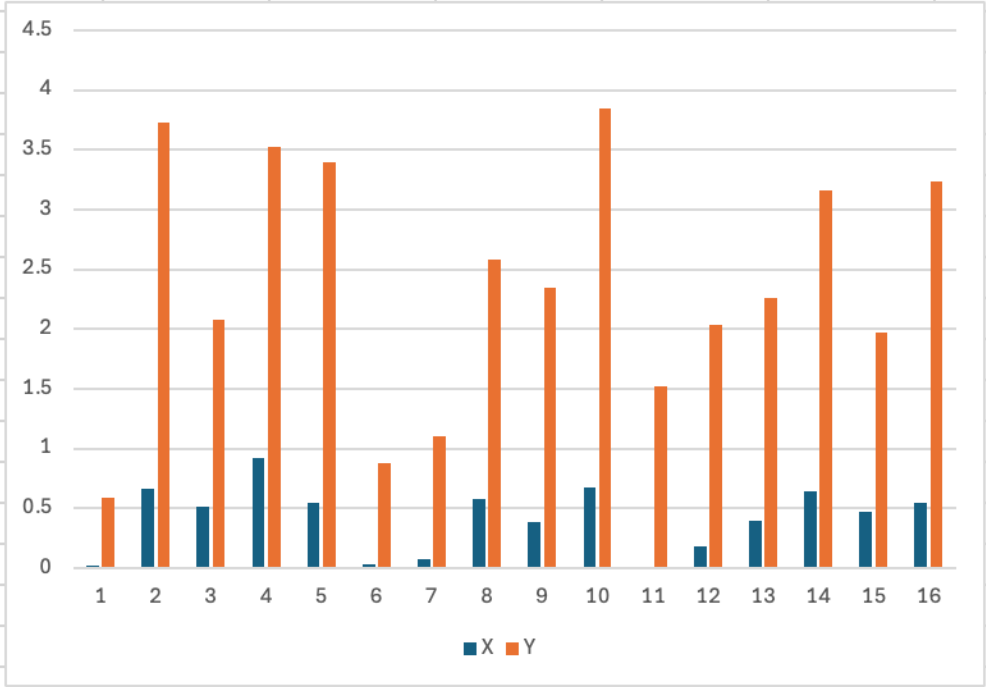
How to decide encoding methods?

Which type of plot should I pick?

X	Y
0.02115262	0.59452469
0.66967361	3.7307747
0.51128188	2.07910542
0.91838337	3.5239892
0.55068483	3.39446135
0.03021709	0.88071304
0.07984255	1.09886156
0.58280969	2.58033226
0.38946946	2.34308904
0.67620784	3.84794921
0.01028188	1.51932511
0.17896576	2.03220989
0.39602253	2.2591193
0.63888962	3.16331261
0.47212667	1.97629893
0.54308658	3.23432797



Scatterplot

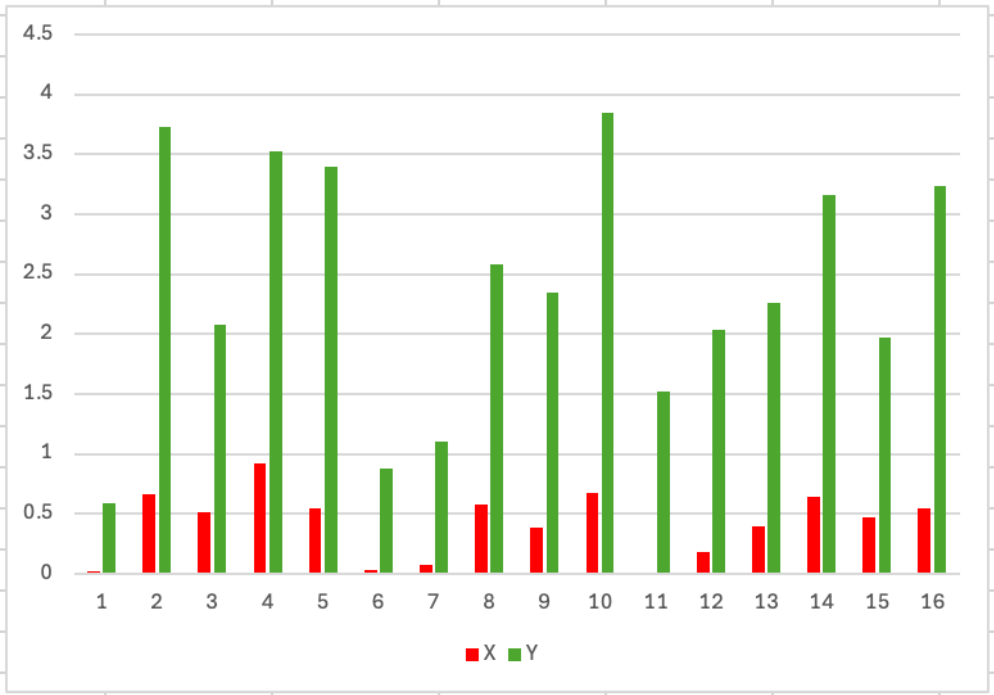


Column

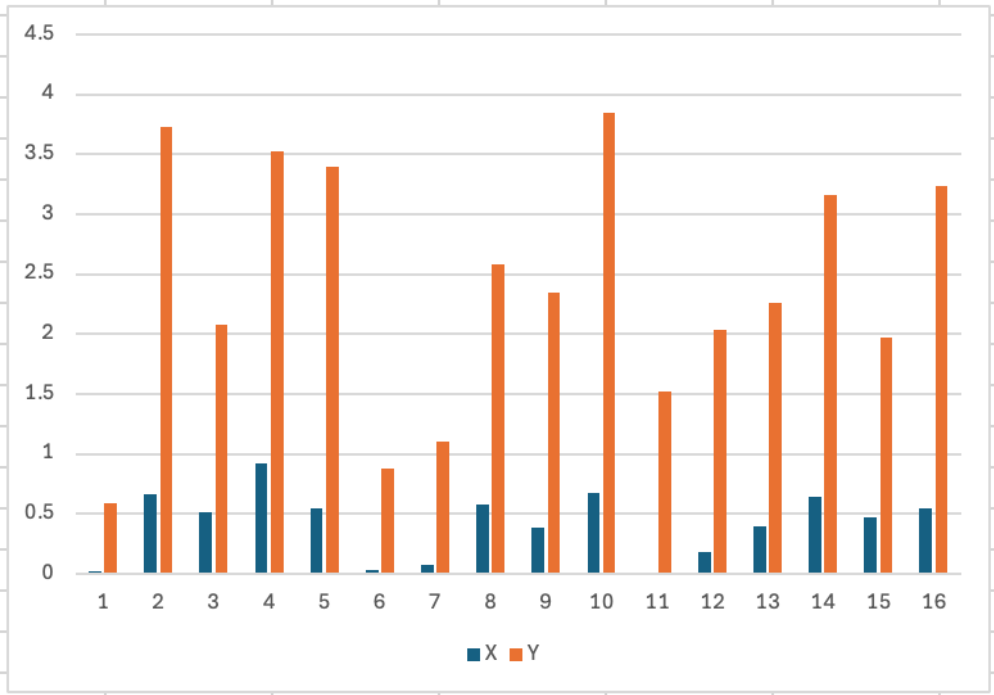
How to decide encoding methods?

Which colour scheme should I pick?

X	Y
0.02115262	0.59452469
0.66967361	3.7307747
0.51128188	2.07910542
0.91838337	3.5239892
0.55068483	3.39446135
0.03021709	0.88071304
0.07984255	1.09886156
0.58280969	2.58033226
0.38946946	2.34308904
0.67620784	3.84794921
0.01028188	1.51932511
0.17896576	2.03220989
0.39602253	2.2591193
0.63888962	3.16331261
0.47212667	1.97629893
0.54308658	3.23432797



Column

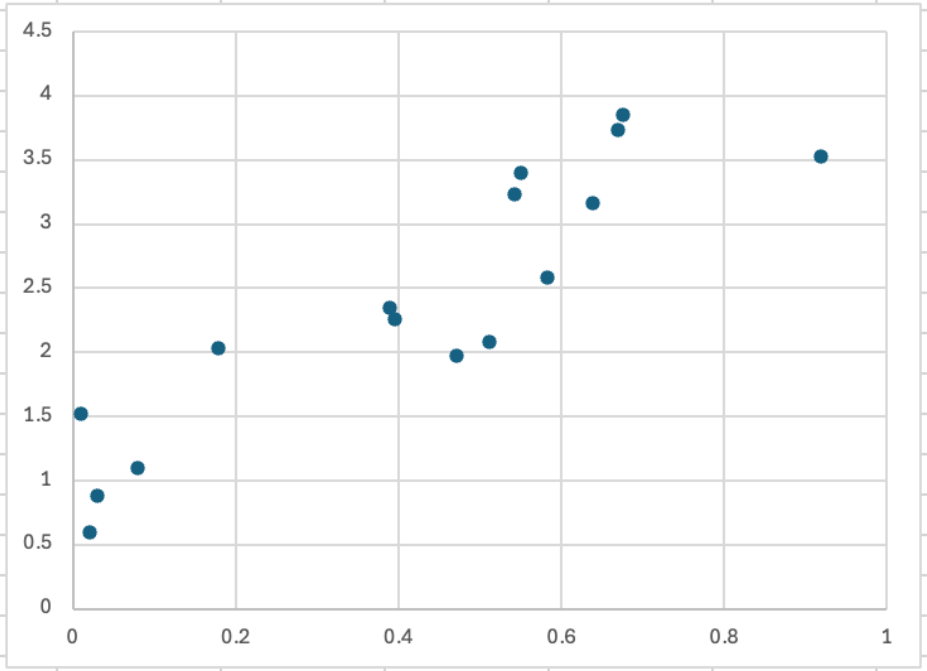


Column

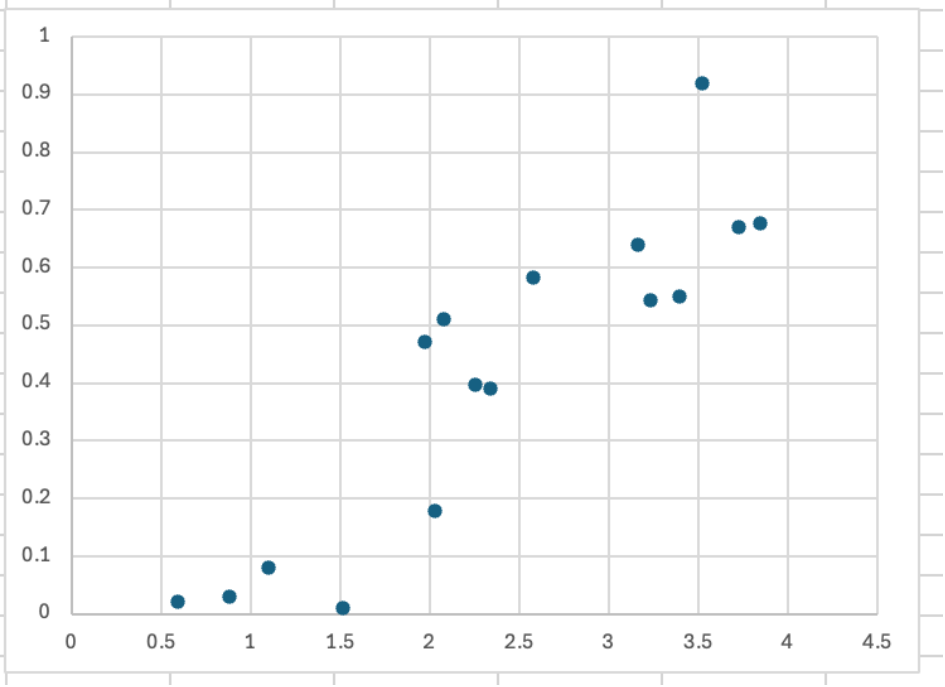
How to decide encoding methods?

Which variable should go on the x and y axis?

X	Y
0.02115262	0.59452469
0.66967361	3.7307747
0.51128188	2.07910542
0.91838337	3.5239892
0.55068483	3.39446135
0.03021709	0.88071304
0.07984255	1.09886156
0.58280969	2.58033226
0.38946946	2.34308904
0.67620784	3.84794921
0.01028188	1.51932511
0.17896576	2.03220989
0.39602253	2.2591193
0.63888962	3.16331261
0.47212667	1.97629893
0.54308658	3.23432797



Scatterplot

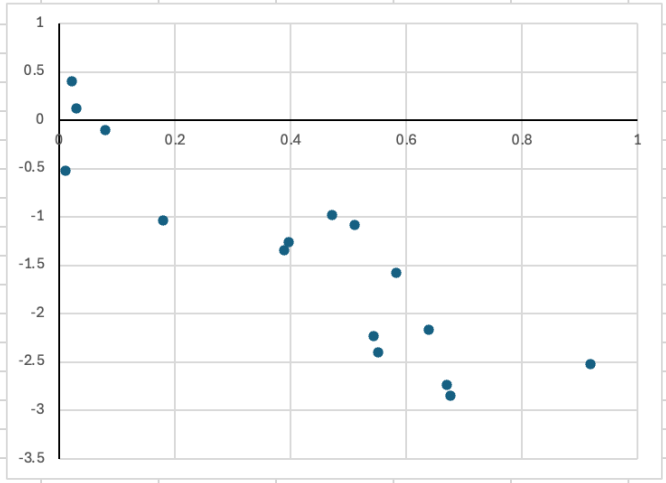


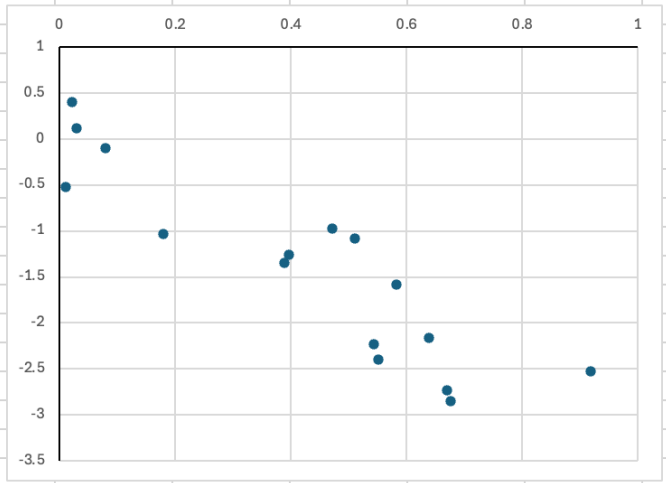
Scatterplot

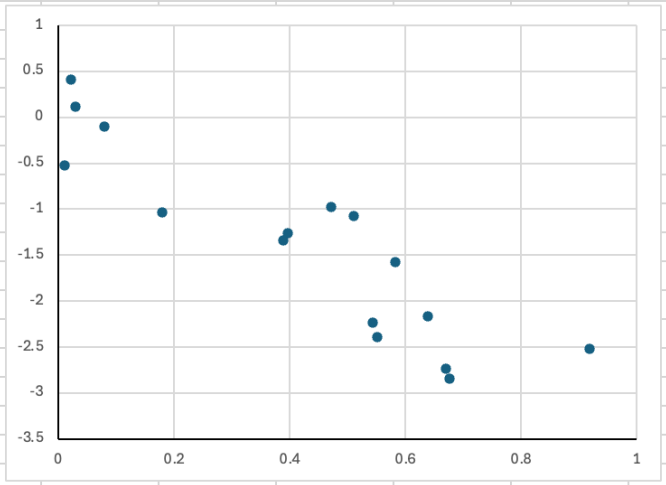
How to decide encoding methods?

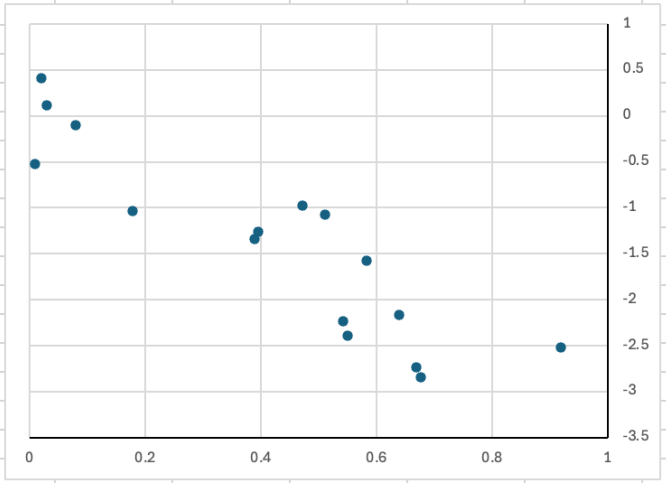
Where should the axes be located?

X	Y
0.02115262	0.40547531
0.66967361	-2.7307747
0.51128188	-1.0791054
0.91838337	-2.5239892
0.55068483	-2.3944613
0.03021709	0.11928696
0.07984255	-0.0988616
0.58280969	-1.5803323
0.38946946	-1.343089
0.67620784	-2.8479492
0.01028188	-0.5193251
0.17896576	-1.0322099
0.39602253	-1.2591193
0.63888962	-2.1633126
0.47212667	-0.9762989
0.54308658	-2.234328







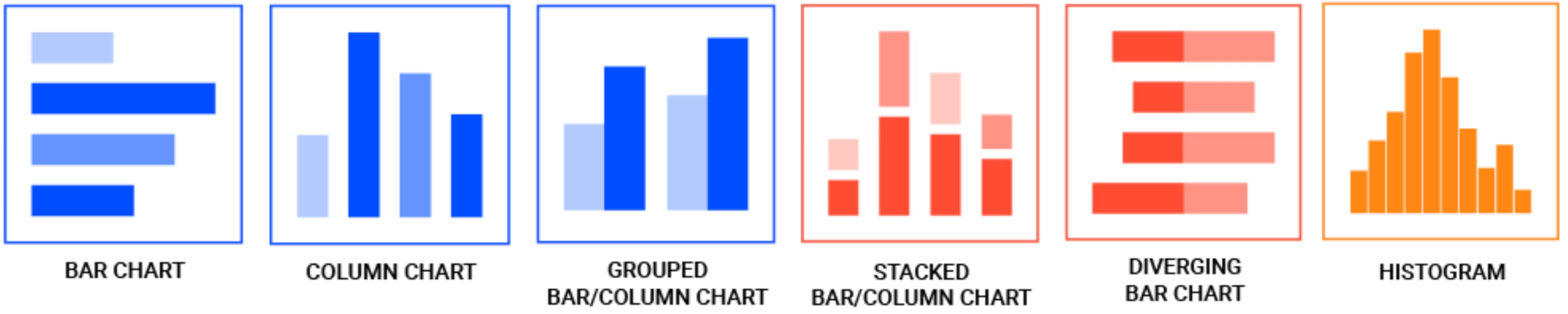


Research Exercise

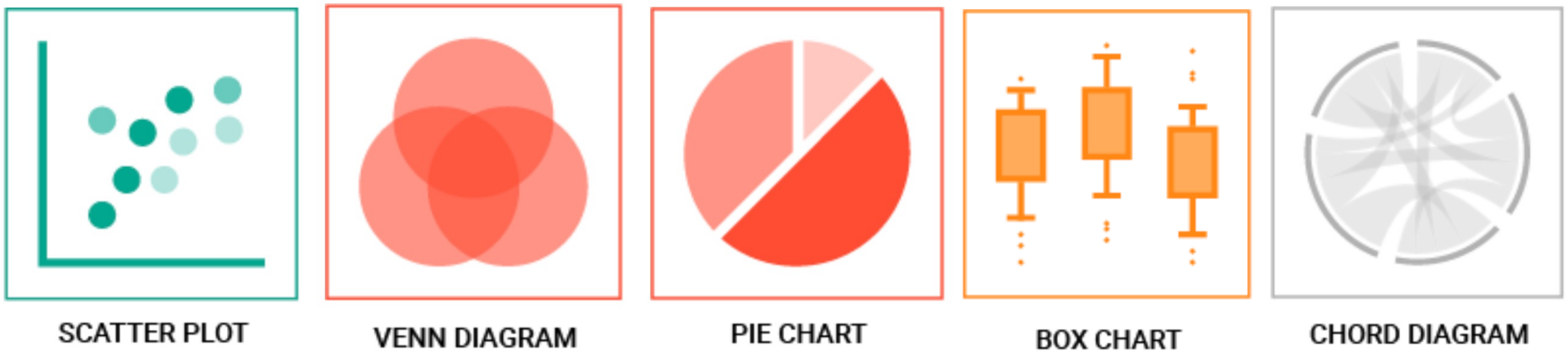
Justify why certain plots are used over another

Research to answer these two qs:

What are the use cases for each of these charts?



What types of data lend themselves to this type of chart?



Content Revision

Matplotlib

Matplotlib

- A powerful library for visualisation in Python

1. **Import libraries**
   - `import pandas as pd`
   - `import matplotlib.pyplot as plt`
2. **Use `pandas` to read and manipulate the data stored in CSV file**
   - `df = pd.read_csv(filename.csv', header = None)`
3. **Use `matplotlib.pylot` to draw plots**
   - `df.plot(kind = 'scatter', x = 'Duration', y = 'Calories')`

6 Key components of Matplotlib plotting

1. Import modules

- Matplotlib
- e.g. Pandas

```
import matplotlib.pyplot as plt
import numpy as np

plt.style.use('_mpl-gallery')

# make data:
x = 0.5 + np.arange(8)
y = [4.8, 5.5, 3.5, 4.6, 6.5, 6.6, 2.6, 3.0]

# plot
fig, ax = plt.subplots()

ax.bar(x, y, width=1, edgecolor="white", linewidth=0.7)

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 8), yticks=np.arange(1, 8))

plt.show()
```

6 Key components of Matplotlib plotting

1. Import modules
- Matplotlib
- e.g. Pandas

```
import matplotlib.pyplot as plt
import numpy as np

plt.style.use('_mpl-gallery')

# make data:
x = 0.5 + np.arange(8)
y = [4.8, 5.5, 3.5, 4.6, 6.5, 6.6, 2.6, 3.0]

# plot
fig, ax = plt.subplots()

ax.bar(x, y, width=1, edgecolor="white", linewidth=0.7)

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 8), yticks=np.arange(1, 8))

plt.show()
```

2. Import data
- Define locally
- Read in

6 Key components of Matplotlib plotting

1. Import modules
- Matplotlib
- e.g. Pandas

```python
import matplotlib.pyplot as plt
import numpy as np

plt.style.use('_mpl-gallery')

# make data:
x = 0.5 + np.arange(8)
y = [4.8, 5.5, 3.5, 4.6, 6.5, 6.6, 2.6, 3.0]

# plot
fig, ax = plt.subplots()

ax.bar(x, y, width=1, edgecolor="white", linewidth=0.7)

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 8), yticks=np.arange(1, 8))

plt.show()
```

2. Import data
- Define locally
- Read in

3. Define figure and axes of plt object

6 Key components of Matplotlib plotting

1. Import modules
- Matplotlib
- e.g. Pandas

```
import matplotlib.pyplot as plt
import numpy as np

plt.style.use('_mpl-gallery')

# make data:
x = 0.5 + np.arange(8)
y = [4.8, 5.5, 3.5, 4.6, 6.5, 6.6, 2.6, 3.0]

# plot
fig, ax = plt.subplots()

ax.bar(x, y, width=1, edgecolor="white", linewidth=0.7)

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 8), yticks=np.arange(1, 8))

plt.show()
```

2. Import data
- Define locally
- Read in

3. Define figure and axes of plt object

4. Pick chart type

6 Key components of Matplotlib plotting

1. Import modules
- Matplotlib
- e.g. Pandas

```
import matplotlib.pyplot as plt
import numpy as np

plt.style.use('_mpl-gallery')

# make data:
x = 0.5 + np.arange(8)
y = [4.8, 5.5, 3.5, 4.6, 6.5, 6.6, 2.6, 3.0]

# plot
fig, ax = plt.subplots()

ax.bar(x, y, width=1, edgecolor="white", linewidth=0.7)

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 8), yticks=np.arange(1, 8))

plt.show()
```

2. Import data
- Define locally
- Read in

3. Define figure and axes of plt object

4. Pick chart type

5. Set parameters

6 Key components of Matplotlib plotting

1. Import modules
- Matplotlib
- e.g. Pandas

```
import matplotlib.pyplot as plt
import numpy as np

plt.style.use('_mpl-gallery')

# make data:
x = 0.5 + np.arange(8)
y = [4.8, 5.5, 3.5, 4.6, 6.5, 6.6, 2.6, 3.0]

# plot
fig, ax = plt.subplots()

ax.bar(x, y, width=1, edgecolor="white", linewidth=0.7)

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 8), yticks=np.arange(1, 8))

plt.show()
```

2. Import data
- Define locally
- Read in

3. Define figure and axes of plt object

4. Pick chart type

5. Set parameters

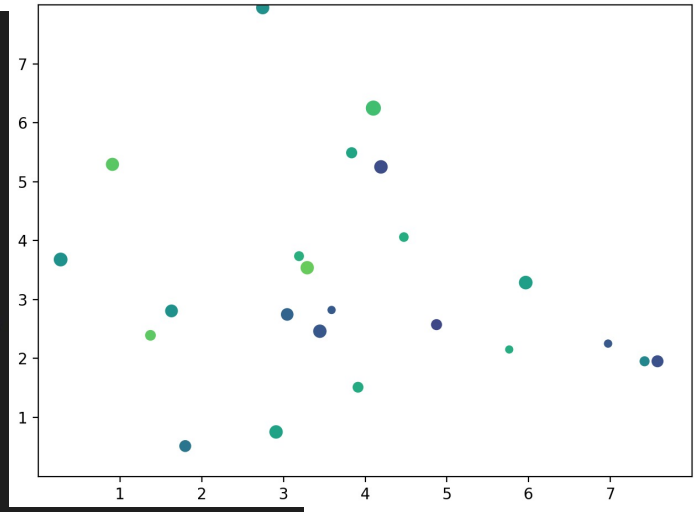
6. Show/Export/Save plot

6 Key components of Matplotlib plotting

```
 9  import matplotlib.pyplot as plt
10  import numpy as np
11
12  plt.style.use('_mpl-gallery')
13
14  # make the data
15  np.random.seed(3)
16  x = 4 + np.random.normal(0, 2, 24)
17  y = 4 + np.random.normal(0, 2, len(x))
18
19  # plot
20  fig, ax = plt.subplots()
21
22  # size and color:
23  sizes = np.random.uniform(15, 80, len(x))
24  colors = np.random.uniform(15, 80, len(x))
25  ax.scatter(x, y, s=sizes, c=colors, vmin=0, vmax=100)
26
27  ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
28         ylim=(0, 8), yticks=np.arange(1, 8))
29
30  plt.show()
```

6 Key components of Matplotlib plotting

```python
import matplotlib.pyplot as plt
import numpy as np

# make the data
np.random.seed(3)
x = 4 + np.random.normal(0, 2, 24)
y = 4 + np.random.normal(0, 2, len(x))

# plot
fig, ax = plt.subplots()

# size and color:
sizes = np.random.uniform(15, 80, len(x))
colors = np.random.uniform(15, 80, len(x))
ax.scatter(x, y, s=sizes, c=colors, vmin=0, vmax=100)

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 8), yticks=np.arange(1, 8))

fig.tight_layout()

plt.show()
```

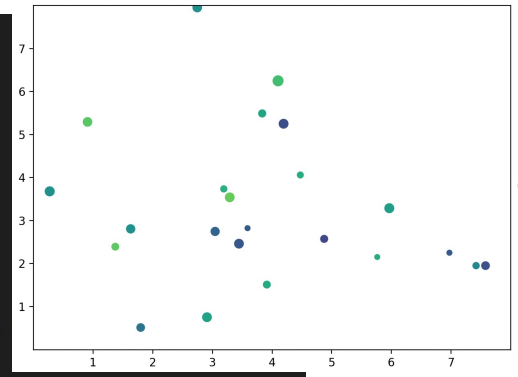


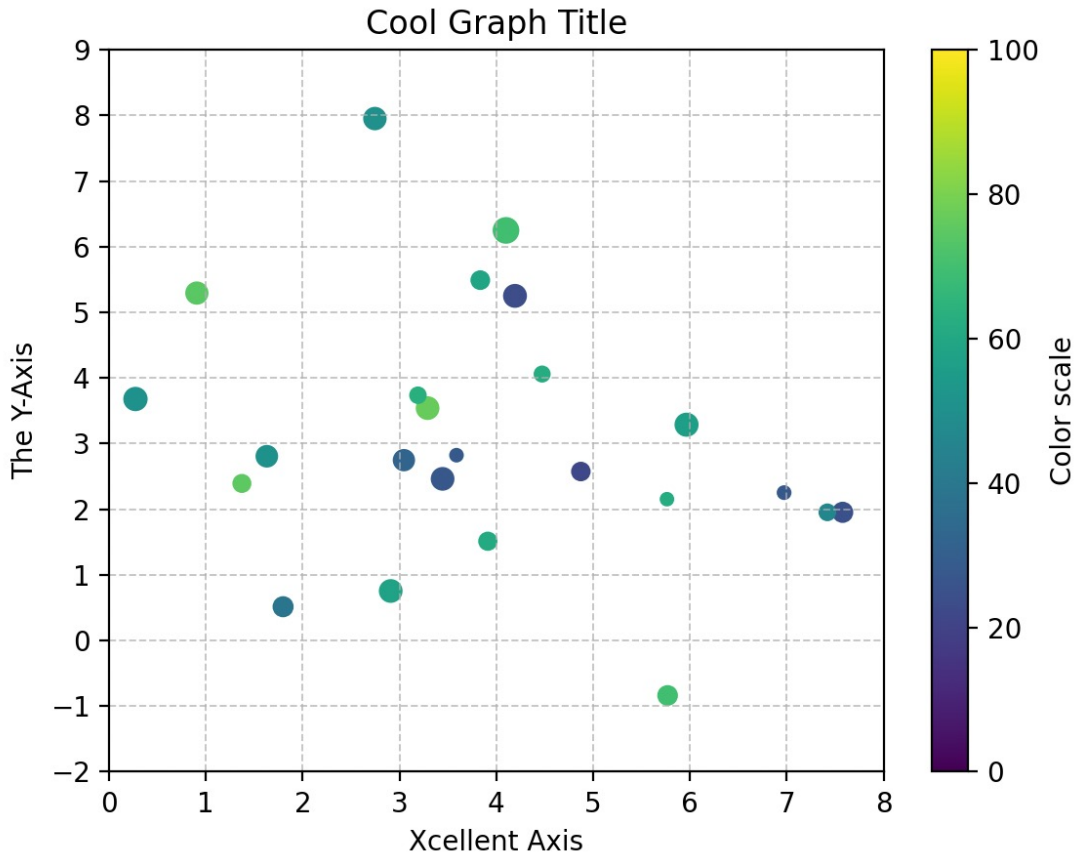
Python Exercise

Pandas & Functions/Errors

Transform this figure from this to that

```python
import matplotlib.pyplot as plt
import numpy as np

# make the data
np.random.seed(3)
x = 4 + np.random.normal(0, 2, 24)
y = 4 + np.random.normal(0, 2, len(x))

# plot
fig, ax = plt.subplots()

# size and color:
sizes = np.random.uniform(15, 80, len(x))
colors = np.random.uniform(15, 80, len(x))
ax.scatter(x, y, s=sizes, c=colors, vmin=0, vmax=100)

ax.set(xlim=(0, 8), xticks=np.arange(1, 8),
       ylim=(0, 8), yticks=np.arange(1, 8))

fig.tight_layout()

plt.show()
```





Let’s Take a Short Break!

Lab Activities

Working on Assignment 1

Activity

For Data1002, the main goal of this meeting is to finalize the report. Specifically, check each item in the marking table at https://canvas.sydney.edu.au/courses/65392/assignments/621825 . Create a to-do list and make sure all items have been covered correctly. If you are unsure, ask your tutor.

Exam-Style Questions

Question 1:

How can visualising data using different types of charts improve the understanding of complex datasets in data science projects?

Provide examples of suitable chart types for various data analysis tasks.

Exam-Style Questions

Visualising data using different types of charts can significantly enhance the comprehension of complex datasets by presenting data in an accessible and interpretable manner.

For example, line charts are ideal for **showing trends over time**, such as tracking sales figures over months. Scatter plots effectively **display relationships between two variables, useful for identifying correlations** in datasets, like age and income levels. Bar charts are excellent for comparing categorical data, such as sales across different regions. Pie charts, though less effective for precise comparisons, **can illustrate proportions within a whole, like market share distribution**.

By selecting the appropriate chart type, data scientists can highlight key insights, making complex data more understandable and actionable.

Exam-Style Questions

Question 2 [DATA1002]:

Discuss the role of ethical considerations in data visualisation within data science projects.

How can misleading charts impact decision-making, and what steps can be taken to ensure ethical visualisation practices?

Exam-Style Questions

Ethical considerations in data visualisation are crucial to maintain trust and integrity in data science projects. Misleading charts can distort data interpretation, leading to poor decision-making.

For instance, truncated y-axes can exaggerate differences between data points, while improper scaling can misrepresent trends. To ensure ethical visualisation practices, data scientists should a**dhere to principles of clarity and accuracy, such as using appropriate scales, avoiding deceptive design choices, and providing necessary context through labels and legends**.

Additionally, transparency about data sources and methods used to create visualisations helps stakeholders understand the limitations and reliability of the presented data. Ethical visualisation fosters informed decision-making and upholds the credibility of the data science profession.

That’s it folks!

Remaining Ed Lessons, Questions, Assignment etc.