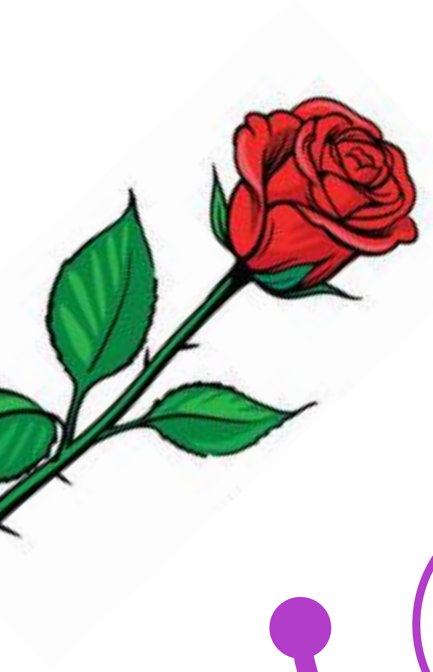
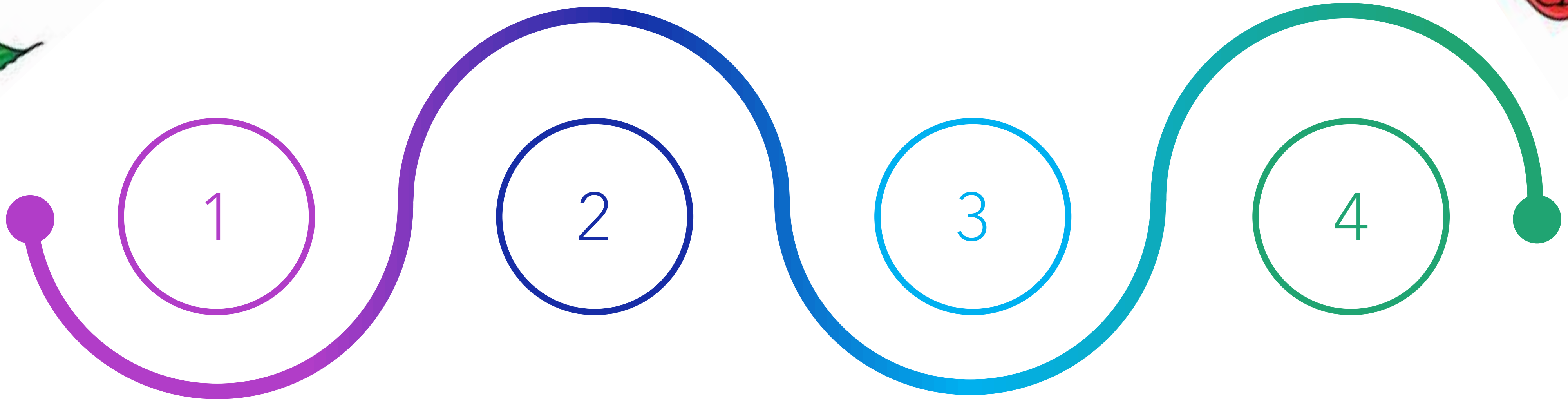
INTRODUCTION TO DATA SCIENCE WITH PYTHON





Welcome!

Please find a seat and get settled

Start your laptop and check internet

We will start at 9.00am!

Welcome!

1

2

3

4

Presenter introduction

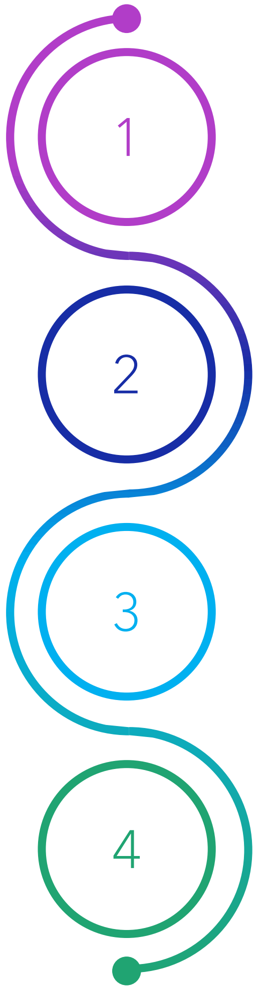
- Lindsay Forestell
- Stuart Hemerling

Acknowledgement

- Indigenous Peoples and Lands (cirnac.gc.ca)
- The First Nations Principles of OCAP® - The First Nations Information Governance Centre (fnigc.ca)
- Native-Land.ca | Our home on native land

The Learning Environment

1. Content is new, developed by the DSP team
2. Geared to the practice of data science
3. Course is interactive – no textbook - you will be coding and saving your work for future reference
4. We will aim to start/end sessions on time



Day 1

- **9:00-9:30** - Course Introduction (20 min) (Stuart)
- **9:30-10:30** - Getting Up and Running (60 min) (Lindsay)
- **10:30-10:45** BREAK ☕
- **10:45-Noon** Working with Code (90 min) (Stuart)
- **Noon-I:00** LUNCH 🍍
- **1:00-2:30** Core Data Structure Concepts (90 min) (Lindsay)
- **2:30-2:45** BREAK 🍩
- **2:45-4:30** Getting Data (90 min) (Lindsay)

Day 2

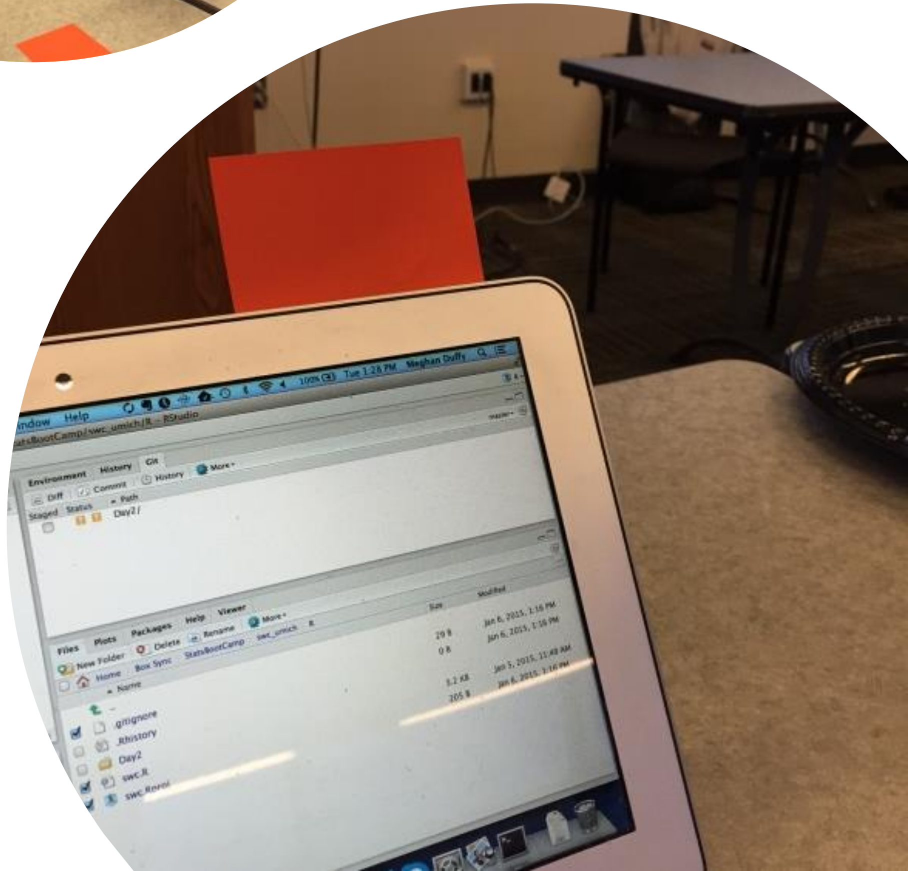
- **9:00-10:30** - Data Cleaning (90 min) (Stuart)
- **10:30-10:45** BREAK ☕
- **10:45-Noon** Exploring Data with Pandas (90 min) (Stuart)
- **Noon-1:00** LUNCH 🍍
- **1:00-2:30** Graphical Depictions of Data (90 min) (Lindsay)
- **2:30-2:45** BREAK 🍩
- **2:45-4:30** Bonus Content: TBD (90 min) (TBD – Learners vote on topic end of Day 1!)

This is a general guide of what will be covered – some sections may start or end earlier or later than outlined!

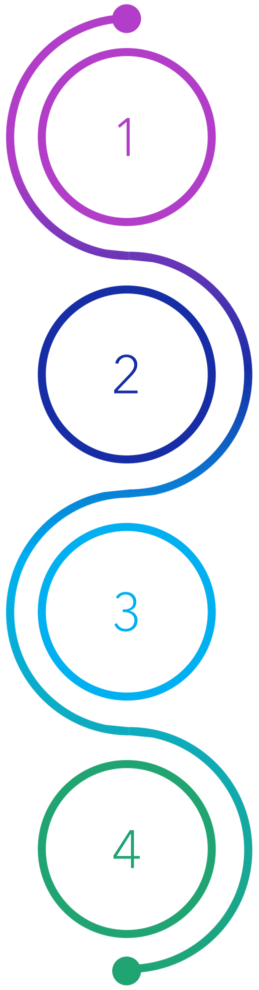
Sticky Notes

- As Status Flags
  - Green is "*I'm done!*"
  - Red is "*I need help!*"
  - No flag means "*Working on it*"

- Please put your name on both so we can get to know each other properly!

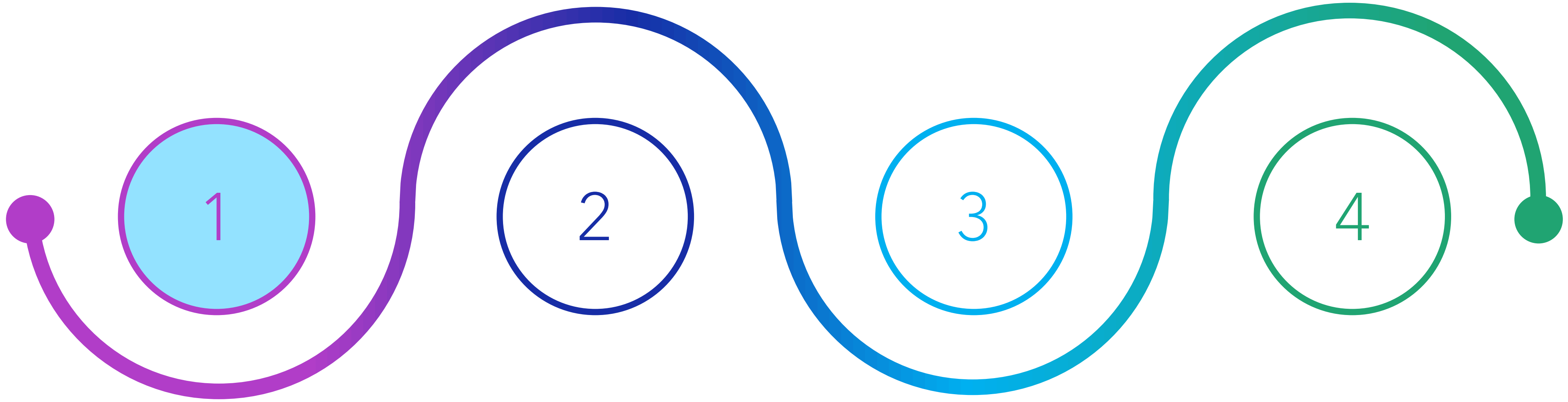


Collaborative Note Taking: Etherpad



- https://etherpad.wikimedia.org/p/bcgov-intro-python-jan23
- Where we can share challenges and solutions throughout course
- Let's start by going there!

GETTING UP AND RUNNING



Introduction to Python

Get And Clean Data

Understand and Analyze Data

Advanced Topics

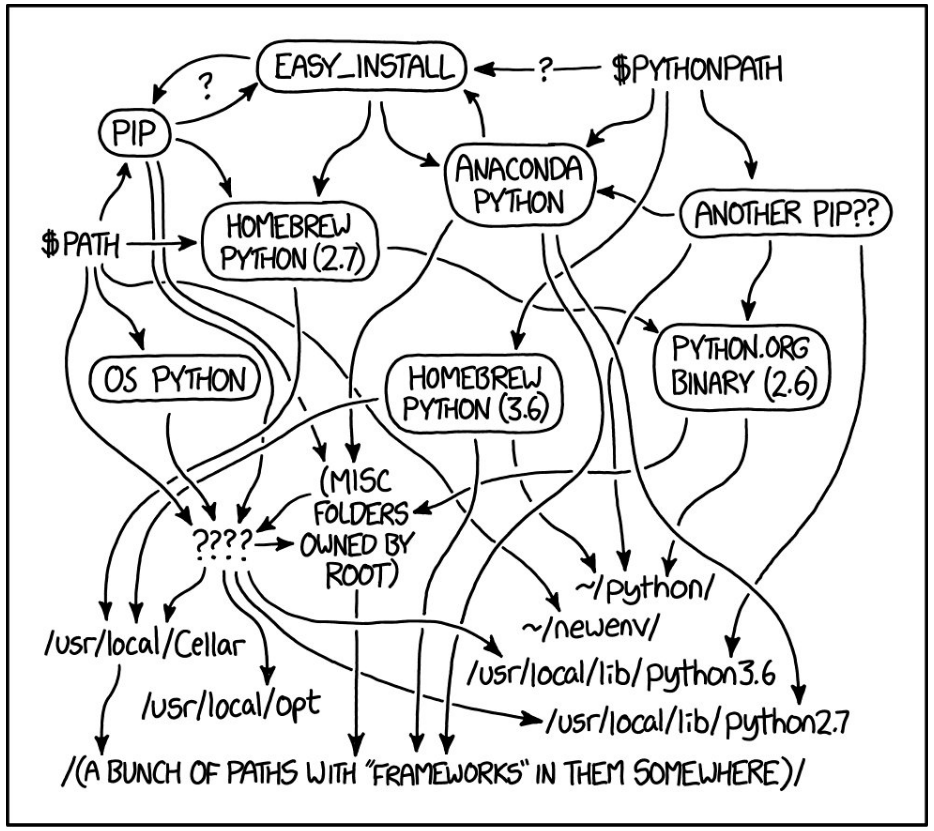
Why use conda?

1

2

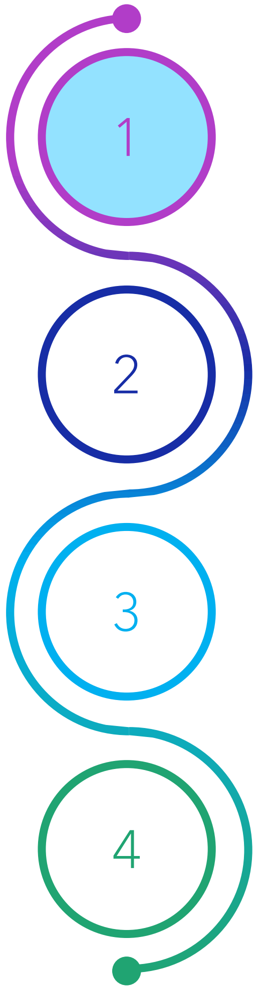
3

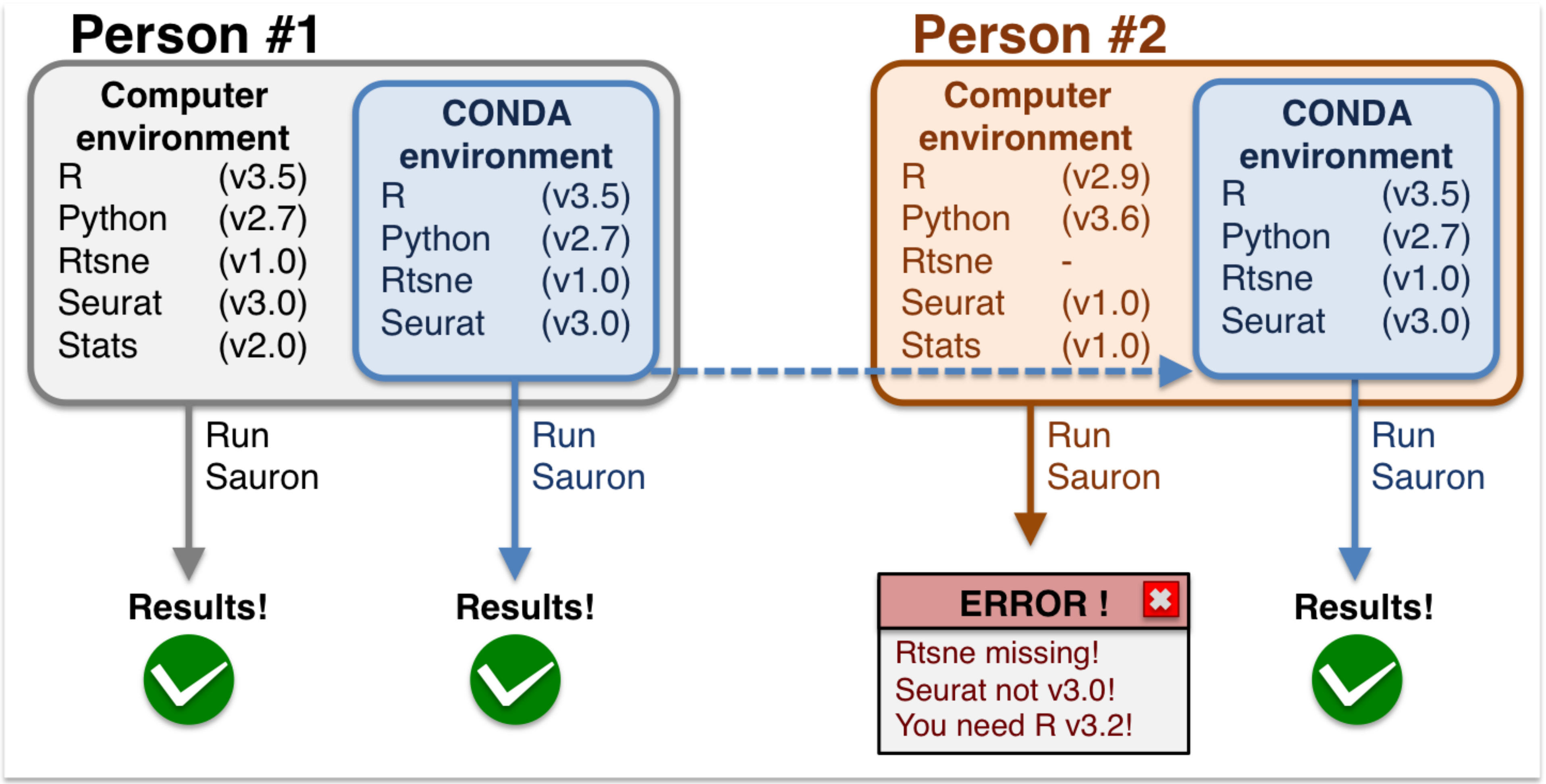
4



MY PYTHON ENVIRONMENT HAS BECOME SO DEGRADED THAT MY LAPTOP HAS BEEN DECLARED A SUPERFUND SITE.

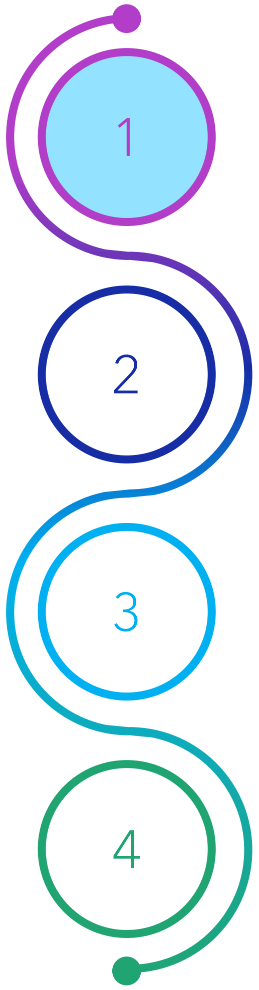
Why use conda?





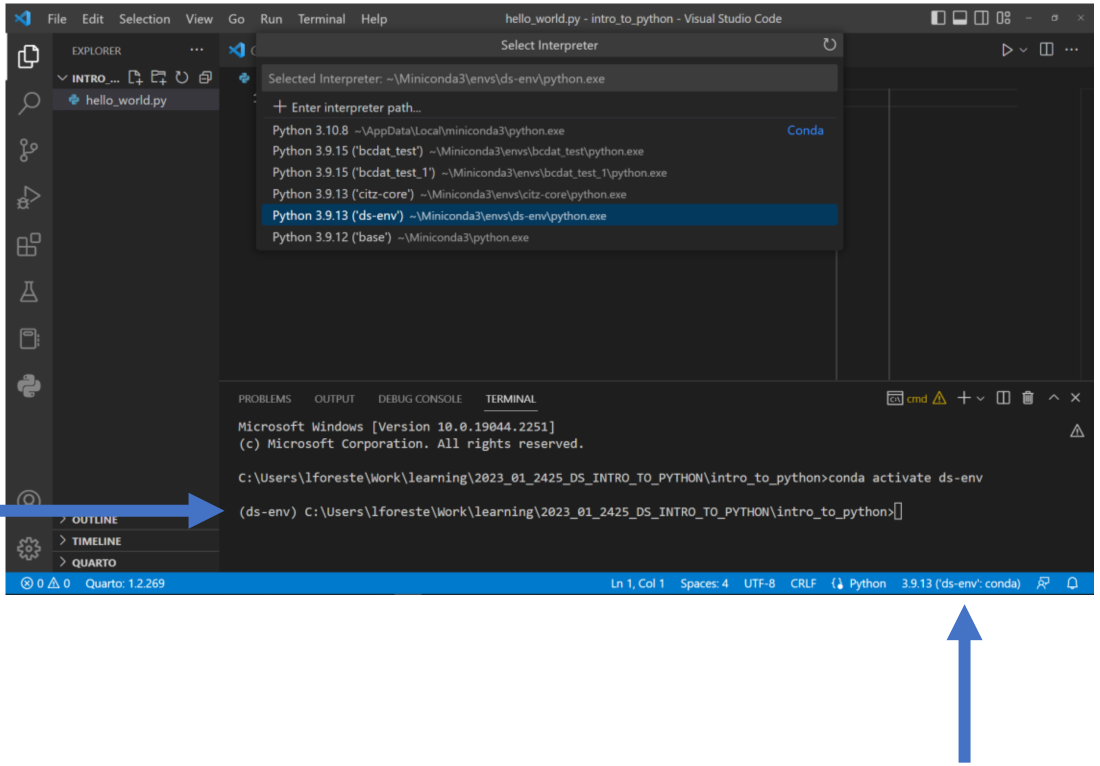
Rtsne missing!
Seurat not v3.0!
You need R v3.2!

Remember to double check:



Does my command line start with the right environment name?

Is VS Code using the right environment?



hello_world.py - intro_to_python - Visual Studio Code

Select Interpreter

Selected Interpreter: ~\Miniconda3\envs\ds-env\python.exe

+ Enter interpreter path...

Python 3.10.8 ~\AppData\Local\miniconda3\python.exe Conda
Python 3.9.15 ('bcdat_test') ~\Miniconda3\envs\bcdat_test\python.exe
Python 3.9.15 ('bcdat_test_1') ~\Miniconda3\envs\bcdat_test_1\python.exe
Python 3.9.13 ('citz-core') ~\Miniconda3\envs\citz-core\python.exe
Python 3.9.13 ('ds-env') ~\Miniconda3\envs\ds-env\python.exe
Python 3.9.12 ('base') ~\Miniconda3\python.exe

```
Microsoft Windows [Version 10.0.19044.2251]
(c) Microsoft Corporation. All rights reserved.

C:\Users\lforeste\Work\learning\2023_01_2425_DS_INTRO_TO_PYTHON\intro_to_python>conda activate ds-env

(ds-env) C:\Users\lforeste\Work\learning\2023_01_2425_DS_INTRO_TO_PYTHON\intro_to_python>
```

⊗ 0 ⚠ 0 Quarto: 1.2.269 Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Python 3.9.13 ('ds-env': conda)

Different Environments for Different Purposes

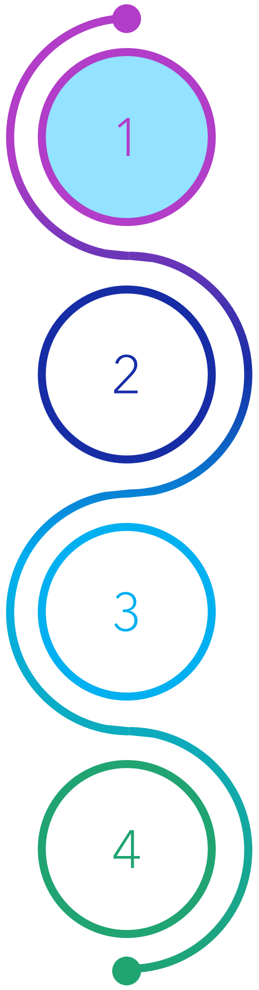
Jupyter Lab

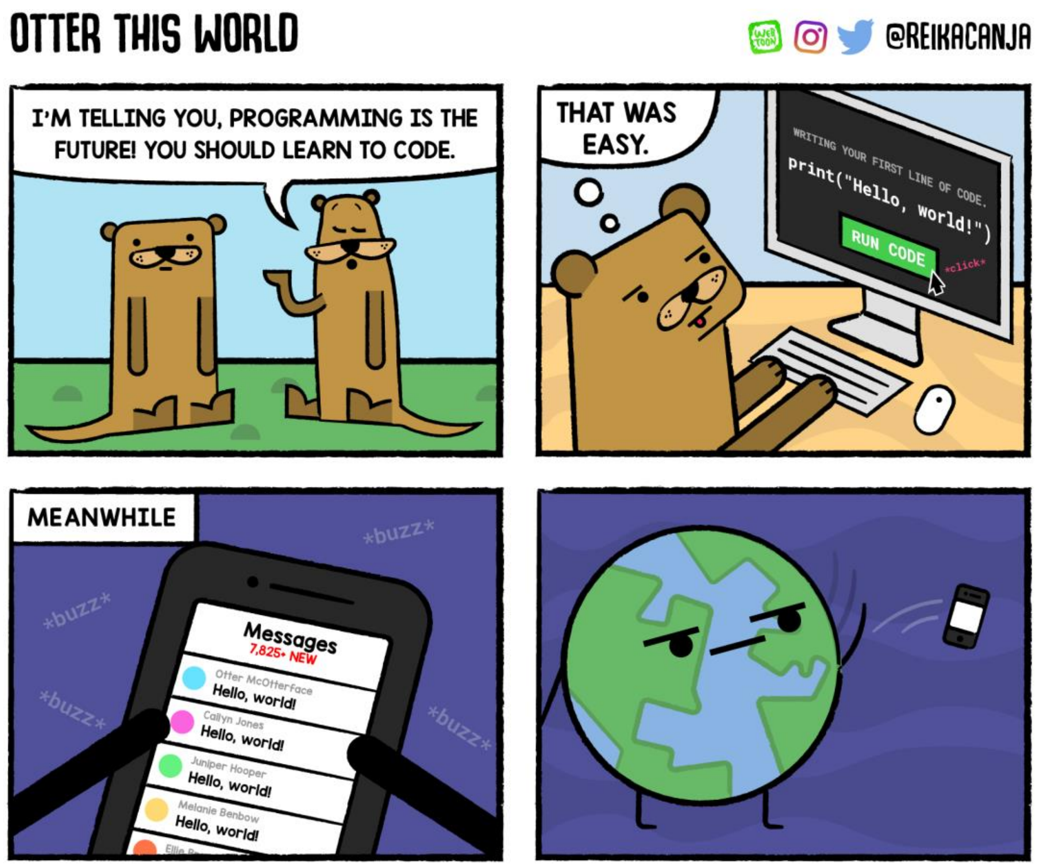
- Barebones IDE
- Useful for exploratory and documented analysis done in **notebooks**
- .ipynb files
- Displays plots inline, between **cells** of code

VS Code

- More wholistic IDE (integrated developing environment)
- Useful for running, testing, debugging **scripts**
- .py files
- Displays plots externally
- No notion of code cells

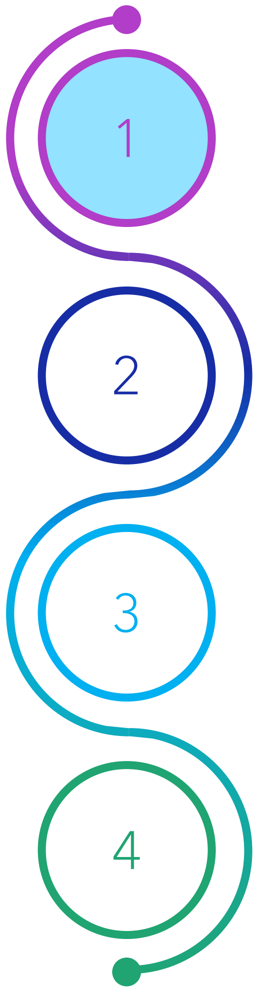
Hello, world!





```
print("Hello, world!")
```

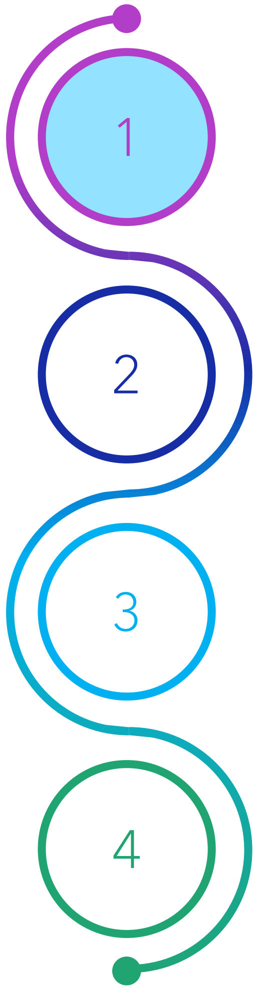
Challenge 1



In the **hello_world.py** file, run the second line of code.

Add a third line of code that prints your favourite TV show, and print this line.

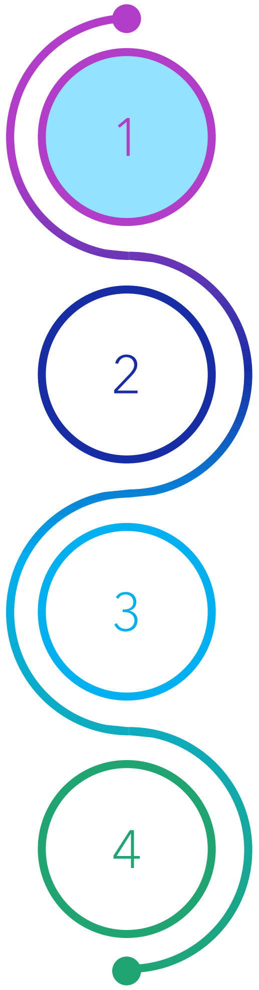
Challenge 2



Try clicking the play button again.

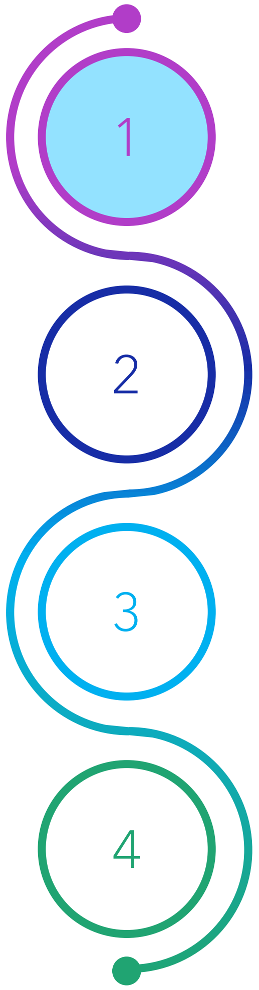
What happens here? Can you explain why?

Challenge 3



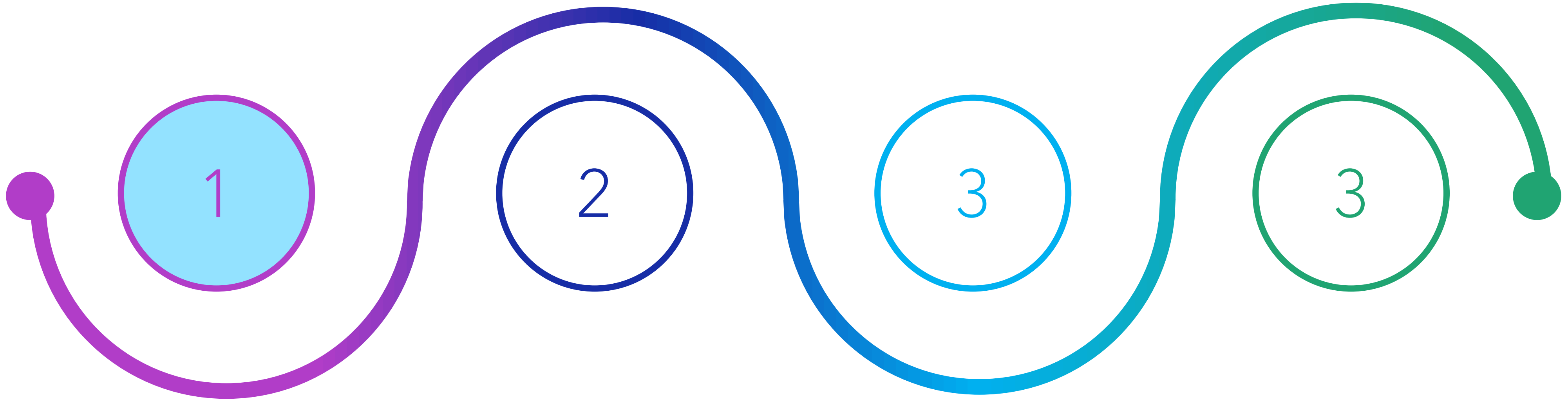
Rename the notebook to **hello_world.ipynb**

Challenge 4



Print ‘Hello World’ inside the Jupyter Notebook.

WORKING WITH CODE



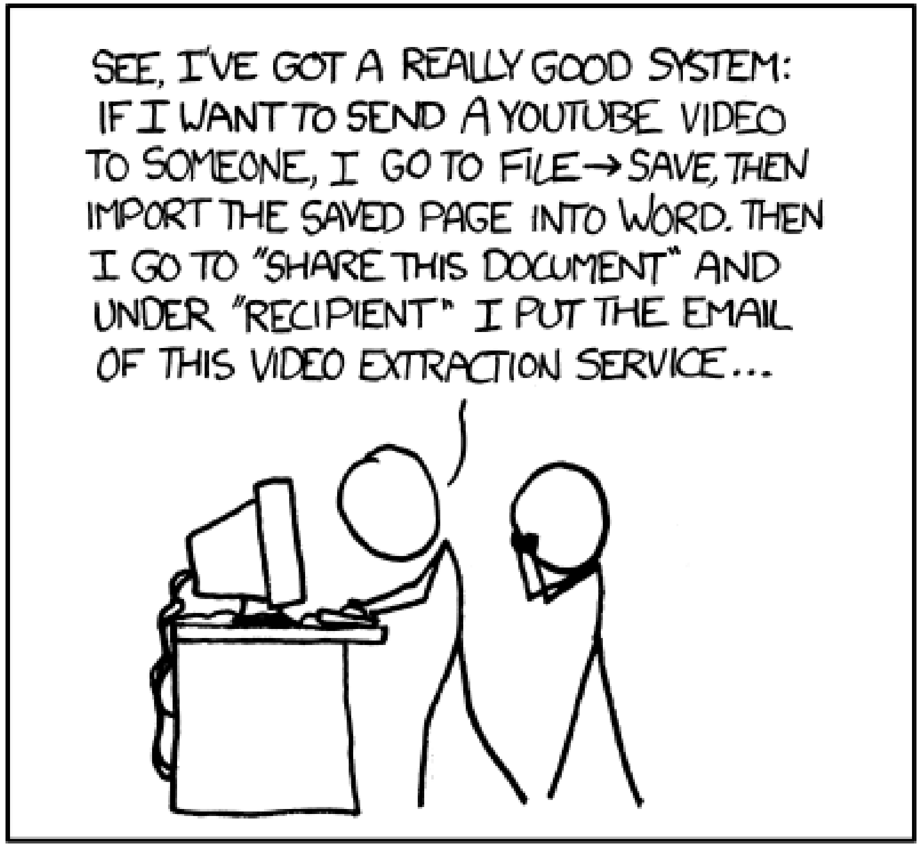
Introduction to Python

Get And Clean Data

Understand and Analyze Data

Advanced Topics

Organizing Matters


SEE, I'VE GOT A REALLY GOOD SYSTEM: IF I WANT TO SEND A YOUTUBE VIDEO TO SOMEONE, I GO TO FILE→SAVE, THEN IMPORT THE SAVED PAGE INTO WORD. THEN I GO TO "SHARE THIS DOCUMENT" AND UNDER "RECIPIENT" I PUT THE EMAIL OF THIS VIDEO EXTRACTION SERVICE...

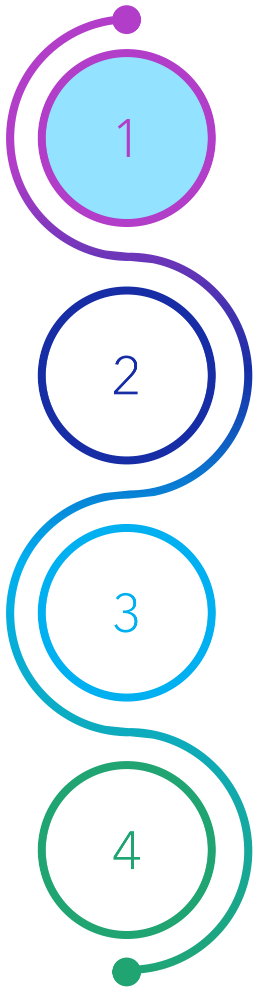
I'LL OFTEN ENCOURAGE RELATIVES TO TRY TO SOLVE COMPUTER PROBLEMS THEMSELVES BY TRIAL AND ERROR.

HOWEVER, I'VE LEARNED AN IMPORTANT LESSON: IF THEY SAY THEY'VE SOLVED THEIR PROBLEM, *NEVER* ASK HOW.

Organizing Principles

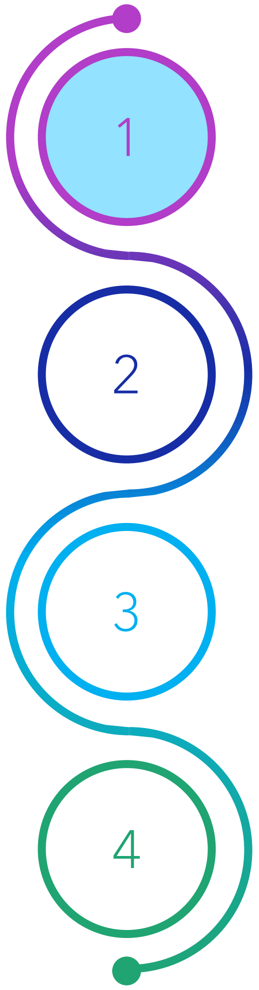
1. Treat data as read only
2. Clean and analyze your data with code
3. Treat generated output as disposable
4. Organize your files

File Organization "Template"



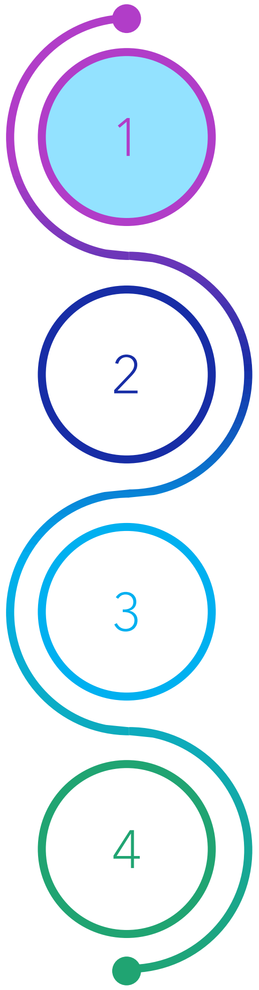
```
/MarmotsBC          top level directory,
|                   contains project name, without spaces
|
└─── /docs          supporting text and word processed documents
|                   that are meant for humans to read
|
└─── /data          the original (meta)data that acts as an input
|                   and will be treated as read-only
|
└─── /src           your scripts, may further be divided
|                   into /functions, /test, etc.
|
└─── /bin           programs brought in from elsewhere
|
└─── /output        data written by your code to this location,
|                   could be input for another process
|
└─── /analysis      the charts, stats, figures and the like
                   generated from your scripts
```

Challenge 1



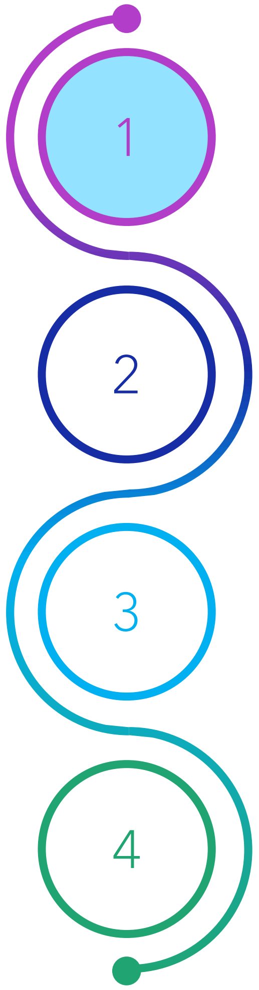
What are three activities associated with a project that are made more difficult and put at risk if project files are not kept well organized? Who is likely to be affected?

Challenge 2

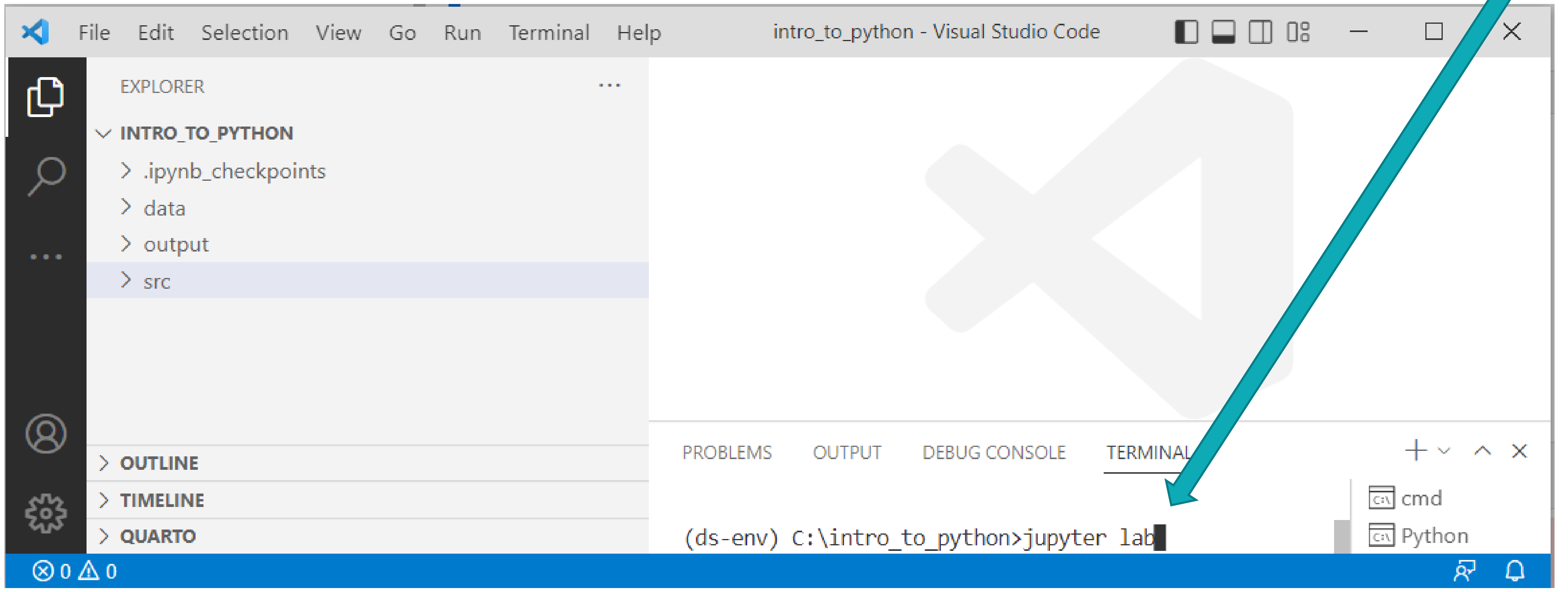


If output is generally considered the most usable thing for end-users of data, why should it be considered as disposable?

Getting Help



Let's get coding!

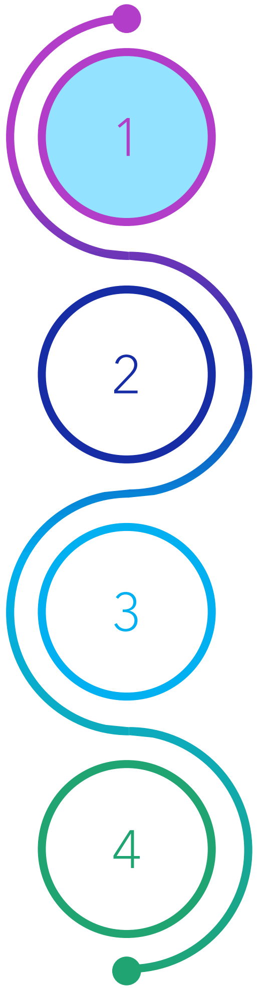


```
(ds-env) C:\intro_to_python>jupyter lab
```

Challenge 3

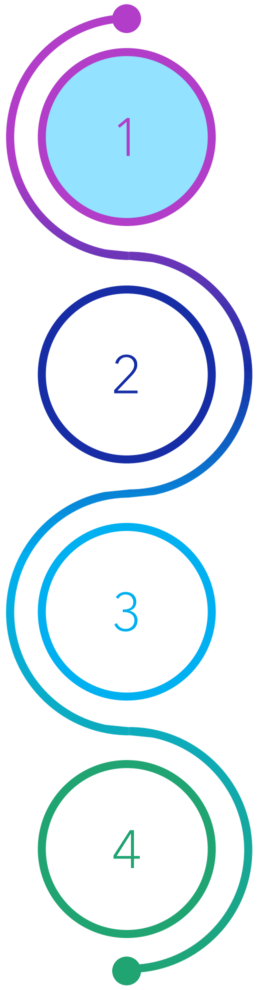
You want to find out what the **`print()`** function does. How would you look to answer this question? What does it do, does it send a print job to a printer?

Challenge 4



Someone asks you want the term “iterable” means in the context of a python function? How would you go about finding an answer? What does it mean?

Coding in Style



Let's get coding!

Challenge 5

Based on what we have covered, what, if any are the style code issues.

```
# Example 1
print (last_name)

# Example 2
celcius = (fahrenheit - 32) / 1.8

# Example 3
index = ['snail', 'pig', 'elephant', 'rabbit', 'giraffe', 'coyote', 'horse', 'giraffe', 'ping pong']
```

CORE CONCEPTS

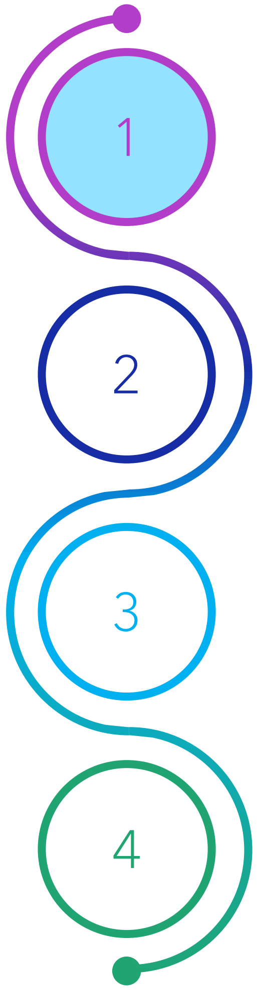
Introduction to Python

Get And Clean Data

Understand and Analyze Data

Advanced Topics

Challenge 1



So far in the notebook we should have:

```
variable_name = 'variable_value’
first_name = 'Loki’
age = 1054
```

In the next cell, run the following command:

```
print(last_name)
```

What happens?

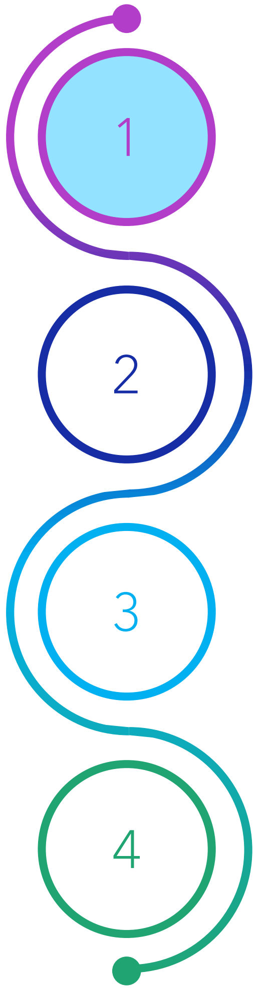
Why?

How can we fix it?

Challenge 2

Fill in the table below, showing the values of each variable in this program *after* each statement is executed.

Command	Value of x	Value of y	Value of swap
x = 1.0	1.0	Undefined	Undefined
y = 3.0	1.0	3.0	Undefined
swap = x	1.0	3.0	1.0
x = y	3.0	3.0	1.0
y = swap	3.0	1.0	1.0



Challenge 3

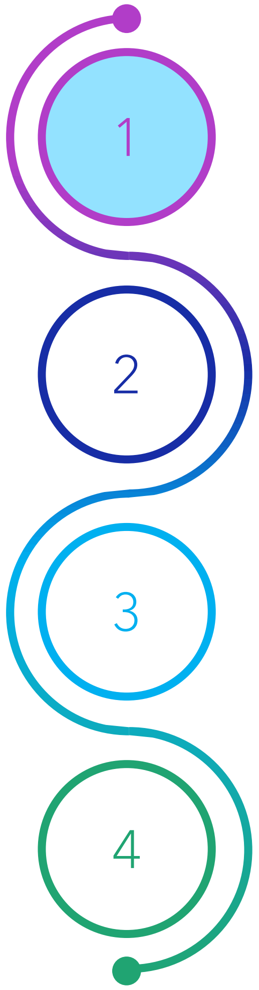
Given the following string:

```
full_name = 'Peregrin Fool of a Took Pippin'
```

What would these expressions return?

1. full_name[2:8]
2. full_name[11:] (without a value after the colon)
3. full_name[:4] (without a value before the colon)
4. full_name[:] (just a colon)
5. full_name[11:-3]
6. full_name[-5:-3]
7. What happens when you choose a stop value which is out of range? (i.e. try full_name[0:42] or full_name[:103]

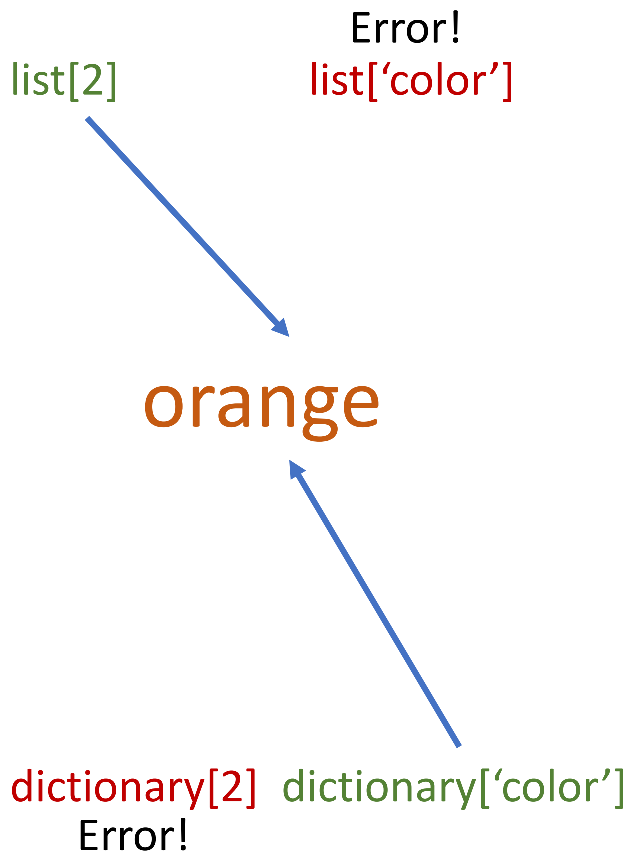
Lists vs Dictionaries



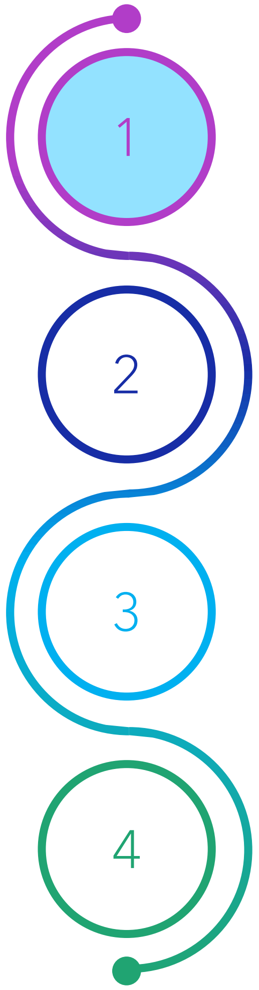
Lists: use an ordered index, starting at 0

0	1	2	3	4
apple |	banana |	orange |	potato |	otter
A	‘key’	‘color’	99	‘cute’

Dictionary: uses an unordered index, using values supplied by you, the coder



Data Types



Integers (int)

positive or negative whole numbers

42, -7, 90_210, 0, …

Floats (float)

real fractional numbers

3.14159, -87.6, 0.0, …

Strings (str)

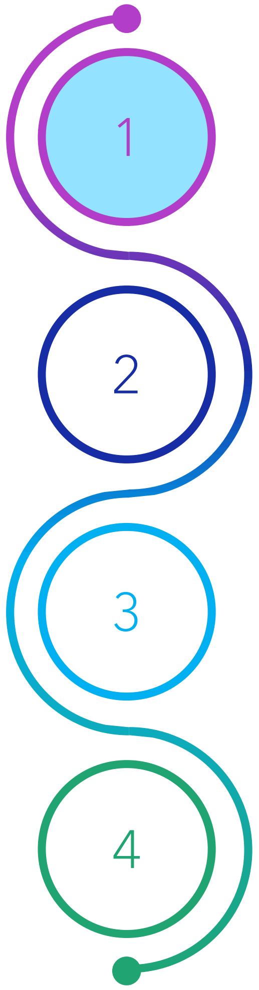
text written either in ‘single’ or “double” quotes

‘Hello, World!’, “42”, ‘zero’

Booleans (bool)

the logical values of True and False

Challenge 4

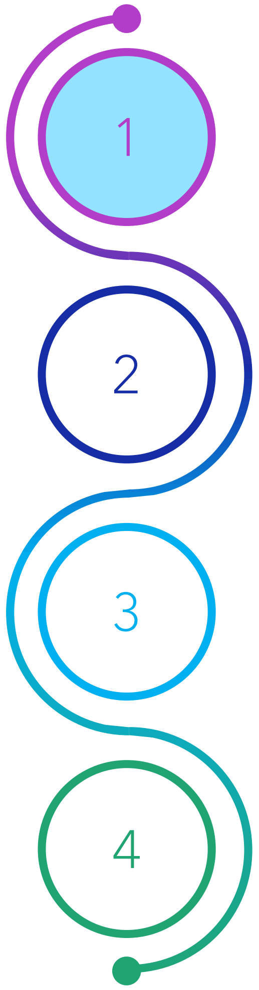


Use the help() and round() functions to print out the value of:

2.718281828459045235 36

to 0 and 2 decimal places

Challenge 5

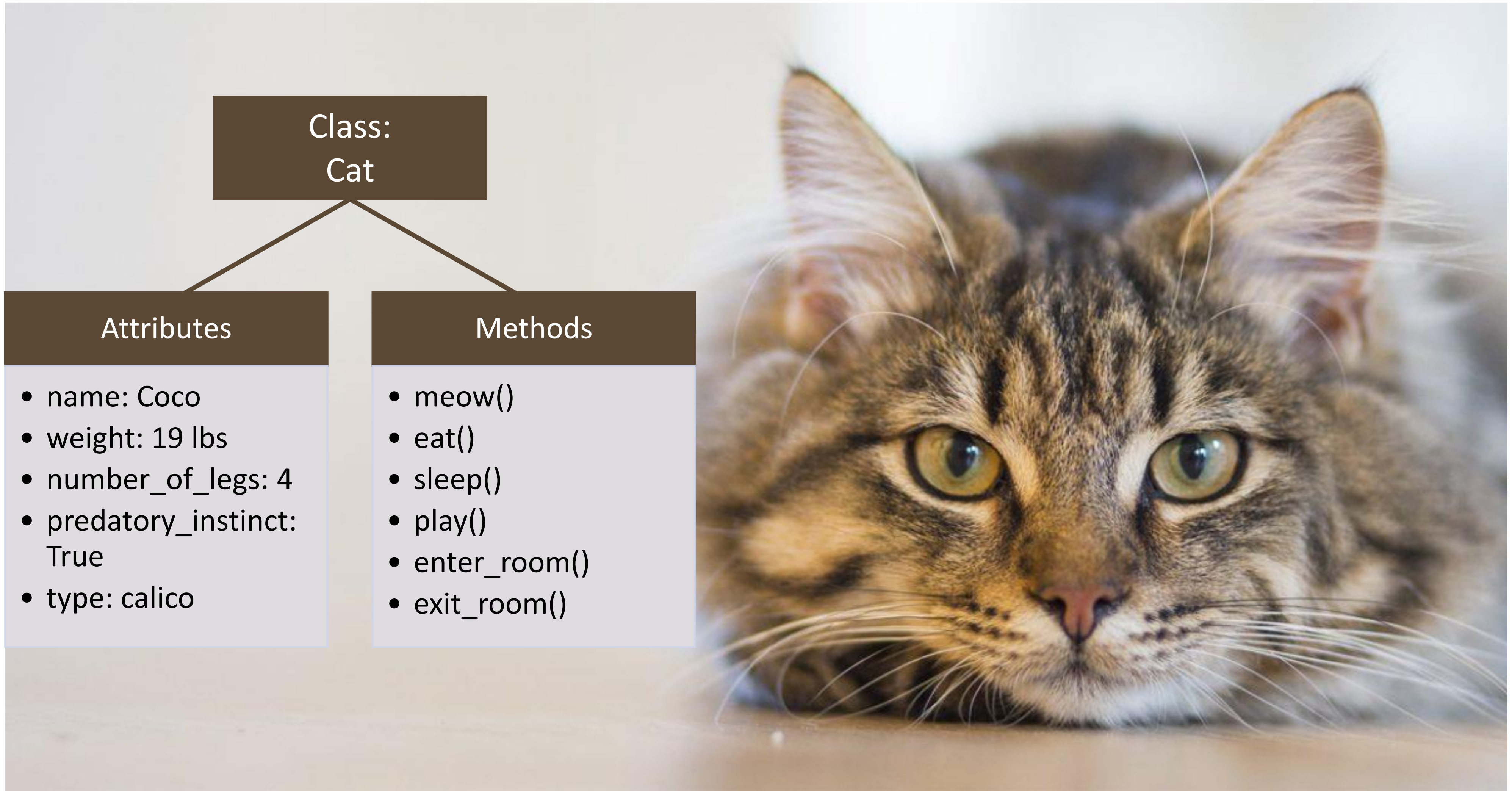


What will be the output of the following block of code?

```
num = 42
animal = 'otter’

if num==42 and animal=='mouse’:
        print('Correct, that is the animal that found the answer’)

elif num==42 and animal!='mouse’:
        print('Almost, the number is correct but not the animal.’)
        animal = 'dolphin’

elif num!=42 or animal=='mouse’:
        print('Almost, the animal is correct but not the number.’)
        num = 5

elif (1>3) or (4>3):
        print('This has nothing to do with it, we just needed an or statement’)

else:
        print('Not even close, those pesky mice need to work harder.’)
        num = 19
        animal = 'kangaroo’

print('The end result is the number', num, 'and animal', animal)
```

1
2
3
4

Objects in Python



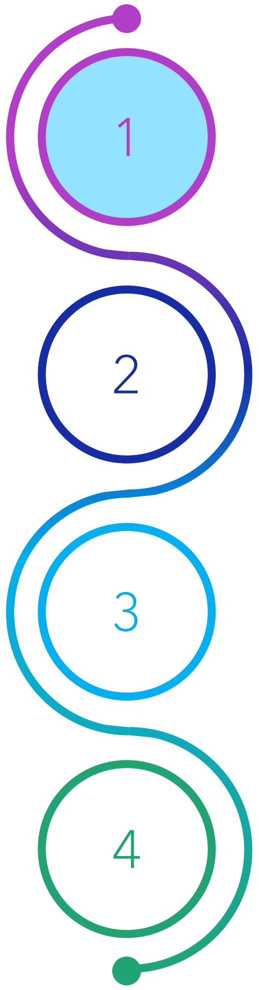
Class:
Cat

Attributes

- name: Coco
- weight: 19 lbs
- number_of_legs: 4
- predatory_instinct: True
- type: calico

Methods

- meow()
- eat()
- sleep()
- play()
- enter_room()
- exit_room()



Challenge 6

We currently have the string:

```
my_string = 'Peter Piper Picked a Peck of Pickled Peppers’
```

What happens if we try to run the following block of code:

```
print(my_string.isnumeric().upper())
```

Challenge 7

A common data string you see across government are various personal IDs. One example is the Personal Education Number (PEN), which is a nine-digit number assigned to you when you enter the K-12 School System. Oftentimes when looking at such an ID, any leading zeros that are an actual part of the PEN get stripped away, leading to unevenly sized strings of IDs across the dataset.

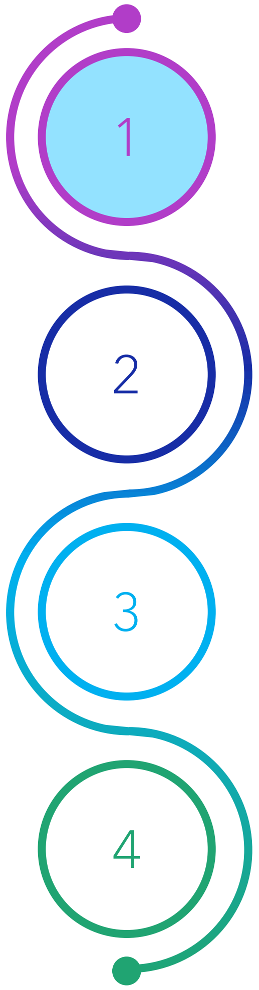
Write a piece of code for PEN IDs that does the following:

- Checks to make sure that the ID is entirely numeric. If it is not, print out a warning that this is an invalid PEN.
- If the PEN is numeric, make sure that it is less than or equal to 9 digits long. If it is longer, print out a warning that this is an invalid PEN.
- If the PEN is too short, make sure to pad it with the appropriate number of 0’s to the left. Print out the correct PEN.

Try your code out with the following PENs:

```
pen_1 = '12345678x’
pen_2 = '123456789’
pen_3 = '1234567890’
pen_4 = 123456789
pen_5 = '123456'
```

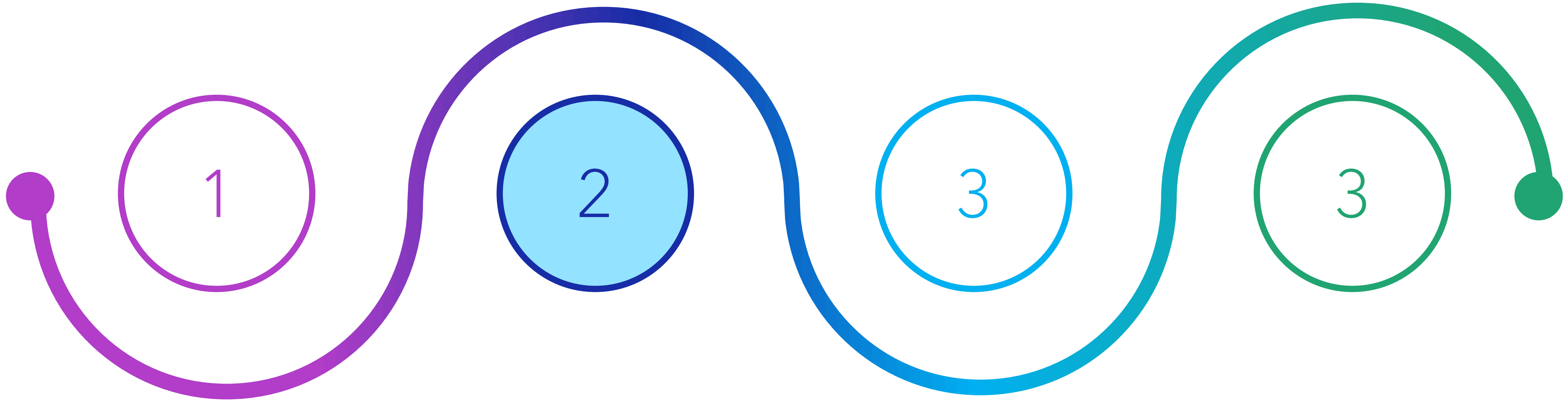
Common Package Aliases:



Package	Alias
pandas	pd
matplotlib.pyplot	plt
seaborn	sns
numpy	np

import PACKAGE **as** ALIAS

GETTING DATA WITH PANDAS



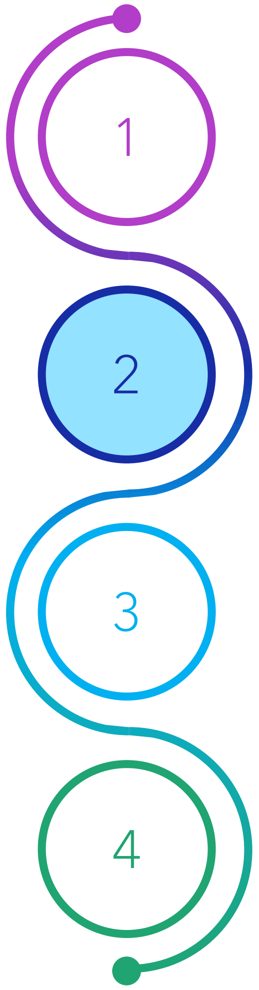
Introduction to Python

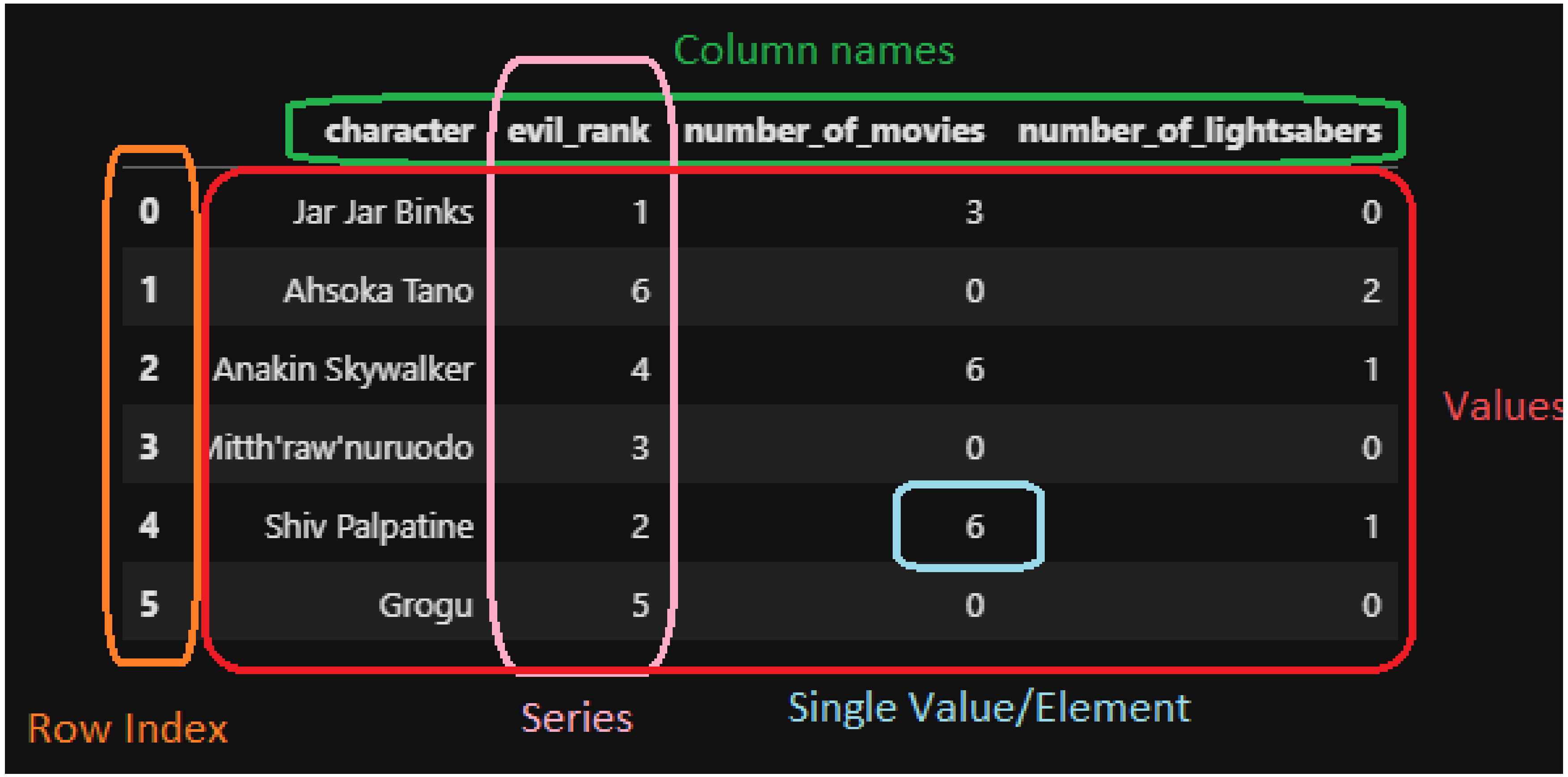
Get And Clean Data

Understand and Analyze Data

Advanced Topics

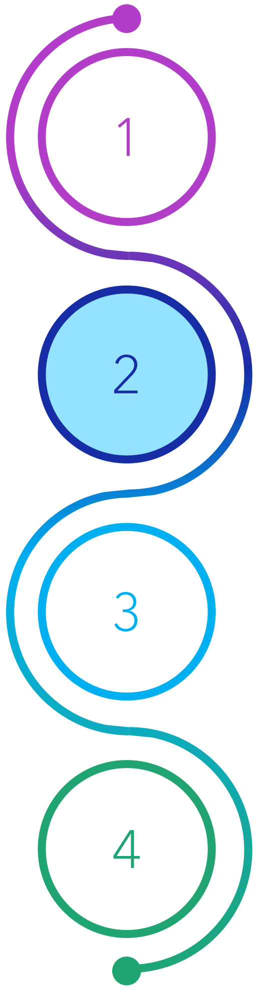
Pandas Dataframes





	character	evil_rank	number_of_movies	number_of_lightsabers
0	Jar Jar Binks	1	3	0
1	Ahsoka Tano	6	0	2
2	Anakin Skywalker	4	6	1
3	Mitth'raw'nuruodo	3	0	0
4	Shiv Palpatine	2	6	1
5	Grogu	5	0	0

Challenge 1



Bring the gapfinder dataset back into pandas.

This time:

- Use the country as the index
- Only include the year, continent and population columns

- **Hint: ?pd.read_csv**
- **Hint 2: What do the arguments index_col and usecols do?**

Challenge 2

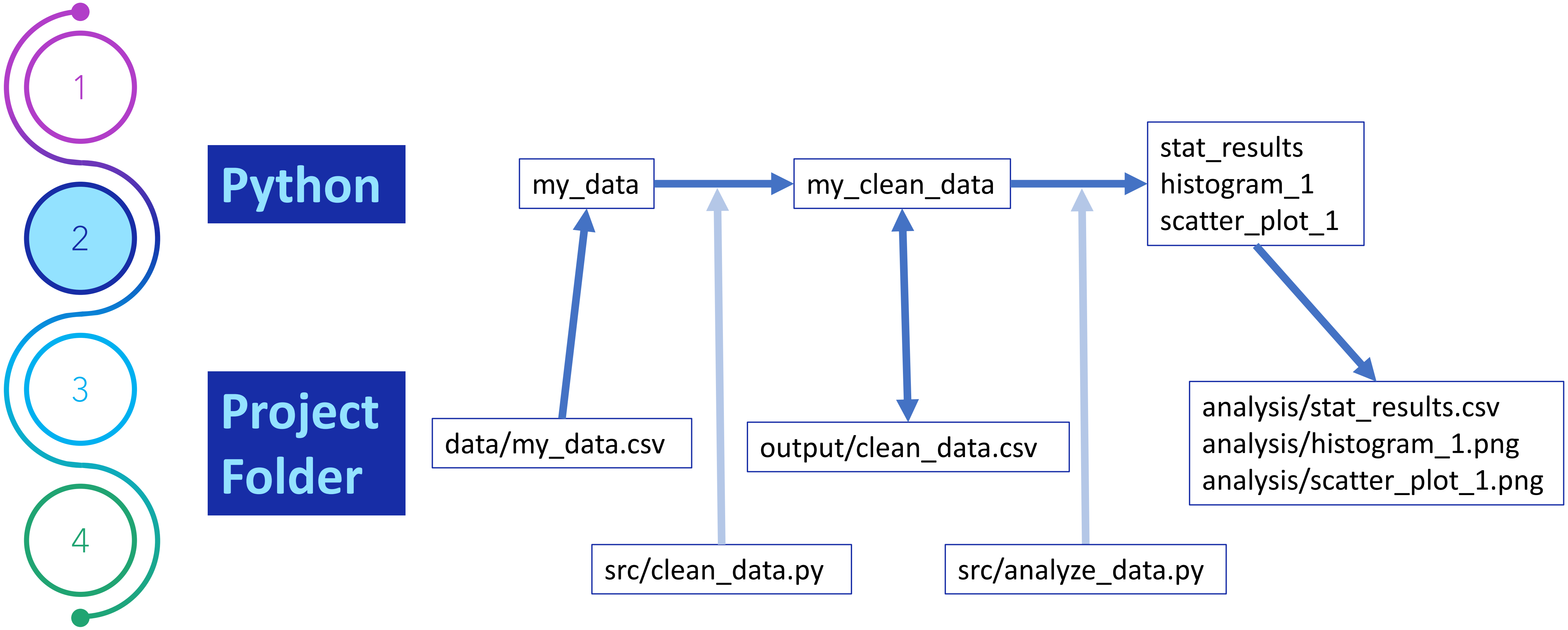
We have the gapfinder dataset loaded into an object called **df**

What will happen if we run the following code:

```
df.sample(42).describe()
```

1. Do we expect the results to be the same as **df.describe()**? Why or why not?
2. Run the code again. Are the results the same or different than before? Can you explain?

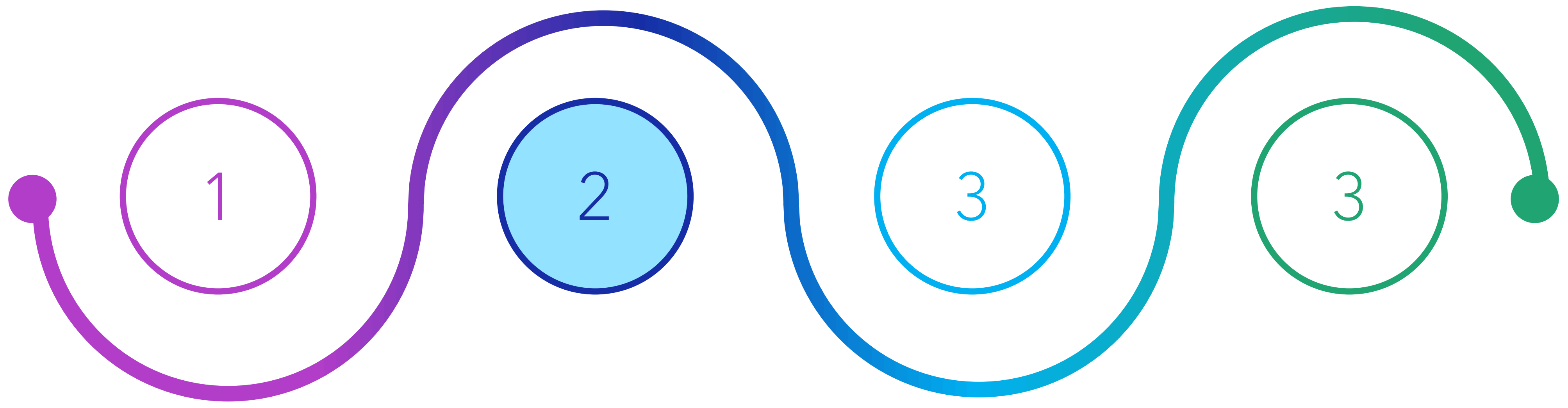
Beginning of a Reproducible Pipeline



Challenge 3

1. Save a copy of all the summary statistics for the gapfinder dataset.
2. Only include the statistics for the **pop** and **lifeExp** columns.
3. What happens when we include or exclude the index?
4. Hint: columns, index

CLEANING DATA



Introduction to Python

Get And Clean Data

Understand and Analyze Data

Advanced Topics

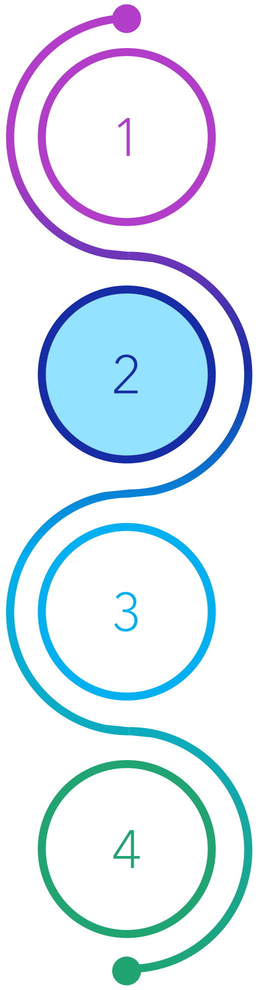
Challenge 1

You are given a dataset for several days time frame in Feb/March 2021 of the average daily temperature in celcius for three Canadian cities. You would like to ultimately create a plot of the temperatures but don’t want to have gaps in the data. Run the code below to get the data and see what it looks like.

```
import pandas as pd


temps = "https://raw.githubusercontent.com/bcgov/" \
        "ds-intro-to-python/main/data/citytemps.csv"
city_temps = pd.read_csv(temps)
city_temps.head(10)
```

Unfortunately you notice that several of the days have data missing. Write code that leverages the interpolate() function to fill in the missing data. Run the code. Is this an appropriate solution for the problem? What are the pros and cons to this approach.

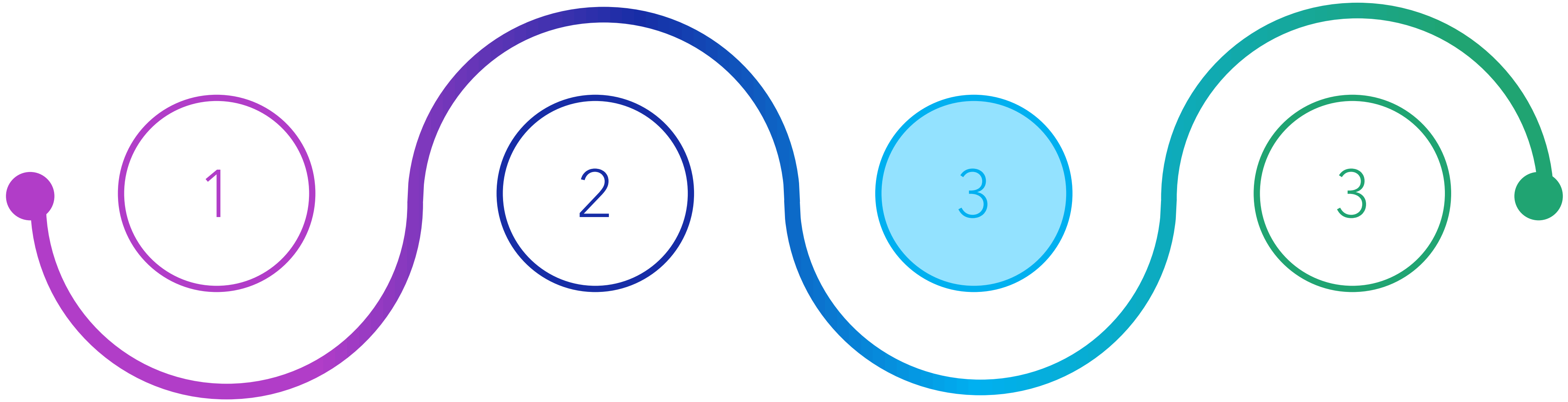
Challenge 2



You would like to clean up the scale used in the question about whether one’s mental illness affects one’s work.

- Find the column (pandas series) that needs to be transformed and transform its values using `replace()` with numerical values that make sense.
- Display the modified dataframe showing the replaced values.

EXPLORING DATA STRUCTURES



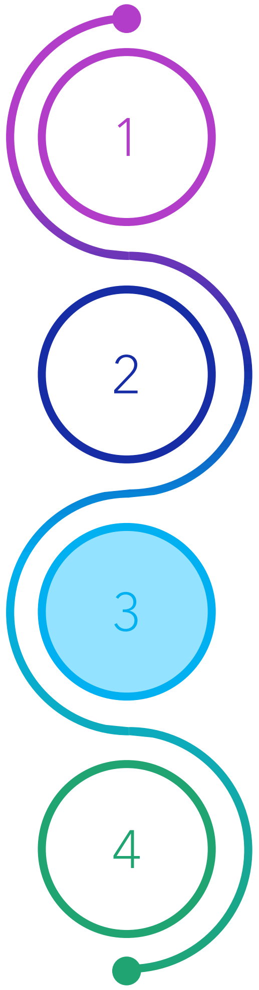
Introduction to Python

Get And Clean Data

Understand and Analyze Data

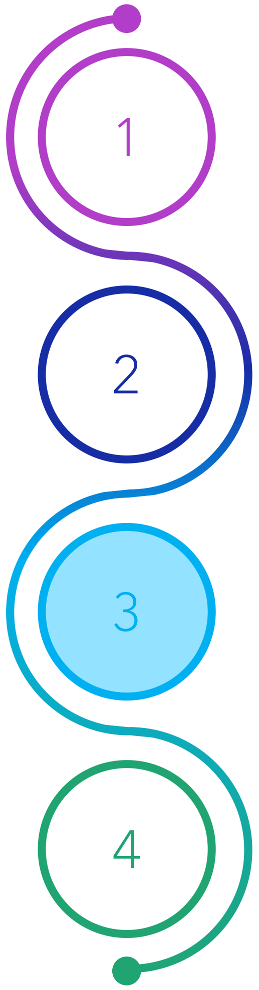
Advanced Topics

Challenge 1



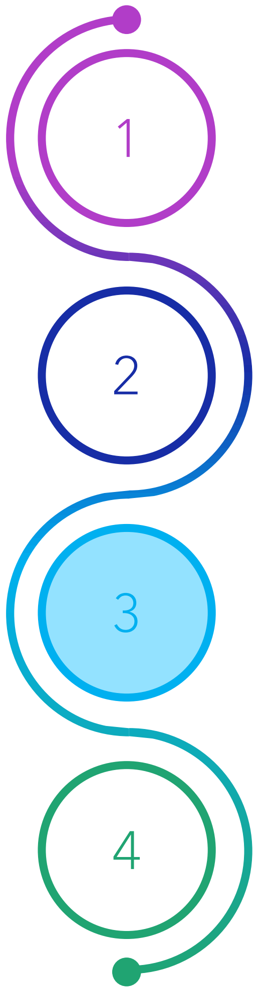
Let’s say you want to narrow down the dataset to include only the country and the year. How would you do it? Don’t forget to run your code so it also shows a view of the result so you can confirm the code worked as you wanted it to.

Challenge 2



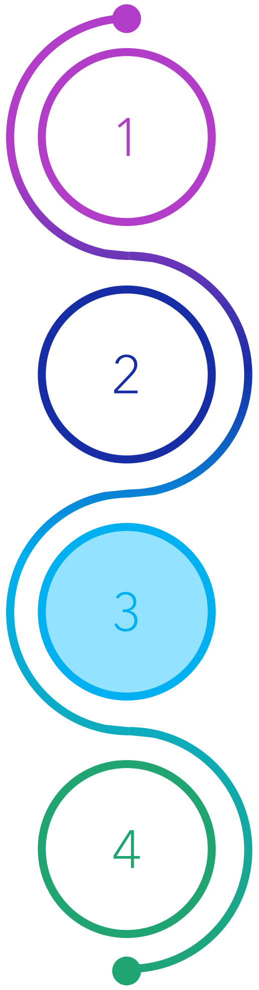
Your director has come to you and asked if you know what the life expectancy has been in Canada since 1992. How would you use pandas code to get the data you need? And after having run the code, what’s the answer?

Challenge 3



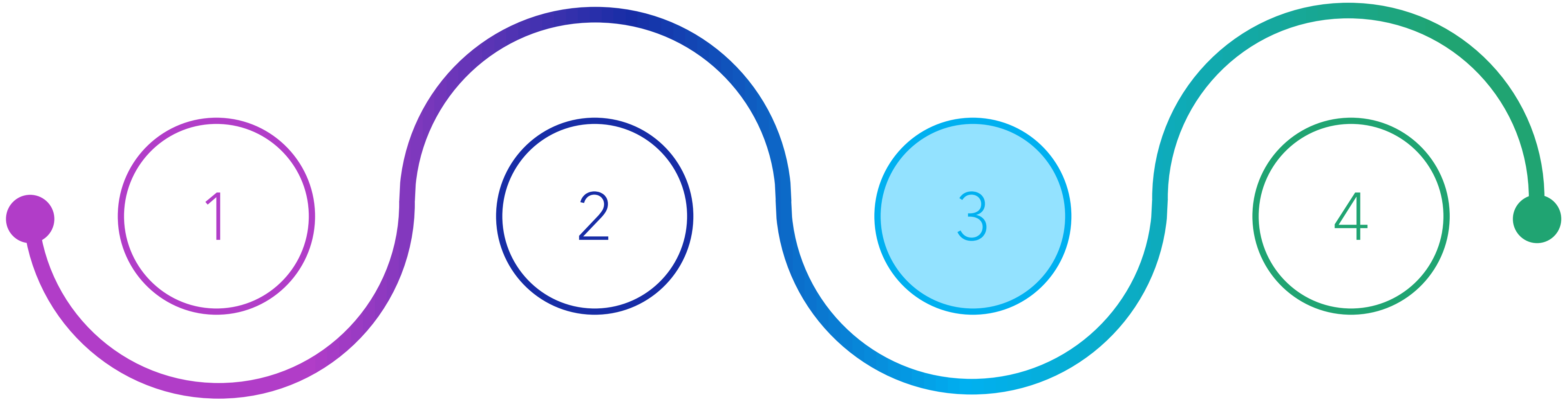
Your director has come back to you and wondered about whether the life expectancy of people changed during the 1970s in Cambodia. Use what you know about selecting rows and sorting data to get the data you need to answer the question, sorted from most recent at the top.

Challenge 4



You would like to summarize population as well as life expectancy by year, grouped by continent. Pick some aggregations that would make sense to look at for this task. Don’t worry about re-setting the index or about creating new labels for the result.

GRAPHICAL DEPICTIONS OF DATA



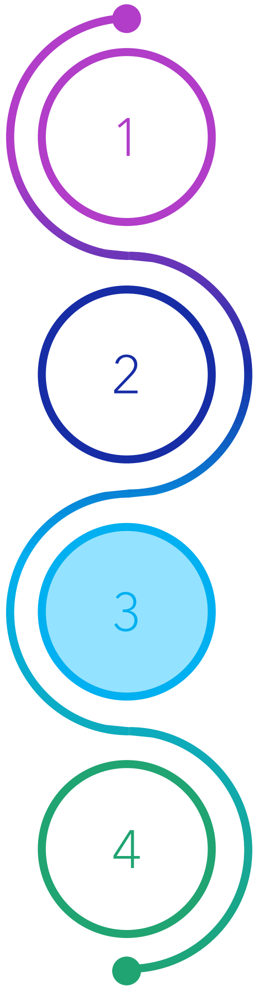
Introduction to Python

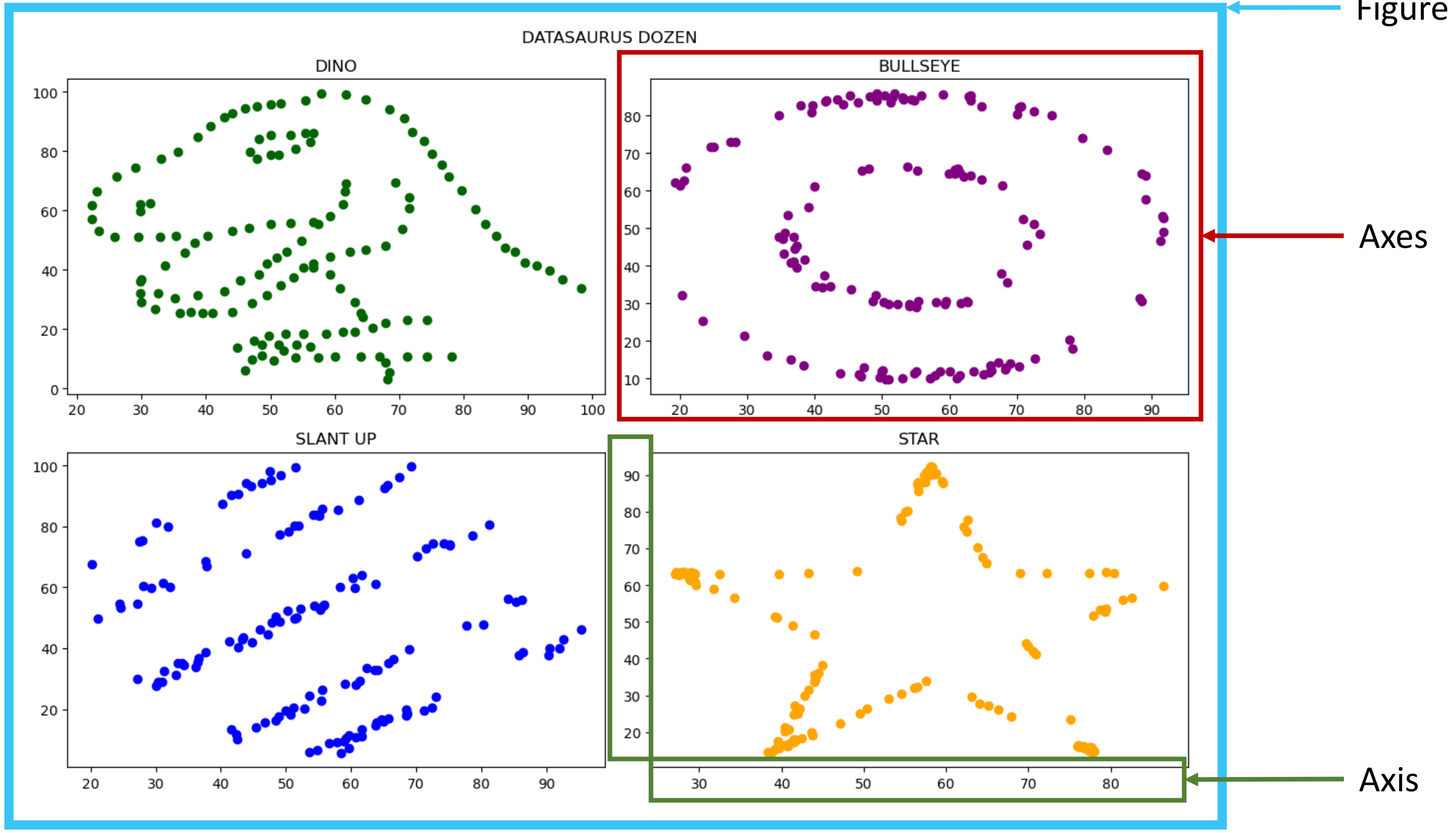
Get And Clean Data

Understand and Analyze Data

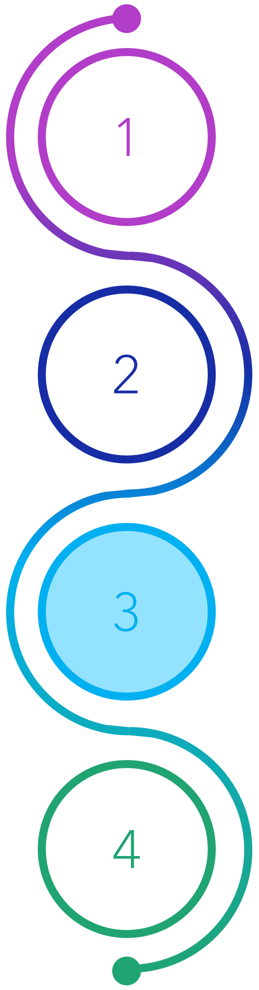
Advanced Topics

Matplotlib Figures

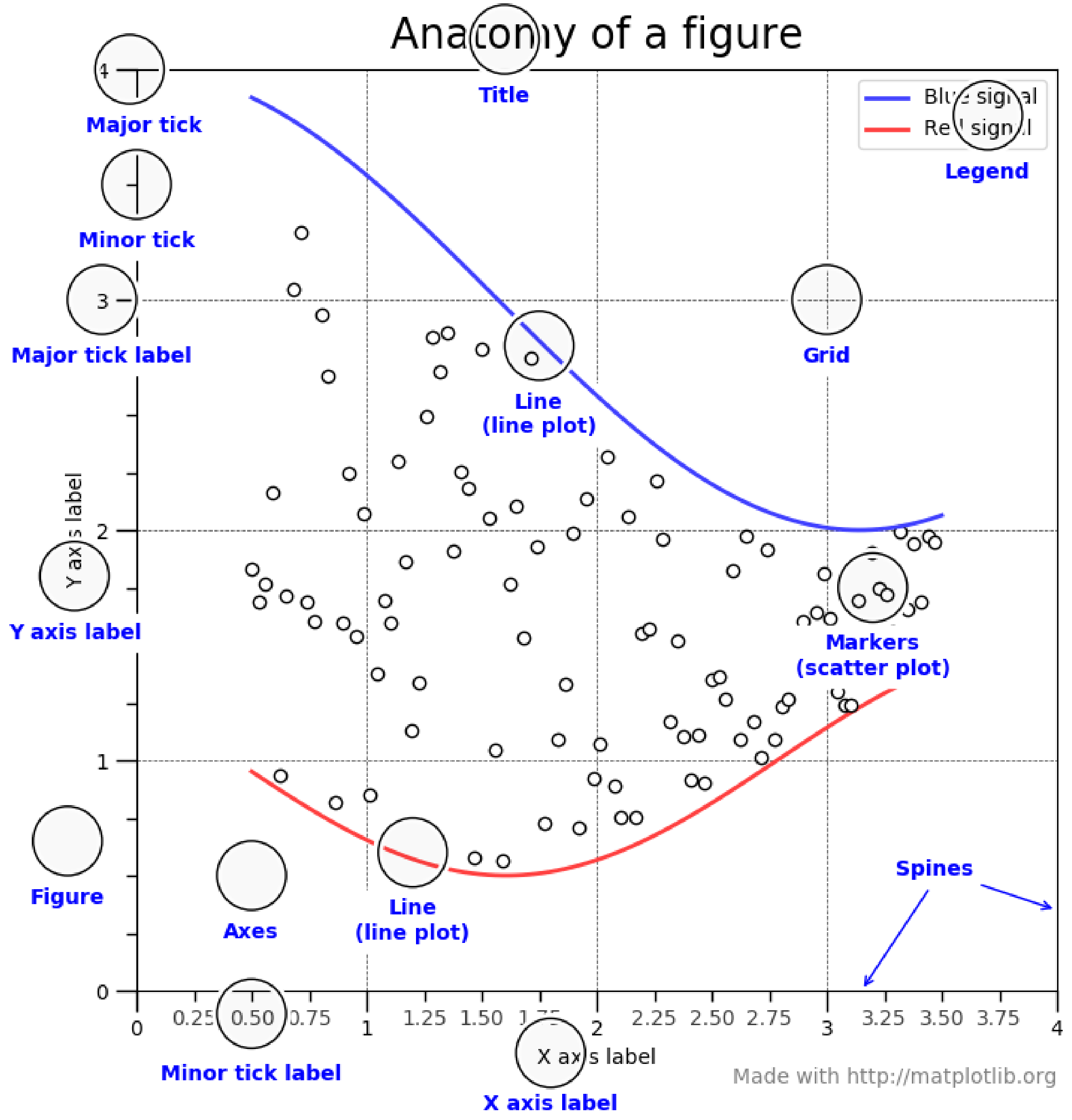




Matplotlib Figures



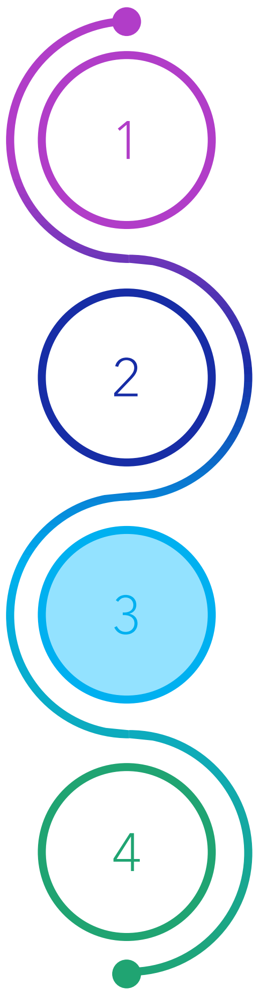
1. Entire figure and axes is initialized when we create the first plot. These are assigned to the **current figure/axes.**
2. Add secondary/tertiary plots to the same axes line by line.
   - Individual plot characteristics are defined on each of these lines (label, plot type, color used, etc.)
3. Label each **axes** as required
   - Titles, legend names, legend values, etc.
4. Save the plot
5. Display the plot



Challenge 1

1. Using what we have learned so far about pandas filtering and matplotlib functionality, produce a plot of bill lengths vs. depths for the penguins dataset.
2. Colour the three penguin species (Adelie, Chinstrap, Gentoo) different colours.
3. Include a legend that identifies which species is which colour!
4. **HINT:** use what we learned with Stu to filter the dataframe to a single species (create three new dataframes…)

Seaborn Plots



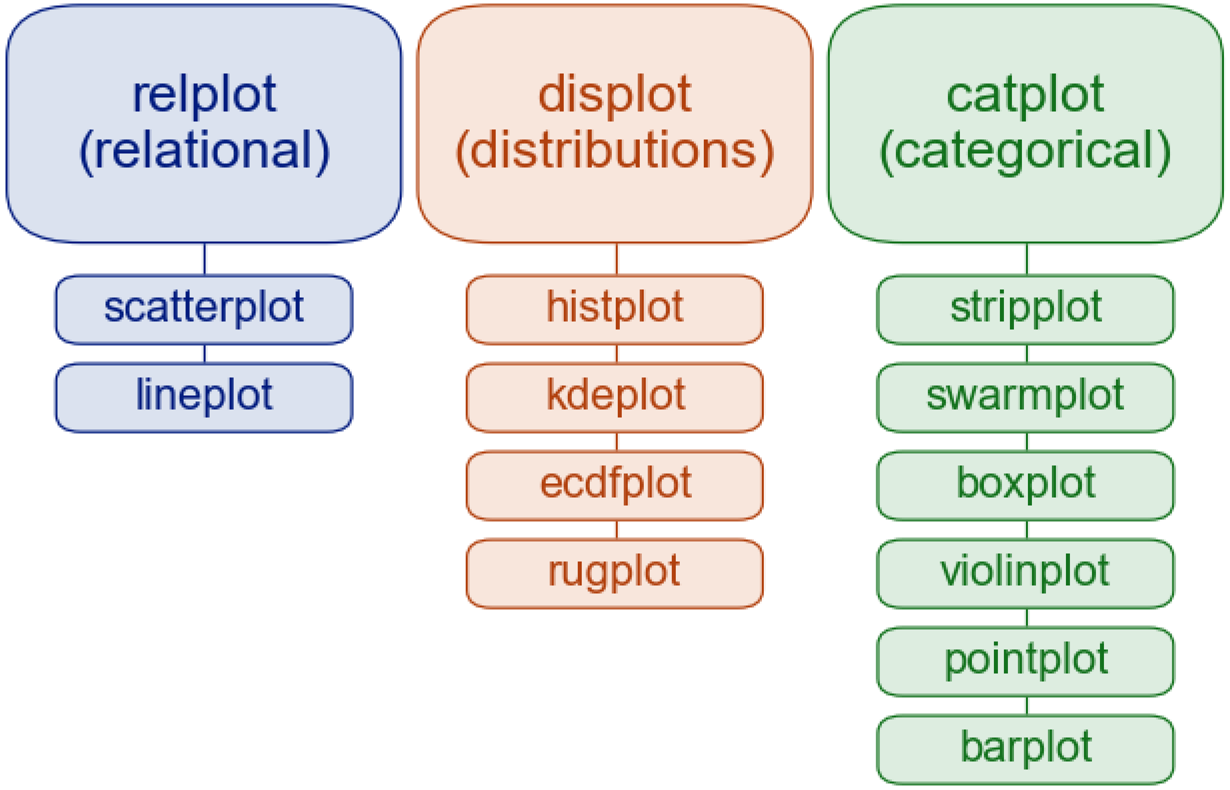


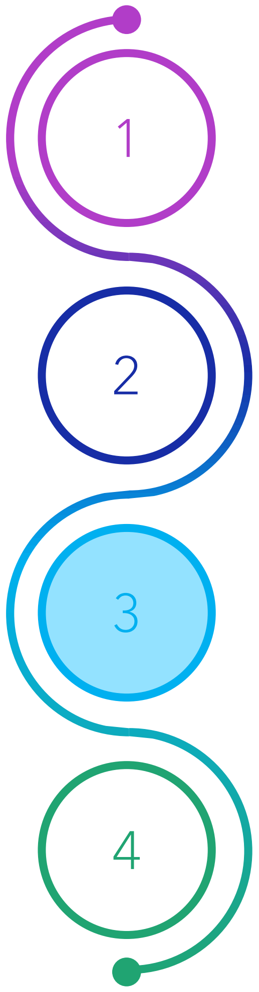
Figure Level:

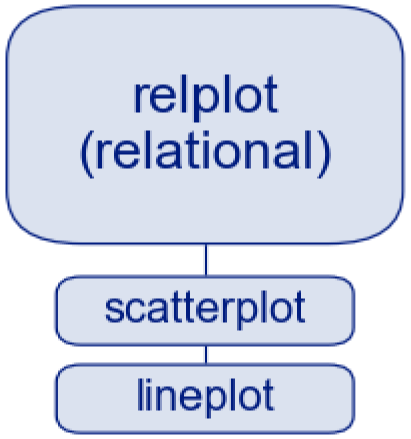
- Simple, single command for multiple different plot types
- Returns a 'fully formed' final figure
- Cannot easily access the underlying matplotlib components

Axes Level:

- Individualized for specific plot types
- Returns a single axes that can be added to other matplotlib figures
- Easier to customize, but more 'fidgety'

Relational Plots





(For regressions): lmplot

Usage

- Let us look for patterns in a dataset between two or more numerical features
- Can be used to identify trends over time
- Can be used to display uncertainty in a model

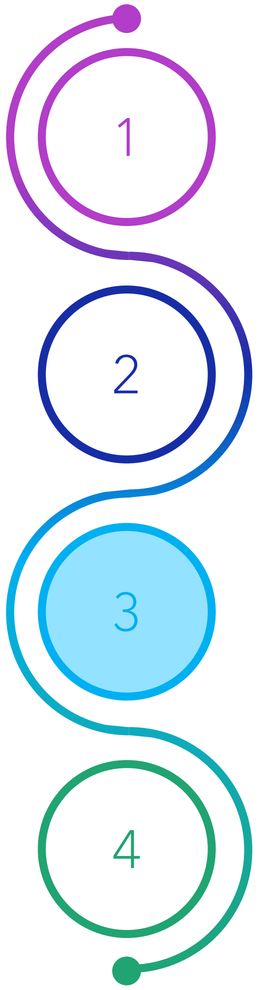
Types

- Scatter plots
- Line plots

Conditional Options

- style, hue, size
- row, col

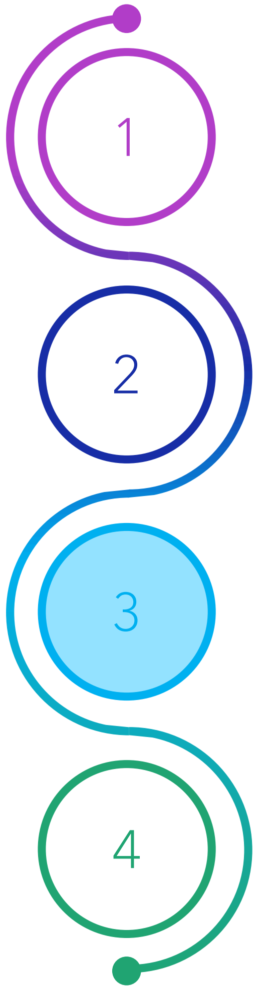
Challenge 2



Using the Seaborn **load_dataset** method, import a dataset called ‘dowjones’.

1. What does this dataset include/represent?
2. On what date does the data represented here reach its highest price?
3. Create a line plot to depict the data. Does the plot agree with the answer you came up with in part 2?

Challenge 3

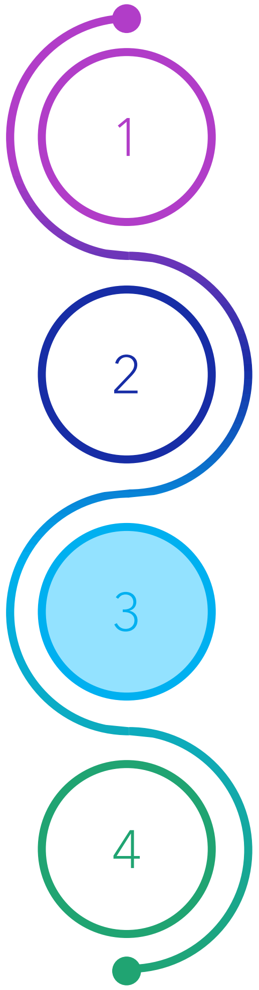


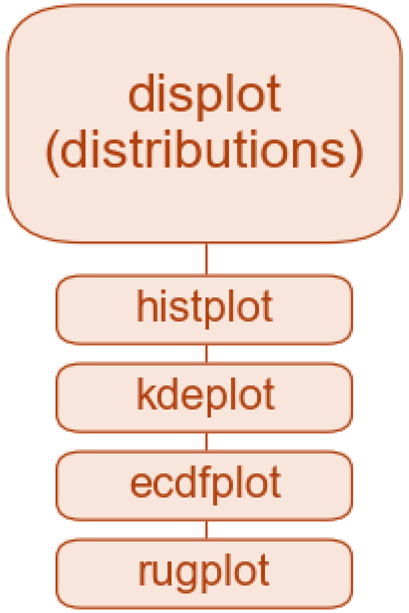
Choose any two numerical features in the penguins dataset.

Produce a scatter plot that highlights the differences between species.

Separate the results into two plots – one for male and one for female penguins.

Distribution Plots





Usage

- Analyze the data to see how features are distributed
- Identify outliers
- Compare subsets

Types

- Histograms
- KDE (Kernel Density Estimation)
- ECDF (Empirical Cumulative Distribution Function)

Conditional Options

- hue
- row/col

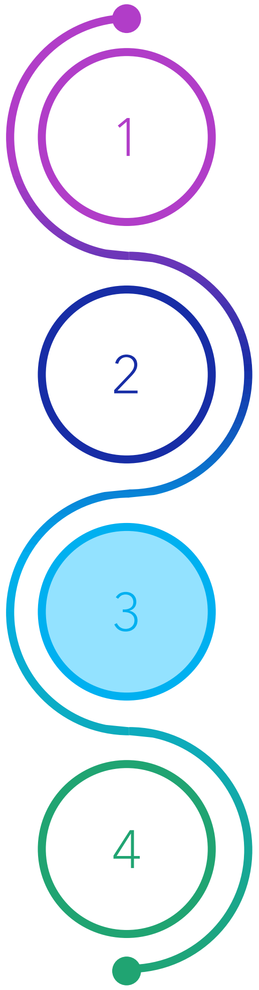
Challenge 4

Create a KDE plot of body mass from the penguins dataset. (body_mass_g)

Does it appear to be bi-modal (2 peaks)?

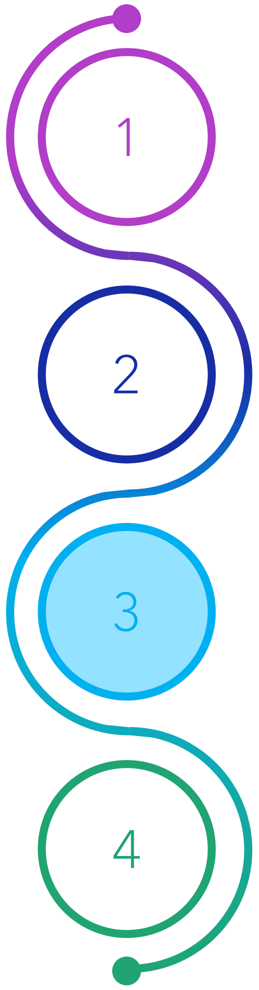
If so, create another plot (or more) to identify what may be the cause. Are male penguins heavier? Is it a specific species? Does the island matter?

Challenge 5

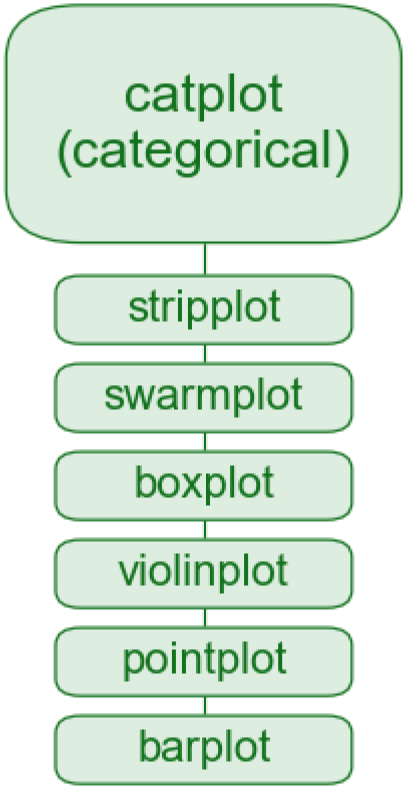


Which penguin species was most frequently included in this dataset?

Display your result visually!



Categorical Plots



Usage

- Let us look for patterns in a dataset using at least one categorical feature
- Default is to display **aggregations** of the data for most types!

Types

- Categorical scatterplots: swarm, strip
- Categorical distribution plots: box, violin, boxen
- Categorical estimate plots: point, bar, count

Conditional Options

- hue
- row/col