Dive into Python

INTRODUCTION TO DATA SCIENCE IN PYTHON

Hillary Green-Lerman
Lead Data Scientist, Looker

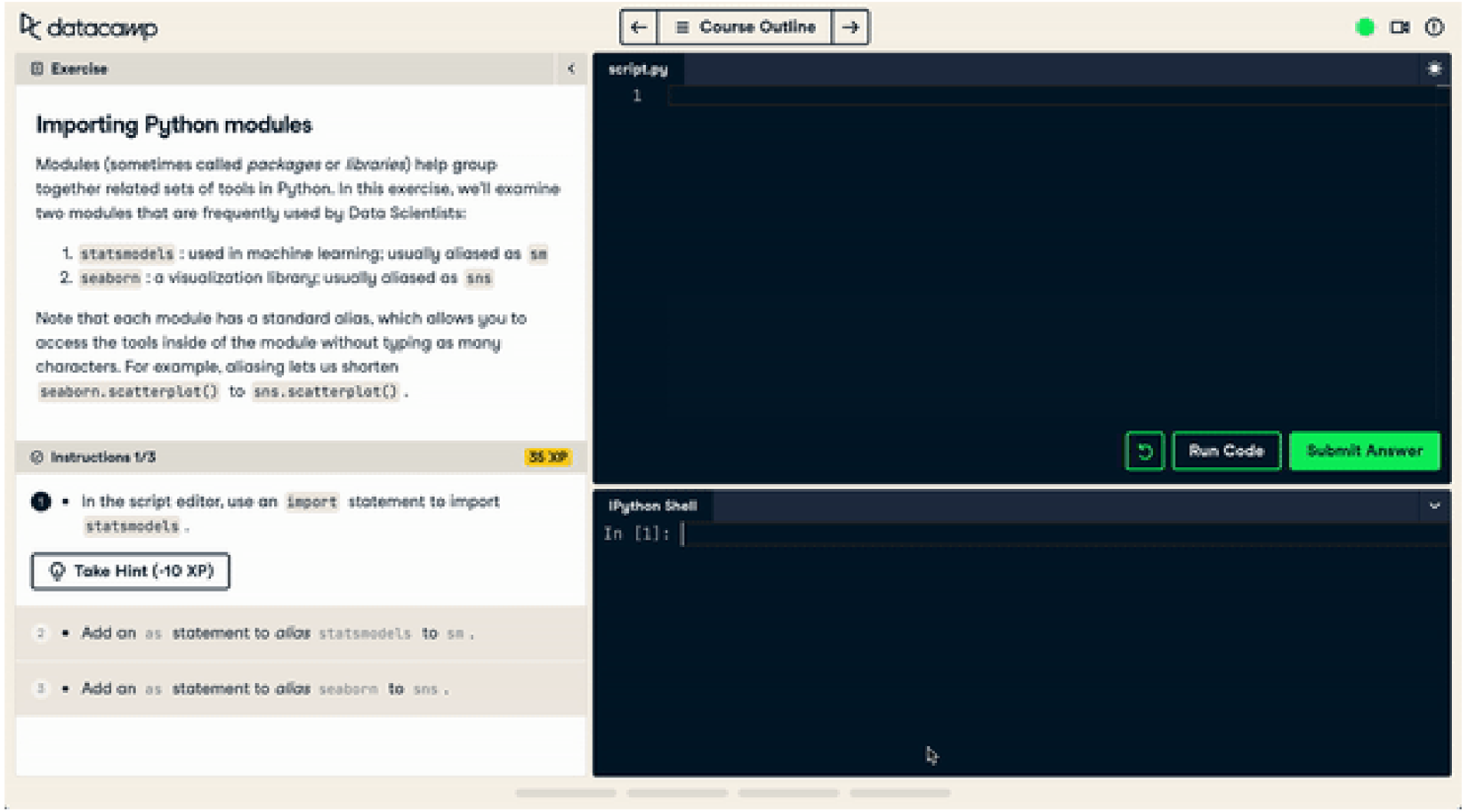
datacamp

What you'll learn

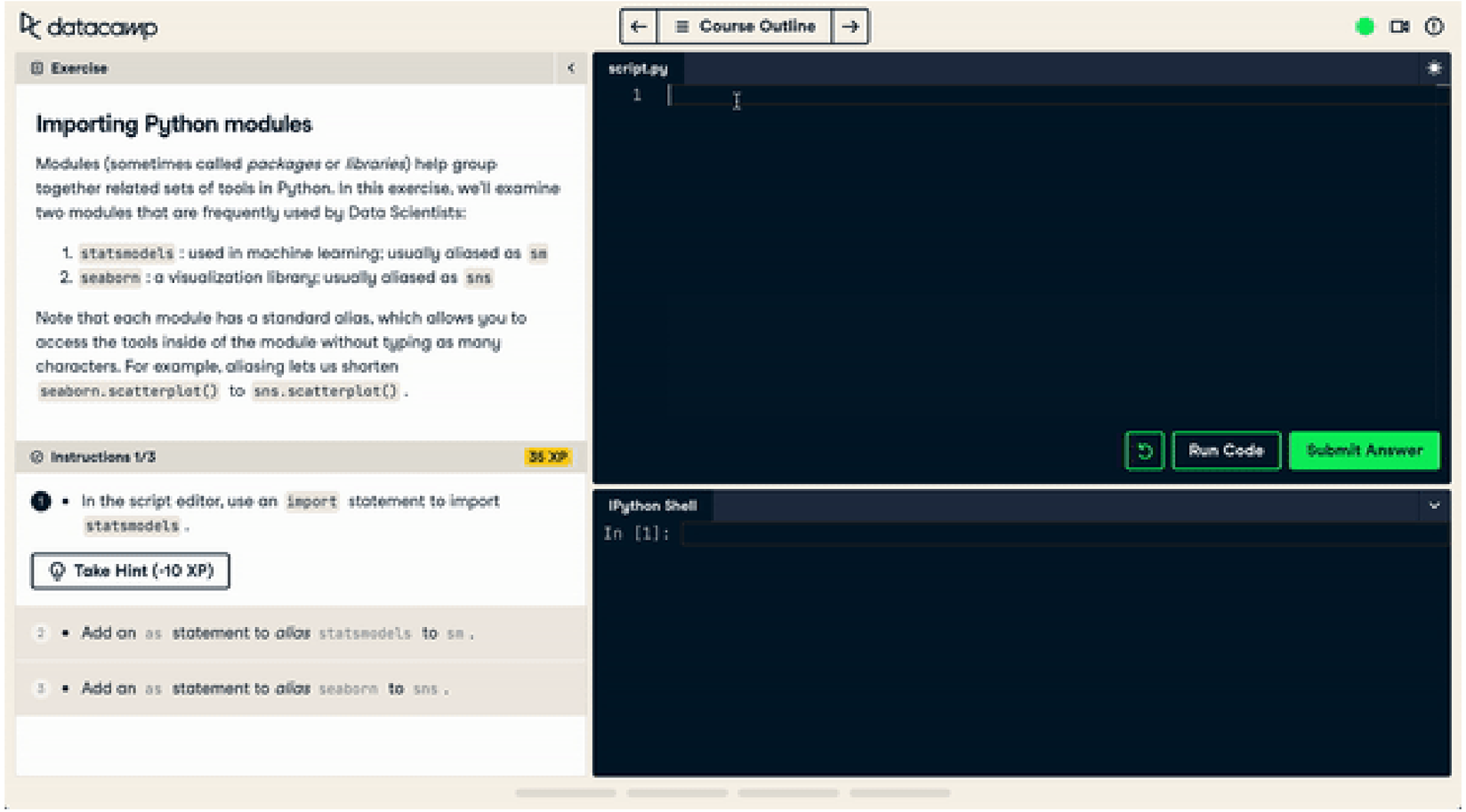
- How to write and execute Python code with DataCamp
- How to load data from a spreadsheet
- How to turn data into beautiful plots

Solving a mystery with data

Using the IPython shell



Using the script editor



What is a module?

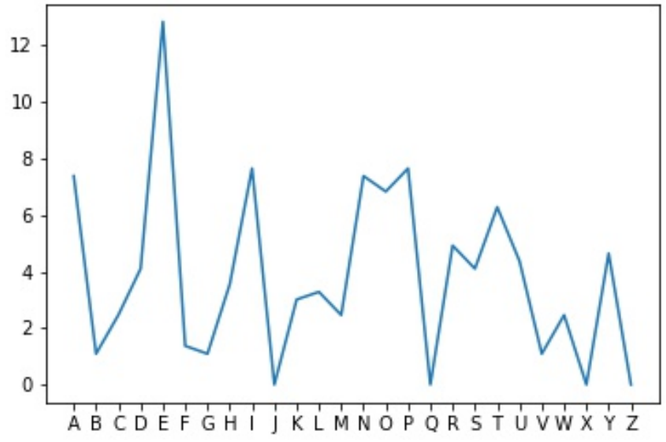
- Groups related tools together
- Makes it easy to know where to look for a particular tool
- Common examples:
  - `matplotlib`
  - `pandas`
  - `scikit-learn`
  - `scipy`
  - `nltk`

Importing pandas and matplotlib

```
import pandas as pd
from matplotlib import pyplot as plt

# Pandas loads our data
df = pd.read_csv('ransom.csv')

# Matplotlib plots and displays
plt.plot(df.letters, df.frequency)
plt.show()
```



Importing a module

- Importing a Module

```
import pandas
```

- Importing a module with an alias

```
import pandas as pd
```

Let's practice!

INTRODUCTION TO DATA SCIENCE IN PYTHON

Creating variables

INTRODUCTION TO DATA SCIENCE IN PYTHON

Hillary Green-Lerman
Lead Data Scientist, Looker

Filing a missing puppy report


```
name = "Bayes"
height = 24
weight = 75.5
```

Rules for variable names

- Must start with a letter (usually lowercase)
- After first letter, can use letters/numbers/underscores
- No spaces or special characters
- Case sensitive (`my_var` is different from `MY_VAR`)

```
# Valid Variables
bayes_weight
b
bayes42
```

```
# Invalid Variables
bayes-height
bayes!
42bayes
```

Error messages

```
bayes-height = 3
```

```
File "<stdin>", line 1
    bayes-height = 3
                 ^
SyntaxError: can't assign to operator
```

Floats and strings

- *float*: represents an integer or decimal number

```
height = 24
weight = 75.5
```

- *string*: represents text; can contain letters, numbers, spaces, and special characters

```
name = 'Bayes'
breed = "Golden Retriever"
```

Common string mistakes

- Without quotes, you'll get a name error.

```
owner = DataCamp
```

```
File "<stdin>", line 1, in <module>
    owner = DataCamp
NameError: name 'DataCamp' is not defined
```

- If you use different quotation marks, you'll get a syntax error.

```
fur_color = "blonde'
```

```
File "<stdin>", line 1
    fur_color = "blonde'
                        ^
SyntaxError: EOL while scanning string literal
```

Displaying variables

```
name = "Bayes"
height = 24
weight = 75

print(height)
```

```
24
```

Let's practice!

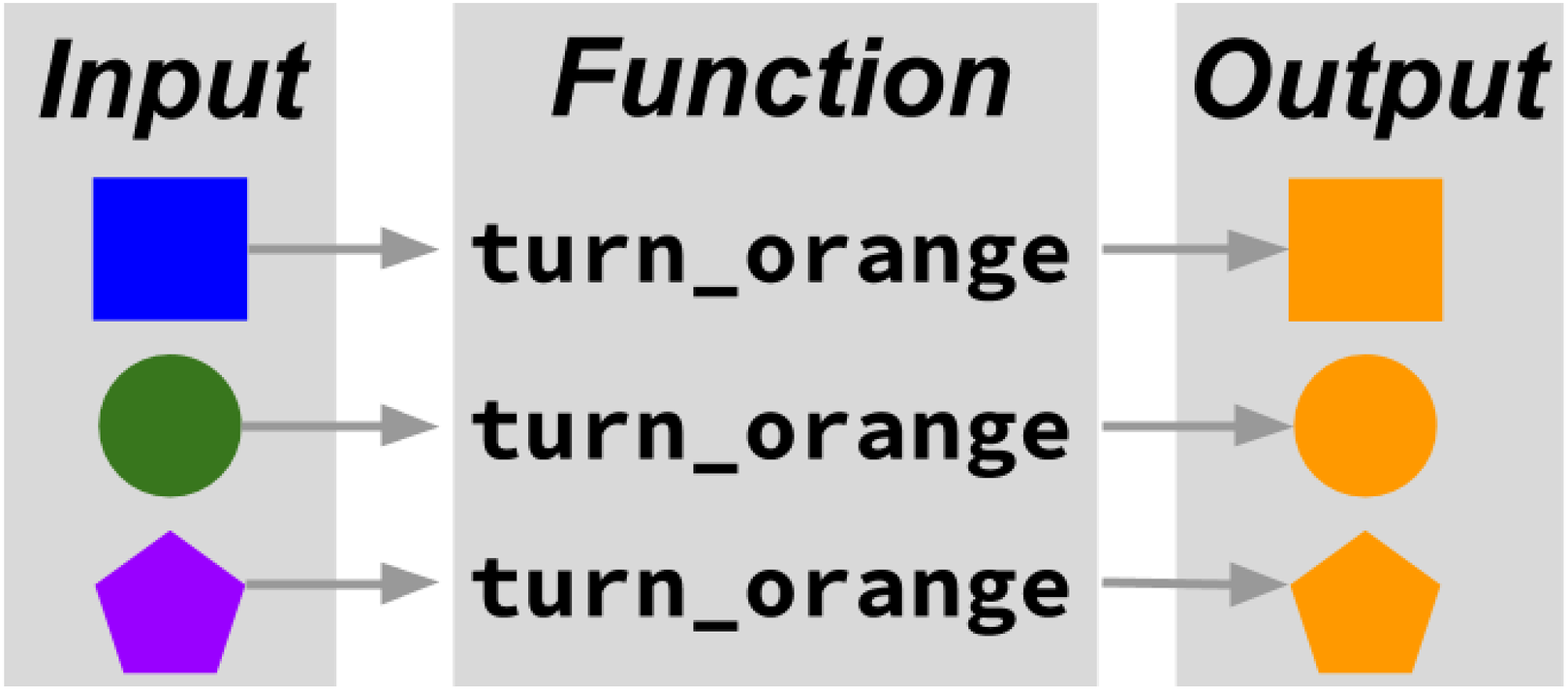
INTRODUCTION TO DATA SCIENCE IN PYTHON

What is a function?

INTRODUCTION TO DATA SCIENCE IN PYTHON

Hillary Green-Lerman
Lead Data Scientist, Looker

A function is an action



Functions in code

```
import pandas as pd
from matplotlib import pyplot as plt

df = pd.read_csv('letter_frequency.csv')

plt.plot(df.letter_index, df.frequency, label='Ransom')
plt.show()
```

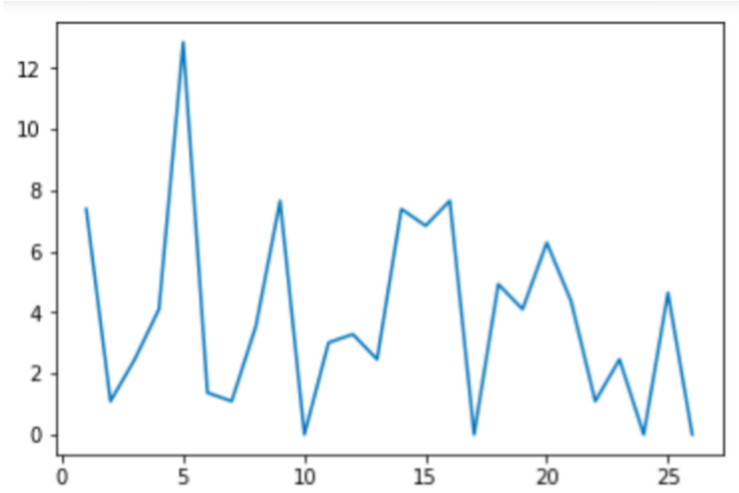
Functions perform actions:

- `pd.read_csv()` turns a csv file into a table in Python
- `plt.plot()` turns data into a line plot
- `plt.show()` displays plot in a new window

```
plt.plot(df.letter_index, df.frequency, label='Ransom')
```

Function — Positional Arguments — Keyword Argument

letter_index	letter	frequency
1	A	7.38
2	B	1.09
3	C	2.46
4	D	4.10
...	...	...



6. Anatomy of a function: positional arguments
Positional arguments are one type of input that a function can have. Positional arguments must come in a specific order. In this case, the first argument is the x-value of each point, and the second argument is the y-value of each point. Each argument is separated by a comma. If you forget the comma, you will get a syntax error. It's good practice to put a space after the comma, but your code will run even if you forget that space. Keyword arguments come after

Anatomy of a function: function name

```
plt.plot(df.letter_index, df.frequency, label='Ransom')
```

Function

Function Name:

- Starts with the module that the function "lives" in (`plt`)
- Followed by the name of the function (`plot`)
- Function name is always followed by parentheses `()`

Anatomy of a function: positional arguments

```
plt.plot(df.letter_index, df.frequency, label='Ransom')
```

Positional Arguments

X Y

Positional Arguments:

- These are *inputs* to a function; they tell the function how to do its job
- Order matters!

Anatomy of a function: keyword arguments

```
plt.plot(df.letter_index, df.frequency, label='Ransom')
```

Keyword Argument

Keyword Arguments:

- Must come *after* positional arguments
- Start with the name of the argument (`label`), then an equals sign (`=`)
- Followed by the argument (`Ransom`)

Common function errors

- Missing commas between arguments

```
plt.plot(df.letter_index df.frequency label='Ransom')
```

- Missing closed parenthesis

```
plt.plot(df.letter_index, df.frequency, label='Ransom'
```

Let's practice!

INTRODUCTION TO DATA SCIENCE IN PYTHON