

Dive into Python

INTRODUCTION TO DATA SCIENCE IN PYTHON



Hillary Green-Lerman
Lead Data Scientist, Looker

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

The screenshot shows a DataCamp exercise interface for "Importing Python modules".

Script Editor: A dark-themed code editor window titled "script.py" containing the number "1". Below the editor are three buttons: a blue "Run Code" button, a green "Submit Answer" button, and a yellow "Check" button.

Instructions: The instructions are divided into three steps:

1. In the script editor, use an `import` statement to import `statsmodels`.
2. Add an `as` statement to alias `statsmodels` to `sm`.
3. Add an `as` statement to alias `seaborn` to `sns`.

A "Take Hint (-10 XP)" button is available for step 1.

IPython Shell: A dark-themed terminal window titled "IPython Shell" showing the prompt "In [1]: |".

Using the script editor

The screenshot shows the DataCamp Python script editor interface. On the left, there's a sidebar with the DataCamp logo, navigation buttons for 'Course Outline' and 'Exercises', and a progress bar indicating 'Instructions 1/3' with '30 XP' available. The main area has a dark theme with a title 'script.py' and a code editor containing the line '1 []'. Below the code editor are three buttons: 'Run Code', 'Submit Answer', and a 'Take Hint (-10 XP)' button. To the right of the code editor is an 'IPython Shell' window titled 'In [1]:' which is currently empty. The overall layout is clean and professional, designed for interactive learning.

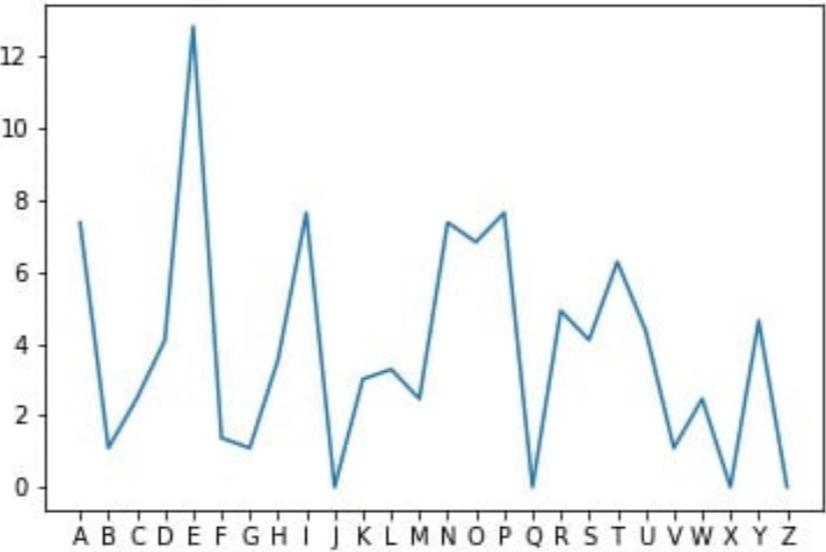
What is a module?

- 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)
```

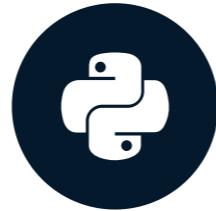
24

Let's practice!

INTRODUCTION TO DATA SCIENCE IN PYTHON

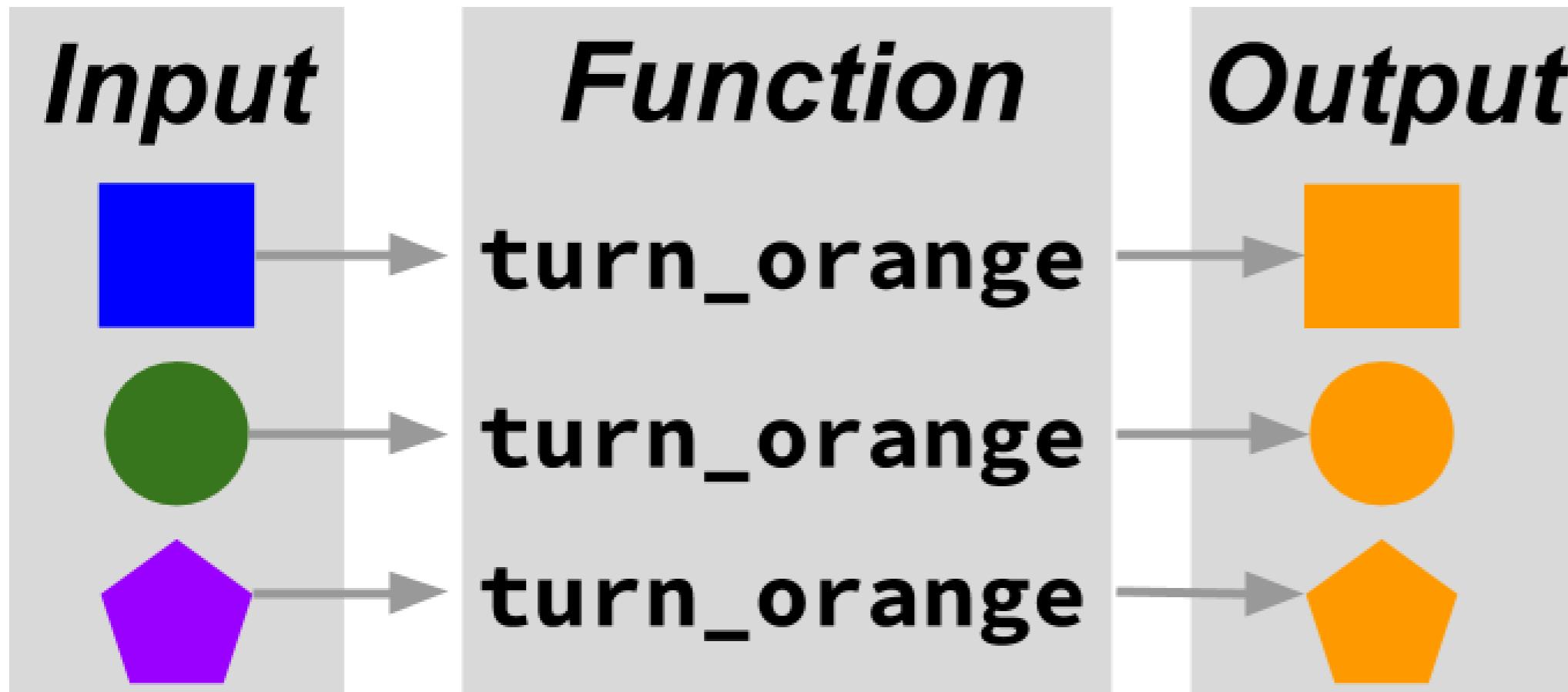
What is a function?

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

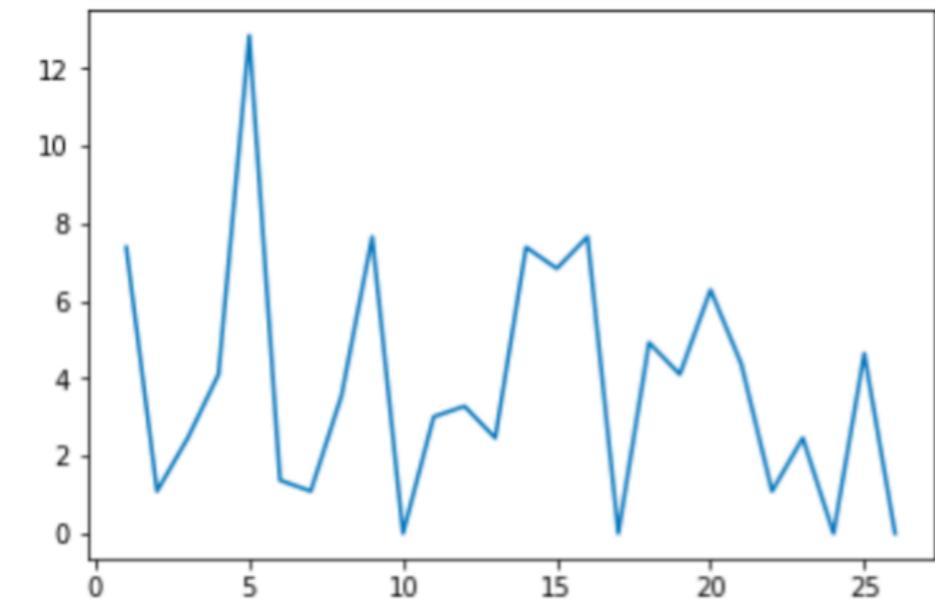
```
plt.plot(df.letter_index, df.frequency, label='Ransom')
```

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
...



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

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 commas!

- Missing closed parenthesis

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



Missing parenthesis!

Let's practice!

INTRODUCTION TO DATA SCIENCE IN PYTHON