Pandas

Session 1

TECH TALENT ACADEMY

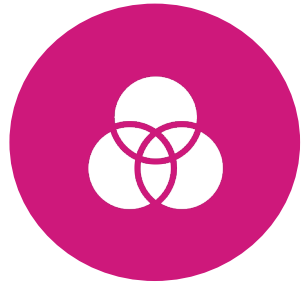
WOMEN IN DATA ACADEMY

Session Content

What is Pandas?

Series/Dataframes

Importing Data



Data Selection & Manipulation

Why Data Analysis?

Trend Spotting

Decision Making

Prediction Making

Problem Solving

What is Pandas?

Software library for use with Python

Uses dataframes

Library facilitates data manipulation, visualisation and analysis

Created by software developer Wes McKinney

Why use Pandas?

You can import, analyse and visualise data easier

Builds on packages such as NumPy

Key concepts of Pandas are indexing and dataframes

What is a Series?

A series is the name for a one-dimensional data structure used within Pandas. If you do not assign an index to your series, Pandas will automatically assign one.

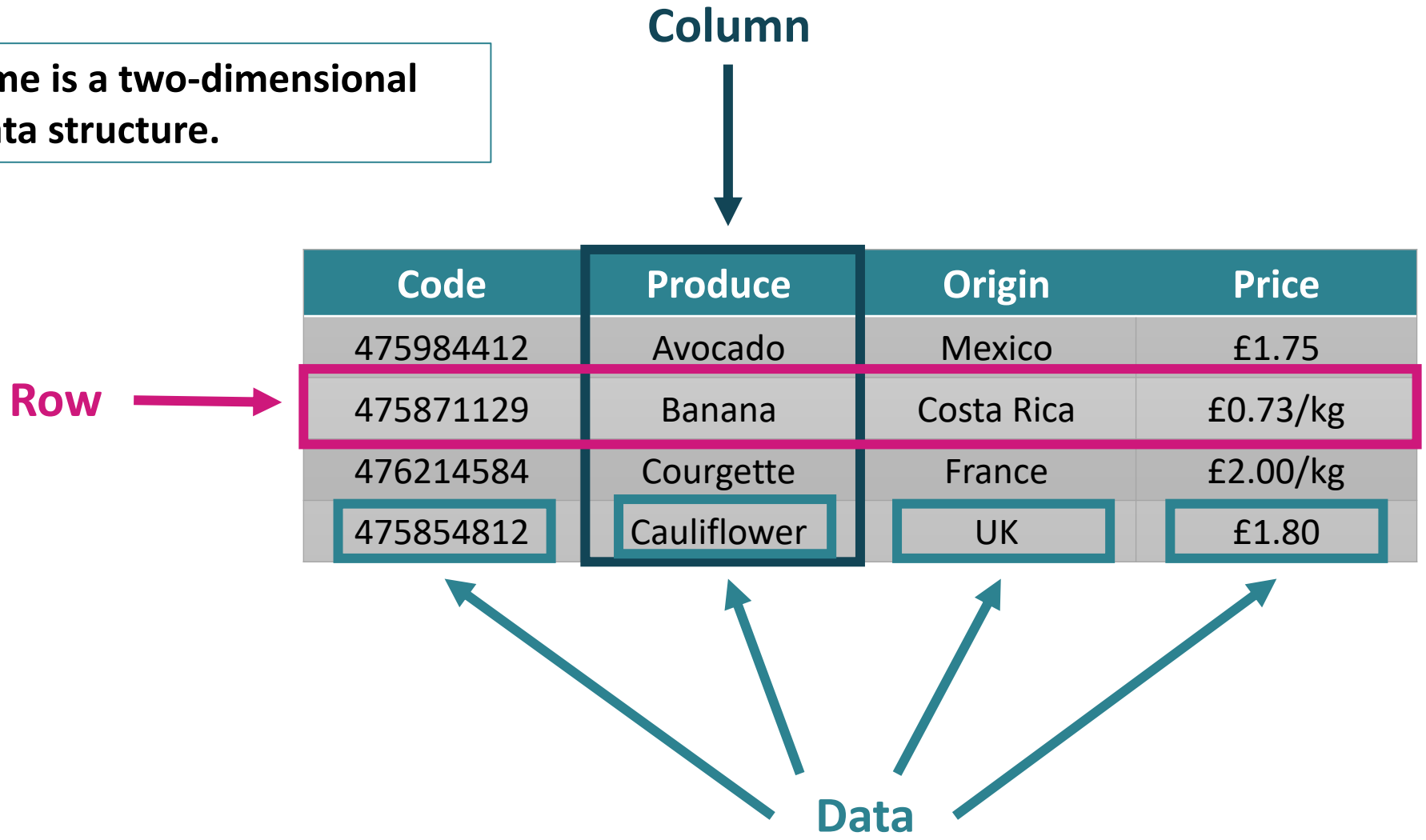
439
98.54
Hello World
Sea Breeze
-342

A series can hold mixed data types

Multiple series can be combined in to a dataframe.

What is a Dataframe?

A dataframe is a two-dimensional tabular data structure.



How to create a dataframe

```
In [2]: # import pandas
        import pandas as pd

        #create data
        inventory = {"Code": [475984412, 475871129, 476214584, 475854812],
                     "Produce": ["Avocado", "Banana", "Courgette", "Cauliflower"],
                     "Origin": ["Mexico", "Costa Rica", "France", "UK"],
                     "Price": [1.75, 0.73, 2.00, 1.80]}
        # create variable
        myframe = pd.DataFrame(inventory)

        # print variable
        print(myframe)
```

```
        Code      Produce      Origin  Price
0  475984412      Avocado      Mexico   1.75
1  475871129       Banana  Costa Rica   0.73
2  476214584    Courgette      France   2.00
3  475854812  Cauliflower          UK   1.80
```

1. **Import pandas**
2. **Create your data**
3. **Create your frame using pd.DataFrame()**

You can introduce different index values such as two letter produce code to make the index value more meaningful.

 # Set the index for frame

frame.index = [“AV", “BN", “CG", "CF"]

 # Print out frame with new index values

print(frame)

```
In [8]: ## Create data
        inventory = {"Code": ["475984412","475871129", "476214584", "475854812"],
                "Produce": ["Avocado", "Banana", "Courgette", "Cauliflower"],
                "Origin": ["Mexico", "Costa Rica", "France", "UK"],
                "Price": [1.75, 0.73, 2.00, 1.80]}
        ## Import Pandas
        import pandas as pd

        ## Create dataframe
        myframe = pd.DataFrame(inventory)

        ## create index
        myframe.index = ["AV", "BN", "CG", "CF"]

        print(myframe)
```

```
        Code      Produce      Origin  Price
AV  475984412      Avocado      Mexico   1.75
BN  475871129       Banana  Costa Rica   0.73
CG  476214584    Courgette      France   2.00
CF  475854812  Cauliflower          UK   1.80
```

Importing external data to Pandas

You can import existing data using Pandas.

The best file type for import is CSV as this is a plain text file.

```
In [9]: ## Import Pandas
        import pandas as pd

        ##read the csv file using pd.read_csv
        vetdata = pd.read_csv("vet_data.csv")

        print(vetdata)
```

```
  Owner_Surname  Pet_Name  Pet_Age  Last_Visit
0         Adams    Fluffy        2  11/05/2020
1         Smith      Zuko        8  30/01/2019
2     Radcliffe      Nala        4  24/11/2019
3       Holland  Mr Chips       10  15/06/2019
4        Potter     Daisy        9  07/04/2020
5        Sorola     Oscar        1  27/02/2020
```

TECH TALENT ACADEMY

Retrieving Specific Data

Use square brackets to retrieve data in specific columns. A single square bracket ([“example”]) will present the data as a series output (i.e without a heading). Double square brackets ([[“example”]]) will retrieve the dataframe heading for that column.

In [17]:
```
## Import Pandas
import pandas as pd

##read the csv file using pd.read_csv
vetdata = pd.read_csv("vet_data.csv")

## retrieving pet name as a series
print(vetdata["Pet_Name"])

##retrieving pet name as a dataframe
print(vetdata[["Pet_Name"]])

##retrieving multiple columns as dataframe
print(vetdata[["Type", "Pet_Age", "Chipped"]])
```

```
0        Fluffy
1          Zuko
2          Nala
3      Mr Chips
4         Daisy
5         Oscar
6         Pepsi
7        George
8         Monty
9           Flo
10        Anton
11        Farah
12        Homer
13         Iggy
14        Bobby
Name: Pet_Name, dtype: object
```

```
     Pet_Name
0      Fluffy
1        Zuko
2        Nala
3    Mr Chips
4       Daisy
5       Oscar
6       Pepsi
7      George
8       Monty
9         Flo
10      Anton
11      Farah
12      Homer
13       Iggy
14      Bobby
```

```
         Type  Pet_Age Chipped
0         Cat        2     Yes
1         Dog        8     Yes
2         Dog        4     Yes
3         Dog       10      No
4      Rabbit        9     Yes
5     Hamster        1      No
6         Cat        6     Yes
7    Tortoise       17      No
8         Dog        3     Yes
9         Cat        7     Yes
10        Cat        2      No
11      Horse        5      No
12        Dog        6     Yes
13        Dog        5     Yes
14     Rabbit        3      No
```


WOMEN IN DATA ACADEMY

Printing Observations (rows)

In [4]:
```
## Import Pandas
import pandas as pd

## read the csv file using pd.read_csv
vetdata = pd.read_csv("vet_data.csv")

## Print the first 4 observations (rows)
print(vetdata[0:4])

## print the 9th, 10th, 11th and 12th observation
print(vetdata[8:12])
```

```
  Owner_Surname  Pet_Name  Pet_Age  Last_Visit Type Chipped
0         Adams    Fluffy        2  11/05/2020  Cat     Yes
1         Smith      Zuko        8  30/01/2019  Dog     Yes
2     Radcliffe      Nala        4  24/11/2019  Dog     Yes
3       Holland  Mr Chips       10  15/06/2019  Dog      No
   Owner_Surname Pet_Name  Pet_Age Last_Visit   Type Chipped
8          Aston    Monty        3        NaN    Dog     Yes
9         Waller      Flo        7        NaN    Cat     Yes
10  De la Force    Anton        2        NaN    Cat      No
11         Reed    Farah        5        NaN  Horse      No
```

Loc and iloc for Data Selection

iloc

Retrieves data based on index location

In [24]:
```
## Import Pandas
import pandas as pd

## read the csv file using pd.read_csv
vetdata = pd.read_csv("vet_data.csv")

## Print out observations for index row 1
print(vetdata.iloc[1])
```

```
Owner_Surname          Smith
Pet_Name                Zuko
Pet_Age                    8
Last_Visit        30/01/2019
Type                     Dog
Chipped                  Yes
Unnamed: 6               NaN
Name: 1, dtype: object
```

loc

Retrieves data based on labels attributed to that data.

In [21]:
```
## Import Pandas
import pandas as pd

## read the csv file using pd.read_csv
vetdata = pd.read_csv("vet_data.csv", index_col ="Owner_Surname")

## Print out observations for Smith and Sorola
print(vetdata.loc[["Smith", "Sorola"]])
```

Owner_Surname	Pet_Name	Pet_Age	Last_Visit	Type	Chipped	Unnamed: 6
Smith	Zuko	8	30/01/2019	Dog	Yes	NaN
Sorola	Oscar	1	27/02/2020	Hamster	No	NaN

Printing columns and ranges using iloc

To select specific rows **and columns** using iloc, you can **separate these with a comma**. Rows always come first.

For example, using **data.iloc[7, 4]** will retrieve you the data that is in the **8th row and the 5th column.**

```
In [15]: import pandas as pd

         data = pd.read_csv("vet_data.csv")

         print(data.iloc[7, 4])

         Tortoise
```

To print a **range** using iloc, you can **use a colon** (:). For example, to **print rows 4-7 and columns 3-4,** you will need the following code:

data.iloc[3:7, 2:4]

```
In [16]: import pandas as pd

         data = pd.read_csv("vet_data.csv")

         print(data.iloc[3:7,2:4])

            Pet_Age  Last_Visit
         3       10  15/06/2019
         4        9  07/04/2020
         5        1  27/02/2020
         6        6   18/092019
```

WOMEN IN DATA ACADEMY

TECH TALENT ACADEMY