Getting Started with Pandas

11 December 2024

15:56

Example of a DataFrame:

A screen shot of a computer program

Description automatically generated

Getting Started with Jupyter Notebook

11 December 2024

16:22

In a Jupyter notebook you can have multiple lines of code in a cell

A white rectangular object with blue lines

Description automatically generated

Enter = new line in cell

Shift + Enter = run code in cell  
Alt + Enter = new cell

D + D = delete cell

\*To select a cell hover mouse + enter

Loading CSV Files

11 December 2024

16:40

A screenshot of a computer

Description automatically generated

Loading JSON Files

11 December 2024

16:40

A screenshot of a computer

Description automatically generated

Loading Excel Files

11 December 2024

16:41

Same as others but requires a sheet name:

A screenshot of a computer

Description automatically generated

Loading Data from Plain Text Files

11 December 2024

16:49

For data separated by commas use pandas.read\_csv()

A screenshot of a computer

Description automatically generated

Otherwise we need to declare a separator:

A screenshot of a computer

Description automatically generated

Set Table Header Row

11 December 2024

16:54

We just need to add **header=** as a parameter:

A screenshot of a computer

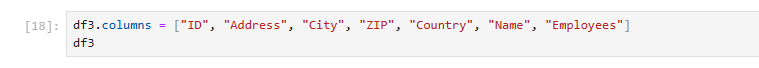
Description automatically generated

Set Column Names:

11 December 2024

16:56

To set column names:



Set Index Column

11 December 2024

16:59

A screenshot of a computer

Description automatically generated

\*\*NOTE\*\* Set index will overlap pervious index

(, drop=False) will keep add the column and also keep as row header

(, inplace=True) saves to Dataframe

Filtering Data from a pandas Dataframe

11 December 2024

17:02

To understand Indexing - we can try and use df.loc to **locate** specific data in a dataframe:

A screenshot of a computer

Description automatically generated

This allows us to use ["Row":"Row","Column":"Column"]

We can also search entire rows / columns:

A screenshot of a computer

Description automatically generated

And also make this a list using Python built in function:

A screenshot of a computer

Description automatically generated

This can also be done with df.iloc and using row / column numbers:

A screenshot of a computer

Description automatically generated

Deleting Columns and Rows

12 December 2024

13:57

To drop columns/rows: df.drop(*Index Number*) - example below misses row 1:

A screenshot of a computer

Description automatically generated

\*\*Note that this does not save to the data - this is only shown as an instance



This can also be done using row Indexes:

A screenshot of a computer

Description automatically generated

Adding Columns and Rows

12 December 2024

14:09

To add a column:



We call DF["Header"]=DF.shape[0]\*["DATA"]

This is because our rows are index of 0 in the tuple in df.shape

\*\*Note this is inplace and will change data permanently

To add a row it is slightly harder:

We have to call df.T to swap the rows and columns:

A screenshot of a computer

Description automatically generated

We then add a column with our data:

A screenshot of a computer

Description automatically generated

And then swap them again using df.T:

A screenshot of a computer

Description automatically generated

----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

To change data in a column:

A screenshot of a computer

Description automatically generated

Example 2:

A screenshot of a computer

Description automatically generated

We call DF["Column Header"]=DF["2nd Column Header"] + "string" + "string2"

df.shape:

A screenshot of a computer

Description automatically generated

This code gives you the shape (rows / columns) of your data frame in a tuple

Data Analysis Example

12 December 2024

14:36

**Note**

We are going to use Nominatim() in the next video. Nominatim() currently has a bug. To fix this problem, whenever you see these lines in the next video:

* 1. from geopy.geocoders import Nominatim
  2. nom = Nominatim()

change them to these

* 1. from geopy.geocoders import ArcGIS
  2. nom = ArcGIS()

The rest of the code remains the same.

Data Analysis Example: Converting Addresses into Coordinates

Using geopy (ArcGIS)



A screenshot of a computer

Description automatically generated

Find Latitude / Longitude: (Stored in variable 'n')

A screenshot of a phone

Description automatically generated

Find Coordinates for multiple rows:

First we need to create a column with all the info needed by geopy:

A screenshot of a computer

Description automatically generated

We then call df["*Coordinates Column*"]=df["*Info Column*"].apply(nom.geopy)

If we want only the latitude / longitude:

We need to use lambda and store .latitude in a temp variable (x)

1. df["Latitude"]=df["Coordinates"].apply(lambda x: x.latitude)
2. Df

If there is None Value (invalid address) we can add an if function:

df["Latitude"]=df["Coordinates"].apply(lambda x: x.latitude if x != None else None)

df