Mkdir pandas

Virtualenv panda

Panda\scripts\activate

Pip install pandas

Pip install jupyterlab

Now go to the folder in command prompt where data is present and type

Jupyter notebook

Click new option and select python

In python terminal type

**Import pandas as pd**

**df = pd.read\_csv(‘data/**[**survey\_results\_public.csv**](http://localhost:8888/edit/data/survey_results_public.csv)’**)**

**df 🡪To read the table copied from csv file to df variable**

**df.shape 🡪 returns the number of colums and records in tuples form.**

**df.info()🡪 returns number of columns,rows,all the column names and their data types**

**pd.set\_option(‘display.max\_columns’,85)**

**pd.set\_option(‘display.max\_rows’,85)**

**df.head🡪To get first five rows**

**df.head(10)🡪To get first 10 rows**

**df.tail🡪To get last 5 rows**

**df.tail(10)🡪To get last 10 rows**

Data Frame is nothing a Table Which Contains Columns and Rows. When we compare pandas Data Frame to Python We can say as list of dictionaries. Where list represents Records or Rows and Keys represents Columns.

**Ex:- people = {‘first’:[‘Karthik’,’Deepthi’,‘Srikanth’],**

**‘last’:[‘Sukavasi’,’Teerthala’,’Sukhavasi’],}**

**Import pandas as pd**

**df=pd.DataFrame(people)**

**Now df variable will get us o/p as dataframe.**

\*\*Data Frame means a container which contains n number of Series objects.

Series is nothing but rows of a \*\* single column.

**df[‘last’]** will return series of data for ‘last’ column.

**df[[‘first’,’last’]]** will return both first and last column records which is also a dataframe because we cant call series because it contains more than one column values.

**df.columns** will return column names of the df data frame.

**df.iloc[0]** will return row data of first index location of a data frame.

**df.iloc[[0,1]]** will return first two rows.

**df.iloc[[0,1],1]** will return 1 index column[‘last’] with first two index of rows.

**df.loc[[0,1],[‘first’,’last’]]** will return first two rows of data for the given columns.

**df.loc[0:1,’first’:’last’]**

**df.set\_index(‘email’,inplace=True)** will set the email as index for the dataset.

**df.loc[‘email\_value’,’first’]** will get the first name record with the given email field.

**df.iloc[0,’first’]** to get the record with giving index value we use iloc.

**df.reset\_index(inplace=True)** to reset the email index of the data frame to normal index values.

**df = pd.read\_csv(‘path\_for\_file’,index\_col=’Respondent’)** will set Respondent Column as a index value to refer. We set Respondent as index value because Respondent Column as unique values.

**df.sort\_index()** will set index column in ascending way

**df.sort\_index(ascending=False)** to set index value in descending order.

**df.sort\_index(inplace=True)** to set index value in ascending order permanently.

**Filters:-**

**filt = (df[‘last’]==’Sukhavasi’)** It returns True or False values for every record.

**df[filt]** It returns all rows where condition is true for that row.

**df.loc[filt,’last’]** It returns only last column Values where condition is true for that row.

**filt=(df[‘last’]==’Sukhavasi) & (df[‘first’]==’Srikanth’)** And operator

**df.loc[filt,’last’]**

**filt=(df[‘last’]==’Sukhavasi) | (df[‘first’]==’Srikanth’)** Or operator

**df.loc[filt,’last’]**

**df.loc[~filt,’last’]** To get the opposite values in filt variable we use tilde(~)

**1.Problem:-** Retrieve country,LanguageWorkedWith,ConvertedComp Column values Where condition should be like ConvertedComp > 7000

To see only the countries listed below rows, We can do as followed 3 lines.

**countries=[‘India’,’United States’,’United Kingdom’]**

**filt = df[‘country’].isin(countries)**

**df.loc[filt,‘country’]**

In every LanguagesWorkedWith Value may include Python with other Languages. To Show all rows which include Python, We can use below two lines.

**filt = df[‘LanguagesWorkedWith’].str.contains(‘Python’,na=False)**

**df.loc[filt,’LanguagesWorkedWith’]** To get only language rows which contains Python String in it.