Data Manipulation

Imorting pandas

In [ ]:

```
import pandas as pd
import numpy as np #For future use
```

Pandas Data Frames

Creating pandas data frames

(01) Using dictionaries

In [ ]:

```
d={"Name":["Sam","Jane","Kane"],"Age":[23,24,25]}
df=pd.DataFrame(d)
df
```

(04) Creating data frames using CSV files

In [ ]:

```
df=pd.read_csv("D:\\Workshops\\W 12 - Python Programming Masterclass Workshop (Batch 02)\\Data\\iris.CSV")
df
```

Top values of the data set

In [ ]:

```
df.head()
```

Bottom values of the data set

In [ ]:

```
df.tail()
```

Columns & indexes of the data set

Accessing columns & elements

```
In [ ]: df=pd.DataFrame({"Col1":[12,22,23,34,45],"Col2":[45,43,44,46,54],"Col3":[56,43
        ,46,67,65]})
        df
```

Accessing columns

```
In [ ]: df.columns
```

```
In [ ]: df["Col1"]
```

```
In [ ]: df[["Col1","Col2"]]
```

Accessing elements

```
In [ ]: df.iloc[0,1]
```

```
In [ ]: df.iat[0,2]
```

```
In [ ]: df.iloc[:3,1]
```

```
In [ ]: df.loc[0,"Col1"]
```

```
In [ ]: df.at[0,"Col2"]
```

Dropping rows & columns

```
In [ ]: df=pd.DataFrame({"Name":["A","B","C"],"Age":[23,33,21],"Result":["Pass","Fail"
        ,"Pass"]},index=["a","b","c"])
        df
```

Dropping rows

```
In [ ]: df1=df.drop("a")
        df1
```

```
In [ ]: df #Oroginal data frame is not affected
```

```
In [ ]: df.drop("a",inplace=True) #Oroginal data frame is affected
        df
```

Dropping columns

```
In [ ]: df=pd.DataFrame({"Name":["A","B","C"],"Age":[23,33,21],"Result":["Pass","Fail"
        ,"Pass"]},index=["a","b","c"])
        df
```

```
In [ ]: df1=df.drop("Result",axis=1)
        df1
```

```
In [ ]: df2=df.drop(["Age","Result"],axis=1)
        df2
```

Concatenating data frames

```
In [ ]: df1=pd.DataFrame(np.random.random((5,2)),columns=["A","B"])
        df2=pd.DataFrame(np.random.random((5,2)),columns=["A","B"])
        df3=pd.DataFrame(np.random.random((5,2)),columns=["C","D"])
```

Concatenating along the rows

```
In [ ]: newdf=pd.concat([df1,df2],ignore_index=True) #ignore_index will reset the inde
        x in the correct way
        newdf
```

Concatenating along the columns

```
In [ ]: newdf=pd.concat([df1,df3],axis=1)
        newdf
```

Counting values

In [ ]:

```
d={"Col1":["P","P","Q"],"Col2":["X","Y","Z"],"Col3":["A","B","C"]}
df=pd.DataFrame(d)
df
```

In [ ]:

```
pd.value_counts(df["Col1"])
```

Working with indexes & columns

In [ ]:

```
d={"Name":["Sam","Pam","Jane"],"Marks":[56,32,78],"Class":[1,2,3]}
df=pd.DataFrame(d,index=["a","b","c"])
df
```

Columns of the data frame

In [ ]:

```
df.columns
```

Indexes of the data frame

In [ ]:

```
df.index
```

Changing the order of the rows

In [ ]:

```
df2=df.reindex(["b","c","a"])
df2
```

Changing the order of the columns

In [ ]:

```
df3=df.reindex(columns=["Marks","Name","Class"])
df3
```

Data Visualization

In [ ]:

```
import seaborn as sns
import matplotlib.pyplot as plt
```

Line Charts

In [ ]:

```
L1=[1,2,3,4]
L2=[10,25,32,58]
plt.plot(L1,L2)
plt.show()
```

In [ ]:

```
L1=[1,2,3,4]
L2=[10,25,32,58]
plt.figure(figsize=(8,7))
plt.plot(L1,L2)
plt.title("A Simple Line Plot")
plt.xlabel("L1")
plt.ylabel("L2")
plt.show()
```

Scatter Plots

In [ ]:

```
height=np.random.normal(160,5,100)
weight=np.random.normal(55,2,100)

plt.scatter(weight,height)
plt.xlabel("Weight")
plt.ylabel("Height")
plt.title("Height VS Weight")
plt.show()
```

In [ ]:

```
df=pd.read_csv("D:\\Workshops\\Python for Data Science Comprehensive Workshop\\Part 03 - Data Visualization\\Datasets\\default.CSV")
df.head()
```

In [ ]:

```
plt.scatter(df["Work.Exp"],df["Credit.Score"])
plt.xlabel("Work Experience")
plt.ylabel("Credit Score")
plt.title("Work Experience VS Credit Score")
plt.show()
```

Heat Maps

In [ ]:

```
df=pd.DataFrame({"Loc1":[0,10,20],"Loc2":[10,0,15],"Loc3":[20,15,0]},index=["Loc1","Loc2","Loc3"])
df
```

```
In [ ]: sns.heatmap(df,annot=True)
        plt.show()
```

```
In [ ]: df=pd.read_csv("D:\\Workshops\\W 12 - Python Programming Masterclass Workshop (Batch 02)\\Data\\iris.CSV")
        df
```

```
In [ ]: df1=df.iloc[:,:4]
        df1.corr()
```

```
In [ ]: sns.heatmap(df1.corr())
        plt.show()
```