To import libraries

- To import pandas:	import pandas as pd
- To import matplotlib:	import matplotlib.pyplot as plt
- To import plotly express (line graph)	import plotly.express as px
- To import plotly go object (scatter graph) [for comparision graph]	import plotly.graph_objects as go
- To import seaborn	import seaborn as sns

To import data from url (use get request):

```
url  = “https://    ”

import requests                                        # response should be 200
response = requests. get(url)
response

with open(“foldername.zip", "wb") as f:                # zipped file data
   f.write(response.content)

import zipfile                                         # unzipped files
with zipfile.ZipFile(“filename.zip") as zipped:
   zipped.extractall(“extract_filename")
```

Pandas

Pandas

	Syntax	**Example**
How to read a csv file	variablename = pd.read_csv (“filepath”)	who = pd.read_csv(“path”)
For Dimensions	filename.ndim	who.ndim
To get column names	filename.columns	who.columns
For Sarting 5 rows	filename.head( )	who.head( )
For Ending 5 rows	filename.tail( )	who.tail( )
For Type	type(filename[“columnname”])	type(who[“Datereported”])
For Information	filename.info( )	who.info( )
For Rows * Columns	filename.shape( )	who.shape( )
For Describe	filename.describe( )	who.describe ( )
For Unique Values	filename.unique( )	who.unique( )
To get unique value from specific column	filename.columnname.unique( )	who.Country.unique( )
For Valuecount (counting of unique values)	filename.columnname.value_counts( )	bike.weathersit.value_counts()

	Syntax	**Example**
For Datatype	filename.dtype	who.dtype
To access any column	filename[“columnname”]	who[“Datereported”]

To replace data in columns
 	
Another way
	filename.columnname.replace({oldname: newname, oldname:newname}, inplace=True)
	
	filename.columnname.map({oldname:newname, oldname:newname})
		bike.yr.replace({0:2011,1:2012}, inplace = True)
		
		bike.yr.map({0:2011, 1:2012})

To drop any column	variablename = filename.drop([‘1st columnname’, ‘2nd columnname’], axis = “columns”).copy( )	incidents = traffic.drop(['Hour (Coded)','Slowness in traffic (%)'], axis = "columns").copy()
For sum of columns	variablename.sum( )	incidents.sum()
For sum of all the sum columns	variablename.sum( ).sum( )	incidents.sum().sum()

	Syntax	**Example**
It gives answer from 0 to 26 index (last value excluded) – For row indexing	df.iloc[:indexnumber]	df.iloc[:27]
It gives answer from 0 to 27 index (last value included)	df.loc[:indexnumber]	df.loc[:27]
For not null value (answer in True or False)	variablename = filename[“columnname”].notnull( )	bol= i_94["holiday"].notnull() i_94[bol]
For null value	variablename = filename[“columnname”].isnull( )	bol1_null = i_94["holiday"].isnull()

For DateTime methods

	Syntax	**Example**
Tells about day in numeric form	filename[“columnname”].dt.day	bike["dteday"].dt.day
Tells about days and months inword form	filename[“columnname”]. dt.strftime ("%A %B")	bike["dteday"].dt.strftime("%A %B")
Tells about month in numeric form	filename.date_time.dt.month	day.date_time.dt.month
Tells about month in word form	filename.date_time.dt. strftime("%B")	day.date_time.dt.strftime("%B")
Tells about year in numeric form	filename. date_time.dt.year	day.date_time.dt.year
For min value of date and time	variablename.date_time.min( )	day.date_time.min()
For max value of date and time	variablename.date_time.max( )	day.date_time.max()