In [5]:

```
import pandas as pd    # dataframe operation
import numpy as np     #math operation
import matplotlib.pyplot as plt   # Diagrams/plots
import seaborn as sns          #Diagrams /plots
```

In [6]:

```
# data set name visa :visadataset
#read csv file : Comma separated value
#extension : .csv
#you can read this using pandas package
# read excel file
# extainsion : .xls
```

In [1]:

```
#path
#file location +filename +extainsion
path =r"C:\Users\tanma\DATASCIENCE\EDA\Visadataset.xlsx"

# C:\Users\tanma\DATASCIENCE\EDA
```

In [12]:

```
pd.read_excel(path)
```

Out[12]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_trainii
0	EZYV01	Asia	High School	N	
1	EZYV02	Asia	Master's	Y	
2	EZYV03	Asia	Bachelor's	N	
3	EZYV04	Asia	Bachelor's	N	
4	EZYV05	Africa	Master's	Y	
...	...	...	...	...	
25475	EZYV25476	Asia	Bachelor's	Y	
25476	EZYV25477	Asia	High School	Y	
25477	EZYV25478	Asia	Master's	Y	
25478	EZYV25479	Asia	Master's	Y	
25479	EZYV25480	Asia	Bachelor's	Y	

25480 rows × 12 columns

In [8]:

```
import os
os.getcwd()
```

Out[8]:

```
'C:\\Users\\tanma\\DATASCIENCE\\EDA'
```

In [16]:

```
path =r'C:\Users\tanma\DATASCIENCE\EDA\bank.csv'
```

In [17]:
```
pd.read_csv(path)
```

Out[17]:
```
      age          job  marital  education  default  balance  housing  loan   contact  day
   0   30   unemployed  married    primary       no     1787       no    no  cellular   19
   1   33     services  married  secondary       no     4789      yes   yes  cellular   11
   2   35   management   single   tertiary       no     1350      yes    no  cellular   16
   3   30   management  married   tertiary       no     1476      yes   yes   unknown    3
   4   59  blue-collar  married  secondary       no        0      yes    no   unknown    5
 ...  ...          ...      ...        ...      ...      ...      ...   ...       ...  ...
4516   33     services  married  secondary       no     -333      yes    no  cellular   30
                 self-
4517   57     employed  married   tertiary      yes    -3313      yes   yes   unknown    9
4518   57   technician  married  secondary       no      295       no    no  cellular   19
4519   28  blue-collar  married  secondary       no     1137       no    no  cellular    6
4520   44 entrepreneur   single   tertiary       no     1136      yes   yes  cellular    3

4521 rows × 17 columns
```

In [ ]:
```
#*** step5th ***
#add new column
name = ['ramesh','suresh','satish']
age = [30,35,40]
df=pd.Dataframe(zip('name','age'))
```

In [ ]:
```
#-if you want to addd a new column
#-df[new column]
#you need to have a list which is having some elements
#that elements need to equal to number of rows
#city _names =['hyderabad','blr','chennai']
#df['city']=city_names
```

In [ ]:
```
city_names=['hyd','blr','chennai']
```

In [ ]:
```

```

In [ ]:
```
#step no6
#update the existing column
-if you want to create new column or update the old column
#both are same way
```

In [ ]:
```
df.drop('city',     # column name
        axis =1,    # column
       inplace=True)  #
```

In [ ]:

```
#step no 7
#DROP THE COLUMN
- in order to drop the the column
- we need to use drop method
-it takes 3 parameters
   - drop column or row
   - mention the column name
   -axis
      -axis =1  reference as column
      - axis =0 reference as row
- you want to create a new dataframe or
-you want to overwrite the existing dataframe
 - inplace true
```

In [20]:

```
name = ['ramesh','suresh','satish']
age =[30,35,40]
name,age
```

Out[20]:

```
(['ramesh', 'suresh', 'satish'], [30, 35, 40])
```

In [22]:

```
#step1
# to create dataframe
pd.DataFrame()
```

Out[22]:

```
—
```

In [23]:

```
#step2
#provide data
pd.DataFrame(zip(name,age))
```

Out[23]:

	0	1
0	ramesh	30
1	suresh	35
2	satish	40

#step no 10 $removetheindex$

In [ ]:

```
#to avoid the above problem
#give index false
df.to_csv("output.csv",index=False)
```

In [ ]:

```

```

$Createadictionaryusing Dataframe$

In [26]:

```
d1={'name':['ramesh','suresh','satish'],
   "age":[30,40,50]}
pd.DataFrame(d1)

# no need of zip
#
```

Out[26]:

	name	age
0	ramesh	30
1	suresh	40
2	satish	50

In [ ]:

In [ ]:

practiceathome

step1

In [14]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [15]:

```
name = ['satish','ramesh','rajesh']
age = [30,35,40]
city = ['hyderabad','banglore','chennai']
marks =[88,90,95]

name,age,city,marks
```

Out[15]:

```
(['satish', 'ramesh', 'rajesh'],
 [30, 35, 40],
 ['hyderabad', 'banglore', 'chennai'],
 [88, 90, 95])
```

In [16]:

```
pd.DataFrame()
```

Out[16]: —

step2

In [17]:

```
#provide data

pd.DataFrame(zip(name,age,city,marks))
```

Out[17]:

	0	1	2	3
0	satish	30	hyderabad	88
1	ramesh	35	banglore	90
2	rajesh	40	chennai	95

In [ ]:

```

```

$step3\ providecolumnname$

In [18]:

```
data = zip(name,age,city,marks)
colms = ['name','age','city','marks']
pd.DataFrame(data ,columns=colms)
```

Out[18]:

	name	age	city	marks
0	satish	30	hyderabad	88
1	ramesh	35	banglore	90
2	rajesh	40	chennai	95

$step4\ provideindex$

In [19]:

```
data = zip(name,age,city,marks)
colms = ['name','age','city','marks']
index = ['a','b','c']
pd.DataFrame(data,columns=colms,index =index)
```

Out[19]:

	name	age	city	marks
a	satish	30	hyderabad	88
b	ramesh	35	banglore	90
c	rajesh	40	chennai	95

$step - 5\ addnewcolumns$

In [12]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

name = ['vikas','ramesh','suresh']
age = [40,45,50]
city =['delhi','culcutta','mumbai']
marks = [56,78,89]

data = zip(name,age,city,marks)
colms = ['name','age','city','marks']
index = ['a','b','c']
df=pd.DataFrame(data,columns=colms,index =index)


fathers_name = ['mr.x','mr.y','mr.z']
df['fathers_name'] = fathers_name
df
```

Out[12]:

	name	age	city	marks	fathers_name
a	vikas	40	delhi	56	mr.x
b	ramesh	45	culcutta	78	mr.y
c	suresh	50	mumbai	89	mr.z

In [13]:

```
data = zip(name,age,city,marks)
colms = ['name','age','city','marks']
index = ['a','b','c']
df=pd.DataFrame(data,columns=colms,index =index)
df
```

Out[13]:

	name	age	city	marks
a	vikas	40	delhi	56
b	ramesh	45	culcutta	78
c	suresh	50	mumbai	89

step6 updatetheexistingcolumns

In [14]:

```
df['name'] = ['dibyanshu','ram kaushik','shaurya']
df
```

Out[14]:

	name	age	city	marks
a	dibyanshu	40	delhi	56
b	ram kaushik	45	culcutta	78
c	shaurya	50	mumbai	89

In [ ]:

```

```

step7 dropthecolumn

In [16]:
```
df.drop('marks',
        axis =1 ,
        inplace =True)
df
```

Out[16]:

	name	age	city
a	dibyanshu	40	delhi
b	ram kaushik	45	culcutta
c	shaurya	50	mumbai

step8 droprows

In [17]:
```
df.drop('c',axis =0 ,inplace = True)
df
```

Out[17]:

	name	age	city
a	dibyanshu	40	delhi
b	ram kaushik	45	culcutta

In [ ]:

step9 savethedataframe

In [18]:
```
df.to_csv("student.csv")
```

In [19]:
```
pd.read_csv("student.csv")
```

Out[19]:

	Unnamed: 0	name	age	city
0	a	dibyanshu	40	delhi
1	b	ram kaushik	45	culcutta

In [ ]:

step10 removetheindex

In [21]:
```
df.to_csv("student.csv",index =False)
```

In [ ]:

Creatdataframesusingdictionary

createdataframeandprovidedata

In [25]:

```
d1 ={"name":['sham','ram','krish'],
       "age":[12,10,13],
    "class":[5,6,7]}
pd.DataFrame(d1)
```

Out[25]:

	name	age	class
0	sham	12	5
1	ram	10	6
2	krish	13	7

providecoumns

In [31]:

```
data={"name":['sham','ram','krish'],
       "age":[12,10,13],
       "class":[5,6,7]}

ind = ['a','b','c']
pd.DataFrame(data ,columns = ['name','age','class'],index= ind)
```

Out[31]:

	name	age	class
a	sham	12	5
b	ram	10	6
c	krish	13	7

In [ ]:

addnewcolumn

In [ ]:

In [ ]:

In [ ]: