Import required packages

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Read the data

In [2]:
```
file_location="C:\\Users\\omkar\\OneDrive\\Documents\\Data science\\Naresh 
visa_df=pd.read_csv(file_location)
visa_df
```

Out[2]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_traini
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

head

In [3]:
```
# your name of the data frame: visa_df
# <package name>.<method name>
visa_df.head()  # It will return top-5
```

Out[3]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_
0	EZYV01	Asia	High School	N	N	
1	EZYV02	Asia	Master's	Y	N	
2	EZYV03	Asia	Bachelor's	N	Y	
3	EZYV04	Asia	Bachelor's	N	N	
4	EZYV05	Africa	Master's	Y	N	

tail

In [4]:
```
visa_df.tail() # LAST 5 Rows
```

Out[4]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_traini
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	

In [5]:
```
visa_df
```

Out[5]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_traini
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

shape

In [6]:
```
visa_df.shape
# shape : (m,n) (rows,columns)
# Number of rows are 25480
# Number of column are 12
```

Out[6]:
```
(25480, 12)
```

In [7]:
```
type(visa_df.shape)
```

Out[7]:
```
tuple
```

In [14]:
```
visa_df.shape[0],visa_df.shape[1]
print("The number of rows are:",visa_df.shape[0])
print("The number of columns are:",visa_df.shape[1])
```

```
The number of rows are: 25480
The number of columns are: 12
```

In [11]:
```
type(visa_df)
```

Out[11]:
```
pandas.core.frame.DataFrame
```

size

In [16]:
```
visa_df.size # rows * columns
visa_df.shape[0]*visa_df.shape[1]
```

Out[16]:
```
305760
```

len

In [17]:
```
len(visa_df)
```

Out[17]:
```
25480
```

columns

In [24]:
```
cols=visa_df.columns
cols
# It will provide all 12 columns
```

Out[24]:
```
Index(['case_id', 'continent', 'education_of_employee', 'has_job_experienc
e',
       'requires_job_training', 'no_of_employees', 'yr_of_estab',
       'region_of_employment', 'prevailing_wage', 'unit_of_wage',
       'full_time_position', 'case_status'],
      dtype='object')
```

In [19]:
```
type(visa_df.columns)
```

Out[19]:
```
pandas.core.indexes.base.Index
```

In [25]:
```
cols_list=cols.to_list()
type(cols_list)
```

Out[25]:
```
list
```

In [26]:
```
# Select cell
# CTRL+A
# CTRL+/

# cols=visa_df.columns
# for i in cols:
#     print(i.capitalize())
```

dtypes

In [27]:
```
visa_df.dtypes
# Object : the values under that column are categorical
# Int/float : The values under that column are numerical
# 1 2 4 8 16 32 64 128 256
```

Out[27]:
```
case_id                  object
continent                object
education_of_employee    object
has_job_experience       object
requires_job_training    object
no_of_employees           int64
yr_of_estab               int64
region_of_employment     object
prevailing_wage         float64
unit_of_wage             object
full_time_position       object
case_status              object
dtype: object
```

In [28]:
```
type(visa_df.dtypes)
```

Out[28]:
```
pandas.core.series.Series
```

info

In [29]:
```
visa_df.info()
# Caseid  25400
# out of 25480 rows 80 values are missed: Null
# Null count   80
# Non Null count 25400
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25480 entries, 0 to 25479
Data columns (total 12 columns):
 #   Column                 Non-Null Count  Dtype
---  ------                 --------------  -----
 0   case_id                25480 non-null  object
 1   continent              25480 non-null  object
 2   education_of_employee  25480 non-null  object
 3   has_job_experience     25480 non-null  object
 4   requires_job_training  25480 non-null  object
 5   no_of_employees        25480 non-null  int64
 6   yr_of_estab            25480 non-null  int64
 7   region_of_employment   25480 non-null  object
 8   prevailing_wage        25480 non-null  float64
 9   unit_of_wage           25480 non-null  object
 10  full_time_position     25480 non-null  object
 11  case_status            25480 non-null  object
dtypes: float64(1), int64(2), object(9)
memory usage: 2.3+ MB
```

isnull

In [35]:
```
visa_df.isna().sum()

# visa_df.isnull().sum()
```

Out[35]:
```
case_id                  0
continent                0
education_of_employee    0
has_job_experience       0
requires_job_training    0
no_of_employees          0
yr_of_estab              0
region_of_employment     0
prevailing_wage          0
unit_of_wage             0
full_time_position       0
case_status              0
dtype: int64
```

- Read the csv file
- head
- tail
- shape
- size
- len
- columns
- dtypes
- info
- isnull

- **Not callable**
  - It is not a function but you provided brackets
  - Remove those brackets
- **Bound method or built in function**
  - Please provide brackets
- **Attribute Error**
  - You provided a method which does not have
  - Please check the spelling
  - or use dir to check the method existed or not

In [53]:
```
visa_df.dtypes
# I want to seperate categorical columns and numerical columns
# Idea: Convert this into dictionary
dtype=dict(visa_df.dtypes)
# for i in dtype:
#     if dtype[i]=='object':
#         print(i,dtype[i])

cat_col=[i for i in dtype if dtype[i]=='object']
num_col=[i for i in dtype if dtype[i]!='object']
cat_col,num_col
```

Out[53]:
```
(['case_id',
  'continent',
  'education_of_employee',
  'has_job_experience',
  'requires_job_training',
  'region_of_employment',
  'unit_of_wage',
  'full_time_position',
  'case_status'],
 ['no_of_employees', 'yr_of_estab', 'prevailing_wage'])
```

In [ ]:
```
file_location="C:\\Users\\omkar\\OneDrive\\Documents\\Data science\\Naresh 
visa_df=pd.read_csv(file_location)
visa_df
visa_df.dtypes
dtype=dict(visa_df.dtypes)
```

In [56]:
```
# visa_df.select_dtypes(include='object'): Dataframe
visa_df.select_dtypes(include='object').columns
visa_df.select_dtypes(exclude='object').columns
```

Out[56]:
```
Index(['no_of_employees', 'yr_of_estab', 'prevailing_wage'], dtype='objec
t')
```

In [ ]:
```
visa_df.dtypes
dtype=dict(visa_df.dtypes)
cat_col=[i for i in dtype if dtype[i]=='object']
num_col=[i for i in dtype if dtype[i]!='object']
cat_col,num_col

visa_df.select_dtypes(include='object').columns
visa_df.select_dtypes(exclude='object').columns
```

In [ ]:
```
# head: first 5 rows
# tail: last 5 rows
# You want some middile rows
# You want specific rows
# you want some specific clumns data
```

take-loc-iloc

In [57]:
```
visa_df.take([1,2,3])
# by default axis=0  -====== > rows
# provide the rows which has index=1,2,3 and all the columns
```

Out[57]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_
1	EZYV02	Asia	Master's	Y	N	
2	EZYV03	Asia	Bachelor's	N	Y	
3	EZYV04	Asia	Bachelor's	N	N	

In [58]:
```
visa_df.take([1,2,3],axis=1)
# The output is : columns
# index=1,2,3 columns and all the rows
```

Out[58]:

	continent	education_of_employee	has_job_experience
0	Asia	High School	N
1	Asia	Master's	Y
2	Asia	Bachelor's	N
3	Asia	Bachelor's	N
4	Africa	Master's	Y
...	...	...	...
25475	Asia	Bachelor's	Y
25476	Asia	High School	Y
25477	Asia	Master's	Y
25478	Asia	Master's	Y
25479	Asia	Bachelor's	Y

25480 rows × 3 columns

In [59]:
```
visa_df.take([100,200,300])
```

Out[59]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	r
100	EZYV101	Asia	Master's	Y	N	
200	EZYV201	Asia	Doctorate	Y	N	
300	EZYV301	Asia	Master's	Y	N	

In [60]:
```
visa_df.take([100,200,300]).take([2,3],axis=1)
```

Out[60]:

	education_of_employee	has_job_experience
100	Master's	Y
200	Doctorate	Y
300	Master's	Y

In [ ]:
```
visa_df.take([1,2,3])   # rows  and all the columns
visa_df.take([1,2,3],axis=1) # columns and all the rows
visa_df.take([100,200,300]).take([2,3],axis=1) # Rows and columns
```

In [ ]:
```
- Read the data
- Quick checks
- type : list ==== methods
         string === str methods
         dict   === dict methods
         index  ==== has some methods==== convert into other
- head/tail/shape/size/len
- columns/dtypes/info
- cat and num

- take based on axis either rows or columns
```

iloc

In [ ]:
```
# Whenever you opened jupyter notebook
# please read the packages and dataframe again
```

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
file_location="C:\\Users\\omkar\\OneDrive\\Documents\\Data science\\Naresh 1
visa_df=pd.read_csv(file_location)
visa_df.head()
```

Out[2]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_
0	EZYV01	Asia	High School	N	N	
1	EZYV02	Asia	Master's	Y	N	
2	EZYV03	Asia	Bachelor's	N	Y	
3	EZYV04	Asia	Bachelor's	N	N	
4	EZYV05	Africa	Master's	Y	N	

In [ ]:
```
visa_df.iloc[<rows>,<columns>]
visa_df.iloc[<start:end>,<start:end>]
visa_df.iloc[[rows index],[columns index]]
```

In [3]:
```
visa_df.iloc[5:10,2:6]
# i want select rows has index 5 to 9
# I want select columns has index 2 to 5
```

Out[3]:

	education_of_employee	has_job_experience	requires_job_training	no_of_employees
5	Master's	Y	N	2339
6	Bachelor's	N	N	4985
7	Bachelor's	Y	N	3035
8	Bachelor's	N	N	4810
9	Doctorate	Y	N	2251

In [5]:
```
visa_df.iloc[5:10,]
visa_df.iloc[5:10,:]
```

Out[5]:

	case_id	continent	education_of_employee	has_job_experience	requires_job_training	no_
5	EZYV06	Asia	Master's	Y	N	
6	EZYV07	Asia	Bachelor's	N	N	
7	EZYV08	North America	Bachelor's	Y	N	
8	EZYV09	Asia	Bachelor's	N	N	
9	EZYV10	Europe	Doctorate	Y	N	

In [7]:
```
visa_df.iloc[:,:2]
```

Out[7]:

	case_id	continent
0	EZYV01	Asia
1	EZYV02	Asia
2	EZYV03	Asia
3	EZYV04	Asia
4	EZYV05	Africa
...	...	...
25475	EZYV25476	Asia
25476	EZYV25477	Asia
25477	EZYV25478	Asia
25478	EZYV25479	Asia
25479	EZYV25480	Asia

25480 rows × 2 columns

In [ ]:
```
visa_df.iloc[5:10,2:6] # 5 to 9 index rows , 2 to 5 index columns
visa_df.iloc[5:10,:] # All the columns , 5 to 9 index rows
visa_df.iloc[:,:2] # All the rows only 0 and 1 column
```

In [8]:
```
rows=[100,200,300]
cols=[2,7]
visa_df.iloc[rows,cols]
```

Out[8]:

	education_of_employee	region_of_employment
100	Master's	Northeast
200	Doctorate	West
300	Master's	Midwest

In [9]:
```
cols=[2,7]
visa_df.iloc[100:103,cols]
```

Out[9]:

	education_of_employee	region_of_employment
100	Master's	Northeast
101	Master's	Midwest
102	Bachelor's	Midwest

In [16]:
```
rows=[100,200,300]
cols=['education_of_employee','region_of_employment']
visa_df.loc[rows,cols]
```

Out[16]:

	education_of_employee	region_of_employment
100	Master's	Northeast
200	Doctorate	West
300	Master's	Midwest

Note

- iloc takes only numerical entry, we can not provide column names directly
- loc takes only column names

In [ ]:
```
```