In [ ]:

```python
import pandas as pd
import numpy as np
import matplotlib as pyplot
import seaborn as sns
```

In [ ]:

```python
data= pd.read_csv("C:\\Users\\Asus\\DS\\Data Sets\\income.csv")
```

In [ ]:

```python
#Print the size of the dataframe
data.shape
```

Out[ ]:

```
(31978, 13)
```

In [ ]:

```python
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 31978 entries, 0 to 31977
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   age            31978 non-null  int64 
 1   JobType        31978 non-null  object
 2   EdType         31978 non-null  object
 3   maritalstatus  31978 non-null  object
 4   occupation     31978 non-null  object
 5   relationship   31978 non-null  object
 6   race           31978 non-null  object
 7   gender         31978 non-null  object
 8   capitalgain    31978 non-null  int64 
 9   capitalloss    31978 non-null  int64 
 10  hoursperweek   31978 non-null  int64 
 11  nativecountry  31978 non-null  object
 12  SalStat        31978 non-null  object
dtypes: int64(4), object(9)
memory usage: 3.2+ MB
```

In [ ]:

```python
#print basic statistical details (Numerical Variables)
data.describe()
```

Out[ ]:

	age	capitalgain	capitalloss	hoursperweek
count	31978.000000	31978.000000	31978.000000	31978.000000
mean	38.579023	1064.360623	86.739352	40.417850
std	13.662085	7298.596271	401.594301	12.345285
min	17.000000	0.000000	0.000000	1.000000
25%	28.000000	0.000000	0.000000	40.000000
50%	37.000000	0.000000	0.000000	40.000000
75%	48.000000	0.000000	0.000000	45.000000
max	90.000000	99999.000000	4356.000000	99.000000

In [ ]:

```python
#print basic statistical details (Categorical Variables)
#pd.set_option("display.max_columns",None)
```

```
data.describe(include="O")
```

Out[ ]:

	JobType	EdType	maritalstatus	occupation	relationship	race	gender	nativecountry	SalS
count	31978	31978	31978	31978	31978	31978	31978	31978	319
unique	9	16	7	15	6	5	2	41	
top	Private	HS-grad	Married-civ-spouse	Prof-specialty	Husband	White	Male	United-States	l thar eq 50,(
freq	22286	10368	14692	4038	12947	27430	21370	29170	242

In [ ]:

```
#Missing Values
#Standard Missing Values: The missing values which are detected by python are called s
#The missing values detected by python include NaN and blank spaces

#Non-Standard Missing Values
#The missing values such as ?, - , NA are not detected by python and are known as the
```

In [ ]:

```
# No missing Values
data.isnull().sum()
```

Out[ ]:

```
age              0
JobType          0
EdType           0
maritalstatus    0
occupation       0
relationship     0
race             0
gender           0
capitalgain      0
capitalloss      0
hoursperweek     0
nativecountry    0
SalStat          0
dtype: int64
```

In [ ]:

```
#Check features individually
#Contains ?
data["JobType"].value_counts()
```

Out[ ]:

```
 Private             22286
 Self-emp-not-inc     2499
 Local-gov            2067
 ?                    1809
 State-gov            1279
 Self-emp-inc         1074
 Federal-gov           943
 Without-pay            14
 Never-worked            7
Name: JobType, dtype: int64
```

In [ ]:

```
data["EdType"].value_counts()
#No missing Value
```

Out[ ]:

```
 HS-grad         10368
 Some-college     7187
 Bachelors        5210
 Masters          1674
 Assoc-voc        1366
 11th             1167
 Assoc-acdm       1055
 10th              921
 7th-8th           627
 Prof-school       559
 9th               506
 12th              417
 Doctorate         390
 5th-6th           318
 1st-4th           163
 Preschool          50
Name: EdType, dtype: int64
```

In [ ]:

```
data["maritalstatus"].value_counts()
#No Missing Values
```

Out[ ]:

```
 Married-civ-spouse       14692
 Never-married            10488
 Divorced                  4394
 Separated                 1005
 Widowed                    979
 Married-spouse-absent      397
 Married-AF-spouse           23
Name: maritalstatus, dtype: int64
```

In [ ]:

```
data["occupation"].value_counts() #contains ?
```

Out[ ]:

```
 Prof-specialty       4038
 Craft-repair         4030
 Exec-managerial      3992
 Adm-clerical         3721
 Sales                3584
 Other-service        3212
 Machine-op-inspct    1966
 ?                    1816
 Transport-moving     1572
 Handlers-cleaners    1350
 Farming-fishing       989
 Tech-support          912
 Protective-serv       644
 Priv-house-serv       143
 Armed-Forces            9
Name: occupation, dtype: int64
```

In [ ]:

```
#Checking Again
print(np.unique(data["JobType"]))
```

```
[' ?' ' Federal-gov' ' Local-gov' ' Never-worked' ' Private'
 ' Self-emp-inc' ' Self-emp-not-inc' ' State-gov' ' Without-pay']
```

In [ ]:

```
print(np.unique(data["occupation"]))
```

```
[' ?' ' Adm-clerical' ' Armed-Forces' ' Craft-repair' ' Exec-managerial'
 ' Farming-fishing' ' Handlers-cleaners' ' Machine-op-inspct'
 ' Other-service' ' Priv-house-serv' ' Prof-specialty' ' Protective-serv'
 ' Sales' ' Tech-support' ' Transport-moving']
```

```
In [ ]: #na_values: This is used to create a string that considers pandas as NaN (Not a Number
        data=pd.read_csv('C:\\Users\\Asus\\DS\\Data Sets\\income.csv',na_values=[' ?'])
```

```
In [ ]: #Shows Missing Values
        #Occupation and JobType has missing values
        data.isnull().sum()
```

```
Out[ ]: age                0
        JobType         1809
        EdType             0
        maritalstatus      0
        occupation      1816
        relationship       0
        race               0
        gender             0
        capitalgain        0
        capitalloss        0
        hoursperweek       0
        nativecountry      0
        SalStat            0
        dtype: int64
```

```
In [ ]: # calculate percentage of the missing values
        #Drop the feature if mising data>40%
        percent = ((data.isnull().sum()/data.shape[0])*100)
        print(percent)
```

```
age              0.000000
JobType          5.657014
EdType           0.000000
maritalstatus    0.000000
occupation       5.678904
relationship     0.000000
race             0.000000
gender           0.000000
capitalgain      0.000000
capitalloss      0.000000
hoursperweek     0.000000
nativecountry    0.000000
SalStat          0.000000
dtype: float64
```

```
In [ ]: #any() returns true if any of the element in the passed list is true
        #axis=1 is across columns
        #axis=0 is across rows
        missing=data[data.isnull().any(axis=1)]
        print(missing)
```

```
       age JobType         EdType        maritalstatus occupation  \
8       17     NaN           11th        Never-married        NaN   
17      32     NaN   Some-college   Married-civ-spouse        NaN   
29      22     NaN   Some-college        Never-married        NaN   
42      52     NaN           12th        Never-married        NaN   
44      63     NaN        1st-4th   Married-civ-spouse        NaN   
...    ...     ...            ...                  ...        ...   
31892   59     NaN      Bachelors   Married-civ-spouse        NaN   
31934   20     NaN        HS-grad        Never-married        NaN   
31945   28     NaN   Some-college   Married-civ-spouse        NaN   
31967   80     NaN        HS-grad              Widowed        NaN   
31968   17     NaN           11th        Never-married        NaN   

          relationship    race   gender  capitalgain  capitalloss  \
8            Own-child   White   Female            0            0   
17             Husband   White     Male            0            0   
29           Own-child   White     Male            0            0   
42      Other-relative   Black     Male          594            0   
44             Husband   White     Male            0            0   
...                ...     ...      ...          ...          ...   
31892          Husband   White     Male            0            0   
31934   Other-relative   White   Female            0            0   
31945             Wife   White   Female            0         1887   
31967    Not-in-family   White     Male            0            0   
31968        Own-child   White     Male            0            0   

       hoursperweek   nativecountry                        SalStat  
8                 5   United-States   less than or equal to 50,000  
17               40   United-States   less than or equal to 50,000  
29               40   United-States   less than or equal to 50,000  
42               40   United-States   less than or equal to 50,000  
44               35   United-States   less than or equal to 50,000  
...             ...             ...                            ...  
31892            40   United-States            greater than 50,000  
31934            35   United-States   less than or equal to 50,000  
31945            40   United-States            greater than 50,000  
31967            24   United-States   less than or equal to 50,000  
31968            40   United-States   less than or equal to 50,000  

[1816 rows x 13 columns]
```

In [ ]:

```
#First Solution
#Drop Null Values
data=data.dropna(axis=0)
```

In [ ]:

```
data.isnull().sum()
```

Out[ ]:

```
age              0
JobType          0
EdType           0
maritalstatus    0
occupation       0
relationship     0
race             0
gender           0
capitalgain      0
capitalloss      0
hoursperweek     0
nativecountry    0
SalStat          0
dtype: int64
```

In [ ]:

```
#Second solution
#Impute Missing Values
#For Numerical data use mean/median (As Discussed in class)
#for Categorical data use mode
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30162 entries, 0 to 31977
Data columns (total 13 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   age            30162 non-null  int64 
 1   JobType        30162 non-null  object
 2   EdType         30162 non-null  object
 3   maritalstatus  30162 non-null  object
 4   occupation     30162 non-null  object
 5   relationship   30162 non-null  object
 6   race           30162 non-null  object
 7   gender         30162 non-null  object
 8   capitalgain    30162 non-null  int64 
 9   capitalloss    30162 non-null  int64 
 10  hoursperweek   30162 non-null  int64 
 11  nativecountry  30162 non-null  object
 12  SalStat        30162 non-null  object
dtypes: int64(4), object(9)
memory usage: 3.2+ MB
```

In [ ]:

```
#Obtain the mode value for JobType
# Private is the mode for  JobType
data['JobType'].mode()
```

Out[ ]:

```
0     Private
dtype: object
```

In [ ]:

```
# replace all the missing values with 'Private'
data.JobType.replace(np.NaN,"Private" ,inplace = True)
```

In [ ]:

```
#Obtain the mode value for Occupation
#Prof-specialty is the mode for occupation
data['occupation'].mode()
```

Out[ ]:

```
0     Prof-specialty
dtype: object
```

In [ ]:

```
# replace all the missing values with 'Prof-specialty'
data.occupation.replace(np.NaN,"Prof-specialty" ,inplace = True)
```

When inplace = True , the data is modified in place, which means it will return nothing and the dataframe is now updated. When inplace = False , which is the default, then the operation is performed and it returns a copy of the object. You then need to save it to something.

In [ ]:

```
data.isnull().sum()
```

Out[ ]:

```
age              0
JobType          0
EdType           0
maritalstatus    0
occupation       0
relationship     0
race             0
gender           0
capitalgain      0
capitalloss      0
hoursperweek     0
nativecountry    0
SalStat          0
dtype: int64
```