```
import pandas as pd
import numpy as np
```

HR DATASET

```
df=pd.read_csv('HR-Employee-Attrition.csv')
```

```
df.head(10)
```

	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Educatic
0	41	Yes	Travel_Rarely	1102	Sales	1	
1	49	No	Travel_Frequently	279	Research & Development	8	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	
3	33	No	Travel_Frequently	1392	Research & Development	3	
4	27	No	Travel_Rarely	591	Research & Development	2	
5	32	No	Travel_Frequently	1005	Research & Development	2	
6	59	No	Travel_Rarely	1324	Research & Development	3	
7	30	No	Travel_Rarely	1358	Research & Development	24	
8	38	No	Travel_Frequently	216	Research & Development	23	
9	36	No	Travel_Rarely	1299	Research & Development	27	

10 rows × 35 columns

```
df.describe()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	Employe
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024
std	9.135373	403.509100	8.106864	1.024165	0.0	602
min	18.000000	102.000000	1.000000	1.000000	1.0	1
25%	30.000000	465.000000	2.000000	2.000000	1.0	491
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068

8 rows × 26 columns

```
print("The mean of monthly income is :",df.loc[:,"MonthlyIncome"].mean())
print("The mean of age is :",df.loc[:,"Age"].mean())
```

```
The mean of monthly income is : 6502.931292517007
The mean of age is : 36.923809523809524
```

```
print("The median of monthly income is :",df.loc[:,"MonthlyIncome"].median())
print("The median of age is :",df.loc[:,"Age"].median())
```

```
The median of monthly income is : 4919.0
The median of age is : 36.0
```

```
print("The mode of monthly income is :",df.loc[:,"MonthlyIncome"].mode())
print("The mode of age is :",df.loc[:,"Age"].mode())
```

```
The mode of monthly income is : 0    2342
Name: MonthlyIncome, dtype: int64
The mode of age is : 0    35
Name: Age, dtype: int64
```

```
print("The standard deviation of monthly income is :",df.loc[:,"MonthlyIncome"].std())
print("The standard deviation of age is :",df.loc[:,"Age"].std())
```

```
The standard deviation of monthly income is : 4707.956783097995
The standard deviation of age is : 9.135373489136734
```

```
array1 = np.array(df['MonthlyIncome'])
array2=np.array(df["Age"])
print("Income",array1)
print("Age array",array2)
print("Maximum income among the employees is :",max(array1))
print("Minimum income among the employees is :",min(array1))
```

```
print("Maximum age among the employees is :",max(array2))
print("Minimum age among the employees is :",min(array2))
```

```
Income [5993 5130 2090 ... 6142 5390 4404]
Age array [41 49 37 ... 27 49 34]
Maximum income among the employees is : 19999
Minimum income among the employees is : 1009
Maximum age among the employees is : 60
Minimum age among the employees is : 18
```

```
df.dtypes
```

	0
Age	int64
Attrition	object
BusinessTravel	object
DailyRate	int64
Department	object
DistanceFromHome	int64
Education	int64
EducationField	object
EmployeeCount	int64
EmployeeNumber	int64
EnvironmentSatisfaction	int64
Gender	object
HourlyRate	int64
JobInvolvement	int64
JobLevel	int64
JobRole	object
JobSatisfaction	int64
MaritalStatus	object
MonthlyIncome	int64
MonthlyRate	int64
NumCompaniesWorked	int64
Over18	object
OverTime	object
PercentSalaryHike	int64
PerformanceRating	int64
RelationshipSatisfaction	int64
StandardHours	int64
StockOptionLevel	int64
TotalWorkingYears	int64
TrainingTimesLastYear	int64
WorkLifeBalance	int64
YearsAtCompany	int64

YearsInCurrentRole	int64
YearsSinceLastPromotion	int64
YearsWithCurrManager	int64

dtype: object

```
df1=df
df1["BusinessTravel"].replace({"Travel_Rarely":1, "Travel_Frequently":0},inplace=True)
df1["Attrition"].replace({ "Yes":1, "No":0}, inplace=True)
df1['BusinessTravel'].head(10)
```

```
<ipython-input-11-85d3939a9c74>:2: FutureWarning: A value is trying to be set on a co
The behavior will change in pandas 3.0. This inplace method will never work because t

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({


  df1["BusinessTravel"].replace({"Travel_Rarely":1, "Travel_Frequently":0},inplace=Tr
<ipython-input-11-85d3939a9c74>:3: FutureWarning: A value is trying to be set on a co
The behavior will change in pandas 3.0. This inplace method will never work because t

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({


  df1["Attrition"].replace({ "Yes":1, "No":0}, inplace=True)
<ipython-input-11-85d3939a9c74>:3: FutureWarning: Downcasting behavior in `replace` i
  df1["Attrition"].replace({ "Yes":1, "No":0}, inplace=True)
```

	BusinessTravel
0	1
1	0
2	1
3	0
4	1
5	0
6	1
7	1
8	0
9	1

dtype: object

IRIS DATASET

```
data= pd.read_csv('iris.csv')
```

```
data.head(5)
```

	sepal_length	sepal_width	petal_length	petal_width	species
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa

Next steps: Generate code with data | View recommended plots | New interactive sheet

```
data.describe()
```

	sepal_length	sepal_width	petal_length	petal_width
count	150.000000	150.000000	150.000000	150.000000
mean	5.843333	3.054000	3.758667	1.198667
std	0.828066	0.433594	1.764420	0.763161
min	4.300000	2.000000	1.000000	0.100000
25%	5.100000	2.800000	1.600000	0.300000
50%	5.800000	3.000000	4.350000	1.300000
75%	6.400000	3.300000	5.100000	1.800000
max	7.900000	4.400000	6.900000	2.500000

```
print("The mean of sepal length is :",data.loc[:,"sepal_length"].mean())
print("The mean of petal length is :",data.loc[:,"petal_length"].mean())
```

```
The mean of sepal length is : 5.843333333333334
The mean of petal length is : 3.758666666666666
```

```
print("The median of sepal length is :",data.loc[:,"sepal_length"].median())
```