TASK - 1 | Run the example as given in Listing - 1. Print top 10 rows of the data and also print the data summary as given in Figure - 2.

Top 10 Rows

In [ ]:

```
import seaborn as sns
import pandas as panda

data_frame_tips = sns.load_dataset("tips")
data_frame_tips.head(10)
```

Out[ ]:

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4
5	25.29	4.71	Male	No	Sun	Dinner	4
6	8.77	2.00	Male	No	Sun	Dinner	2
7	26.88	3.12	Male	No	Sun	Dinner	4
8	15.04	1.96	Male	No	Sun	Dinner	2
9	14.78	3.23	Male	No	Sun	Dinner	2

Data Summary

In [ ]:

```
data_frame_tips.describe()
```

Out[ ]:

	total_bill	tip	size
count	244.000000	244.000000	244.000000
mean	19.785943	2.998279	2.569672
std	8.902412	1.383638	0.951100
min	3.070000	1.000000	1.000000
25%	13.347500	2.000000	2.000000
50%	17.795000	2.900000	2.000000
75%	24.127500	3.562500	3.000000
max	50.810000	10.000000	6.000000

TASK - 2 | Repeat Task 1 on pima-indians-diabetes.csv dataset.

Top 10 Rows

In [ ]:

```
data_diabetes = panda.read_csv("diabetes.csv")
data_diabetes.head(10)
```

Out[ ]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunc
0	6	148	72	35	0	33.6	0
1	1	85	66	29	0	26.6	0
2	8	183	64	0	0	23.3	0
3	1	89	66	23	94	28.1	0
4	0	137	40	35	168	43.1	2
5	5	116	74	0	0	25.6	0
6	3	78	50	32	88	31.0	0
7	10	115	0	0	0	35.3	0
8	2	197	70	45	543	30.5	0
9	8	125	96	0	0	0.0	0

Data Summary

In [ ]:

```
data_diabetes.describe().T
```

Out[ ]:

	count	mean	std	min	25%	50%	
Pregnancies	768.0	3.845052	3.369578	0.000	1.00000	3.0000	6.0
Glucose	768.0	120.894531	31.972618	0.000	99.00000	117.0000	140.2
BloodPressure	768.0	69.105469	19.355807	0.000	62.00000	72.0000	80.0
SkinThickness	768.0	20.536458	15.952218	0.000	0.00000	23.0000	32.0
Insulin	768.0	79.799479	115.244002	0.000	0.00000	30.5000	127.2
BMI	768.0	31.992578	7.884160	0.000	27.30000	32.0000	36.6
DiabetesPedigreeFunction	768.0	0.471876	0.331329	0.078	0.24375	0.3725	0.6
Age	768.0	33.240885	11.760232	21.000	24.00000	29.0000	41.0
Outcome	768.0	0.348958	0.476951	0.000	0.00000	0.0000	1.0

TASK - 3 | Generate dot plot and distribution plot on heart.csv dataset.

DotPlot

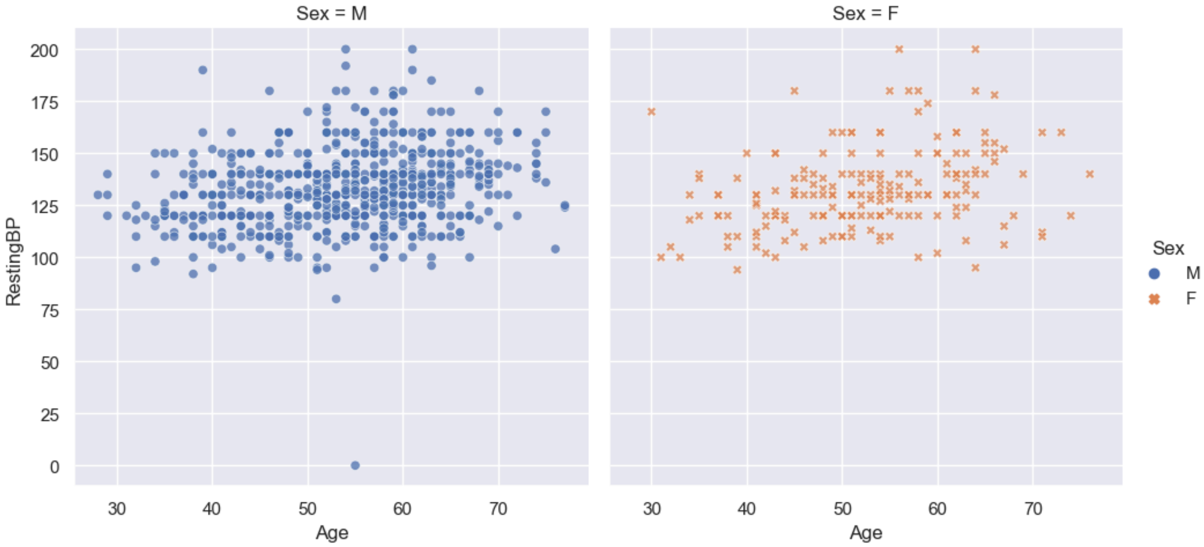
In [ ]:

```
data_heart = panda.read_csv("heart.csv")
data_heart.head()
sns.relplot(data=data_heart,x="Age",y="RestingBP",col="Sex",hue="Sex",style="Sex",a
```

```
D:\ANACONDA\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layou
t has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[ ]:

```
<seaborn.axisgrid.FacetGrid at 0x22d0d4f8210>
```



Distribution Plot (Histogram)

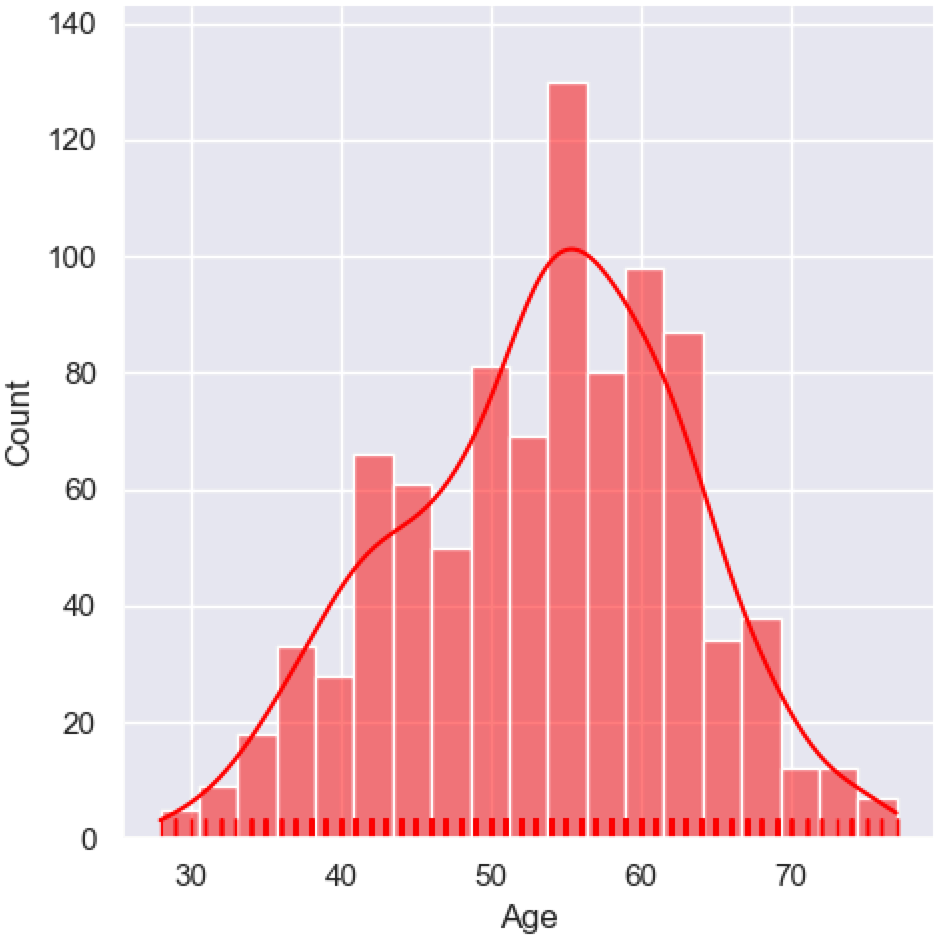
In [ ]:

```
sns.displot(data=data_heart,x='Age',kde=True,rug=True,color="red")
```

```
D:\ANACONDA\Lib\site-packages\seaborn\axisgrid.py:118: UserWarning: The figure layou
t has changed to tight
  self._figure.tight_layout(*args, **kwargs)
```

Out[ ]:

```
<seaborn.axisgrid.FacetGrid at 0x22d6bc2ab50>
```



TASK - 4 | Run the example as given in Listing - 3 and review the number of missing values in the dataset before and after the data imputation transform.

In [ ]:

```
# statistical imputation transfo rm for the horse colic dataset
from numpy import isnan
from pandas import read_csv
from sklearn.impute import SimpleImputer
# load dataset
url = "https://raw.githubusercontent.com/jbrownlee/Datasets/master/horse-colic.csv"
dataframe = read_csv(url,header=None, na_values = "?")
# split into input and output elements
data = dataframe.values
ix = [i for i in range ( data.shape [1]) if i != 23]
X , y = data [: , ix ], data [: , 23]
# print total missing
print ( "Missing(Before) : % d " % sum ( isnan(X).flatten ()))
```

```
# define imputer
imputer = SimpleImputer(strategy="constant")
# fit on the dataset
imputer.fit (X)
# transform the dataset
Xtrans = imputer.transform(X)
# print total missing
print ("Missing(After) : % d " % sum (isnan(Xtrans).flatten()))
```

```
Missing(Before) :  1605
Missing(After) :  0
```

TASK - 5 | Repeat Task 4 on heart.csv dataset.

In [ ]:

```
from pandas import read_csv
from sklearn.impute import SimpleImputer

# Load dataset
dataframe = read_csv("heart.csv", header=None, na_values="?")

# Split into input and output elements
x = dataframe.drop(columns=[11])  # Exclude the target variable (assuming it's the
y = dataframe[11]  # Assuming the target variable is in the last column

# Print total missing values
print("Missing(Before): %d" % x.isnull().sum().sum())

# Define imputer
imputer = SimpleImputer(strategy="mean")

# Fit and transform the dataset
Xtrans = imputer.fit_transform(X)

# Print total missing values after transformation
print("Missing(After): %d" % panda.DataFrame(Xtrans).isnull().sum().sum())
```

```
Missing(Before): 1
Missing(After): 0
```