In [133…
```
import pandas as pd,numpy as np,os
import matplotlib.pyplot as plt
%matplotlib inline
```

In [134…
```
os.chdir('D:\machine learning\Raw data')
```

In [135…
```
os.listdir()
```

Out[135…
```
['a SQL',
 'Advertising.csv',
 'automobile_data.sas7bdat',
 'Automobile_data.xlsx',
 'Automobile_data2.csv',
 'Automobile_data2.xlsx',
 'bank_data.sas7bdat',
 'bigmart_data.csv',
 'Book1.xlsx',
 'carsnew2.xlsx',
 'casnew.csv',
 'churn.csv',
 'churn.xlsx',
 'churn2.csv',
 'churn_data.pickle',
 'churn_data.xlsx',
 'chur_12.xlsx',
 'cleaned data',
 'concrete_data.csv',
 'Covid_data.xlsx',
 'CREDIT_DISCOVERY_FOR_DS.csv',
 'data.csv',
 'dubai_refreshments_final.sas7bdat',
 'employees.csv',
 'employee_detail.sas7bdat',
 'german.data.txt',
 'Gold.xlsx',
 'House Price.csv',
 'House_Price_Scoring.csv',
 'machine learning',
 'MANJU.csv',
 'marks',
 'merging',
 'nortel.csv',
 'payroll2.csv',
 'Problem Statement.docx',
 'state gdp',
 'test.csv',
 'Titanic_data.csv',
 'train.csv',
 'user devise',
 'user_usage.xlsx']
```

In [136…
```
df=pd.read_csv('Titanic_data.csv',usecols=['Age','Fare','Survived'])
```

In [137…
```
df.head()
```

Out[137…

	Survived	Age	Fare
0	0	22.0	7.2500
1	1	38.0	71.2833
2	1	26.0	7.9250
3	1	35.0	53.1000
4	0	35.0	8.0500

In [138…
```
df.isnull().sum()
```

Out[138…
```
Survived      0
Age         177
Fare          0
dtype: int64
```

In [139…
```
df.isnull().mean()
```

Out[139…
```
Survived    0.000000
Age         0.198653
Fare        0.000000
dtype: float64
```

In [140…
```
df['Age'].isnull().sum()
```

Out[140…
```
177
```

In [141…
```
df['Age'].dropna().sample(df['Age'].isnull().sum(),random_state=0)
```

Out[141…
```
423    28.00
177    50.00
305     0.92
292    36.00
889    26.00
       ...
539    22.00
267    25.00
352    15.00
99     34.00
689    15.00
Name: Age, Length: 177, dtype: float64
```

In [142…
```
df[df['Age'].isnull()].index
```

Out[142…
```
Int64Index([  5,  17,  19,  26,  28,  29,  31,  32,  36,  42,
            ...
            832, 837, 839, 846, 849, 859, 863, 868, 878, 888],
           dtype='int64', length=177)
```

In [143…
```
df[df['Age'].isnull()].index
```

Out[143…
```
Int64Index([  5,  17,  19,  26,  28,  29,  31,  32,  36,  42,
            ...
            832, 837, 839, 846, 849, 859, 863, 868, 878, 888],
           dtype='int64', length=177)
```

In [144…
```
def impute_nan(df,variable,median):
    df[variable+"_median"]=df[variable].fillna(median)
    df[variable+"_random"]=df[variable]
    ##It will have the random sample to fill the na
    random_sample=df[variable].dropna().sample(df[variable].isnull().sum(),random_state=0)
    ##pandas need to have same index in order to merge the dataset
    random_sample.index=df[df[variable].isnull()].index
    df.loc[df[variable].isnull(),variable+'_random']=random_sample
```

In [145…
```
median=df.Age.median()
```

In [146…
```
impute_nan(df,"Age",median)
```

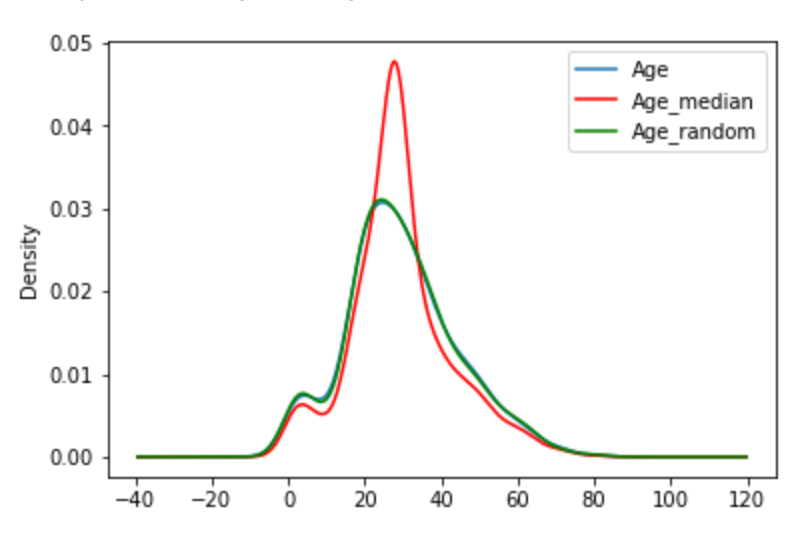
In [147…
```
df.head()
```

Out[147…

	Survived	Age	Fare	Age_median	Age_random
0	0	22.0	7.2500	22.0	22.0
1	1	38.0	71.2833	38.0	38.0
2	1	26.0	7.9250	26.0	26.0
3	1	35.0	53.1000	35.0	35.0
4	0	35.0	8.0500	35.0	35.0

In [148…
```
fig = plt.figure()
ax = fig.add_subplot(111)
df['Age'].plot(kind='kde', ax=ax)
df.Age_median.plot(kind='kde', ax=ax, color='red')
df.Age_random.plot(kind='kde', ax=ax, color='green')
lines, labels = ax.get_legend_handles_labels()
ax.legend(lines, labels, loc='best')
```

Out[148…
```
<matplotlib.legend.Legend at 0x11eca511070>
```



In [149…
```
df=pd.read_csv('Titanic_data.csv',usecols=['Age','Fare','Survived'])
df[:5]
```

Out[149…

	Survived	Age	Fare
0	0	22.0	7.2500
1	1	38.0	71.2833
2	1	26.0	7.9250
3	1	35.0	53.1000
4	0	35.0	8.0500

In [150…
```
df['Age_new']=np.where(df['Age'].isnull(),1,0)
```

In [151…
```
df[:10]
```

Out[151…

	Survived	Age	Fare	Age_new
0	0	22.0	7.2500	0
1	1	38.0	71.2833	0
2	1	26.0	7.9250	0
3	1	35.0	53.1000	0
4	0	35.0	8.0500	0
5	0	NaN	8.4583	1
6	0	54.0	51.8625	0
7	0	2.0	21.0750	0
8	1	27.0	11.1333	0
9	1	14.0	30.0708	0

In [152…
```
df["Age"].median()
```

Out[152…
```
28.0
```

In [153…
```
df["Age"].fillna(df["Age"].median(),inplace=True)
```

In [154…
```
df[:10]
```

Out[154…

	Survived	Age	Fare	Age_new
0	0	22.0	7.2500	0
1	1	38.0	71.2833	0
2	1	26.0	7.9250	0
3	1	35.0	53.1000	0
4	0	35.0	8.0500	0
5	0	28.0	8.4583	1
6	0	54.0	51.8625	0
7	0	2.0	21.0750	0
8	1	27.0	11.1333	0
9	1	14.0	30.0708	0

Advantages

1.Easy to implement 2.Captures the importance of missing values

Disadvantages

1. Creating Additional Features(Curse of Dimensionality)

End of Distribution imputation

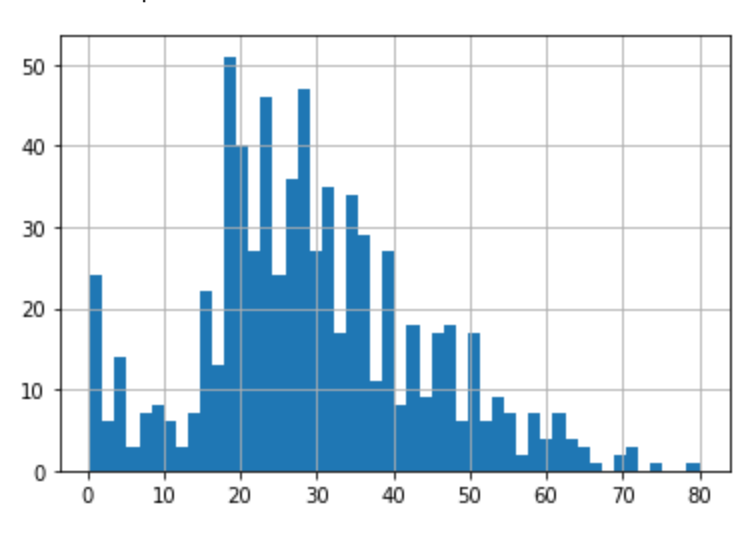
In [155…
```
df=pd.read_csv('Titanic_data.csv',usecols=['Age','Fare','Survived'])
df[:5]
```

Out[155…

	Survived	Age	Fare
0	0	22.0	7.2500
1	1	38.0	71.2833
2	1	26.0	7.9250
3	1	35.0	53.1000
4	0	35.0	8.0500

In [156…
```
df.Age.hist(bins=50)
```

Out[156…
```
<AxesSubplot:>
```

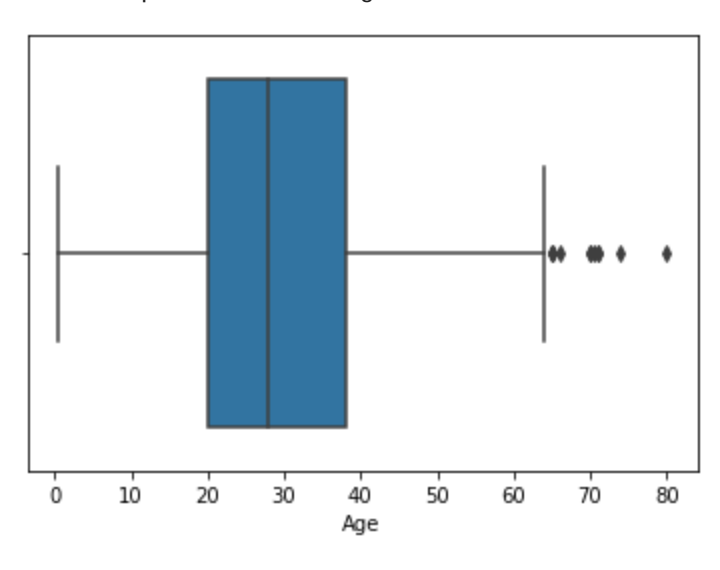


In [157…
```
extreme=df.Age.mean()+3*df.Age.std()
```

In [158…
```
import seaborn as sea
sea.boxplot('Age',data=df)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[158…
```
<AxesSubplot:xlabel='Age'>
```



In [159…
```
def impute_nan(df,variable,median,extreme):
    df['variable_extrm_val'] =df[variable].fillna(extreme)
    df[variable].fillna(median,inplace=True)
```

In [160…
```
impute_nan(df,'Age',df.Age.median(),extreme)
```

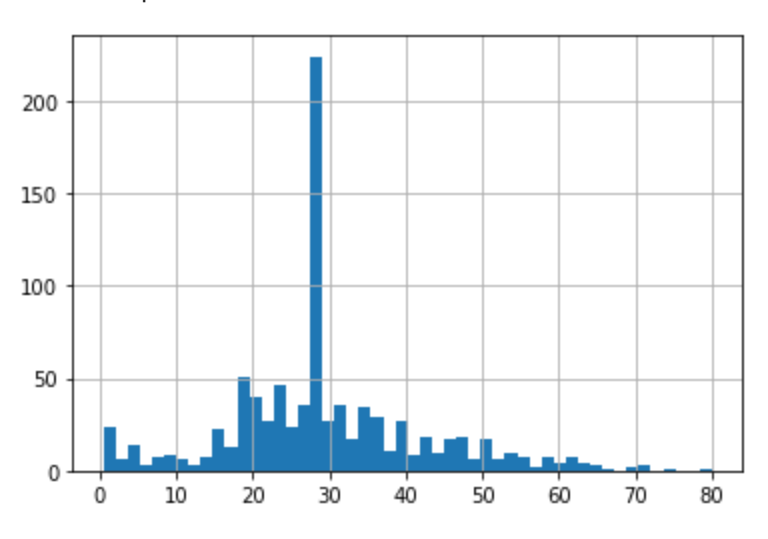
In [161…
```
df.head(10)
```

Out[161…

	Survived	Age	Fare	variable_extrm_val
0	0	22.0	7.2500	22.00000
1	1	38.0	71.2833	38.00000
2	1	26.0	7.9250	26.00000
3	1	35.0	53.1000	35.00000
4	0	35.0	8.0500	35.00000
5	0	28.0	8.4583	73.27861
6	0	54.0	51.8625	54.00000
7	0	2.0	21.0750	2.00000
8	1	27.0	11.1333	27.00000
9	1	14.0	30.0708	14.00000

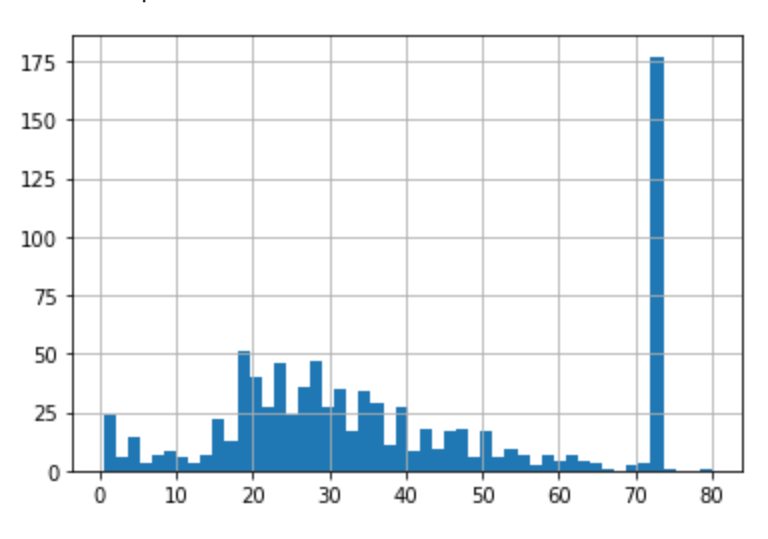
In [162…
```
df['Age'].hist(bins=50)
```

Out[162…
```
<AxesSubplot:>
```



In [163…
```
df['variable_extrm_val'].hist(bins=50)
```

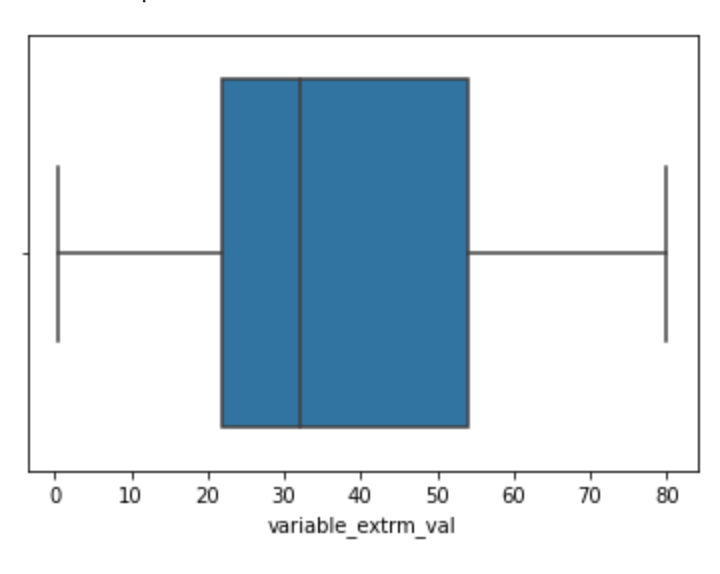
Out[163…
```
<AxesSubplot:>
```



In [164…
```
sea.boxplot('variable_extrm_val',data=df)
```

```
C:\Users\Admin\anaconda3\lib\site-packages\seaborn\_decorators.py:36: FutureWarning: Pass the following variable as a keyword arg: x. From version 0.12, the only valid positional a
rgument will be `data`, and passing other arguments without an explicit keyword will result in an error or misinterpretation.
  warnings.warn(
```

Out[164…
```
<AxesSubplot:xlabel='variable_extrm_val'>
```



Arbitrary Value Imputation

this technique was derived from kaggle competition It consists of replacing NAN by an arbitrary value

In [165…
```
df=pd.read_csv('Titanic_data.csv',usecols=['Age','Fare','Survived'])
df[:5]
```

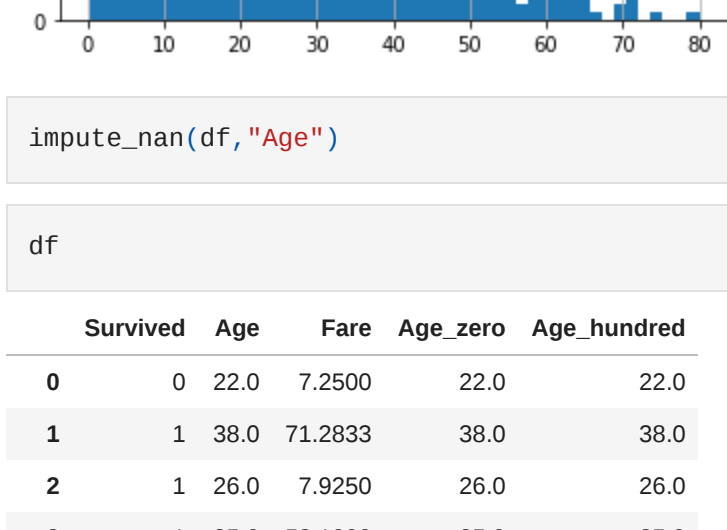
Out[165…

	Survived	Age	Fare
0	0	22.0	7.2500
1	1	38.0	71.2833
2	1	26.0	7.9250
3	1	35.0	53.1000
4	0	35.0	8.0500

In [166…
```
def impute_nan(df,variable):
    df[variable+'_zero']=df[variable].fillna(0)
    df[variable+'_hundred']=df[variable].fillna(100)
```

In [167…
```
df['Age'].hist(bins=50)
```

Out[167…
```
<AxesSubplot:>
```



In [168…
```
impute_nan(df,"Age")
```

In [169…
```
df
```

Out[169…

	Survived	Age	Fare	Age_zero	Age_hundred
0	0	22.0	7.2500	22.0	22.0
1	1	38.0	71.2833	38.0	38.0
2	1	26.0	7.9250	26.0	26.0
3	1	35.0	53.1000	35.0	35.0
4	0	35.0	8.0500	35.0	35.0
...	...	...	...	...	...
886	0	27.0	13.0000	27.0	27.0
887	1	19.0	30.0000	19.0	19.0
888	0	NaN	23.4500	0.0	100.0
889	1	26.0	30.0000	26.0	26.0
890	0	32.0	7.7500	32.0	32.0

891 rows × 5 columns

In [ ]: