Day 11 | Mean & Median imputation

Import Libraries

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

Import sklearn Libraries

In [3]:
```
from sklearn.model_selection import train_test_split
from sklearn.impute import SimpleImputer
from sklearn.compose import ColumnTransformer
```

Import Dataset

In [4]:
```
df = pd.read_csv('titanic_toy.csv')
```

In [10]:
```
df
```

Out[10]:

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

891 rows × 4 columns

In [9]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 4 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   Age       714 non-null    float64
 1   Fare      846 non-null    float64
 2   Family    891 non-null    int64  
 3   Survived  891 non-null    int64  
dtypes: float64(2), int64(2)
memory usage: 28.0 KB
```

Perform Train Test Split

In [11]:
```
X = df.drop(columns=['Survived'])
y = df['Survived']
```

In [12]:
```
X_train,X_test,y_train,y_test = train_test_split(X,y,test_size=0.2,random_state=2)
```

In [13]:
```
X_train.shape, X_test.shape
```

Out[13]:
```
((712, 3), (179, 3))
```

In [14]:
```
X_train.isnull().mean()
```

Out[14]:
```
Age       0.207865
Fare      0.050562
Family    0.000000
dtype: float64
```

Calculate Mean and Median (Age | Fare)

In [15]:
```
mean_age = X_train['Age'].mean()
median_age = X_train['Age'].median()
mean_fare = X_train['Fare'].mean()
median_fare = X_train['Fare'].median()
```

Create new column & impute missing values

In [17]:
```
X_train['Age_median'] = X_train['Age'].fillna(median_age)
X_train['Age_mean'] = X_train['Age'].fillna(mean_age)
X_train['Fare_median'] = X_train['Fare'].fillna(median_fare)
X_train['Fare_mean'] = X_train['Fare'].fillna(mean_fare
```

In [18]:
```
X_train.sample(8)
```

Out[18]:

	Age	Fare	Family	Age_median	Age_mean	Fare_median	Fare_mean
255	29.0	15.2458	2	29.00	29.000000	15.2458	15.245800
420	NaN	7.8958	0	28.75	29.785904	7.8958	7.895800
673	31.0	NaN	0	31.00	31.000000	14.4583	32.617597
761	41.0	7.1250	0	41.00	41.000000	7.1250	7.125000
82	NaN	NaN	0	28.75	29.785904	14.4583	32.617597
269	35.0	135.6333	0	35.00	35.000000	135.6333	135.633300
254	41.0	20.2125	2	41.00	41.000000	20.2125	20.212500
440	45.0	26.2500	2	45.00	45.000000	26.2500	26.250000

Review Variance

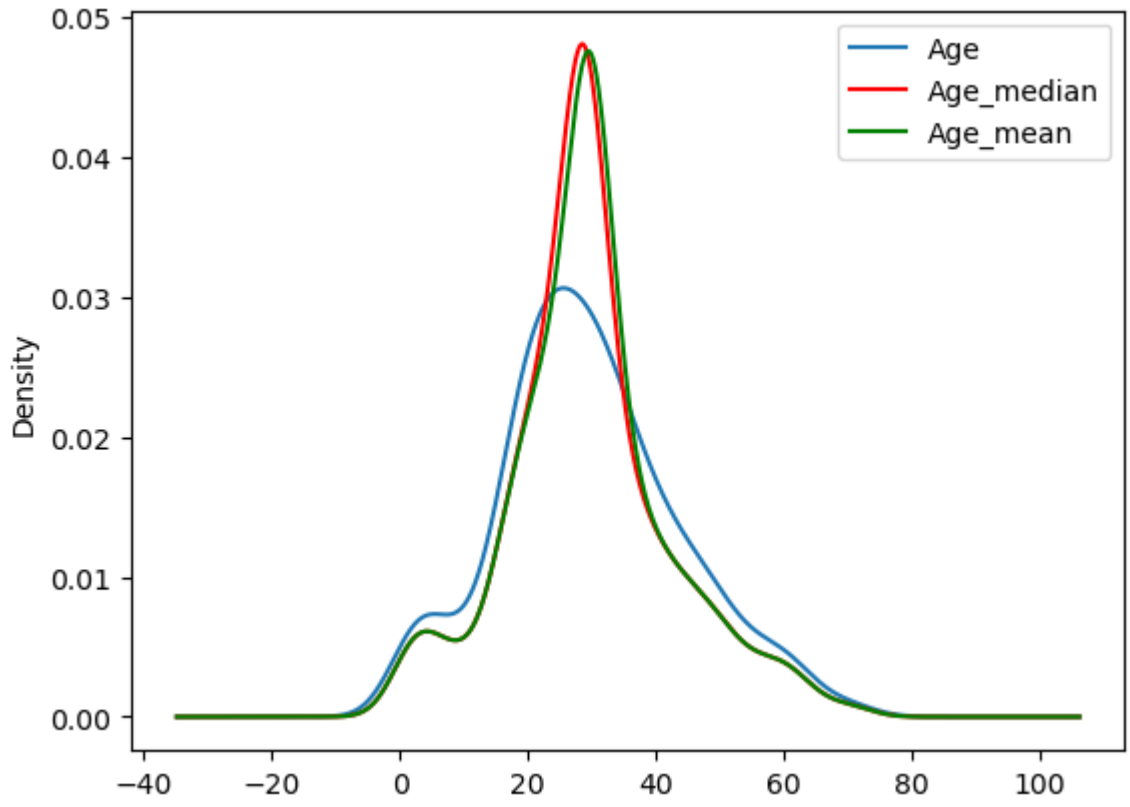
In [19]:
```
print('Original Age variable variance: ', X_train['Age'].var())
print('Age Variance after median imputation: ', X_train['Age_median'].var())
print('Age Variance after mean imputation: ', X_train['Age_mean'].var())

print('Original Fare variable variance: ', X_train['Fare'].var())
print('Fare Variance after median imputation: ', X_train['Fare_median'].var())
print('Fare Variance after mean imputation: ', X_train['Fare_mean'].var())
```

```
Original Age variable variance:  204.3495133904614
Age Variance after median imputation:  161.9895663346054
Age Variance after mean imputation:  161.81262452718673
Original Fare variable variance:  2448.197913706318
Fare Variance after median imputation:  2340.0910219753637
Fare Variance after mean imputation:  2324.2385256705547
```

Changes in Distribution in Age

In [20]:
```
fig = plt.figure()
ax = fig.add_subplot(111)
# original variable distribution
X_train['Age'].plot(kind='kde', ax=ax)
# variable imputed with the median
X_train['Age_median'].plot(kind='kde', ax=ax, color='red')
# variable imputed with the mean
X_train['Age_mean'].plot(kind='kde', ax=ax, color='green')
# add legends
lines, labels = ax.get_legend_handles_labels()
ax.legend(lines, labels, loc='best')
```

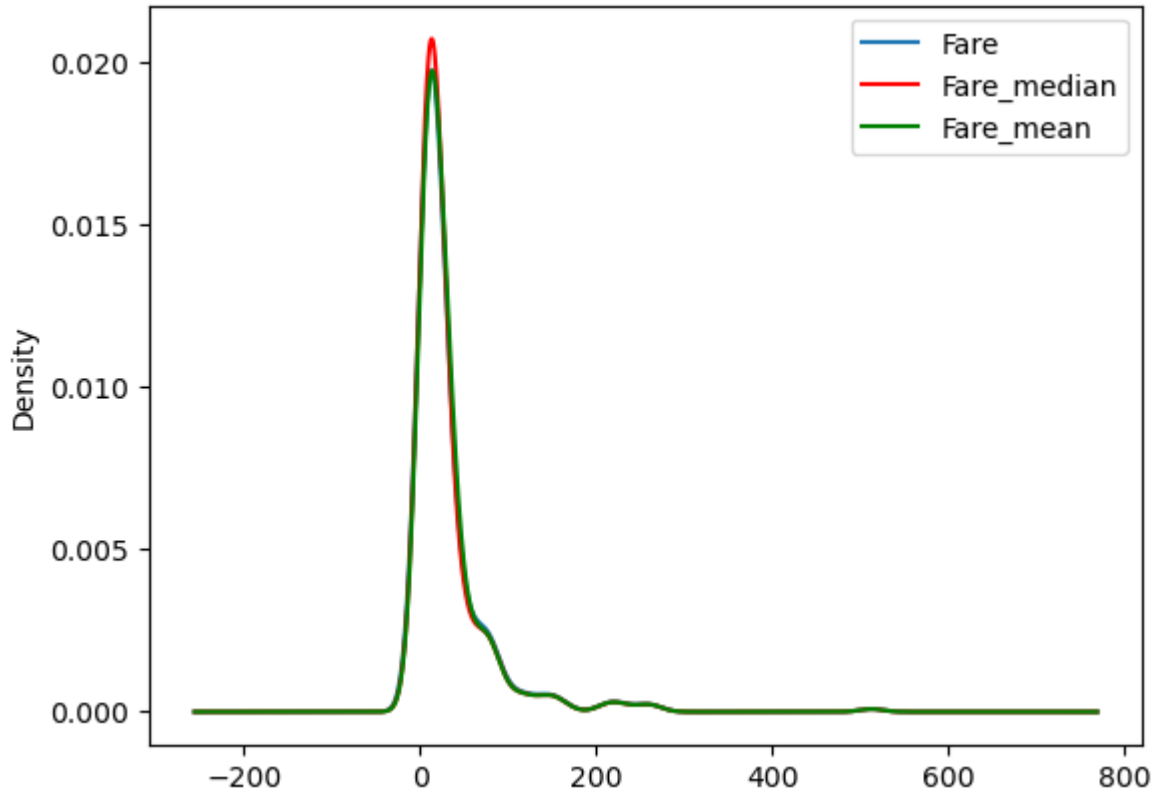
Out[20]: <matplotlib.legend.Legend at 0x2bca7545e90>



Changes in Distribution in Fare

In [21]:
```
fig = plt.figure()
ax = fig.add_subplot(111)
# original variable distribution
X_train['Fare'].plot(kind='kde', ax=ax)
# variable imputed with the median

X_train['Fare_median'].plot(kind='kde', ax=ax, color='red')
# variable imputed with the mean
X_train['Fare_mean'].plot(kind='kde', ax=ax, color='green')
# add legends
lines, labels = ax.get_legend_handles_labels()
ax.legend(lines, labels, loc='best')
```

Out[21]: <matplotlib.legend.Legend at 0x2bca45f4490>



Check Covariance

In [22]:
```
X_train.cov()
```

Out[22]:

	Age	Fare	Family	Age_median	Age_mean	Fare_median	Fare_mean
Age	204.349513	70.719262	-6.498901	204.349513	204.349513	64.858859	66.665205
Fare	70.719262	2448.197914	17.258917	57.957599	55.603719	2448.197914	2448.197914
Family	-6.498901	17.258917	2.735252	-5.112563	-5.146106	16.476305	16.385048
Age_median	204.349513	57.957599	-5.112563	161.989566	161.812625	53.553455	55.023037
Age_mean	204.349513	55.603719	-5.146106	161.812625	161.812625	51.358000	52.788341
Fare_median	64.858859	2448.197914	16.476305	53.553455	51.358000	2340.091022	2324.238526
Fare_mean	66.665205	2448.197914	16.385048	55.023037	52.788341	2324.238526	2324.238526

Check Correlation

In [24]:
```
X_train.corr()
```

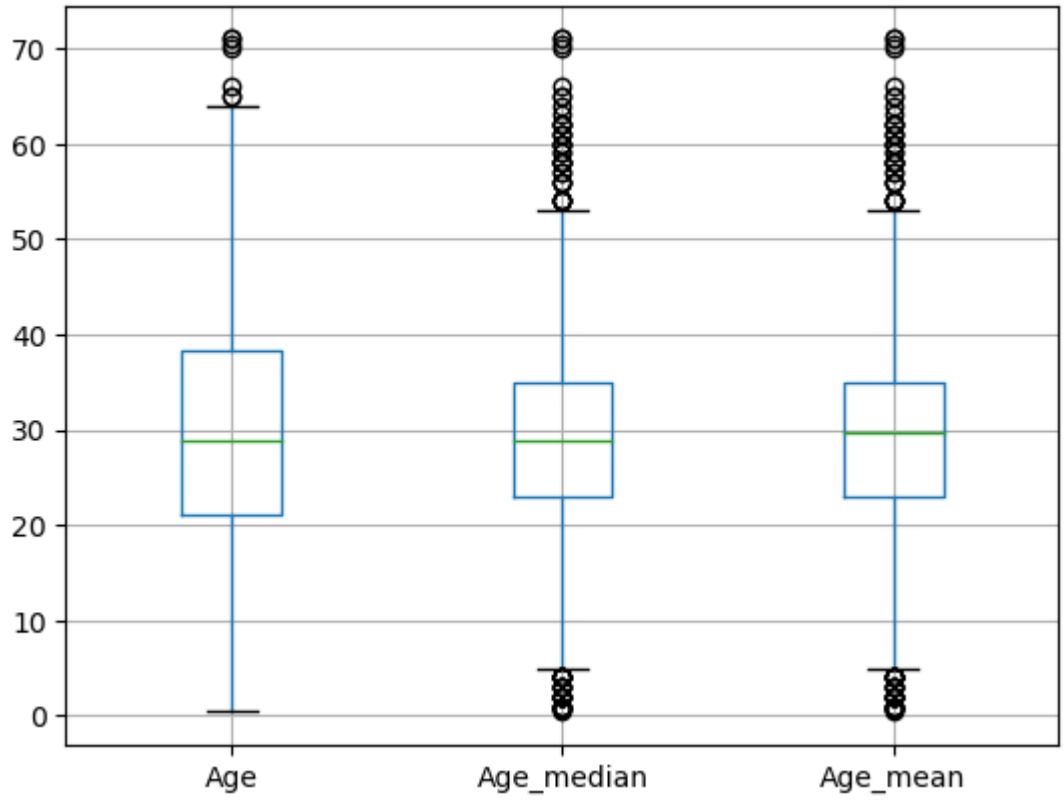
Out[24]:

	Age	Fare	Family	Age_median	Age_mean	Fare_median	Fare_mean
Age	1.000000	0.092644	-0.299113	1.000000	1.000000	0.087356	0.090156
Fare	0.092644	1.000000	0.208268	0.091757	0.088069	1.000000	1.000000
Family	-0.299113	0.208268	1.000000	-0.242883	-0.244610	0.205942	0.205499
Age_median	1.000000	0.091757	-0.242883	1.000000	0.999454	0.086982	0.089673
Age_mean	1.000000	0.088069	-0.244610	0.999454	1.000000	0.083461	0.086078
Fare_median	0.087356	1.000000	0.205942	0.086982	0.083461	1.000000	0.996607
Fare_mean	0.090156	1.000000	0.205499	0.089673	0.086078	0.996607	1.000000

Box Plot for Age

In [25]:
```
X_train[['Age', 'Age_median',
'Age_mean']].boxplot()
```

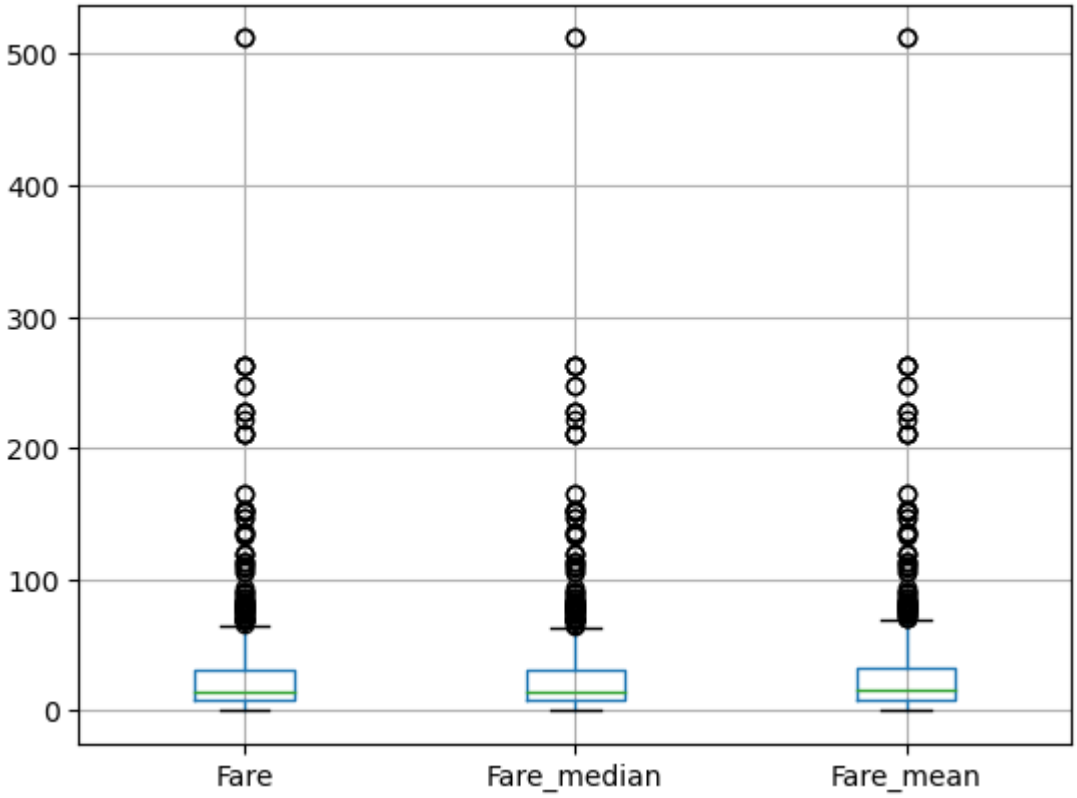
Out[25]: <Axes: >



Box Plot for Fare

In [26]:
```
X_train[['Fare', 'Fare_median', 'Fare_mean']].boxplot()
```

Out[26]: <Axes: >



In [ ]: