handle the missing values

December 20, 2023

[1]:
```
##data manupulation
import pandas as pd
## data visuali
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
```

[ ]:
```
## life cycle of data science projects
##data collection strategy-- from company side 3rd party side survey,survey
## data should collecting from multiple sources
```

[16]:
```
df = pd.read_csv("C:/Users/Vikas/Downloads/titanic.csv")
df.head(2)
```

[16]:
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1

                                                 Name     Sex   Age  SibSp  \
0                             Braund, Mr. Owen Harris    male  22.0      1
1  Cumings, Mrs. John Bradley (Florence Briggs Th…  female  38.0      1

   Parch     Ticket     Fare Cabin Embarked
0      0  A/5 21171   7.2500   NaN        S
1      0   PC 17599  71.2833   C85        C
```

[17]:
```
df.columns
```

[17]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

[18]:
```
df.shape
```

[18]:
```
(891, 12)
```

[19]:
```
df.describe()
```

[19]:
```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

            Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

[20]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

[21]:
```
df.isnull().sum()
```

[21]:
```
PassengerId      0
Survived         0
Pclass           0
Name             0
```

```
Sex              0
Age            177
SibSp            0
Parch            0
Ticket           0
Fare             0
Cabin          687
Embarked         2
dtype: int64
```

[22]:
```
## missing complety at random
df[df['Embarked'].isnull()]
```

[22]:
```
     PassengerId  Survived  Pclass                                          Name  \
61            62         1       1                        Icard, Miss. Amelie
829          830         1       1  Stone, Mrs. George Nelson (Martha Evelyn)

        Sex   Age  SibSp  Parch  Ticket  Fare Cabin Embarked
61   female  38.0      0      0  113572  80.0   B28      NaN
829  female  62.0      0      0  113572  80.0   B28      NaN
```

[26]:
```
## missing data not at random(mnar): systematic missing values
## inside the cabin column missing values is yes1 otherwise is no 0
df['cabin_null']=np.where(df['Cabin'].isnull(),1,0)

## find the percentage of null values
df['cabin_null'].mean()
```

[26]:
```
0.7710437710437711
```

[31]:
```
df.columns
```

[31]:
```
Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp',
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked', 'cabin_null'],
      dtype='object')
```

[32]:
```
## how much people survived missing values percentage
df.groupby(['Survived'])['cabin_null'].mean()
```

[32]:
```
Survived
0    0.876138
1    0.602339
Name: cabin_null, dtype: float64
```

[ ]:
```
## missing at random(MAR)
Men---hide there salary
women--- hide there age
```

[ ]:
```
## all the technic of handling of missing values
## mean ,median, mode replacement
## random sample imputation
## capturing nan value with new feature
## end of destribution imputation
## orbitrary impution
## frequent category imputation
##
```

1 mean ,median, mode replacement

when should we apply

[33]:
```
df = pd.read_csv('C:/Users/Vikas/Downloads/titanic.
 ↪csv',usecols=['Age','Fare','Survived'])
df.head()
```

[33]:

	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

[34]:
```
## lets go and see percentage of missing values
df.isnull().mean()
```

[34]:
```
Survived    0.000000
Age         0.198653
Fare        0.000000
dtype: float64
```

[35]:
```
def impute_nan(df,variable,median):
    df[variable+"_median"]=df[variable].fillna(median)
```

[36]:
```
median = df.Age.median()
median
```

[36]:
```
28.0
```

[37]:
```
impute_nan(df,'Age',median)
df.head()
```

[37]:

	Survived	Age	Fare	Age_median
0	0	22.0	7.2500	22.0
1	1	38.0	71.2833	38.0
2	1	26.0	7.9250	26.0

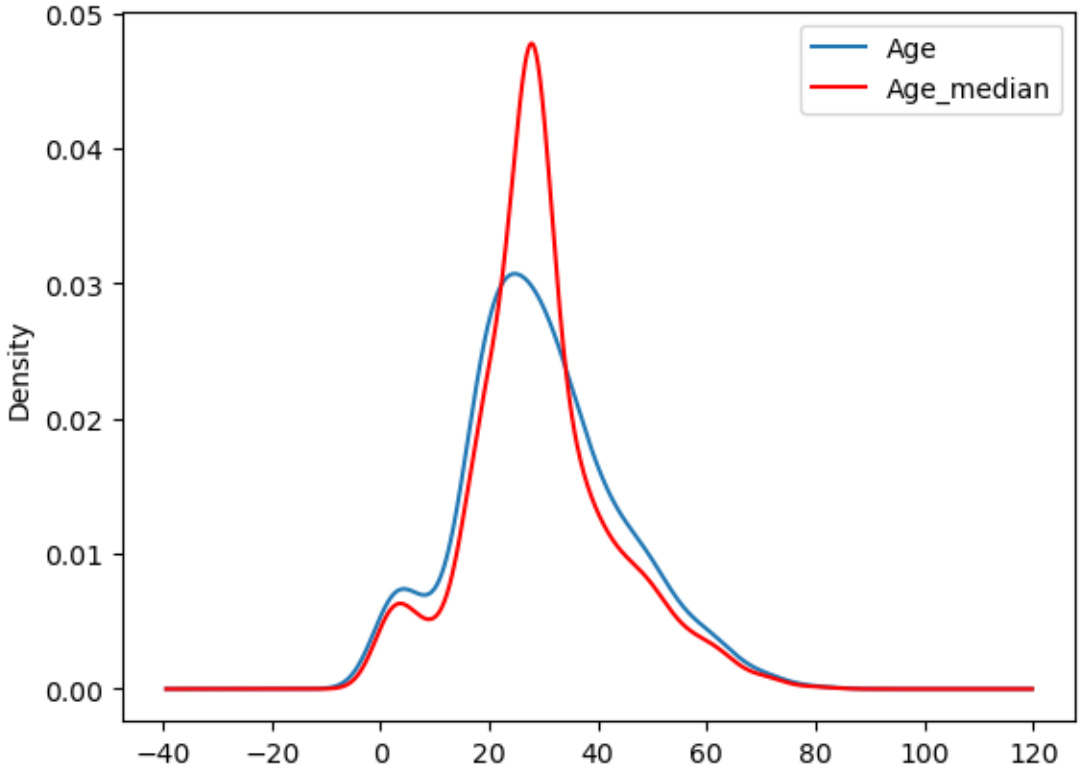
```
3          1  35.0  53.1000        35.0
4          0  35.0   8.0500        35.0
```

[38]:
```
print(df['Age'].std())
print(df['Age_median'].std())
```

```
14.526497332334044
13.019696550973194
```

[41]:
```
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')
lines,labels = ax.get_legend_handles_labels()
ax.legend(lines,labels,loc='best')
```

[41]: <matplotlib.legend.Legend at 0x15b5e613100>



[ ]: