avib013/College/blob/main/Data_mining/titanic.ipynb

me 1252 lines (1252 loc) · 118 KB Code 55% faster with GitHub Copilot Raw

[1]:
```
import pandas as pd
import numpy as np
```

[6]:
```
df=pd.read_csv('test.csv')
```

[7]:
```
df
```

[7]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

College / Data_mining / **titanic.ipynb**

Blame 1252 lines (1252 loc) · 118 KB Code 55% faster with GitHub Copilot

In [13]:
```
df.head()
```

Out[13]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S

In [8]:
```
df.columns
```

Out[8]:
```
Index(['PassengerId', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch',
       'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

Data Understanding

In [9]:
```
df.shape
```

Out[9]:
```
(418, 11)
```

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
```

In [12]:
```
df.describe()
```

Out[12]:

	PassengerId	Pclass	Age	SibSp	Parch	Fare
count	418.000000	418.000000	332.000000	418.000000	418.000000	417.000000
mean	1100.500000	2.265550	30.272590	0.447368	0.392344	35.627188
std	120.810458	0.841838	14.181209	0.896760	0.981429	55.907576
min	892.000000	1.000000	0.170000	0.000000	0.000000	0.000000
25%	996.250000	1.000000	21.000000	0.000000	0.000000	7.895800
50%	1100.500000	3.000000	27.000000	0.000000	0.000000	14.454200
75%	1204.750000	3.000000	39.000000	1.000000	0.000000	31.500000
max	1309.000000	3.000000	76.000000	8.000000	9.000000	512.329200

In [23]:
```
df.isna().mean() *100#here we get the total in percentage
```

Out[23]:
```
PassengerId     0.000000
Pclass          0.000000
Name            0.000000
Sex             0.000000
Age            20.574163
SibSp           0.000000
Parch           0.000000
Ticket          0.000000
Fare            0.239234
Cabin          78.229665
Embarked        0.000000
dtype: float64
```

30]:
```
df.duplicated().sum()
```

30]: 0

34]:
```
df=df.drop_duplicates()
df
```

34]:

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	Q
1	893	3	Wilkes, Mrs. James (Ellen Needs)	female	47.0	1	0	363272	7.0000	NaN	S
2	894	2	Myles, Mr. Thomas Francis	male	62.0	0	0	240276	9.6875	NaN	Q
3	895	3	Wirz, Mr. Albert	male	27.0	0	0	315154	8.6625	NaN	S
4	896	3	Hirvonen, Mrs. Alexander (Helga E Lindqvist)	female	22.0	1	1	3101298	12.2875	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...
413	1305	3	Spector, Mr. Woolf	male	NaN	0	0	A.5. 3236	8.0500	NaN	S
414	1306	1	Oliva y Ocana, Dona. Fermina	female	39.0	0	0	PC 17758	108.9000	C105	C
415	1307	3	Saether, Mr. Simon Sivertsen	male	38.5	0	0	SOTON/O.Q. 3101262	7.2500	NaN	S
416	1308	3	Ware, Mr. Frederick	male	NaN	0	0	359309	8.0500	NaN	S
417	1309	3	Peter, Master. Michael J	male	NaN	1	1	2668	22.3583	NaN	C

418 rows × 11 columns

EDA

[38]:
```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 418 entries, 0 to 417
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype  
---  ------       --------------  -----  
 0   PassengerId  418 non-null    int64  
 1   Pclass       418 non-null    int64  
 2   Name         418 non-null    object 
 3   Sex          418 non-null    object 
 4   Age          332 non-null    float64
 5   SibSp        418 non-null    int64  
 6   Parch        418 non-null    int64  
 7   Ticket       418 non-null    object 
 8   Fare         417 non-null    float64
 9   Cabin        91 non-null     object 
 10  Embarked     418 non-null    object 
dtypes: float64(2), int64(4), object(5)
memory usage: 36.1+ KB
```

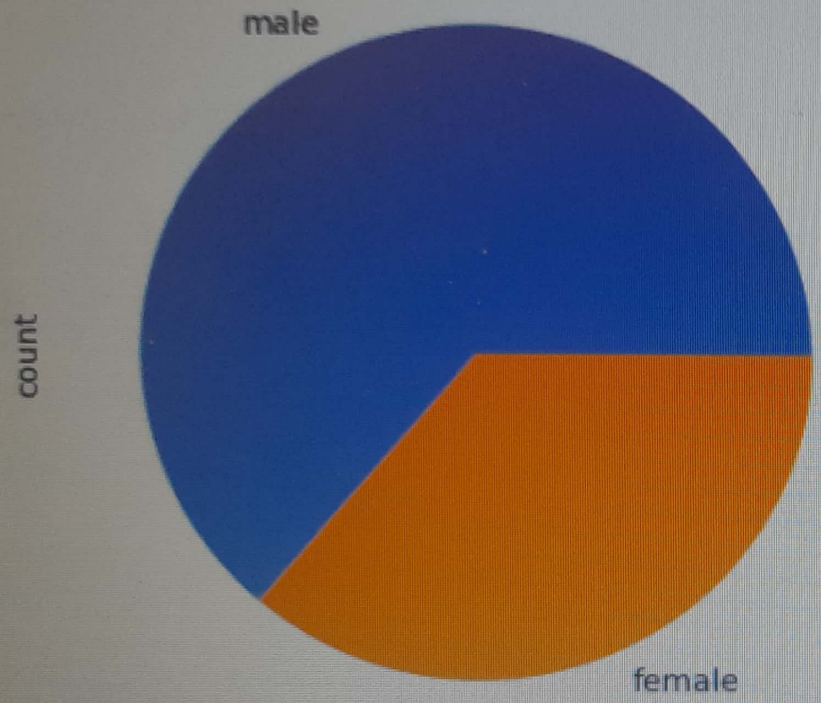
In [41]:
```
set(df['Sex']) #categorical data
```

Out[41]:
```
{'female', 'male'}
```

In [43]:
```
df['Sex'].value_counts().plot(kind='pie')
```

Out[43]:
```
<Axes: ylabel='count'>
```

ut[43]: <Axes: ylabel='count'>



[44]: set(df['Pclass'])

ut[44]: {1, 2, 3}

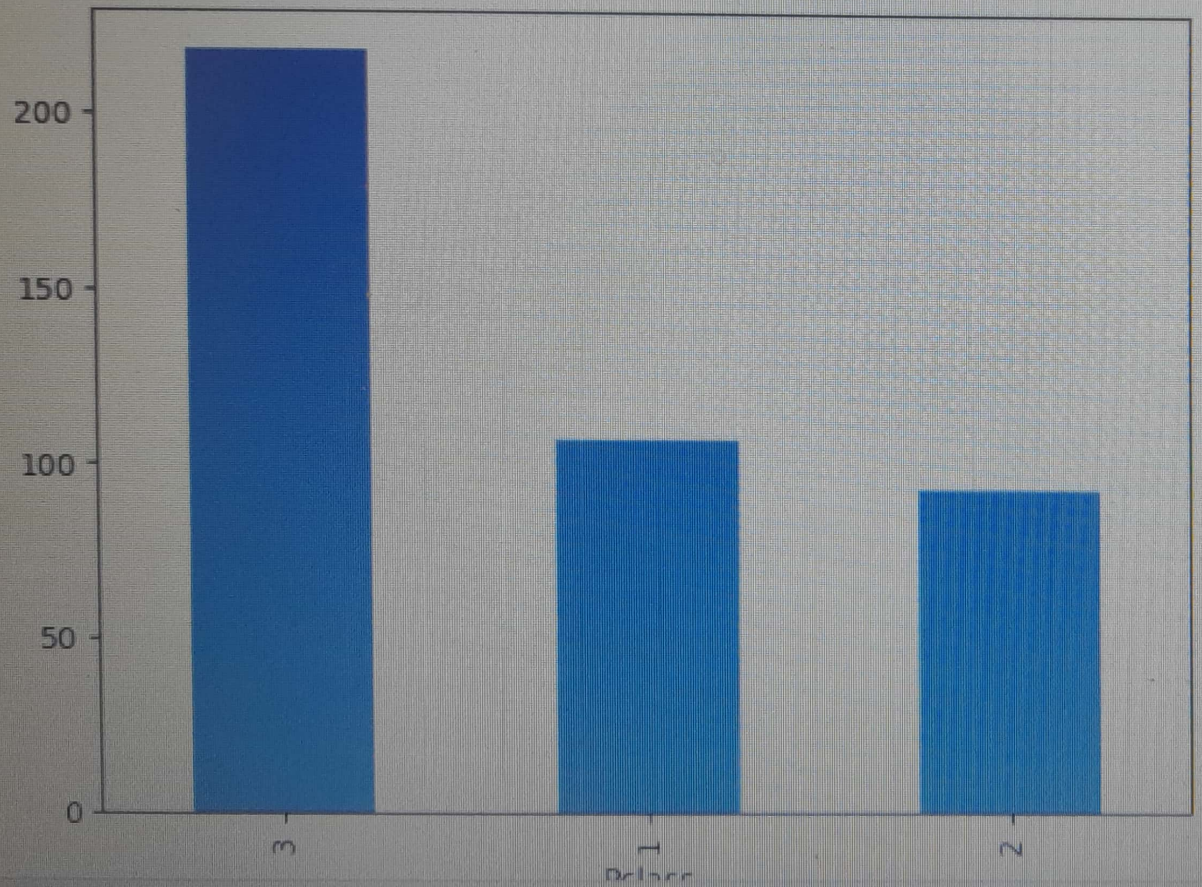
[28]: import seaborn as sns
import matplotlib.pyplot as plt

Out[44]: {1, 2, 3}

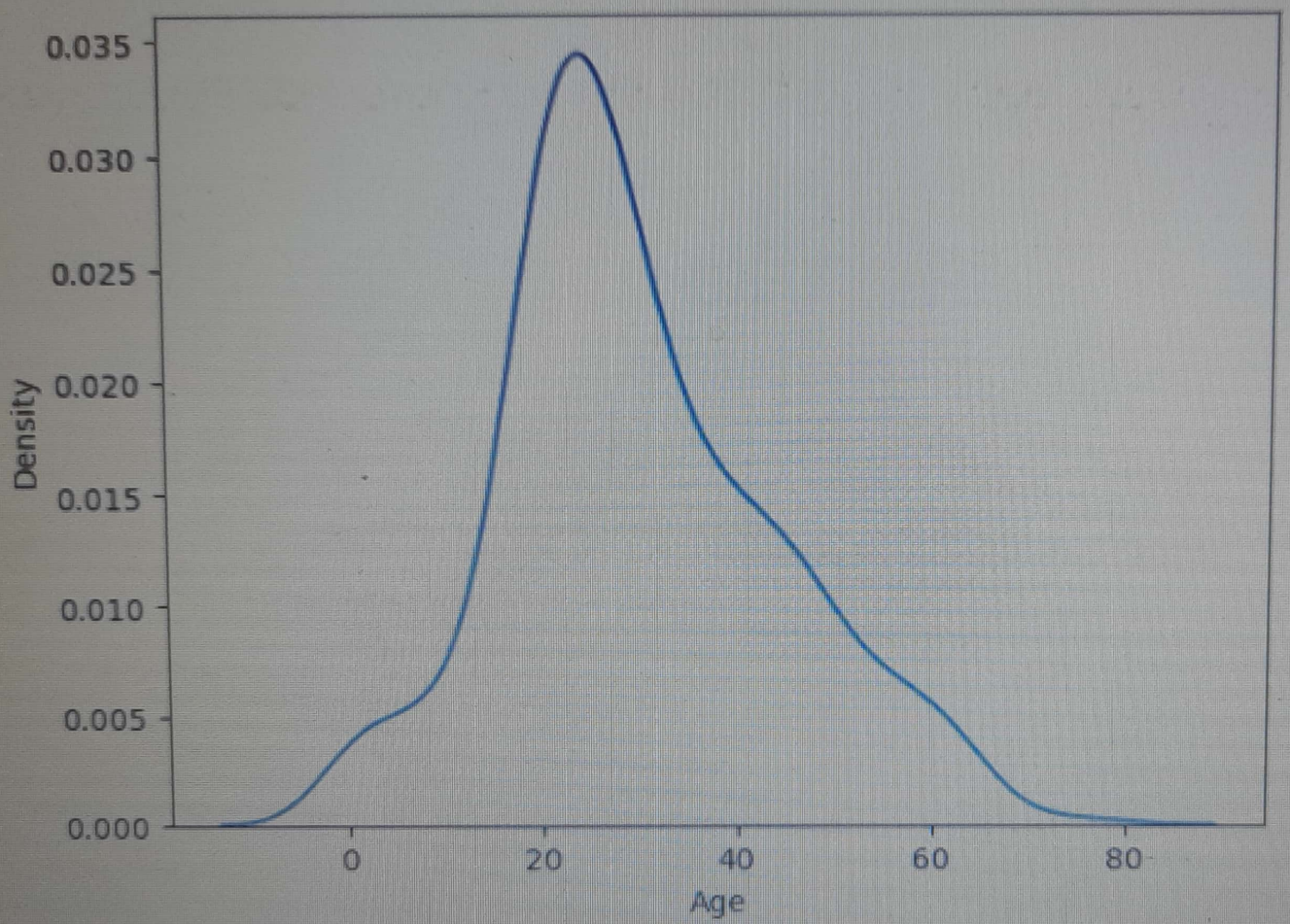
In [28]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

In [50]:
```
df['Pclass'].value_counts().plot(kind='bar')
```

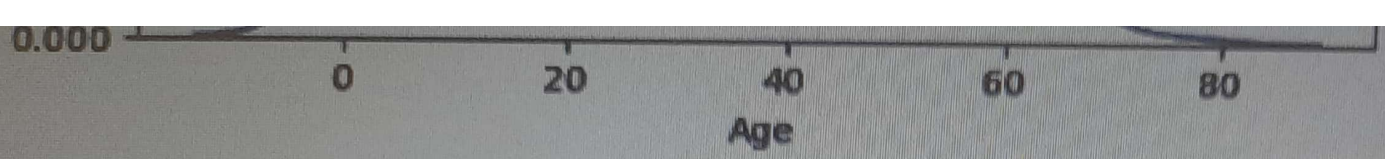
Out[50]: <Axes: xlabel='Pclass'>



Out[52]: <Axes: xlabel='Age', ylabel='Density'>



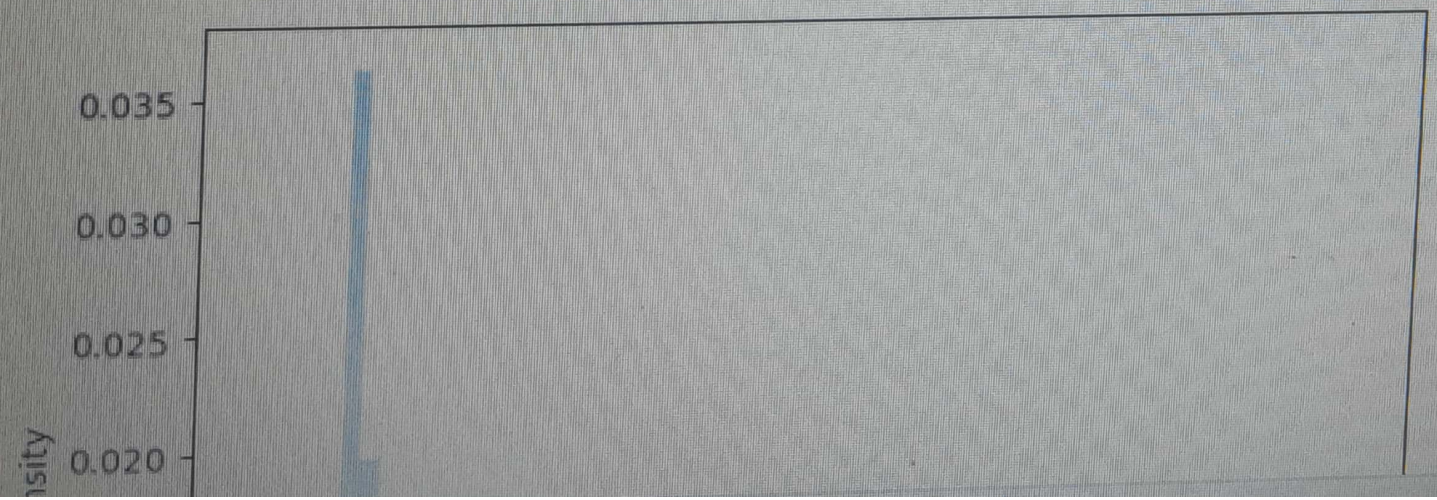
In [49]: sns.distplot(df['Fare'])



49]:
```
sns.distplot(df['Fare'])
```

```
C:\Users\Vaishnavi\AppData\Local\Temp\ipykernel_11928\3425841524.py:1: UserWarning:

`distplot` is a deprecated function and will be removed in seaborn v0.14.0.

Please adapt your code to use either `displot` (a figure-level function with
similar flexibility) or `histplot` (an axes-level function for histograms).

For a guide to updating your code to use the new functions, please see
https://gist.github.com/mwaskom/de44147ed2974457ad6372750bbe5751

  sns.distplot(df['Fare'])
C:\Users\Vaishnavi\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated an
d will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

]: <Axes: xlabel='Fare', ylabel='Density'>

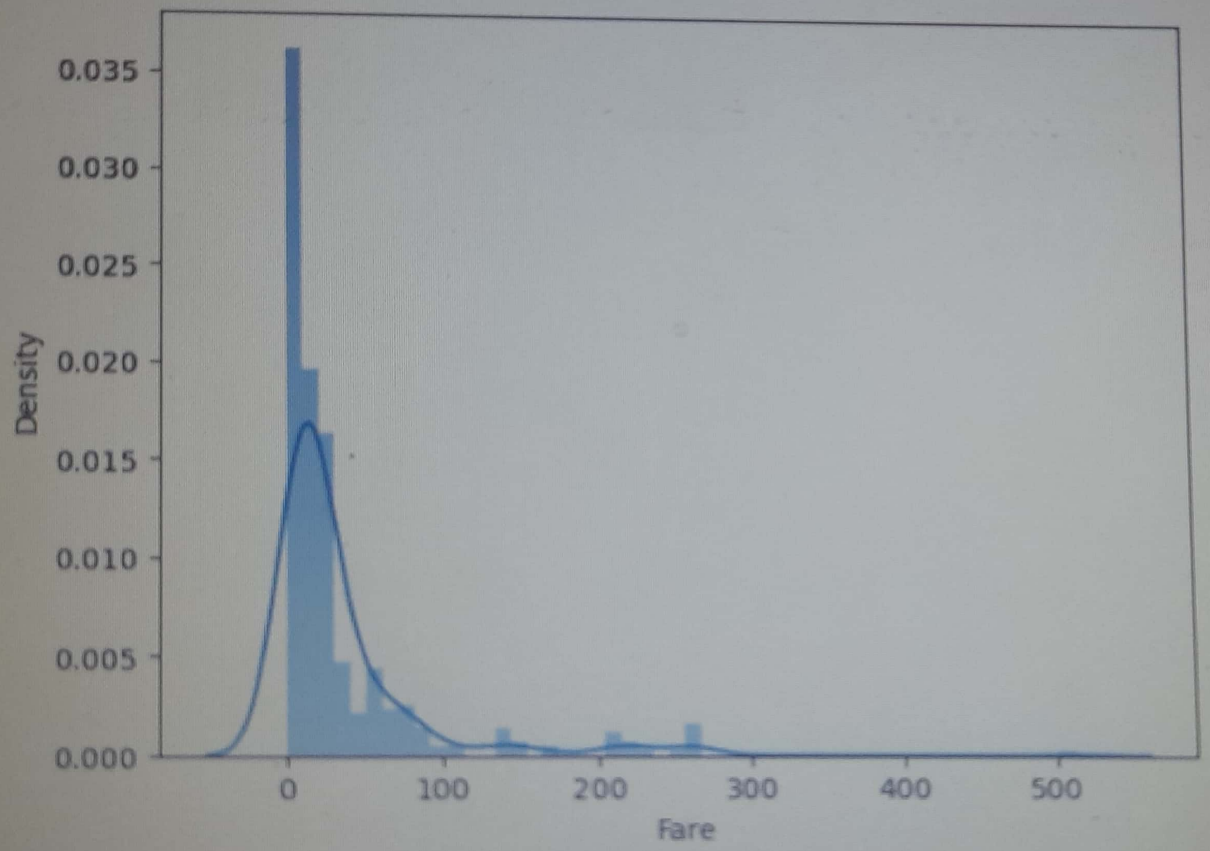


College / Data_mining / titanic.ipynb

Blame 1252 lines (1252 loc) · 118 KB Code 55% faster with GitHub Copilot Raw

```
sns.distplot(df['Fare'])
C:\Users\Vaishnavi\anaconda3\Lib\site-packages\seaborn\_oldcore.py:1119: FutureWarning: use_inf_as_na option is deprecated an
d will be removed in a future version. Convert inf values to NaN before operating instead.
  with pd.option_context('mode.use_inf_as_na', True):
```

Out[49]: <Axes: xlabel='Fare', ylabel='Density'>



Page No.
Date

Title:- Use any suitable dataset & perform following operation on given dataset suitable programming language.

(a) Find missing values & replace missing values with suitable alternative.
(b) Remove inconsistency in dataset.
(c) Prepare boxplot analysis for each numerical attribute. Find outliers in each attribute in dataset.
(d) Draw histogram for any 2 suitable attributes.
(e) Find datatype of each column.
(f) Finding out zeros.
(g) Find mean of any numerical value.
(h) Find shape of data.

* Theory.

(1) What is dataset & type of dataset?

→ (1) Dataset is a collection of related data that is stored in a structured format.
(2) It consists of row (records) & column (features)
(3) Datasets can be used for analysis. ML or data visualize.

(2) What are types of dataset?

→ (1) Record based dataset:- These datasets consist of rows & column. Each row represents an individual datapoint & columns represent features of those datapoints.
eg:- employee details: Name, age, salary.
used in relational databases.

(2) Graph & Network Datasets:-
These attributes model relationships between entities in the form of nodes & edges.
ex- social gathering networks.
Transportation networks: location (nodes) & roads (edges).

(3) Ordered Datasets:- There datasets have a sequence or temporal order associated the datapoints. ex. time series.

(4) Spatial:- Images & multimedia.
Spatial- associated with geographic eg- GPS.
Image- consists of images eg- images.
Multimedia- contains multiple type of media
ex- Speech datasets.

(3) What is data objects?
→ (1) Datasets are made up of dataobjects.
(2) Data objects represents an entity.
(3) Also known as samples, eg, instances, datapoints, objects & tuples.
It is described by objects attributes.

(4) What are different types of attributes?
→ Attributes a data field, representing characteristic of features of data object. eg:- name, address.
Types:-
1) Nominal - represents name, labels.
2) Binary - Special types of Nomial attribute that has only 2 possible values.

① Symmetric :- both O/P / Outcome equally imp.
② Assymmetric :- Outcomes not equally imp.
eg (medical test (+ve or -ve)
③ Ordinal :- Attributes has meaningful order. eg :- size.
④ Numeric :- numerical values can be counted.
⑤ Discrete :- Numeric attributes that can take distinct values. eg :- zip code.
⑥ Continuous :- Numeric attribute that can take any value within range. eg :- Height.

5) Data preprocessing Operation - Handling missing value
→ ① Identify Missing Values :- Check for blank entries in dataset
② Methods To Handle :-
(A) Remove missing data : Delete rows/columns with missing values if their impact on data is minimal
(B) Input missing Data :- Replacing missing values :- with mean / median / mode.
③ Advanced Techniques : Apply ML methods.

6) Explain about Boxplot & histogram.
① A boxplot is visual summary of data distribution
It is useful for detecting Outliers & understanding data variability. It shows median, quartiles, whiskers & outliers.
② A histogram displays frequency distribution of numeric data. It divides data into bins & show how many values fall into bins.

* Code :-
import pandas as pd.
import numpy as np.

```
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv("data")
df.fillna(df.mean(), inplace=True)
print(df.duplicated().sum())
df.drop_duplicates(inplace=True)
plt.figure(figsize=(6,4))
sns.boxplot(x=df['age'])
plt.title('Boxplot of age')
plt.savefig("Boxplot_age.png")
plt.show()
Q1 = df['age'].quantile(0.25)
Q3 = df['age'].quantile(0.75)
IQR = Q3 - Q1
Outliers = df[(df['age'] < Q1 - 1.5*IQR) | (df['age']
    > Q3 + 1.5*IQR)]
plt.figure(figsize=(6,4))
plt.hist(df['age'], bins=10, color='blue', edgecolor='black')
plt.show()
print(df.types)
zero_count = (df['age'] == 0).sum()
print("Number of zero in age", zero_count)
mean_age = df['age'].mean()
print("Mean of Age", mean_age)
print("shape of dataset", df.shape)
```

Conclusion :- In this assignment we are able to understand about operations of the data handling and data preprocessing.