

Practical 1

Head, describe, dtypes, shape

Practical 2

Shape

df.isnull()

df.isnull().any()

df.isnull().sum()

df['Rollno'].mean()

df['Phy_marks'] = df['Phy_marks'].fillna(df['Phy_marks'].mean())

df['Phy_marks'] = df['Phy_marks'].fillna(df['Phy_marks'].mode())

df['Phy_marks'] = df['Phy_marks'].fillna(df['Phy_marks'].median())

df['Phy_marks'] = df['Phy_marks'].fillna(df['Phy_marks'].std())

col = ['Phy_marks', 'Che_marks', 'EM1_marks', 'PPS_marks']

df.boxplot(col)

sorted_rscores = sorted(df['Rollno'])

sorted_rscores

q1 = np.percentile(sorted_rscores, 25)

q3 = np.percentile(sorted_rscores, 75)

IQR=q3-q1

lwr_bound = q1-(1.5*IQR)

upr_bound = q3+(1.5*IQR)

print(lwr_bound, upr_bound)

• r_outliers = []

for i in sorted_rscores:

if (i < lwr_bound or i > upr_bound):

r_outliers.append(i)

print(r_outliers)

Practical 3

df.head()

df.dtypes

mean, median, mode

```
df[["Annual Income (k$)", "Age"]].groupby("Age").describe()  
df[["Annual Income (k$)", "Age"]].groupby("Age").mean()  
my_list = list()  
my_list.append(df[["Annual Income (k$)", "Age"]].groupby("Age").sum())  
df1[Species = 'Iris-setosa'].quantile()
```

Practical 7

```
sentence1 = "I will walk 500 miles and I would walk 500 more. Just to be the man who walks " + \  
    "a thousand miles!"  
  
sentence2 = "I played the play playfully as the players were playing in the play with playfullness"  
  
from nltk import word_tokenize, sent_tokenize  
  
print('Tokenized words:', word_tokenize(sentence1))  
  
print('\nTokenized sentences:', sent_tokenize(sentence1))  
  
from nltk import pos_tag  
  
token = word_tokenize(sentence1) + word_tokenize(sentence2)  
  
tagged = pos_tag(token)  
  
print("Tagging Parts of Speech:", tagged)  
  
from nltk.corpus import stopwords  
  
stop_words = stopwords.words('english')  
  
token = word_tokenize(sentence1)  
  
cleaned_token = []  
  
for word in token:  
  
    if word not in stop_words:  
  
        cleaned_token.append(word)  
  
print('Unclean version:', token)  
  
print('\nCleaned version:', cleaned_token)  
  
from nltk.stem import PorterStemmer  
  
stemmer = PorterStemmer()  
  
token = word_tokenize(sentence2)  
  
stemmed = [stemmer.stem(word) for word in token]  
  
print(" ".join(stemmed))  
  
from nltk.stem import WordNetLemmatizer
```

```
lemmatizer = WordNetLemmatizer()
token = word_tokenize(sentence2)
lemmatized_output = [lemmatizer.lemmatize(word) for word in token]
print(" ".join(lemmatized_output))
```

Practical 8

```
dataset.describe()
sns.histplot(dataset['fare'], kde=True, linewidth=0);
sns.jointplot(x='age', y='fare', data=dataset);
sns.boxplot(x='sex', y='age', data=dataset, hue="survived");
```

Practical 9

```
sns.boxplot(x='sex', y='age', data=dataset, hue="survived");
```

Practical 10

```
df.head()
fig, axes = plt.subplots(2, 2, figsize=(16, 8))
axes[0,0].set_title("Distribution of First Column")
axes[0,0].hist(df["SepalLengthCm"]);
axes[0,1].set_title("Distribution of Second Column")
axes[0,1].hist(df["SepalWidthCm"]);
axes[1,0].set_title("Distribution of Third Column")
axes[1,0].hist(df["PetalLengthCm"]);
axes[1,1].set_title("Distribution of Fourth Column")
axes[1,1].hist(df["PetalWidthCm"]);
data_to_plot = df[df.columns[1:-1]]
fig, axes = plt.subplots(1, figsize=(12,8))
bp = axes.boxplot(data_to_plot)
```