

```

import seaborn as sns
import pandas as pd
import numpy as np
import scipy.stats as st
import matplotlib.pyplot as plt
from scipy.stats import gmean

###-----Q1-----
print("-----Q1-----")
datasets= sns.get_dataset_names()
print("Datasets in seaborn package: ")
print(datasets)

###-----Q2-----
print("\n\n-----Q2-----")
dataset_list=['diamonds','iris','tips','penguins','titanic']
for i in dataset_list:
    df=sns.load_dataset(i)
    print(f"Number of observations in {i}: ",df.shape[0])
    print("\n")

###-----Q3-----
print("\n\n-----Q3-----")
df=sns.load_dataset('titanic')
df_summary= df.describe()
print(df_summary.round(2))
if df.isna().sum().sum()>0:
    print("Yes! There are missing observations in this titanic dataset")
    print("Total number of missing observations in this titanic dataset : ",df.isna().sum().sum())
    missing_vals = df.isnull().sum()
    print("These are the counts of missing observations column wise:")
    print(missing_vals)
else:
    print("There are no missing observations in this titanic dataset")

###-----Q4-----
print("\n\n-----Q4-----")
titanic_df=sns.load_dataset('titanic')
print("Displaying first 5 rows of titanic dataset : ")
print(titanic_df.head())
cols=[3,4,5,6]
numerical_df=titanic_df[titanic_df.columns[cols]]
print("Displaying first 5 rows of numerical dataset : ")
print(numerical_df.head())

###-----Q5-----
print("\n\n-----Q5-----")
print("Total number of missing observations in each feature: ")
print(numerical_df.isna().sum())
print("Total number of missing observations in numerical dataset: ")
print(numerical_df.isna().sum().sum())
missing_observations_before = numerical_df.isna().sum().sum()
total_rows_before = numerical_df.shape[0]
df_clean= numerical_df.dropna()
missing_observations_after = df_clean.isna().sum().sum()
total_rows_after = df_clean.shape[0]
print("Total number of observations after cleaning up: ")
print(total_rows_after)

```

```
percentage_cleaned = ((total_rows_before -
total_rows_after)*100/(total_rows_before))
percentage_cleaned = round(percentage_cleaned,2)
print(f"% of data eliminated to clean dataset: {percentage_cleaned}")
```

```
###-----Q8-----
from scipy.stats import gmean
from scipy.stats import hmean
print("\n\n-----Q8-----")
numerical_df=numerical_df.dropna()
def arithmetic_mean(data):
    total_sum = sum(data)
    count = len(data)
    return round(total_sum / count, 2)
```

```
def geometric_mean(data):
    product = 1
    count = len(data)
    for num in data:
        product *= num
    return round(product**(1/count), 2)
```

```
def harmonic_mean(data):
    if 0 in data or len(data) == 0:
        return 0
    count = len(data)
    reciprocal_sum = sum(1 / num for num in data)
    return round(count / reciprocal_sum, 2)
```

```
numerical_df = numerical_df.dropna()
list_cols=['sibsp', 'parch', 'fare']
print(f"Arithmetic mean of age is :",arithmetic_mean(data=numerical_df['age']))
print(f"Geometric mean of age is :",round(gmean(numerical_df['age']), 2))
print(f"Harmonic mean of age is :", round(hmean(numerical_df['age']), 2))
print()
for i in list_cols:
    print(f"Arithmetic mean of {i} is :",arithmetic_mean(data=numerical_df[i]))
    print(f"Geometric mean of {i} is :", geometric_mean(data=numerical_df[i]))
    print(f"Harmonic mean of {i} is :", harmonic_mean(data=numerical_df[i]))
    print()
```

```
###-----Q9-----
print("\n\n-----Q9-----")
print("Histograms...")
sns.histplot(numerical_df['age'], bins=10, label='Age',kde=True,
color='green',edgecolor='black')
plt.xlabel('Age')
plt.legend()
plt.ylabel('Frequency')
plt.legend()
plt.title('Titanic Dataset')
plt.show()
sns.histplot(numerical_df['fare'], bins=10, label='Fare',kde=True,
color='blue',edgecolor='black')
plt.xlabel('Fare')
plt.legend()
```

```
plt.ylabel('Frequency')
plt.title('Titanic Dataset')
plt.show()
```

```
###-----Q10-----
print("\n\n-----Q10-----")
print("Pairwise Distribution...")
sns.pairplot(numerical_df,diag_kind='kde',kind="scatter")
plt.show()
```