Import libraries

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

Load the Titanic dataset

```
titanic= pd.read_csv("/content/train.csv")
titanic
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Emba
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	
				Allen, Mr.								

```
# Question 1: What are the dimensions of the dataset?
```

```
print(titanic.shape)
```

```
(891, 12)
```

```
# Question 2: What are the column names?
```

```
titanic.columns.values.tolist()
```

```
['PassengerId',
 'Survived',
 'Pclass',
 'Name',
 'Sex',
 'Age',
 'SibSp',
 'Parch',
 'Ticket',
 'Fare',
 'Cabin',
 'Embarked']
```

```
# Question 3: Are there any missing values in the dataset?
```

```
titanic.isnull().sum()
```

```
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
```

```
# Question 4: What are the data types of the columns?
```

```
data_types = titanic.dtypes
print("Data Types:\n", data_types)
```

```
Data Types:
 PassengerId      int64
Survived         int64
Pclass           int64
Name            object
Sex             object
Age            float64
SibSp            int64
Parch            int64
Ticket          object
Fare           float64
Cabin           object
Embarked        object
dtype: object
```

```
# Question 5: What are the summary statistics of the dataset?
```

```
titanic.describe()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
count	891.000000	891.000000	891.000000	714.000000	891.000000	891.000000	891.000000
mean	446.000000	0.383838	2.308642	29.699118	0.523008	0.381594	32.204208
std	257.353842	0.486592	0.836071	14.526497	1.102743	0.806057	49.693429
min	1.000000	0.000000	1.000000	0.420000	0.000000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	20.125000	0.000000	0.000000	7.910400
50%	446.000000	0.000000	3.000000	28.000000	0.000000	0.000000	14.454200
75%	668.500000	1.000000	3.000000	38.000000	1.000000	0.000000	31.000000
max	891.000000	1.000000	3.000000	80.000000	8.000000	6.000000	512.329200

```
# Question 6: What is the correlation between different columns?
```

```
titanic.corr()
```

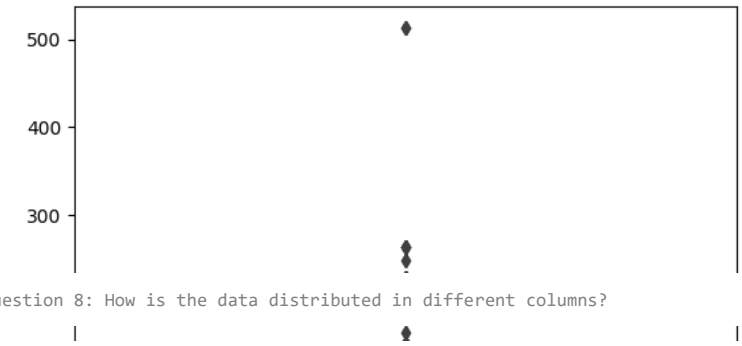
```
<ipython-input-208-c1c691e9860d>:1: FutureWarning: The default value of numeric_only in DataFrame.corr
  titanic.corr()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fare
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.012658
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.257307
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.549500
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.096067
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.159651
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.216225
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.000000

```
# Question 7: Are there any outliers in the dataset?
```

```
import matplotlib.pyplot as plt
import seaborn as sns

# Calculate the IQR for the Fare column
q1 = titanic["Fare"].quantile(0.25)
q3 = titanic["Fare"].quantile(0.75)
iqr = q3 - q1

# Find the outliers in the Fare column
outliers = titanic[(titanic["Fare"] < q1 - 1.5 * iqr) | (titanic["Fare"] > q3 + 1.5 * iqr)]

sns.boxplot(titanic["Fare"])
```

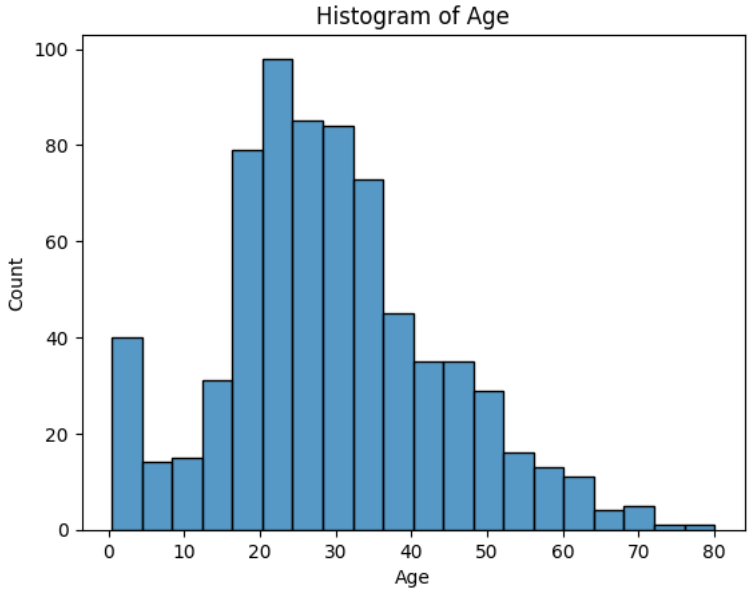
```
<Axes: >
```




```
# Question 8: How is the data distributed in different columns?
```



```
# Histgram

import seaborn as sns
import matplotlib.pyplot as plt
sns.histplot(titanic["Age"], bins=20)
plt.xlabel("Age")
plt.ylabel("Count")
plt.title("Histogram of Age")
plt.show()
```




```
# Question 9: Are there any categorical variables in the dataset?


print(titanic.select_dtypes(include="category"))
```

```
Empty DataFrame
Columns: []
Index: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36,

[891 rows x 0 columns]
```

```
# Question 10: What is the distribution of categorical variables?


cat_features = [feature for feature in titanic.columns if titanic[feature].dtypes == 'O']
print('Number of categorical variables: ', len(cat_features))
print('-'*80)
print('Categorical variables column name:',cat_features)
```

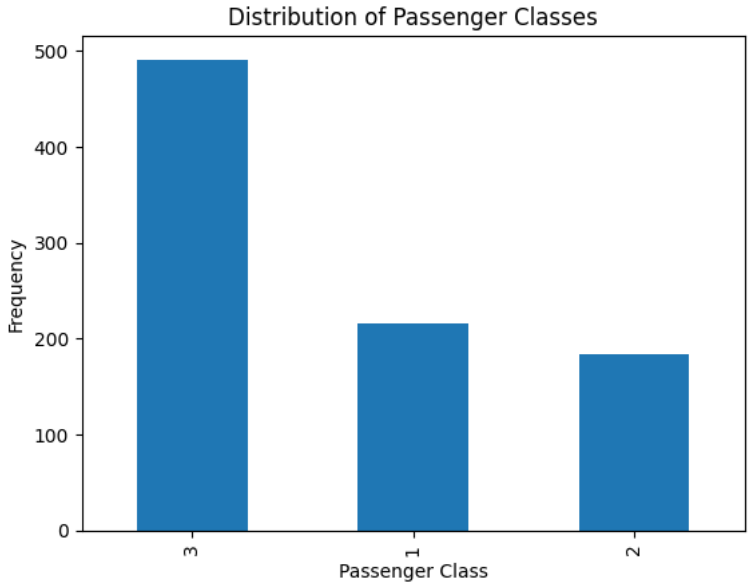
```
Number of categorical variables:  5
--------------------------------------------------------------------------------
Categorical variables column name: ['Name', 'Sex', 'Ticket', 'Cabin', 'Embarked']
```

```
# Question 11: How many passengers survived (1) and died (0)?


titanic['Survived'].value_counts()
```

```
0    549
1    342
Name: Survived, dtype: int64
```

```
# Question 12: What is the distribution of passenger classes (1st, 2nd, 3rd)?


#url = "https://www.kaggle.com/c/titanic/data?select=train.csv"
titanic = pd.read_csv("/content/train.csv")
pclass = titanic["Pclass"]
freq = pclass.value_counts()
print(freq)
freq.plot.bar()
plt.xlabel('Passenger Class')
plt.ylabel("Frequency")
plt.title("Distribution of Passenger Classes")
plt.show()
```

```
3    491
1    216
2    184
Name: Pclass, dtype: int64
```




```
# Question 13: What is the distribution of passengers by gender?


sex = titanic["Sex"]
gender = sex.value_counts()
print(gender)
gender.plot.bar()
plt.xlabel('Passenger by gender')
plt.ylabel("Sex")
plt.title("Distribution of Passenger by gender")
plt.show()
```

```
male      577
female    314
Name: Sex, dtype: int64
```

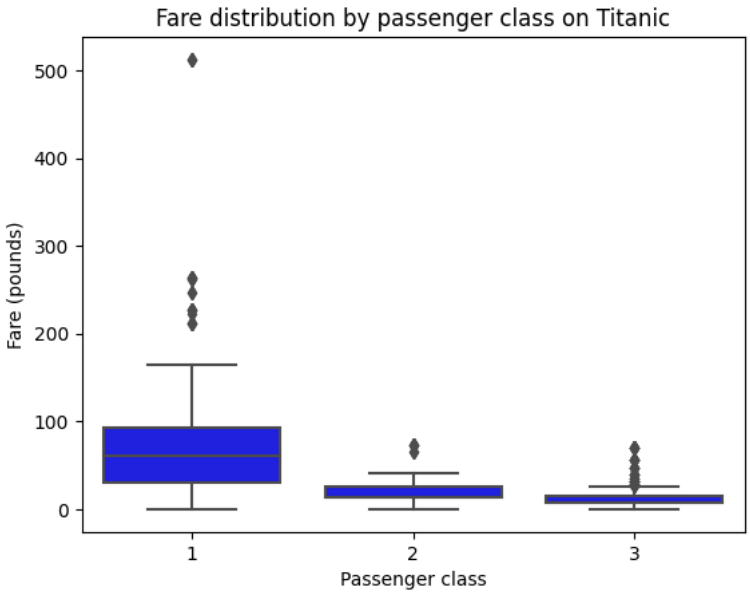
```
# Question 14: What is the average age of passengers?
```

```
titanic["Age"].mean()
```

```
29.69911764705882
```

```
# Question 15: What is the fare distribution for each passenger class?
```

```
sns.boxplot(x="Pclass", y="Fare", data=titanic, color="blue")

plt.xlabel("Passenger class")
plt.ylabel("Fare (pounds)")
plt.title("Fare distribution by passenger class on Titanic")
plt.show()
```



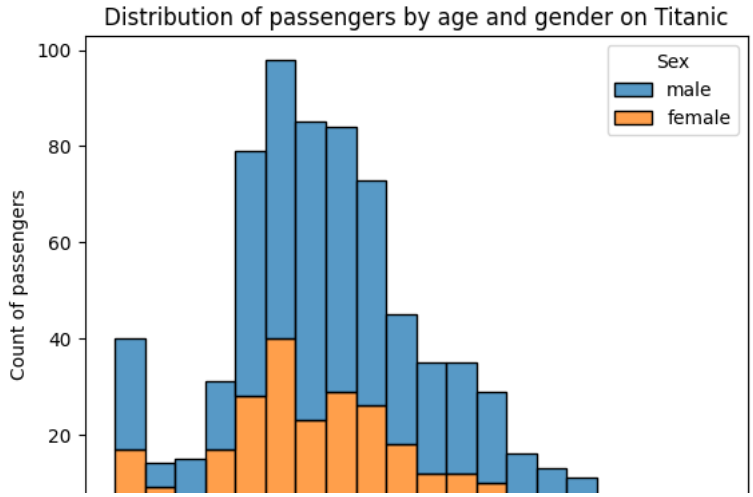

```
# Question 16: What is the survival rate based on passenger class?


titanic["Survived"] = titanic["Survived"].astype(int)

survival_rate = titanic.groupby("Pclass")["Survived"].mean()

print(survival_rate)
```

```
Pclass
1    0.629630
2    0.472826
3    0.242363
Name: Survived, dtype: float64
```

```
# Question 17: What is the survival rate based on gender?


survival_rate = titanic.groupby("Sex")["Survived"].mean()
print(survival_rate)
```

```
Sex
female    0.742038
male      0.188908
Name: Survived, dtype: float64
```

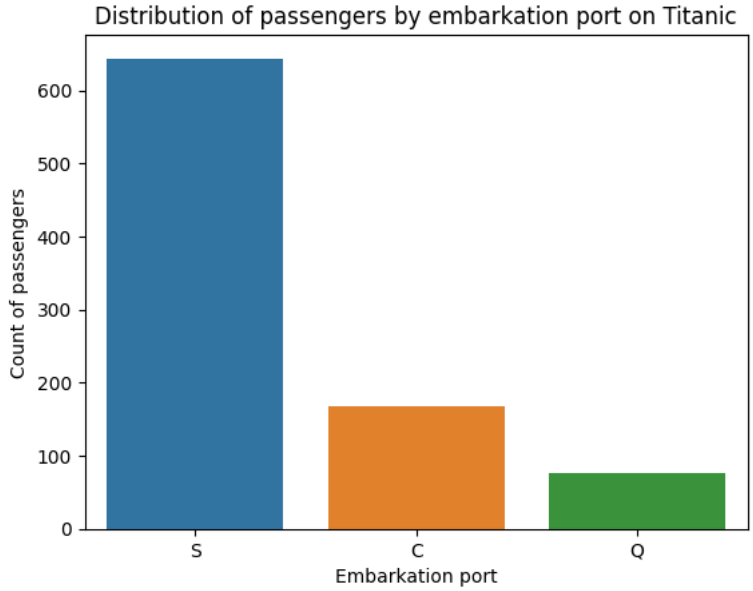
```
# Question 18: What is the distribution of passengers by age and gender?


sns.histplot(data=titanic, x="Age", hue="Sex", multiple="stack", bins=20)
plt.xlabel("Age (years)")
plt.ylabel("Count of passengers")
plt.title("Distribution of passengers by age and gender on Titanic")
plt.show()
```




```
# Question 19: What is the distribution of passengers by embarkation port?
```

+ Code + Text

```
sns.countplot(x="Embarked", data=titanic)
plt.xlabel("Embarkation port")
plt.ylabel("Count of passengers")
plt.title("Distribution of passengers by embarkation port on Titanic")
plt.show()
```




```
# Question 20: What is the survival rate based on embarkation port?


survival_rate = titanic.groupby("Embarked")["Survived"].mean()
print(survival_rate)
```

```
Embarked
C    0.553571
Q    0.389610
S    0.336957
Name: Survived, dtype: float64
```

```
# Question 21: What is the survival rate based on the number of siblings/spouses aboard?


survival_rate = titanic.groupby("SibSp")["Survived"].mean()
print(survival_rate)
```

```
SibSp
0    0.345395
1    0.535885
2    0.464286
3    0.250000
4    0.166667
5    0.000000
8    0.000000
Name: Survived, dtype: float64
```

```
# Question 22: What is the survival rate based on the number of parents/children aboard?


survival_rate = titanic.groupby("Parch")["Survived"].mean()
```

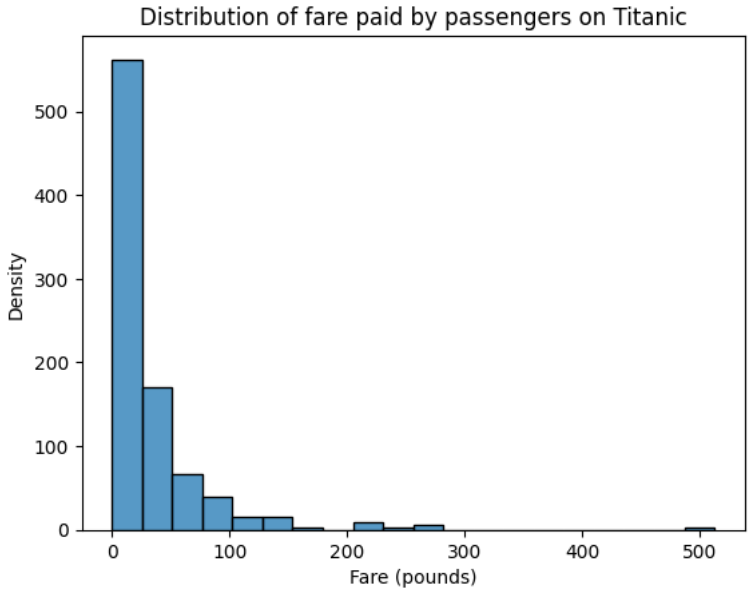
```
survival_rate = titanic.groupby("Parch")["Survived"].mean()
print(survival_rate)
```

```
Parch
0    0.343658
1    0.550847
2    0.500000
3    0.600000
4    0.000000
5    0.200000
6    0.000000
Name: Survived, dtype: float64
```

```
# Question 23: What is the survival rate based on the combination of siblings/spouses and parents/children aboard?


titanic["FamilySize"] = titanic["SibSp"] + titanic["Parch"] + 1
titanic["FamilyType"] = titanic["FamilySize"].map(lambda x: "Alone" if x == 1 else "Small" if x <= 4 else "Large")
survival_rate = titanic.groupby("FamilyType")["Survived"].mean()
print(survival_rate)
```

```
FamilyType
Alone    0.303538
Large    0.161290
Small    0.578767
Name: Survived, dtype: float64
```

```
# Question 24: What is the distribution of fare paid by passengers?


sns.histplot(titanic["Fare"], bins=20)
plt.xlabel("Fare (pounds)")
plt.ylabel("Density")
plt.title("Distribution of fare paid by passengers on Titanic")
plt.show()
```




```
# Question 25: What is the average fare paid by passengers who survived and those who did not?


average_fare = titanic.groupby("Survived")["Fare"].mean()
print(average_fare)
```

```
Survived
0    22.117887
1    48.395408
Name: Fare, dtype: float64
```

```
# Question 26: What is the survival rate based on the cabin class (if available)?


titanic["CabinClass"] = titanic["Cabin"].str[0]
survival_rate = titanic.groupby("CabinClass")["Survived"].mean()
print(survival_rate)
```

```
CabinClass
A    0.466667
B    0.744681
C    0.593220
D    0.757576
E    0.750000
F    0.615385
G    0.500000
```

```
T    0.000000
Name: Survived, dtype: float64
```

```
# Question 27: What is the survival rate based on whether the passenger had a cabin or not?
```

```
titanic["CabinStatus"] = titanic["Cabin"].notnull().map(lambda x: "Had a cabin" if x else "Did not have a cabin")
survival_rate = titanic.groupby("CabinStatus")["Survived"].mean()
print(survival_rate)
```

```
CabinStatus
Did not have a cabin    0.299854
Had a cabin             0.666667
Name: Survived, dtype: float64
```