3.1 Problem1

```
[ ] df= pd.read_csv('/content/drive/MyDrive/Concept and technology of AI/bank .csv')
    df.head(3)
```

	age	job	marital	education	default	balance	housing	loan	contact	day	month	duration	campaign	pdays	previous	poutcome	y
0	58	management	married	tertiary	no	2143	yes	no	unknown	5	may	261	1	-1	0	unknown	no
1	44	technician	single	secondary	no	29	yes	no	unknown	5	may	151	1	-1	0	unknown	no
2	33	entrepreneur	married	secondary	no	2	yes	yes	unknown	5	may	76	1	-1	0	unknown	no

Next steps: Generate code with df | View recommended plots | New interactive sheet

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 45211 entries, 0 to 45210
Data columns (total 17 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   age        45211 non-null  int64
 1   job        45211 non-null  object
 2   marital    45211 non-null  object
 3   education  45211 non-null  object
 4   default    45211 non-null  object
 5   balance    45211 non-null  int64
 6   housing    45211 non-null  object
 7   loan       45211 non-null  object
 8   contact    45211 non-null  object
 9   day        45211 non-null  int64
 10  month      45211 non-null  object
```

✓ 0s completed at 9:14 PM

```
[ ] object_columns = df.select_dtypes(include='object').columns
    object_columns
```

```
Index(['job', 'marital', 'education', 'default', 'housing', 'loan', 'contact',
       'month', 'poutcome', 'y'],
      dtype='object')
```

```
for col in object_columns:
  print(f'{col}: {df[col].unique()}')
```

```
job: ['management' 'technician' 'entrepreneur' 'blue-collar' 'unknown'
 'retired' 'admin.' 'services' 'self-employed' 'unemployed' 'housemaid'
 'student']
marital: ['married' 'single' 'divorced']
education: ['tertiary' 'secondary' 'unknown' 'primary']
default: ['no' 'yes']
housing: ['yes' 'no']
loan: ['no' 'yes']
contact: ['unknown' 'cellular' 'telephone']
month: ['may' 'jun' 'jul' 'aug' 'oct' 'nov' 'dec' 'jan' 'feb' 'mar' 'apr' 'sep']
poutcome: ['unknown' 'failure' 'other' 'success']
y: ['no' 'yes']
```

```
[ ] null = df.isnull().sum()
    null
```

	0
age	0
job	0
marital	0

✓ 0s completed at 9:14 PM

```
[ ] new_df = df.select_dtypes(exclude=['object'])
    new_df.to_csv('new_df.csv', index=False)
    new_df_tab = pd.read_csv('new_df.csv')
    new_df_tab.head(3)
```

	age	balance	day	duration	campaign	pdays	previous
0	58	2143	5	261	1	-1	0
1	44	29	5	151	1	-1	0
2	33	2	5	76	1	-1	0

Next steps: Generate code with new_df_tab | View recommended plots | New interactive sheet

```
[ ] dropped = df.select_dtypes(exclude='object')
    dropped.to_csv('banknumericdata.csv', index=False)
```

```
[ ] reading_csv = pd.read_csv('banknumericdata.csv')
    reading_csv.head(3)
```

	age	balance	day	duration	campaign	pdays	previous
0	58	2143	5	261	1	-1	0
1	44	29	5	151	1	-1	0
2	33	2	5	76	1	-1	0

Next steps: Generate code with reading_csv | View recommended plots | New interactive sheet

Problem 2

```
df2 = pd.read_csv('/content/drive/MyDrive/Concept and technology of AI/medical_students_dataset.csv')
df2.head(3)
```

	Student ID	Age	Gender	Height	Weight	Blood Type	BMI	Temperature	Heart Rate	Blood Pressure	Cholesterol	Diabetes	Smoking
0	1.0	18.0	Female	161.777924	72.354947	O	27.645835	NaN	95.0	109.0	203.0	No	NaN
1	2.0	NaN	Male	152.069157	47.630941	B	NaN	98.714977	93.0	104.0	163.0	No	No
2	3.0	32.0	Female	182.537664	55.741083	A	16.729017	98.260293	76.0	130.0	216.0	Yes	No

```
[ ] df2.info()
    df2.isnull().sum()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200000 entries, 0 to 199999
Data columns (total 13 columns):
 #   Column          Non-Null Count   Dtype
---  ------          --------------   -----
 0   Student ID      180000 non-null  float64
 1   Age             180000 non-null  float64
 2   Gender          180000 non-null  object
 3   Height          180000 non-null  float64
 4   Weight          180000 non-null  float64
 5   Blood Type      180000 non-null  object
 6   BMI             180000 non-null  float64
 7   Temperature     180000 non-null  float64
 8   Heart Rate      180000 non-null  float64
 9   Blood Pressure  180000 non-null  float64
 10  Cholesterol     180000 non-null  float64
 11  Diabetes        180000 non-null  object
 12  Smoking         180000 non-null  object
```

✓ 0s completed at 9:14 PM

```
[ ] #if the age is null then we can fill the null with
    df2["Age"].fillna(df2["Age"].mean(), inplace = True)
```

```
<ipython-input-16-eb4bc4a36b31>:2: FutureWarning: A value is trying to be set on a copy of a DataFrame or Series through chained assignment using an inplace method.
The behavior will change in pandas 3.0. This inplace method will never work because the intermediate object on which we are setting values always behaves as a copy.

For example, when doing 'df[col].method(value, inplace=True)', try using 'df.method({col: value}, inplace=True)' or df[col] = df[col].method(value) instead, to perform the

  df2["Age"].fillna(df2["Age"].mean(), inplace = True)
```

```
[ ] df2.duplicated().sum()
    df2.drop_duplicates(inplace=True)
```

```
df3=pd.read_csv('/content/drive/MyDrive/Concept and technology of AI/Titanic-Dataset.csv')
df3.head(3)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S

Next steps: Generate code with df3 | View recommended plots | New interactive sheet

3.2 Problem 1:

0s completed at 9:14 PM

Search

```
[6] df3_subset = df3[['Name', 'Pclass', 'Sex', 'Age', 'Fare', 'Survived']]
    df3_subset_first_class = df3_subset[df3_subset['Pclass'] == 1]
    fare_mean = df3_subset_first_class['Fare'].mean()
    fare_median = df3_subset_first_class['Fare'].median()
    fare_max = df3_subset_first_class['Fare'].max()
    fare_min = df3_subset_first_class['Fare'].min()
    print("Mean:", fare_mean)
    print("Median:", fare_median)
    print("Max:", fare_max)
    print("Min:", fare_min)
```

```
Mean: 84.1546875
Median: 60.287499999999994
Max: 512.3292
Min: 0.0
```

Problem 2:

```
age = df3_subset_first_class['Age'].isnull().sum()
print(f"Number of missing values in Age: {age}")
df3_drop_missing = df3_subset_first_class.dropna(subset=['Age'])
print(df3_drop_missing.head())
```

```
Number of missing values in Age: 30
                                                 Name  Pclass     Sex   Age  \
1   Cumings, Mrs. John Bradley (Florence Briggs Th...       1  female  38.0
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)       1  female  35.0
6                              McCarthy, Mr. Timothy J       1    male  54.0
11                            Bonnell, Miss. Elizabeth       1  female  58.0
23                         Sloper, Mr. William Thompson       1    male  28.0
```

Problem 3:

```
df_one_hot = pd.get_dummies(df3['Embarked'], prefix='Embarked')
df_with_embarked = pd.concat([df3, df_one_hot], axis=1)
df_with_embarked = df_with_embarked.drop(columns=['Embarked'])
print(df_with_embarked.head())
```

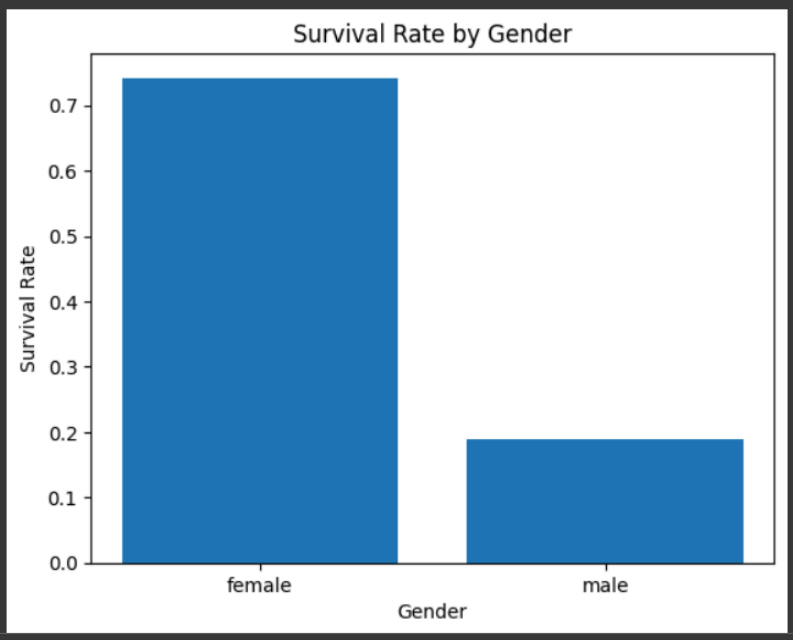
```
   PassengerId  Survived  Pclass  \
0            1         0       3
1            2         1       1
2            3         1       3
3            4         1       1
4            5         0       3

                                                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
2                             Heikkinen, Miss. Laina  female  26.0      0
3       Futrelle, Mrs. Jacques Heath (Lily May Peel)  female  35.0      1
4                           Allen, Mr. William Henry    male  35.0      0

   Parch            Ticket     Fare Cabin  Embarked_C  Embarked_Q  Embarked_S
0      0         A/5 21171   7.2500   NaN       False       False        True
1      0          PC 17599  71.2833   C85        True       False       False
2      0  STON/O2. 3101282   7.9250   NaN       False       False        True
3      0            113803  53.1000  C123       False       False        True
4      0            373450   8.0500   NaN       False       False        True
```

```
[9] import matplotlib.pyplot as plt
    survival_rate_by_gender = df3.groupby('Sex')['Survived'].mean()
    plt.bar(survival_rate_by_gender.index, survival_rate_by_gender.values)
    plt.title('Survival Rate by Gender')
```

0s completed at 9:14 PM

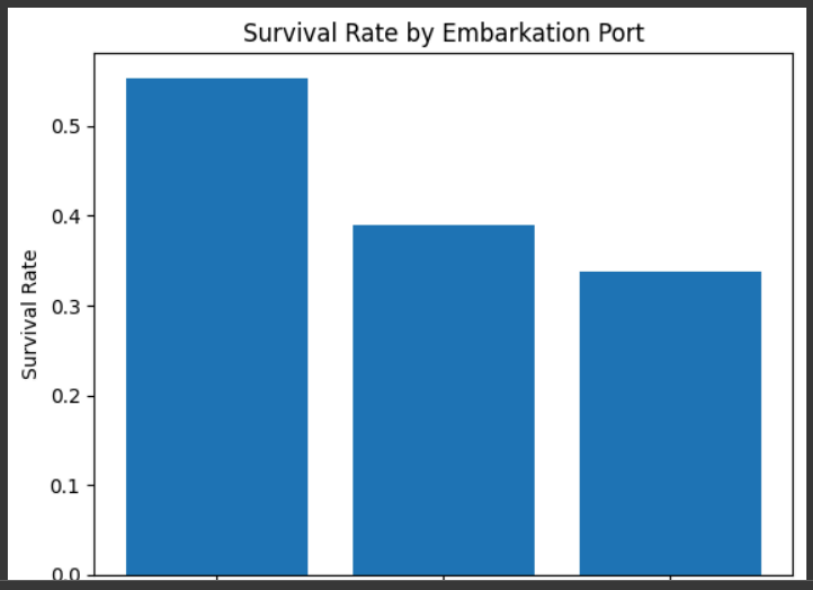
Search

```python
import matplotlib.pyplot as plt
survival_rate_by_gender = df3.groupby('Sex')['Survived'].mean()
plt.bar(survival_rate_by_gender.index, survival_rate_by_gender.values)
plt.title('Survival Rate by Gender')
plt.xlabel('Gender')
plt.ylabel('Survival Rate')
plt.show()
```



✓ 0s completed at 9:14 PM

```python
import matplotlib.pyplot as plt

survival_rate_by_embarked = df3.groupby('Embarked')['Survived'].mean()

survival_rate_by_embarked = survival_rate_by_embarked[['C', 'Q', 'S']]
plt.bar(survival_rate_by_embarked.index, survival_rate_by_embarked.values)
plt.title('Survival Rate by Embarkation Port')
plt.xlabel('Embarkation Port')
plt.ylabel('Survival Rate')
plt.show()
```



✓ 0s completed at 9:14 PM