

To - Do - Task Please Complete all the problem listed below. 3.1 Warming Up Exercises - Basic Inspection and Exploration: Problem 1 - Data Read, Write and Inspect: Complete all following Task:

- Dataset for the Task: "bank.csv"

1. Load the provided dataset and import in pandas DataFrame.

```
[94] 0s
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv("/content/drive/MyDrive/ConceptAndTechnologiesOfAI/Copy of bank.csv");
```

2. Check info of the DataFrame and identify following:

(a) columns with dtypes=object

```
[70] 0s
col = df.select_dtypes(include='object').columns
col

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

(b) unique values of those columns.

```
[71] 0s
for x in col:
    print(df[x].unique())

▼
['management' 'technician' 'entrepreneur' 'blue-collar' 'unknown'
 'retired' 'admin.' 'services' 'self-employed' 'unemployed' 'housemaid'
 'student']
['married' 'single' 'divorced']
['tertiary' 'secondary' 'unknown' 'primary']
['no' 'yes']
['yes' 'no']
['no' 'yes']
['unknown' 'cellular' 'telephone']
['may' 'jun' 'jul' 'aug' 'oct' 'nov' 'dec' 'jan' 'feb' 'mar' 'apr' 'sep']
['unknown' 'failure' 'other' 'success']
['no' 'yes']
```

(c) check for the total number of null values in each column.

```
[72] 0s
df.isnull().sum()

▼
0
```

3. Drop all the columns with dtypes object and store in new DataFrame, also write the DataFrame in ".csv" with name "banknumericdata.csv"

```
[73] ✓ 0s
km = df.copy()
df_numeric = km.drop(columns=col)
df_numeric.to_csv("banknumericdata.csv", index=False)
df_numeric.head()
```

	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
3	47	1506	5	92	1	-1	0
4	33	1	5	198	1	-1	0

Next steps: [Generate code with df_numeric](#) [New interactive sheet](#)

4. Read "banknumericdata.csv" and Find the summary statistics.

```
[74] ✓ 0s
bnk = pd.read_csv("/content/banknumericdata.csv");
bnk.describe()
```

	age	balance	day	duration	campaign	pdays	previous
count	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000	45211.000000
mean	40.936210	1362.272058	15.806419	258.163080	2.763841	40.197828	0.580323
std	10.618762	3044.765829	8.322476	257.527812	3.098021	100.128746	2.303441
min	18.000000	-8019.000000	1.000000	0.000000	1.000000	-1.000000	0.000000
25%	33.000000	72.000000	8.000000	103.000000	1.000000	-1.000000	0.000000
50%	39.000000	448.000000	16.000000	180.000000	2.000000	-1.000000	0.000000
75%	48.000000	1428.000000	21.000000	319.000000	3.000000	-1.000000	0.000000
max	95.000000	102127.000000	31.000000	4918.000000	63.000000	871.000000	275.000000

Problem 2 - Data Imputations: Complete all the following Task:

- Dataset for the Task: "medical_student.csv"

1. Load the provided dataset and import in pandas DataFrame.

```
[75] ✓ 1s
med = pd.read_csv("/content/drive/MyDrive/ConceptAndTechnologiesOfAI/Copy of medical_students_dataset.csv")
med.head()
```

	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
3	NaN	30.0	Male	182.112867	63.332207	B	19.096042	98.839605	99.0	112.0	141.0	No	Yes
4	5.0	23.0	Female	NaN	46.234173	O	NaN	98.480008	95.0	NaN	231.0	No	No

2. Check info of the DataFrame and identify column with missing (null) values.

+ Code

```
med.isnull().sum()
```

0

Student ID	20000
Age	20000
Gender	20000
Height	20000
Weight	20000
Blood Type	20000
BMI	20000
Temperature	20000
Heart Rate	20000
Blood Pressure	20000
Cholesterol	20000
Diabetes	20000

Diabetes	20000
Smoking	20000

dtype: int64

3. For the column with missing values fill the values using various techniques we discussed above. Try to explain why did you select the particular methods for particular column.

```
med.describe()
```

	Student ID	Age	Height	Weight	BMI	Temperature	Heart Rate	Blood Pressure	Cholesterol
count	180000.000000	180000.000000	180000.000000	180000.000000	180000.000000	180000.000000	180000.000000	180000.000000	180000.000000
mean	49974.042078	26.021561	174.947103	69.971585	23.338869	98.600948	79.503767	114.558033	184.486361
std	28879.641657	4.890528	14.447560	17.322574	7.033554	0.500530	11.540755	14.403353	37.559678
min	1.000000	18.000000	150.000041	40.000578	10.074837	96.397835	60.000000	90.000000	120.000000
25%	24971.750000	22.000000	162.476110	54.969838	17.858396	98.264750	70.000000	102.000000	152.000000
50%	49943.500000	26.000000	174.899914	69.979384	22.671401	98.599654	80.000000	115.000000	184.000000
75%	74986.000000	30.000000	187.464417	84.980097	27.997487	98.940543	90.000000	127.000000	217.000000
max	100000.000000	34.000000	100.000000	90.000007	44.355113	100.821857	90.000000	130.000000	240.000000

I see the data for age,temperature and BMI is mostly normal. The standard deviation is low and the mean and median is close to each other. And the quartiles are evenly spaced. So I'll fill the age,temperature and BMI with the value of mean.

```
[78] ✓ 0s
med['Age'] = med['Age'].fillna(med['Age'].mean())
med['Temperature'] = med['Temperature'].fillna(med['Temperature'].mean())
med['BMI'] = med['BMI'].fillna(med['BMI'].mean())
med.isnull().sum()
```

	0
Student ID	20000
Age	0
Gender	20000
Height	20000
Weight	20000
Blood Type	20000
BMI	0
Temperature	0
Heart Rate	20000

dtype: int64

I see the data for height,weight,heart rate,blood pressure and cholesterol is skewed. The standard deviation is fairly high. And the quartiles aren't evenly spaced. So I'll fill the age,temperature and BMI with the value of median. Also the difference between minimum and maximum values are high which means there are outliers in the data.

[+ Code](#) [+ Text](#)

med.describe()

	Student ID	Age	Height	Weight	BMI	Temperature	Heart Rate	Blood Pressure	Cholesterol
count	180000.000000	200000.000000	180000.000000	180000.000000	200000.000000	200000.000000	180000.000000	180000.000000	180000.000000
mean	49974.042078	26.021561	174.947103	69.971585	23.338869	98.600948	79.503767	114.558033	184.486361
std	28879.641657	4.639561	14.447560	17.322574	6.672613	0.474844	11.540755	14.403353	37.559678
min	1.000000	18.000000	150.000041	40.000578	10.074837	96.397835	60.000000	90.000000	120.000000
25%	24971.750000	22.000000	162.476110	54.969838	18.382809	98.306875	70.000000	102.000000	152.000000
50%	49943.500000	26.021561	174.899914	69.979384	23.338869	98.600948	80.000000	115.000000	184.000000
75%	74986.000000	30.000000	187.464417	84.980097	27.255521	98.897102	90.000000	127.000000	217.000000
max	100000.000000	34.000000	199.998639	99.999907	44.355113	100.824857	99.000000	139.000000	249.000000

4. Check for any duplicate values present in Dataset and do necessary to manage the duplicate items. {Hint: dataset.duplicated.sum()}

```
[8] 0s ⏎ med.duplicated().sum()  
... np.int64(7644)
```

3.2 Exercises - Data Cleaning and Transformations with "Titanic Dataset": Dataset Used: "titanic.csv"

Problem - 1:

Create a DataFrame that is subsetted for the columns 'Name', 'Pclass', 'Sex', 'Age', 'Fare', and 'Survived'. Retain only those rows where 'Pclass' is equal to 1, representing first-class passengers. What is the mean, median, maximum value, and minimum value of the 'Fare' column?

```
[1] 0s tic = pd.read_csv("/content/drive/MyDrive/ConceptAndTechnologiesOfAI/Copy_of_Titanic-Dataset.csv")  
tic.head()
```

	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	Allen, Mr. William Henry	male	35.0	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
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	Nan	S

Next steps: [Generate code with tic](#) [New interactive sheet](#)

```
subset = tic[['Name', 'Pclass', 'Sex', 'Age', 'Fare', 'Survived']]  
first_class = subset[subset['Pclass'] == 1]  
print(first_class['Fare'].describe())
```

```
count    216.000000  
mean     84.154687  
std      78.380373  
min      0.000000  
25%     30.923950  
50%     60.287500  
75%     93.500000  
max     512.329200  
Name: Fare, dtype: float64
```

Problem - 2:

How many null values are contained in the 'Age' column in your subsetted DataFrame? Once you've found this out, drop them from your DataFrame.

```
[83] 0s tic1 = tic.copy()  
tic1 = tic1.dropna(subset=['Age'])  
tic1.isnull().sum()
```

	0
PassengerId	0
Survived	0
Pclass	0
Name	0
Sex	0
Age	0
SibSp	0
Parch	0
Ticket	0

Problem - 3:

The 'Embarked' column in the Titanic dataset contains categorical data representing the ports of embarkation:

- 'C' for Cherbourg
 - 'Q' for Queenstown
 - 'S' for Southampton

Task:

1. Use one-hot encoding to convert the 'Embarked' column into separate binary columns ('Embarked C', 'Embarked Q', 'Embarked S').
 2. Add these new columns to the original DataFrame.

```

] os
tic2 = tic.copy()
tic2.head()

tic2["C"] = np.where(tic["Embarked"] == "C", 1, 0)
tic2["Q"] = np.where(tic["Embarked"] == "Q", 1, 0)
tic2["S"] = np.where(tic["Embarked"] == "S", 1, 0)
tic2.head()

```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	C	Q	S
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	0	0	1

PassengerId	Survived	Pclass		Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	C	Q	S
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S	0	0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C85	C	1	0	0
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S	0	0	1
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	0	0	1
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S	0	0	1

Next steps: [Generate code with tic2](#) [New interactive sheet](#)

3. Drop the original 'Embarked' column.

tic2.drop(columns = ["Embarked"])																
	PassengerId	Survived	Pclass	Name			Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	C	Q	S
0	1	0	3	Braund, Mr. Owen Harris			male	22.0	1	0	A/5 21171	7.2500	NaN	0	0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Price Th...)			female	38.0	1	0	PC 17599	71.2833	C85	1	0	0

tic2.drop(columns = ["Embarked"])																
PassengerId	Survived	Pclass				Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	C	Q	S
			0	1	2											
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	0	0	1		
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th... Heikkinen, Miss. Laina	female	38.0	1	0	PC 17599	71.2833	C85	1	0	0		
2	3	1	3	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	0	0	1		
3	4	1	1	Allen, Mr. William Henry	male	35.0	1	0	113803	53.1000	C123	0	0	1		
4	5	0	3	Montvila, Rev. Juozas	male	27.0	0	0	373450	8.0500	NaN	0	0	1		
...		
886	887	0	2	Graham, Miss. Margaret Edith	female	19.0	0	0	211536	13.0000	NaN	0	0	1		
887	888	1	1	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	112053	30.0000	B42	0	0	1		
888	889	0	3	Behr, Mr. Karl Howell	male	26.0	0	0	W./C. 6607	23.4500	NaN	0	0	1		
889	890	1	1	Dooley, Mr. Patrick	male	32.0	0	0	111369	30.0000	C148	1	0	0		
890	891	0	3						370376	7.7500	NaN	0	1	0		

4. Print the first few rows of the modified DataFrame to verify the changes.

```
[86] 0s
tic2.head()
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked	C	Q	S
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	Nan	S	0	0	1
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...)	female	38.0	1	0	PC 17599	71.2833	C85	C	1	0	0
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	Nan	S	0	0	1
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S	0	0	1
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	Nan	S	0	0	1

Next steps: [Generate code with tic2](#) [New interactive sheet](#)

Problem - 4:

Compare the mean survival rates ('Survived') for the different groups in the 'Sex' column. Draw a visualization to show how the survival

Problem - 4:

Compare the mean survival rates ('Survived') for the different groups in the 'Sex' column. Draw a visualization to show how the survival distributions vary by gender.

```
[93] 0s
# Number of records
print(len(tic2))
# No.of survived people
print(tic2["Survived"].sum())

mean = tic2["Survived"].sum() / len(tic2)
print(mean)

891
342
0.3838383838383838
```

Problem - 5:

Draw a visualization that breaks your visualization from Exercise 3 down by the port of embarkation ('Em- barked'). In this instance, compare the ports 'C' (Cherbourg), 'Q' (Queenstown), and 'S' (Southampton).

```
[96]
name = ["C" (Cherbourg),"Q" (Queenstown),"S" (Southampton)]
```

Problem - 5:

Draw a visualization that breaks your visualization from Exercise 3 down by the port of embarkation ('Em- barked'). In this instance, compare the ports 'C' (Cherbourg), 'Q' (Queenstown), and 'S' (Southampton).

```
[96]
name = ["C" (Cherbourg),"Q" (Queenstown),"S" (Southampton)]
value = [tic2["C"].sum(),tic2["Q"].sum(),tic2["S"].sum()]

plt.bar(name,value)
plt.xlabel("Port of Embark")
plt.ylabel("Number of passengers")
plt.title("Visualization")
plt.show()
```



