In this Assignment, you will do-&-excel in :

> 1. importing datasets, dealing with missing values, changing data types.
>
> 2. filtering, sorting, selecting specific column(s).
>
> 3. dealing with duplicate values, dropping and adding rows and columns.
>
> 4. counting values, counting unique values.

Follow the hints, limited googling is allowed

In [ ]:
```
import pandas as pd
```

In [ ]:
```
# Steps to upload any dataset into your Colab NB :
# step 1 : First Download the dataset to your local PC.
#          The link for downloading our dataset for practicing is https://drive.google.com/open?id=1uwGlwHjyBd4XECeoNzcY4zzjLB
# step 2 : Run the below code and select the (above downloaded) dataset.
# from google.colab import files
# files.upload()
```

In [ ]:
```
# Import dataset from datasets/titanic/train.csv   with read_csv().
data = pd.read_csv("train.csv")
```

In [ ]:
```
# Check first 7 rows of the DataFrame with .head() method.
data.head(7)
```

Out[ ]:

	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
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
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S

In [ ]:
```
# Check last 4 rows of the DataFrame with .tail() method.
data.tail(4)
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45	NaN	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.75	NaN	Q

In [ ]:
```
# how many rows or records are their in the df or dataset ?
# or simply said whats the shape of the dataframe ?
data.shape[0]
```

Out[ ]:
```
891
```

In [ ]:
```
# You don’t want all of the columns from the CSV file.
# Instead You just want 3 of them namely PassengerId, Survived, Pclass
# usecols argument to specify the column names that we want to work with.
# as well display first five records.
data[["PassengerId","Survived","Pclass"]].head()
```

Out[ ]:

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

In [ ]:
```
# Getting some information about dataset with .describe() and .info()
# try .describe() over df
data.describe()


# Note : describe() gets a summary of numeric values in your dataset.
# It calculates the mean, standard deviation, minimum value, maximum value,
# 1st percentile, 2nd percentile, 3rd percentile of the columns
# with numeric values. It also counts the number of variables in the dataset.
# So, we will be able to see if there are missing values in columns.
```

Out[ ]:

	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

In [ ]:
```
# try .info()
data.info()

# Note : This method (.info) prints information about a DataFrame
# including column dtypes, non-null values and memory usage.
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 891 entries, 0 to 890
Data columns (total 12 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   PassengerId  891 non-null    int64
 1   Survived     891 non-null    int64
 2   Pclass       891 non-null    int64
 3   Name         891 non-null    object
 4   Sex          891 non-null    object
 5   Age          714 non-null    float64
 6   SibSp        891 non-null    int64
 7   Parch        891 non-null    int64
 8   Ticket       891 non-null    object
 9   Fare         891 non-null    float64
 10  Cabin        204 non-null    object
 11  Embarked     889 non-null    object
dtypes: float64(2), int64(5), object(5)
memory usage: 83.7+ KB
```

In [ ]:
```
# print only data types of columns with .dtypes attribute
data.dtypes
```

Out[ ]:
```
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
```

In [ ]:
```
# SORTING THE DATA

# show me first 10 rows sorted according to the 'Fare' in non-decreasing order.
data.sort_values(by=["Fare"],ascending=True).head(10)
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
271	272	1	3	Tornquist, Mr. William Henry	male	25.0	0	0	LINE	0.0	NaN	S
597	598	0	3	Johnson, Mr. Alfred	male	49.0	0	0	LINE	0.0	NaN	S
302	303	0	3	Johnson, Mr. William Cahoone Jr	male	19.0	0	0	LINE	0.0	NaN	S
633	634	0	1	Parr, Mr. William Henry Marsh	male	NaN	0	0	112052	0.0	NaN	S
277	278	0	2	Parkes, Mr. Francis "Frank"	male	NaN	0	0	239853	0.0	NaN	S
413	414	0	2	Cunningham, Mr. Alfred Fleming	male	NaN	0	0	239853	0.0	NaN	S
674	675	0	2	Watson, Mr. Ennis Hastings	male	NaN	0	0	239856	0.0	NaN	S
263	264	0	1	Harrison, Mr. William	male	40.0	0	0	112059	0.0	B94	S
466	467	0	2	Campbell, Mr. William	male	NaN	0	0	239853	0.0	NaN	S
732	733	0	2	Knight, Mr. Robert J	male	NaN	0	0	239855	0.0	NaN	S

In [ ]:
```
# show me first 8 passengers who paid the highest ticket fares.
data.sort_values(by=["Fare"],ascending=False).head(8)
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	NaN	C
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292	B101	C
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.3292	B51 B53 B55	C
88	89	1	1	Fortune, Miss. Mabel Helen	female	23.0	3	2	19950	263.0000	C23 C25 C27	S
27	28	0	1	Fortune, Mr. Charles Alexander	male	19.0	3	2	19950	263.0000	C23 C25 C27	S
341	342	1	1	Fortune, Miss. Alice Elizabeth	female	24.0	3	2	19950	263.0000	C23 C25 C27	S
438	439	0	1	Fortune, Mr. Mark	male	64.0	1	4	19950	263.0000	C23 C25 C27	S
311	312	1	1	Ryerson, Miss. Emily Borie	female	18.0	2	2	PC 17608	262.3750	B57 B59 B63 B66	C

In [ ]:
```
# above sorting did not sort the original dataset
# if you print the df.head() you would see the original dataset itself
# Sort the df rows in descending order on "Fare".
# Also make it effective on origional dataset
data = data.sort_values(by=["Fare"],ascending=False)
data
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	NaN	C
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292	B101	C
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.3292	B51 B53 B55	C
88	89	1	1	Fortune, Miss. Mabel Helen	female	23.0	3	2	19950	263.0000	C23 C25 C27	S
27	28	0	1	Fortune, Mr. Charles Alexander	male	19.0	3	2	19950	263.0000	C23 C25 C27	S
...	...	...	...	...	...	...	...	...	...	...	...	...
633	634	0	1	Parr, Mr. William Henry Marsh	male	NaN	0	0	112052	0.0000	NaN	S
413	414	0	2	Cunningham, Mr. Alfred Fleming	male	NaN	0	0	239853	0.0000	NaN	S
822	823	0	1	Reuchlin, Jonkheer. John George	male	38.0	0	0	19972	0.0000	NaN	S
732	733	0	2	Knight, Mr. Robert J	male	NaN	0	0	239855	0.0000	NaN	S
674	675	0	2	Watson, Mr. Ennis Hastings	male	NaN	0	0	239856	0.0000	NaN	S

891 rows × 12 columns

In [ ]:
```
# Sort df with “Cabin” column.
# As there are lots of missing (NaN) values in this column.
data = data.sort_values(by=["Cabin"],ascending=True)
```

In [ ]:
```
# check if the nan values are at the bottom of our dataset ?
data.tail()
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
633	634	0	1	Parr, Mr. William Henry Marsh	male	NaN	0	0	112052	0.0	NaN	S
413	414	0	2	Cunningham, Mr. Alfred Fleming	male	NaN	0	0	239853	0.0	NaN	S
822	823	0	1	Reuchlin, Jonkheer. John George	male	38.0	0	0	19972	0.0	NaN	S
732	733	0	2	Knight, Mr. Robert J	male	NaN	0	0	239855	0.0	NaN	S
674	675	0	2	Watson, Mr. Ennis Hastings	male	NaN	0	0	239856	0.0	NaN	S

In [ ]:
```
# Counting the occurences of variables
data.value_counts()
# finding how many unique variables are there in a column
# or the occurence of each item in a column.
```

Out[ ]:
```
PassengerId  Survived  Pclass  Name                                                Sex     Age   SibSp  Parch  Ticket    Fa
re      Cabin  Embarked
2            1         1       Cumings, Mrs. John Bradley (Florence Briggs Thayer)   female  38.0  1      0      PC 17599  7
1.2833  C85    C           1
572          1         1       Appleton, Mrs. Edward Dale (Charlotte Lamson)         female  53.0  2      0      11769     5
1.4792  C101   S           1
578          1         1       Silvey, Mrs. William Baird (Alice Munger)             female  39.0  1      0      13507     5
5.9000  E44    S           1
582          1         1       Thayer, Mrs. John Borland (Marian Longstreth Morris)  female  39.0  1      1      17421     11
0.8833  C68    C           1
584          0         1       Ross, Mr. John Hugo                                   male    36.0  0      0      13049     4
0.1250  A10    C           1

..
328          1         2       Ball, Mrs. (Ada E Hall)                               female  36.0  0      0      28551     1
3.0000  D      S           1
330          1         1       Hippach, Miss. Jean Gertrude                          female  16.0  0      1      111361    5
7.9792  B18    C           1
332          0         1       Partner, Mr. Austen                                   male    45.5  0      0      113043    2
8.5000  C124   S           1
333          0         1       Graham, Mr. George Edward                             male    38.0  0      1      PC 17582  15
3.4625  C91    S           1
890          1         1       Behr, Mr. Karl Howell                                 male    26.0  0      0      111369    3
0.0000  C148   C           1
Name: count, Length: 183, dtype: int64
```

In [ ]:
```
# Using .nunique() to count number of unique values that occur in dataset
# or in a column
data.nunique()
# try df.nunique()
```

Out[ ]:
```
PassengerId    891
Survived         2
Pclass           3
Name           891
Sex              2
Age             88
SibSp            7
Parch            7
Ticket         681
Fare           248
Cabin          147
Embarked         3
dtype: int64
```

In [ ]:
```
# try finding unique values for Embarked
data["Embarked"].unique()
```

Out[ ]:
```
array(['C', 'S', nan, 'Q'], dtype=object)
```

We checked the data types of the columns in Titanic dataset. We saw that the type of ***Embarked*** column is object. After counting the unique values in Embarked column with .unique(), we can see that there are 3 unique values in that column. So we can consider that the data type should be categorical.

change the datatype of ***Embarked*** column to categorical.

In [ ]:
```
# change the datatype of Embarked column to categorical
# use .astype("category")
data["Embarked"] = data["Embarked"].astype("category")
```

In [ ]:
```
# Filtering

# find passengers who Embarked from port "C"
data[data["Embarked"] == "C"]
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
583	584	0	1	Ross, Mr. John Hugo	male	36.0	0	0	13049	40.1250	A10	C
556	557	1	1	Duff Gordon, Lady. (Lucille Christiana Sutherl...	female	48.0	1	0	11755	39.6000	A16	C
599	600	1	1	Duff Gordon, Sir. Cosmo Edmund ("Mr Morgan")	male	49.0	1	0	PC 17485	56.9292	A20	C
647	648	1	1	Simonius-Blumer, Col. Oberst Alfons	male	56.0	0	0	13213	35.5000	A26	C
209	210	1	1	Blank, Mr. Henry	male	40.0	0	0	112277	31.0000	A31	C
...	...	...	...	...	...	...	...	...	...	...	...	...
875	876	1	3	Najib, Miss. Adele Kiamie "Jane"	female	15.0	0	0	2667	7.2250	NaN	C
354	355	0	3	Yousif, Mr. Wazli	male	NaN	0	0	2647	7.2250	NaN	C
522	523	0	3	Lahoud, Mr. Sarkis	male	NaN	0	0	2624	7.2250	NaN	C
843	844	0	3	Lemberopolous, Mr. Peter L	male	34.5	0	0	2683	6.4375	NaN	C
378	379	0	3	Betros, Mr. Tannous	male	20.0	0	0	2648	4.0125	NaN	C

168 rows × 12 columns

In [ ]:
```
# another way to do the above query could be as :
c_embark = data["Embarked"] == "C"
data[c_embark]
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
583	584	0	1	Ross, Mr. John Hugo	male	36.0	0	0	13049	40.1250	A10	C
556	557	1	1	Duff Gordon, Lady. (Lucille Christiana Sutherl...	female	48.0	1	0	11755	39.6000	A16	C
599	600	1	1	Duff Gordon, Sir. Cosmo Edmund ("Mr Morgan")	male	49.0	1	0	PC 17485	56.9292	A20	C
647	648	1	1	Simonius-Blumer, Col. Oberst Alfons	male	56.0	0	0	13213	35.5000	A26	C
209	210	1	1	Blank, Mr. Henry	male	40.0	0	0	112277	31.0000	A31	C
...	...	...	...	...	...	...	...	...	...	...	...	...
875	876	1	3	Najib, Miss. Adele Kiamie "Jane"	female	15.0	0	0	2667	7.2250	NaN	C
354	355	0	3	Yousif, Mr. Wazli	male	NaN	0	0	2647	7.2250	NaN	C
522	523	0	3	Lahoud, Mr. Sarkis	male	NaN	0	0	2624	7.2250	NaN	C
843	844	0	3	Lemberopolous, Mr. Peter L	male	34.5	0	0	2683	6.4375	NaN	C
378	379	0	3	Betros, Mr. Tannous	male	20.0	0	0	2648	4.0125	NaN	C

168 rows × 12 columns

Filtering under two or more condition

We are going to use AND and OR operator to filter with more than one condition. Let's assume that we want to see the passengers whose *Fare is lesser than 100* and *who are female*. We are going to create 2 new masks to complete that.

follow this (below) solution

In [ ]:
```
# finding female passengers having ticket fare < $100

cond = (data["Fare"] < 100) & (data["Sex"] == "female")
data[cond]
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
556	557	1	1	Duff Gordon, Lady. (Lucille Christiana Sutherl...	female	48.0	1	0	11755	39.6000	A16	C
523	524	1	1	Hippach, Mrs. Louis Albert (Ida Sophia Fischer)	female	44.0	0	1	111361	57.9792	B18	C
329	330	1	1	Hippach, Miss. Jean Gertrude	female	16.0	0	1	111361	57.9792	B18	C
781	782	1	1	Dick, Mrs. Albert Adrian (Vera Gillespie)	female	17.0	1	0	17474	57.0000	B20	S
540	541	1	1	Crosby, Miss. Harriet R	female	36.0	0	2	WE/P 5735	71.0000	B22	S
...	...	...	...	...	...	...	...	...	...	...	...	...
367	368	1	3	Moussa, Mrs. (Mantoura Boulos)	female	NaN	0	0	2626	7.2292	NaN	C
780	781	1	3	Ayoub, Miss. Banoura	female	13.0	0	0	2687	7.2292	NaN	C
19	20	1	3	Masselmani, Mrs. Fatima	female	NaN	0	0	2649	7.2250	NaN	C
875	876	1	3	Najib, Miss. Adele Kiamie "Jane"	female	15.0	0	0	2667	7.2250	NaN	C
654	655	0	3	Hegarty, Miss. Hanora "Nora"	female	18.0	0	0	365226	6.7500	NaN	Q

280 rows × 12 columns

In [ ]:
```
# do this
# find passenger's whose fare is > $500 or older than 70 years.
# Note : or operator is indicated as pipe ->    |

cond = (data["Fare"] > 500) | (data["Age"] > 70)
data[cond]
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0	27042	30.0000	A23	S
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0	PC 17754	34.6542	A5	C
737	738	1	1	Lesurer, Mr. Gustave J	male	35.0	0	0	PC 17755	512.3292	B101	C
679	680	1	1	Cardeza, Mr. Thomas Drake Martinez	male	36.0	0	1	PC 17755	512.3292	B51 B53 B55	C
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	NaN	C
493	494	0	1	Artagaveytia, Mr. Ramon	male	71.0	0	0	PC 17609	49.5042	NaN	C
851	852	0	3	Svensson, Mr. Johan	male	74.0	0	0	347060	7.7750	NaN	S
116	117	0	3	Connors, Mr. Patrick	male	70.5	0	0	370369	7.7500	NaN	Q

In [ ]:
```
# Finding the null values with .isnull()
data[data.isnull()]
# find passengers whose cabin is unknown i.e NaN
data[data["Cabin"].isna()]
```

Out[ ]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
258	259	1	1	Ward, Miss. Anna	female	35.0	0	0	PC 17755	512.3292	NaN	C
557	558	0	1	Robbins, Mr. Victor	male	NaN	0	0	PC 17757	227.5250	NaN	C
380	381	1	1	Bidois, Miss. Rosalie	female	42.0	0	0	PC 17757	227.5250	NaN	C
856	857	1	1	Wick, Mrs. George Dennick (Mary Hitchcock)	female	45.0	1	1	36928	164.8667	NaN	S
708	709	1	1	Cleaver, Miss. Alice	female	22.0	0	0	113781	151.5500	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
633	634	0	1	Parr, Mr. William Henry Marsh	male	NaN	0	0	112052	0.0000	NaN	S
413	414	0	2	Cunningham, Mr. Alfred Fleming	male	NaN	0	0	239853	0.0000	NaN	S
822	823	0	1	Reuchlin, Jonkheer. John George	male	38.0	0	0	19972	0.0000	NaN	S
732	733	0	2	Knight, Mr. Robert J	male	NaN	0	0	239855	0.0000	NaN	S
674	675	0	2	Watson, Mr. Ennis Hastings	male	NaN	0	0	239856	0.0000	NaN	S

687 rows × 12 columns

In [ ]:
```
# find count of passengers whose cabin is not known.
# use .isnull().sum()  functions
data["Cabin"].isna().sum()
```

Out[ ]:
```
687
```

Dealing with missing values

There are lots of ways to deal with missing values but most common are **to "drop the tuple"** or to **"fill it with median".**

We are going to ignore the "Cabin" column since 70% of that column is missing. And we are going to fill the missing Ages with median value of that column.

In [ ]:
```
# drop the "Cabin" column
data.drop(columns=["Cabin"],inplace=True)
```

In [ ]:
```
# drop those rows which have No Age information
data['Age'].dropna()
# making new data frame with dropped NA values
clean_data = data['Age'].dropna()



# comparing sizes of data frames
print("Old data frame length: {}\nNew data frame length: {}\nNumber of rows with at least 1 NA value: {}".format(data.shape[0]
```

```
Old data frame length: 891
New data frame length: 714
Number of rows with at least 1 NA value: 891
```

In [ ]:
```
# Filling missing values with .fillna()

# fill the missing values with the median of Age column
data["Age"].fillna(data['Age'].median(),inplace=True)
data["Age"].isna().sum()
```

Out[ ]:
```
0
```