```
import pandas as pd
dataframe = pd.DataFrame()
```

```
url = "/content/test (2).csv"
df = pd.read_csv(url)
# show first two rows
print(df.head (2)) # also try tail (2) for last two rows
```

```
   PassengerId  Pclass                              Name     Sex   Age  SibSp  \
0          892       3                  Kelly, Mr. James    male  34.5      0
1          893       3  Wilkes, Mrs. James (Ellen Needs)  female  47.0      1

   Parch  Ticket    Fare Cabin Embarked
0      0  330911  7.8292   NaN        Q
1      0  363272  7.0000   NaN        S
```

```
# select the first row
print("First row : \n",df.iloc[0])

# select three rows
print("Three rows : \n",df.iloc[1:4])

# all rows up to and including the fourth row
print("All rows up to and including the fourth row : \n",df.iloc[:4])

# set index
df = df.set_index(df['Name'])
```

```
First row :
 PassengerId                 892
Pclass                        3
Name           Kelly, Mr. James
Sex                        male
Age                        34.5
SibSp                         0
Parch                         0
Ticket                   330911
Fare                     7.8292
Cabin                       NaN
Embarked                      Q
Name: 0, dtype: object
Three rows :
    PassengerId  Pclass                              Name     Sex   Age  SibSp  \
1          893       3  Wilkes, Mrs. James (Ellen Needs)  female  47.0      1
2          894       2         Myles, Mr. Thomas Francis    male  62.0      0
3          895       3                  Wirz, Mr. Albert    male  27.0      0

   Parch  Ticket    Fare Cabin Embarked
1      0  363272  7.0000   NaN        S
2      0  240276  9.6875   NaN        Q
3      0  315154  8.6625   NaN        S
All rows up to and including the fourth row :
    PassengerId  Pclass                              Name     Sex   Age  SibSp  \
0          892       3                  Kelly, Mr. James    male  34.5      0
1          893       3  Wilkes, Mrs. James (Ellen Needs)  female  47.0      1
2          894       2         Myles, Mr. Thomas Francis    male  62.0      0
3          895       3                  Wirz, Mr. Albert    male  27.0      0

   Parch  Ticket    Fare Cabin Embarked
0      0  330911  7.8292   NaN        Q
1      0  363272  7.0000   NaN        S
2      0  240276  9.6875   NaN        Q
3      0  315154  8.6625   NaN        S
```

```
# select top two rows where column 'sex' is 'female'
df [df ['Sex'] == 'female'].head (2)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
Name										
Wilkes, Mrs. James	893	3	Wilkes, Mrs. James	female	47.0	1	0	363272	7.0000	

```
# multiple conditions
df [(df['Sex'] == 'female') & (df ['Age'] >= 65)]
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cab
Name										
Cavendish,			Cavendish							

```
# replace any instance of 'female' with Woman
df ['Sex'].replace('female', 'Woman').head (2)
```

```
Name
Kelly, Mr. James                    male
Wilkes, Mrs. James (Ellen Needs)    Woman
Name: Sex, dtype: object
```

```
# replace any instance of 'female' with Woman
df ['Sex'].replace(['female', 'male'], ['Woman', 'Man']). head (5)
```

```
Name
Kelly, Mr. James                                  Man
Wilkes, Mrs. James (Ellen Needs)                Woman
Myles, Mr. Thomas Francis                         Man
Wirz, Mr. Albert                                  Man
Hirvonen, Mrs. Alexander (Helga E Lindqvist)    Woman
Name: Sex, dtype: object
```

```
df.replace(1, "One").head (2)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Em
Name											
Kelly, Mr. James	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	

```
df.rename(columns={'PClass': 'Passenger Class'}).head (2)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Em
Name											
Kelly, Mr. James	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	

```
print('Maximum: {}'.format(df ['Age'].max()))
print('Minimum: {}'.format(df['Age'].min()))
print('Mean: {}'.format(df ['Age'].mean()))
print('Sum: {}'.format(df ['Age'].sum()))
print('Count: {}'.format(df ['Age'].count()))
```

```
Maximum: 76.0
Minimum: 0.17
Mean: 30.272590361445783
Sum: 10050.5
Count: 332
```

```
df.rename (columns={'PClass': 'Passenger Class', 'Sex': 'Gender'}).head (2)
```

	PassengerId	Pclass	Name	Gender	Age	SibSp	Parch	Ticket	Fare	Cabin	Eı
Name											
Kelly, Mr. James	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	

```
# unique will return an array of all unique values in a column
df ['Sex'].unique()
```

```
array(['male', 'female'], dtype=object)
```

```
# value_counts will display all unique values with the number of times each value appears
df ['Sex'].value_counts()
```

```
Sex
male      266
female    152
Name: count, dtype: int64
```

```
# select missing values, show 2 rows
df [df ['Age'].isnull()].head (2)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
Name									
Ilieff, Mr. Ylio	902	3	Ilieff, Mr. Ylio	male	NaN	0	0	349220	7.8958

```
# axis=1 means the column axis
df.drop('Age', axis=1).head (2)
```

	PassengerId	Pclass	Name	Sex	SibSp	Parch	Ticket	Fare	Cabin	Embarke
Name										
Kelly, Mr. James	892	3	Kelly, Mr. James	male	0	0	330911	7.8292	NaN	(

```
# create new dataframe excluding the rows you want to delete
df [df ['Sex'] != 'male'].head (2)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Ca
Name										
Wilkes, Mrs. James	893	3	Wilkes, Mrs. James	female	47.0	1	0	363272	7.0000	I

```
# delete a row by matching a unique value
df [df['Name'] != 'Allison, Miss Helen Loraine'].head(2)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Em
Name											
Kelly, Mr. James	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	

```
# delete a row by index df [df.index
df[df.index != 0].head (2)
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Em
Name											
Kelly, Mr. James	892	3	Kelly, Mr. James	male	34.5	0	0	330911	7.8292	NaN	

```
df.groupby( 'Cabin') ['Name'].count()
```

```
Cabin
A11      1
A18      1
A21      1
A29      1
A34      2
        ..
F G63    1
F2       1
F33      1
F4       2
```

```
G6       1
Name: Name, Length: 76, dtype: int64
```

```
# for .. in.. loop
for name in df ['Name'][0:2]:
  print(name.upper())
```

```
KELLY, MR. JAMES
WILKES, MRS. JAMES (ELLEN NEEDS)
```

```
# list comprehension (more "pythonic")
[name.upper() for name in df ['Name'] [0:2]]
```

```
['KELLY, MR. JAMES', 'WILKES, MRS. JAMES (ELLEN NEEDS)']
```

```
def uppercase(x):
  return x.upper()
df ['Name'].apply(uppercase) [0:2]
```

```
Name
Kelly, Mr. James                                    KELLY, MR. JAMES
Wilkes, Mrs. James (Ellen Needs)    WILKES, MRS. JAMES (ELLEN NEEDS)
Name: Name, dtype: object
```

```
df.groupby('Sex').apply(lambda x: x.count())
```

	PassengerId	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
Sex											
female	152	152	152	152	127	152	152	152	152	44	152
male	266	266	266	266	205	266	266	266	265	47	266

+ Code + Text

```
import numpy as np
numerical_df=df.select_dtypes(include=[np.number])
numerical_df.head()
numerical_df=df.select_dtypes(include=[np.number])
numerical_df.head()
categorical_df = pd.DataFrame({
# Define the columns and data for the categorical DataFrame here
})
comb_df = pd.concat([numerical_df,categorical_df],axis=1)
comb_df.head()
```

	PassengerId	Pclass	Age	SibSp	Parch	Fare
Name						
Kelly, Mr. James	892	3	34.5	0	0	7.8292
Wilkes, Mrs. James (Ellen Needs)	893	3	47.0	1	0	7.0000
Myles, Mr. Thomas Francis	894	2	62.0	0	0	9.6875
Wirz, Mr. Albert	895	3	27.0	0	0	8.6625
Hirvonen, Mrs. Alexander (Helga E Lindqvist)	896	3	22.0	1	1	12.2875

Next steps: Generate code with comb_df View recommended plots