Importing Pandas

```
import pandas as pd
```

Loading Dataset

```
df = pd.read_csv(r'C:\Users\hp\Desktop\100DaysOfDataScience\Day 14\
Titanic-Dataset.csv',index_col=0,header=0)
df.head()
```

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

                                                          Name     Sex
Age  \
PassengerId

1                                       Braund, Mr. Owen Harris    male
22.0
2            Cumings, Mrs. John Bradley (Florence Briggs Th...  female
38.0
3                                        Heikkinen, Miss. Laina  female
26.0
4                 Futrelle, Mrs. Jacques Heath (Lily May Peel)  female
35.0
```

```
5                                    Allen, Mr. William Henry    male
35.0

             SibSp  Parch            Ticket     Fare Cabin Embarked
PassengerId
1                1      0         A/5 21171   7.2500   NaN        S
2                1      0          PC 17599  71.2833   C85        C
3                0      0  STON/O2. 3101282   7.9250   NaN        S
4                1      0            113803  53.1000  C123        S
5                0      0            373450   8.0500   NaN        S
```

Checking Columns

.columns - The column labels of the DataFrame.

```
df.columns
```

```
Index(['Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch',
'Ticket',
       'Fare', 'Cabin', 'Embarked'],
      dtype='object')
```

Checking Shape

.shape - Return a tuple representing the dimensionality and size of the DataFrame.

```
df.shape
```

```
(891, 11)
```

Checking Information

.info() - This method prints information about a DataFrame including the index dtype and columns, non-null values and memory usage.

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 891 entries, 1 to 891
Data columns (total 11 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   Survived  891 non-null    int64
 1   Pclass    891 non-null    int64
 2   Name      891 non-null    object
 3   Sex       891 non-null    object
 4   Age       714 non-null    float64
 5   SibSp     891 non-null    int64
 6   Parch     891 non-null    int64
```

```
 7   Ticket    891 non-null    object
 8   Fare      891 non-null    float64
 9   Cabin     204 non-null    object
 10  Embarked  889 non-null    object
dtypes: float64(2), int64(4), object(5)
memory usage: 83.5+ KB
```

Checking Description

.describe() -

- Descriptive statistics include those that summarize the central tendency, dispersion and shape of a dataset's distribution, excluding NaN values.
- Analyzes both numeric and object series, as well as DataFrame column sets of mixed data types.

```
df.describe()
```

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

Checking Datatypes

.dtypes() -

- This returns a Series with the data type of each column.
- The result's index is the original DataFrame's columns.
- Columns with mixed types are stored with the object dtype.

```
df.dtypes
```

```
Survived      int64
Pclass        int64
Name         object
```

```
Sex          object
Age         float64
SibSp         int64
Parch         int64
Ticket       object
Fare        float64
Cabin        object
Embarked     object
dtype: object
```

Location Specific Data Using iloc & loc

iloc[] -

- It allows users to select specific rows and columns by providing integer indices, making it a valuable tool for data manipulation and extraction based on numerical positions within the DataFrame.
- iloc[rows,columns]

loc[] -

- It is a method that takes only index labels and returns row or dataframe if the index label exists in the caller data frame.
- loc[rows,columns]

```
df.iloc[0:20,0:5]
```

PassengerId	Survived	Pclass	\
1	0	3	
2	1	1	
3	1	3	
4	1	1	
5	0	3	
6	0	3	
7	0	1	
8	0	3	
9	1	3	
10	1	2	
11	1	3	
12	1	1	
13	0	3	
14	0	3	
15	0	3	
16	1	2	
17	0	3	
18	1	2	
19	0	3	
20	1	3	

```
                                                                Name     Sex
Age
PassengerId

1                                        Braund, Mr. Owen Harris    male
22.0
2            Cumings, Mrs. John Bradley (Florence Briggs Th...  female
38.0
3                                         Heikkinen, Miss. Laina  female
26.0
4                 Futrelle, Mrs. Jacques Heath (Lily May Peel)  female
35.0
5                                       Allen, Mr. William Henry    male
35.0
6                                               Moran, Mr. James    male
NaN
7                                        McCarthy, Mr. Timothy J    male
54.0
8                                Palsson, Master. Gosta Leonard    male
2.0
9            Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)  female
27.0
10                           Nasser, Mrs. Nicholas (Adele Achem)  female
14.0
11                               Sandstrom, Miss. Marguerite Rut  female
4.0
12                                     Bonnell, Miss. Elizabeth  female
58.0
13                               Saundercock, Mr. William Henry    male
20.0
14                                  Andersson, Mr. Anders Johan    male
39.0
15                          Vestrom, Miss. Hulda Amanda Adolfina  female
14.0
16                              Hewlett, Mrs. (Mary D Kingcome)   female
55.0
17                                          Rice, Master. Eugene    male
2.0
18                                 Williams, Mr. Charles Eugene    male
NaN
19           Vander Planke, Mrs. Julius (Emelia Maria Vande...  female
31.0
20                                       Masselmani, Mrs. Fatima  female
NaN

df.iloc[700:721,2:7]

                                                                Name     Sex
Age  \
PassengerId
```

```
701          Astor, Mrs. John Jacob (Madeleine Talmadge Force)  female
18.0
702                           Silverthorne, Mr. Spencer Victor    male
35.0
703                                       Barbara, Miss. Saiide  female
18.0
704                                       Gallagher, Mr. Martin    male
25.0
705                                    Hansen, Mr. Henrik Juul    male
26.0
706             Morley, Mr. Henry Samuel ("Mr Henry Marshall")    male
39.0
707                                Kelly, Mrs. Florence "Fannie"  female
45.0
708                            Calderhead, Mr. Edward Pennington    male
42.0
709                                       Cleaver, Miss. Alice  female
22.0
710          Moubarek, Master. Halim Gonios ("William George")    male
NaN
711           Mayne, Mlle. Berthe Antonine ("Mrs de Villiers")  female
24.0
712                                         Klaber, Mr. Herman    male
NaN
713                                    Taylor, Mr. Elmer Zebley    male
48.0
714                                  Larsson, Mr. August Viktor    male
29.0
715                                       Greenberg, Mr. Samuel    male
52.0
716                  Soholt, Mr. Peter Andreas Lauritz Andersen    male
19.0
717                               Endres, Miss. Caroline Louise  female
38.0
718                          Troutt, Miss. Edwina Celia "Winnie"  female
27.0
719                                         McEvoy, Mr. Michael    male
NaN
720                                Johnson, Mr. Malkolm Joackim    male
33.0
721                            Harper, Miss. Annie Jessie "Nina"  female
6.0

             SibSp  Parch
PassengerId
701              1      0
702              0      0
703              0      1
```

```
704                0       0
705                1       0
706                0       0
707                0       0
708                0       0
709                0       0
710                1       1
711                0       0
712                0       0
713                1       0
714                0       0
715                0       0
716                0       0
717                0       0
718                0       0
719                0       0
720                0       0
721                0       1
```

```
df.loc[21:100,['Survived','Age','Name']]
```

```
             Survived   Age                                      Name
PassengerId
21                  0  35.0                   Fynney, Mr. Joseph J
22                  1  34.0                  Beesley, Mr. Lawrence
23                  1  15.0            McGowan, Miss. Anna "Annie"
24                  1  28.0           Sloper, Mr. William Thompson
25                  0   8.0          Palsson, Miss. Torborg Danira
...               ...   ...                                      ...
96                  0   NaN            Shorney, Mr. Charles Joseph
97                  0  71.0              Goldschmidt, Mr. George B
98                  1  23.0        Greenfield, Mr. William Bertram
99                  1  34.0  Doling, Mrs. John T (Ada Julia Bone)
100                 0  34.0                       Kantor, Mr. Sinai

[80 rows x 3 columns]
```

```
df.loc[121:310,]
```

```
             Survived  Pclass  \
PassengerId
121                 0       2
122                 0       3
123                 0       2
124                 1       2
125                 0       1
...               ...     ...
306                 1       1
307                 1       1
308                 1       1
```

```
309                  0       2
310                  1       1

                                                          Name     Sex
Age  \
PassengerId

121                                  Hickman, Mr. Stanley George    male
21.00
122                                   Moore, Mr. Leonard Charles    male
NaN
123                                        Nasser, Mr. Nicholas    male
32.50
124                                         Webber, Miss. Susan  female
32.50
125                                  White, Mr. Percival Wayland    male
54.00
...                                                         ...     ...
...
306                               Allison, Master. Hudson Trevor    male
0.92
307                                     Fleming, Miss. Margaret  female
NaN
308          Penasco y Castellana, Mrs. Victor de Satode (M...  female
17.00
309                                       Abelson, Mr. Samuel    male
30.00
310                               Francatelli, Miss. Laura Mabel  female
30.00

             SibSp  Parch        Ticket      Fare    Cabin Embarked
PassengerId
121              2      0  S.O.C. 14879   73.5000      NaN        S
122              0      0     A4. 54510    8.0500      NaN        S
123              1      0        237736   30.0708      NaN        C
124              0      0         27267   13.0000     E101        S
125              0      1         35281   77.2875      D26        S
...            ...    ...           ...       ...      ...      ...
306              1      2        113781  151.5500  C22 C26        S
307              0      0         17421  110.8833      NaN        C
308              1      0      PC 17758  108.9000      C65        C
309              1      0     P/PP 3381   24.0000      NaN        C
310              0      0      PC 17485   56.9292      E36        C

[190 rows x 11 columns]

df.loc[:,['Survived','Age','Name']]

             Survived   Age
Name
```

```
PassengerId

1                    0  22.0                              Braund, Mr.
Owen Harris
2                    1  38.0  Cumings, Mrs. John Bradley (Florence
Briggs Th...
3                    1  26.0                               Heikkinen,
Miss. Laina
4                    1  35.0       Futrelle, Mrs. Jacques Heath (Lily
May Peel)
5                    0  35.0                             Allen, Mr.
William Henry
...                ...   ...
...
887                  0  27.0                               Montvila,
Rev. Juozas
888                  1  19.0                        Graham, Miss.
Margaret Edith
889                  0   NaN           Johnston, Miss. Catherine Helen
"Carrie"
890                  1  26.0                               Behr, Mr.
Karl Howell
891                  0  32.0                                 Dooley, Mr.
Patrick

[891 rows x 3 columns]
```

Checking Value Counts of Specific Columns

.value_counts() - It returns a Series containing counts of unique values.

```
df.Sex.value_counts()

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

df['Embarked'].value_counts()

Embarked
S    644
C    168
Q     77
Name: count, dtype: int64

df['Embarked'].value_counts(normalize = True)*100

Embarked
S    72.440945
```

```
C    18.897638
Q     8.661417
Name: proportion, dtype: float64
```

Combining & Comparing Two Columns

.crosstab() - It is used to compute a simple cross-tabulation of two factors.

```
pd.crosstab(df['Sex'],df['Survived']) #1st data becomes the rows and
2nd becomes the columns
```

```
Survived    0    1
Sex
female     81  233
male      468  109
```

```
pd.crosstab(df['Pclass'],df['Survived'], margins=True,
margins_name='Total')
```

```
Survived    0    1  Total
Pclass
1          80  136    216
2          97   87    184
3         372  119    491
Total     549  342    891
```

Getting Data Whose Age Is Greater Than 55

```
Agemorethen50 = df[df['Age']>55]
print("Shape: ",Agemorethen50.shape)
Agemorethen50
```

```
Shape:  (40, 11)
```

```
             Survived  Pclass  \
PassengerId
12                  1       1
34                  0       2
55                  0       1
95                  0       3
97                  0       1
117                 0       3
153                 0       3
171                 0       1
175                 0       1
196                 1       1
233                 0       2
253                 0       1
269                 1       1
276                 1       1
```

281	0	3
327	0	3
367	1	1
439	0	1
457	0	1
468	0	1
484	1	3
488	0	1
494	0	1
546	0	1
556	0	1
571	1	2
588	1	1
626	0	1
627	0	2
631	1	1
648	1	1
660	0	1
673	0	2
685	0	2
695	0	1
746	0	1
773	0	2
830	1	1
852	0	3
880	1	1

PassengerId	Name	Sex	Age
12	Bonnell, Miss. Elizabeth	female	58.0
34	Wheadon, Mr. Edward H	male	66.0
55	Ostby, Mr. Engelhart Cornelius	male	65.0
95	Coxon, Mr. Daniel	male	59.0
97	Goldschmidt, Mr. George B	male	71.0
117	Connors, Mr. Patrick	male	70.5
153	Meo, Mr. Alfonzo	male	55.5
171	Van der hoef, Mr. Wyckoff	male	61.0
175	Smith, Mr. James Clinch	male	56.0

```
196                                      Lurette, Miss. Elise  female
58.0
233                                   Sjostedt, Mr. Ernst Adolf    male
59.0
253                                   Stead, Mr. William Thomas    male
62.0
269              Graham, Mrs. William Thompson (Edith Junkins)  female
58.0
276                           Andrews, Miss. Kornelia Theodosia  female
63.0
281                                        Duane, Mr. Frank    male
65.0
327                                   Nysveen, Mr. Johan Hansen    male
61.0
367          Warren, Mrs. Frank Manley (Anna Sophia Atkinson)  female
60.0
439                                       Fortune, Mr. Mark    male
64.0
457                                   Millet, Mr. Francis Davis    male
65.0
468                                  Smart, Mr. John Montgomery    male
56.0
484                                     Turkula, Mrs. (Hedwig)  female
63.0
488                                     Kent, Mr. Edward Austin    male
58.0
494                                     Artagaveytia, Mr. Ramon    male
71.0
546                                Nicholson, Mr. Arthur Ernest    male
64.0
556                                       Wright, Mr. George    male
62.0
571                                       Harris, Mr. George    male
62.0
588                             Frolicher-Stehli, Mr. Maxmillian    male
60.0
626                                     Sutton, Mr. Frederick    male
61.0
627                               Kirkland, Rev. Charles Leonard    male
57.0
631                         Barkworth, Mr. Algernon Henry Wilson    male
80.0
648                          Simonius-Blumer, Col. Oberst Alfons    male
56.0
660                                   Newell, Mr. Arthur Webster    male
58.0
673                                  Mitchell, Mr. Henry Michael    male
70.0
685                             Brown, Mr. Thomas William Solomon    male
```

```
60.0
695                                          Weir, Col. John    male
60.0
746                              Crosby, Capt. Edward Gifford    male
70.0
773                                       Mack, Mrs. (Mary)  female
57.0
830                   Stone, Mrs. George Nelson (Martha Evelyn)  female
62.0
852                                     Svensson, Mr. Johan    male
74.0
880               Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)  female
56.0

             SibSp  Parch       Ticket      Fare        Cabin Embarked
PassengerId
12               0      0       113783   26.5500         C103        S
34               0      0   C.A. 24579   10.5000          NaN        S
55               0      1       113509   61.9792          B30        C
95               0      0       364500    7.2500          NaN        S
97               0      0     PC 17754   34.6542           A5        C
117              0      0       370369    7.7500          NaN        Q
153              0      0   A.5. 11206    8.0500          NaN        S
171              0      0       111240   33.5000          B19        S
175              0      0        17764   30.6958           A7        C
196              0      0     PC 17569  146.5208          B80        C
233              0      0       237442   13.5000          NaN        S
253              0      0       113514   26.5500          C87        S
269              0      1     PC 17582  153.4625         C125        S
276              1      0        13502   77.9583           D7        S
281              0      0       336439    7.7500          NaN        Q
327              0      0       345364    6.2375          NaN        S
367              1      0       110813   75.2500          D37        C
```

439	1	4	19950	263.0000	C23 C25 C27		S
457	0	0	13509	26.5500	E38		S
468	0	0	113792	26.5500	NaN		S
484	0	0	4134	9.5875	NaN		S
488	0	0	11771	29.7000	B37		C
494	0	0	PC 17609	49.5042	NaN		C
546	0	0	693	26.0000	NaN		S
556	0	0	113807	26.5500	NaN		S
571	0	0	S.W./PP 752	10.5000	NaN		S
588	1	1	13567	79.2000	B41		C
626	0	0	36963	32.3208	D50		S
627	0	0	219533	12.3500	NaN		Q
631	0	0	27042	30.0000	A23		S
648	0	0	13213	35.5000	A26		C
660	0	2	35273	113.2750	D48		C
673	0	0	C.A. 24580	10.5000	NaN		S
685	1	1	29750	39.0000	NaN		S
695	0	0	113800	26.5500	NaN		S
746	1	1	WE/P 5735	71.0000	B22		S
773	0	0	S.O./P.P. 3	10.5000	E77		S
830	0	0	113572	80.0000	B28		NaN
852	0	0	347060	7.7750	NaN		S
880	0	1	11767	83.1583	C50		C

Getting Data Whose Gender Is Equal To Male

```
Sexequaltomale = df[df['Sex'] == 'male']
print("Shape: ",Sexequaltomale.shape)
Sexequaltomale
```

```
Shape:  (577, 11)
```

```
             Survived  Pclass                               Name   Sex
Age  \
PassengerId

1                   0       3          Braund, Mr. Owen Harris  male
22.0
5                   0       3         Allen, Mr. William Henry  male
35.0
6                   0       3                 Moran, Mr. James  male
NaN
7                   0       1          McCarthy, Mr. Timothy J  male
54.0
8                   0       3  Palsson, Master. Gosta Leonard  male
2.0
...               ...     ...                              ...   ...
...
884                 0       2   Banfield, Mr. Frederick James  male
28.0
885                 0       3          Sutehall, Mr. Henry Jr  male
25.0
887                 0       2            Montvila, Rev. Juozas  male
27.0
890                 1       1            Behr, Mr. Karl Howell  male
26.0
891                 0       3              Dooley, Mr. Patrick  male
32.0

             SibSp  Parch            Ticket     Fare Cabin Embarked
PassengerId
1                1      0         A/5 21171   7.2500   NaN        S
5                0      0            373450   8.0500   NaN        S
6                0      0            330877   8.4583   NaN        Q
7                0      0             17463  51.8625   E46        S
8                3      1            349909  21.0750   NaN        S
...            ...    ...               ...      ...   ...      ...
884              0      0  C.A./SOTON 34068  10.5000   NaN        S
885              0      0   SOTON/OQ 392076   7.0500   NaN        S
887              0      0            211536  13.0000   NaN        S
890              0      0            111369  30.0000  C148        C
891              0      0            370376   7.7500   NaN        Q

[577 rows x 11 columns]
```

Getting Data Whose Age Is Greater Than Or Equal To 55 And Whose Gender Is Equal To Male And Who Survived

```
passenger = df[(df['Sex'] == 'male') & (df['Age'] >= 55) &
(df['Survived'] == 1)]
print('Shape: ',passenger.shape)
passenger
```

```
Shape:  (4, 11)
```

```
             Survived  Pclass                                     Name
Sex  \
PassengerId

571                 1       2                    Harris, Mr. George
male
588                 1       1      Frolicher-Stehli, Mr. Maxmillian
male
631                 1       1  Barkworth, Mr. Algernon Henry Wilson
male
648                 1       1   Simonius-Blumer, Col. Oberst Alfons
male

               Age  SibSp  Parch       Ticket  Fare Cabin Embarked
PassengerId
571          62.0      0      0  S.W./PP 752  10.5   NaN        S
588          60.0      1      1        13567  79.2   B41        C
631          80.0      0      0        27042  30.0   A23        S
648          56.0      0      0        13213  35.5   A26        C
```

Getting Data Whose Age Is In Between 20 To 40 And Whose Gender Is Equal To Male And Who Survived

```
passenger1 = df[(df['Age'] >= 20) & (df['Age'] <= 40) & (df['Sex'] ==
'female') & (df['Survived'] == 1)]
print('Shape: ',passenger1.shape)
passenger1
```

```
Shape:  (107, 11)
```

```
             Survived  Pclass  \
PassengerId
2                   1       1
3                   1       3
4                   1       1
9                   1       3
26                  1       3
...               ...     ...
843                 1       1
859                 1       3
867                 1       2
```

```
875                  1       2
881                  1       2

                                                          Name     Sex
Age  \
PassengerId

2            Cumings, Mrs. John Bradley (Florence Briggs Th...  female
38.0
3                                       Heikkinen, Miss. Laina  female
26.0
4                 Futrelle, Mrs. Jacques Heath (Lily May Peel)  female
35.0
9            Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)  female
27.0
26           Asplund, Mrs. Carl Oscar (Selma Augusta Emilia...  female
38.0
...                                                        ...     ...
...
843                                    Serepeca, Miss. Augusta  female
30.0
859                      Baclini, Mrs. Solomon (Latifa Qurban)  female
24.0
867                               Duran y More, Miss. Asuncion  female
27.0
875                      Abelson, Mrs. Samuel (Hannah Wizosky)  female
28.0
881                Shelley, Mrs. William (Imanita Parrish Hall)  female
25.0

             SibSp  Parch            Ticket     Fare Cabin Embarked
PassengerId
2                1      0          PC 17599  71.2833   C85        C
3                0      0  STON/O2. 3101282   7.9250   NaN        S
4                1      0            113803  53.1000  C123        S
9                0      2            347742  11.1333   NaN        S
26               1      5            347077  31.3875   NaN        S
...            ...    ...               ...      ...   ...      ...
843              0      0            113798  31.0000   NaN        C
859              0      3              2666  19.2583   NaN        C
867              1      0     SC/PARIS 2149  13.8583   NaN        C
875              1      0         P/PP 3381  24.0000   NaN        C
881              0      1            230433  26.0000   NaN        S

[107 rows x 11 columns]
```

Checking Null Values

.isnull() - Return a boolean same-sized object indicating if the values are NA.

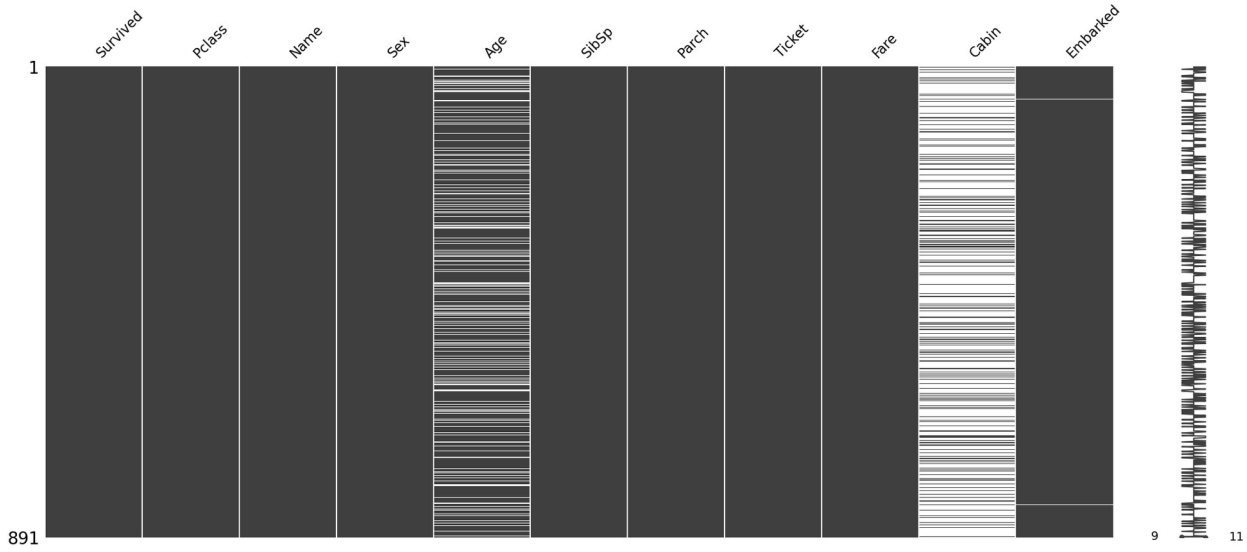
```
df.isnull().sum()
```

```
Survived      0
Pclass        0
Name          0
Sex           0
Age         177
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin       687
Embarked      2
dtype: int64
```

Missing Data Visualization Module For Python.

```
#!pip install missingno
import missingno as msno
msno.matrix(df)
```

```
<Axes: >
```



Dropping Null values

.dropna() - It allows the user to analyze and drop Rows/Columns with Null values in different ways.

```
df.dropna(how='any',subset=['Embarked'],inplace=True)
print(df.shape)
print(df.isnull().sum())
```

```
(889, 11)
Survived      0
Pclass        0
Name          0
Sex           0
Age         177
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin       687
Embarked      0
dtype: int64
```

Checking Statstics Details

```
print(df['Age'].describe())
print('----------------------------------')
print("Median of Age: ",df['Age'].median())
print("Mean of Age: ",df['Age'].mean())
print("Mode of Age: ",df['Age'].mode())
```

```
count    712.000000
mean      29.642093
std       14.492933
min        0.420000
25%       20.000000
50%       28.000000
75%       38.000000
max       80.000000
Name: Age, dtype: float64
----------------------------------
Median of Age:  28.0
Mean of Age:  29.64209269662921
Mode of Age:  0    24.0
Name: Age, dtype: float64
```

Filling Null Values With Mean

.fillna() -It let the user replace NaN values with some value of their own.

```
df['Age'].fillna(df['Age'].mean(), inplace = True)
print(df.shape)
print(df.isnull().sum())
```

```
(889, 11)
Survived      0
Pclass        0
Name          0
```

```
Sex            0
Age            0
SibSp          0
Parch          0
Ticket         0
Fare           0
Cabin        687
Embarked       0
dtype: int64
```

```
print(df['Cabin'].describe())
print('----------------------------------')
print("Mode of Cabin: ",df['Cabin'].mode())
print('----------------------------------')
print("First Mode of Cabin: ",df['Cabin'].mode()[0])
```

```
count          202
unique         146
top       B96 B98
freq             4
Name: Cabin, dtype: object
----------------------------------
Mode of Cabin:  0        B96 B98
1    C23 C25 C27
2             G6
Name: Cabin, dtype: object
----------------------------------
First Mode of Cabin:  B96 B98
```

Filling Null Values With Mode

```
df['Cabin'].fillna(df['Cabin'].mode()[0], inplace = True)
print(df.shape)
print(df.isnull().sum())
```

```
(889, 11)
Survived    0
Pclass      0
Name        0
Sex         0
Age         0
SibSp       0
Parch       0
Ticket      0
Fare        0
Cabin       0
Embarked    0
dtype: int64
```

Checking First Five Rows Of DataFrame

.head() -

- It returns a specified number of rows, string from the top.
- By default it returns the first 5 rows if a number is not specified.

```
df.head()
```

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

                                                          Name     Sex
Age  \
PassengerId

1                                       Braund, Mr. Owen Harris    male
22.0
2             Cumings, Mrs. John Bradley (Florence Briggs Th...  female
38.0
3                                        Heikkinen, Miss. Laina  female
26.0
4                 Futrelle, Mrs. Jacques Heath (Lily May Peel)  female
35.0
5                                     Allen, Mr. William Henry    male
35.0

             SibSp  Parch            Ticket     Fare    Cabin Embarked

PassengerId

1                1      0         A/5 21171   7.2500  B96 B98        S

2                1      0          PC 17599  71.2833      C85        C

3                0      0  STON/O2. 3101282   7.9250  B96 B98        S

4                1      0            113803  53.1000     C123        S

5                0      0            373450   8.0500  B96 B98        S
```

Checking Last Five Rows Of DataFrame

.tail() -

- It returns a specified number of rows, string from the bottom.

- By default it returns the last 5 rows if a number is not specified.

```
df.tail()

df['Age'].describe()

count    889.000000
mean      29.642093
std       12.968346
min        0.420000
25%       22.000000
50%       29.642093
75%       35.000000
max       80.000000
Name: Age, dtype: float64
```

Creating Bins

Use cut when you need to segment and sort data values into bins. This function is also useful for going from a continuous variable to a categorical variable.

```
PassengerAge = df['Age']
PassengerAge = PassengerAge.dropna()

bins = [PassengerAge.min(),15,21,60,PassengerAge.max()]
binlabels = ['Children', 'Adolescents', 'Adult', 'Senior']

categories = pd.cut(PassengerAge,bins,labels = binlabels)
categories.head(20)

PassengerId
1            Adult
2            Adult
3            Adult
4            Adult
5            Adult
6            Adult
7            Adult
8         Children
9            Adult
10        Children
11        Children
12           Adult
13     Adolescents
14           Adult
15        Children
16           Adult
17        Children
18           Adult
19           Adult
```

```
20          Adult
Name: Age, dtype: category
Categories (4, object): ['Children' < 'Adolescents' < 'Adult' <
'Senior']
```

Mapping The Created Bin To The DataFrame

```
df['AgeGroup'] = categories
df.tail(5)
```

```
             Survived  Pclass
Name  \
PassengerId

887                 0       2                     Montvila, Rev.
Juozas
888                 1       1              Graham, Miss. Margaret
Edith
889                 0       3  Johnston, Miss. Catherine Helen
"Carrie"
890                 1       1                     Behr, Mr. Karl
Howell
891                 0       3                       Dooley, Mr.
Patrick

                Sex        Age  SibSp  Parch      Ticket   Fare
Cabin  \
PassengerId

887            male  27.000000      0      0      211536  13.00  B96
B98
888          female  19.000000      0      0      112053  30.00
B42
889          female  29.642093      1      2  W./C. 6607  23.45  B96
B98
890            male  26.000000      0      0      111369  30.00
C148
891            male  32.000000      0      0      370376   7.75  B96
B98

            Embarked     AgeGroup
PassengerId
887                S        Adult
888                S  Adolescents
889                S        Adult
890                C        Adult
891                Q        Adult
```

```
pd.crosstab(df['AgeGroup'],df['Sex'], margins=True,
margins_name='Total')
```

```
Sex          female  male  Total
AgeGroup
Children         43    39     82
Adolescents      41    80    121
Adult           226   438    664
Senior            2    19     21
Total           312   576    888
```

Grouping And Comparing The Columns

.groupby() -

- A groupby operation involves some combination of splitting the object, applying a function, and combining the results.
- This can be used to group large amounts of data and compute operations on these groups.

```
df.groupby(['Survived','Sex','Pclass'])['Sex'].count()
```

```
Survived  Sex     Pclass
0         female  1           3
                  2           6
                  3          72
          male    1          77
                  2          91
                  3         300
1         female  1          89
                  2          70
                  3          72
          male    1          45
                  2          17
                  3          47
Name: Sex, dtype: int64
```

```
df.groupby(['AgeGroup','Sex','Survived'])['Survived'].count()
```

```
AgeGroup     Sex     Survived
Children     female  0            15
                     1            28
             male    0            19
                     1            20
Adolescents  female  0            12
                     1            29
             male    0            71
                     1             9
Adult        female  0            54
                     1           172
             male    0           361
                     1            77
Senior       female  0             0
```

```
                      1             2
             male     0            17
                      1             2
Name: Survived, dtype: int64
```

Exporting Dataset

```
df.to_csv('Updated_Titanic.csv')
```