In this analysis, we are going to deal only analyzing and exploring the datasets in a simple way.

- Here we are not going to drop any rows/columns,nor na values.
- also we are not going to do imputation here.
- No visualization too.
- this is just a pure exploration of datasets surfacely.

import dependencies/libraries.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
# %matplotlib inline ensures that the plots will be rendered within
the notebook itself.
```

Extract data from csv file

```
data=pd.read_csv("titanic_train.csv")
data
```

```
     PassengerId  Survived  Pclass  \
0              1         0       3
1              2         1       1
2              3         1       3
3              4         1       1
4              5         0       3
..           ...       ...     ...
886          887         0       2
887          888         1       1
888          889         0       3
889          890         1       1
890          891         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
..                                                  ...     ...   ...
...
886                              Montvila, Rev. Juozas    male  27.0
0
887                       Graham, Miss. Margaret Edith  female  19.0
0
888           Johnston, Miss. Catherine Helen "Carrie"  female   NaN
1
889                              Behr, Mr. Karl Howell    male  26.0
0
890                                Dooley, Mr. Patrick    male  32.0
0

     Parch            Ticket     Fare Cabin Embarked
0        0         A/5 21171   7.2500   NaN        S
1        0          PC 17599  71.2833   C85        C
2        0  STON/O2. 3101282   7.9250   NaN        S
3        0            113803  53.1000  C123        S
4        0            373450   8.0500   NaN        S
..     ...               ...      ...   ...      ...
886      0            211536  13.0000   NaN        S
887      0            112053  30.0000   B42        S
888      2        W./C. 6607  23.4500   NaN        S
889      0            111369  30.0000  C148        C
890      0            370376   7.7500   NaN        Q

[891 rows x 12 columns]

# from above dataset it is seen clear that the dataset has 891 rows
and 12 columns.

data.head()
# just check the first five top rows.
# here we found some null n/a values.

   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
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S

data.tail()
# just check the last bottom rows.

     PassengerId  Survived  Pclass
Name  \
886          887         0       2                     Montvila, Rev.
Juozas
887          888         1       1              Graham, Miss. Margaret
Edith
888          889         0       3  Johnston, Miss. Catherine Helen
"Carrie"
889          890         1       1                     Behr, Mr. Karl
Howell
890          891         0       3                       Dooley, Mr.
Patrick

        Sex   Age  SibSp  Parch      Ticket   Fare Cabin Embarked
886    male  27.0      0      0      211536  13.00   NaN        S
887  female  19.0      0      0      112053  30.00   B42        S
888  female   NaN      1      2  W./C. 6607  23.45   NaN        S
889    male  26.0      0      0      111369  30.00  C148        C
890    male  32.0      0      0      370376   7.75   NaN        Q
```

check the columns names

```
data.columns

Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age',
'SibSp',
```

```
       'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
      dtype='object')

# here we get columns name present in the dataset.
```

check the dimension

```
data.ndim
# this check the whole dataset dimension.

2

# from above we found dataset is of 2 dimension.

data['Survived'].ndim

1

# we just checked the dimension of individual columns i.e Survived
column so which is a series. thats why the dimension is 1.

# similarly check other columns also.
```

Shape of the dataset.

```
data.shape

(891, 12)

# shape gives the no of rows and columns present in the dataset.
# 891 no of rows and
# 12 columns
```

data types present in dataset.

```
data.dtypes

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

# dtypes gives the data type of each columns.
```

info regarding the dataset.

```
data.info()

<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

# here info gives us the information about no of rows and columns.
# Column Name present in dataset.
# data types of each columns
# null value present in columns.
# Age out of 891 714 is present and remaining 183 is null value.
# Cabin out of 891 204 is present and remaining 687 is null value.
# Embarked out of 891 889 is present and remaining 2 is null value.
```

filter categorical and numerical columns

```
categorical_columns=data.select_dtypes(include=['object']).columns
display(categorical_columns)

Index(['Name', 'Sex', 'Ticket', 'Cabin', 'Embarked'], dtype='object')

# above code gives us categorical columns in series form.
```

```
categorical_columns=data[data.select_dtypes(include=['object']).column
s]
display(categorical_columns)
# this code gives us in 2D form.
```

```
                                                    Name     Sex  \
0                              Braund, Mr. Owen Harris    male
1    Cumings, Mrs. John Bradley (Florence Briggs Th...  female
2                               Heikkinen, Miss. Laina  female
3         Futrelle, Mrs. Jacques Heath (Lily May Peel)  female
4                             Allen, Mr. William Henry    male
..                                                 ...     ...
886                              Montvila, Rev. Juozas    male
887                       Graham, Miss. Margaret Edith  female
888           Johnston, Miss. Catherine Helen "Carrie"  female
889                              Behr, Mr. Karl Howell    male
890                                Dooley, Mr. Patrick    male

               Ticket Cabin Embarked
0           A/5 21171   NaN        S
1            PC 17599   C85        C
2    STON/O2. 3101282   NaN        S
3              113803  C123        S
4              373450   NaN        S
..                ...   ...      ...
886            211536   NaN        S
887            112053   B42        S
888        W./C. 6607   NaN        S
889            111369  C148        C
890            370376   NaN        Q

[891 rows x 5 columns]
```

```
numerical_columns=data.select_dtypes(include=['number']).columns
display(numerical_columns)
```

```
Index(['PassengerId', 'Survived', 'Pclass', 'Age', 'SibSp', 'Parch',
'Fare'], dtype='object')
```

```
# above code gives us numerical columns in series form.
```

```
numerical_columns=data[data.select_dtypes(include=['number']).columns]
display(numerical_columns)
```

```
     PassengerId  Survived  Pclass   Age  SibSp  Parch     Fare
0              1         0       3  22.0      1      0   7.2500
1              2         1       1  38.0      1      0  71.2833
2              3         1       3  26.0      0      0   7.9250
3              4         1       1  35.0      1      0  53.1000
4              5         0       3  35.0      0      0   8.0500
..           ...       ...     ...   ...    ...    ...      ...
```

```
886          887         0       2  27.0      0      0  13.0000
887          888         1       1  19.0      0      0  30.0000
888          889         0       3   NaN      1      2  23.4500
889          890         1       1  26.0      0      0  30.0000
890          891         0       3  32.0      0      0   7.7500

[891 rows x 7 columns]
```

length of dataset

```
len(data)
```

```
891
```

```
# 891 is length of data i.e rows.
```

index

```
data.index
```

```
RangeIndex(start=0, stop=891, step=1)
```

```
# this provides information on the data is starting from 0 and
stops/ends at 891 with a step=1 value.
```

value counts

```
data.value_counts()
# it gives the value counts of all columns
```

```
PassengerId  Survived  Pclass  Name
Sex     Age   SibSp  Parch  Ticket    Fare      Cabin  Embarked
2            1         1       Cumings, Mrs. John Bradley (Florence
Briggs Thayer)   female  38.0  1      0      PC 17599  71.2833   C85
C           1
572          1         1       Appleton, Mrs. Edward Dale (Charlotte
Lamson)          female  53.0  2      0      11769     51.4792   C101
S           1
578          1         1       Silvey, Mrs. William Baird (Alice
Munger)             female  39.0  1      0      13507     55.9000
E44    S           1
582          1         1       Thayer, Mrs. John Borland (Marian
Longstreth Morris)  female  39.0  1      1      17421     110.8833
C68    C           1
584          0         1       Ross, Mr. John Hugo
```

```
male    36.0  0      0      13049     40.1250   A10    C           1

..
328          1         2       Ball, Mrs. (Ada E Hall)
female  36.0  0      0      28551     13.0000   D      S           1
330          1         1       Hippach, Miss. Jean Gertrude
female  16.0  0      1      111361    57.9792   B18    C           1
332          0         1       Partner, Mr. Austen
male    45.5  0      0      113043    28.5000   C124   S           1
333          0         1       Graham, Mr. George Edward
male    38.0  0      1      PC 17582  153.4625  C91    S           1
890          1         1       Behr, Mr. Karl Howell
male    26.0  0      0      111369    30.0000   C148   C           1
Length: 183, dtype: int64

data['Survived'].value_counts()
# it gives the value count of individual column i.e Survived column

0    549
1    342
Name: Survived, dtype: int64

# value count for survived column is 1 and 0.
#you can check for other columns too.
```

unique values

- it only gives the uniqueness/ unique value present in individual columns not as a whole.

```
data['Embarked'].unique()

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

# here embarked columns has three unique values i.e S , C, Q . Here
nan means not a unique value but a null value.

data['Pclass'].unique()

array([3, 1, 2], dtype=int64)

# Pclass has three unique value 3, 1, 2.
```

number of unique values

- it is same as unique
- but it gives us whole number.

```
data['Embarked'].nunique()
```

```
3

# nunique means number of unique values. from above, there are 3
unique values.
```

descriptive statisctics

```
data.describe()
```

```
       PassengerId    Survived      Pclass         Age       SibSp  \
count   891.000000  891.000000  891.000000  714.000000  891.000000
mean    446.000000    0.383838    2.308642   29.699118    0.523008
std     257.353842    0.486592    0.836071   14.526497    1.102743
min       1.000000    0.000000    1.000000    0.420000    0.000000
25%     223.500000    0.000000    2.000000   20.125000    0.000000
50%     446.000000    0.000000    3.000000   28.000000    0.000000
75%     668.500000    1.000000    3.000000   38.000000    1.000000
max     891.000000    1.000000    3.000000   80.000000    8.000000

             Parch        Fare
count  891.000000  891.000000
mean     0.381594   32.204208
std      0.806057   49.693429
min      0.000000    0.000000
25%      0.000000    7.910400
50%      0.000000   14.454200
75%      0.000000   31.000000
max      6.000000  512.329200
```

```
# describe function gives the descriptive statistics of only numerical
columns.
# count of all numerical columns is given as 891.............
# mean of every individual numerical columns.
# standard deviation of all indiviual numerical columns.
# minimum (min) values of every individual numerical columns
# 25% percentile: 25 percentage of data is below 223.
# 50 % percentile: 50 percentage of data is below 446.
# 75 % percentile: 75 percentage of data is below 668.5
# maximum (max) values of every individual numerical columns
# mean and standard deviation provides insight on central
tendency/average and dispersion of dataset.
# dispersion/spread of data.: it provides how much the data deviates
from mean.
# A low standard deviation indicates that the data points are close to
the mean,
# while a high standard deviation suggests that the data points are
more spread out from the mean.
```

```
# The one with the smaller standard deviation will have data points
more tightly clustered around the mean, indicating less variability.

data.describe(include='all')
```

```
        PassengerId    Survived      Pclass                          Name
Sex  \
count    891.000000  891.000000  891.000000                           891
891
unique          NaN         NaN         NaN                           891
2
top             NaN         NaN         NaN  Braund, Mr. Owen Harris
male
freq            NaN         NaN         NaN                             1
577
mean     446.000000    0.383838    2.308642                           NaN
NaN
std      257.353842    0.486592    0.836071                           NaN
NaN
min        1.000000    0.000000    1.000000                           NaN
NaN
25%      223.500000    0.000000    2.000000                           NaN
NaN
50%      446.000000    0.000000    3.000000                           NaN
NaN
75%      668.500000    1.000000    3.000000                           NaN
NaN
max      891.000000    1.000000    3.000000                           NaN
NaN

               Age       SibSp       Parch  Ticket        Fare
Cabin  \
count   714.000000  891.000000  891.000000     891  891.000000
204
unique         NaN         NaN         NaN     681         NaN
147
top            NaN         NaN         NaN  347082         NaN  B96
B98
freq           NaN         NaN         NaN       7         NaN
4
mean     29.699118    0.523008    0.381594     NaN   32.204208
NaN
std      14.526497    1.102743    0.806057     NaN   49.693429
NaN
min       0.420000    0.000000    0.000000     NaN    0.000000
NaN
25%      20.125000    0.000000    0.000000     NaN    7.910400
NaN
50%      28.000000    0.000000    0.000000     NaN   14.454200
NaN
```

```
75%      38.000000    1.000000    0.000000     NaN   31.000000
NaN
max      80.000000    8.000000    6.000000     NaN  512.329200
NaN
```

```
       Embarked
count       889
unique        3
top           S
freq        644
mean        NaN
std         NaN
min         NaN
25%         NaN
50%         NaN
75%         NaN
max         NaN
```

```
# this include='all' provides all insight on descriptive statistics of
both numerical and categorical columns.
# but this is not relevant for our problem.
# it is not relevant because you can see there are null values and
Letters present which donot give insights to our data here.
```

```
data.describe(include='object')
```

```
                            Name   Sex  Ticket    Cabin Embarked
count                        891   891     891      204      889
unique                       891     2     681      147        3
top     Braund, Mr. Owen Harris  male  347082  B96 B98        S
freq                           1   577       7        4      644
```

```
# here include='object' provides insight of categorical columns only.
```

```
data['Survived'].mean()
```

```
0.3838383838383838
```

```
data['Embarked'].mode()[0]
```

```
'S'
```

```
data['Embarked'].mode()
```

```
0    S
Name: Embarked, dtype: object
```

```
data['Fare'].max()
```

```
512.3292
```

```
data['Fare'].min()
```

```
0.0

# you can do yourself and check other statistical function with the
help of this.
```

to change case of letter format in dataframe.

- uppercase to lowercase and viceversa.

```
# i am chaging the data case in the dataframe.

data['Sex']=data['Sex'].str.upper()
data.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
0      0         A/5 21171   7.2500   NaN        S
1      0          PC 17599  71.2833   C85        C
2      0  STON/O2. 3101282   7.9250   NaN        S
3      0            113803  53.1000  C123        S
4      0            373450   8.0500   NaN        S
```

```
# now see the above Sex column[previously in sex column: male was in
lowercase now it is changed to uppercase as MALE]

# using swap to change case

data['Embarked']=data['Embarked'].str.swapcase()
data.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
0      0         A/5 21171   7.2500   NaN        s
1      0          PC 17599  71.2833   C85        c
2      0  STON/O2. 3101282   7.9250   NaN        s
3      0            113803  53.1000  C123        s
4      0            373450   8.0500   NaN        s
```

```
# previously in Embarked, S C, were in uppercase now it is in lower
case by using swap case.
```

Replace

```
data['Sex']=data['Sex'].replace({'MALE':'male','FEMALE':'female'})
data.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
0      0         A/5 21171   7.2500   NaN        s
1      0          PC 17599  71.2833   C85        c
2      0  STON/O2. 3101282   7.9250   NaN        s
3      0            113803  53.1000  C123        s
4      0            373450   8.0500   NaN        s
```

```
# here we change the uppercase value to lower case in Sex column
# we can change to other values too.

data=pd.DataFrame(data.rename(columns={
    'PassengerId':'PASSENGERID',
    'Survived':'SURVIVED'
}))
data.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
0      0         A/5 21171   7.2500   NaN        s
1      0          PC 17599  71.2833   C85        c
2      0  STON/O2. 3101282   7.9250   NaN        s
3      0            113803  53.1000  C123        s
4      0            373450   8.0500   NaN        s
```

set index

- previously index was set as PassengerId.
- if we want to change the index to other columns then

```
data.set_index('Name')
```

```
                                                    PASSENGERID
SURVIVED  \
Name

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

                                                    Pclass     Sex
Age  \
Name

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

```
Montvila, Rev. Juozas                                        2    male
27.0
Graham, Miss. Margaret Edith                                 1  female
19.0
Johnston, Miss. Catherine Helen "Carrie"                     3  female
NaN
Behr, Mr. Karl Howell                                        1    male
26.0
Dooley, Mr. Patrick                                          3    male
32.0

                                                    SibSp  Parch  \
Name
Braund, Mr. Owen Harris                                 1      0
Cumings, Mrs. John Bradley (Florence Briggs Tha...      1      0
Heikkinen, Miss. Laina                                  0      0
Futrelle, Mrs. Jacques Heath (Lily May Peel)            1      0
Allen, Mr. William Henry                                0      0
...                                                   ...    ...
Montvila, Rev. Juozas                                   0      0
Graham, Miss. Margaret Edith                            0      0
Johnston, Miss. Catherine Helen "Carrie"                1      2
Behr, Mr. Karl Howell                                   0      0
Dooley, Mr. Patrick                                     0      0

                                                           Ticket
Fare  \
Name

Braund, Mr. Owen Harris                                  A/5 21171
7.2500
Cumings, Mrs. John Bradley (Florence Briggs Tha...        PC 17599
71.2833
Heikkinen, Miss. Laina                                STON/O2. 3101282
7.9250
Futrelle, Mrs. Jacques Heath (Lily May Peel)                113803
53.1000
Allen, Mr. William Henry                                    373450
8.0500
...                                                            ...
...
Montvila, Rev. Juozas                                       211536
13.0000
Graham, Miss. Margaret Edith                                112053
30.0000
Johnston, Miss. Catherine Helen "Carrie"                 W./C. 6607
23.4500
Behr, Mr. Karl Howell                                       111369
30.0000
Dooley, Mr. Patrick                                         370376
```

```
7.7500
```

```
                                                Cabin Embarked
Name
Braund, Mr. Owen Harris                           NaN        s
Cumings, Mrs. John Bradley (Florence Briggs Tha...   C85        c
Heikkinen, Miss. Laina                            NaN        s
Futrelle, Mrs. Jacques Heath (Lily May Peel)      C123        s
Allen, Mr. William Henry                          NaN        s
...                                               ...      ...
Montvila, Rev. Juozas                             NaN        s
Graham, Miss. Margaret Edith                      B42        s
Johnston, Miss. Catherine Helen "Carrie"          NaN        s
Behr, Mr. Karl Howell                             C148        c
Dooley, Mr. Patrick                               NaN        q

[891 rows x 11 columns]
```

```
# now see Name is set as index/unique identifer. But here i am not
using drop, and inplace=True.
# if you need another index permanently then use inplace=True.
```

Pandas Profiling report

```
import ydata_profiling

data.profile_report()
```


```
{"model_id":"7af6f8524a364da495219372615cdf3a","version_major":2,"vers
ion_minor":0}

{"model_id":"891b67c37bc7402b99e8cdd91711a48f","version_major":2,"vers
ion_minor":0}

{"model_id":"3584e490f2d14ebbbb20a7ff2ea010f1","version_major":2,"vers
ion_minor":0}
```


```
<IPython.core.display.HTML object>
```