Pandas complete notes Lab2

Analayze data:

In [51]:
```
import pandas as pd
```

In [2]:
```
data = pd.read_csv("titanic.csv")
```

In [3]:
```
data.head()
```

Out[3]:

	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

To manually store data in a table, create a DataFrame.

in python,The dictionary keys will be used as column headers and the values in each list as columns of the DataFrame. A DataFrame is a 2-dimensional data structure that can store data.

In [4]:
```
a = pd.DataFrame({
    "Name": [
        "Anaya",
        "zahra",
        "Fatima"
    ],
    "Degree": [
        "BSIT",
        "BSCS",
        "BSSE"
    ]
})

print(a)
```

```
     Name Degree
0   Anaya   BSIT
1   zahra   BSCS
2  Fatima   BSSE
```

In [5]:
```
# When selecting a single column of a pandas DataFrame,use the column label in between square brackets [].
a["Name"]
```

Out[5]:
```
0     Anaya
1     zahra
2    Fatima
Name: Name, dtype: object
```

we can also create a series also called column

In [6]:
```
a=pd.Series([11,22,33], name="age")
print(a)                                  # A pandas Series has no column labels, as it is just a single column of a DataFrame.
                                         # A Series does have row labels.
```

```
0    11
1    22
2    33
Name: age, dtype: int64
```

In [7]:
```
a.max()   # for series (syntax)
```

Out[7]:
```
33
```

In [8]:
```
data["Age"].max()   # for dataframe (syntax)
```

Out[8]:
```
80.0
```

In [9]:
```
data.describe()   # show all numerical values from dataset
```

Out[9]:

	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

Note:

-A table of data is stored as a pandas DataFrame
-Each column in a DataFrame is a Series

In [10]:
```
data.head()
```

Out[10]:

	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

In [11]:
```
data.tail()
```

Out[11]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00	NaN	S
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 [12]:
```
data.dtypes # dtypes is an attribute of a DataFrame and Series so do not need brackets
```

Out[12]:
```
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 [13]:
```
#The method info() provides technical information about a DataFrame we use brackets here bcz describe something.
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
```

In [14]:
```
a=data["Age"]   # to select a single column, use square brackets
print(a)
```

```
0      22.0
1      38.0
2      26.0
3      35.0
4      35.0
       ...
886    27.0
887    19.0
888     NaN
889    26.0
890    32.0
Name: Age, Length: 891, dtype: float64
```

As a single column is selected, the returned object is a pandas Series.

In [15]:
```
type(data["Age"])  # output in series
```

Out[15]:
```
pandas.core.series.Series
```

Select multiple columns

To select multiple columns, use a list of column names within the selection brackets [ ] The inner square brackets define a Python list with column names, whereas the outer brackets are used to select the data from a pandas DataFrame

In [16]:
```
multiple_columns=data[["Age", "Fare"]]
print(multiple_columns)
```

```
      Age     Fare
0    22.0   7.2500
1    38.0  71.2833
2    26.0   7.9250
3    35.0  53.1000
4    35.0   8.0500
..    ...      ...
886  27.0  13.0000
887  19.0  30.0000
888   NaN  23.4500
889  26.0  30.0000
890  32.0   7.7500

[891 rows x 2 columns]
```

The selection returned a DataFrame with 891 rows and 2 columns. Remember, a DataFrame is 2-dimensional with both a row and column dimension.

In [17]:
```
type(data[["Age","Fare"]])
```

Out[17]:
```
pandas.core.frame.DataFrame
```

In [18]:
```
data.shape
```

Out[18]:
```
(891, 12)
```

In [19]:
```
data[["Age","Fare"]].shape
```

Out[19]: (891, 2)

Conditional operator

In [20]:
```
data[data["Age"]>35]
```

Out[20]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
11	12	1	1	Bonnell, Miss. Elizabeth	female	58.0	0	0	113783	26.5500	C103	S
13	14	0	3	Andersson, Mr. Anders Johan	male	39.0	1	5	347082	31.2750	NaN	S
15	16	1	2	Hewlett, Mrs. (Mary D Kingcome)	female	55.0	0	0	248706	16.0000	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
865	866	1	2	Bystrom, Mrs. (Karolina)	female	42.0	0	0	236852	13.0000	NaN	S
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S
873	874	0	3	Vander Cruyssen, Mr. Victor	male	47.0	0	0	345765	9.0000	NaN	S
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250	NaN	Q

217 rows × 12 columns

In [21]:
```
data["Age"]>35
```

Out[21]:
```
0      False
1       True
2      False
3      False
4      False
       ...
886    False
887    False
888    False
889    False
890    False
Name: Age, Length: 891, dtype: bool
```

In [22]:
```
data[data["Age"]<=35]
```

Out[22]:

	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	7.250	NaN	S
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.925	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.100	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.050	NaN	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.075	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.050	NaN	S
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.000	B42	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.750	NaN	Q

497 rows × 12 columns

isin()

conditional expression, the isin() conditional function returns a True

(works lika or operator)

In [23]:
```
data[data["Pclass"].isin([2,3])]
```

Out[23]:

	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
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	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
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
...	...	...	...	...	...	...	...	...	...	...	...	...
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.0500	NaN	S
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250	NaN	Q
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	S
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

675 rows × 12 columns

In [24]:
```
data[(data["Pclass"]==2) | (data["Pclass"]==3)].head(5)      #or operator
```

Out[24]:

	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
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	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
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S

In [25]:
```
data[(data["Pclass"]==2) & (data["Pclass"]==3)].head(5)    # and operator
```

Out[25]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked

In [26]:
```
data[(data["Pclass"]==3) & (data["Pclass"]==3)].head(5)
```

Out[26]:

	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
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	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
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S

notna()

The notna() conditional function returns a True for each row the values are not a Null value.

In [27]:
```
data[data["Age"].notna()]
```

Out[27]:

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.1250	NaN	Q
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.0000	NaN	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.7500	NaN	Q

714 rows × 12 columns

In [28]:
```
data[data["Age"].notna()].shape
```

Out[28]:
```
(714, 12)
```

loc & iloc operators

loc:

- Select specific rows and/or columns using loc when using the row and column names.

In [29]:
```
data.loc[data["Age"] > 35, "Name"]
```

Out[29]:
```
1      Cumings, Mrs. John Bradley (Florence Briggs Th...
6                               McCarthy, Mr. Timothy J
11                             Bonnell, Miss. Elizabeth
13                           Andersson, Mr. Anders Johan
15                       Hewlett, Mrs. (Mary D Kingcome)
                              ...
865                             Bystrom, Mrs. (Karolina)
871     Beckwith, Mrs. Richard Leonard (Sallie Monypeny)
873                           Vander Cruyssen, Mr. Victor
879          Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)
885                   Rice, Mrs. William (Margaret Norton)
Name: Name, Length: 217, dtype: object
```

Set "Pclass" = 45 in row 3: (column name pclass)

In [30]:
```
data.loc[3, 'Pclass'] = 45
```

In [31]:
```
data.head()
```

Out[31]:

	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	45	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

Delete rows where "Age" is higher than 38:

In [32]:
```
for x in data.index:
  if data.loc[x, "Age"] > 38:
    data.drop(x, inplace = True)
```

In [33]:
```
data.head()
```

Out[33]:

	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	45	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

iloc:

```
-Select specific rows and/or columns using iloc when using the positions in the table.
-for example: [9:15,2:5]
-9:15 for rows 9 include and 15 exclude
-after comma 2:5 means colums 2 exclude and 5 include
```

In [34]:
```
data.iloc[9:15,2:5]
```

Out[34]:

	Pclass	Name	Sex
10	3	Sandstrom, Miss. Marguerite Rut	female
12	3	Saundercock, Mr. William Henry	male
14	3	Vestrom, Miss. Hulda Amanda Adolfina	female
16	3	Rice, Master. Eugene	male
17	2	Williams, Mr. Charles Eugene	male
18	3	Vander Planke, Mrs. Julius (Emelia Maria Vande...	female

You can assign new values to a selection based on loc/iloc.

In [35]:
```
data.iloc[1:5, 3]="write something"
```

In [36]:
```
data.head()
```

Out[36]:

	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	write something	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	write something	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	45	write something	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	write something	male	35.0	0	0	373450	8.0500	NaN	S

Data cleaning:

Data cleaning means fixing bad data in your data set.Bad data could be:

```
-Empty cells
-Data in wrong format
-Wrong data
-Duplicates
```

Dropna():

By default, the dropna() method returns a new DataFrame, and will not change the original.

In [37]:
```
data.dropna()       # Return a new Data Frame with no empty cells
```

Out[37]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	write something	female	38.0	1	0	PC 17599	71.2833	C85	C
3	4	1	45	write something	female	35.0	1	0	113803	53.1000	C123	S
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S
21	22	1	2	Beesley, Mr. Lawrence	male	34.0	0	0	248698	13.0000	D56	S
23	24	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.5000	A6	S
...	...	...	...	...	...	...	...	...	...	...	...	...
853	854	1	1	Lines, Miss. Mary Conover	female	16.0	0	1	PC 17592	39.4000	D28	S
867	868	0	1	Roebling, Mr. Washington Augustus II	male	31.0	0	0	PC 17590	50.4958	A24	S
872	873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0000	B51 B53 B55	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0000	C148	C

109 rows × 12 columns

The dropna(inplace = True) will NOT return a new DataFrame, but it will remove all rows containing NULL values from the original DataFrame.

In [38]:
```
data.dropna(inplace=True)
```

In [39]:
```
data.head()
```

Out[39]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	write something	female	38.0	1	0	PC 17599	71.2833	C85	C
3	4	1	45	write something	female	35.0	1	0	113803	53.1000	C123	S
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S
21	22	1	2	Beesley, Mr. Lawrence	male	34.0	0	0	248698	13.0000	D56	S
23	24	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.5000	A6	S

Fillna():

The fillna() method allows us to replace empty cells with a value

In [40]:
```
#Replace NULL values with the number 130:
data.fillna(130, inplace = True)
```

In [41]:
```
data.tail(3)
```

Out[41]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
872	873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0	B51 B53 B55	S
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0	B42	S
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.0	C148	C

In [42]:
```
#Replace NULL values in the "Calories" columns with the number 130:
data["Age"].fillna(13, inplace = True)
```

Mean:

In [43]:
```
data["Survived"].mean()
```

Out[43]:
```
0.7614678899082569
```

Median:

Median = the value in the middle, after you have sorted all values ascending.

In [44]:
```
data["Age"].median()
```

Out[44]:
```
27.0
```

Mode:

Mode = the value that appears most frequently.

In [45]:
```
data["Pclass"].mode()
```

Out[45]:
```
0    1
Name: Pclass, dtype: int64
```

In [46]:
```
data.shape
```

Out[46]:
```
(109, 12)
```

Remove duplicates:

In [48]:
```
#Remove all duplicates:
data.drop_duplicates(inplace = True)
```

In [49]:
```
data.head()
```

Out[49]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
1	2	1	1	write something	female	38.0	1	0	PC 17599	71.2833	C85	C
3	4	1	45	write something	female	35.0	1	0	113803	53.1000	C123	S
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	PP 9549	16.7000	G6	S
21	22	1	2	Beesley, Mr. Lawrence	male	34.0	0	0	248698	13.0000	D56	S
23	24	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.5000	A6	S

In [50]:
```
data.shape
```

Out[50]:
```
(109, 12)
```