

```
In [1]: import numpy as np
import pandas as pd
```

```
In [3]: df=pd.read_csv("titanic-training-data.csv")
```

```
In [4]: df.head(10)
```

Out[4]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	237736	30.0708

In [5]: `df.tail(10)`

Out[5]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
881	882	0	3	Markun, Mr. Johann	male	33.0	0	0	349257	7.89
882	883	0	3	Dahlberg, Miss. Gerda Ulrika	female	22.0	0	0	7552	10.51
883	884	0	2	Banfield, Mr. Frederick James	male	28.0	0	0	C.A./SOTON 34068	10.51
884	885	0	3	Sutehall, Mr. Henry Jr	male	25.0	0	0	SOTON/OQ 392076	7.09
885	886	0	3	Rice, Mrs. William (Margaret Norton)	female	39.0	0	5	382652	29.12
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536	13.00
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.00
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.45
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369	30.00
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376	7.73

In [6]: `df.sample(10)`

Out[6]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare
830	831	1	3	Yasbeck, Mrs. Antoni (Selini Alexander)	female	15.0	1	0	2659	14.4542
315	316	1	3	Nilsson, Miss. Helmina Josefina	female	26.0	0	0	347470	7.8542
345	346	1	2	Brown, Miss. Amelia "Mildred"	female	24.0	0	0	248733	13.0000
552	553	0	3	O'Brien, Mr. Timothy	male	NaN	0	0	330979	7.8292
740	741	1	1	Hawksford, Mr. Walter James	male	NaN	0	0	16988	30.0000
472	473	1	2	West, Mrs. Edwy Arthur (Ada Mary Worth)	female	33.0	1	2	C.A. 34651	27.7500
538	539	0	3	Risien, Mr. Samuel Beard	male	NaN	0	0	364498	14.5000
340	341	1	2	Navratil, Master. Edmond Roger	male	2.0	1	1	230080	26.0000
848	849	0	2	Harper, Rev. John	male	28.0	0	1	248727	33.0000
485	486	0	3	Lefebvre, Miss. Jeannie	female	NaN	3	1	4133	25.4667

In [7]: `df.shape`

Out[7]: (891, 12)

In [8]: `df.columns`

Out[8]: Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex', 'Age', 'SibSp', 'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'], dtype='object')

```
In [9]: df.dtypes
```

```
Out[9]: 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 [10]: df.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 [12]: df.isnull().sum()
```

```
Out[12]: PassengerId      0
Survived      0
Pclass        0
Name          0
Sex           0
Age           177
SibSp         0
Parch         0
Ticket        0
Fare          0
Cabin         687
Embarked      2
dtype: int64
```

In [13]: `df.describe()`

Out[13]:

	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 [14]: `df.describe(include="all")`

Out[14]:

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch
count	891.000000	891.000000	891.000000	891	891	714.000000	891.000000	891.000000
unique	NaN	NaN	NaN	891	2	NaN	NaN	NaN
top	NaN	NaN	NaN	Braund, Mr. Owen Harris	male	NaN	NaN	NaN
freq	NaN	NaN	NaN	1	577	NaN	NaN	NaN
mean	446.000000	0.383838	2.308642	NaN	NaN	29.699118	0.523008	0.381594
std	257.353842	0.486592	0.836071	NaN	NaN	14.526497	1.102743	0.806057
min	1.000000	0.000000	1.000000	NaN	NaN	0.420000	0.000000	0.000000
25%	223.500000	0.000000	2.000000	NaN	NaN	20.125000	0.000000	0.000000
50%	446.000000	0.000000	3.000000	NaN	NaN	28.000000	0.000000	0.000000
75%	668.500000	1.000000	3.000000	NaN	NaN	38.000000	1.000000	0.000000
max	891.000000	1.000000	3.000000	NaN	NaN	80.000000	8.000000	6.000000

In [15]: `ds=pd.read_csv("World Happiness Report.csv")`

In [16]: `ds.head(10)`

Out[16]:

	Country	Happiness Rank	Happiness Score	Economy	Family	Health	Freedom	Generosity	C
0	Norway	1	7.537	1.616463	1.533524	0.796667	0.635423	0.362012	
1	Denmark	2	7.522	1.482383	1.551122	0.792566	0.626007	0.355280	
2	Iceland	3	7.504	1.480633	1.610574	0.833552	0.627163	0.475540	
3	Switzerland	4	7.494	1.564980	1.516912	0.858131	0.620071	0.290549	
4	Finland	5	7.469	1.443572	1.540247	0.809158	0.617951	0.245483	
5	Netherlands	6	7.377	1.503945	1.428939	0.810696	0.585384	0.470490	
6	Canada	7	7.316	1.479204	1.481349	0.834558	0.611101	0.435540	
7	New Zealand	8	7.314	1.405706	1.548195	0.816760	0.614062	0.500005	
8	Sweden	9	7.284	1.494387	1.478162	0.830875	0.612924	0.385399	
9	Australia	10	7.284	1.484415	1.510042	0.843887	0.601607	0.477699	

In [17]: `ds.tail()`

Out[17]:

	Country	Happiness Rank	Happiness Score	Economy	Family	Health	Freedom	Generosity	C
148	Rwanda	151	3.471	0.368746	0.945707	0.326425	0.581844	0.252756	
149	Syria	152	3.462	0.777153	0.396103	0.500533	0.081539	0.493664	
150	Tanzania	153	3.349	0.511136	1.041990	0.364509	0.390018	0.354256	
151	Burundi	154	2.905	0.091623	0.629794	0.151611	0.059901	0.204435	
152	Central African Republic	155	2.693	0.000000	0.000000	0.018773	0.270842	0.280876	

In [19]: `ds.sample()`

Out[19]:

	Country	Happiness Rank	Happiness Score	Economy	Family	Health	Freedom	Generosity	Co
87	Portugal	89	5.195	1.315175	1.367043	0.795844	0.498465	0.095103	0

In [20]: `ds.shape`

Out[20]: (153, 12)

In [21]: `ds.columns`

Out[21]: Index(['Country', 'Happiness Rank', 'Happiness Score', 'Economy', 'Family',
 'Health', 'Freedom', 'Generosity', 'Corruption', 'Dystopia',
 'Job Satisfaction', 'Region'],
 dtype='object')

In [23]: `df.dtypes`

Out[23]: 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 [24]: `df.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 [25]: `ds.isnull().sum()`

```
Out[25]: Country          0
Happiness Rank          0
Happiness Score          0
Economy                 0
Family                  0
Health                  0
Freedom                 0
Generosity              0
Corruption              0
Dystopia                 0
Job Satisfaction         2
Region                  0
dtype: int64
```

In [27]: `ds.describe()`

```
Out[27]:
```

	Happiness Rank	Happiness Score	Economy	Family	Health	Freedom	Generosity
count	153.000000	153.000000	153.000000	153.000000	153.000000	153.000000	153.000000
mean	78.169935	5.349281	0.982433	1.186630	0.550117	0.408489	0.245324
std	45.008741	1.134997	0.421901	0.288441	0.237769	0.150744	0.134395
min	1.000000	2.693000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	40.000000	4.497000	0.659517	1.041990	0.364509	0.300741	0.153075
50%	78.000000	5.279000	1.064578	1.251826	0.606042	0.437454	0.231503
75%	117.000000	6.098000	1.315175	1.416404	0.719217	0.518631	0.322228
max	155.000000	7.537000	1.870766	1.610574	0.949492	0.658249	0.838075

In [28]: `ds.describe(include="all")`

```
Out[28]:
```

	Country	Happiness Rank	Happiness Score	Economy	Family	Health	Freedom	Generosity
count	153	153.000000	153.000000	153.000000	153.000000	153.000000	153.000000	153.000000
unique	153	NaN	NaN	NaN	NaN	NaN	NaN	NaN
top	Norway	NaN	NaN	NaN	NaN	NaN	NaN	NaN
freq	1	NaN	NaN	NaN	NaN	NaN	NaN	NaN
mean	NaN	78.169935	5.349281	0.982433	1.186630	0.550117	0.408489	0.245324
std	NaN	45.008741	1.134997	0.421901	0.288441	0.237769	0.150744	0.134395
min	NaN	1.000000	2.693000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	NaN	40.000000	4.497000	0.659517	1.041990	0.364509	0.300741	0.153075
50%	NaN	78.000000	5.279000	1.064578	1.251826	0.606042	0.437454	0.231503
75%	NaN	117.000000	6.098000	1.315175	1.416404	0.719217	0.518631	0.322228
max	NaN	155.000000	7.537000	1.870766	1.610574	0.949492	0.658249	0.838075

