Titanic dataset investigation with Pandas

Stages:

1. Primary Dataset analysis
2. Filtering Data
3. Merging Dataframes
4. Analytics
5. Data Visualization
6. Changing Data

```
import pandas as pd
```

```
df = pd.read_csv('https://raw.githubusercontent.com/datasciencedojo/datasets/master/titanic.csv
```

Primary Dataset analysis

```
type(df)
```

```
pandas.core.frame.DataFrame
```

Class Dataframe is two-dimensional (columns and rows) tabular data

```
df
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450

```
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
```

```
df.shape
```

```
(891, 12)
```

891 rows and 12 columns

```
df.columns
```

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

```
# First 5 entries
df.head(5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450

```
# Last 5 entries
df.tail(5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607
889	890	1	1	Behr, Mr. Karl Howell	male	26.0	0	0	111369
890	891	0	3	Dooley, Mr. Patrick	male	32.0	0	0	370376

```
# Types of columns
df.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
```

```
df['Name']
```

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

```
type(df['Name'])
```

```
pandas.core.series.Series
```

Class Series is a one-dimensional array

```
df['Name'].shape
```

```
(891,)
```

2. Filtering Data

```
# Name, Age and Sex of 5 first passengers
df[['Name', 'Age', 'Sex']].head(5)
```

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

```
# Name and Age of the 1st, 91st and 202nd passenger by the Index
df.loc[[0, 90, 201], ['Name', 'Age']]
```

	Name	Age
0	Braund, Mr. Owen Harris	22.0
90	Christmann, Mr. Emil	29.0
201	Sage, Mr. Frederick	NaN

```
# Return first 5 columns for Index [10:21]
df.iloc[10:21, :4]
```

	PassengerId	Survived	Pclass	Name
10	11	1	3	Sandstrom, Miss. Marguerite Rut
11	12	1	1	Bonnell, Miss. Elizabeth
12	13	0	3	Saundercock, Mr. William Henry
13	14	0	3	Andersson, Mr. Anders Johan
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina
15	16	1	2	Hewlett, Mrs. (Mary D Kingcome)
16	17	0	3	Rice, Master. Eugene
17	18	1	2	Williams, Mr. Charles Eugene
18	19	0	3	Vander Planke, Mrs. Julius (Emelia Maria Vande...
19	20	1	3	Masselmani, Mrs. Fatima
20	21	0	2	Fynney, Mr. Joseph J

```
# All passengers under 18 years old using Boolean Mask
df[df['Age'] < 18]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticke
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	34990
9	10	1	2	Nasser, Mrs. Nicholas (Adele Achem)	female	14.0	1	0	23773
10	11	1	3	Sandstrom, Miss. Marguerite Rut	female	4.0	1	1	P 954
14	15	0	3	Vestrom, Miss. Hulda Amanda Adolfina	female	14.0	0	0	35040
16	17	0	3	Rice, Master. Eugene	male	2.0	4	1	38265

```
# Are there 80 year old people among the passengers?
df[df['Age'].isin([80])]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	F
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0	27042	3

```
# Passengers either younger than 1 or older than 70 years old
df[(df['Age'] <= 1) | (df['Age'] >= 70)]
```

381	382	1	3	Nakid, Miss. Maria ("Mary")	female	1.00	0	2	
386	387	0	3	Goodwin, Master. Sidney Leonard	male	1.00	5	2	C
469	470	1	3	Baclini, Miss. Helene Barbara	female	0.75	2	1	
493	494	0	1	Artagaveytia, Mr. Ramon	male	71.00	0	0	PC
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.00	0	0	
644	645	1	3	Baclini, Miss. Eugenie	female	0.75	2	1	
672	673	0	2	Mitchell, Mr. Henry Michael	male	70.00	0	0	C.A
745	746	0	1	Crosby, Capt. Edward Gifford	male	70.00	1	1	WE
755	756	1	2	Hamalainen, Master. Viljo	male	0.67	1	1	
788	789	1	3	Dean, Master. Bertram Vere	male	1.00	1	2	C.
803	804	1	3	Thomas, Master. Assad Alexander	male	0.42	0	1	
827	828	1	2	Mallet, Master. Andre	male	1.00	0	2	S.C
831	832	1	2	Richards, Master. George Sibley	male	0.83	1	1	
851	852	0	3	Svensson, Mr. Johan	male	74.00	0	0	

```
# How many people did not survive?
df[df['Survived'] == 1].sum()
```

```
<ipython-input-154-692532fc8ffb>:2: FutureWarning: The default value of numeric_only in Da
  df[df['Survived'] == 1].sum()
PassengerId                                            151974
Survived                                                  342
Pclass                                                    667
Name           Cumings, Mrs. John Bradley (Florence Briggs Th...
Sex            femalefemalefemalefemalefemalefemalefemalefema...
Age                                                   8219.67
SibSp                                                     162
Parch                                                     159
Ticket         PC 17599STON/O2. 3101282113803347742237736PP 9...
Fare                                                16551.2294
dtype: object
```

```
# Sorting by Age
df.sort_values('Age').head(10)
```

```
# Sorting by Age in reverse order
df.sort_values(['Age', 'Name'], ascending=[False, True]).head(10)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
630	631	1	1	Barkworth, Mr. Algernon Henry Wilson	male	80.0	0	0	27042
851	852	0	3	Svensson, Mr. Johan	male	74.0	0	0	347060
493	494	0	1	Artagaveytia, Mr. Ramon	male	71.0	0	0	PC 17609
96	97	0	1	Goldschmidt, Mr. George B	male	71.0	0	0	PC 17754
116	117	0	3	Connors, Mr. Patrick	male	70.5	0	0	370369
745	746	0	1	Crosby, Capt. Edward Gifford	male	70.0	1	1	WE/P 5735
672	673	0	2	Mitchell, Mr. Henry Michael	male	70.0	0	0	C.A. 24580
33	34	0	2	Wheadon, Mr. Edward H	male	66.0	0	0	C.A. 24579
280	281	0	3	Duane, Mr. Frank	male	65.0	0	0	336439
456	457	0	1	Millet, Mr. Francis Davis	male	65.0	0	0	13509

Merging Dataframes

```
# Creating temporal Dataframe
df2 = df.copy(deep=True)
```

```
# Concatenation (rows)
concat_df = pd.concat([df, df2])
```

```
concat_df.shape
```

```
(1782, 12)
```

```
df.shape
```

```
(891, 12)
```

```
# Concatenation (columns)
concat_df2 = pd.concat([df, df2], axis=1)
```

```
concat_df2.shape
```

```
(891, 24)
```

```
mdf = pd.DataFrame(index=df.index)
mdf['PassengerId'] = df['PassengerId']
mdf['evenId'] = mdf['PassengerId'].apply(lambda x: x % 2 == 0)
```

```
mdf
```

	PassengerId	evenId
0	1	False
1	2	True
2	3	False
3	4	True
4	5	False
...	...	...
886	887	False
887	888	True
888	889	False
889	890	True
890	891	False

891 rows × 2 columns

```
pd.merge(df, mdf, how='inner')
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2 3101282
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450
...	...	...	...	...	...	...	...	...	..
886	887	0	2	Montvila, Rev. Juozas	male	27.0	0	0	211536

Analytics

```
# Count number of non empty elements (NaN) for each column
df.count()
```

```
PassengerId    891
Survived       891
Pclass         891
Name           891
Sex            891
Age            714
SibSp          891
Parch          891
Ticket         891
Fare           891
Cabin          204
Embarked       889
dtype: int64
```

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

```
891
```

```
# Mean Age
df['Age'].mean()
```

```
29.69911764705882
```

```
# Mean Age by Sex
df.groupby('Sex')['Age'].mean()
```

```
Sex
female    27.915709
male      30.726645
Name: Age, dtype: float64
```

```
df.groupby('Sex')['Age'].describe()
```

Sex	count	mean	std	min	25%	50%	75%	max
female	261.0	27.915709	14.110146	0.75	18.0	27.0	37.0	63.0
male	453.0	30.726645	14.678201	0.42	21.0	29.0	39.0	80.0

```
df.groupby(['Sex','Survived'])['Age'].agg(['mean', 'median'])
```

Sex	Survived	mean	median
female	**0**	25.046875	24.5
	1	28.847716	28.0
male	**0**	31.618056	29.0
	1	27.276022	28.0

```
# count number of men and women
df['Sex'].value_counts()
```

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

```
# Correlation
df.corr()
```

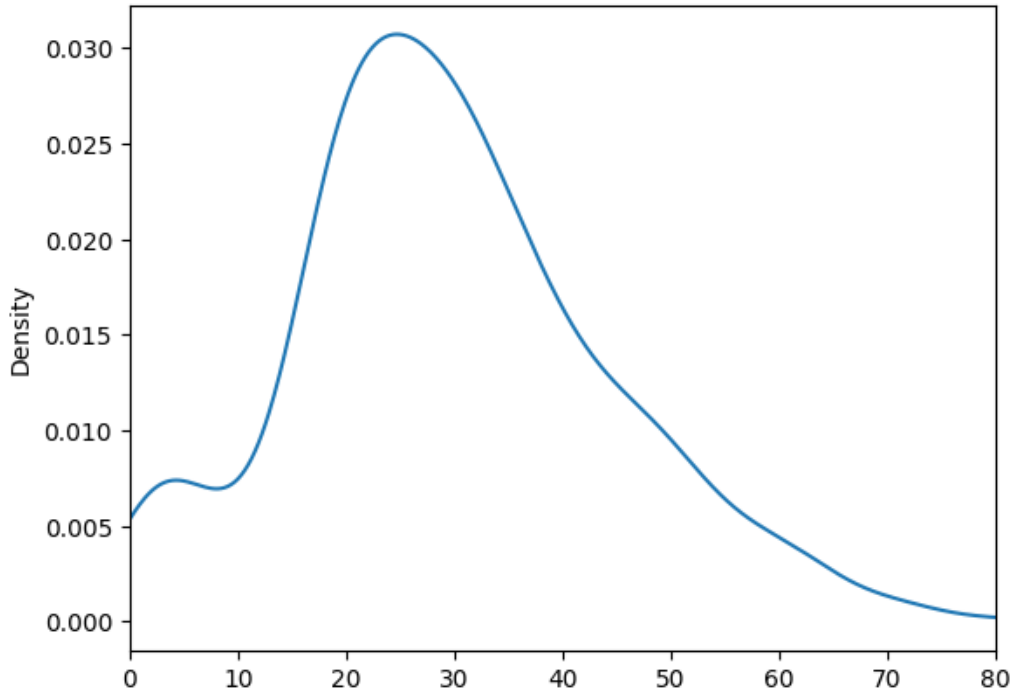
```
<ipython-input-173-6fc543ac6a6f>:2: FutureWarning: The default value of numer
  df.corr()
```

	PassengerId	Survived	Pclass	Age	SibSp	Parch	Fa
PassengerId	1.000000	-0.005007	-0.035144	0.036847	-0.057527	-0.001652	0.0126
Survived	-0.005007	1.000000	-0.338481	-0.077221	-0.035322	0.081629	0.2573
Pclass	-0.035144	-0.338481	1.000000	-0.369226	0.083081	0.018443	-0.5495
Age	0.036847	-0.077221	-0.369226	1.000000	-0.308247	-0.189119	0.0960
SibSp	-0.057527	-0.035322	0.083081	-0.308247	1.000000	0.414838	0.1596
Parch	-0.001652	0.081629	0.018443	-0.189119	0.414838	1.000000	0.2162
Fare	0.012658	0.257307	-0.549500	0.096067	0.159651	0.216225	1.0000

Data Visualization

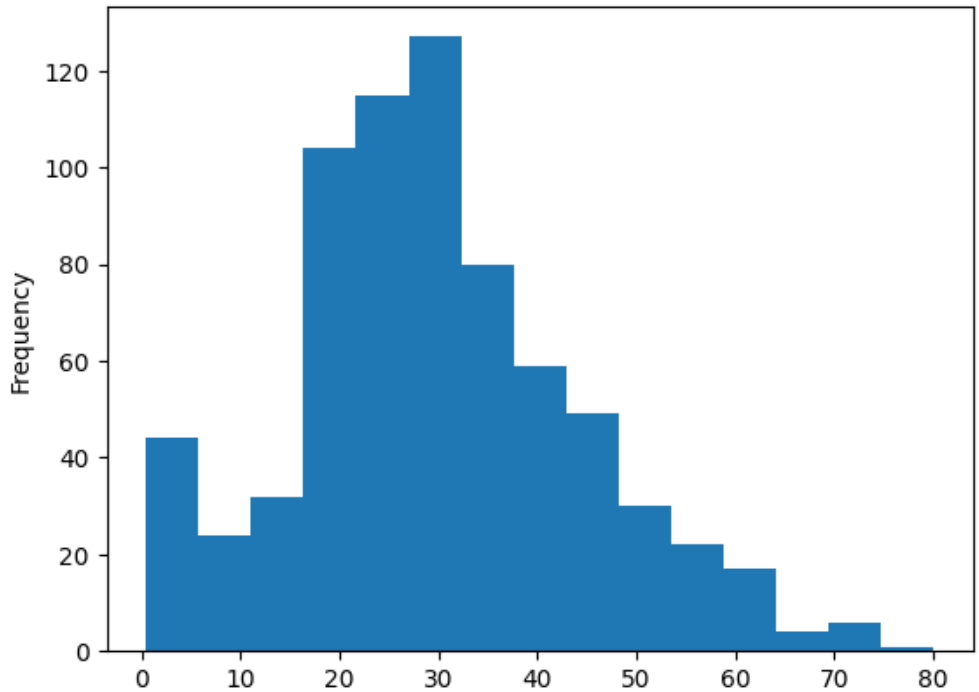
```
# Age distribution
df['Age'].plot(kind='kde', xlim=[0, 80])
```

```
<Axes: ylabel='Density'>
```



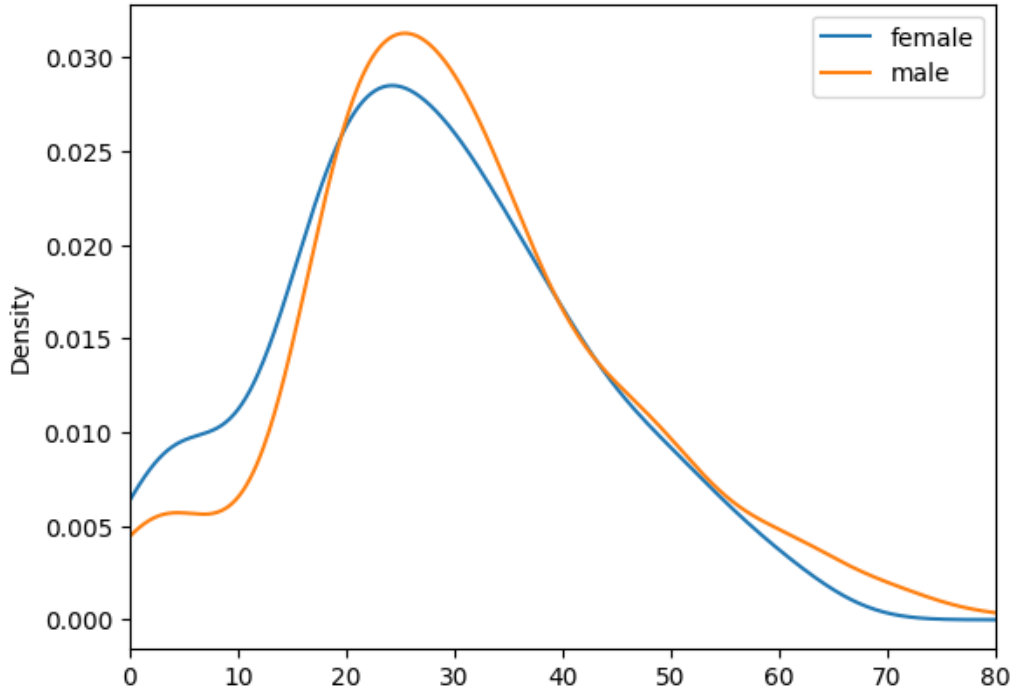

```
# Passengers by Age group
df['Age'].plot(kind='hist', bins=15)
```

```
<Axes: ylabel='Frequency'>
```




```
# Distribution of men and women by Age
df.groupby('Sex')['Age'].plot(kind='kde', xlim=[0, 80], legend=True)
```

```
Sex
female    Axes(0.125,0.11;0.775x0.77)
male      Axes(0.125,0.11;0.775x0.77)
Name: Age, dtype: object
```



Changing Data

```
temp_df = df.copy(deep=True)
```

```
# Let's save the Titanic passengers
temp_df['Survived'] = 1
```

```
temp_df['Survived'].value_counts()
```

```
1    891
Name: Survived, dtype: int64
```

```
temp_df['isChild'] = temp_df['Age'] <= 18
```

```
temp_df.head(10)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123
4	5	1	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN
5	6	1	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN
6	7	1	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46
7	8	1	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN

```
# Finding survived children
temp_df['isChildSurvived'] = temp_df['isChild']
temp_df.loc[temp_df['Survived']==0, 'isChildSurvived'] = False
```

(Elisabeth

```
temp_df.head(10)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123
4	5	1	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN
5	6	1	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN
6	7	1	1	McCarthy, Mr.	male	54.0	0	0	17463	51.8625	E46

```
# Rename columns
temp_df.rename(columns={'isChildSurvived': 'ChildSurvived'}).head(5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin
0	1	1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123
4	5	1	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN

```
# Column title to uppercase
```

```
# Column title to uppercase
temp_df.rename(columns=str.upper).head(5)
```

	PASSENGERID	SURVIVED	PCLASS	NAME	SEX	AGE	SIBSP	PARCH	TICKET	FARE	CABIN
0	1	1	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123
4	5	1	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN

```
# Names of passengers to uppercase
temp_df['upper_case_name'] = temp_df['Name'].str.upper()
```

```
temp_df.head(5)
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket
				Braund,					

```
temp_df['upper_case_name']
```

```
0                                BRAUND, MR. OWEN HARRIS
1      CUMINGS, MRS. JOHN BRADLEY (FLORENCE BRIGGS TH...
2                               HEIKKINEN, MISS. LAINA
3           FUTRELLE, MRS. JACQUES HEATH (LILY MAY PEEL)
4                             ALLEN, MR. WILLIAM HENRY
                                ...
886                             MONTVILA, REV. JUOZAS
887                       GRAHAM, MISS. MARGARET EDITH
888            JOHNSTON, MISS. CATHERINE HELEN "CARRIE"
889                             BEHR, MR. KARL HOWELL
890                               DOOLEY, MR. PATRICK
Name: upper_case_name, Length: 891, dtype: object
```

3	4	1	1	Jacques Heath (Lily May Peel)	female	35.0	1	0	113803
4	5	1	3	Allen, Mr. William Henry	male	35.0	0	0	373450

✓ 0s completed at 05:42