
telesoft ai

# PANDAS BASICS

John Fedinandi

## What is Pandas?

- **Pandas** is a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real world data analysis in Python.
- The two primary data structures of pandas, Series (1-dimensional) and DataFrame (2-dimensional), handle the vast majority of typical use cases in finance, statistics, social science, and many areas of engineering.
- Data frames are tabular, meaning that they are based on rows and columns like you would see in a spreadsheet.
- Pandas is built on top of NumPy and is intended to integrate well within a scientific computing environment with many other 3rd party libraries.

## Here are just a few of the things that pandas does well:

- Easy handling of missing data (represented as NaN) in floating point as well as non-floating point data.
- Size mutability: columns can be inserted and deleted from DataFrame and higher dimensional objects.
- Automatic and explicit data alignment: objects can be explicitly aligned to a set of labels, or the user cansimply ignore the labels and let Series, DataFrame, etc. automatically align the data for you in computations.
- Powerful, flexible group by functionality to perform split-apply-combine operations on data sets, for both aggregating and transforming data.
- Make it easy to convert ragged, differently-indexed data in other Python and NumPy data structures into DataFrame objects.
- Intelligent label-based slicing, fancy indexing, and subsetting of large data sets.
- Intuitive merging and joining data sets.
- Flexible reshaping and pivoting of data sets.

# Pandas Installation

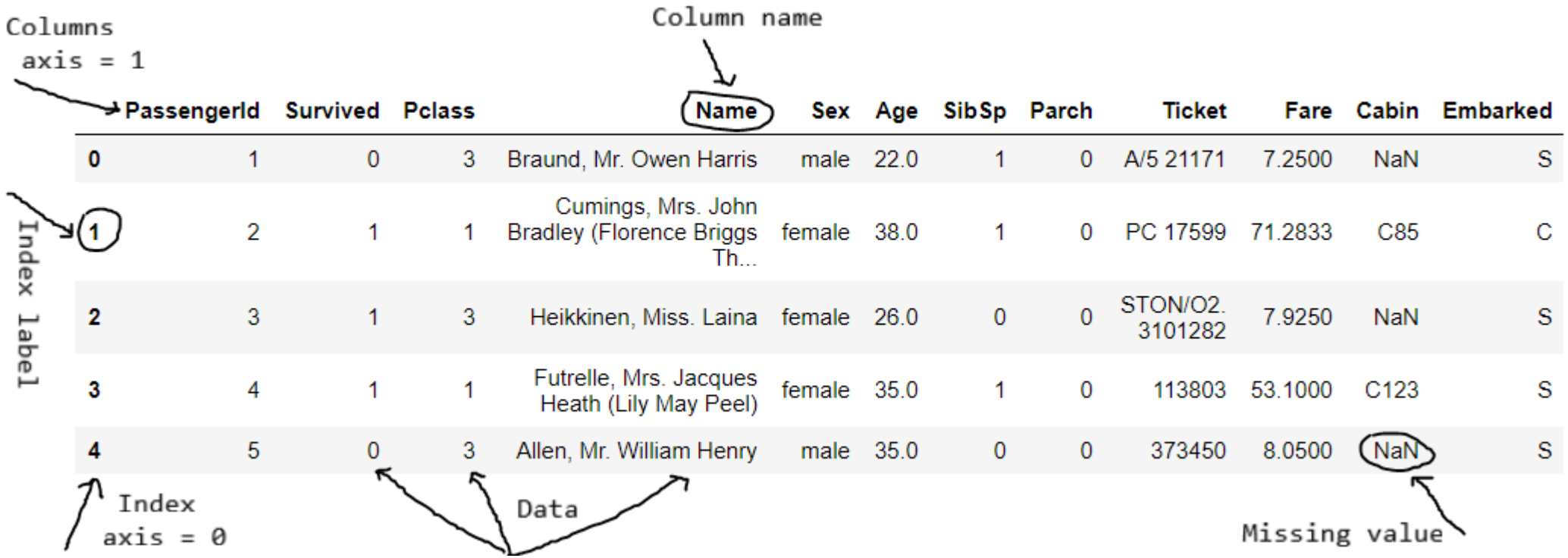
- conda environment conda install pandas
- Installing from PyPI

  python -m pip install pandas
- **Installing pandas on Linux**
  - In the following table, we will present some of the common Linux distributions package names for Matplotlib and the tools we can use to install the package:

| | |
|---|---|
| Debian or Ubuntu (And other Debian derivatives) | sudo apt-get install python3-pandas |
| Fedora | sudo dnf install python3-pandas |
| Red Hat | sudo yum install python3-matplotlib |
| Centos/RHEL | sudo dnf install python3-pandas |

# 1.Understanding a pandas DataFrame

- A pandas DataFrame (in a Jupyter Notebook) appears to be nothing more than an ordinary table of data consisting of rows and columns. Hiding beneath the surface are the three components, the index, columns, and data (also known as values) that you must be aware of in order to maximize the DataFrame's full potential.
- Analyse the labeled anatomy of the DataFrame:
- **Note**
  - In this Notebook we will be using a **Titanic** dataset. A dataset about passengers in Titanic.




| | 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 |

# The variables that describe the passengers are:

**PassengerId**: and id given to each traveller on the boat.

**Pclass**: the passenger class. It has three possible values: 1,2,3.

**The Name**: a word or set of words by which a person or thing is usually known.

**The Sex**: males or females considered as separate groups.

**The Age**: the number of years that someone has lived.

**SibSp**: number of siblings and spouses traveling with the passenger.

**Parch**: number of parents and children traveling with the passenger.

**The ticket number**: a number (identifier) piece of paper that shows you have paid for a journey.

**The ticket Fare**: amount paid for a ticket.

**The cabin number**: a number for private room on a ship for a passenger.

**The embarkation**: It has three possible values S,C,Q.

telesoft ai

```
In [1]: # Load library

In [2]: import pandas as pd

In [3]: #Create url

In [4]: url = 'Data/Titanic.csv'

In [5]: # Load data as a DataFrame

In [6]: dataframe = pd.read_csv(url)

In [7]: # Show first 5 rows

In [8] dataframe.head(5)
```

| | 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 |

## Things to notice in this DataFrame

- First, in a data frame each row corresponds to one observation (e.g., a passenger) and each column corresponds to one feature (gender, age, etc.). For example, bylooking at the first observation we can see that **Heikkinen, Miss. Laina** stayed in first class, was 26 years old, was female, and survived the disaster.
- Second, each column contains a name (e.g., Name, PClass, Age) and each row contains an index number (e.g., 0 for the lucky Miss Elisabeth Walton Allen). We will use these to select and manipulate observations and features.

# 2.Creating a DataFrame

- First method :
  - Create a dataframe and add columns independently.

```
In [1]: #Load library

In [2]: import pandas as pd

In [3]: # Create a DataFrame

In [4]: df = pd.DataFrame()

In [5]: #Add columns to a DataFrame

In [6]: df['Name'] = ['John', 'Rebecca', 'Lisa', 'Godfrey', 'Vivan']

In [7]: df['Age'] = [19,16,27,18,91]

In [8]: df['Country'] = ['Kenya', 'Uganda', 'Rwanda‘ ,'Tanzania', 'Burundi']

In [9]: #show DataFrame

In [10] df
```

| | Name | Age | Country |
|---|---|---|---|
| 0 | John | 19 | Kenya |
| 1 | Rebecca | 16 | Uganda |
| 2 | Lisa | 27 | Rwanda |
| 3 | Godfrey | 18 | Tanzania |
| 4 | Vivan | 91 | Burundi |

telesoft ai

- Second method :

  Create a dataframe and add columns at the same time.

```
In [1]: #Load library
In [2]: import pandas as pd
In [3]: # Create a DataFrame
In [4]: df = pd.DataFrame(columns=['Name', 'Age', 'Country'],
                  data=[
                      ['John',19,'Kenya'],
                      ['Rebecca',16,'Uganda'],
                      ['Lisa',27,'Rwanda'],
                      ['Godfrey',18,'Tanzania'],
                      ['Vivan',91,'Burundi']
                     ])
In [5]: df
```

| | Name | Age | Country |
|---|---|---|---|
| 0 | John | 19 | Kenya |
| 1 | Rebecca | 16 | Uganda |
| 2 | Lisa | 27 | Rwanda |
| 3 | Godfrey | 18 | Tanzania |
| 4 | Vivan | 91 | Burundi |

telesoft ai

# 3.Creating a Series

```
In [1]: #Load library

In [2]: import pandas as pd

In [3]: #Create a Series

In [4]: series = pd.Series(index=['Name', 'Age', 'Country'], data=['John', 19, 'Uganda'])

In [5]: #show series

In [6]:Series

Out [1]: Name          John
         Age           19
         Country       Uganda
         dtype : object
```

# A series can be used to create a DataFrame as follows

```
In [1]: #Load library

In [2]: import pandas as pd

In [3]: #Create a DataFrame

In [4]: df = pd.DataFrame().append(series, ignore_index=True)

In [5]: #show DataFrame

In [6]: df
```

| | Age | Country | Name |
|---|---|---|---|
| **0** | 19.0 | Uganda | John |

telesoft ai

# 4.Describing a DataFrame

- Describing a DataFrame involve looking at its short summary of descriptive statistical measures.

```
In [1]: #Load library

In [2]: import pandas as pd

In [3]: #Create url

In [4]: url = 'Data/Titanic.csv'

In [5]: #Load data as a DataFrame

In [6]: dataframe = pd.read_csv(url)

In [7]: #show statistics

In [8]: dataframe.describe()
```

- We can also take a look at the number of row and columns.

```
In [1]: dataframe.shape

Out [1]: (891,12)
```

- DataFrame has 891 rows(instances/samples) and 12 columns(features).

## 5.Navigating DataFrames.

- You need to select individual data or slices of a DataFrame.
  - **loc**
    - is useful when the index of the DataFrame is a label (e.g., a string).
  - **iloc**
    - works by looking for the position in the DataFrame. For example, iloc[0] will return the first row regardless of whether the index is an integer or a label.

```
In [1]: # Select three rows

In [2]: dataframe.iloc[1:4] # also dataframe.iloc[:4]
```

telesoft ai

| | 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 |
| **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 |

- DataFrames do not need to be numerically indexed. We can set the index of a DataFrame to any value where the value is unique to each row. For example, we can set the index to be passenger names and then select rows using a name:

```
In [1]: #set index

In [2]: dataframe = dataframe.set_index(dataframe['Name'])

In [3]: #use index to slice and show row

In [4]: dataframe.loc['Heikkinen, Miss. Laina']
```

Out [1]:

| | |
|---|---|
| PassangerId | 3 |
| Survived | 1 |
| Pclass | 3 |
| Name | Heikkinen, Miss. Laina |
| Sex | Female |
| Age | 26 |
| SibSp | 0 |
| Patch | 0 |
| Ticket | STON/02. 3101282 |
| Fare | 7.925 |
| Cabin | NaN |
| Embarked | S |

Name: Heikkinen, Miss. Laina dtype:object

# 6.Selecting Rows Based on Conditionals

```
In [1]: # Load library

In [2]: import pandas as pd

In [3]: # Create URL

In [4]: url = 'Data/Titanic.csv'

In [5]: # Load data

In [6]: dataframe = pd.read_csv(url)

In [7]: # Show top two rows where column 'sex' is 'female'

In [8]: dataframe[dataframe['Sex'] == 'female'].head(2)
```

| | 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 |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | NaN | S |

- Multiple conditions are easy as well. For example, here we select all the rows where the passenger is a female 65 or older:

```
In [1]: # Show top two rows where column 'sex' is 'female' and 'age' >=27

In [2]: dataframe[(dataframe['Sex'] == 'female') & (dataframe['Age'] >= 27) ].head(2)
```

| | 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 |
| **3** | 4 | 1 | 1 | Futrelle, Mrs. Jacques Heath (Lily May Peel) | female | 35.0 | 1 | 0 | 113803 | 53.1000 | C123 | S |

telesoft ai

# 7.Replacing Values

- pandas' replace() is an easy way to find and replace values. For example, we can replace any instance of "female" in the Sex column with "Woman":

```
In [1]: # Load library
In [2]: import pandas as pd
In [3]: # Create URL
In [4]: url = 'Data/Titanic.csv'
In [5]: # Load data
In [6]: dataframe = pd.read_csv(url)
In [7]: # Replace values, show two rows
In [8]: dataframe['Sex'].replace("female", "woman").head(2)
Out [1]: 0        male
         1       woman
         Name : Sex, dtype:object
```

telesoft ai

- We can also replace multiple values at the same time:

```
In [1]: # Replace "female" and "male with "Woman" and "Man"

In [2]: dataframe['Sex'].replace(["female", "male"], ["Woman", "Man"]).head(5)

Out [1]: 0       Man
         1       Woman
         2       Woman
         3       Woman
         4       Man
         Name: Sex, dtype:object
```

# 8.Renaming Columns

- Rename columns using the rename() method:

```
In [1]: # Load library

In [2]: import pandas as pd

In [3]: # Create URL

In [4]: url = 'Data/Titanic.csv'

In [5]: # Load data

In [6]: dataframe = pd.read_csv(url)

In [7]: # Rename column, show two rows

In [8]: dataframe.rename(columns={'PClass': 'Passenger Class'}).head(2)
```

| | PassengerId | Survived | Passenger Class | 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 |

- Notice that the rename() method can accept a dictionary as a parameter. We can use the dictionary to change multiple column names at once:

```
In [1]: # Rename columns, show two rows

In [2]: dataframe.rename(columns={'PClass': 'Passenger Class', 'Sex': 'Gender'}).head(2)
```

| | PassengerId | Survived | Passenger Class | Name | Gender | 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 |

# 9.Finding the Minimum, Maximum, Sum, Average, and Count

```
In [1]: # Load library
In [2]: import pandas as pd
In [3]: # Create URL
In [4]: url = 'Data/Titanic.csv'
In [5]: # Load data
In [6]: dataframe = pd.read_csv(url)
In [7]: # Calculate statistics
In [8]: print('Maximum:', dataframe['Age'].max())
In [9]: print('Minimum:', dataframe['Age'].min())
In [10]: print('Mean:', dataframe['Age'].mean())
In [11]: print('Sum:', dataframe['Age'].sum())
In [12]: print('Count:', dataframe['Age'].count())
```

telesoft ai

Out [1]: Maximum: 80.0

Minimum: 0.42

Mean: 29.966117647058 82

Sum: 21205.17

Count: 714

# 10.Finding Unique Values

- Use unique to view an array of all unique values in a column:

```
In [1]: # Load library
In [2]: import pandas as pd
In [3]: # Create URL
In [4]: url = 'Data/Titanic.csv'
In [5]: # Load data
In [6]: dataframe = pd.read_csv(url)
In [7]: # Select unique values
In[8]: dataframe['Sex'].unique()
Out [1]: array([‘male’, ’female’],dtype:object)
```

telesoft ai

- Alternatively, value_counts() will display all unique values with the number of times each value appears:

```
In [1] dataframe['Sex'].value_counts()

Out [1]:  male          577

          female      314

          Name: Sex, dtype: object
```

# 11.Handling Missing Values

- isnull() and notnull() return booleans indicating whether a value is missing:

```
In [1]: # Load library
In [2]: import pandas as pd
In [3]: # Create URL
In [4]: url = 'Data/Titanic.csv'
In [5] # Load data
In [6]: dataframe = pd.read_csv(url)
In [7]: # Select missing values, show two rows
In [6]: dataframe[dataframe['Age'].isnull()].head(2)
```

| | PassengerId | Survived | Pclass | Name | Sex | Age | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **5** | 6 | 0 | 3 | Moran, Mr. James | male | NaN | 0 | 0 | 330877 | 8.4583 | NaN | Q |
| **17** | 18 | 1 | 2 | Williams, Mr. Charles Eugene | male | NaN | 0 | 0 | 244373 | 13.0000 | NaN | S |

# 12.Deleting a Column

- The best way to delete a column is to use drop() with the parameter axis=1 (i.e., the column axis):

```
In [1]: # Load library

In [2]: import pandas as pd

In [3]: # Create URL

In [4]: url = 'Data/Titanic.csv'

In [5]: # Load data

In [6]: dataframe = pd.read_csv(url)

In [7]: # Delete column

In [8]: dataframe.drop('Age', axis=1).head(2)
```

| | PassengerId | Survived | Pclass | Name | Sex | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | male | 1 | 0 | A/5 21171 | 7.2500 | NaN | S |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | female | 1 | 0 | PC 17599 | 71.2833 | C85 | C |

- You can also use a list of column names as the main argument to drop multiple columns at once:

```
In [1]: # Drop columns

In [2]: dataframe.drop(['Age', 'Sex'], axis=1).head(2)
```

| | PassengerId | Survived | Pclass | Name | SibSp | Parch | Ticket | Fare | Cabin | Embarked |
|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 0 | 3 | Braund, Mr. Owen Harris | 1 | 0 | A/5 21171 | 7.2500 | NaN | S |
| **1** | 2 | 1 | 1 | Cumings, Mrs. John Bradley (Florence Briggs Th... | 1 | 0 | PC 17599 | 71.2833 | C85 | C |

# 13.Deleting a Row

- Use a boolean condition to create a new DataFrame excluding the rows you want to delete:

```
In [1]: # Load library

In [2]: import pandas as pd

In [3]: # Create URL

In [4]: url = 'Data/Titanic.csv'

In [5]: # Load data

In [6]: dataframe = pd.read_csv(url)

In [7]: # Delete rows, show first two rows of output

In [8]: dataframe[dataframe['Sex'] != 'male'].head(2)
```

telesoft ai

| | 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 |
| **2** | 3 | 1 | 3 | Heikkinen, Miss. Laina | female | 26.0 | 0 | 0 | STON/O2. 3101282 | 7.9250 | NaN | S |

# 14.Dropping Duplicate Rows

- Use drop_duplicates(), but be mindful of the parameters:

```
In [1]: # Load library

In [2]: import pandas as pd

In [3]: # Create URL

In [4]: url = 'Data/Titanic.csv'

In [5]: # Load data

In [6]: dataframe = pd.read_csv(url)

In [8]: # Drop duplicates, show first two rows of output

In [9]: dataframe.drop_duplicates(keep='last').head(2)
```

| | 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 |

# 15.Grouping Rows by Values

- groupby() is one of the most powerful features in pandas:

```
In [1]: # Load library
In [2]: import pandas as pd
In [3]: # Create URL
In [4]: url = 'Data/Titanic.csv'
In [5]: # Load data
In [6]: dataframe = pd.read_csv(url)
In [7]: # Group rows by the values of the column 'Sex', calculate mean
In [8]: # of each group
In [8]: dataframe.groupby('Sex').mean()
```

| Sex | PassengerId | Survived | Pclass | Age | SibSp | Parch | Fare |
|---|---|---|---|---|---|---|---|
| **female** | 431.028662 | 0.742038 | 2.159236 | 27.915709 | 0.694268 | 0.649682 | 44.479818 |
| **male** | 454.147314 | 0.188908 | 2.389948 | 30.726645 | 0.429809 | 0.235702 | 25.523893 |

# 15.Concatenating DataFrames

- Use concat with axis=0 to concatenate along the row axis:

```
In [1]: # Load library

In [2]: import pandas as pd

In [3]: # Create DataFrame

In [4]: data_a = {'id': ['1', '2', '3'],

'first': ['Alex', 'Amy', 'Allen'],

'last': ['Anderson', 'Ackerman', 'Ali']}

In [5]: dataframe_a = pd.DataFrame(data_a, columns = ['id', 'first', 'last'])
```

```
In [6]: # Create DataFrame

In [7]: data_b = {'id': ['4', '5', '6'],

'first': ['Billy', 'Brian', 'Bran'],

'last': ['Bonder', 'Black', 'Balwner']}

In [8]:dataframe_b = pd.DataFrame(data_b, columns = ['id', 'first', 'last'])

In [9]: # Concatenate DataFrames by rows

In[10]: pd.concat([dataframe_a, dataframe_b], axis=0)
```

| | id | first | last |
|---|---|---|---|
| **0** | 1 | Alex | Anderson |
| **1** | 2 | Amy | Ackerman |
| **2** | 3 | Allen | Ali |
| **0** | 4 | Billy | Bonder |
| **1** | 5 | Brian | Black |
| **2** | 6 | Bran | Balwner |

# References


- pandas: powerful Python data analysis toolkit, Release 0.18.1, Wes McKinney & PyData Development Team,2016.
- Python programming -Pandas, Finn Arup


Write to : telesoftai@gmail.com

telesoft ai