BEGINNER (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/BEGINNER/)

DATA CLEANING (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/DATA-
EXPLORATION/DATA-CLEANING/)

DATA EXPLORATION (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/DATA-
EXPLORATION/)

DATA SCIENCE (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/DATA-SCIENCE/)

LIBRARIES (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/LIBRARIES/)

PANDAS (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/PYTHON-2/PANDAS/)

PYTHON (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/PYTHON/)

STRUCTURED DATA (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/CATEGORY/STRUCTURED-
DATA/)

Data Manipulation Using Pandas | Essential Functionalities of Pandas you need to know!

SRIVIGNESH_R (HTTPS://WWW.ANALYTICSVIDHYA.COM/BLOG/AUTHOR/SRIVIGNESH...

Article   🔒 Video Book

This article was published as a part of the Data Science Blogathon (https://datahack.analyticsvidhya.com/contest/data-science-blogathon-9/)

Pandas

Pandas is an open-source data analysis and data manipulation library written in python. Pandas provide you with data structures and functions to work on structured data seamlessly. The name Pandas refer to "Panel Data", which means a structured dataset. Pandas have two main classes to work on, DataFrame and Series. Let us explore more on this later in this article.

Key Features of Pandas

- Perform Group by operation seamlessly
- Datasets are mutable using pandas which means we can add new rows and columns to them.
- Easy to handle missing data
- Merge and join datasets
- Indexing and subsetting data

Installation

Install via pip using the following command,

```
pip install pandas
```

Install via anaconda using the following command,

```
conda install pandas
```

DataFrame in Pandas

POPULAR POSTS

Donut Plots : Data Visualization With Python (https://www.analyticsvidhya.com/blog/2021/06/donut-plots-data-visualization-with-python/)

Build Treemaps in Python using Squarify (https://www.analyticsvidhya.com/blog/2021/06/build-treemaps-in-python-using-squarify/)

Logistic Regression-Supervised Learning Algorithm for Classification (https://www.analyticsvidhya.com/blog/2021/05/logistic-regression-supervised-learning-algorithm-for-classification/)

How to Build Word Cloud in Python? (https://www.analyticsvidhya.com/blog/2021/05/how-to-build-word-cloud-in-python/)

How to Use Progress Bars in Python? (https://www.analyticsvidhya.com/blog/2021/05/how-to-use-progress-bars-in-python/)

Introduction to Python Programming (Beginner's Guide) (https://www.analyticsvidhya.com/blog/2021/05/introduction-to-python-programming-beginners-guide/)

Making Programming with Date and Time, less painless (https://www.analyticsvidhya.com/blog/2021/05/making-programming-with-date-and-time-less-painless/)

Python List Programs For Absolute Beginners –

A DataFrame is a two-dimensional table in pandas. Each column can have different data types like int, float, or string. Each column is of class Series in pandas, we'll discuss this later in this article.

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

DataFrame
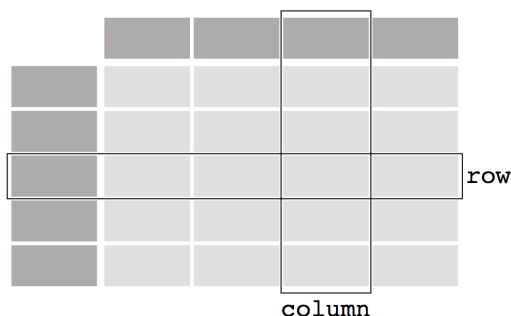
Image Source (https://pandas.pydata.org/docs/getting_started/index.html#getting-started)

Creating a DataFrame in Pandas

```
# import the library as pd
import pandas as pd
df = pd.DataFrame(
    {
        'Name': ['Srivignesh', 'Hari'],
        'Age': [22, 11],
        'Country': ['India', 'India']
    }
)
print(df)
# output
#          Name  Age Country
# 0  Srivignesh   22   India
# 1        Hari   11   India
```

pd.DataFrame is a class available in pandas. Here we provide a dictionary whose keys are the column names ('Name', 'Age', 'Country') and the values are the values in those columns. Here each column is of class pandas.Series. Series is a one-dimensional data used in pandas.

```
# accessing the column 'Name' in df
print(df['Name'])
# Output
# 0      Srivignesh
# 1            Hari
# Name: Name, dtype: object
print(type(df['Name']))
# Output
# <class 'pandas.core.series.Series'>
```

Let's get started with Data Manipulation using Pandas!

For this purpose, we are going to use Titanic Dataset
(https://www.kaggle.com/c/titanic/data) which is available on Kaggle.

```
import pandas as pd
path_to_data = 'path/to/titanic_dataset'
# read the csv data using pd.read_csv function
data = pd.read_csv(path_to_data)
data.head()
```

	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

Dropping columns in the data

```
df_dropped = data.drop('Survived', axis=1)
df_dropped.head()
```

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

The 'Survived' column is dropped in the data. The axis=1 denotes that
it 'Survived' is a column, so it searches 'Survived' column-wise to drop.

Drop multiple columns using the following code,

```
df_dropped_multiple = data.drop(['Survived', 'Name'], axis=1)
df_dropped_multiple.head()
```

	PassengerId	Pclass	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	3	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	3	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	female	35.0	1	0	113803	53.1000	C123	S
4	5	3	male	35.0	0	0	373450	8.0500	NaN	S

become-a-data-

7-signs-snow-
you-have-data-
scientist-
potential/?
&utm_source=Blog&utm_medium=CareerResou
DECEMBER 3, 2020

How To Have a
Career in Data
Science
(Business
Analytics)?
(https://www.analyticsvidhya.com/blog/2020/1
to-have-a-career-
in-data-science-
business-
analytics/?
&utm_source=Blog&utm_medium=CareerResou
NOVEMBER 26,
2020

RECENT POSTS

Exploring Mito: Automatic Python Code for SpreadSheet Operations (https://www.analyticsvidh ya.com/blog/2021/06/explo ring-mito-automatic-python-code-for-spreadsheet-operations/)
JUNE 20, 2021

Part- 4: Step by Step Guide to Master Natural Language Processing in Python (https://www.analyticsvidh ya.com/blog/2021/06/part-4-step-by-step-guide-to-master-natural-language-processing-in-python/)
JUNE 20, 2021

The columns 'Survived' and 'Name' are dropped in the data.

Dropping rows in the data

```
df_row_dropped = data.drop(2, axis=0)
df_row_dropped.head()
```

	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
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
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q

The row with index 2 is dropped in the data. The axis=0 denotes that index 2 is a row, so it searches the index 2 column-wise.

Drop multiple rows using the following code,

```
df_row_dropped_multiple = data.drop([2, 3], axis=0)
df_row_dropped_multiple.head()
```

	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
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
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S

The rows with indexes 2 and 3 are dropped in the data.

Renaming a column in the dataset

```
data.columns
# Output
# Index(['PassengerId', 'Survived', 'Pclass', 'Name', 'Sex',
'Age', 'SibSp',
#        'Parch', 'Ticket', 'Fare', 'Cabin', 'Embarked'],
#        dtype='object')
df_renamed = data.rename(columns={'PassengerId': 'Id'})
df_renamed.head()
```

	Id	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 column 'PassengerId' is renamed to 'Id' in the data. Do not forget to mention the dictionary inside the columns parameter.

Rename multiple columns using the following code,
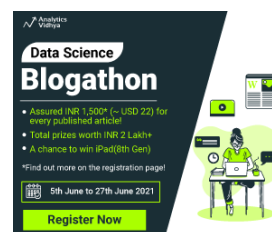
```
df_renamed_multiple = data.rename(

    columns={

        'PassengerId': 'Id',

        'Sex': 'Gender',

    }

)

df_renamed_multiple.head()
```

	Id	Survived	Pclass	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
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 columns 'PassengerId' and 'Sex' are renamed to 'Id' and 'Gender' respectively.

Select columns with specific data types

```
integer_data = data.select_dtypes('int')

integer_data.head()
```

	PassengerId	Survived	Pclass	SibSp	Parch
0	1	0	3	1	0
1	2	1	1	1	0
2	3	1	3	0	0
3	4	1	1	1	0
4	5	0	3	0	0

The above code selects all columns with integer data types.

```
float_data = data.select_dtypes('float')

float_data.head()
```

	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

The above code selects all columns with float data types.

Slicing the dataset

```
data.iloc[:5, 0]
```

```
0    1
1    2
2    3
3    4
4    5
Name: PassengerId, dtype: int64
```

The above code returns the first five rows of the first column. The ':5' in the iloc denotes the first five rows and the number 0 after the comma denotes the first column, iloc is used to locate the data using numbers or integers.

```
data.loc[:5, 'PassengerId']
```

```
0    1
1    2
2    3
3    4
4    5
5    6
Name: PassengerId, dtype: int64
```

The above code does the same but we can use the column names directly using loc in pandas. Here the index 5 is inclusive.

Handle Duplicates in Dataset

Since there are no duplicate data in the titanic dataset, let us first add a duplicated row into the data and handle it.

```
df_dup = data.copy()
# duplicate the first row and append it to the data
row = df_dup.iloc[:1]
df_dup = df_dup.append(row, ignore_index=True)
df_dup
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
887	888	1	1	Graham, Miss. Margaret Edith	female	19.0	0	0	112053	30.0000	B42	S
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	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
891	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S

```
df_dup[df_dup.duplicated()]
```

	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
891	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.25	NaN	S

The above code returns the duplicated rows in the data.

```
df_dup.drop_duplicates()
```

	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
...	...	...	...	...	...	...	...	...	...	...	...	...
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
888	889	0	3	Johnston, Miss. Catherine Helen "Carrie"	female	NaN	1	2	W./C. 6607	23.4500	NaN	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

The above code drops the duplicated rows in the data.

Select specific values in the column

```
data[data['Pclass'] == 1]
```

	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
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
23	24	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.5000	A6	S
...	...	...	...	...	...	...	...	...	...	...	...	...
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S
872	873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0000	B51 B53 B55	S
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C
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

The above code returns the values which are equal to one in the column 'Pclass' in the data.

Select multiple values in the column using the following code,

	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
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
23	24	1	1	Sloper, Mr. William Thompson	male	28.0	0	0	113788	35.5000	A6	S
...	...	...	...	...	...	...	...	...	...	...	...	...
871	872	1	1	Beckwith, Mrs. Richard Leonard (Sallie Monypeny)	female	47.0	1	1	11751	52.5542	D35	S
872	873	0	1	Carlsson, Mr. Frans Olof	male	33.0	0	0	695	5.0000	B51 B53 B55	S
879	880	1	1	Potter, Mrs. Thomas Jr (Lily Alexenia Wilson)	female	56.0	0	1	11767	83.1583	C50	C
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

The above code returns the values which are equal to one and zero in the column 'Pclass' in the data.

Group by in DataFrame

```
data.groupby('Sex').agg({'PassengerId': 'count'})
```

PassengerId

Sex	
female	314
male	577

The above code groups the values of the column 'Sex' and aggregates the column 'PassengerId' by the count of that column.

```
data.groupby('Sex').agg({'Age':'mean'})
```

Age

Sex	
female	27.915709
male	30.726645

The above code groups the values of the column 'Sex' and aggregates the column 'Age' by mean of that column.

Group multiple columns using the following code,

```
data.groupby(['Pclass', 'Sex']).agg({'PassengerId': 'count'})
```

PassengerId

Pclass	Sex	
1	female	94
	male	122
2	female	76
	male	108
3	female	144
	male	347

Map in Pandas

```
data['Survived'].map(lambda x: 'Survived' if x==1 else 'Not-Su
rvived')
```

```
0      Not-Survived
1          Survived
2          Survived
3          Survived
4      Not-Survived
          ...
886    Not-Survived
887        Survived
888    Not-Survived
889        Survived
890    Not-Survived
Name: Survived, Length: 891, dtype: object
```

The above code maps the values 0 to 'Not-Survived' and 1 to 'Survived'. You can alternatively use the following code to obtain the same results.

```
data['Survived'].map({1: 'Survived', 0: 'Not-Survived'})
```

Replacing values in a DataFrame

```
data['Sex'].replace(['male', 'female'], ["M", "F"])
```