**Introduction to Pandas**

Lesson 3: Data Manipulation & Processing

An initiative 

https://creativecommons.org/licenses/by-nc-nd/4.0/

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

What is Pandas?

● Officially stands for ‘Python Data Analysis Library’

● Powerful Python library for data manipulation and processing ● Your data’s home

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Pandas install + import

conda install pandas **or** pip install pandas ← from terminal / command line !pip install pandas ← in jupyter notebook / Colab notebook

import pandas as pd ← or can just import pandas in the Colab

3

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Core components of Pandas Series Source: https://www.learndatasci.com/tutorials/python-pandas-tutorial-complete-introduction-for-beginners/

Dataframe

4

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Creating series from scratch

INPUT: pd.Series([1, 3, 5, np.nan, 6, 8])

OUTPUT:

0 1.0

1 3.0

2 5.0

3 NaN

4 6.0

5 8.0

Source: https://pandas.pydata.org/pandas-docs/stable/getting\_started/10min.html 

5

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Creating dataframes from scratch

INPUT: data = { 'Amy': [80, 60, 90, 80], 'Ben': [78, 50, 100, 80]} marks = pd.DataFrame(data)

marks

Amy Ben

0 80 78

1 60 50

OUT: 

2 90 100

3 80 80

6

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Read and export to different formats

**Data format Read Save**

.csv pd.read\_csv() pd.to\_csv()

json pd.read\_json() pd.to\_json()

Excel pd.read\_excel() pd.to\_excel()

SQL pd.read\_sql() pd.to\_sql()

7

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Importing .csv files in Colab

There are different ways of importing .csv files to a colab environment. For example:

– Your . csv file is on a local drive:

– Your .csv file is on GitHub:

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND 

Main questions data scientists face

• How do we view or get info from data?

• What if there are duplicate rows?

• Why and how to change the column names?

• What do I do with missing values?

• What do I do if I just want part of the data frame?

9

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

How to view data

head() shows the be~~ginning of a file~~

INPUT: movies\_df.head()

tail() shows the end of a file

INPUT: movies\_df.tail()

-



y 0 a 0.4 u 0 b 1.5 y 1 c 6.3 x 0 d 5.3 i 0 e 9.2 ... ... ... ...

e 1 h 1.3 t 0 h 4.5 y 0 w 5.3 t 1 d 1.4

i 0 e 2.3 10

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Another viewing option in Pandas y 0 a 0.4

>>> **pandas.set\_~~option~~**~~(‘display.max\_rows’, 10)~~

INPUT: print(movies\_df)

-

https://dev.to/chanduthedev/how-to-display-all-rows-from-data-frame-using-pandas-dha

u 0 b 1.5

y 1 c 6.3

x 0 d 5.3

i 0 e 9.2

... ... ... ...

e 1 h 1.3

t 0 h 4.5

y 0 w 5.3

t 1 d 1.4

i 0 e 2.3

11

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Getting info on the data

movie\_df.info() 

OUTPUT:

12

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

How to know the shape of the data

OUTPUT: 

(#row, #col)

22

3

13 

https://giphy.com/gifs/morphiing-shapes-animation-xT4uQB3ugjlDMGblle

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Data’s statistical overview

for all data fields for specific data fields

14

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

describe() more in depth

df.describe()

df.describe(include = “all”)



https://images.app.goo.gl/byfMyVrWQKqEsFNk9 

15

https://images.app.goo.gl/6iiMJu9f9ohzVHcK9

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Handling duplicates

Duplicates are managed using drop\_duplicates()

INPUT: movie\_df.drop\_duplicates**( ) inplace = True** movie\_df movie\_df

\* \* \* \* \* \* 

\* \*

\* \*

\* \*

16

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

drop\_duplicates() options

INPUT: movie\_df.drop\_duplicates**( ) keep = ???**

• **first**: Drop duplicates except for the first occurrence. **(default)**

• **last**: Drop duplicates except for the last occurrence.

• **False**: Drop all duplicates.

Movie\_df **- first** Movie\_df **- last** Movie\_df - **False**

\* \* \* \* \* \*

\* \* \* \* \* \*

\* \*

\* \*

\* \*

17

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Get columns names with .columns **IN:**

**Popularity Movie\_type Screening**

**Title**

Kati kati 2 Drama 2016 The letter 1 Documentary 2019

.columns

**OUT:**

Index([’Popularity’, 'Movie\_type', 'Screening’], dtype='object')

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Rename columns

1. movies\_df.**rename**(columns={

‘Popularity': 'Rank',

'Movie\_type': 'Genre'

‘Screening’: ‘Year’ },inplace=True) Or …

2. movies\_df.**columns** = [‘Rank’, ‘Genre’, ‘Year’]

**Rank Genre Year Title**

Kati kati 2 Drama 2016 The letter 1 Documentary 2019

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Using list comprehension

You can iterate over an iterable, to modify a member object using an expressionmovies\_df.columns = [col.lower() for col in movies\_df]

**rank genre year**

**Title**

Kati kati 2 Drama 2016

The letter 1 Documentary 2019

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Identify missing values Where are the missing values?

movies\_df.isnull().sum()

**rank genre year**

**Title**

.isnull() 

Kati kati 2 2016 The letter 1 Documentary

**rank genre year**

**Title**

Kati kati False True False The letter False False True

rank 0 **genre 1 year 1** dtype: int64

.sum()

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Removing missing values in rows

Before .dropna(): After .dropna():

.dropna()

**rank genre year**

**Title**

Kati kati 2

The letter 1 Documentary 2019

**rank genre year**

**Title**

The letter 1 Documentary 2019

.dropna() is equal to .dropna(axis=0). ‘axis’ refers to the shape: 0=row, 1=column

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Removing missing values in columns

Before .dropna(axis = 1):

.dropn~~a(axis=1)~~

After .dropna(axis = 1):



**rank genre year**

**Title**

Kati kati 2

The letter 1 Documentary 2019

**rank**

**Title**

Kati kati 2

The letter 1

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Filling missing values (Imputation)

**rank genre year revenues\_m$**

**Title**

Kati kati 2 Drama 2016 5.6 The letter 1 Documentary 2019 8.9 Nairobi half-life 3 Drama 2012 Soul boy 4 Drama 2010 1.2

col.mean()= 5.23 **revenues\_m$**

col = movie\_df[‘revenues\_m$’] mean=col.mean()

col.fillna(mean, inplace = True)

col

5.6 8.9

col.fillna(movie\_df[‘revenues\_m$’] .mean(), inplace=True)

5.23 1.2

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Series and dataframes

movies\_df['genre']

type: series

movies\_df[['genre']] type: dataframe

**rank genre year revenues\_m$ Title**

Kati kati 2 Drama 2016 5.6 The letter 1 Documentary 2019 8.9 Nairobi half-life 3 Drama 2012 4.3 Soul boy 4 Drama 2010 1.2

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Extracting columns

movies\_df[['genre','year']]

type: dataframe

**rank genre year revenues\_m$**

**Title**

**Title**

**genre year**

Kati kati 2 Drama 2016 5.6 The letter 1 Documentary 2019 8.9

Nairobi half-life 3 Drama 2012 4.3 Soul boy 4 Drama 2010 1.2

Kati kati Drama 2016 The letter Documentary 2019

Nairobi half-life Drama 2012 Soul boy Drama 2010

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Extract a row by name

**rank genre year revenues\_m$**

**Title**

Kati kati 2 Drama 2016 5.6

The letter 1 Documentary 2019 8.9

Nairobi half-life 3 Drama 2012 4.3

Soul boy 4 Drama 2010 1.2

movie\_df.loc[‘The letter’]

**rank genre year revenues\_m$**

**Title**

****The letter 1 Documentary 2019 8.9

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Extract a row by index

**rank genre year revenues\_m$**

**Title**

Kati kati 2 Drama 2016 5.6

The letter 1 Documentary 2019 8.9

Nairobi half-life 3 Drama 2012 4.3

Soul boy 4 Drama 2010 1.2

movie\_df.iloc[1]

**rank genre year revenues\_m$**

**Title**

****The letter 1 Documentary 2019 8.9

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Extract multiple rows

**rank genre year revenues\_m$**

**Title**

Kati kati 2 Drama 2016 5.6 The letter 1 Documentary 2019 8.9 Nairobi half-life 3 Drama 2012 4.3

movie\_df.iloc[: 1] **Title**

movie\_df.loc[‘Kati kati’, ‘The letter’]**rank genre year revenues\_m$**

Kati kati 2 Drama 2016 5.6 The letter 1 Documentary 2019 8.9

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Conditional selections dataframesProprietary material, under Creative Commons 4.0 CC-BY-NC-ND 

Data Transformation

To make the data normally distributed and make them able to meet the assumptions of parametric statistical tests.



The Effect of Log Transformation. Source: https://www.medcalc.org/manual/log\_transformation.php

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Time Data

Convert time data to a usable format for manipulations.

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND 

Questions?

Thank you for your attention!



Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

33

https://images.app.goo.gl/TAaHUsDu1aVW2qc58

References and further readings

• https://pythongeeks.net/python-tutorials/python-pandas-for-beginners-a-complete-guide/ • https://towardsdatascience.com/python-for-data-science-basics-of-pandas-5f8d9680617e • https://www.learndatasci.com/tutorials/python-pandas-tutorial-complete-introduction-for-begin

ners/https://pandas.pydata.org/pandas-docs/stable/getting\_started/10min.html • https://www.shanelynn.ie/select-pandas-dataframe-rows-and-columns-using-iloc-loc-and-ix/ • https://www.dataquest.io/blog/pandas-cheat-sheet/

34

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND

Contributors

Elias Williams Jose Ignacio Diaz Pavithra Rajasekar Gilles Fayad

Proprietary material, under Creative Commons 4.0 CC-BY-NC-ND