PANDAS TOOL FOR DATA SCIENTIST

NOËL MOSES MWADENDE

ABOUT THE AUTHOR

Noel Moses Mwadende is currently 2 year Computer Science and Information Security student at University of Dodoma , Data Scientist and well specialized in python for ethical hacking ,Penetration Tester, Malware Analiysist,Hacking tools writer already written “INFORMATION GATHERING TOOL” mostly used for penetration testing phases of Information Gathering and Reconnaisance,writer of different books about programming language,hacking, malware and Computer security in general also a member of Udom CyberSec lab, After making several experiments in Udom CyberSec lab , Noel started to write different books concern security with great passion starting with his book of "WIFI HACKING IN FEWS STEPS" ,"PANDAS TOOL FOR DATA SCIENTIST","HOW TO MAN IN THE MIDDLE ATTACK",”MAKING OUR ANDROID TROJAN-HORSE”,”MY WIN TROJAN-HORSE” and "WRITE VIRUS BY BATCH PROGRAMMING" and still different books are being written , and I will keep on releasing my different written books for you my reader , but don’t get tired.

ACKNOWELDGEMENT

First and foremost, thanks to my instructors of Data Science Sir Salim Diwani, Mr Eliah and Miss Martha Shaka,without you all i couldn't be good Data Scientist and I could not be able to write this book , I appreciate you so much , as I did my Data Science project during IPT and it was very enjoyable can't forget those moments.

Thanks to my aunt Doctor Ignasia Mligo , currently instructor at UDOM, you have been encouraging me , also you have been my close parent since I arrived here at University.

Thanks to my security instructor Sir L Mutembei and my brother Joachim Mawole for insipering me , giving different ideas about data science and security issues, to be honest your inspiration to me is like the burning fuel to the fire.

Thanks to my best friend Damalicous Jons for encouraging me in my efforts of writing books , since we are friends ,you have been the closest friend for different

advice together with sharing different ideas.

Thanks to you everyone , I hope without reader no book can exist in this world.

TABLE OF CONTENTS

1. Introduction to Pandas
2. How Create And Start Project
3. Basic Anaconda Prompt Commands

4. Data cleaning

5. Advanced Ways Of Dealing With Data

6. Statistical Analysis

7. Data Visualization

CHAPTER ONE

INTODUCTION TO PANDAS

THE MEANING OF PANDAS

is the software library written for python programming for data manipulation and analysis.in particular it offers data structures and operations for

manipulating numerical tables and time series.it used to make the analysis of data which may be in the form of series,dataframe and panel, all those in pandas

we use to call them data structure, so manipulation of data and it's analysis is done by using data structure, in following chapter I will explain a little bit

about the above mentioned data structure in pandas , with full examples on each , though I will mainly base on data frame which is the mostly used type of data structure

in pandas , do not confuse with data structure being studied in programming language like C++, data structure in pandas has different meaing , now it is 2009, but pandas was

initial release in 11 January 2008, it is now about 11 years, as long the market for data science has been growing each and everyday also pandas as tool for data analysis has been getting repution.

CHAPTER TWO

HOW TO CREARE AND START A PROJECT

1.CREATING DATA SCIENCE PROJECT FOLDER

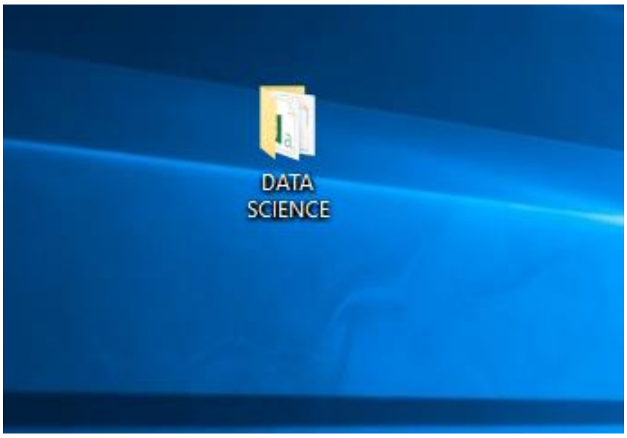
2.HOW TO CREATE JUPYTER NOTEBOOK PAGE AND RUNNIG IT

3. LOADING DATA SET ON JUPYTER NOOTEBOOK

4.HOW TO IMPORT DATA SCIENCE LIBRARIES

1. CREATING DATA SCIENCE PROJECT FOLDER

Step 1 : Create any folder , for the easiest and flexibility of the work save it on your desktop, in my computer I have create folder called DATA SCINCE on my desktop.



Step 2 : Click on panel labelled Desktop" , and you should have seen everything present on your desktop.

Step 3 : Click the folder you have created in step one , on my side I click the folder named DATA SCIENCE, finally you will be able to see everything inside it.

Output From My Screen :

2. HOW TO CREATE JUPYTER NOTEBOOK PAGE AND RUNNIG IT

I. Underneath of right right corner you will see the button named new, right click on that , you will see the following options

-Python 3

I. -other

-Text File

-Folder

-Terminal

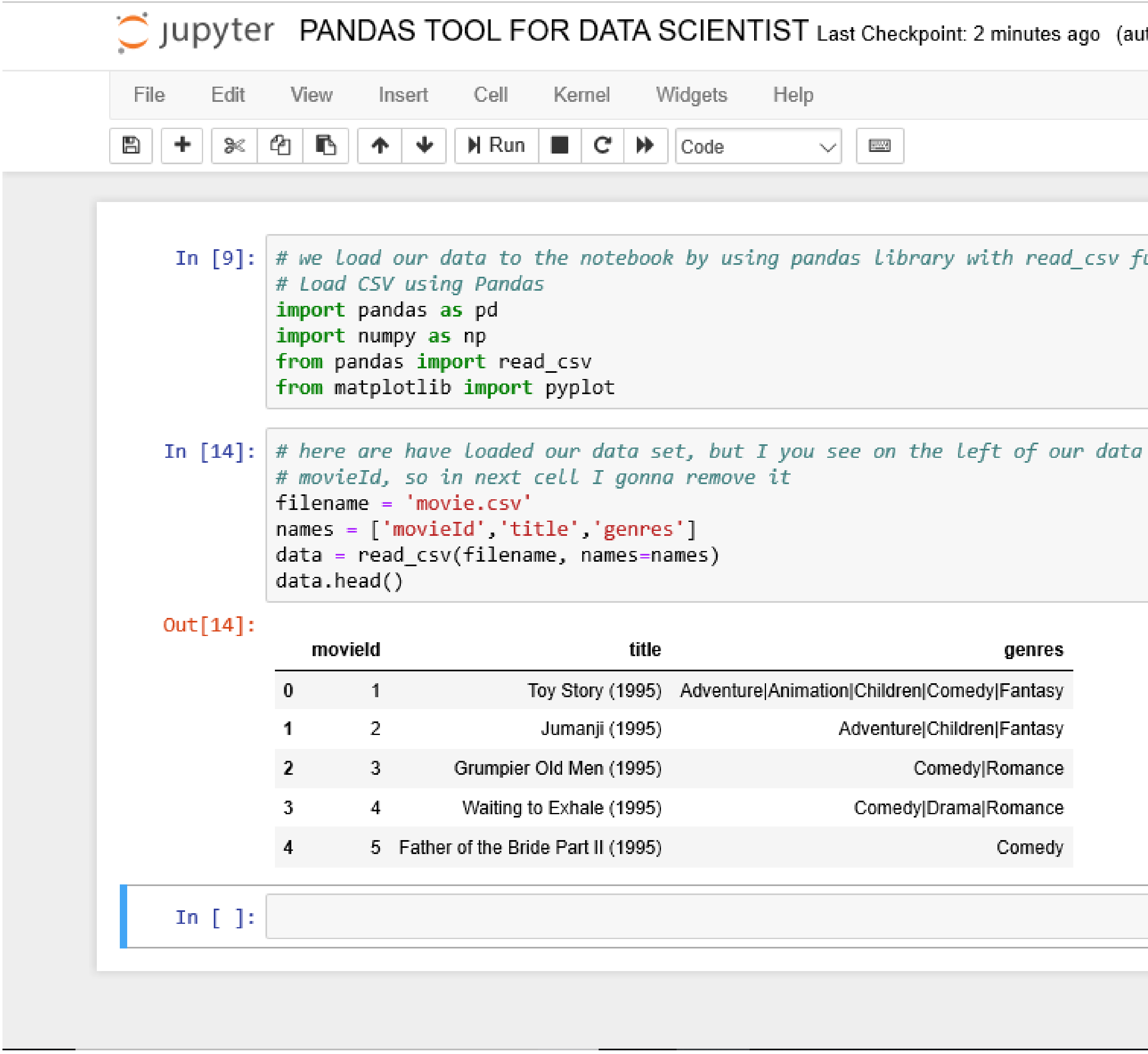
click on Python 3 , then will have created untittle notebook

II. On the top click the place named untitled and give the name to your page , on my side I gave the name "my first notebook"

III. Testing our notebook , write the following print("Pandas Tool For Data Scientist")

IV. Run it , on the top of your notebook , click the button name to run

LOADING DATA SET ON JUPYTER NOOTEBOOK


jupyter
PANDAS TOOL FOR DATA SCIENTIST Last Checkpoint: 2 minutes ago (au
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

In [9]:

```
# we load our data to the notebook by using pandas library with read_csv fu
# Load CSV using Pandas
import pandas as pd
import numpy as np
from pandas import read_csv
from matplotlib import pyplot
```

In [14]:

```
# here are have loaded our data set, but I you see on the left of our data
# movieId, so in next cell I gonna remove it
filename = 'movie.csv'
names = ['movieId','title','genres']
data = read_csv(filename, names=names)
data.head()
```

Out[14]:

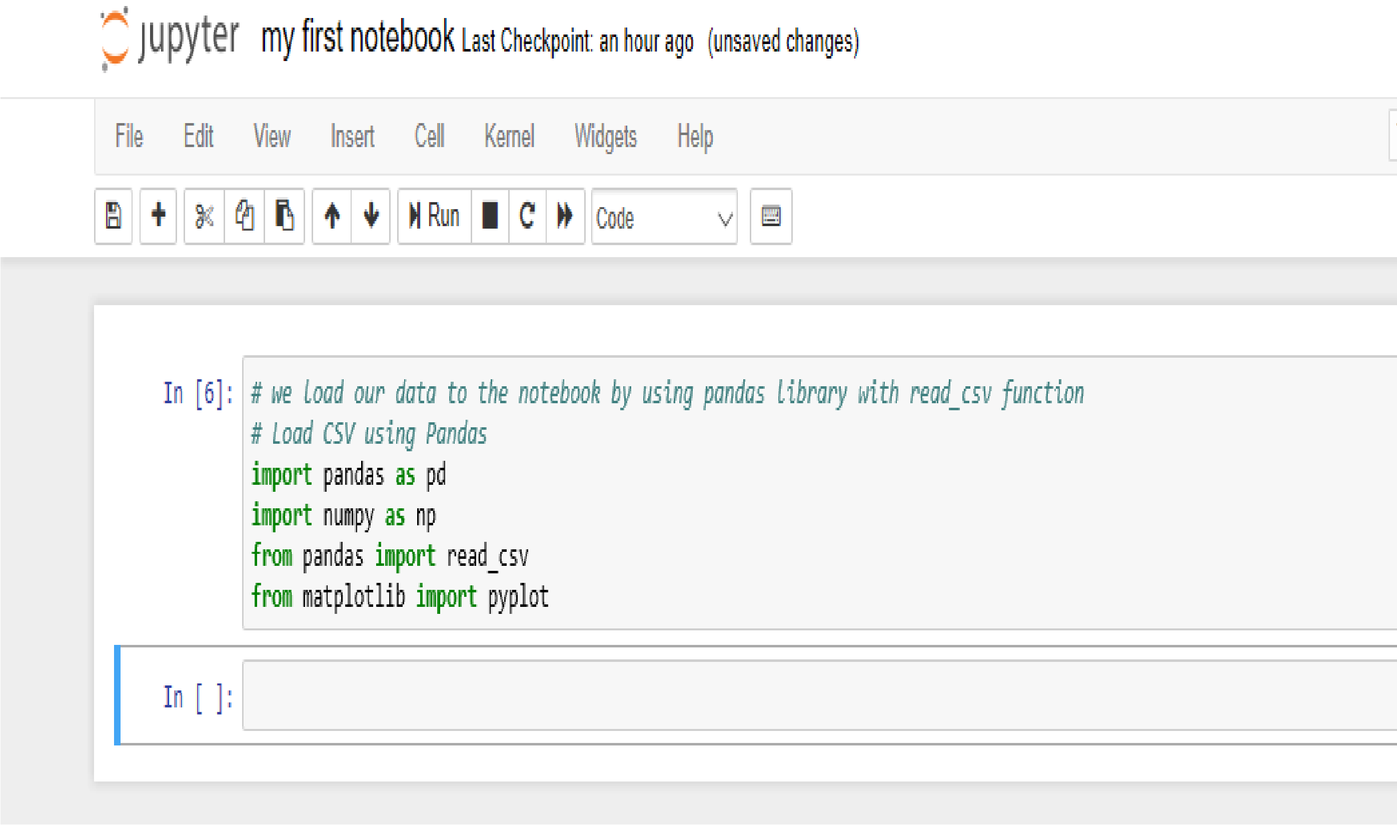
```
   movieId                        title                                       genres
0        1             Toy Story (1995)  Adventure|Animation|Children|Comedy|Fantasy
1        2               Jumanji (1995)                   Adventure|Children|Fantasy
2        3      Grumpier Old Men (1995)                               Comedy|Romance
3        4     Waiting to Exhale (1995)                         Comedy|Drama|Romance
4        5  Father of the Bride Part II (1995)                                Comedy
```

In [ ]:

HOW TO IMPORT DATA SCIENCE LIBRARIES

I recommend to import all libraries and modules you know you will be using on the top of your notebook, After importing them click the button named run to include them in you project.

Output From My Screen :


jupyter my first notebook Last Checkpoint: an hour ago (unsaved changes)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

```
In [6]: # we load our data to the notebook by using pandas library with read_csv function
        # Load CSV using Pandas
        import pandas as pd
        import numpy as np
        from pandas import read_csv
        from matplotlib import pyplot

In [ ]:
```

Two ways to import libraries or Modules

A. from library import module , here you will have to know what library and what module you're importing

Example:

```
from pandas import read_csv
```

On above code of line , you can see that we use library called Pandas to import read_csv function which is for reading file in csv format.

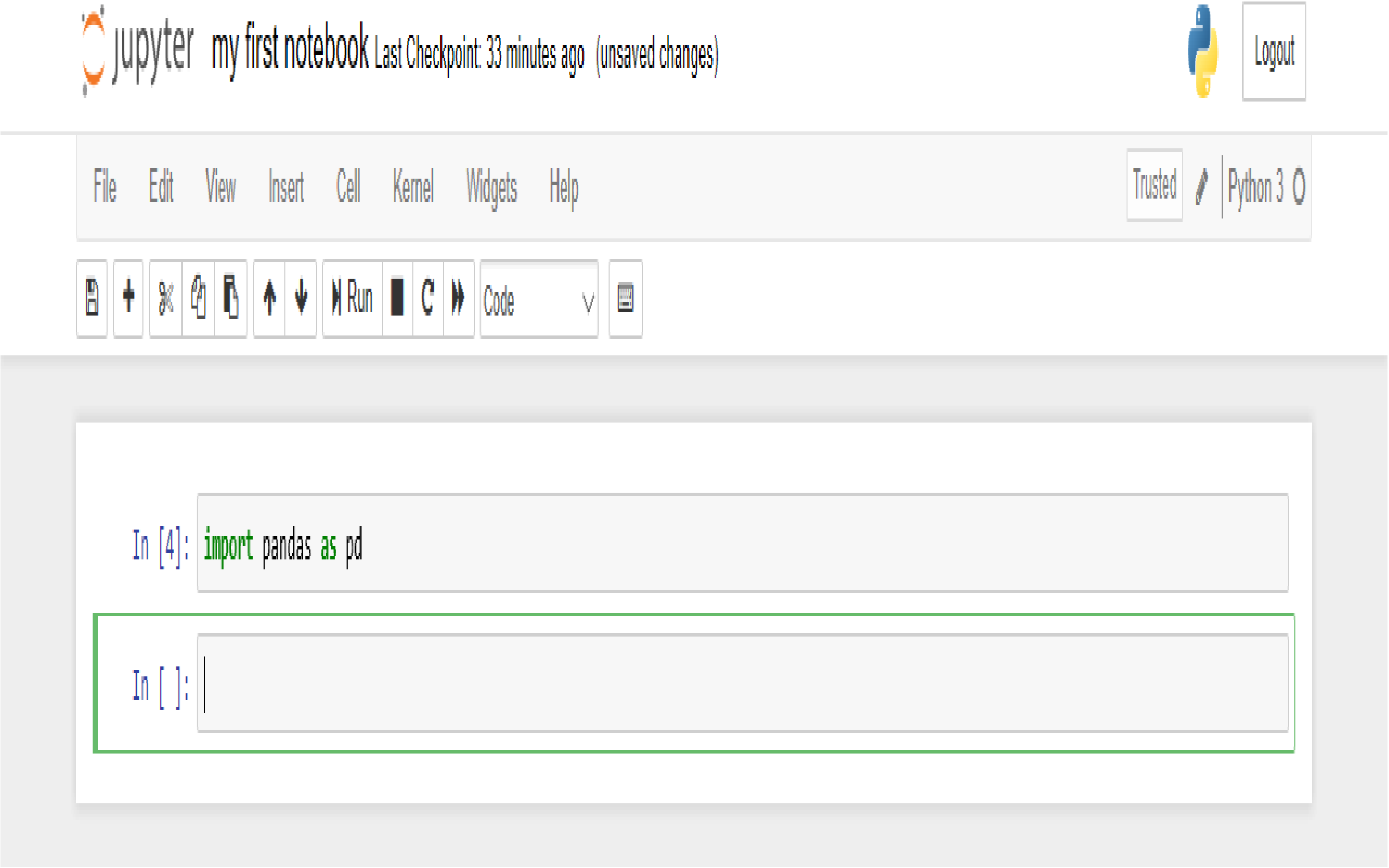
Output From My Screen :

B. import library , here you just need to know the name of library

Example:

```
import pandas as pd
```

Output From My Screen :



```
In [4]: import pandas as pd

In [ ]:
```

CHAPTER THREE

BASIC ANACONDA PROMPT COMMANDS

- anaconda-navigator
- conda info
- conda update -n base conda
- conda update anaconda
- conda clean –index-cache
- conda list
- conda config --show

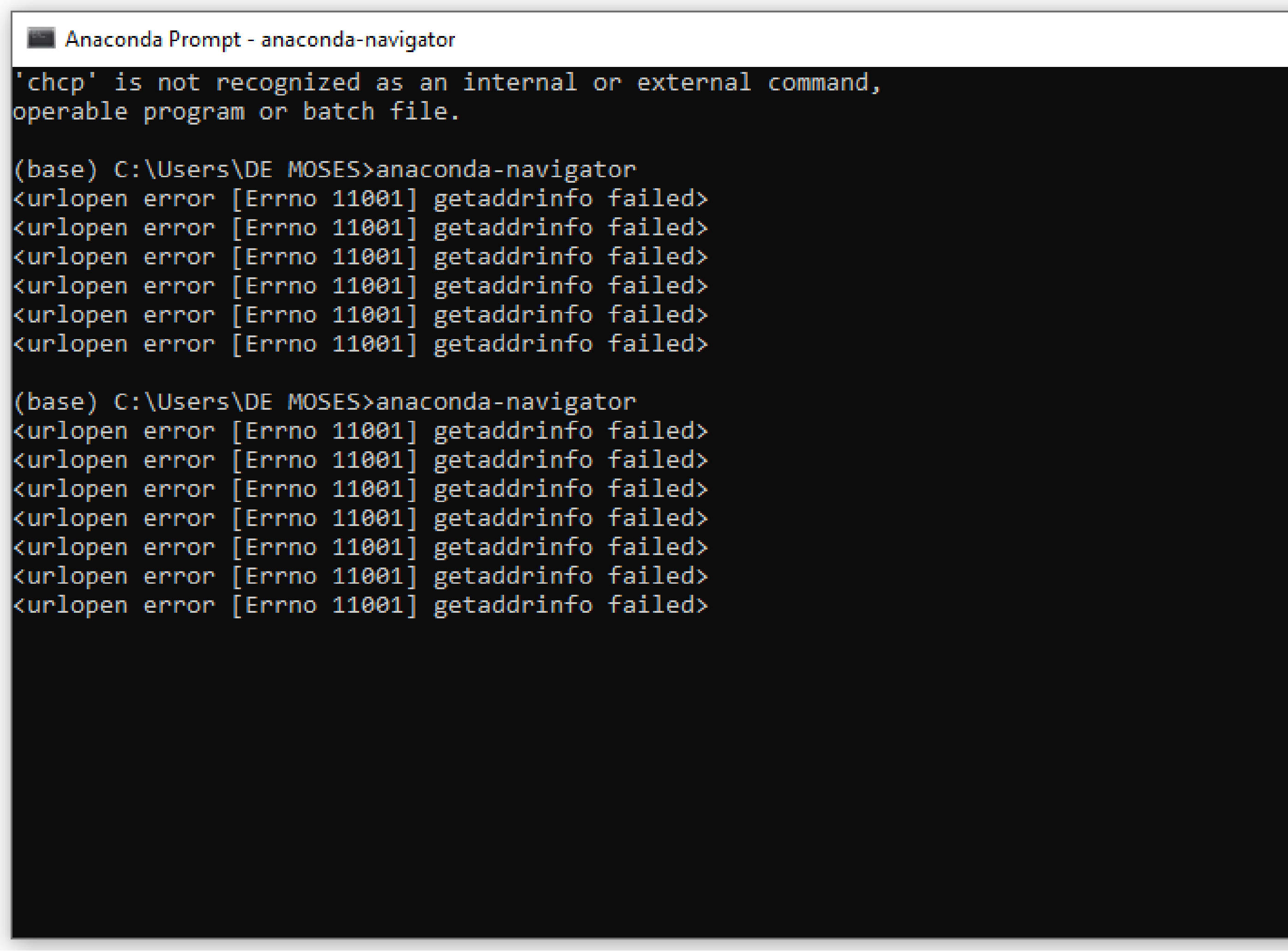
I have added this part as I know how it is important for , most of Data Scientist have came across Ed Linux and they are also familiar with Command Promt (CMD),

so I hope you the power of command , it works fine, here below am going to explain som few command with their function and how you can benefit from them

- anaconda-navigator

This command is used for starting Anaconda Navigator

Output From My Screen :

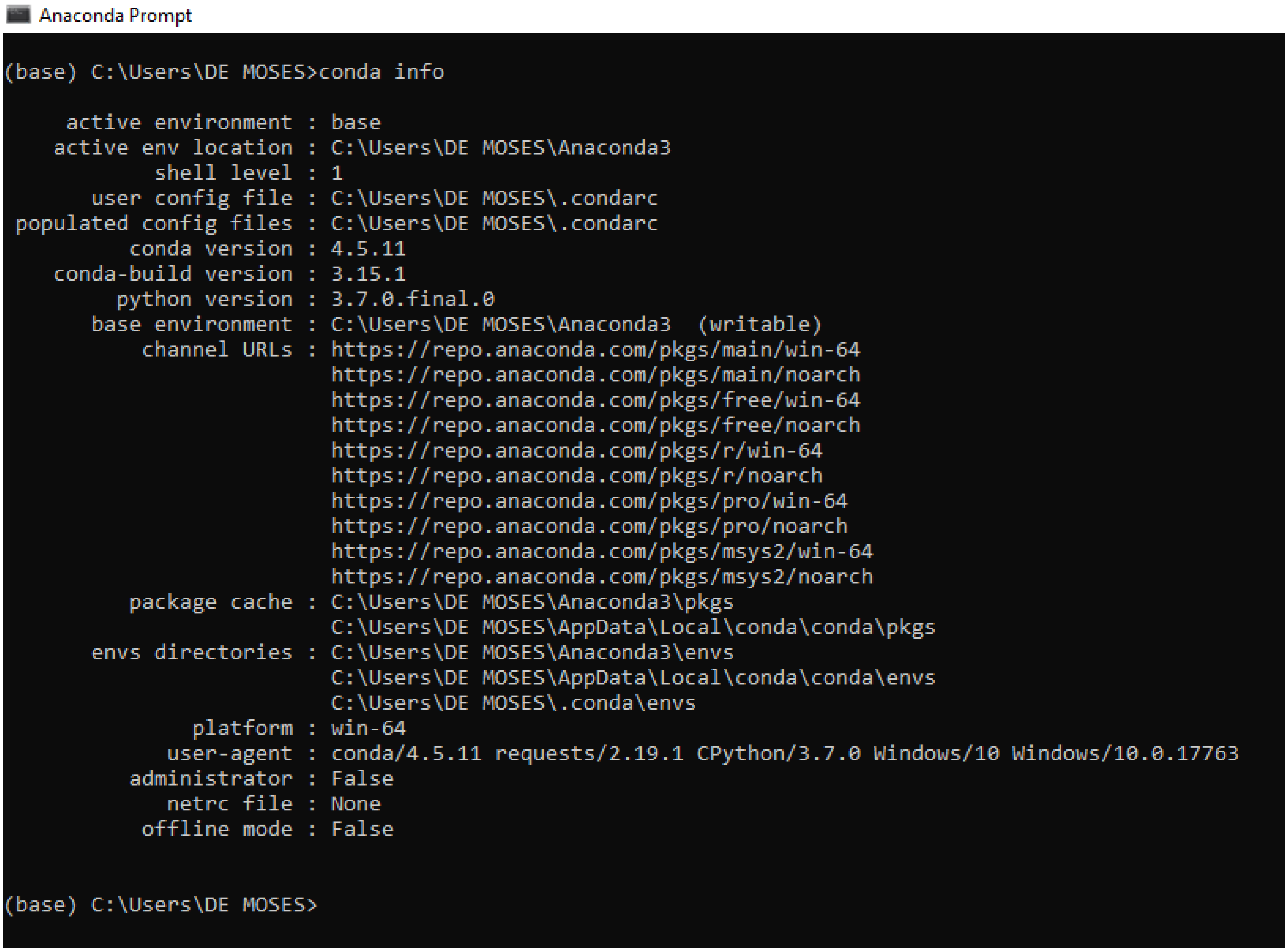


```
'chcp' is not recognized as an internal or external command,
operable program or batch file.

(base) C:\Users\DE MOSES>anaconda-navigator
<urlopen error [Errno 11001] getaddrinfo failed>
<urlopen error [Errno 11001] getaddrinfo failed>
<urlopen error [Errno 11001] getaddrinfo failed>
<urlopen error [Errno 11001] getaddrinfo failed>
<urlopen error [Errno 11001] getaddrinfo failed>
<urlopen error [Errno 11001] getaddrinfo failed>

(base) C:\Users\DE MOSES>anaconda-navigator
<urlopen error [Errno 11001] getaddrinfo failed>
<urlopen error [Errno 11001] getaddrinfo failed>
<urlopen error [Errno 11001] getaddrinfo failed>
<urlopen error [Errno 11001] getaddrinfo failed>
<urlopen error [Errno 11001] getaddrinfo failed>
<urlopen error [Errno 11001] getaddrinfo failed>
<urlopen error [Errno 11001] getaddrinfo failed>
```

- conda info

This is used to check different information about Anaconda , it is like "systeminfo" in windows Command Prompt.

Output From My Screen :

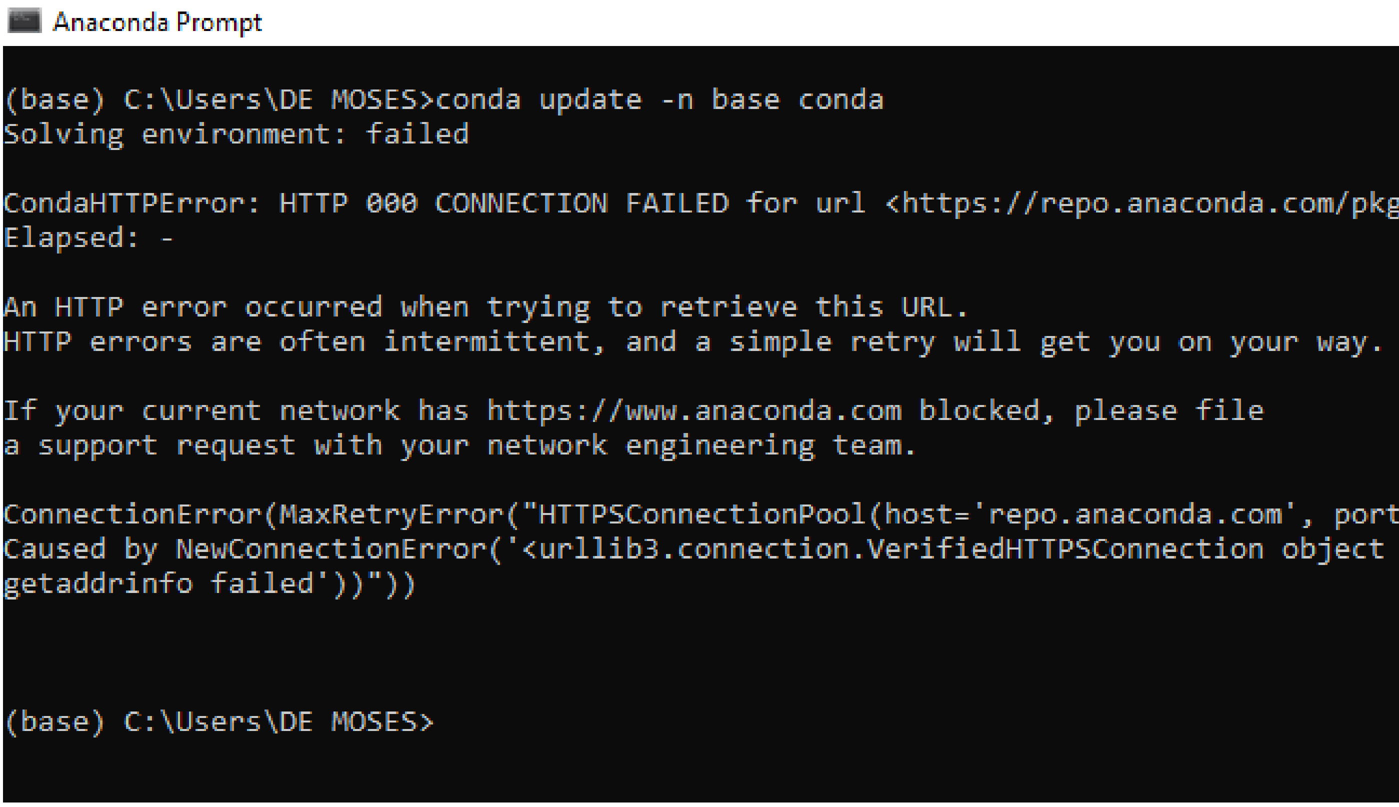


```
(base) C:\Users\DE MOSES>conda info

     active environment : base
    active env location : C:\Users\DE MOSES\Anaconda3
            shell level : 1
       user config file : C:\Users\DE MOSES\.condarc
 populated config files : C:\Users\DE MOSES\.condarc
          conda version : 4.5.11
    conda-build version : 3.15.1
         python version : 3.7.0.final.0
       base environment : C:\Users\DE MOSES\Anaconda3  (writable)
           channel URLs : https://repo.anaconda.com/pkgs/main/win-64
                          https://repo.anaconda.com/pkgs/main/noarch
                          https://repo.anaconda.com/pkgs/free/win-64
                          https://repo.anaconda.com/pkgs/free/noarch
                          https://repo.anaconda.com/pkgs/r/win-64
                          https://repo.anaconda.com/pkgs/r/noarch
                          https://repo.anaconda.com/pkgs/pro/win-64
                          https://repo.anaconda.com/pkgs/pro/noarch
                          https://repo.anaconda.com/pkgs/msys2/win-64
                          https://repo.anaconda.com/pkgs/msys2/noarch
          package cache : C:\Users\DE MOSES\Anaconda3\pkgs
                          C:\Users\DE MOSES\AppData\Local\conda\conda\pkgs
       envs directories : C:\Users\DE MOSES\Anaconda3\envs
                          C:\Users\DE MOSES\AppData\Local\conda\conda\envs
                          C:\Users\DE MOSES\.conda\envs
               platform : win-64
             user-agent : conda/4.5.11 requests/2.19.1 CPython/3.7.0 Windows/10 Windows/10.0.17763
          administrator : False
             netrc file : None
           offline mode : False


(base) C:\Users\DE MOSES>
```

❖ conda update -n base conda

This command is used for updating Anaconda to the current version

Output From My Screen :



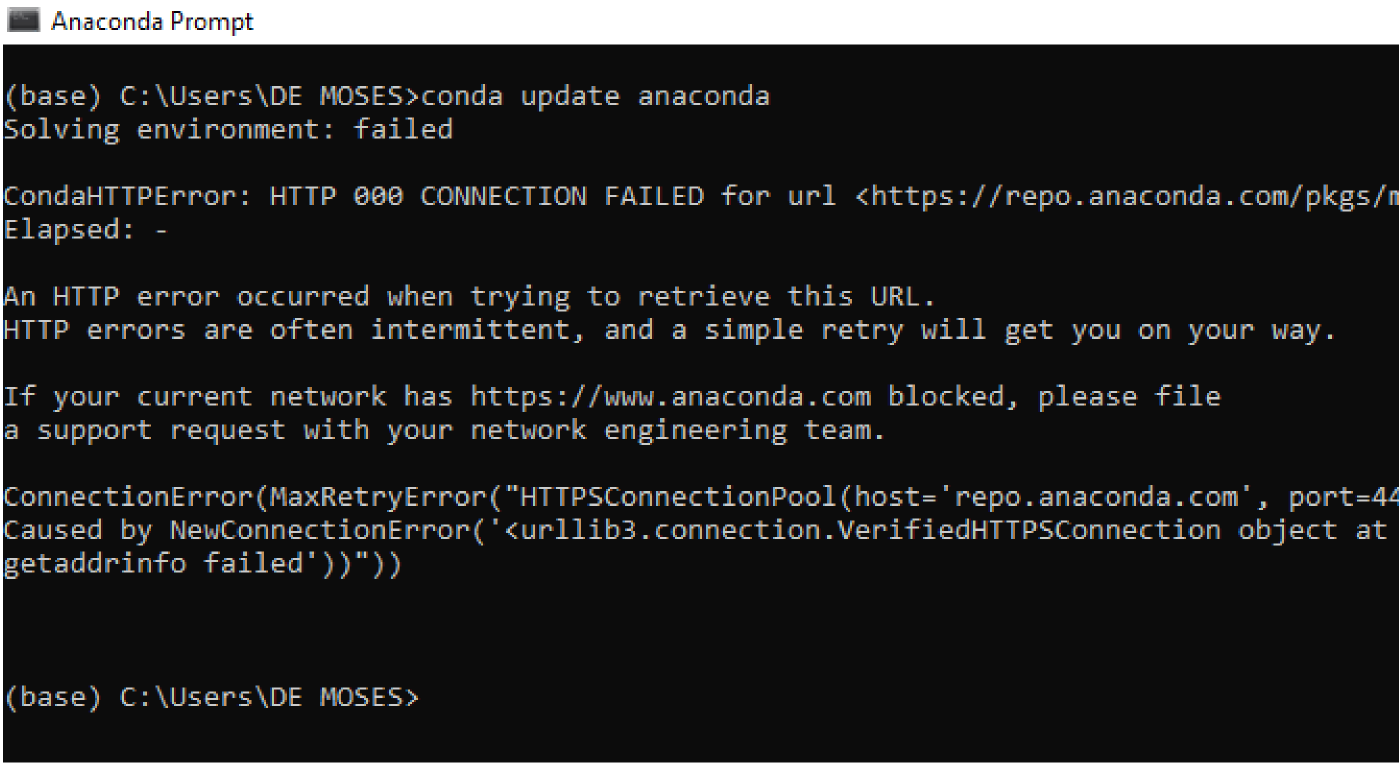
Anaconda Prompt

```
(base) C:\Users\DE MOSES>conda update -n base conda
Solving environment: failed

CondaHTTPError: HTTP 000 CONNECTION FAILED for url <https://repo.anaconda.com/pkg
Elapsed: -

An HTTP error occurred when trying to retrieve this URL.
HTTP errors are often intermittent, and a simple retry will get you on your way.

If your current network has https://www.anaconda.com blocked, please file
a support request with your network engineering team.

ConnectionError(MaxRetryError("HTTPSConnectionPool(host='repo.anaconda.com', port
Caused by NewConnectionError('<urllib3.connection.VerifiedHTTPSConnection object
getaddrinfo failed'))"))



(base) C:\Users\DE MOSES>
```

- conda update anaconda

This is used for updating all packages in anaconda.

Output From My Screen :



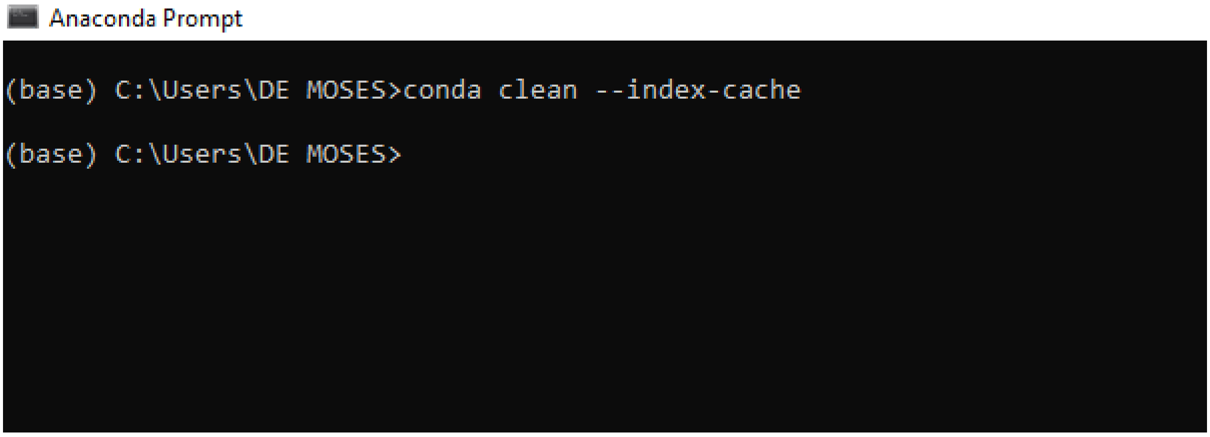
Anaconda Prompt

```
(base) C:\Users\DE MOSES>conda update anaconda
Solving environment: failed

CondaHTTPError: HTTP 000 CONNECTION FAILED for url <https://repo.anaconda.com/pkgs/m
Elapsed: -

An HTTP error occurred when trying to retrieve this URL.
HTTP errors are often intermittent, and a simple retry will get you on your way.

If your current network has https://www.anaconda.com blocked, please file
a support request with your network engineering team.

ConnectionError(MaxRetryError("HTTPSConnectionPool(host='repo.anaconda.com', port=44
Caused by NewConnectionError('<urllib3.connection.VerifiedHTTPSConnection object at
getaddrinfo failed'))"))



(base) C:\Users\DE MOSES>
```

❖ conda clean –index-cache

This command is used to clean all unused cached files, and unused package , so it will boost up Anaconda performance.

Output From My Screen :



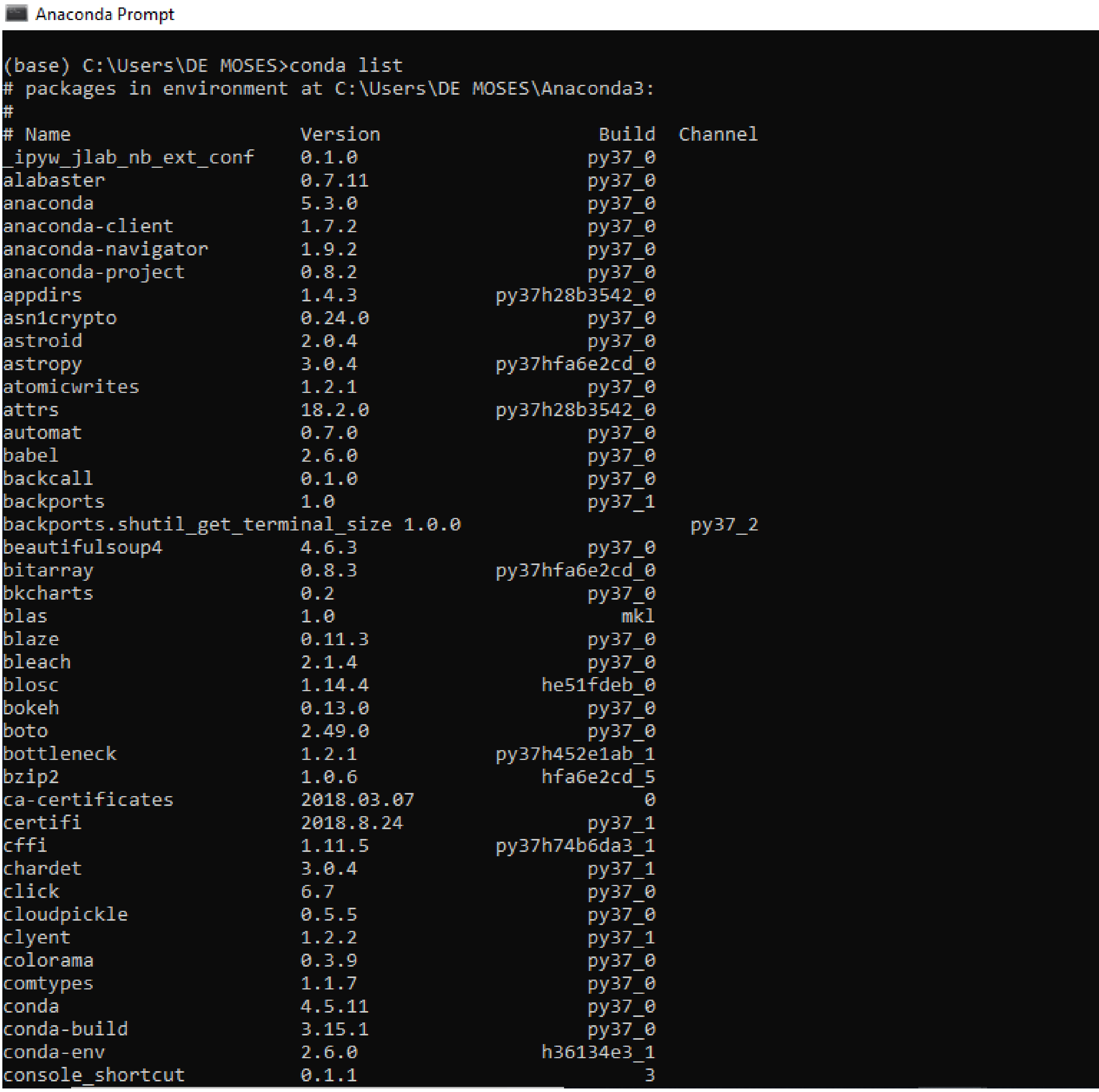
Anaconda Prompt

```
(base) C:\Users\DE MOSES>conda clean --index-cache

(base) C:\Users\DE MOSES>
```

❖ conda list

This command will list packages and versions in the active environment.

Output From My Screen :

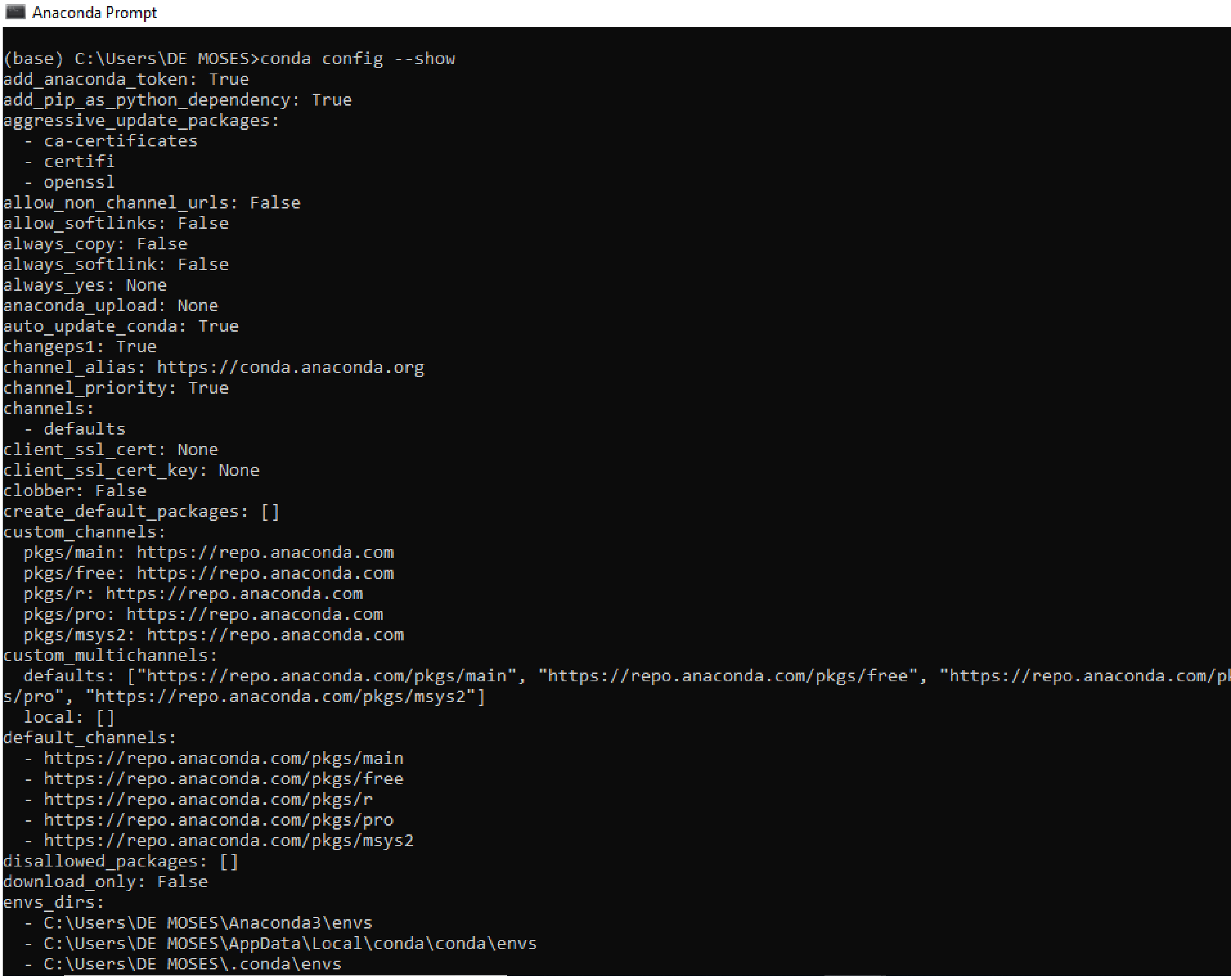


```
(base) C:\Users\DE MOSES>conda list
# packages in environment at C:\Users\DE MOSES\Anaconda3:
#
# Name                    Version                   Build  Channel
_ipyw_jlab_nb_ext_conf    0.1.0                    py37_0
alabaster                 0.7.11                   py37_0
anaconda                  5.3.0                    py37_0
anaconda-client           1.7.2                    py37_0
anaconda-navigator        1.9.2                    py37_0
anaconda-project          0.8.2                    py37_0
appdirs                   1.4.3            py37h28b3542_0
asn1crypto                0.24.0                   py37_0
astroid                   2.0.4                    py37_0
astropy                   3.0.4            py37hfa6e2cd_0
atomicwrites              1.2.1                    py37_0
attrs                     18.2.0           py37h28b3542_0
automat                   0.7.0                    py37_0
babel                     2.6.0                    py37_0
backcall                  0.1.0                    py37_0
backports                 1.0                      py37_1
backports.shutil_get_terminal_size 1.0.0                    py37_2
beautifulsoup4            4.6.3                    py37_0
bitarray                  0.8.3            py37hfa6e2cd_0
bkcharts                  0.2                      py37_0
blas                      1.0                         mkl
blaze                     0.11.3                   py37_0
bleach                    2.1.4                    py37_0
blosc                     1.14.4               he51fdeb_0
bokeh                     0.13.0                   py37_0
boto                      2.49.0                   py37_0
bottleneck                1.2.1            py37h452e1ab_1
bzip2                     1.0.6                hfa6e2cd_5
ca-certificates           2018.03.07                    0
certifi                   2018.8.24                py37_1
cffi                      1.11.5           py37h74b6da3_1
chardet                   3.0.4                    py37_1
click                     6.7                      py37_0
cloudpickle               0.5.5                    py37_0
clyent                    1.2.2                    py37_1
colorama                  0.3.9                    py37_0
comtypes                  1.1.7                    py37_0
conda                     4.5.11                   py37_0
conda-build               3.15.1                   py37_0
conda-env                 2.6.0                h36134e3_1
console_shortcut          0.1.1                         3
```

❖ conda config --show

This command would show all Anaconda configuration information.

Output From My Screen :



```
(base) C:\Users\DE MOSES>conda config --show
add_anaconda_token: True
add_pip_as_python_dependency: True
aggressive_update_packages:
  - ca-certificates
  - certifi
  - openssl
allow_non_channel_urls: False
allow_softlinks: False
always_copy: False
always_softlink: False
always_yes: None
anaconda_upload: None
auto_update_conda: True
changeps1: True
channel_alias: https://conda.anaconda.org
channel_priority: True
channels:
  - defaults
client_ssl_cert: None
client_ssl_cert_key: None
clobber: False
create_default_packages: []
custom_channels:
  pkgs/main: https://repo.anaconda.com
  pkgs/free: https://repo.anaconda.com
  pkgs/r: https://repo.anaconda.com
  pkgs/pro: https://repo.anaconda.com
  pkgs/msys2: https://repo.anaconda.com
custom_multichannels:
  defaults: ["https://repo.anaconda.com/pkgs/main", "https://repo.anaconda.com/pkgs/free", "https://repo.anaconda.com/p
s/pro", "https://repo.anaconda.com/pkgs/msys2"]
  local: []
default_channels:
  - https://repo.anaconda.com/pkgs/main
  - https://repo.anaconda.com/pkgs/free
  - https://repo.anaconda.com/pkgs/r
  - https://repo.anaconda.com/pkgs/pro
  - https://repo.anaconda.com/pkgs/msys2
disallowed_packages: []
download_only: False
envs_dirs:
  - C:\Users\DE MOSES\Anaconda3\envs
  - C:\Users\DE MOSES\AppData\Local\conda\conda\envs
  - C:\Users\DE MOSES\.conda\envs
```

Those are few basic Anaconda Prompt , not much necessary for beginner in this field but as long as become big data scientist , you should know a lot of things as I say

“Powerful minds are those with many datas”

CHAPTER FOUR

HOW TO CREATE AND START A PROJECT

1. HOW TO LOAD DATA VIA CSV

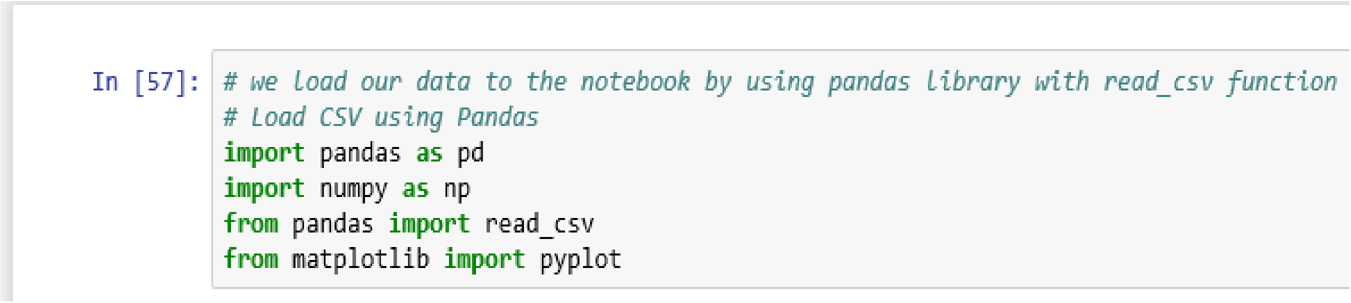
2.RUNNING YOUR FILES

3.TROUBLESHOTING KERNEL

1. HOW TO LOAD DATA VIA CSV

This is one of the foremost important issue to know in data science, because nothing can happen without loading your data on jupyter notebook, we read data by function called read_csv(“parameter to be passed”), by issuing data.head() you can see your first five rows of your dataframe. But you should import all needed and important libraries , without forgetting importing from pandas import read_csv(), which is function we are using.

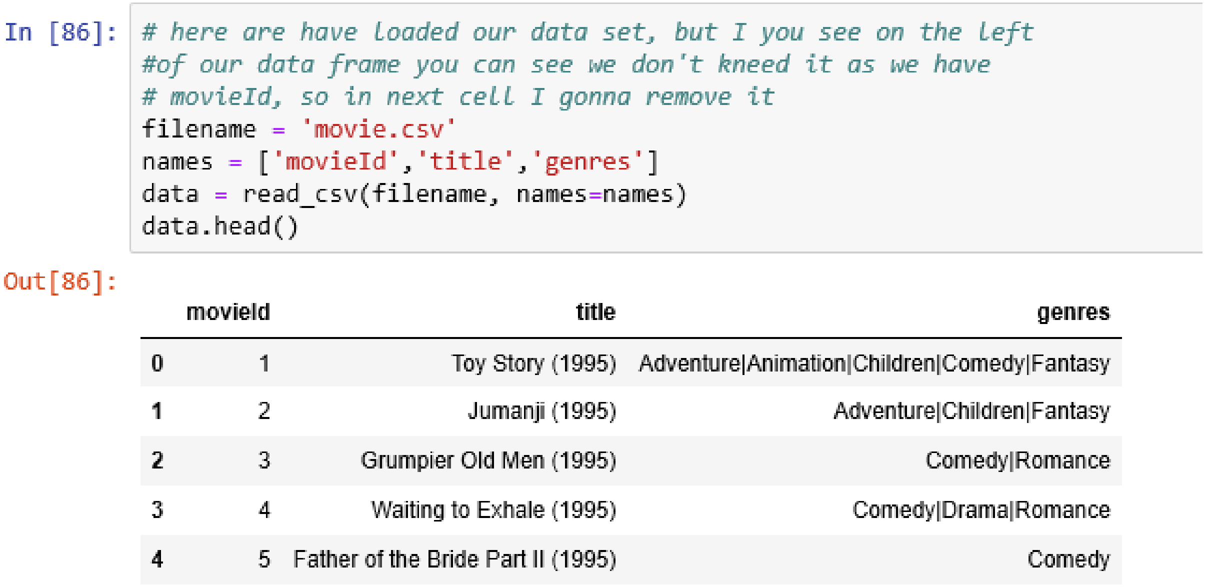
IMPORTING LIBRARIES



In [57]:

```
# we load our data to the notebook by using pandas library with read_csv function
# Load CSV using Pandas
import pandas as pd
import numpy as np
from pandas import read_csv
from matplotlib import pyplot
```

LOADING DATA



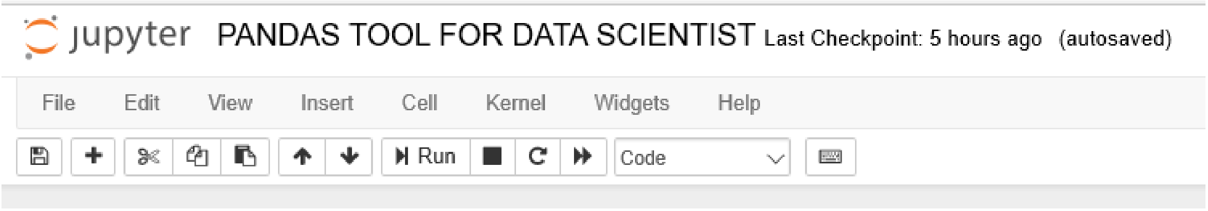
In [86]:

```
# here are have loaded our data set, but I you see on the left
#of our data frame you can see we don't kneed it as we have
# movieId, so in next cell I gonna remove it
filename = 'movie.csv'
names = ['movieId','title','genres']
data = read_csv(filename, names=names)
data.head()
```

Out[86]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

3.RUNNING YOUR FILES


jupyter
PANDAS TOOL FOR DATA SCIENTIST Last Checkpoint: 5 hours ago (autosaved)
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

-PANDAS TOOL FOR DATA SCIENTIST

can be escaped. This is the name of our notebook,file or page being used

-KERNEL

This is like engine of our anaconda navigator, due to some issues it may unexpected stopped , so remember to restart and run all cell.

-CELL

This is the single panel you use for the project.

-RUN

This is the button which when clicked ,it will bring the output.

-EDIT

This is like it express it'selft it is used for cutting cell, move cell, and all about editing cell.

-FILE

This is used for creating new notebook .

CHAPTER FIVE

DATA CLEANING

1.DEAL WITH MISSING VALUE

2.REMOVING COLUMN FROM DATA FRAME

3.REMOVING ROW FROM DATA FRAME

4. REMOVING DUPLICATED DATA

1.DEALING WITH MISSING VALUE

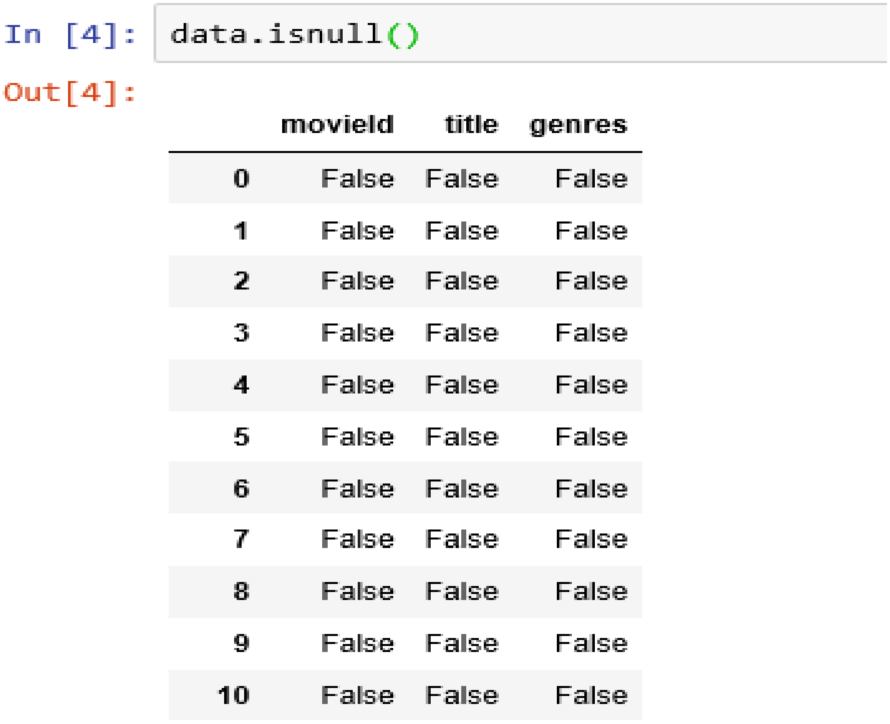
Here we are first to check which data is missing in our data set , if yes we gonna fix this issue , but if no missing data this can be escaped.

Isnull(), can be used to check if there are data missing as the function sai it self is null, so if there is no missing data it will return false, but if there is missing data it will return true, it is like you are asking the function

You : isnull()

Function : false if no missing data or true if there is data missing

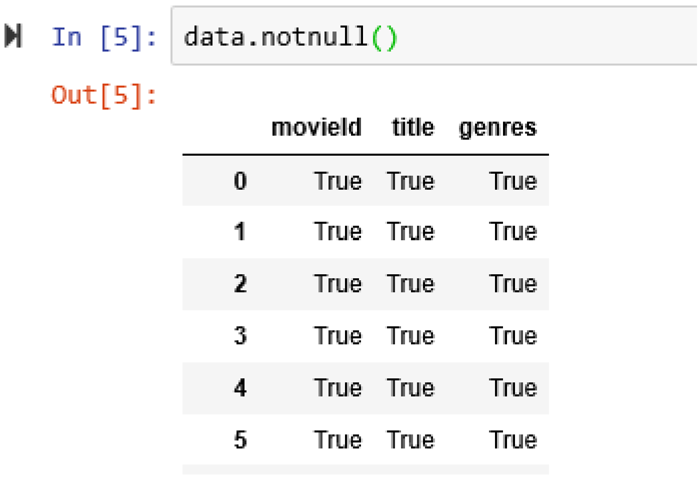
The function is clear as you can see that it self describe it’s work.



```
In [4]: data.isnull()
```

Out[4]:

	movieId	title	genres
0	False	False	False
1	False	False	False
2	False	False	False
3	False	False	False
4	False	False	False
5	False	False	False
6	False	False	False
7	False	False	False
8	False	False	False
9	False	False	False
10	False	False	False

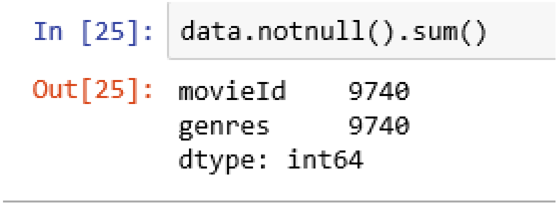


```
In [5]: data.notnull()
Out[5]:
```

	movieId	title	genres
0	True	True	True
1	True	True	True
2	True	True	True
3	True	True	True
4	True	True	True
5	True	True	True

notnull() is also used to check if there is missing value, but it will return true if non data is missing.

data.notnull().sum() is used to bring the sum of all missing values , let’s work it works

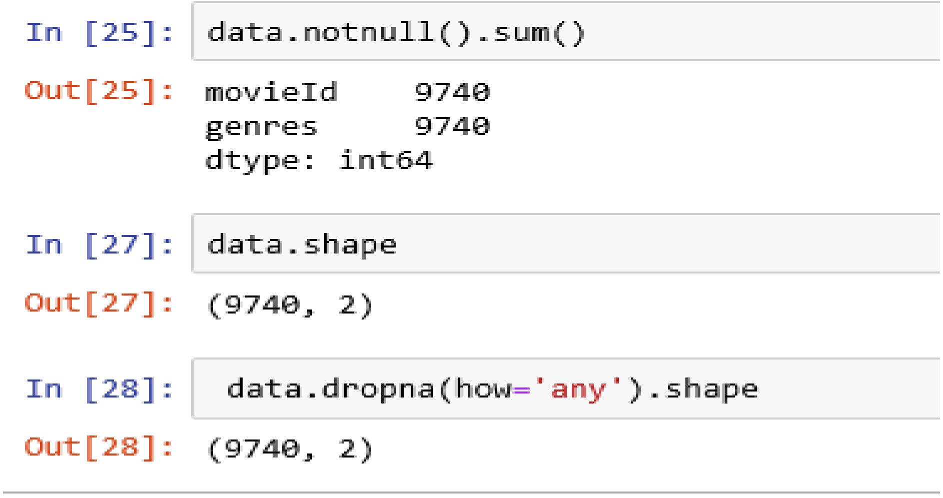


```
In [25]: data.notnull().sum()
Out[25]: movieId     9740
         genres      9740
         dtype: int64
```

It works, You can see all data are full non is missing.

HOW TO DROP MISSING VALUES

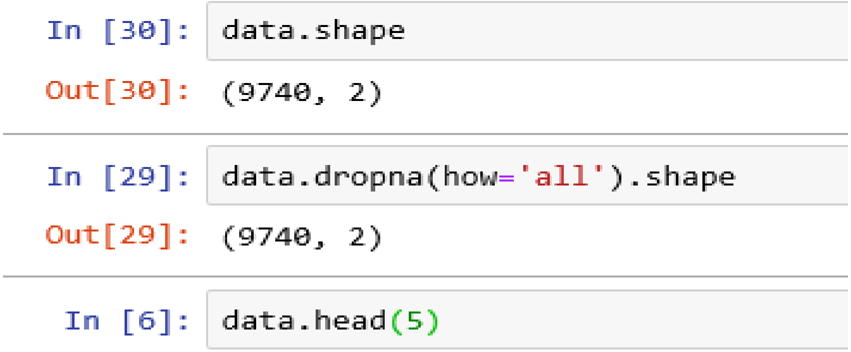
data.dropna(how=’any’).shape, this function will delete any row with missing values.



```
In [25]: data.notnull().sum()

Out[25]: movieId     9740
         genres      9740
         dtype: int64

In [27]: data.shape

Out[27]: (9740, 2)

In [28]:  data.dropna(how='any').shape

Out[28]: (9740, 2)
```

values. If you are carefully , you may have observe that the shape of dataset before and after dropping missing value remain the same, and our shape is (9740,2), that means no any row or column has been dropped as we don’t have missing values.

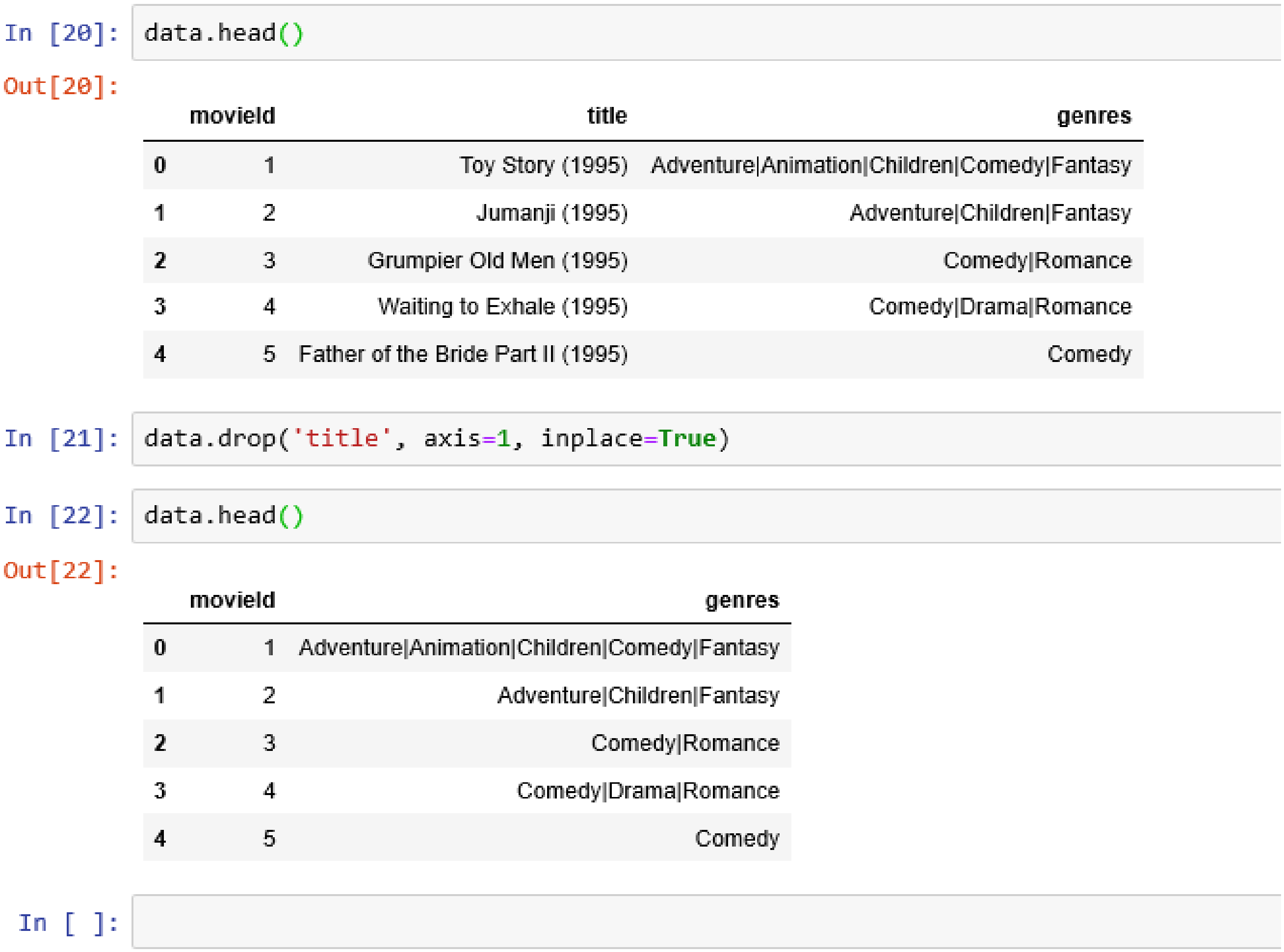
But data.dropna(how=’all’).shape function works the same , let’s do it and see below the output it must be the same with data.dropna(how=’any’).shape,



```
In [30]: data.shape

Out[30]: (9740, 2)

In [29]: data.dropna(how='all').shape

Out[29]: (9740, 2)

 In [6]: data.head(5)
```

2.REMOVING COLUMN FROM DATA FRAME

The function drop can be used to remove a column but thing to remember is axis=1 means it is row.



```
In [20]: data.head()
Out[20]:
      movieId                         title                                        genres
   0        1              Toy Story (1995)   Adventure|Animation|Children|Comedy|Fantasy
   1        2                Jumanji (1995)                    Adventure|Children|Fantasy
   2        3       Grumpier Old Men (1995)                                Comedy|Romance
   3        4      Waiting to Exhale (1995)                          Comedy|Drama|Romance
   4        5  Father of the Bride Part II (1995)                                  Comedy
```

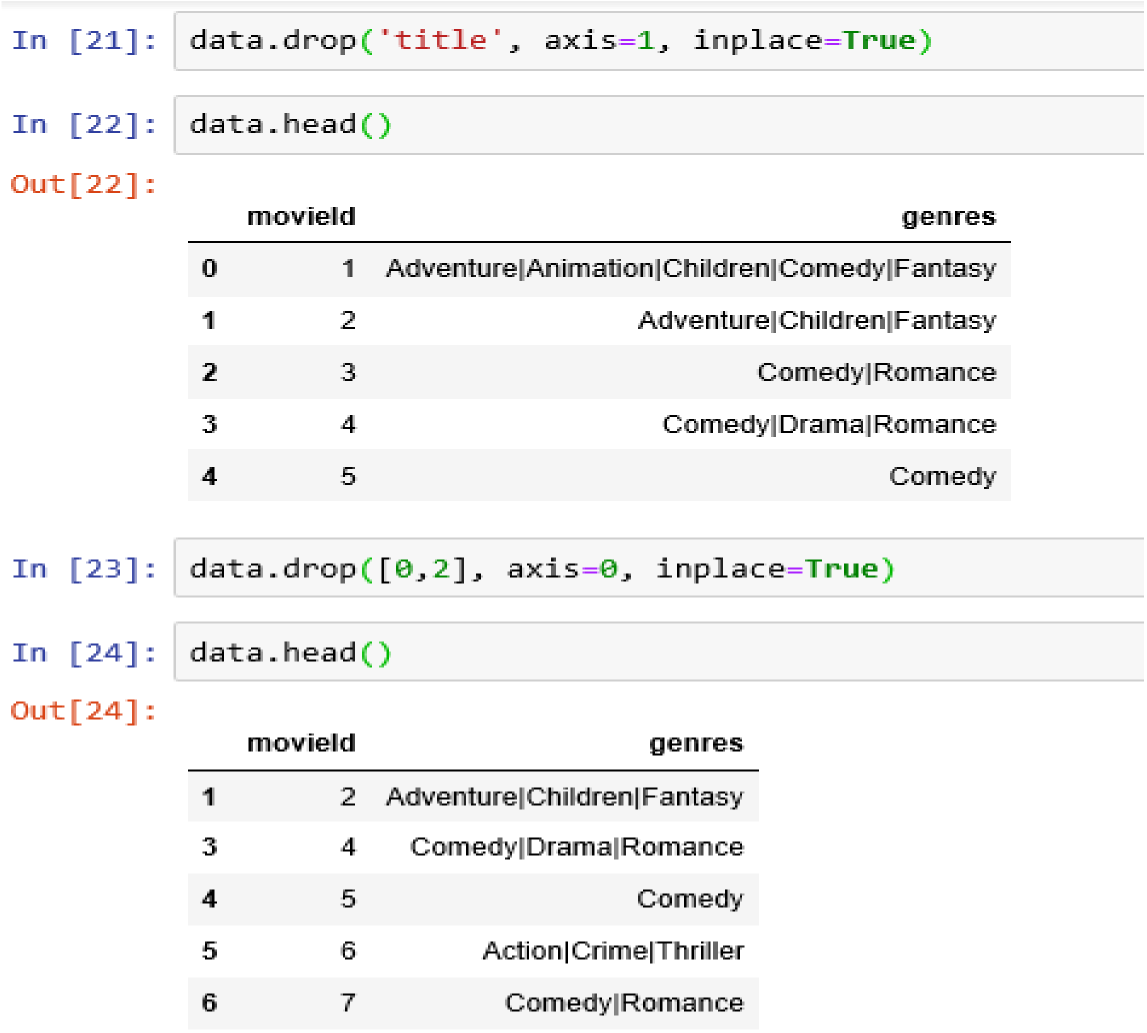
```
In [21]: data.drop('title', axis=1, inplace=True)
```

```
In [22]: data.head()
Out[22]:
      movieId                                        genres
   0        1   Adventure|Animation|Children|Comedy|Fantasy
   1        2                    Adventure|Children|Fantasy
   2        3                                Comedy|Romance
   3        4                          Comedy|Drama|Romance
   4        5                                        Comedy
```

```
In [ ]:
```

3.REMOVING ROW FROM DATA FRAME

The function drop can be used to remove a row but thing to remember is axis=o means it is row.



```
In [21]: data.drop('title', axis=1, inplace=True)
```

```
In [22]: data.head()
```

Out[22]:

	movieId	genres
0	1	Adventure|Animation|Children|Comedy|Fantasy
1	2	Adventure|Children|Fantasy
2	3	Comedy|Romance
3	4	Comedy|Drama|Romance
4	5	Comedy

```
In [23]: data.drop([0,2], axis=0, inplace=True)
```

```
In [24]: data.head()
```

Out[24]:

	movieId	genres
1	2	Adventure|Children|Fantasy
3	4	Comedy|Drama|Romance
4	5	Comedy
5	6	Action|Crime|Thriller
6	7	Comedy|Romance

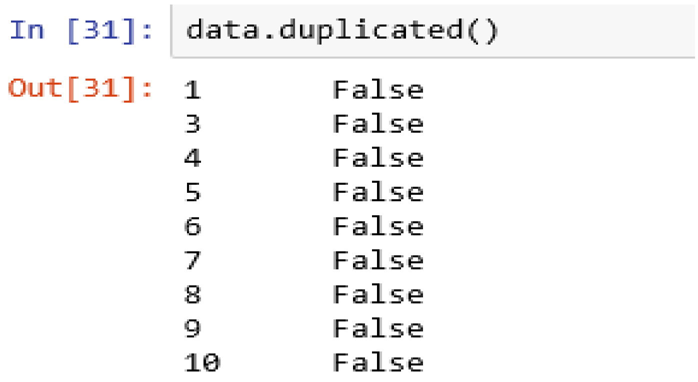
From above you can observe that

0 1 Adventure|Animation|Children|Comedy|Fantasy

Has been removed .

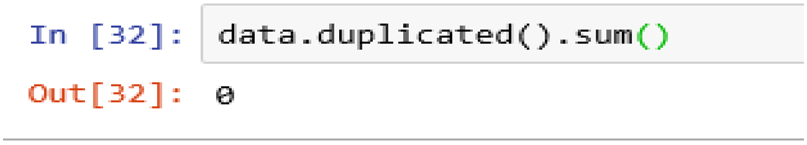
4. REMOVING DUPLICATED DATA

As we find that duplicated data brings confusion to the project, you should know ways of dropping the duplicated data, and this is the , data.duplicated() will return true if there are are duplicated data.



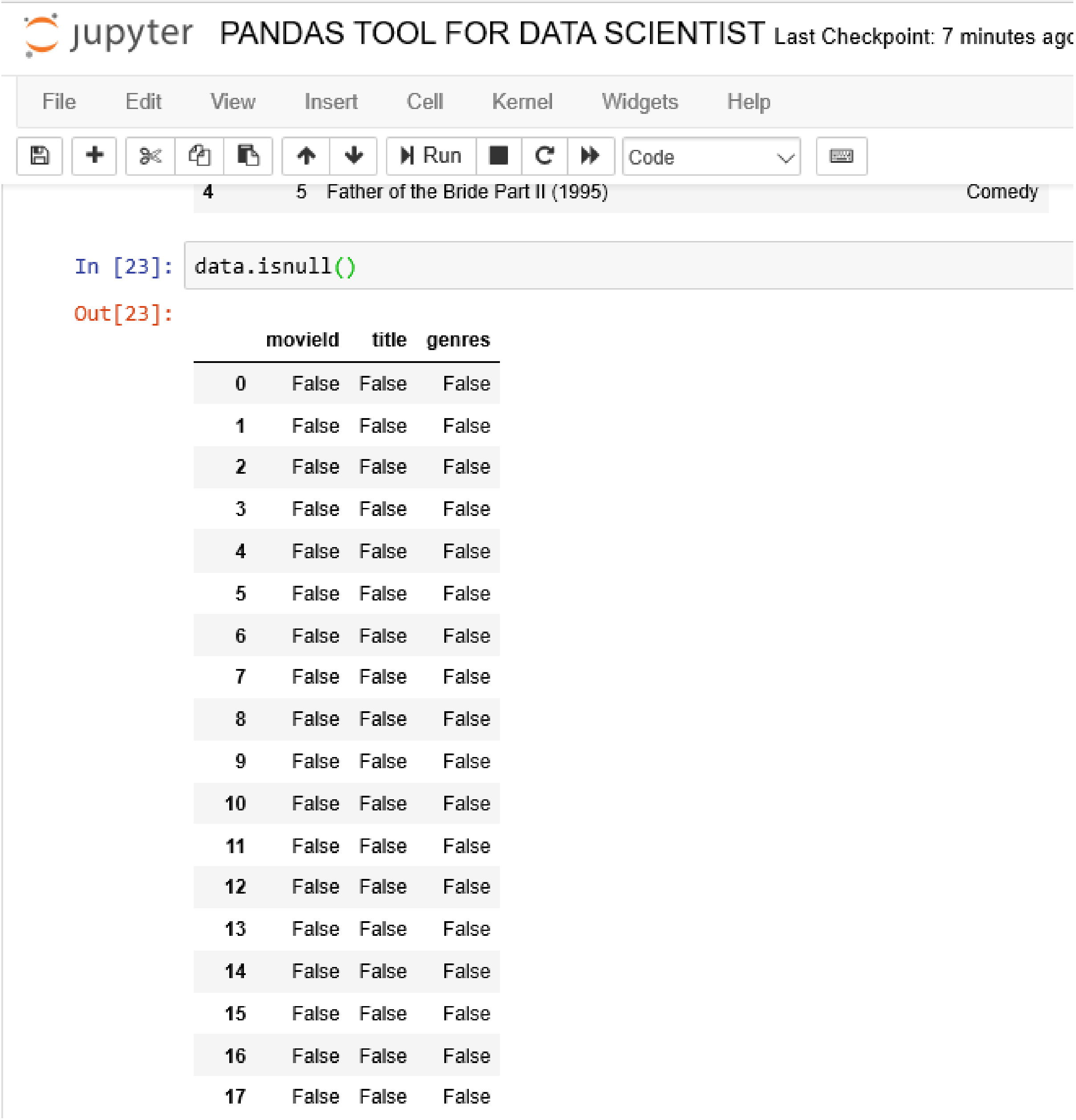
```
In [31]: data.duplicated()
Out[31]: 1        False
         3        False
         4        False
         5        False
         6        False
         7        False
         8        False
         9        False
         10       False
```

data.duplicated().sum(), this will result total number of duplicated rows, in our case you will zero as we don't have duplicated rows.



```
In [32]: data.duplicated().sum()
Out[32]: 0
```

data.duplicates(keep='first').shape, thi s function will delete all duplicated data while leaving first appered data , before those duplicated.

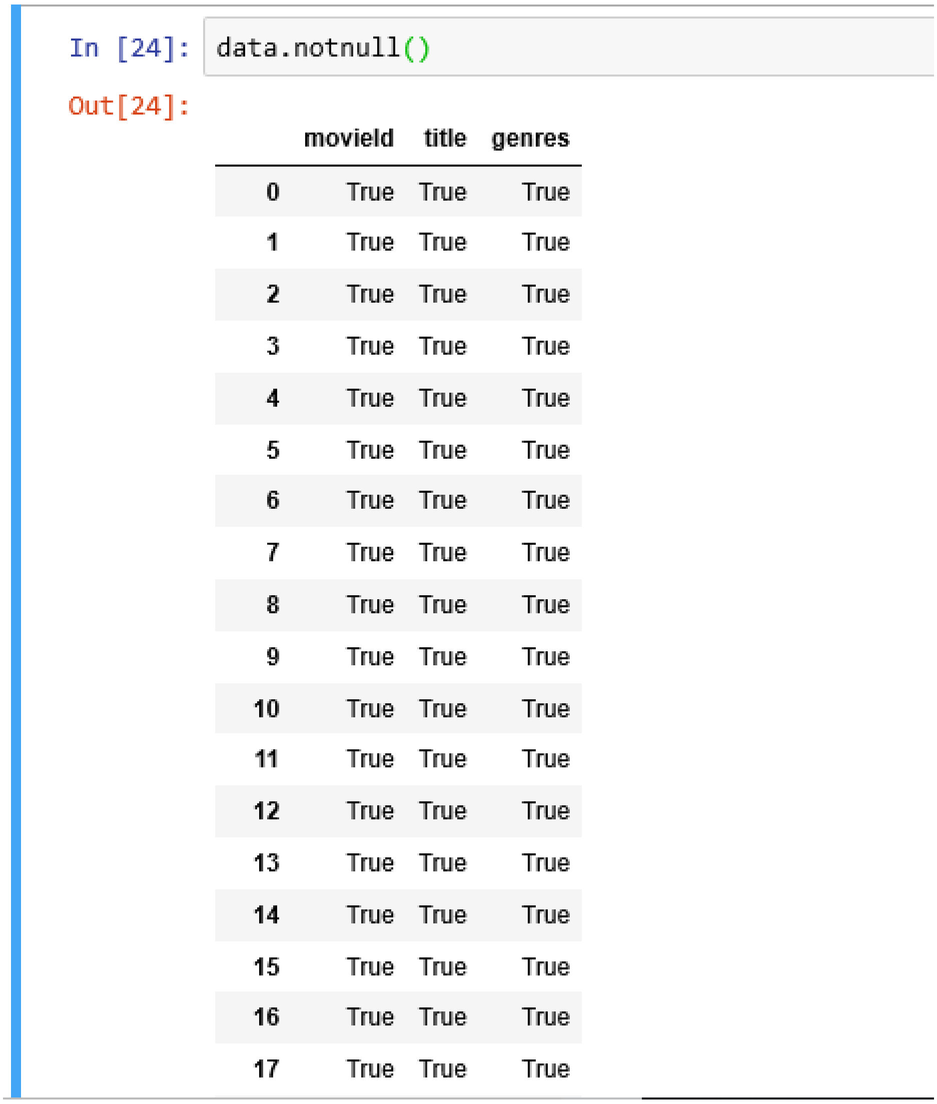

jupyter
PANDAS TOOL FOR DATA SCIENTIST Last Checkpoint: 7 minutes agc
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code

```
4          5   Father of the Bride Part II (1995)                               Comedy
```

```
In [23]: data.isnull()
```

```
Out[23]:
          movieId  title  genres
      0     False  False   False
      1     False  False   False
      2     False  False   False
      3     False  False   False
      4     False  False   False
      5     False  False   False
      6     False  False   False
      7     False  False   False
      8     False  False   False
      9     False  False   False
     10     False  False   False
     11     False  False   False
     12     False  False   False
     13     False  False   False
     14     False  False   False
     15     False  False   False
     16     False  False   False
     17     False  False   False
```

But another function you can use to check if is there missing data before looking , how to handle or fix the missing data is notnull(),if there is no missing data it will return true, but if there are missing data it will return false.



```
In [24]: data.notnull()
```

Out[24]:

	movieId	title	genres
0	True	True	True
1	True	True	True
2	True	True	True
3	True	True	True
4	True	True	True
5	True	True	True
6	True	True	True
7	True	True	True
8	True	True	True
9	True	True	True
10	True	True	True
11	True	True	True
12	True	True	True
13	True	True	True
14	True	True	True
15	True	True	True
16	True	True	True
17	True	True	True

As you have seen the effiency of these two functions isnull() and isnotnull() , works the same and provide.

CHAPTER SIX

ADVANCED WAYS OF DEALING WITH DATA

1.DEALING WITH STRING

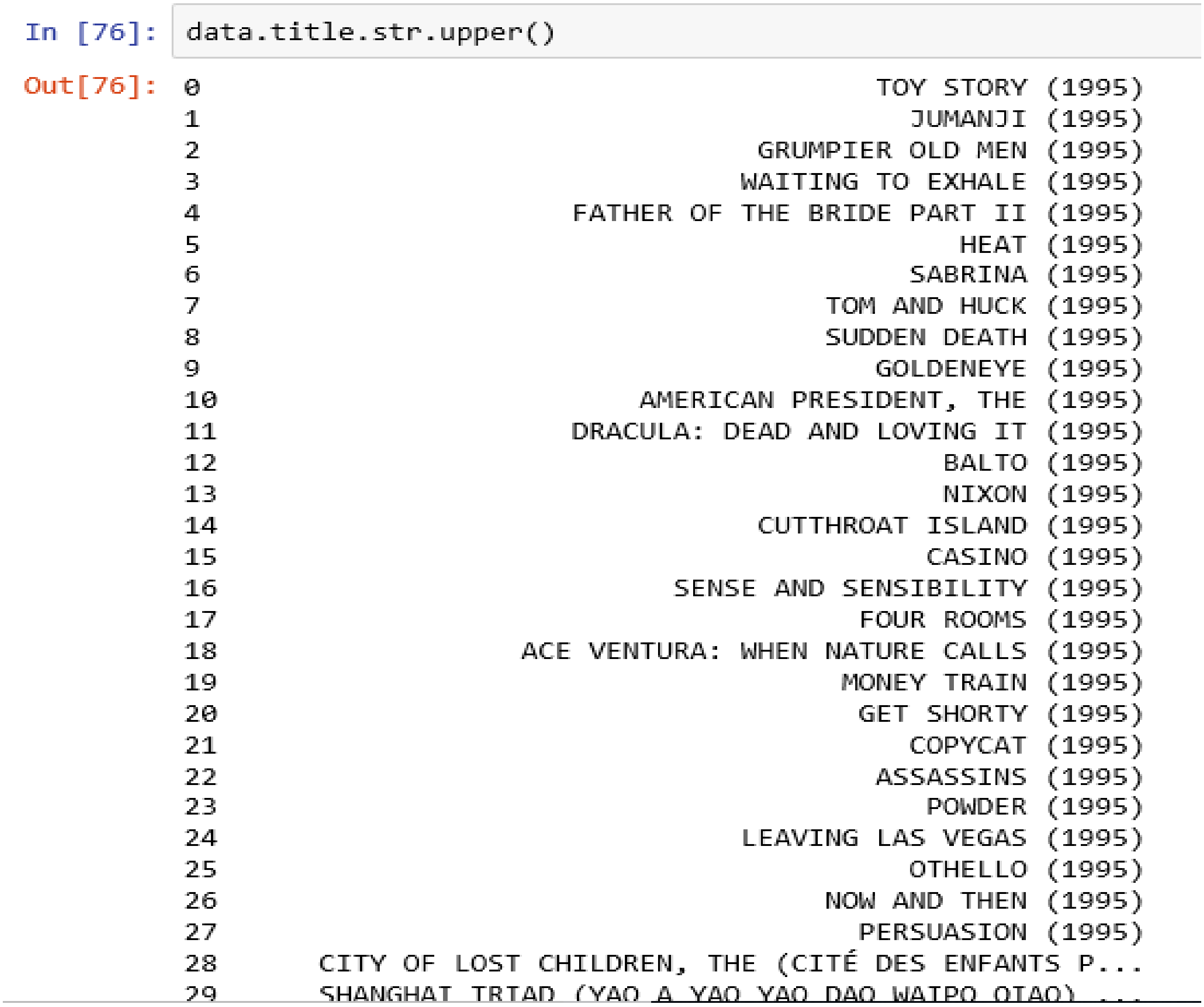
2.FILTERING COLUMN AND ROW

3.DATA SORTING

4.RENAMING OF COLUMN AND ROW

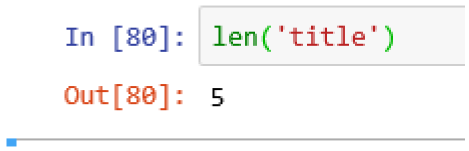
1.DEALING WITH STRING

UPPERCASE CONVERTION



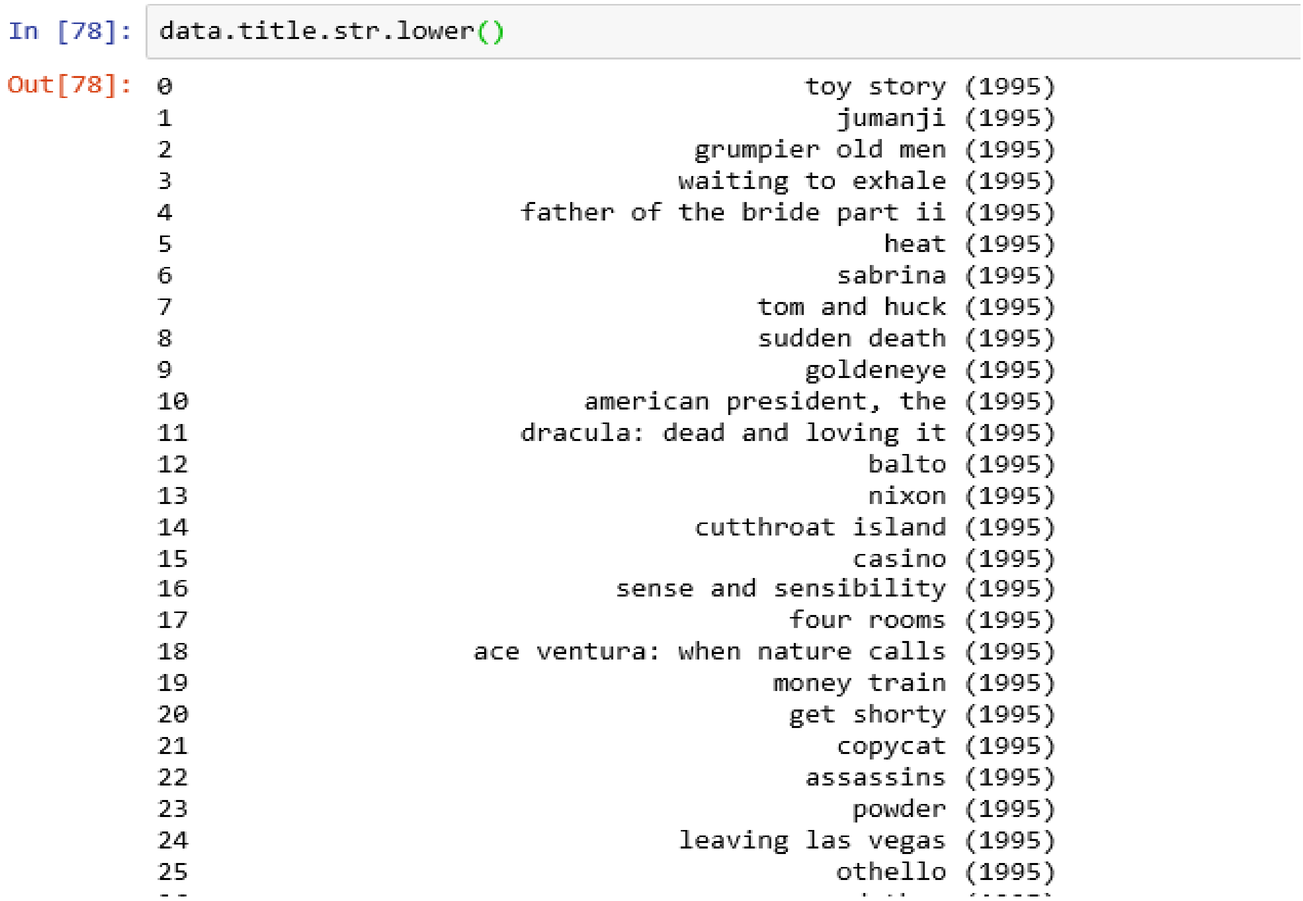
```
In [76]: data.title.str.upper()

Out[76]: 0                                                  TOY STORY (1995)
         1                                                    JUMANJI (1995)
         2                                           GRUMPIER OLD MEN (1995)
         3                                          WAITING TO EXHALE (1995)
         4                                FATHER OF THE BRIDE PART II (1995)
         5                                                       HEAT (1995)
         6                                                    SABRINA (1995)
         7                                               TOM AND HUCK (1995)
         8                                               SUDDEN DEATH (1995)
         9                                                  GOLDENEYE (1995)
         10                                   AMERICAN PRESIDENT, THE (1995)
         11                               DRACULA: DEAD AND LOVING IT (1995)
         12                                                     BALTO (1995)
         13                                                     NIXON (1995)
         14                                          CUTTHROAT ISLAND (1995)
         15                                                    CASINO (1995)
         16                                     SENSE AND SENSIBILITY (1995)
         17                                                FOUR ROOMS (1995)
         18                            ACE VENTURA: WHEN NATURE CALLS (1995)
         19                                               MONEY TRAIN (1995)
         20                                               GET SHORTY (1995)
         21                                                  COPYCAT (1995)
         22                                                ASSASSINS (1995)
         23                                                   POWDER (1995)
         24                                         LEAVING LAS VEGAS (1995)
         25                                                  OTHELLO (1995)
         26                                             NOW AND THEN (1995)
         27                                               PERSUASION (1995)
         28        CITY OF LOST CHILDREN, THE (CITÉ DES ENFANTS P...
         29        SHANGHAI TRIAD (YAO A YAO YAO DAO WAIPO QIAO) ...
```

STRING LENGH



```
In [80]: len('title')

Out[80]: 5
```

LOWERCASE CONVERTION



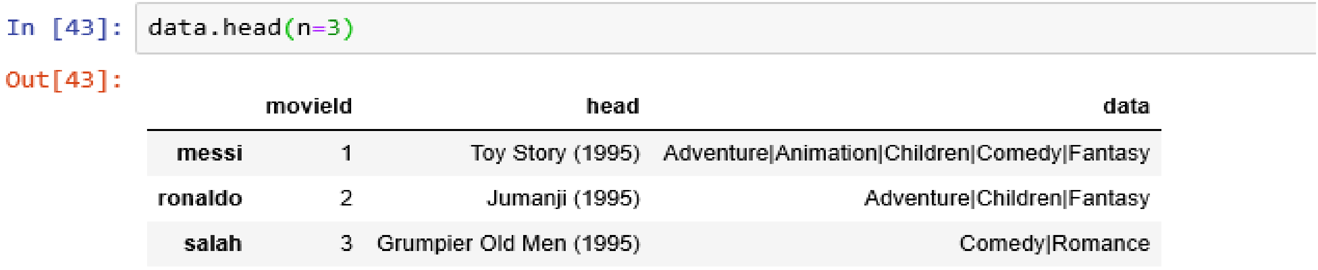
```
In [78]: data.title.str.lower()
```

```
Out[78]: 0                              toy story (1995)
         1                                jumanji (1995)
         2                       grumpier old men (1995)
         3                      waiting to exhale (1995)
         4            father of the bride part ii (1995)
         5                                   heat (1995)
         6                                sabrina (1995)
         7                           tom and huck (1995)
         8                           sudden death (1995)
         9                              goldeneye (1995)
         10               american president, the (1995)
         11           dracula: dead and loving it (1995)
         12                                 balto (1995)
         13                                 nixon (1995)
         14                      cutthroat island (1995)
         15                                casino (1995)
         16                 sense and sensibility (1995)
         17                            four rooms (1995)
         18        ace ventura: when nature calls (1995)
         19                           money train (1995)
         20                            get shorty (1995)
         21                               copycat (1995)
         22                             assassins (1995)
         23                                powder (1995)
         24                     leaving las vegas (1995)
         25                               othello (1995)
```

2.FILTERING COLUMN AND ROW

FILTERING ROW

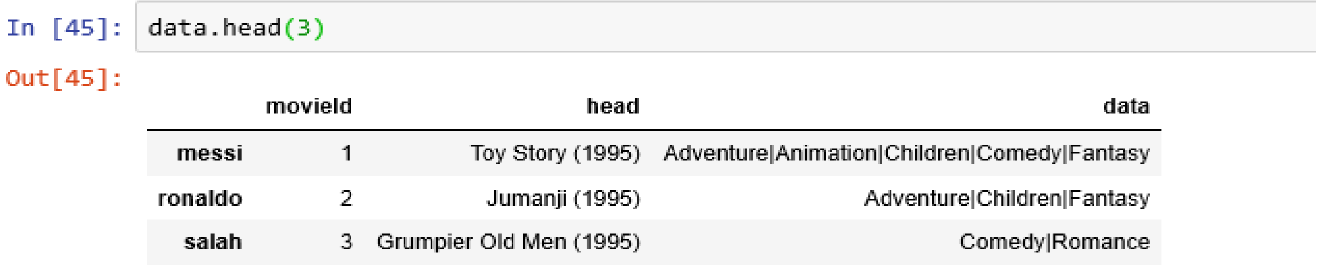
You can specify how many rows to view data.head() by default displays first five rows, but you can use head(n=3) to see first three rows or head(3) to view first three row.



```
In [43]: data.head(n=3)
```

Out[43]:

	movieId	head	data
messi	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
ronaldo	2	Jumanji (1995)	Adventure|Children|Fantasy
salah	3	Grumpier Old Men (1995)	Comedy|Romance



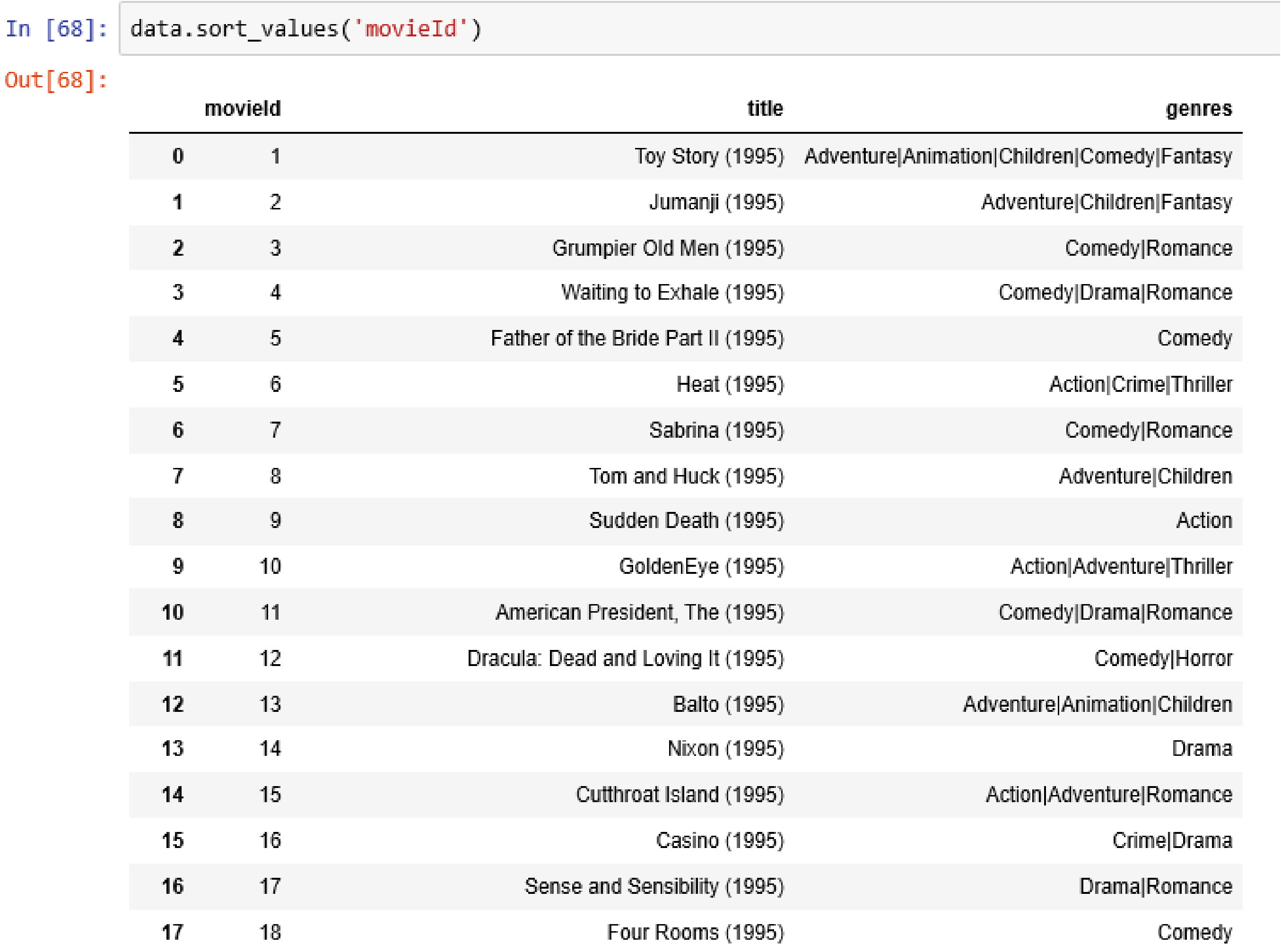
```
In [45]: data.head(3)
```

Out[45]:

	movieId	head	data
messi	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
ronaldo	2	Jumanji (1995)	Adventure|Children|Fantasy
salah	3	Grumpier Old Men (1995)	Comedy|Romance

3.DATA SORTING

Now we are going see how you cans sort you data in different ways.



In [68]:

```
data.sort_values('movieId')
```

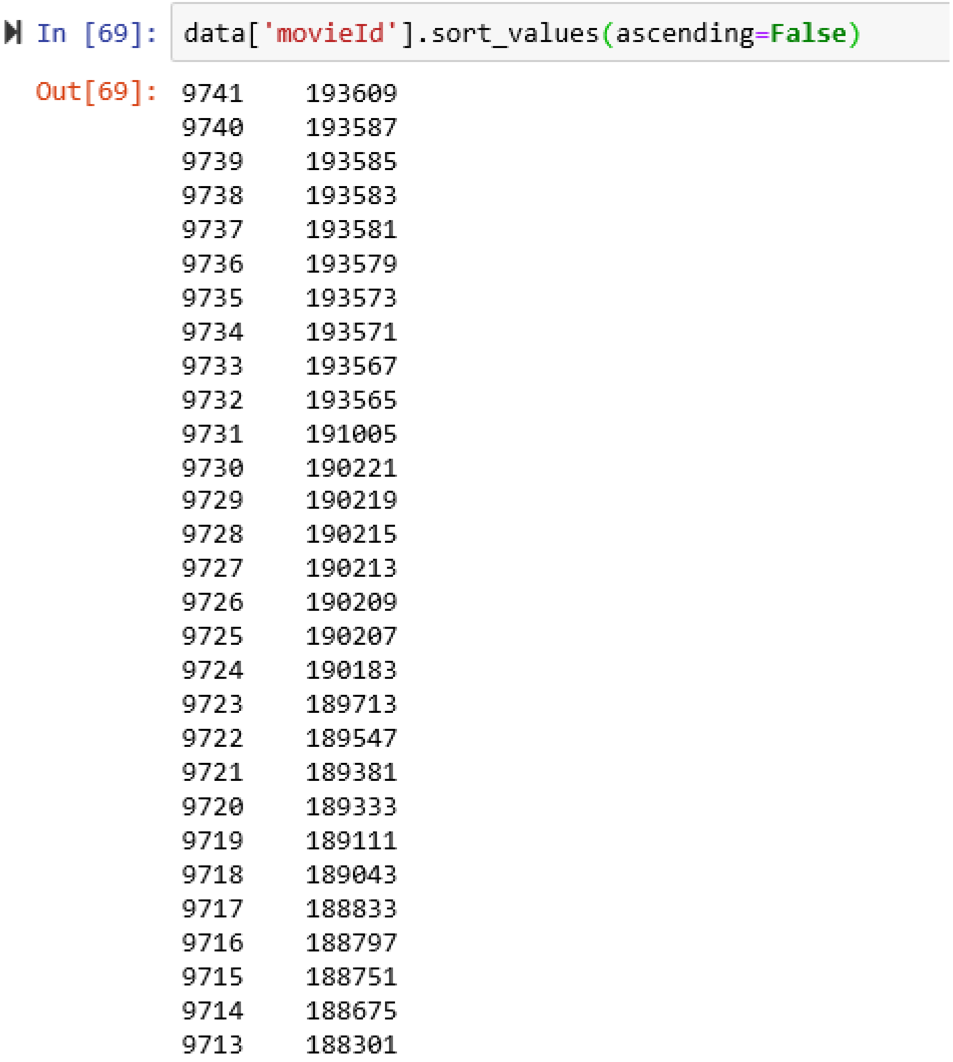
Out[68]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy
5	6	Heat (1995)	Action|Crime|Thriller
6	7	Sabrina (1995)	Comedy|Romance
7	8	Tom and Huck (1995)	Adventure|Children
8	9	Sudden Death (1995)	Action
9	10	GoldenEye (1995)	Action|Adventure|Thriller
10	11	American President, The (1995)	Comedy|Drama|Romance
11	12	Dracula: Dead and Loving It (1995)	Comedy|Horror
12	13	Balto (1995)	Adventure|Animation|Children
13	14	Nixon (1995)	Drama
14	15	Cutthroat Island (1995)	Action|Adventure|Romance
15	16	Casino (1995)	Crime|Drama
16	17	Sense and Sensibility (1995)	Drama|Romance
17	18	Four Rooms (1995)	Comedy

As you can see from above that I have used sort_values() function by sorting movieId , and as you have seen it sort from descending to ascending order, but also you can sort you data according to duration, if your data set have years or number like attributes.

SORTING BY DESCENDING ORDER

attributes. When sorting you have many options, you can sorting by descending order and this is what is done right here, the function being used is sort_values(),but what you need is to specify either ascending or descending order.



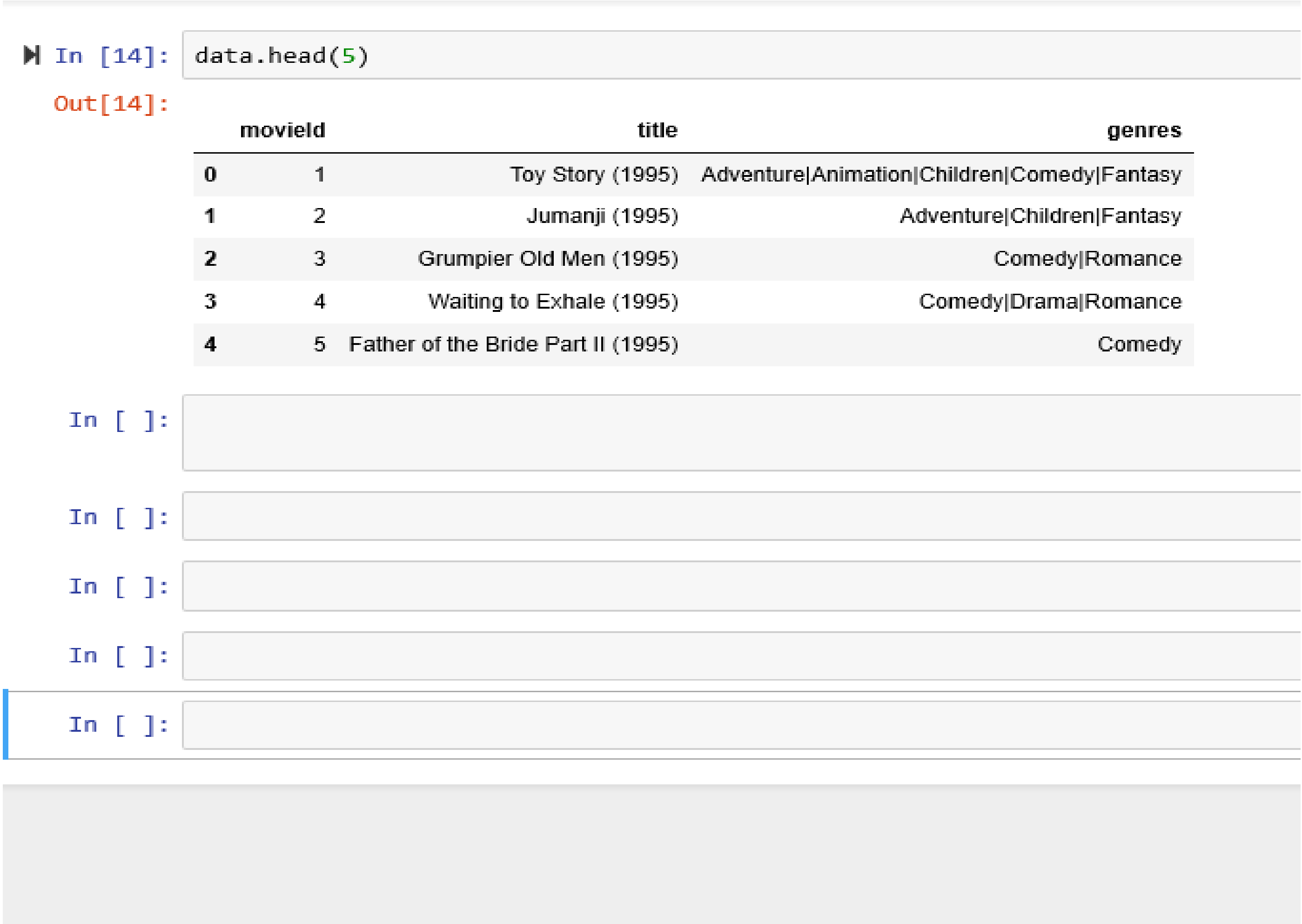
```
In [69]: data['movieId'].sort_values(ascending=False)

Out[69]: 9741    193609
         9740    193587
         9739    193585
         9738    193583
         9737    193581
         9736    193579
         9735    193573
         9734    193571
         9733    193567
         9732    193565
         9731    191005
         9730    190221
         9729    190219
         9728    190215
         9727    190213
         9726    190209
         9725    190207
         9724    190183
         9723    189713
         9722    189547
         9721    189381
         9720    189333
         9719    189111
         9718    189043
         9717    188833
         9716    188797
         9715    188751
         9714    188675
         9713    188301
```

4. RENAMING OF COLUMN AD ROW

But another, here we are going to see how to rename both column and rows, it may happen that you want to change the names of you attributes, and it is when this knowledge is needed, or it may happen that you gave wrong name to the column or row, not only those reason but also for increasing your data flexibility, and increasing the ability to play with your data.

Let’s see how you can rename single column.

Before renaming of single column



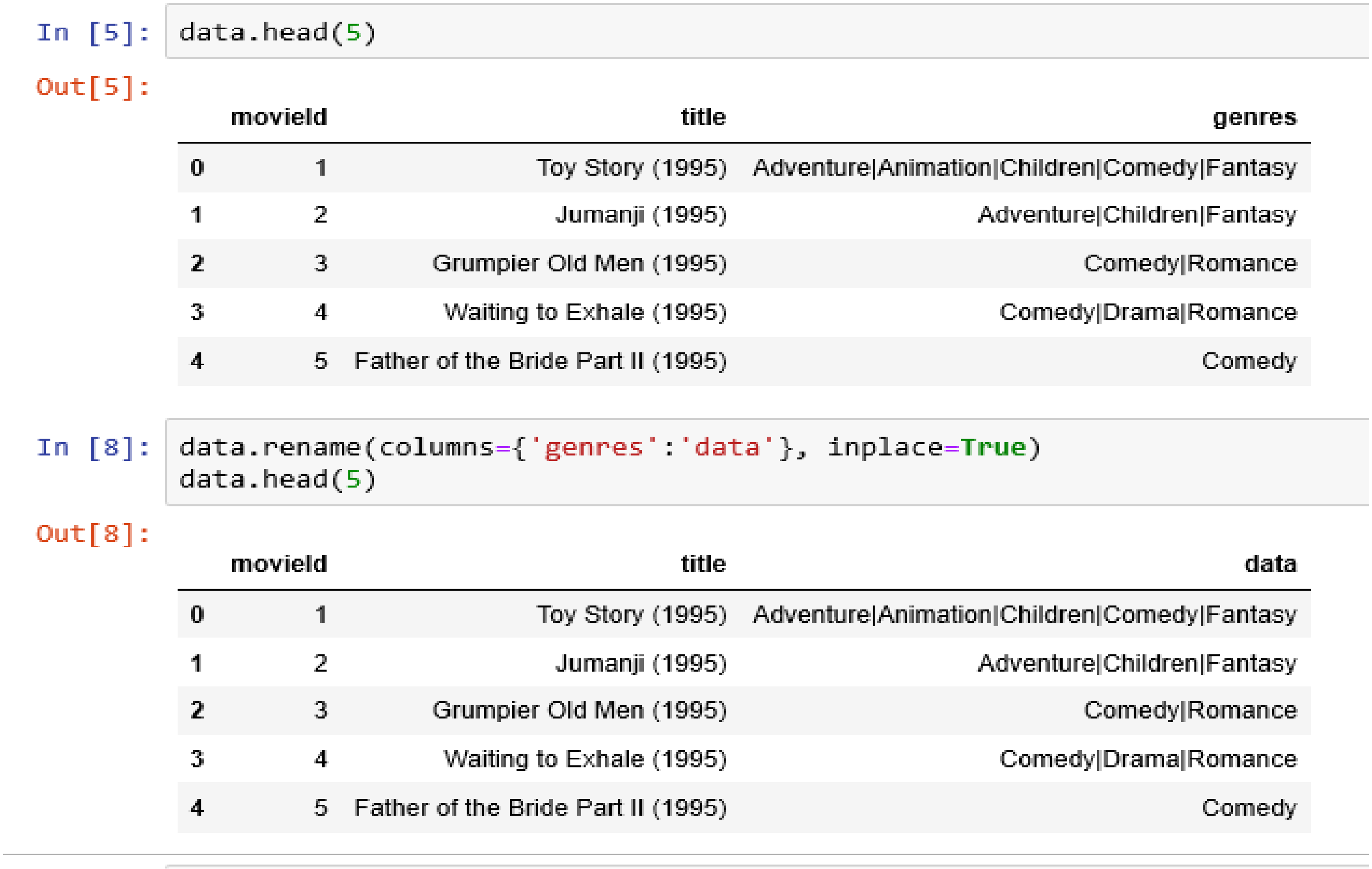
```
In [14]: data.head(5)
Out[14]:
```

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

```
In [ ]:

In [ ]:

In [ ]:

In [ ]:

In [ ]:
```

As you can see our last column is called genres, so am going to rename it from genres to data, so the last column should be read as data and not genres.

After renaming of single column



```
In [5]: data.head(5)
```

Out[5]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

```
In [8]: data.rename(columns={'genres':'data'}, inplace=True)
        data.head(5)
```

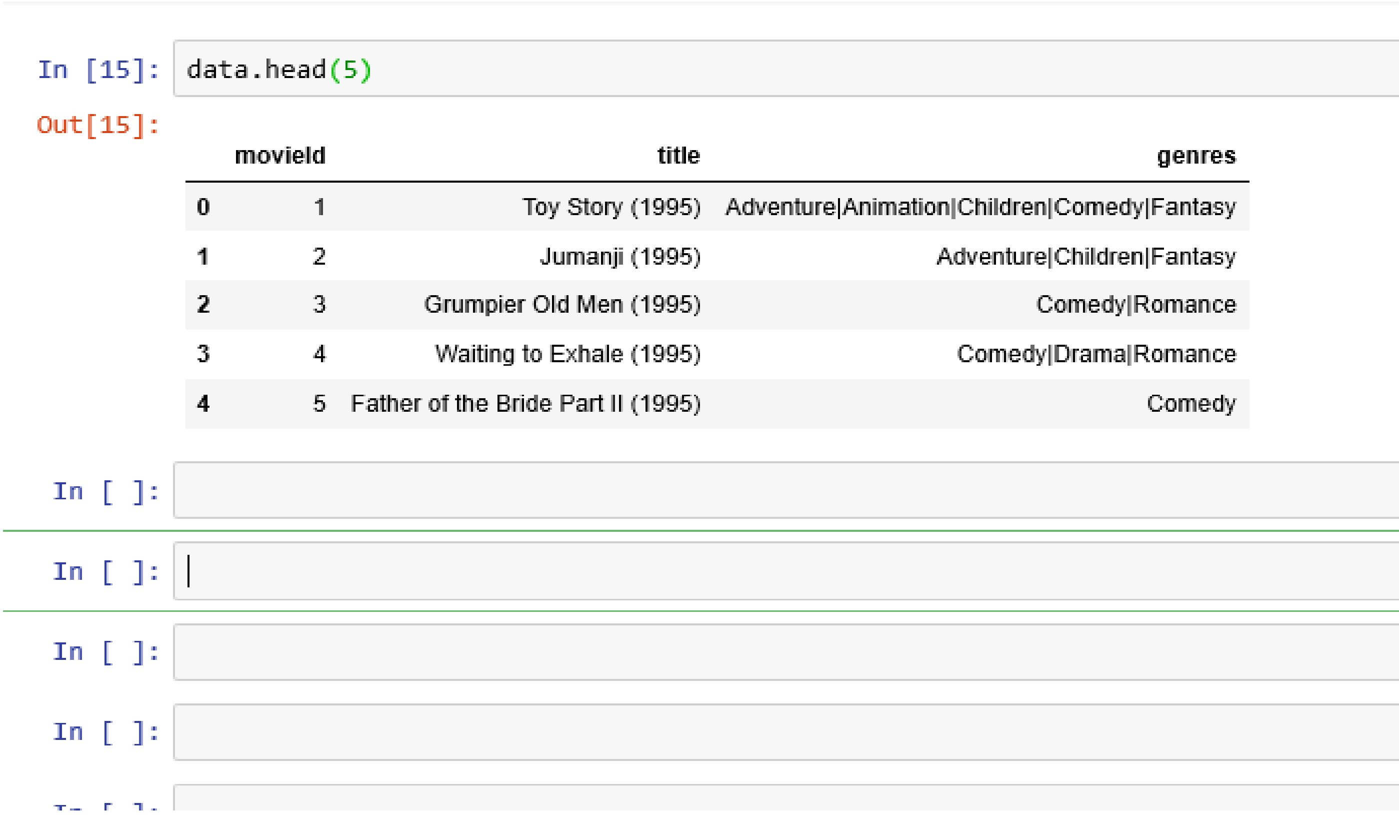
Out[8]:

	movieId	title	data
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

As you can see from above, you may have noticed that last column is already changed from genres to data, rename() is the function which is responsible for renaming, columns specify that what is being renamed is column, genres is the name of old column , while data is the name of our new column in dataframe.

NOTE : inplace=False by default, so when renaming , make sure you don’t forget inplace=True

Before renaming of more than one column



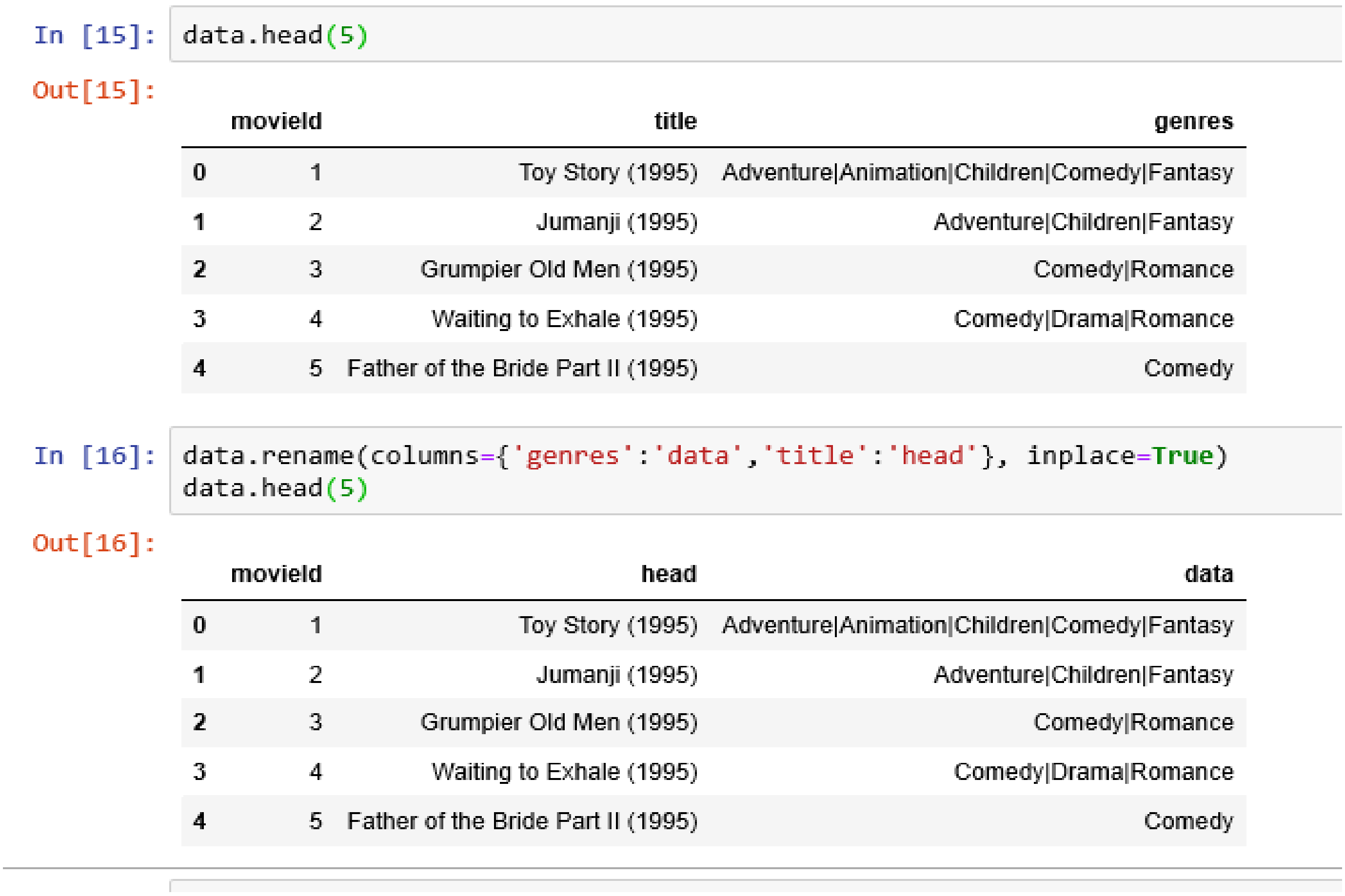
```
In [15]: data.head(5)
Out[15]:
```

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

```
In [ ]:

In [ ]:

In [ ]:

In [ ]:
```

From above picture you can see column two and column three are title and genres but here we suppose to rename them column title to head, and column genres to data , the procedures are the same genres as when we did single column renaming , it you will get trouble about this two parts of single and more than one column renaming, go back to python and get the review of python lists and dictionary, hopefully after that everything will be made easy for you as you know data science is the simplest field to enjoy your skills.

After renaming of more than one column



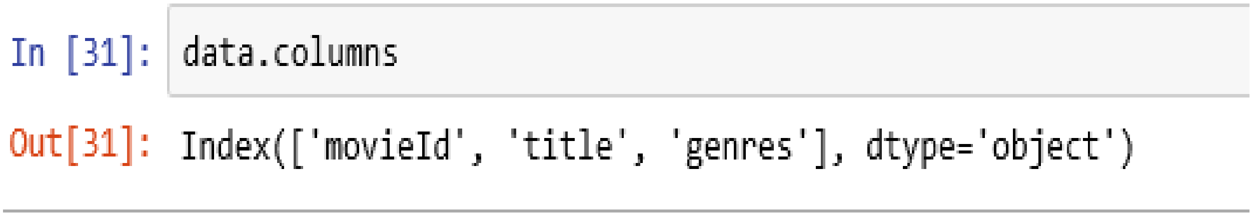
```
In [15]: data.head(5)
Out[15]:
```

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

```
In [16]: data.rename(columns={'genres':'data','title':'head'}, inplace=True)
         data.head(5)
Out[16]:
```

	movieId	head	data
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

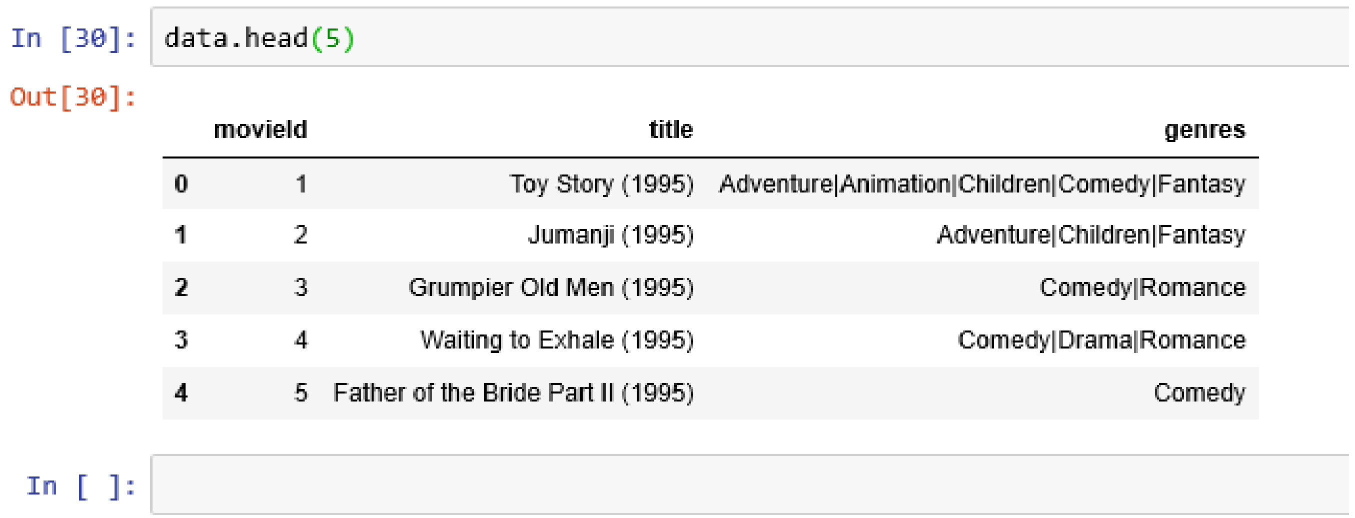
Additional thing is that, if you want to see only columns you have, you can issue this code of line, data.columns, then you will your columns.



```
In [31]: data.columns
Out[31]: Index(['movieId', 'title', 'genres'], dtype='object')
```

Before renaming of single row

Below is how our dataframe looks like before renaming of any single row.



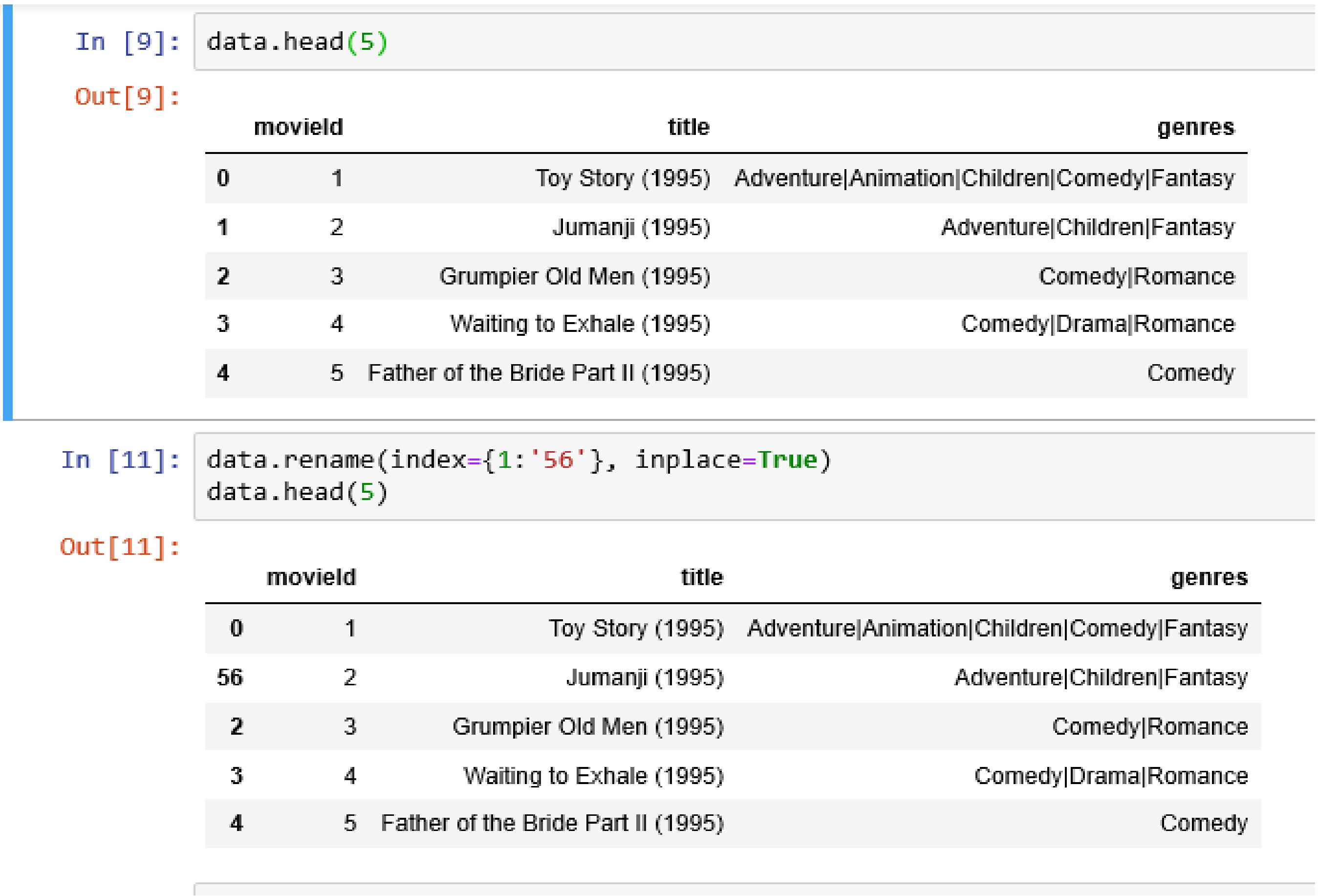
```
In [30]: data.head(5)
Out[30]:
```

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

```
In [ ]:
```

After renaming of single row

Below is how our dataframe looks like after renaming of single row, if you’re carefully you might have notice the changes which have been made.



```
In [9]: data.head(5)
```

Out[9]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

```
In [11]: data.rename(index={1:'56'}, inplace=True)
         data.head(5)
```

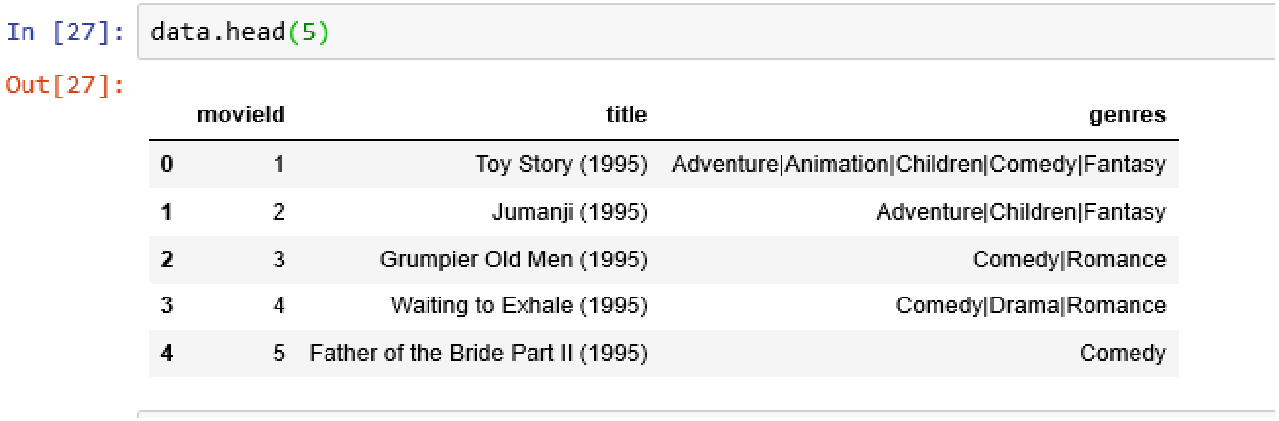
Out[11]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
56	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

As you have seen that the second row or index number have been changed from 1 to 56 , that is how it works, 1:'56', 1 means the index to be replaced and 56 is the index to replace.

Before renaming of single row

In previous cell I showed you how you can rename only one row , now am going to show you how it is possible to rename more than one row.

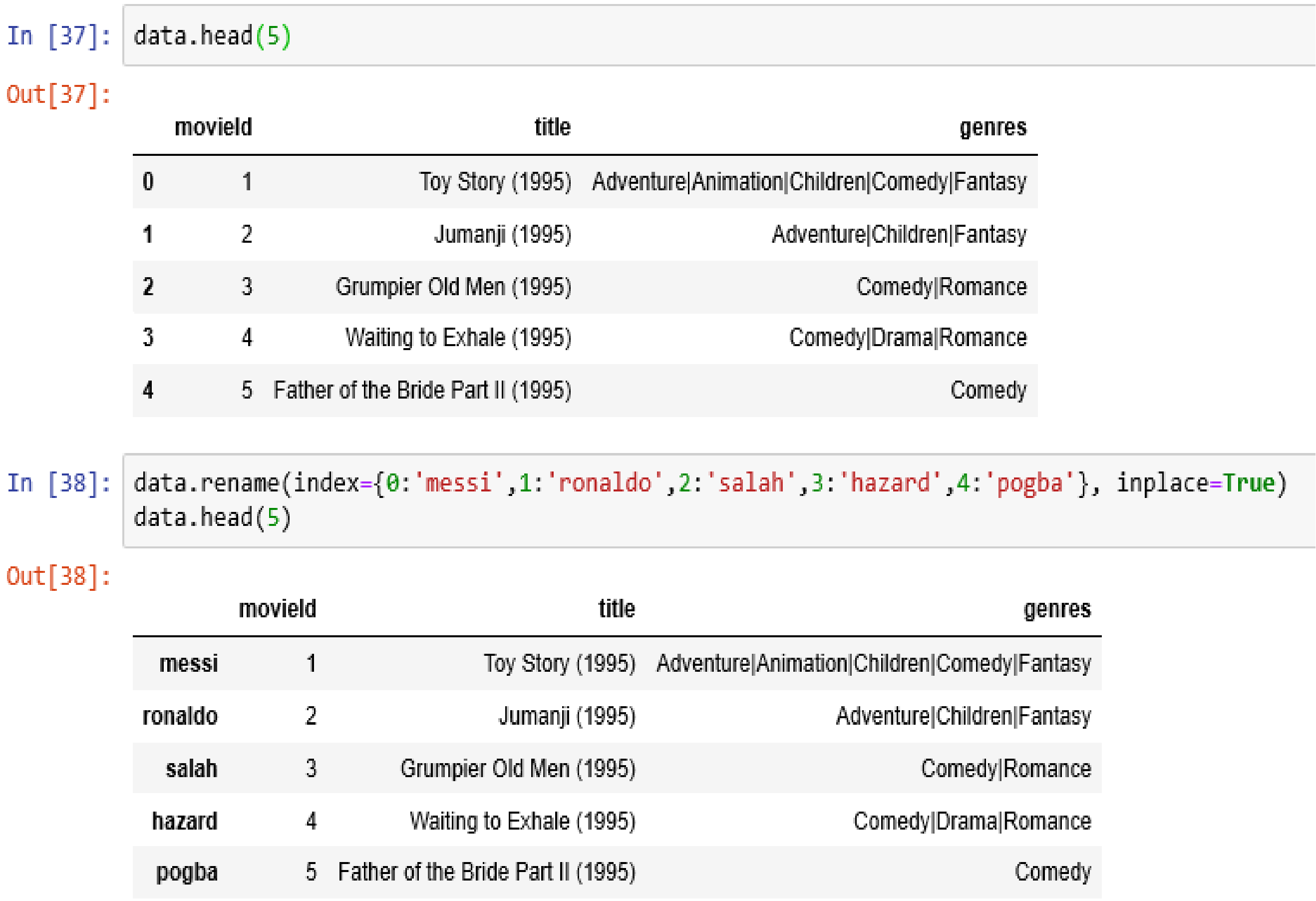


```
In [27]: data.head(5)
```

Out[27]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

After renaming of single row



```
In [37]: data.head(5)
```

Out[37]:

	movieId	title	genres
0	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
1	2	Jumanji (1995)	Adventure|Children|Fantasy
2	3	Grumpier Old Men (1995)	Comedy|Romance
3	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
4	5	Father of the Bride Part II (1995)	Comedy

```
In [38]: data.rename(index={0:'messi',1:'ronaldo',2:'salah',3:'hazard',4:'pogba'}, inplace=True)
         data.head(5)
```

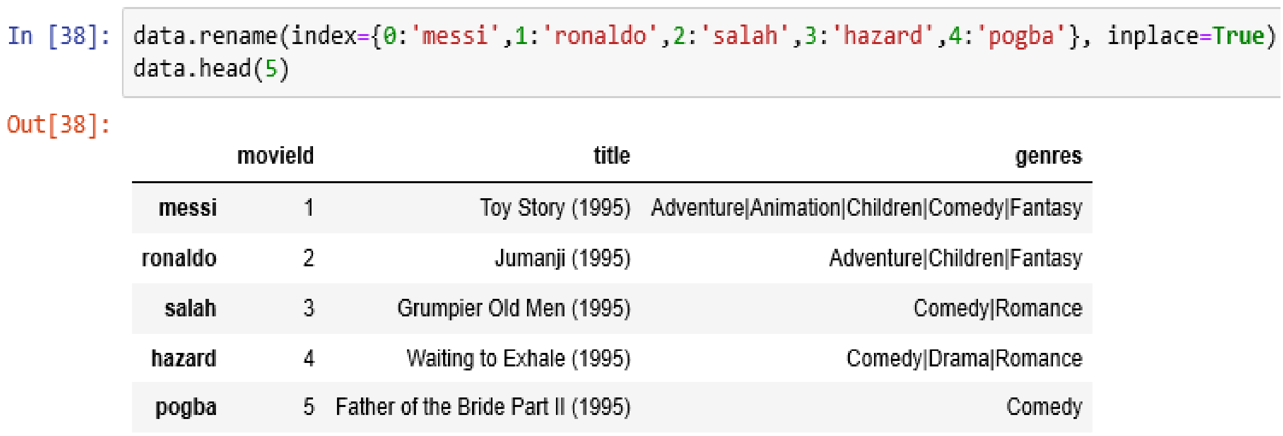
Out[38]:

	movieId	title	genres
messi	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
ronaldo	2	Jumanji (1995)	Adventure|Children|Fantasy
salah	3	Grumpier Old Men (1995)	Comedy|Romance
hazard	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
pogba	5	Father of the Bride Part II (1995)	Comedy

The procedure to do it are the same, what is needed is just taking your time. I believe in you, you can do it the same.

Renaming of row with it's corresponding column

Before

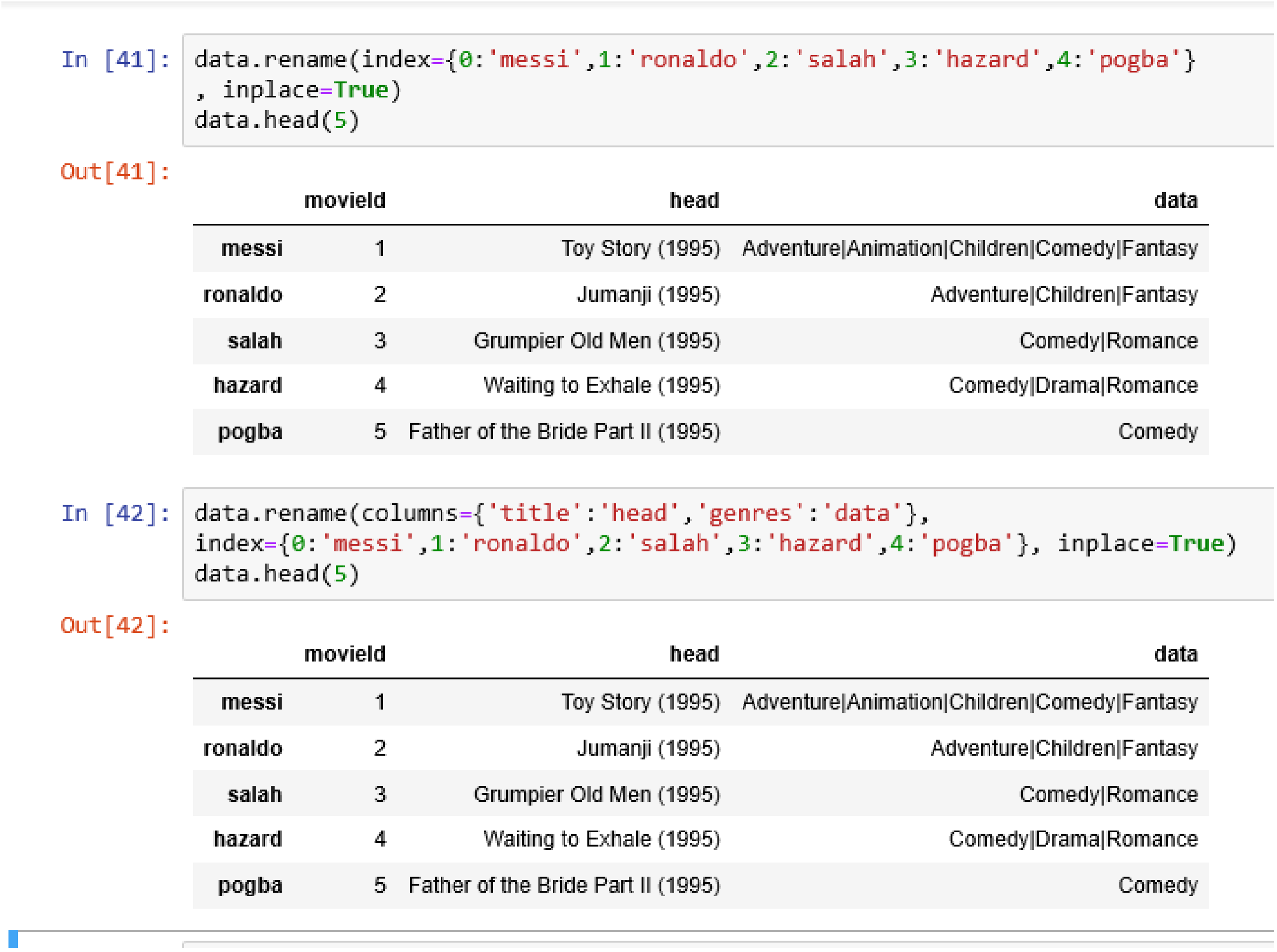


```
In [38]: data.rename(index={0:'messi',1:'ronaldo',2:'salah',3:'hazard',4:'pogba'}, inplace=True)
         data.head(5)
```

Out[38]:

	movieId	title	genres
messi	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
ronaldo	2	Jumanji (1995)	Adventure|Children|Fantasy
salah	3	Grumpier Old Men (1995)	Comedy|Romance
hazard	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
pogba	5	Father of the Bride Part II (1995)	Comedy

After



```
In [41]: data.rename(index={0:'messi',1:'ronaldo',2:'salah',3:'hazard',4:'pogba'}
         , inplace=True)
         data.head(5)
```

Out[41]:

	movieId	head	data
messi	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
ronaldo	2	Jumanji (1995)	Adventure|Children|Fantasy
salah	3	Grumpier Old Men (1995)	Comedy|Romance
hazard	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
pogba	5	Father of the Bride Part II (1995)	Comedy

```
In [42]: data.rename(columns={'title':'head','genres':'data'},
         index={0:'messi',1:'ronaldo',2:'salah',3:'hazard',4:'pogba'}, inplace=True)
         data.head(5)
```

Out[42]:

	movieId	head	data
messi	1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
ronaldo	2	Jumanji (1995)	Adventure|Children|Fantasy
salah	3	Grumpier Old Men (1995)	Comedy|Romance
hazard	4	Waiting to Exhale (1995)	Comedy|Drama|Romance
pogba	5	Father of the Bride Part II (1995)	Comedy

CHAPTER SEVEN

STASTICAL ANALYSIS

1.CORRELATION OF DATA

2.MEAN

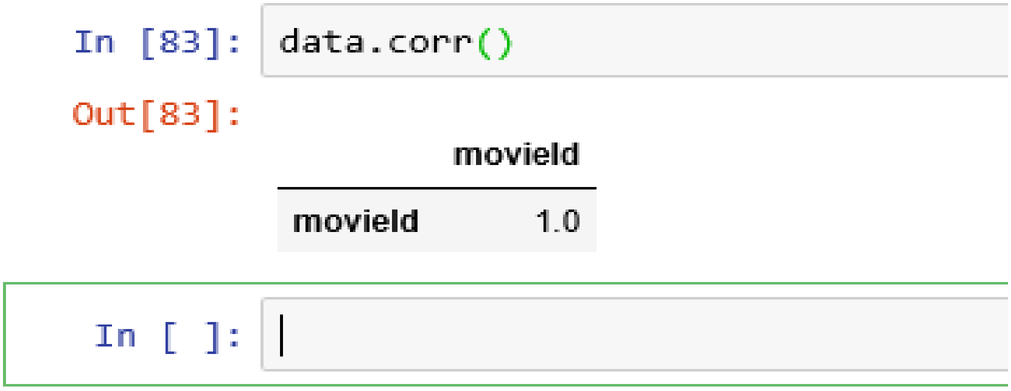
3.MEDIUM

4. MAXIMUM DATA

5. MINUMUM DATA

1.CORRELATION OF DATA

this function is used to find the mean of your attributes,It is also important to know how does your data set correlate, because it becomes easly to work with data which have high positive correlation, unlike working with data with negative correlation, and this must be considered when doing machine learning with algorithm such as logistic regression as it is said to be statistical based.

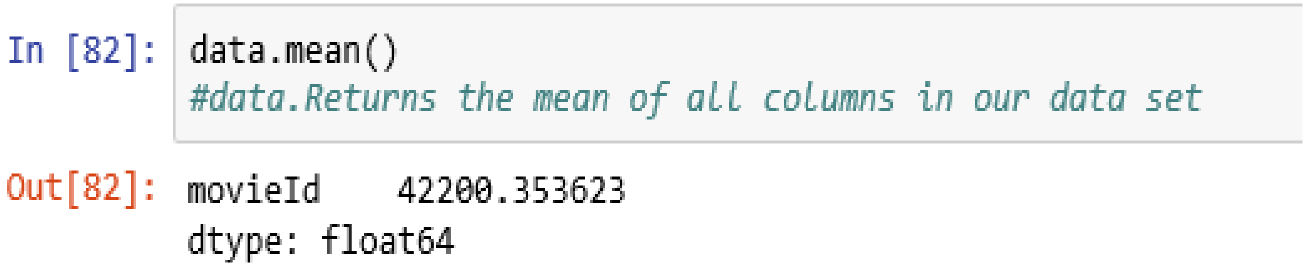


```
In [83]: data.corr()
Out[83]:
                    movieId
         movieId        1.0

In [ ]:
```

You can that moveild has correlation of 1.0 which is said to be the highest correlation, no correlation exceed that , but if you will have started working with machine learning remember to do data transformation first , for example transforming subject to interger , then it is where you will get correlation of all attributes of your dataframe.

2.MEAN

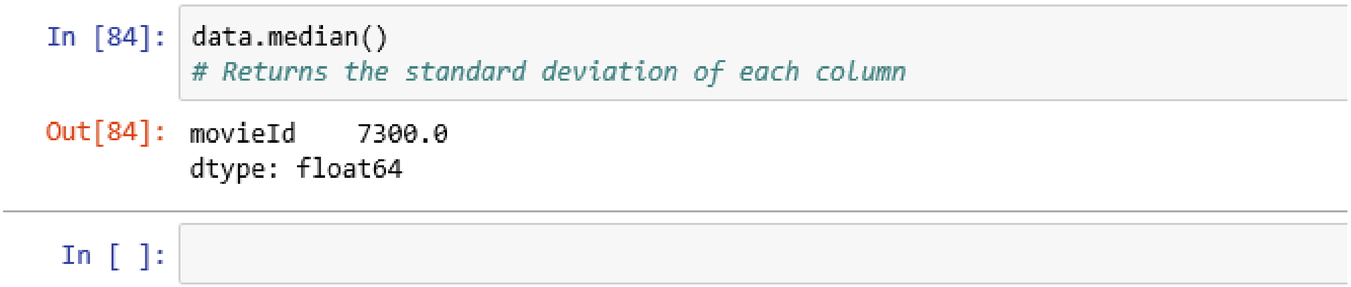
data.mean(),this function is used to find the mean of your attributes, and why knowing mean is very important for data scientist, because it may happen your age column has few missing values , so use can use the mean value you get to fill in those space with missing values, but this is mostly used for numerical missing values , and not string.



```
In [82]: data.mean()
         #data.Returns the mean of all columns in our data set

Out[82]: movieId    42200.353623
         dtype: float64
```

3.MEDIUM

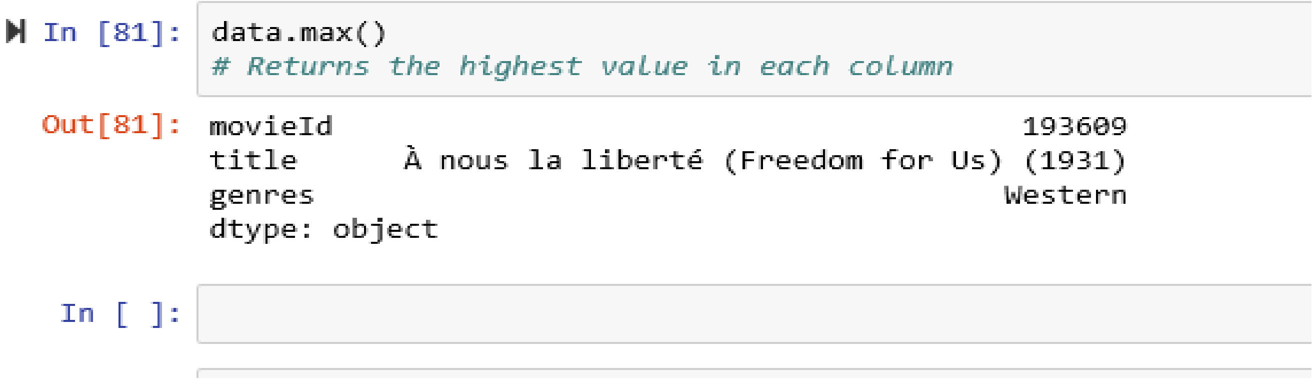
Also median is one of the important thing to look, by using meadian you can know the middle age of value in your data set which can be helpful I more data analysis.



```
In [84]: data.median()
         # Returns the standard deviation of each column

Out[84]: movieId    7300.0
         dtype: float64

In [ ]:
```

4. MAXIMUM DATA

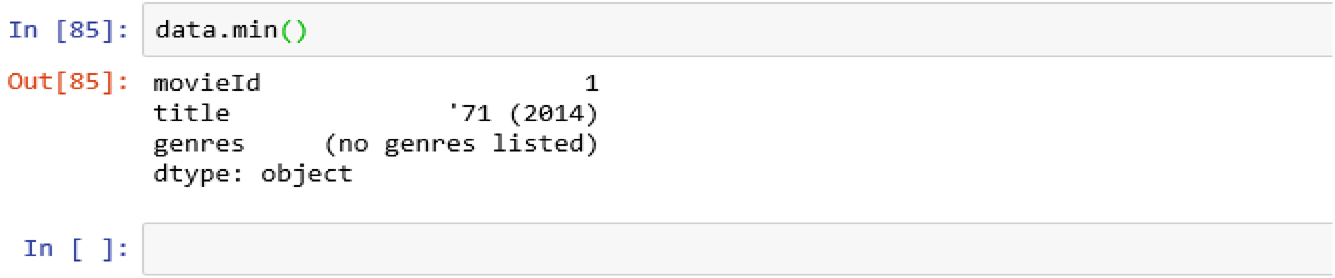
data.max(), this function will always return the highest data or value in your dataframe, this is one one the important part in the statistical analysis of you're data.



```
In [81]: data.max()
         # Returns the highest value in each column

Out[81]: movieId                                          193609
         title      À nous la liberté (Freedom for Us) (1931)
         genres                                          Western
         dtype: object

In [ ]:
```

5.MINUMUM DATA

data.min(),this will return the minimum data or value in your data set.



```
In [85]: data.min()

Out[85]: movieId                    1
         title           '71 (2014)
         genres     (no genres listed)
         dtype: object

In [ ]:
```

CHAPTER EIGHT

DATA VISUALIZATION AND EXPLORATION

1.SCATTER PLOT

2.HISTOGRAM

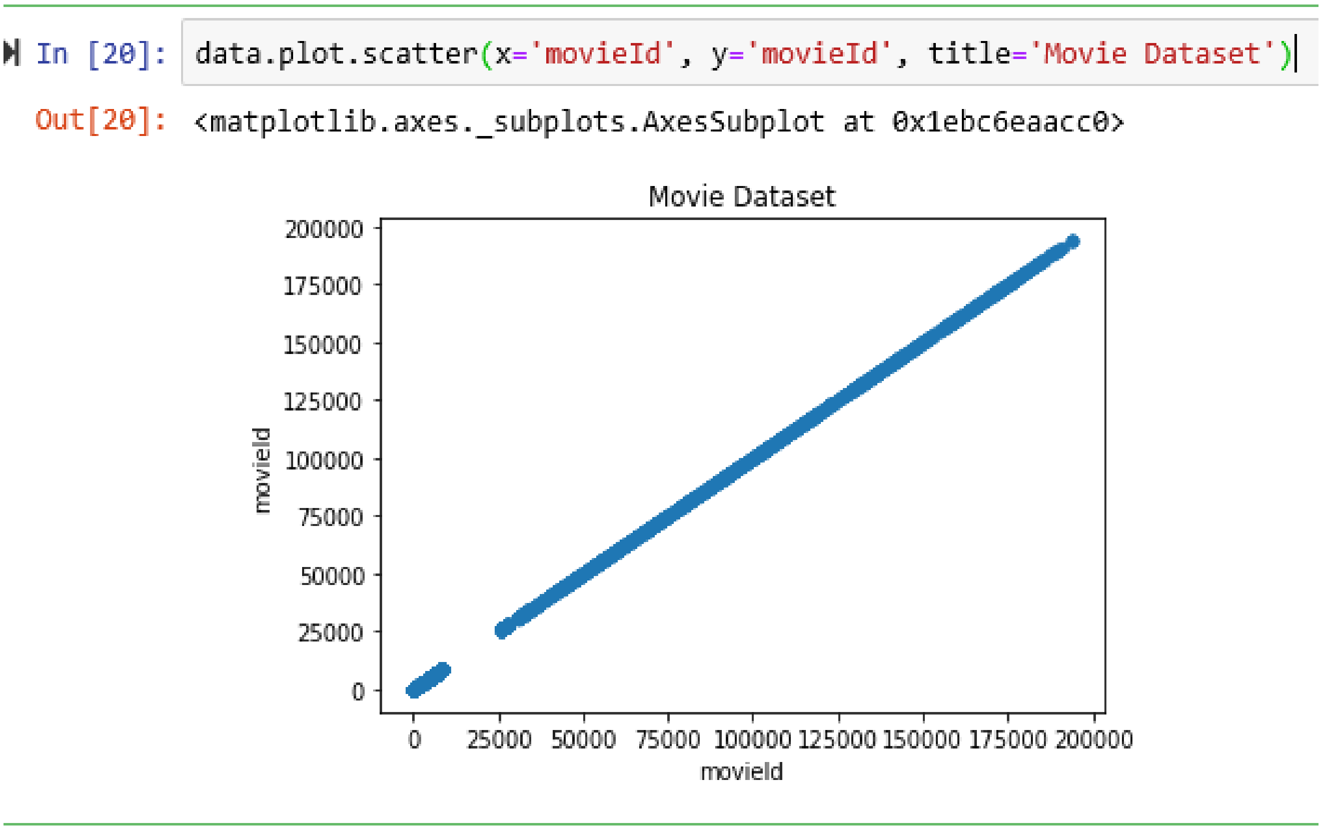
3.BAR CHART

4. LINE CHART

1.SCATTER PLOT

To create a scatter plot in pandas is pretty easy plot.scatter() method can easily plot scatter plot and this method take two arguments, x-column and y-column

data.plot.scatter(x='movieId', y='movieId', title='Movie Dataset')



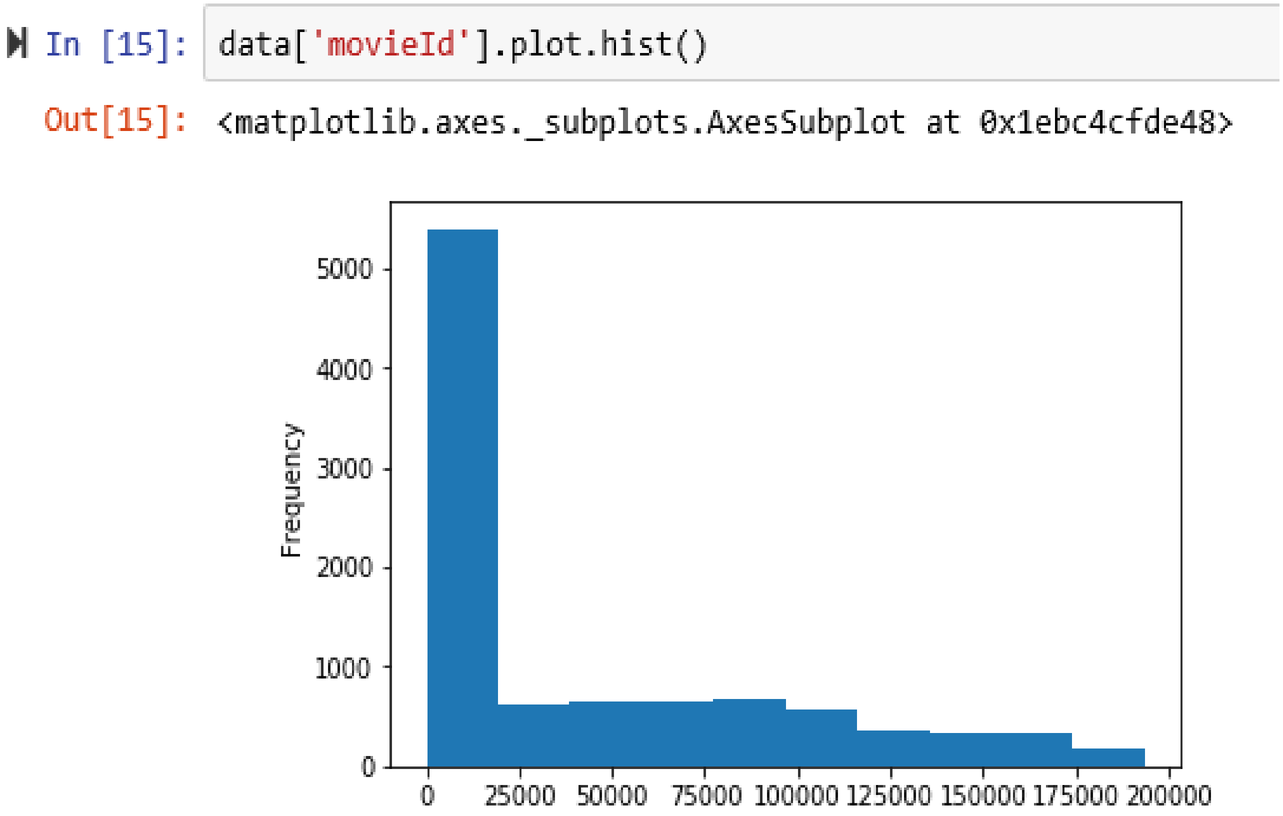
```
In [20]: data.plot.scatter(x='movieId', y='movieId', title='Movie Dataset')
Out[20]: <matplotlib.axes._subplots.AxesSubplot at 0x1ebc6eaacc0>
```

y-column , as you can see I have drawn scatter plot of movieId against movieId , but this is not really, I have done so because movieId is the only numeric data in my data set, for undersanding tell simple example I have data science where I trained data set by using different machine learning algorithm, and the issue was to predict whether a patient taking diagnosis has breast cancer or not , so patient with breast cancer=1 and those without brest cancer=0 ,so for such or similar scenario it is easy to have scatter plot of x and y.

2.HISTOGRAM

in pandas we may create histogram with plot.hist method , in this method you pass the name of colum you want to plot as an argument

data['movieId'].plot.hist(), as you can see that by attribute or column I want to plot is movieId



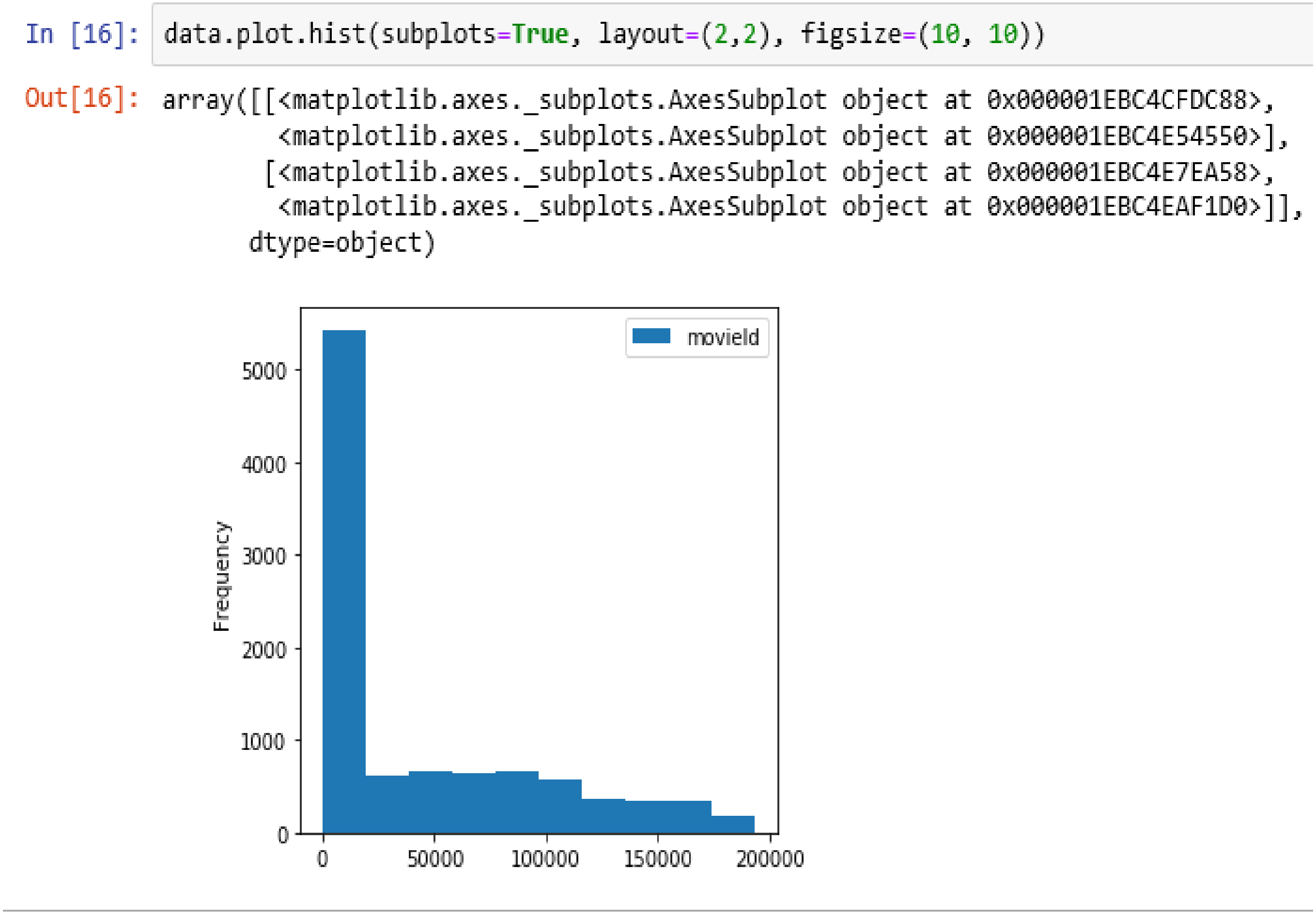
```
In [15]: data['movieId'].plot.hist()
```

```
Out[15]: <matplotlib.axes._subplots.AxesSubplot at 0x1ebc4cfde48>
```

but also you can create multiple histogram it will look like this

data.plot.hist(subplots=True, layout=(2,2), figsize=(10, 10))

the subplots argument specifies that we want a separate plot for each feature and the layout specifies the number per row and column, but for my case you case just single plot of movieId as it is only one numeric data type.



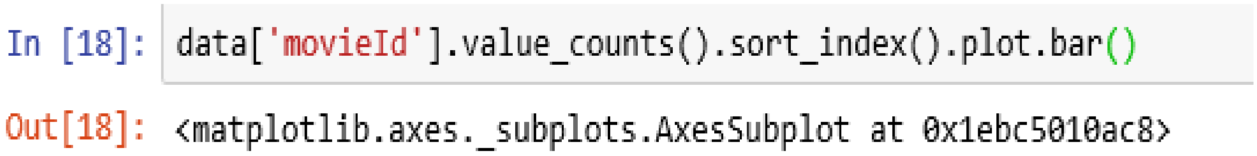
In [16]:
```
data.plot.hist(subplots=True, layout=(2,2), figsize=(10, 10))
```

Out[16]:
```
array([[<matplotlib.axes._subplots.AxesSubplot object at 0x000001EBC4CFDC88>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001EBC4E54550>],
       [<matplotlib.axes._subplots.AxesSubplot object at 0x000001EBC4E7EA58>,
        <matplotlib.axes._subplots.AxesSubplot object at 0x000001EBC4EAF1D0>]],
      dtype=object)
```

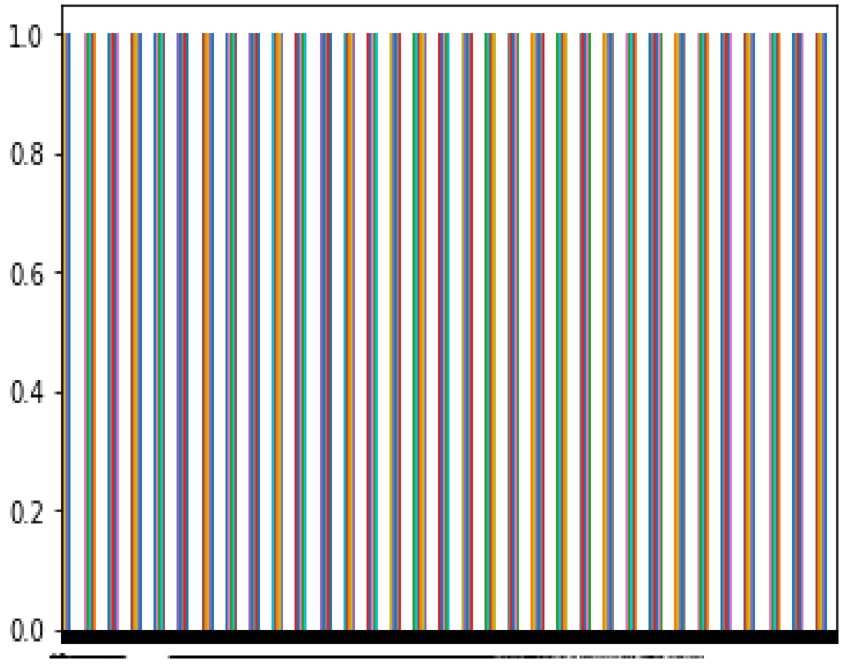
3.BAR CHART

It is easy to plot bar in pandas, we can use plot.bar() method

data['movieId'].value_counts().sort_index().plot.bar()



```
In [18]: data['movieId'].value_counts().sort_index().plot.bar()

Out[18]: <matplotlib.axes._subplots.AxesSubplot at 0x1ebc5010ac8>
```

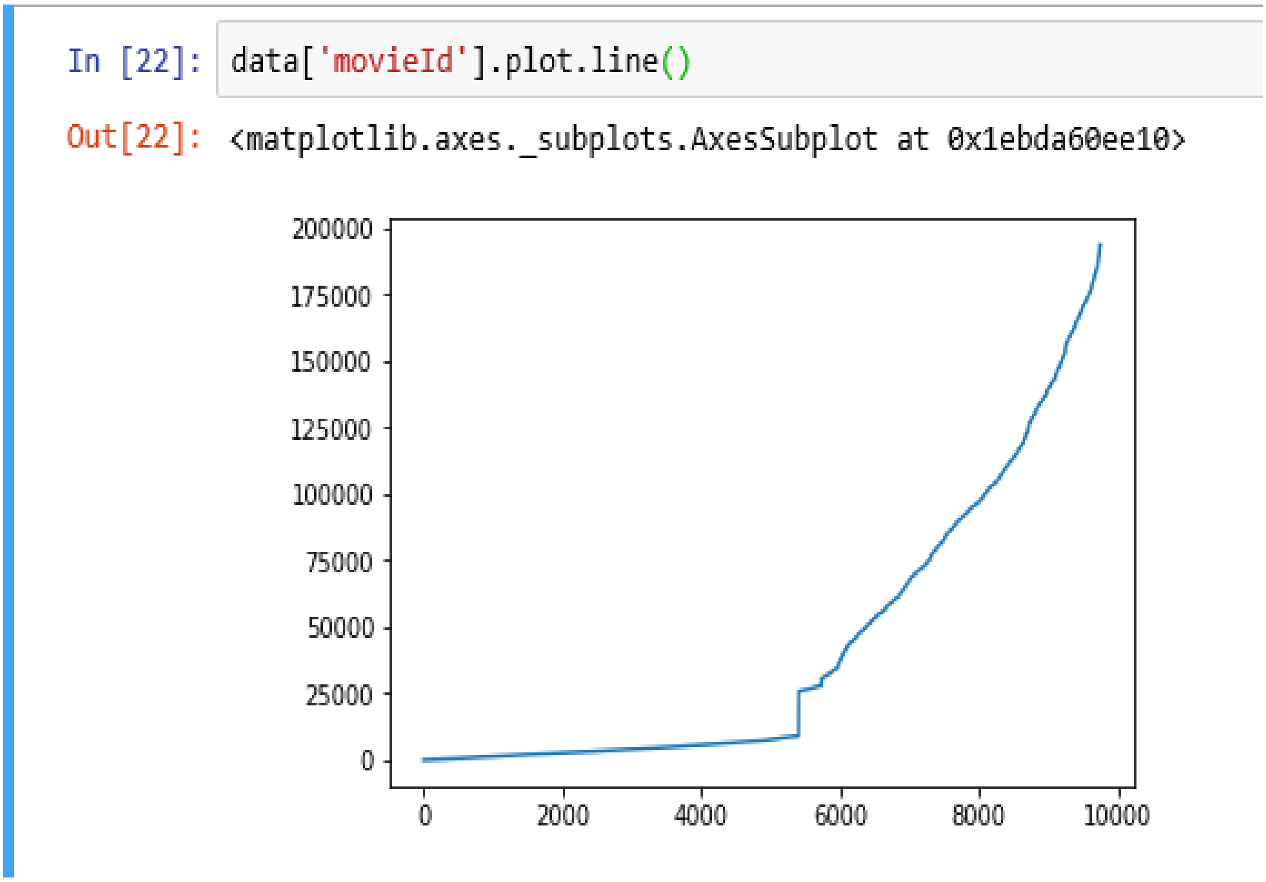


as there are many different ways for plotig bar , you can also plot horizontal bar by using plot.barh() method, everything is simple

but also you can plot bar by sorting data in either ascending or descending order

4. LINE CHART

To create line chart in pandas is pretty easy also, you just need plot.line() method to make it happen, data['movieId'].plot.line(), in this line I have ploted a line chart of movieId attribute, and that is how it looks like.



```
In [22]: data['movieId'].plot.line()

Out[22]: <matplotlib.axes._subplots.AxesSubplot at 0x1ebda60ee10>
```

CONCLUSION

This was just how you can simple activity in data science with pandas I know that one part of machine learning has not been touched,by reading this book carefully you can see how pandas is impoertant for any data scienctist , also you have seen it play a great part in your project, mastring pandas will make you comfortable in data analysis and exploration, you're data. Thank you for your attention , I hope you enjoyed this book, I wish to write another book which will concern about machine learning,explaining different machine learning algorithm.

REFERENCES

-Wikipedia.org/wiki/Pandas_(software)

-python data science handbook by Jake VanderPlas

-think python by Allen B. Downey