

Ex: 9	Interface python to Tableau
--------------	------------------------------------

Aim:

To connect to a data source and create a dashboard to visualize all the spreadsheets.

What is TabPy?

TabPy (the Tableau Python Server) is an Analytics Extension implementation that expands Tableau's capabilities by allowing users to execute Python scripts and saved functions via Tableau's table calculations.

Python

Python is an object-oriented, open-source, high-level programming language with dynamic semantics. Though Python has many high-level data structures it is very user-friendly and simple to learn. Python also supports packages, and modules that are useful to increase program modularity and allows code to be integrated and reused with other technologies.

Tableau

Tableau is a self-service data visualization tool that helps customers to view and present data in the form of interactive dashboards and charts to showcase insights and perform real-time data analytics. Tableau is very user-friendly as it provides a drag-drop user interface to visualize the available data with minimal scripting required only for calculated fields.

How to integrate Python with Tableau?

TabPy is the API that enables the working of Python code from within a Tableau workbook.

Steps to integrate Python with Tableau:

Step - 1 - After downloading Anaconda Navigator, the next step would be to download the Tabpy server. TabPy server can be downloaded by typing `conda install -c anaconda TabPy-server` in the anaconda prompt.

```
(base) C:\Users\ISHWARYA LAKSHMI S.S>conda install -c anaconda tabpy-server
Collecting package metadata (current_repodata.json): done
Solving environment: -
```

After all the packages are installed it will ask for yes or no to proceed, press y to install the server.

```
## Package Plan ##

environment location: C:\Users\ISHWARYA LAKSHMI S.S\anaconda3

added / updated specs:
- tabpy-server

The following packages will be downloaded:
```

package	build		
ca-certificates-2020.1.1	0	165 KB	anaconda
certifi-2019.11.28	py37_0	157 KB	anaconda
conda-4.8.3	py37_0	3.0 MB	anaconda
genson-1.1.0	py_0	19 KB	anaconda
openssl-1.1.1d	he774522_4	5.7 MB	anaconda
simplejson-3.17.0	py37he774522_0	105 KB	anaconda
tabpy-client-0.2	py37_1	27 KB	anaconda
tabpy-server-0.2	py37_1	64 KB	anaconda
tornado-json-1.3.3	py37_0	26 KB	anaconda
Total:		9.3 MB	

```
The following NEW packages will be INSTALLED:

genson                anaconda/noarch::genson-1.1.0-py_0
simplejson             anaconda/win-64::simplejson-3.17.0-py37he774522_0
tabpy-client          anaconda/win-64::tabpy-client-0.2-py37_1
tabpy-server          anaconda/win-64::tabpy-server-0.2-py37_1
tornado-json          anaconda/win-64::tornado-json-1.3.3-py37_0

The following packages will be UPDATED:

conda                pkgs/main::conda-4.8.2-py37_0 --> anaconda::conda-4.8.3-py37_0

The following packages will be SUPERSEDED by a higher-priority channel:

ca-certificates      pkgs/main --> anaconda
certifi              pkgs/main --> anaconda
openssl              pkgs/main --> anaconda

Proceed ([y]/n)?
```

Step - 2 - After the TabPy server is installed, the server should be started in order to connect with Tableau. To start the TabPy-server we should change the directory from the root directory to the folder where TabPy-server is installed. This can be done by typing `cd C:\Users*your username*\Anaconda3\pkgs\tabpy-server-0.2-py37_1\Lib\site-packages\tabpy_server` command in anaconda prompt. This command changes directory to the folder where tabpy_server is installed.

Step - 3 - The next command startup.bat can be typed to start the server.

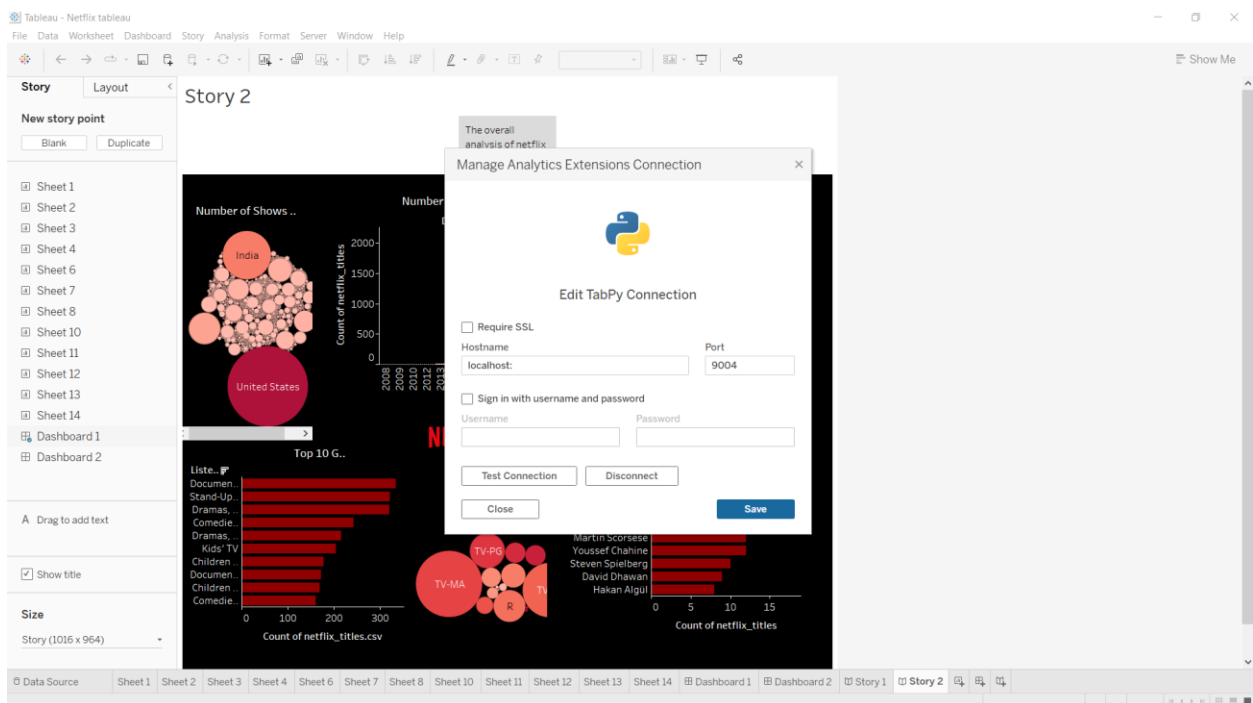
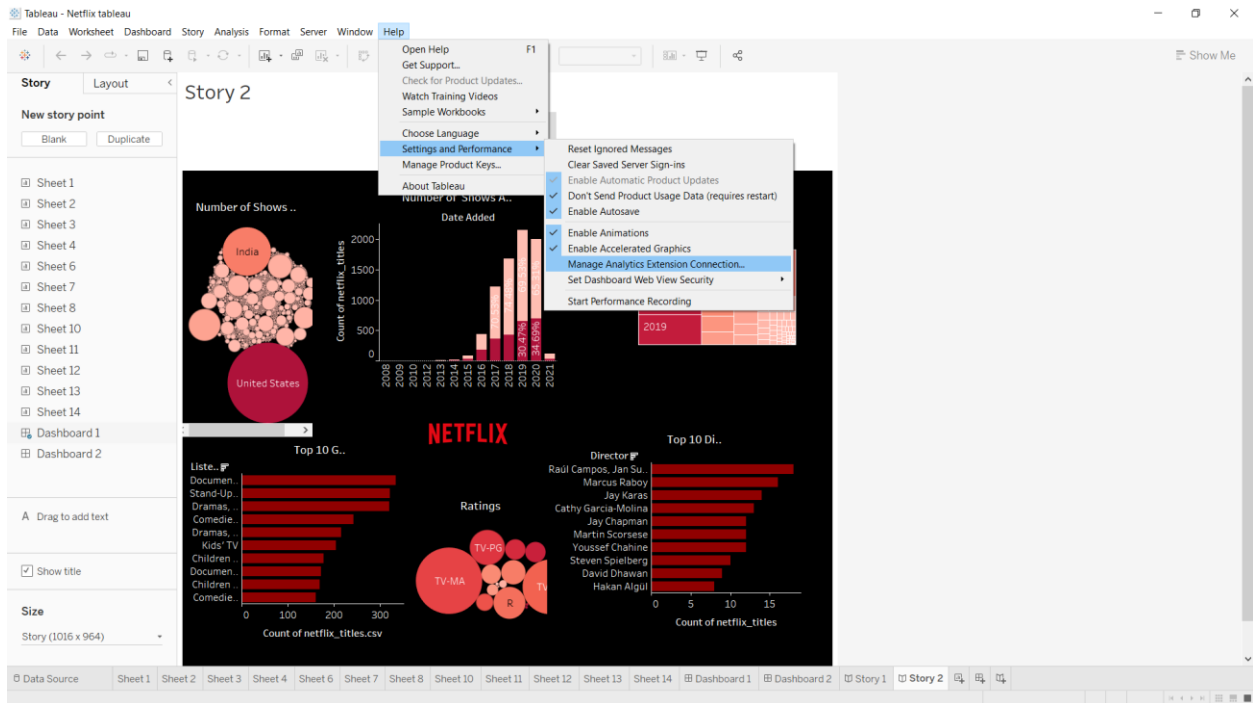
```
(base) C:\Users\ISHWARYA LAKSHMI S.S>cd C:\Users\ISHWARYA LAKSHMI S.S\Anaconda3\Lib\site-packages\tabpy_server
(base) C:\Users\ISHWARYA LAKSHMI S.S\Anaconda3\Lib\site-packages\tabpy_server>startup.bat
Initializing TabPy...
Done initializing TabPy.
Web service listening on port 9004
```

Step - 4 - After the command startup.bat, the prompt displays port number 9004 on which TabPy-server is initialized. After initializing the server, the next part is to connect the server with Tableau. Open Tableau desktop.

Step - 5 - In Tableau go to:

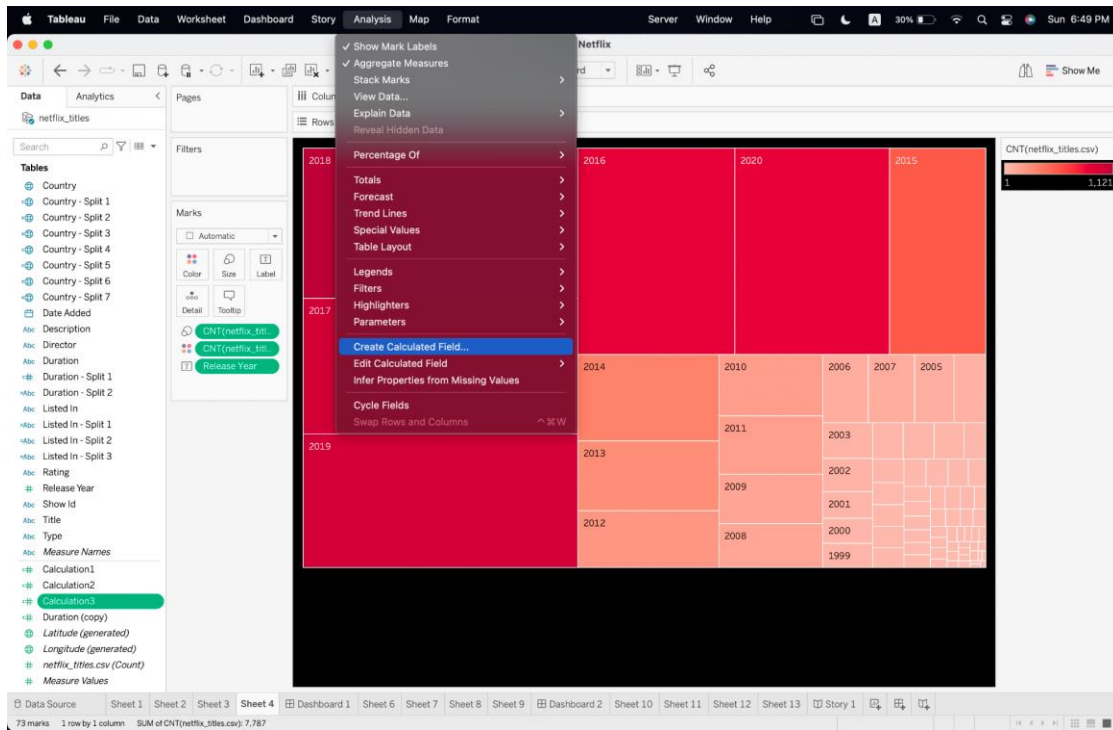
- Open Help menu.
- In that choose settings and performance
- From settings and performance choose to manage the external connection.
- Select TabPy/External API
- Select localhost
- Make sure the port number is 9004

Click on test connection to cross-check the connectivity.

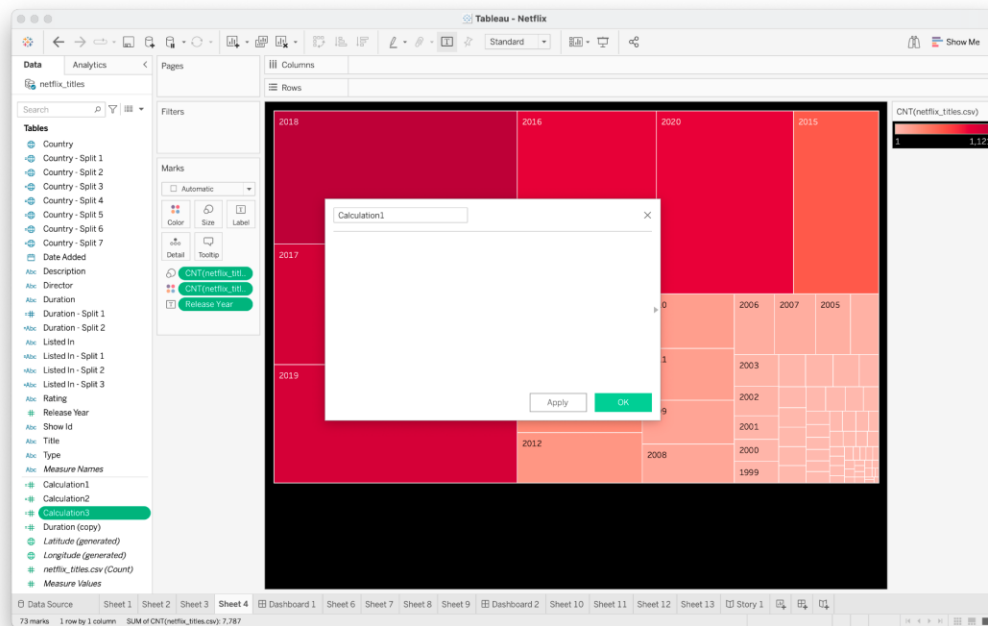


Once the connection is successful then click on the OK button to approve the external connection.

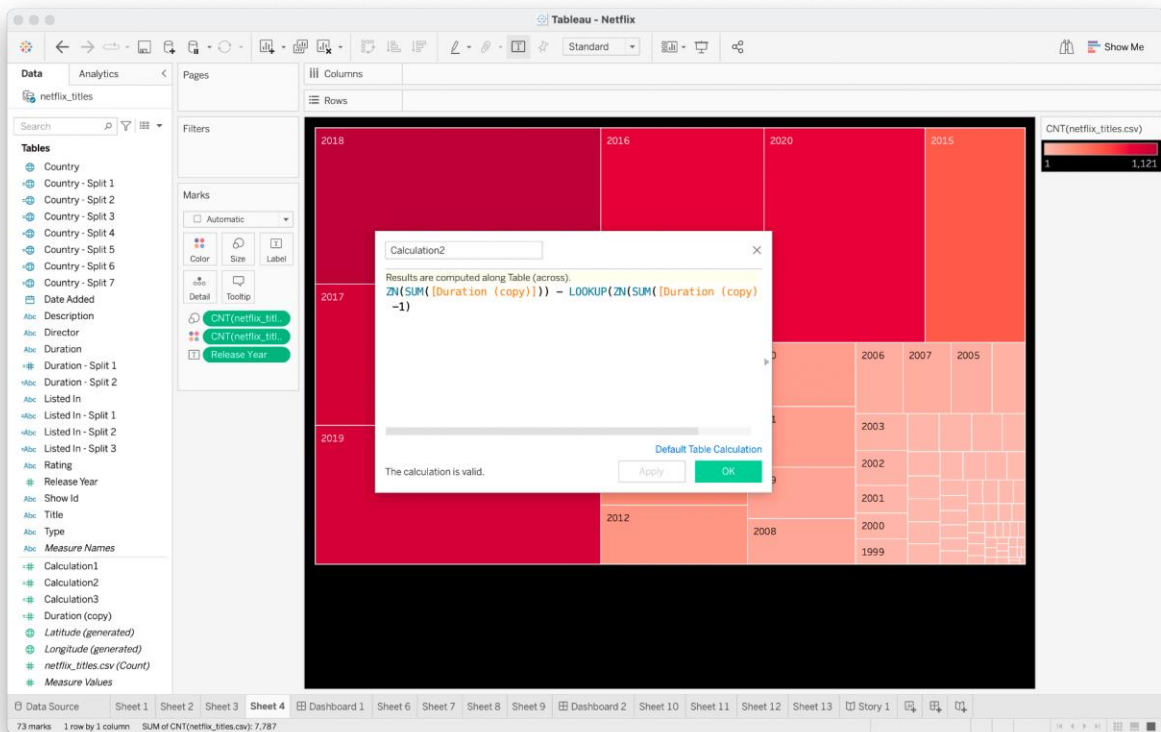
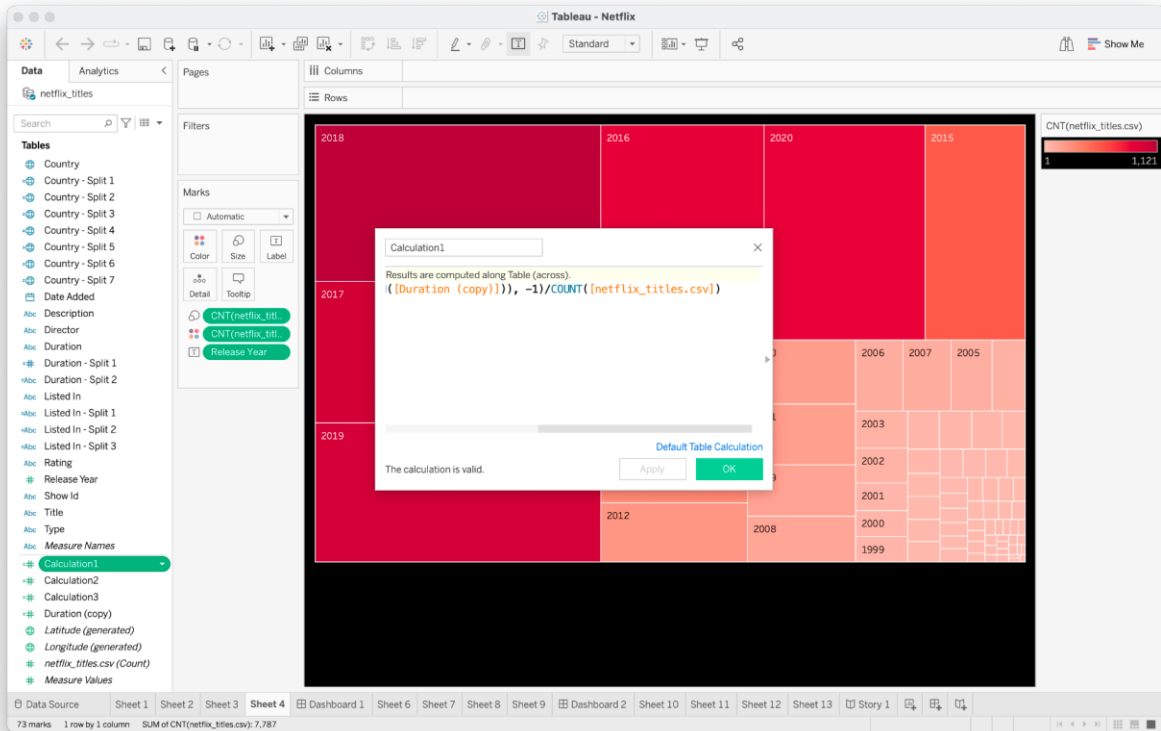
Step - 6 - Create calculated fields to do operations on the charts using TabPy. From the Analyse option from the tableau taskbar, click create collections field to add a new field.



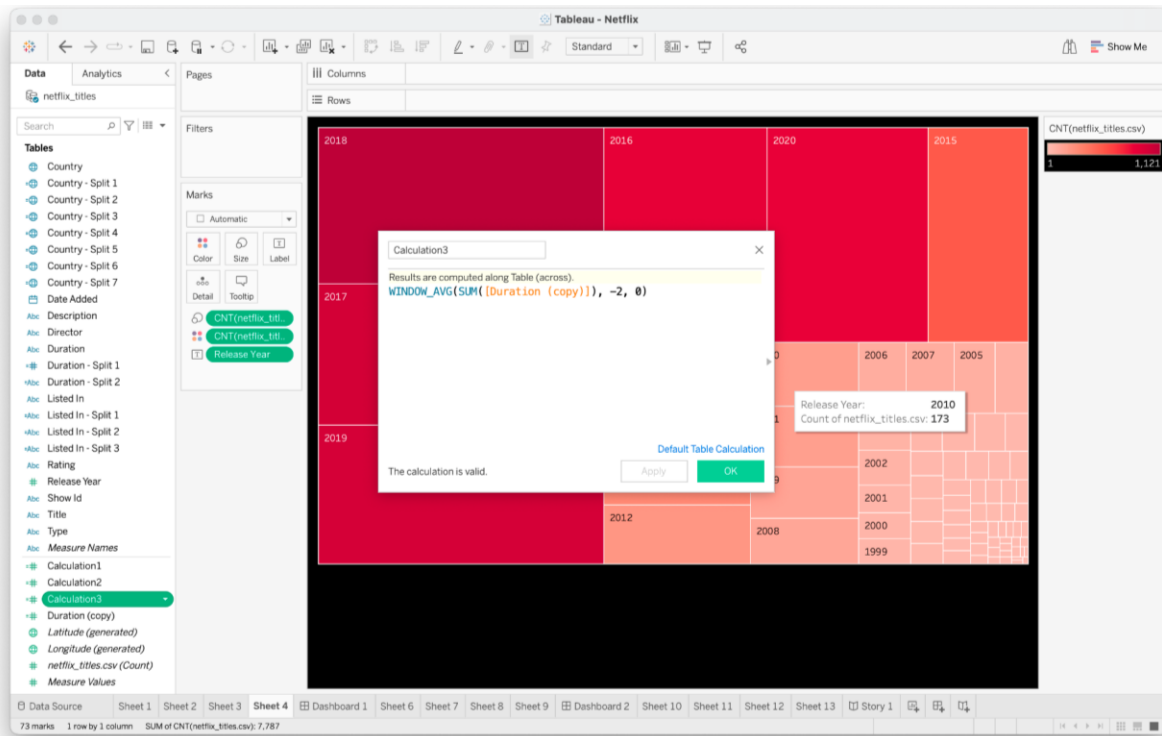
Step - 7 - The empty collections field has been created and we can add the code to analyse the data from within python interpreter



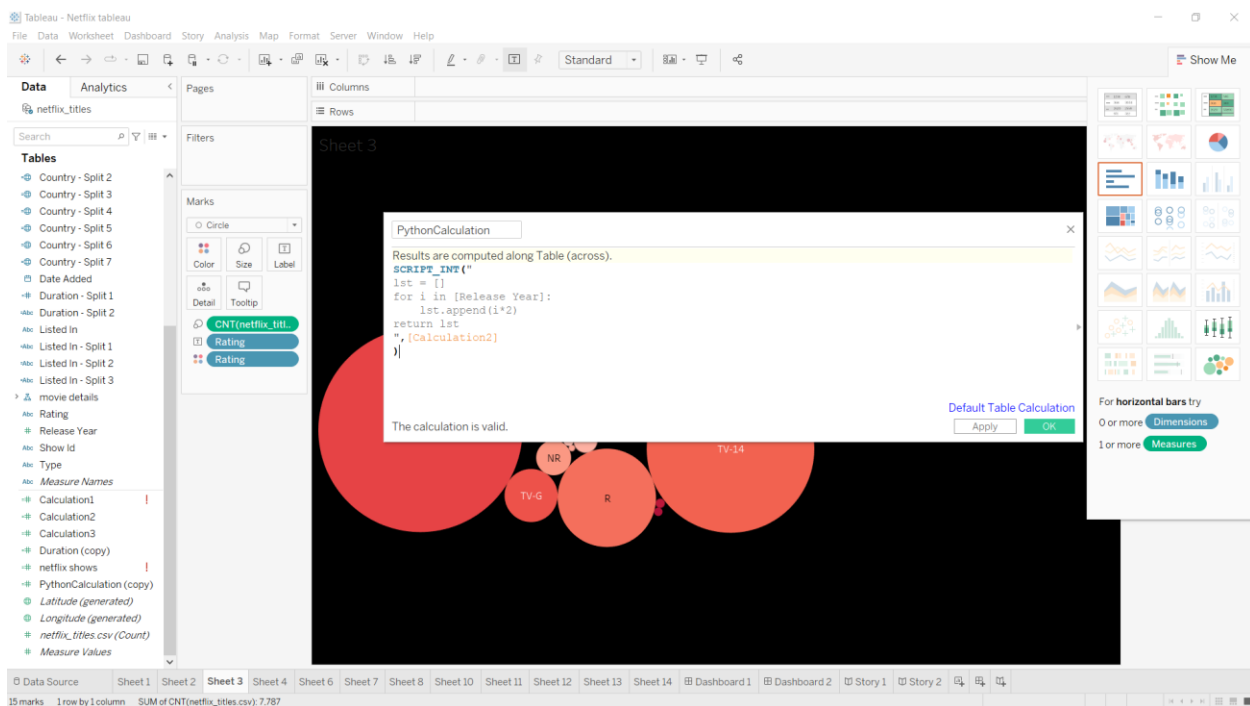
Step - 8 - Add the necessary code into the collections field to get the valid visualization.



Step - 9 - The calculations field has been created and the data is visualized successfully



Using python code to manipulate the data



Why Python + Tableau?

When TabPy is used with Tableau, calculated fields can be defined in Python which enables us to use the power of many machine-learning libraries right from Tableau visualizations. It enables many new features like Machine learning predictions, sentimental analysis, and time series forecasting using various models by customizing calculated fields.

Limitations of integrating Python with Tableau

Though there are many advantages of enabling Tabpy there are also certain limitations.

- When a large dataset is used the waiting time will be more while the script runs each time you make a change to the view.
- The Python-generated calculated fields will not be extracted if you create a Tableau extract.
- The Python script will run only when you put it into the view.
- You cannot use the TabPy calculations to create values and base additional calculations on those values unless you can use both calculated fields in the view.

When deployed together, Python integrated with Tableau can help deliver a scalable, flexible and advanced analytics platform.

Result:

The tableau interface with python is successfully done and the server is connected with the local host.