Hands-on: Python Integration to Tableau

(note that this tutorial may increase anxiety and effective learning, thus, enjoyment)

The goal of this hands-on activity is to demonstrate how Tableau and Python can be integrated. The calculated outputs in Python such as supervised learning results can be used in Tableau visualizations. Interactive dashboard design in Tableau which uses the Python outputs would be ideal practice of this hands-on activity.

Briefly you just need Python and TabPy Server installed and connect the server in Tableau Desktop. To practice, you need to use the same data set, Sample - Superstore.xls, from the previous hands-on activity. Let's start.

- Zeroth, you should install Tableau Desktop. Tableau Public doesn't integrate Python. Tableau will have some issues in the integration so be patient with this tutorial.

- First, I would recommend having Anaconda Distribution on your computer. It is a very rich open-source Python distribution supporting many packages and software platforms. Just download and install it from https://www.anaconda.com/products/distribution though this is not necessary for this hands-on if you have any latest Python installed (we will need it for JupyterLab for the next hands-on's).

- Install Python>=3.7 from https://www.python.org/downloads/. Start taking a look at https://www.python.org/doc/ for resources. I will provide first session workshop and more resources. For now, just be familiar with it from python.org. Then, practice https://towardsdatascience.com/simple-linear-regression-in-python-numpy-only-130a988c0212 (this is step by step building simple linear regression to be familiar with Python) or https://towardsdatascience.com/simple-linear-regression-model-using-python-machine-learning-eab7924d18b4.

- The first chunk of the Python code below calculates the Pearson's correlation coefficient (rho) between two variables, namely _arg1 and _arg2. The second chunk gets the p-value when testing the equality of rho to zero. The third chunk fits the data to a simple linear regression that produces a vector of fitted values. The fourth one gets the prediction of a nonparametric model.

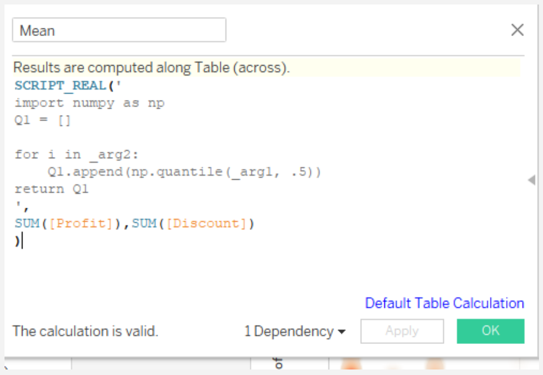
```
 import scipy.stats as ss
 return ss.pearsonr(_arg1,_arg2)[0]

 import scipy.stats as ss
 return ss.pearsonr(_arg1,_arg2)[1]

import numpy as np
from sklearn import linear_model as lm
model=lm.LinearRegression()
x=np.transpose(np.array([_arg1]))
y=np.array(_arg2)
model.fit(x,y)
return model.predict(x).tolist()

 import statsmodels.api as sm
 lowess=sm.nonparametric.lowess(_arg1,_arg2,frac=.75)
 return lowess.T[1].tolist()
```

- Some other Python examples from students:



Mean

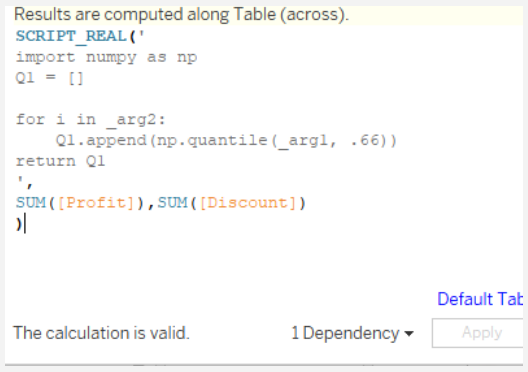
Results are computed along Table (across).

```
SCRIPT_REAL('
import numpy as np
Q1 = []

for i in _arg2:
    Q1.append(np.quantile(_arg1, .5))
return Q1
',
SUM([Profit]),SUM([Discount])
)
```

Default Table Calculation

The calculation is valid. 1 Dependency Apply OK



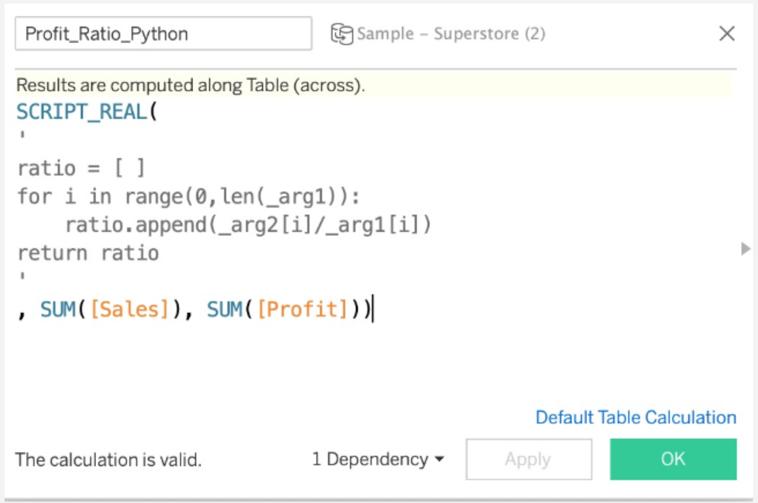
Results are computed along Table (across).

```
SCRIPT_REAL('
import numpy as np
Q1 = []

for i in _arg2:
    Q1.append(np.quantile(_arg1, .66))
return Q1
',
SUM([Profit]),SUM([Discount])
)
```

Default Tab

The calculation is valid. 1 Dependency Apply



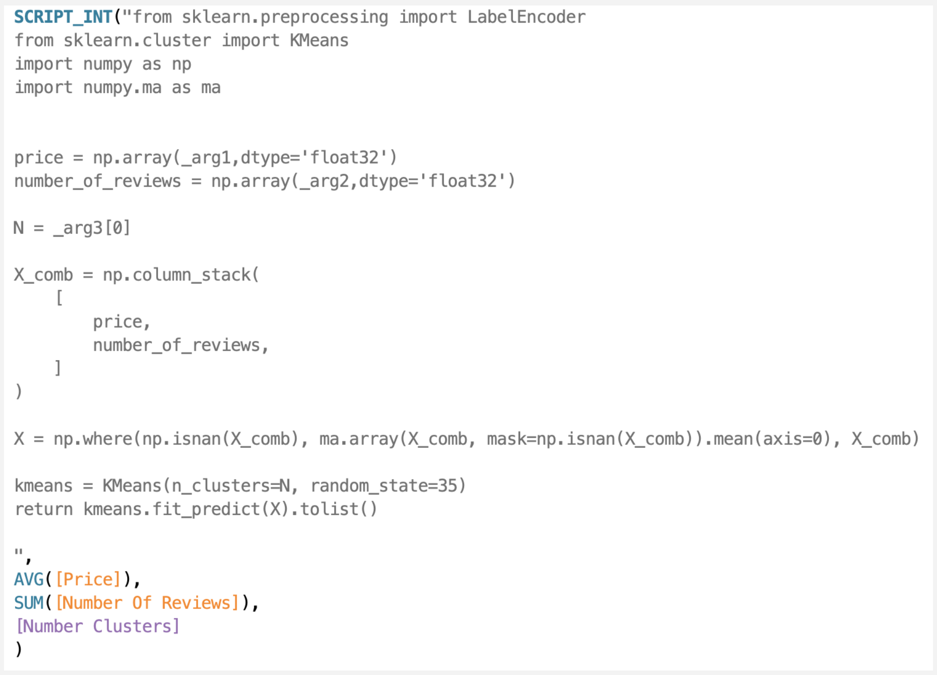
Profit_Ratio_Python Sample – Superstore (2)

Results are computed along Table (across).

```
SCRIPT_REAL(
'
ratio = [ ]
for i in range(0,len(_arg1)):
    ratio.append(_arg2[i]/_arg1[i])
return ratio
'
, SUM([Sales]), SUM([Profit]))
```

Default Table Calculation

The calculation is valid. 1 Dependency Apply OK



```
SCRIPT_INT("from sklearn.preprocessing import LabelEncoder
from sklearn.cluster import KMeans
import numpy as np
import numpy.ma as ma


price = np.array(_arg1,dtype='float32')
number_of_reviews = np.array(_arg2,dtype='float32')

N = _arg3[0]

X_comb = np.column_stack(
    [
        price,
        number_of_reviews,
    ]
)

X = np.where(np.isnan(X_comb), ma.array(X_comb, mask=np.isnan(X_comb)).mean(axis=0), X_comb)

kmeans = KMeans(n_clusters=N, random_state=35)
return kmeans.fit_predict(X).tolist()

",
AVG([Price]),
SUM([Number Of Reviews]),
[Number Clusters]
)
```

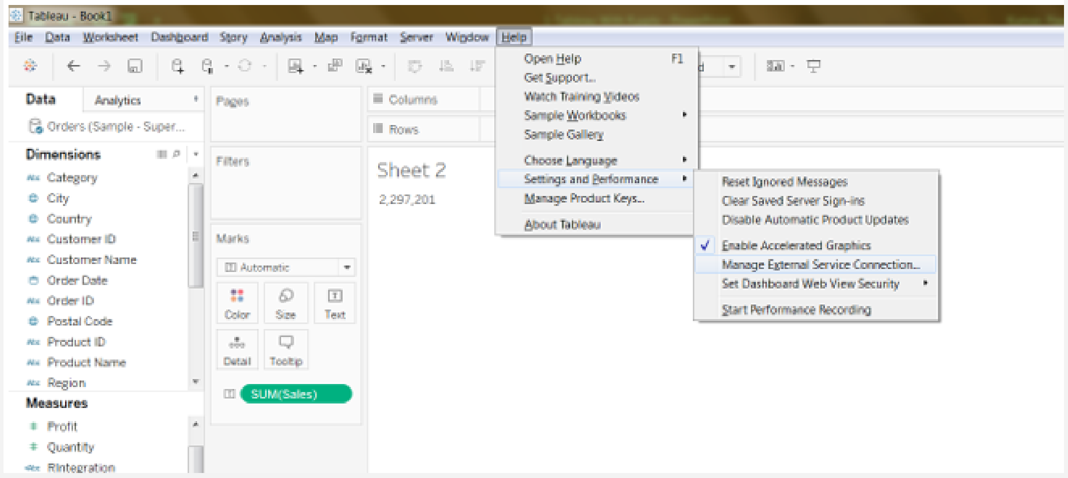
- Now, time to integrate:

Read the guide briefly https://community.tableau.com/s/news/a0A4T000002O55NUAS/tableau-and-python-integration.

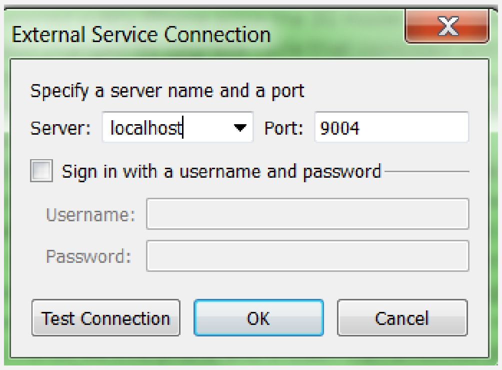
Basically, you don't need to change any configuration by default. Just run these one by one in terminal:

```
python -m pip install --upgrade pip
pip install tabpy
tabpy
```

Now, TabPy server is running in a port. Configure a TabPy Connection on Tableau. On the Help menu in Tableau Desktop, choose Settings and Performance > Manage External Service Connection to open the TabPy connection dialog box.



- Specify a port. Port 9004 is the default port for TabPy servers. Click Test Connection. Click OK.


External Service Connection

Specify a server name and a port

Server: localhost Port: 9004

[ ] Sign in with a username and password

Username:

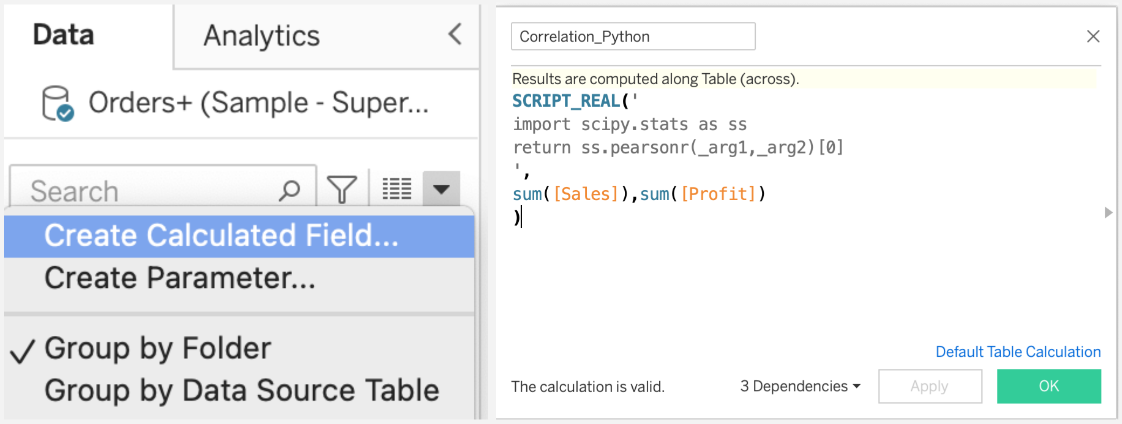
Password:

- In order to let Tableau know that the calculations need to go to Python, it must be passed through one of the 4 functions. These 4 functions are SCRIPT_BOOL , SCRIPT_INT , SCRIPT_REAL , SCRIPT_STR. Python Functions are computed as Table calculations in Tableau. Since these are table calculations, all the Fields being passed to Python must be aggregated like Sum(PROFIT), MIN(Profit), Max (Profit), ATTR(Category) etc.

Run a Python script on Tableau: for example, SCRIPT_BOOL returns a Boolean result from the specified expression. The expression is passed directly to a running external service instance. In Python expressions, use *_argn* (with a leading underscore) to reference parameters (_arg1, _arg2, etc.). See _arg1 is equal to SUM([Profit]).

Just be careful when dealing with aggregated data. All the Fields being passed to python must be aggregated like Sum(PROFIT), MIN(Profit), Max (Profit), ATTR(Category) etc.

For example, in Tableau, create a calculated field for the Pearson's correlation coefficient as below:

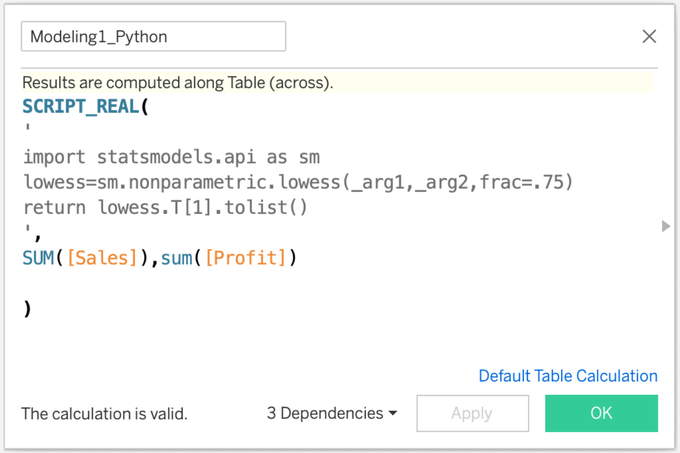

Data
Analytics
Orders+ (Sample - Super...
Search
Create Calculated Field...
Create Parameter...
✓ Group by Folder
Group by Data Source Table
Correlation_Python
Default Table Calculation
3 Dependencies
Apply
OK

Results are computed along Table (across).

```
SCRIPT_REAL('
import scipy.stats as ss
return ss.pearsonr(_arg1,_arg2)[0]
',
sum([Sales]),sum([Profit])
)
```

The calculation is valid.

What about vector outputs? Our first calculation yielded only one data point. Now, the code that generates a vector in modeling will look like this (using the LOWESS method, let's call this is our **first model** with Python):



Results are computed along Table (across).

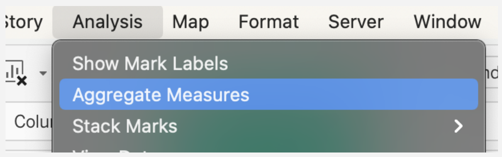
```
SCRIPT_REAL(
'
import statsmodels.api as sm
lowess=sm.nonparametric.lowess(_arg1,_arg2,frac=.75)
return lowess.T[1].tolist()
',
SUM([Sales]),sum([Profit])

)
```

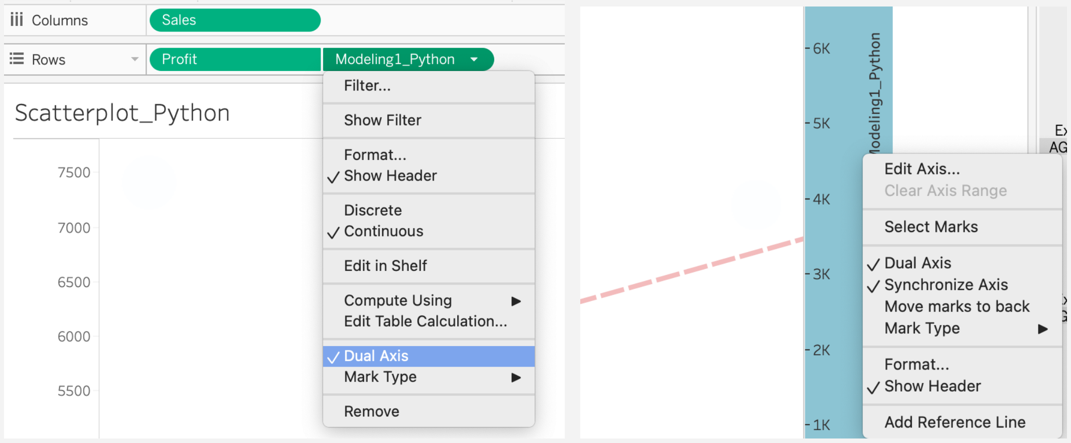
The calculation is valid.

Also, try fitting a simple linear regression with the code above.

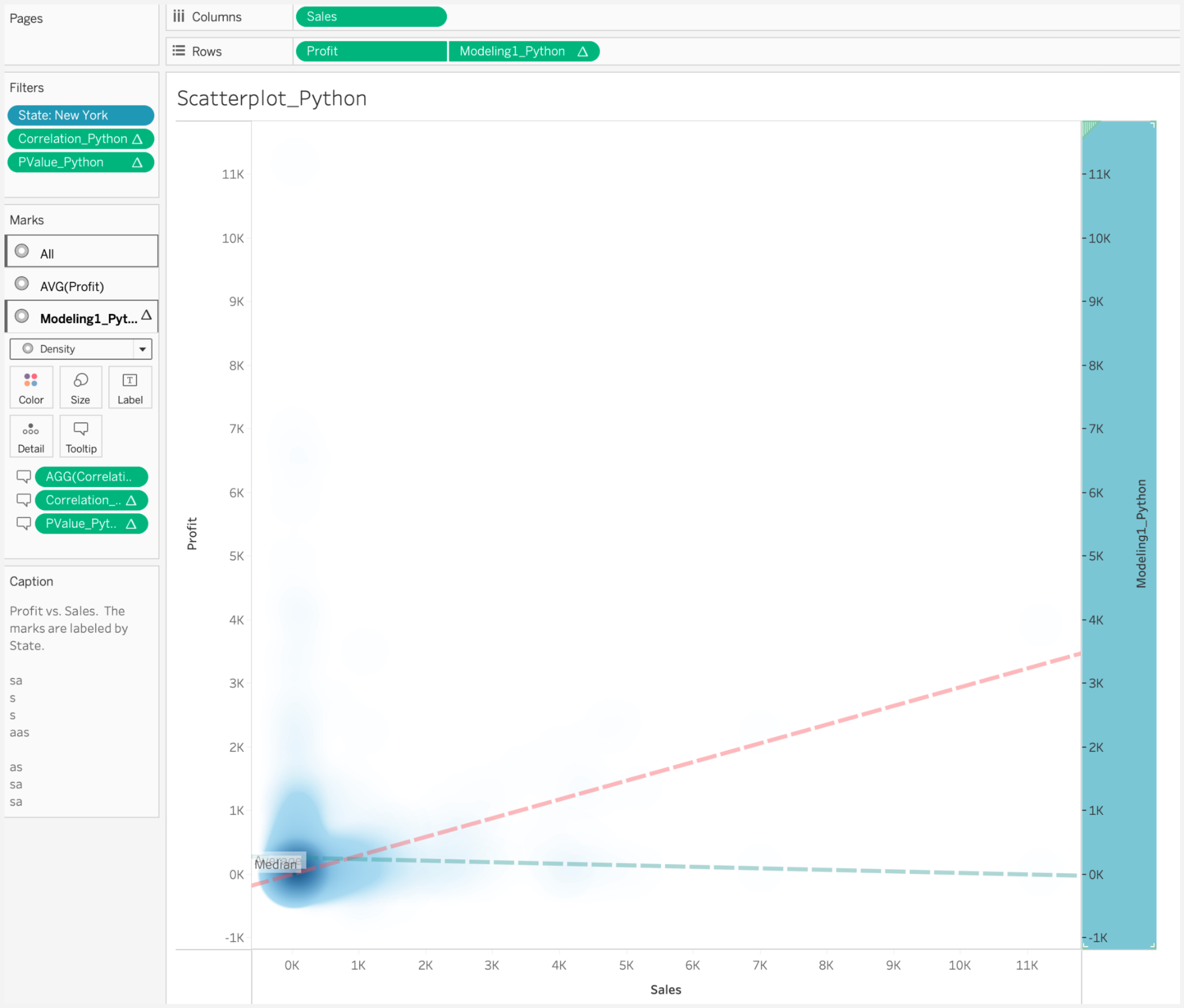
- Let's obtain a scatterplot of Sales and Profit in Tableau. Make it density if too many data points are dealt. Add a trend line (linear, our **second model**). Disaggregate the data. For more info, visit https://help.tableau.com/current/pro/desktop/en-us/buildexamples_scatter.htm or https://data-flair.training/blogs/tableau-scatter-plot/.



Drag the new measure, Modeling1_Python, on Rows. Make Dual Axis and Synchronize Axis.



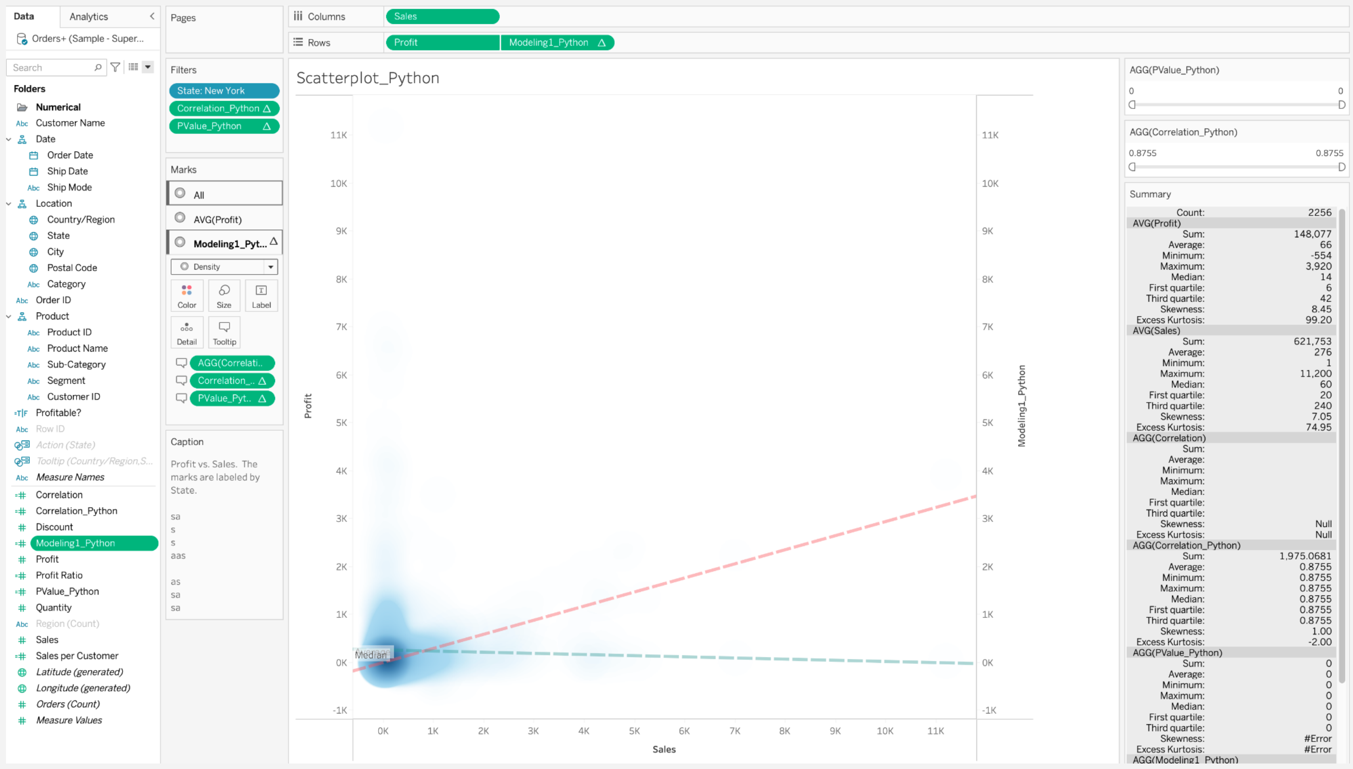
- At the end, you will get the visualization of the two simple models (linear and LOWESS) fitted to the data using Tableau and Python integration like this: see the models are shown as lines. You can work with the SLR code without use of the trend line.



As much as possible, complete the instruction. If you use Tableau Prep, this may help to use the integration within it: [Python Integration in Tableau](#).

- **Submission**

You have employed Python in Tableau Desktop. Keep working on the dashboard you designed previously. Add story points and integrate Python on it. Create a calculated measure (any in your interest) using Python, and incorporate it in visualizations (any). Then just **submit the two screenshots of the workbook or dashboard that shows the calculated measure name, Python code, and its use in the visualization on Discussion**. An example:



Columns: Sales
Rows: Profit, Modeling1_Python

Filters: State: New York; Correlation_Python; PValue_Python

Caption: Profit vs. Sales. The marks are labeled by State.

sa
s
s
aas

as
sa
sa

AGG(PValue_Python): 0 – 0

AGG(Correlation_Python): 0.8755 – 0.8755

Summary

Count:	2256
AVG(Profit)	
Sum:	148,077
Average:	66
Minimum:	-554
Maximum:	3,920
Median:	14
First quartile:	6
Third quartile:	42
Skewness:	8.45
Excess Kurtosis:	99.20
AVG(Sales)	
Sum:	621,753
Average:	276
Minimum:	1
Maximum:	11,200
Median:	60
First quartile:	20
Third quartile:	240
Skewness:	7.05
Excess Kurtosis:	74.95
AGG(Correlation)	
Sum:	
Average:	
Minimum:	
Maximum:	
Median:	
First quartile:	
Third quartile:	
Skewness:	Null
Excess Kurtosis:	Null
AGG(Correlation_Python)	
Sum:	1,975.0681
Average:	0.8755
Minimum:	0.8755
Maximum:	0.8755
Median:	0.8755
First quartile:	0.8755
Third quartile:	0.8755
Skewness:	1.00
Excess Kurtosis:	-2.00
AGG(PValue_Python)	
Sum:	0
Average:	0
Minimum:	0
Maximum:	0
Median:	0
First quartile:	0
Third quartile:	0
Skewness:	#Error
Excess Kurtosis:	#Error
AGG(Modeling1_Python)	

- **BONUS:** practice https://www.tableau.com/about/blog/2017/1/building-advanced-analytics-applications-tabpy-64916 or, practice the forecasting example posted at https://towardsdatascience.com/forecasting-with-python-and-tableau-dd37a218a1e5. Submit the designed dashboard on Discussions under Bonus category.

<u>Troubleshooting:</u>

- Visit the Slack server and start sharing issues and suggestions.
- https://www.tableau.com/about/blog/2021/2/tableau-online-analytics-extensions-now-available