Tableau for Data Scientists

Joel Hutchison
Customer Consultant
Tableau
Understanding the Why
Why Python?  Why R?  Why Tableau?

"Visualization of data (static or interactive).

Storytelling with data. This is a critical skill.

In essence, can someone with no background in whatever area your project is in look at your project and gain some new understandings from it?"

https://www.forbes.com/sites/louiscolumbus/2019/04/14/how-to-get-your-data-scientist-career-started/#3eeb52a47e5c
We help people see and understand their data.
Telling your story.

Advanced Analytical Languages

- Peer-reviewed mathematical and statistics packages built by domain experts
- Enrich data with machine learning and natural language processing libraries
- Perform heavy statistical testing
- Create and iterate on regression model

Visual Analytics in Tableau

- Tableau’s visual analytics makes it faster and easier to identify patterns, trends and relationships
- Tableau allows users to easily share and communicate insights
- Tableau enables users to ask and answer their own questions
Combined Benefits

- Enable broader audiences to use sophisticated models and statistics in decision-making
- Empower analytical package power-users to uncover more through fluid data exploration
- Enhance the OOTB function-library with available statistical libraries and centralized algorithms
- Easily tell your data story!
Understanding the How
How does it work?

Data Sources
- Files
- Databases
- Big Data
- Cloud
- Apps

Tableau Desktop
- extract
- connect live
- publish workbooks

Tableau Server
- extract
- connect live

External Services
- R code
- Python code
- RServe
- TabPy
Preprocessing the data

Data Sources
- Files
- Databases
- Big Data
- Cloud
- Apps

Preprocess Data
- R
- Python

Write to a database or a Tableau Hyper Extract

Tableau Desktop
Tableau Server
The TabPy server allows for the remote execution of Python code. It has two components:

- A server process built on Tornado, which allows for the remote execution of Python code through a set of REST APIs.
- A tools library that enables the deployment of such endpoints, based on Python functions

https://github.com/tableau/TabPy/blob/master/docs/about.md

Rserve is a TCP/IP server which allows other programs to use facilities of R from various languages without the need to initialize R or link against R library.

- Rserve supports remote connection, authentication and file transfer.

https://www.rforge.net/Rserve/
**SCRIPT_*() functions in Tableau**

1. Functions telling Tableau to use an external service.
   - `SCRIPT_REAL()` returns real or decimal numbers
   - `SCRIPT_INT()` returns integers or whole numbers
   - `SCRIPT_STR()` returns strings (words and text)
   - `SCRIPT_BOOL()` returns Booleans (true/false)
SCRIPT_*() functions in Tableau

2. The actual R / Python code to be executed.
   - Tableau treats this as a string, sends it to Rserve / TabPy to interpret
SCRIPT_*() functions in Tableau

3. The data from Tableau.
   - As many arguments as needed
   - Can be [fields] or [parameters]
   - All fields must be aggregated
     \text{MIN()}, \text{MAX()}, \text{SUM}(), \text{etc.}
4. The data from Tableau is passed in the code as arguments

- arg1, arg2, arg3, etc. indicates where to put the data into the code
- In example on the left
  .arg1 = MAX([Timestamp]), .arg2 = SUM([Tweets])
- R: .arg1, .arg2, etc.
- Python: _arg1, _arg2, etc.
The Nuts and Bolts
Installing TabPy

1. Install Python

2. Install TabPy
   • pip install tabpy-server

1. Install required python modules
   • python -m pip install numpy scipy pandas statsmodels patsy sklearn nltk

2. Initialize sentiment lexicon on Python console
   • import nltk
   • nltk.download('vader_lexicon')

3. Start Tabpy from the command line

More details on the install can be found on Github.
Install RServe

1. Install \texttt{R}
2. Optionally install \texttt{Rstudio}
3. Run \texttt{R} (IDE like RStudio, GUI, CLI)
4. Install required packages
   \begin{itemize}
   \item \texttt{install.packages(c("Rserve", "forecast", "dbscan", "dplyr", "tidytext"))}
   \end{itemize}
5. Start \texttt{Rserve} session
   \begin{itemize}
   \item \texttt{library(Rserve)}
   \item \texttt{run.Rserve()}
   \end{itemize}
Connect Tableau Desktop to Rserve / TabPy
Connect Tableau Server to Rserve / TabPy

Tableau Server

TablePy or Rserve

Rserve
TabPy

workbooks
external services

IP & port

```
tsm configuration set -k vizqlserver.extsvc.host -v <IP>
tsm configuration set -k vizqlserver.extsvc.port -v <port>
```
Additional Considerations
Additional Considerations

1. Tableau Desktop and Server currently only support **one External Service**

2. No support for External Services with **Tableau Online** and **Tableau Public**

3. Security and best practices require putting External Services on a **separate machine** and limiting access.

4. If latency for calculation processing times are high, consider **pre-processing data** before analyzing it in Tableau.
Use Cases
Forecasting Time Series Data

Python - Forecasting - Frankfurt Temperatures

R - Forecasting - Frankfurt Temperatures
Forecasting Time Series Data

**R**

```r
library(forecast)

inputData = na.omit(.arg1)
startDate = as.Date(min(na.omit(.arg2)))

timeSeries = ts(inputData,
               start = startDate,
               deltat = 1/52)

timeSeriesForecast = forecast(timeSeries,
                               h = length(.arg1) -
                               length(inputData),
                               level = 95)

append(inputData,
       timeSeriesForecast$mean)
```

**Python**

```python
import numpy as np
import pandas as pd
from statsmodels.tsa.holtwinters import ExponentialSmoothing

series = pd.DataFrame.from_items([('ts', _arg1), ('y', _arg2)])
last_week = np.where(pd.isnull(series))[0][0]
weeks_to_forecast = len(series) - last_week

model_fit = ExponentialSmoothing(series.iloc[:last_week, 1],
                                  seasonal_periods=52,
                                  trend='add',
                                  seasonal='add').fit()

yhat = model_fit.forecast(weeks_to_forecast)

return np.concatenate([series.iloc[:last_week, 1],
                       yhat]).tolist()
```

```
AVG([Temperature]),
MAX([forecastWeek]))
```
Clustering Crime
Clustering Crime

R

```
library(dbscan)

data <- cbind((.arg1 * pi) / 180, (.arg2 * pi) / 180)

db <- dbscan(data,
              eps = 1/39590,
              minPts = .arg3[1])$cluster

db[db > 0] <- 'Yes'
db[db == 0] <- 'No'

db
```

Python

```
import numpy as np
from sklearn.cluster import DBSCAN

X = np.column_stack([np.radians(_arg1), np.radians(_arg2)])

db = DBSCAN(eps=_arg3[1], min_samples=_arg4[1],
            metric='haversine').fit(X)

return np.where(db.labels_ == np.array(-1), \
                'No', 'Yes').tolist()

", 
AVG([Latitude]),
AVG([Longitude]),
AVG([Incident Count])
```

\( \text{AVG} \left( \text{Latitude} \right) \),
\( \text{AVG} \left( \text{Longitude} \right) \),
\( \text{AVG} \left( \text{Incident Count} \right) \)
Thank You

Tableau