Data Science: Data Analytics and Visualizations Using Python

ASSOC. PROF. DR SITI SOPHIAYATI YUHANIZ

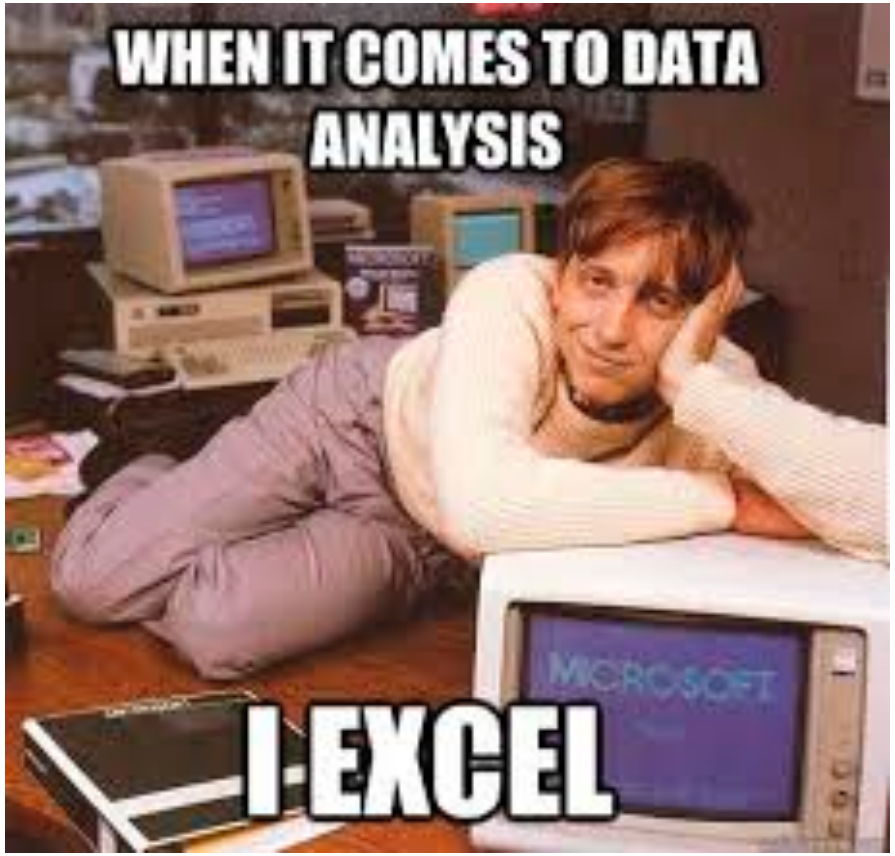
MACHINE LEARNING FOR DATA SCIENCE INTEREST GROUP

UNIVERSITI TEKNOLOGI MALAYSIA

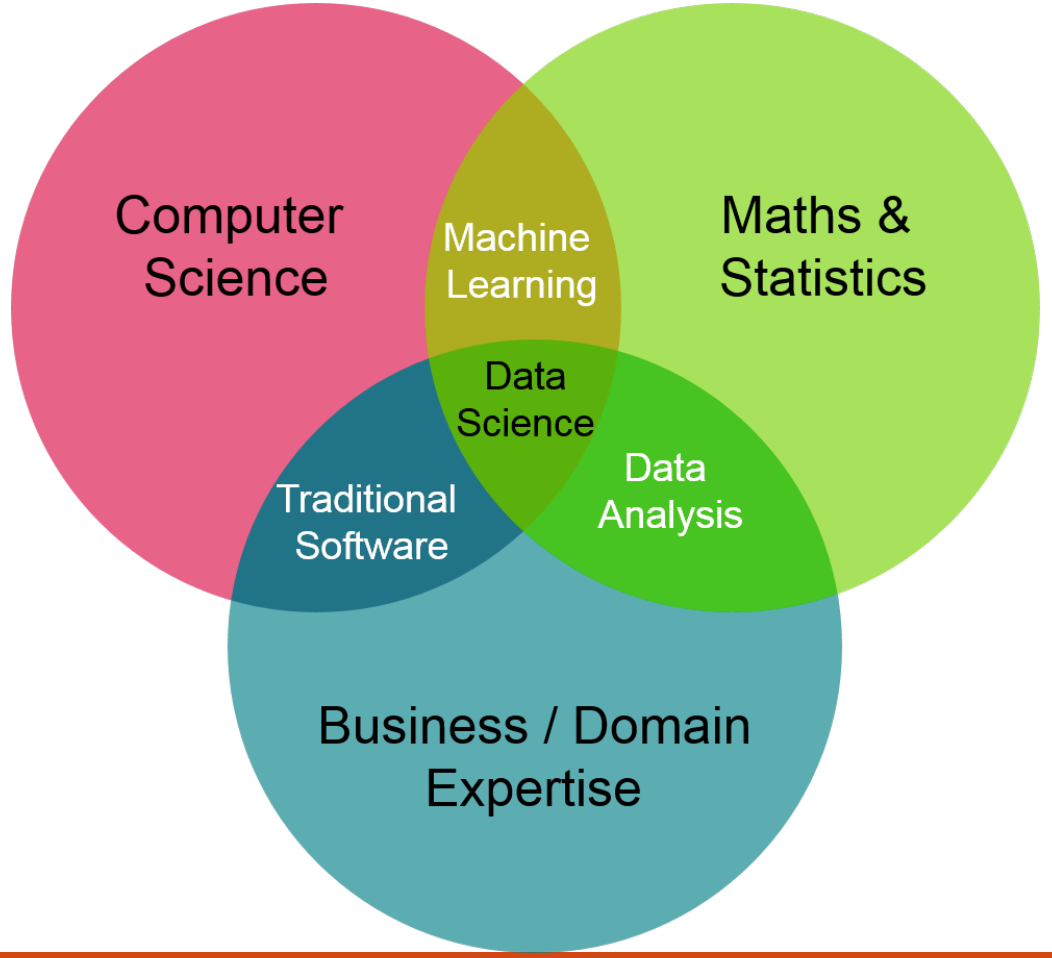
UTM
UNIVERSITI TEKNOLOGI MALAYSIA

Outline

- Data Science
- IOSEMN
- Example of IOSEMN in action

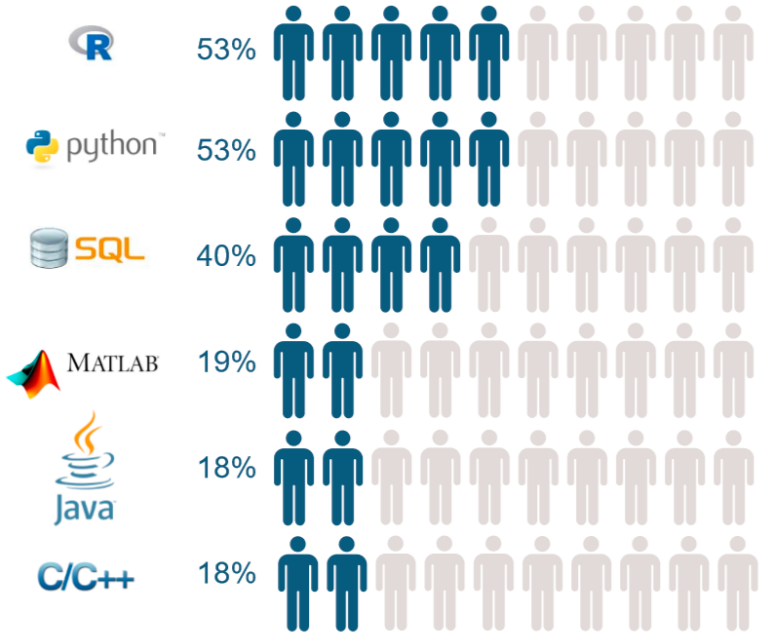


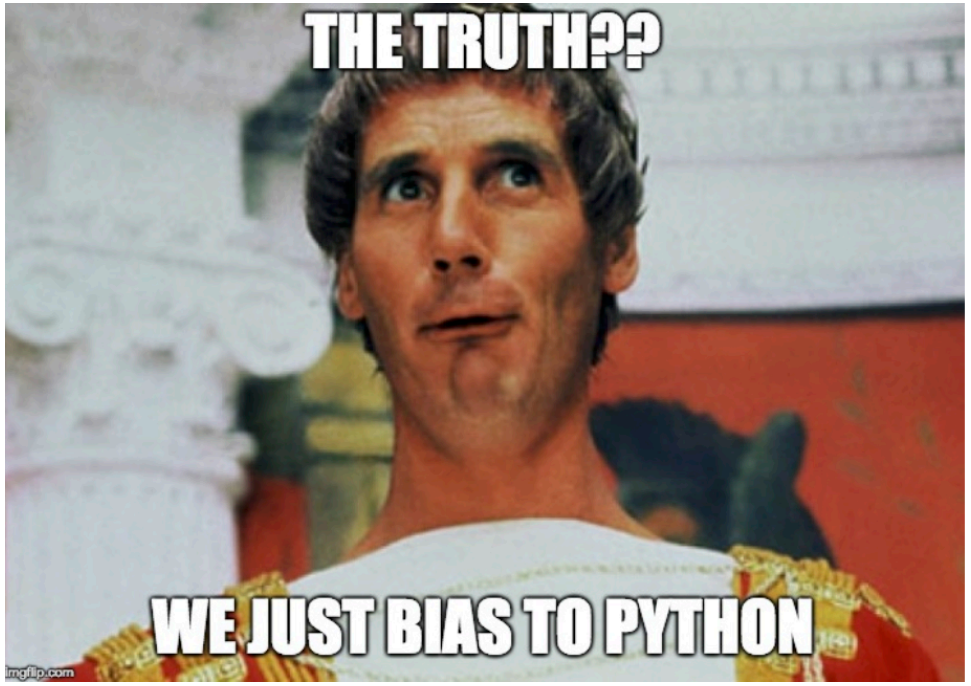
What is Data Science



Why Python?

- Readability
- Supported libraries
- General purpose language





Why Python?

Speech Recognition in Just 5 Lines of Code!

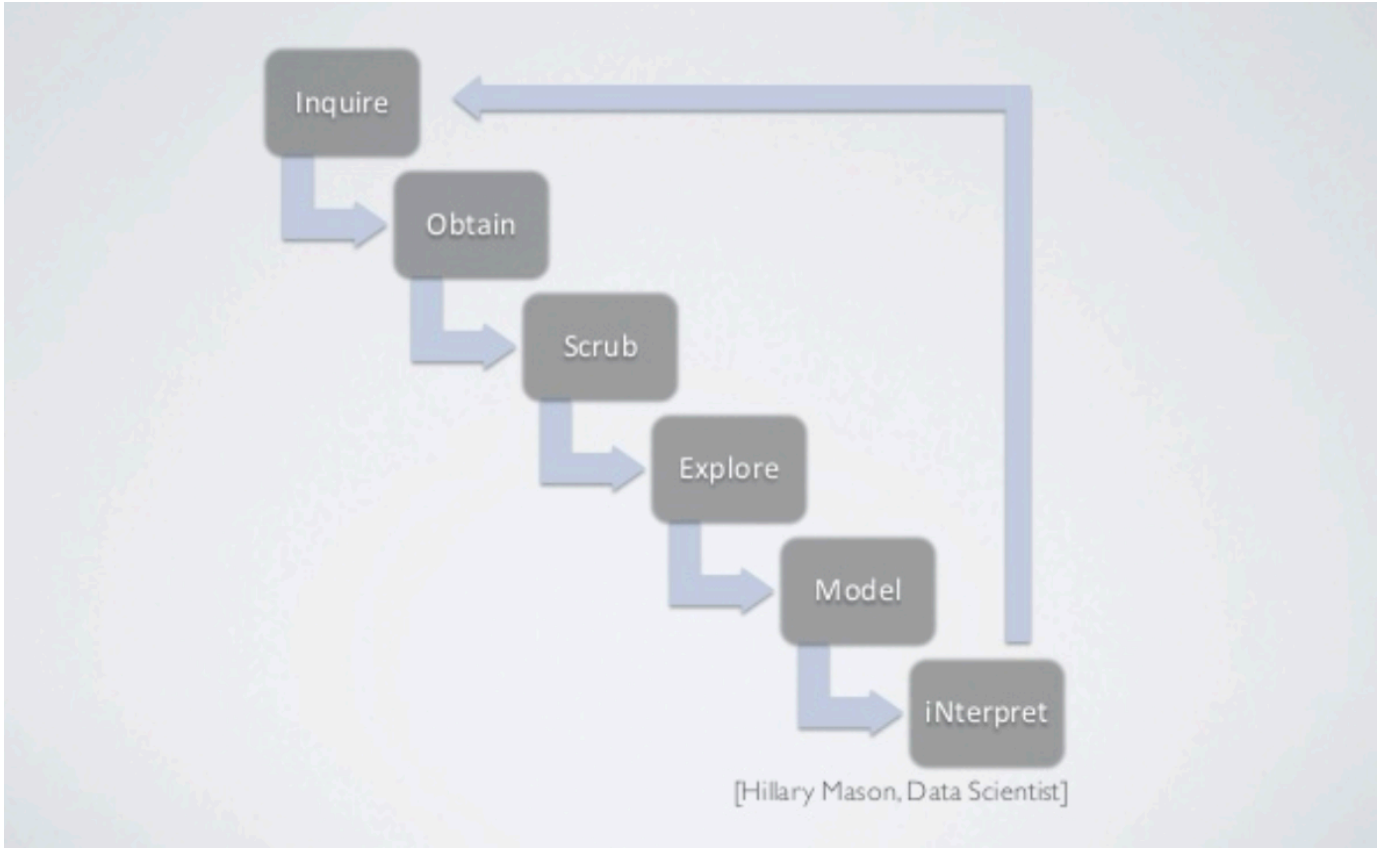
```
#Step 1 - import library
import cmu_sphinx4
#Step 2 - read audio file
audio_URL = 'http://some.site.com/audio.wav'
transcriber = cmu_sphinx4.Transcriber(audio_URL)
#Step 3 - Print it out
for line in transcriber.transcript_stream():
    print line
```

Why Python?

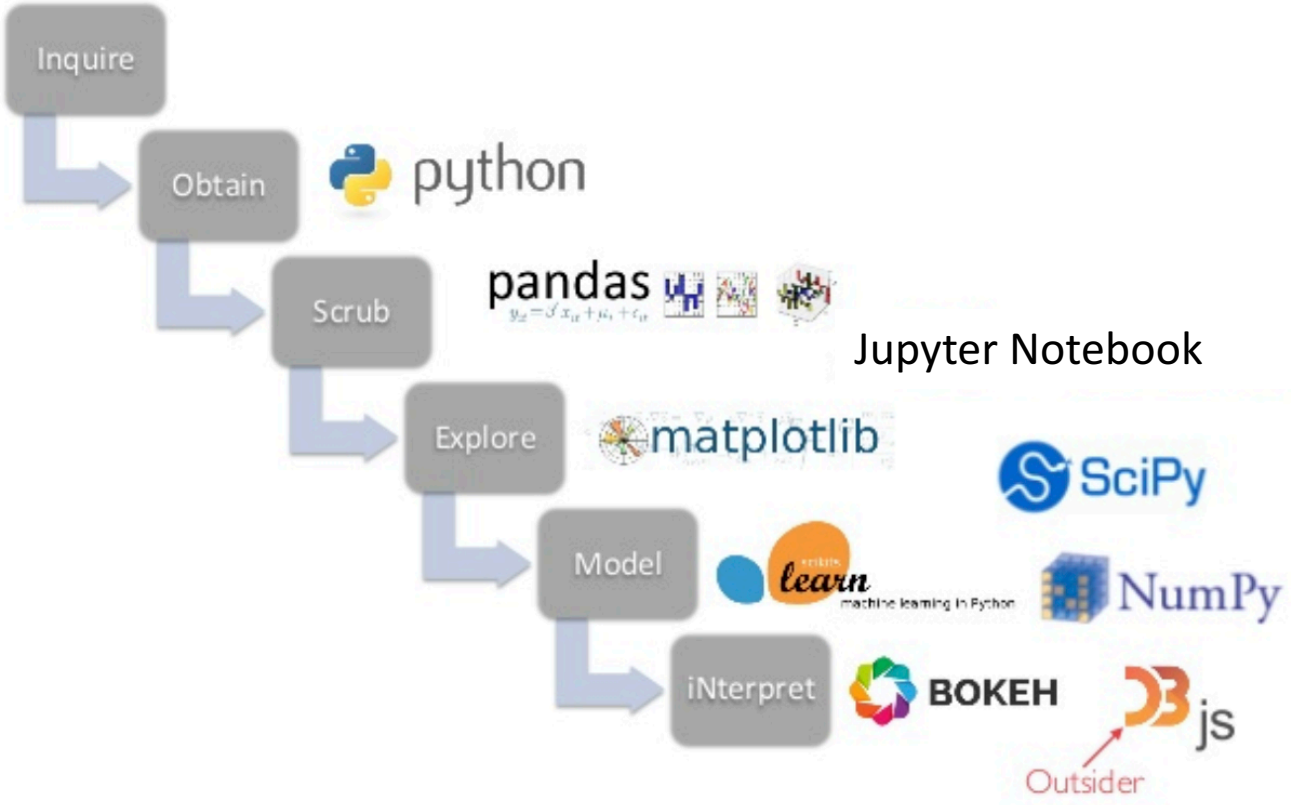
C++??

That’s over 100 lines!!

Data Science is **IOSEMN**



Python is **IOSEMN**



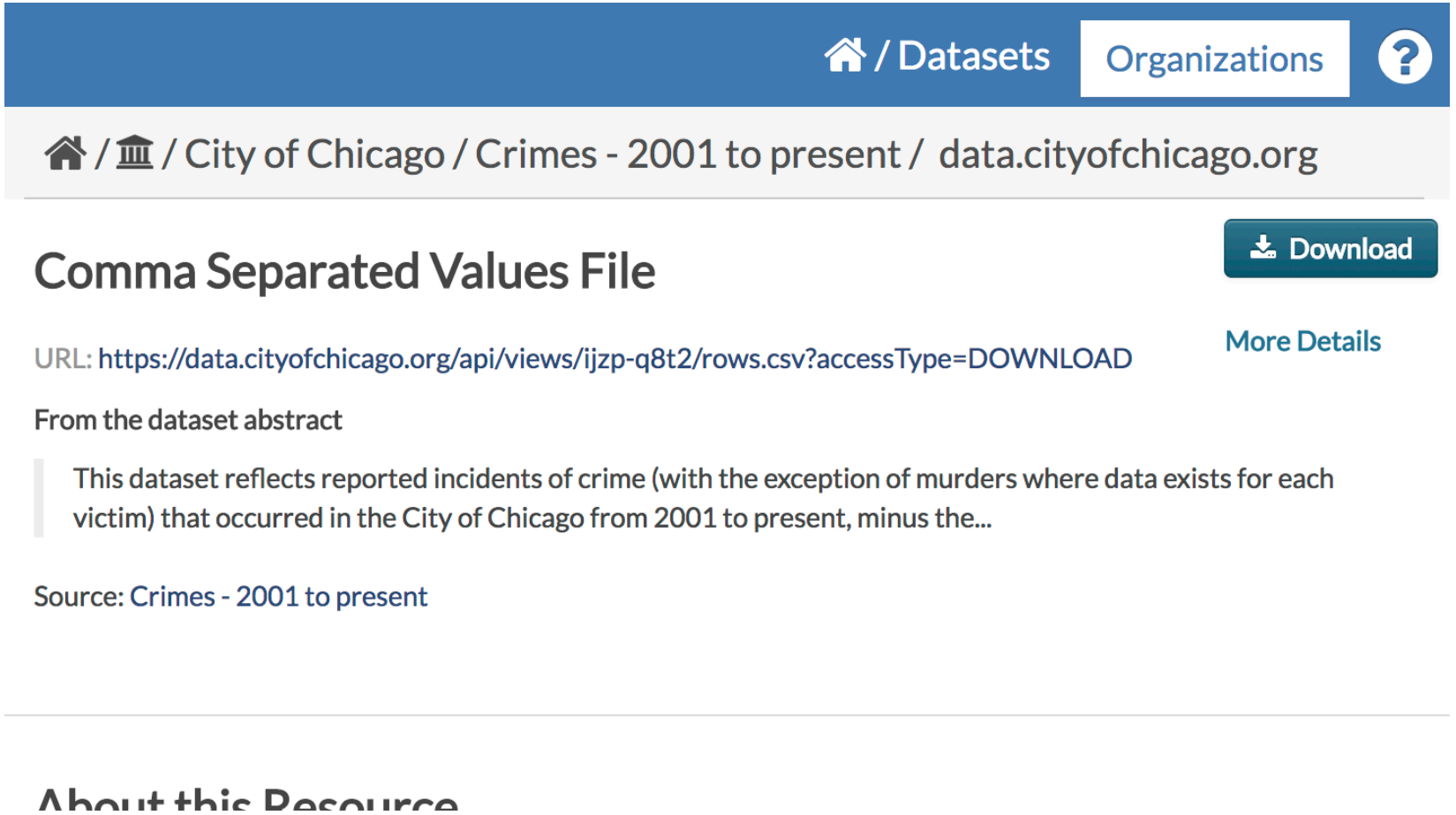
Inquire

1. Which communities are more popular?
2. Is the engagement of users in corporate communities increasing?
3. What is the distribution of posts publishing time, during the day?
4. What is the percentage of interactions (likes and comments)?
5. How is the likes distribution by user?
6. Is there a relationship between publishing hour and number of interactions?
7. What communities are more engaging (greater avg. interactions on posts)?
8. What are the most relevant words in the posts?
9. How to group posts about similar subjects?

Obtain

- Download data from another location (e.g., a web page or server)
- Query data from a database (e.g., MySQL or Oracle)
- Extract data from an API (e.g., Twitter, Facebook)
- Extract data from another file (e.g., an HTML file or spreadsheet)
- Generate data yourself (e.g., reading sensors or taking surveys)

https://catalog.data.gov/dataset/crimes-2001-to-present-398a4



Downloading data

import urllib.request url=https://data.cityofnewyork.us/api/views/5t4n-d72c/rows.csv?accessType=DOWNLOAD

print('Beginning file download with urllib2...')

urllib.request.urlretrieve(url, './data/homeless.csv')

Scrub: the world is a messy place

- Data cleaning (or scrubbing) necessary before analysis, reasons: missing data and inconsistent labels, or via a website with an awkward choice of data formatting.
- The least sexy part of the analysis process.
- The most basic form of scrubbing data is just making sure that it's read cleanly, stripped of extraneous characters, and parsed into a usable format.
- "New York, NY", "NYC", "New York City", "Manhattan, NY"

Explore: You can see a lot by looking

- Visualizing, clustering, performing dimensionality reduction: these are all part of `looking at data.'
- These tasks are sometimes described as "**exploratory**" in that no hypothesis is being tested, no predictions are attempted.
- Example tasks:
  - *more* or *less*. E.g. View the top 5 of data
  - Single-feature histograms
  - Feature-pair scatter plots
  - Dimensionality reduction
  - Clustering

Models: always bad, sometimes ugly

- Why we build models in the first place: to **predict** and to **interpret**.
- Is it `better' to fit one's data to a straight line or a fifth-order polynomial? Should one combine a weighted sum of 10 rules or 10,000?
- Model that gives the least errors tend to be our best predictive model.
- However, how can we interpret the model?

i**N**terpret: “The purpose of computing is insight, not numbers.”

- Here are conducted important activities, like:
  - Drawing conclusions from the data
  - Evaluating what your results mean
  - Communicating your result
- Data products are a very nice way to allow your users to browse the data and get their own conclusions. Python's Bokeh project is really nice for building interactive data products. You can also export your results to JSON and present them in the great D3.js

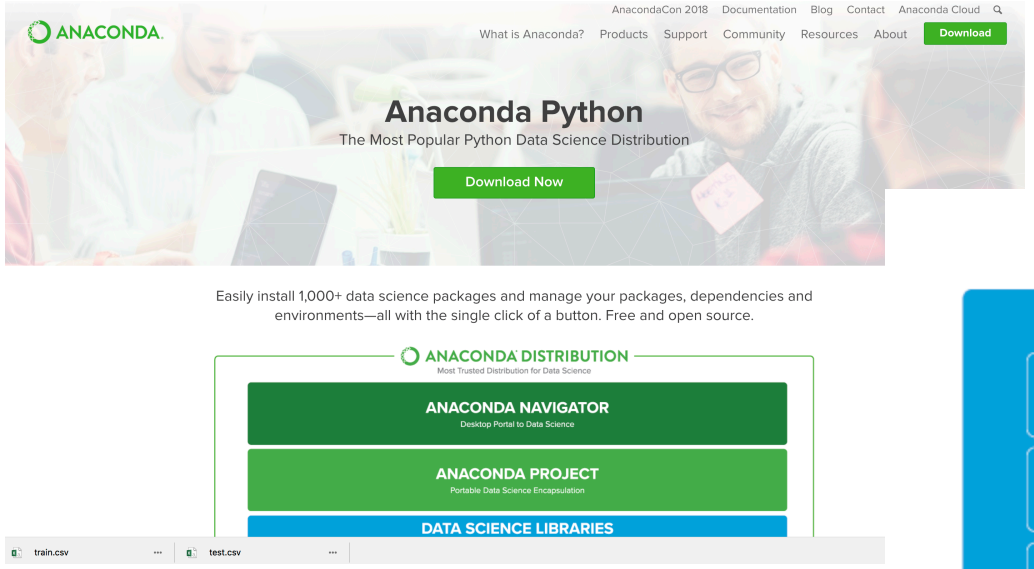
DEMONSTRATION

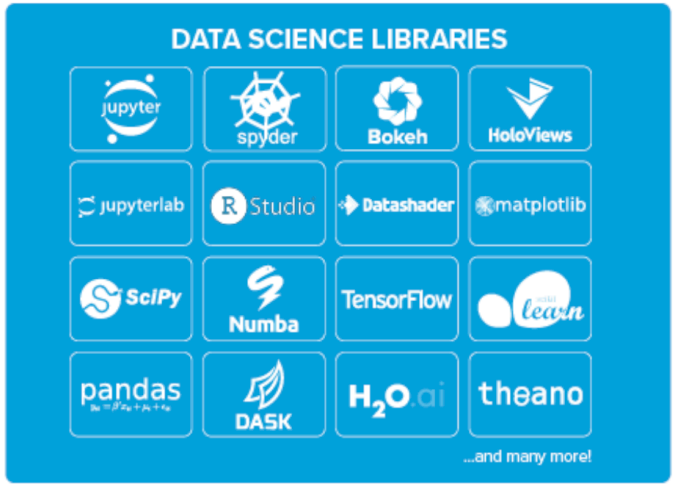
Setup

In this demonstration, we are going to use Python and the required libraries.

If you are new to Python, I recommend for you to download and install Anaconda (https://www.anaconda.com/distribution/), a Python distributor, with readily data science packages such as jupyter notebook, pandas, matplotlib, scikit-learn.

Setup



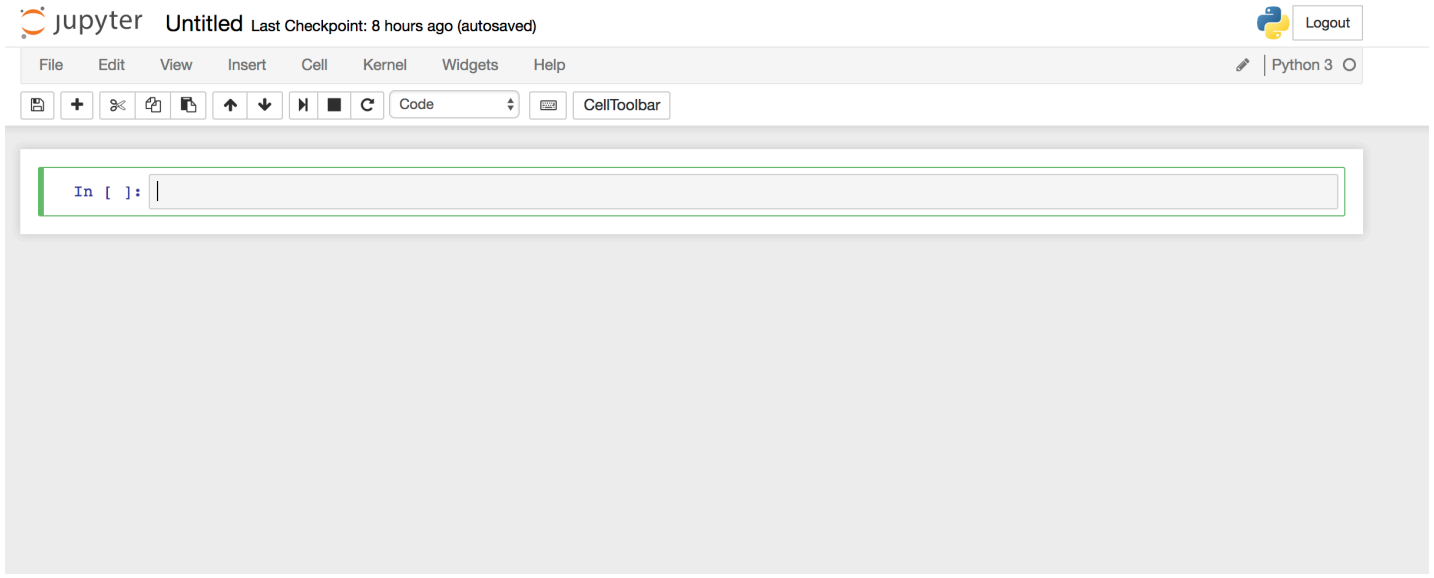


Anaconda Data Science Libraries

- Over 1,000 Anaconda-curated and community data science packages
- Develop data science projects using your favorite IDEs, including Jupyter, JupyterLab, Spyder and RStudio
- Analyze data with scalability and performance with Dask, numpy, pandas and Numba
- Visualize your data with Matplotlib, Bokeh, Datashader and Holoviews
- Create machine learning and deep learning models with Scikit-learn, Tensorflow, h20 and Theano

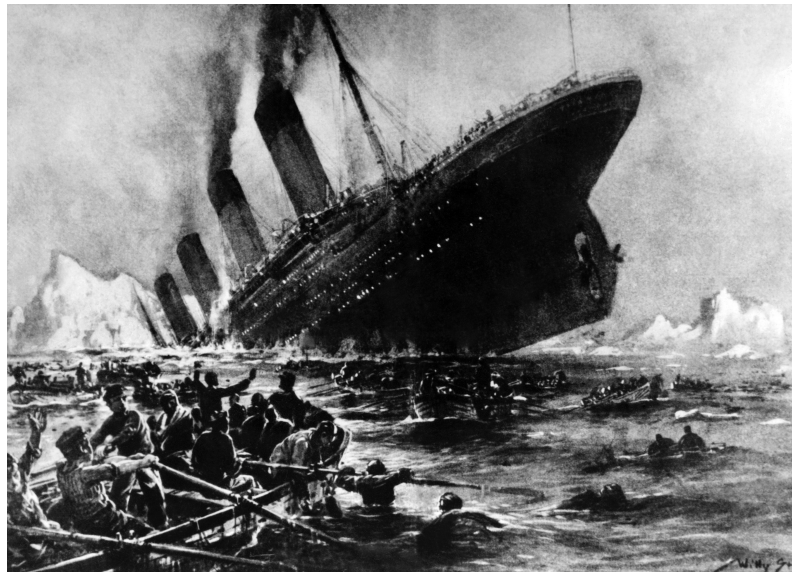
Jupyter Notebook

Notebook documents (or “notebooks”, all lower case) are documents produced by the Jupyter Notebook App, which contain both computer code (e.g. python) and rich text elements (paragraph, equations, figures, links, etc…). Notebook documents are both human-readable documents containing the analysis description and the results (figures, tables, etc..) as well as executable documents which can be run to perform data analysis.



Titanic: Predicting the Survivor

The sinking of the RMS Titanic is one of the most infamous shipwrecks in history. On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew. This sensational tragedy shocked the international community and led to better safety regulations for ships.



One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew. Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others, such as women, children, and the upper-class.

In this challenge, we ask you to complete the analysis of **what sorts of people were likely to survive.**

Adapted from kaggle.com

Inquiry

What sorts of people were likely to survive?

Obtain

Download the data, either from:

1) Kaggle: https://www.kaggle.com/c/titanic/data

2) My github repo: https://github.com/arachnie/hackathon-ericsson

You can directly download, or use Python command to download.

Example:

```
import urllib.request
url='https://raw.githubusercontent.com/arachnie/hackathon-ericsson/master/test.csv’
print('Beginning file download with urllib2...')
urllib.request.urlretrieve(url, 'test.csv')
```

Obtain

- For easy manipulation, in Python, we use Pandas library and save the data as Dataframe.
- DataFrame is a way to store data in grids than be easily overviewed
- Each row of these grids corresponds to measurements or values of an instance, while each column is a vector containing data for a specific variable. This means that a data frame's rows do not need to contain, but can contain, the same type of values: they can be numeric, character, logical, etc.

```
import pandas as pd
train = pd.read_csv("train.csv")
test   = pd.read_csv("test.csv")
titanic= train.append( test , ignore_index = True )
```

Scrub and Explore

- We need to understand the data, thus we need to get the variables details. For these dataset, it can be viewed at: https://www.kaggle.com/c/titanic/data

Data Dictionary

Variable	Definition	Key
survival	Survival	0 = No, 1 = Yes
pclass	Ticket class	1 = 1st, 2 = 2nd, 3 = 3rd
sex	Sex	
Age	Age in years	
sibsp	# of siblings / spouses aboard the Titanic	
parch	# of parents / children aboard the Titanic	
ticket	Ticket number	
fare	Passenger fare	
cabin	Cabin number	
embarked	Port of Embarkation	C = Cherbourg, Q = Queenstown, S = Southampton

Scrub and Explore

- For scrubbing, common tasks are: finding missing values, duplication.

titanic.head()

	Age	Cabin	Embarked	Fare	Name	Parch	PassengerId	Pclass	Sex	SibSp	Survived	Ticket
0	22.0	NaN	S	7.2500	Braund, Mr. Owen Harris	0	1	3	male	1	0.0	A/5 21171
1	38.0	C85	C	71.2833	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	2	1	female	1	1.0	PC 17599
2	26.0	NaN	S	7.9250	Heikkinen, Miss. Laina	0	3	3	female	0	1.0	STON/O2. 3101282
3	35.0	C123	S	53.1000	Futrelle, Mrs. Jacques Heath (Lily May Peel)	0	4	1	female	1	1.0	113803
4	35.0	NaN	S	8.0500	Allen, Mr. William Henry	0	5	3	male	0	0.0	373450

Scrub and **E**xplore

- Find duplications.

```
titanic[titanic.duplicated(keep=False)]
```

Scrub and Explore

To view the data in overall, we can use the *describe* function from pandas to calculate summary statistics of your data.

titanic.describe()

	Age	Fare	Parch	PassengerId	Pclass	SibSp	Survived
count	714.000000	891.000000	891.000000	891.000000	891.000000	891.000000	891.000000
mean	29.699118	32.204208	0.381594	446.000000	2.308642	0.523008	0.383838
std	14.526497	49.693429	0.806057	257.353842	0.836071	1.102743	0.486592
min	0.420000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000
25%	20.125000	7.910400	0.000000	223.500000	2.000000	0.000000	0.000000
50%	28.000000	14.454200	0.000000	446.000000	3.000000	0.000000	0.000000
75%	38.000000	31.000000	0.000000	668.500000	3.000000	1.000000	1.000000
max	80.000000	512.329200	6.000000	891.000000	3.000000	8.000000	1.000000

More on Scrubbing…

- See which columns have missing values in training data:

```
titanic.isnull().sum()
```

```
Age            263
Cabin         1014
Embarked         2
Fare             1
Name             0
Parch            0
PassengerId      0
Pclass           0
Sex              0
SibSp            0
Survived       418
Ticket           0
dtype: int64
```

Scrub

There are 2 missing values for the port that the passenger embarked on in training data. Let's have a look at these two passengers:

```
titanic[pd.isnull(titanic['Embarked'])]
```

	Age	Cabin	Embarked	Fare	Name	Parch	PassengerId	Pclass	Sex	SibSp	Survived	Ticket
61	38.0	B28	NaN	80.0	Icard, Miss. Amelie	0	62	1	female	0	1.0	113572
829	62.0	B28	NaN	80.0	Stone, Mrs. George Nelson (Martha Evelyn)	0	830	1	female	0	1.0	113572

Both are first class passengers, that paid £80. Let's have a look if we can find similar passengers:

```
titanic[(titanic['Pclass'] == 1) & (titanic['Fare'] > 70) & (titanic['Fare'] < 90)]
```

	Age	Cabin	Embarked	Fare	Name	Parch	PassengerId	Pclass	Sex	SibSp	Survived	Ticket
1	38.0	C85	C	71.2833	Cumings, Mrs. John Bradley (Florence Briggs Th...	0	2	1	female	1	1.0	PC 17599
34	28.0	NaN	C	82.1708	Meyer, Mr. Edgar Joseph	0	35	1	male	1	0.0	PC 17604
52	49.0	D33	C	76.7292	Harper, Mrs. Henry Sleeper (Myna Haxtun)	0	53	1	female	1	1.0	PC 17572
61	38.0	B28	NaN	80.0000	Icard, Miss. Amelie	0	62	1	female	0	1.0	113572
62	45.0	C83	S	83.4750	Harris, Mr. Henry Birkhardt	0	63	1	male	1	0.0	36973
102	21.0	D26	S	77.2875	White, Mr. Richard Frasar	1	103	1	male	0	0.0	35281
124	54.0	D26	S	77.2875	White, Mr. Percival Wayland	1	125	1	male	0	0.0	35281
139												PC

Scrub

```
data =titanic[(titanic['Pclass'] == 1) & (titanic['Fare'] > 70) & (titanic['Fare'] < 90)]
```

```
data['Embarked'].value_counts()
```

```
C    29
S    23
```

Fill the 2 missing values on Embarked as C

```
titanic.iloc[:, 2] = titanic.iloc[:, 2].fillna('C')
```

More Scrubbing and Exploring

Fill in other missing values

```
Age            263
Cabin         1014
Embarked         2
Fare             1
Name             0
Parch            0
PassengerId      0
Pclass           0
Sex              0
SibSp            0
Survived       418
Ticket           0
dtype: int64
```

Strategies to fill in missing values:

1. Mode values (most frequent)
2. Mean values
3. Remove it

Explore

- We want to investigate which column (feature) that is most relevant to the surviving rate.

Let's have a look if Passenger Class had an influence on survival rate:

```
X_train[["Pclass", "Survived"]].groupby(['Pclass'], as_index=False).mean()
```

	Pclass	Survived
0	1	0.629630
1	2	0.472826
2	3	0.242363

Passenger class seems to be another good predictor of survival rate, where 1st class passengers were most likely to survive and 3rd class passengers - least likely.

Explore

- How about other features? Such as: Sex, Fare, Age?

Model

- Let say we have decided that the most relevant features are:

Sex	Pclass	Fare	Age

Model

We can start apply classification models on the selected features that we have identified: Sex, Pclass, Fare, Age

```
 # Modelling Algorithms
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC, LinearSVC
from sklearn.cross_validation import train_test_split , StratifiedKFold

# Divide the dataset into training, validation and testing data
train_valid_X = full_X[ 0:891 ]
train_valid_y = titanic.Survived[0:891]
test_X = full_X[ 891: ]
train_X , valid_X , train_y , valid_y = train_test_split( train_valid_X , train_valid_y , train_size = .7 )
```

Model

- Calculate the performance:

Model Performance

```
print (model.score( train_X , train_y ) , model.score( valid_X , valid_y ))
```

```
0.983948635634 0.809701492537
```

iNterpret

- Drawing conclusions from your data
- Evaluating what your results mean
- Communicating your result

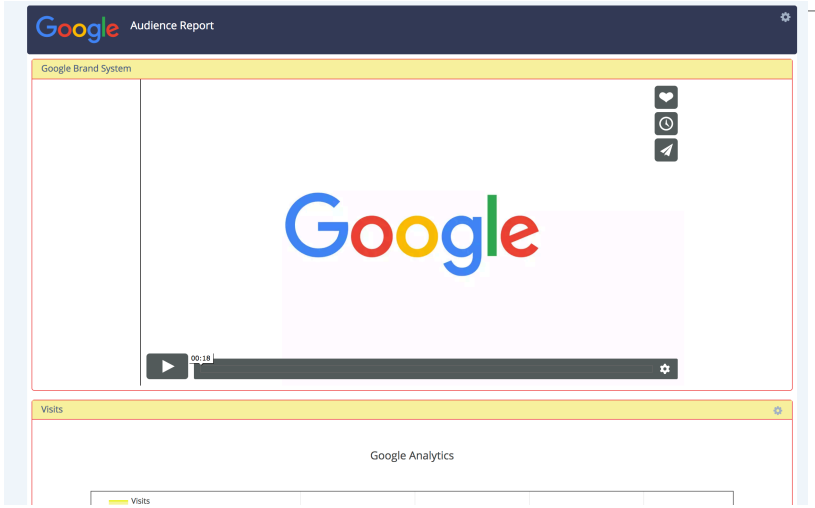
Developing Data Products

So what is a data product? A data product is any App, or software, or presentation, or reproducible report based on data or something that helps people analyze a common form of data.

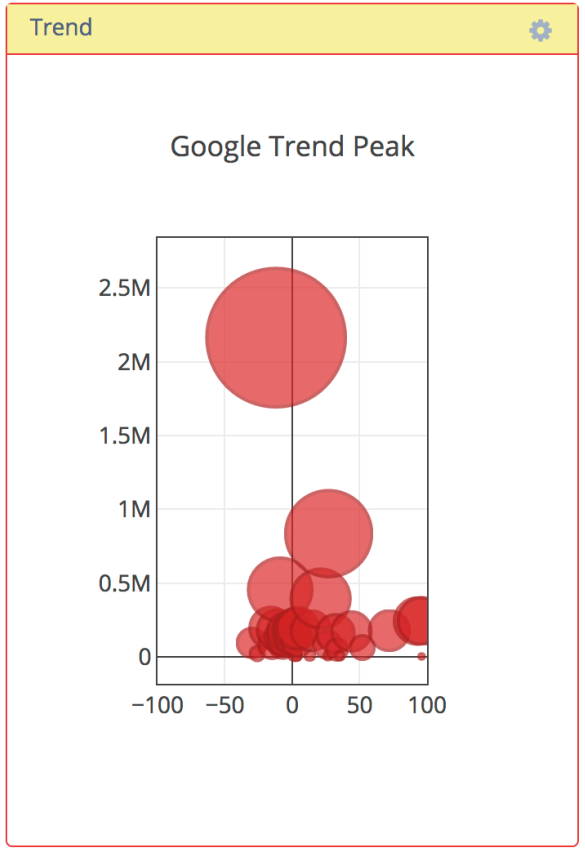
2 approaches:

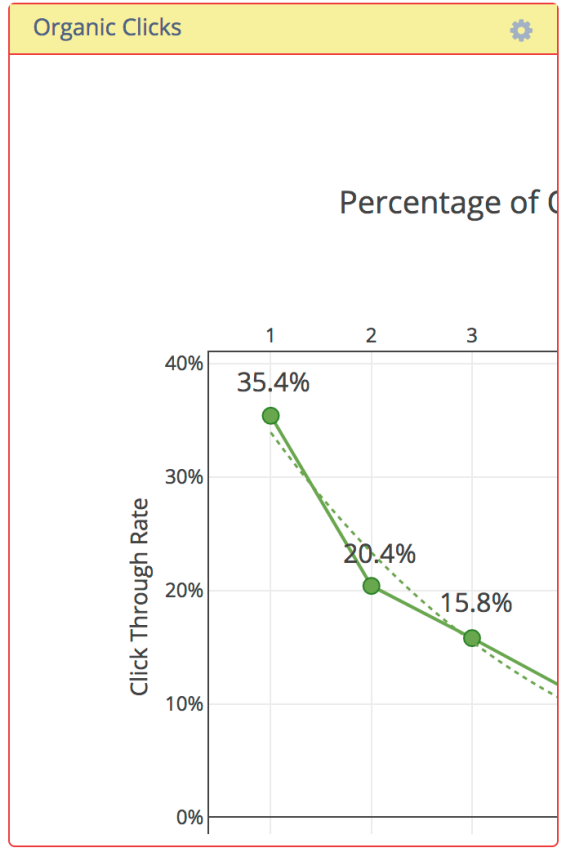
- Apps for creating data products: Tableau, Qlik, PowerBI, Plotly Dashboards
- Coding using Python, R, C++, Java

Example of data products: Dashboard (web app)

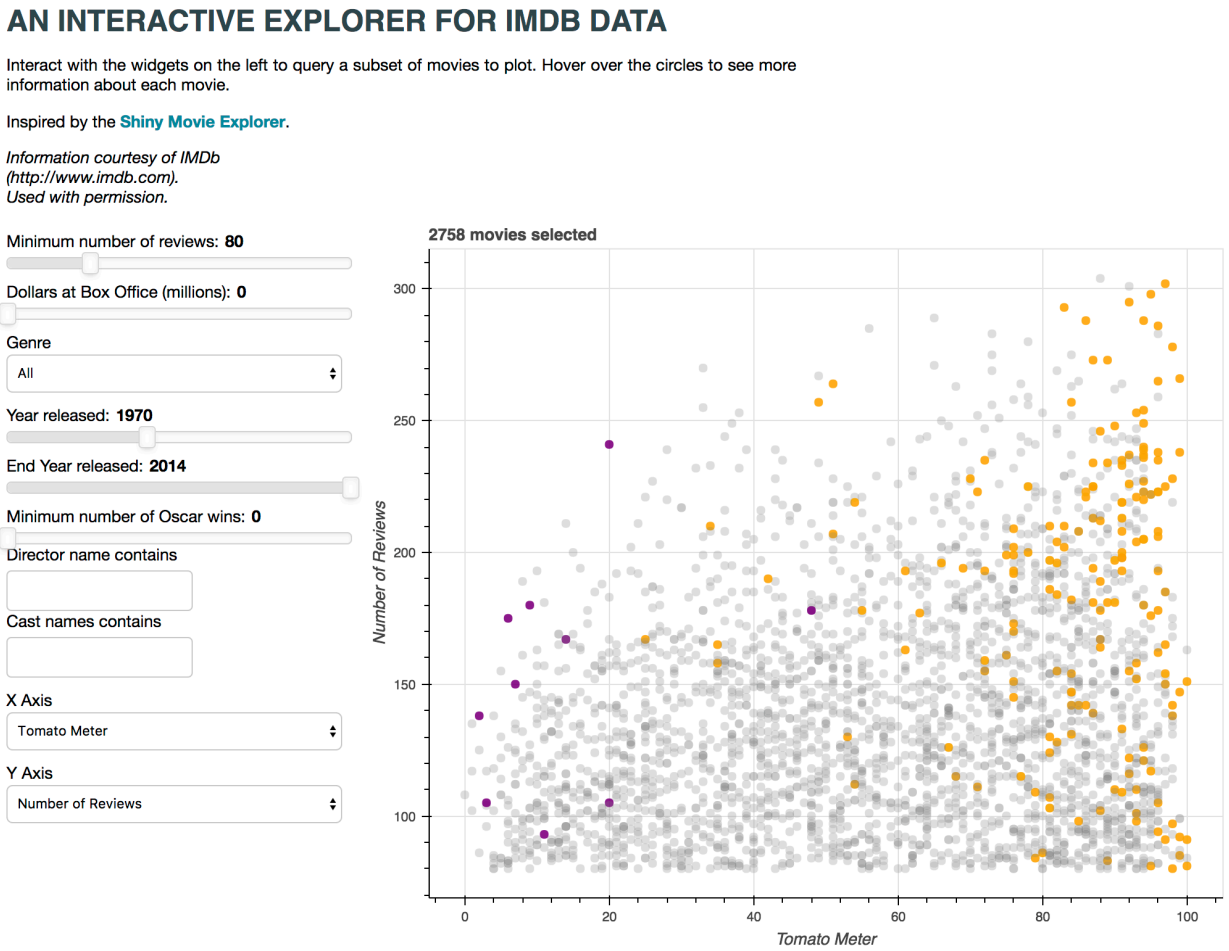


https://plot.ly/dashboard/plotly2_demo:237/view





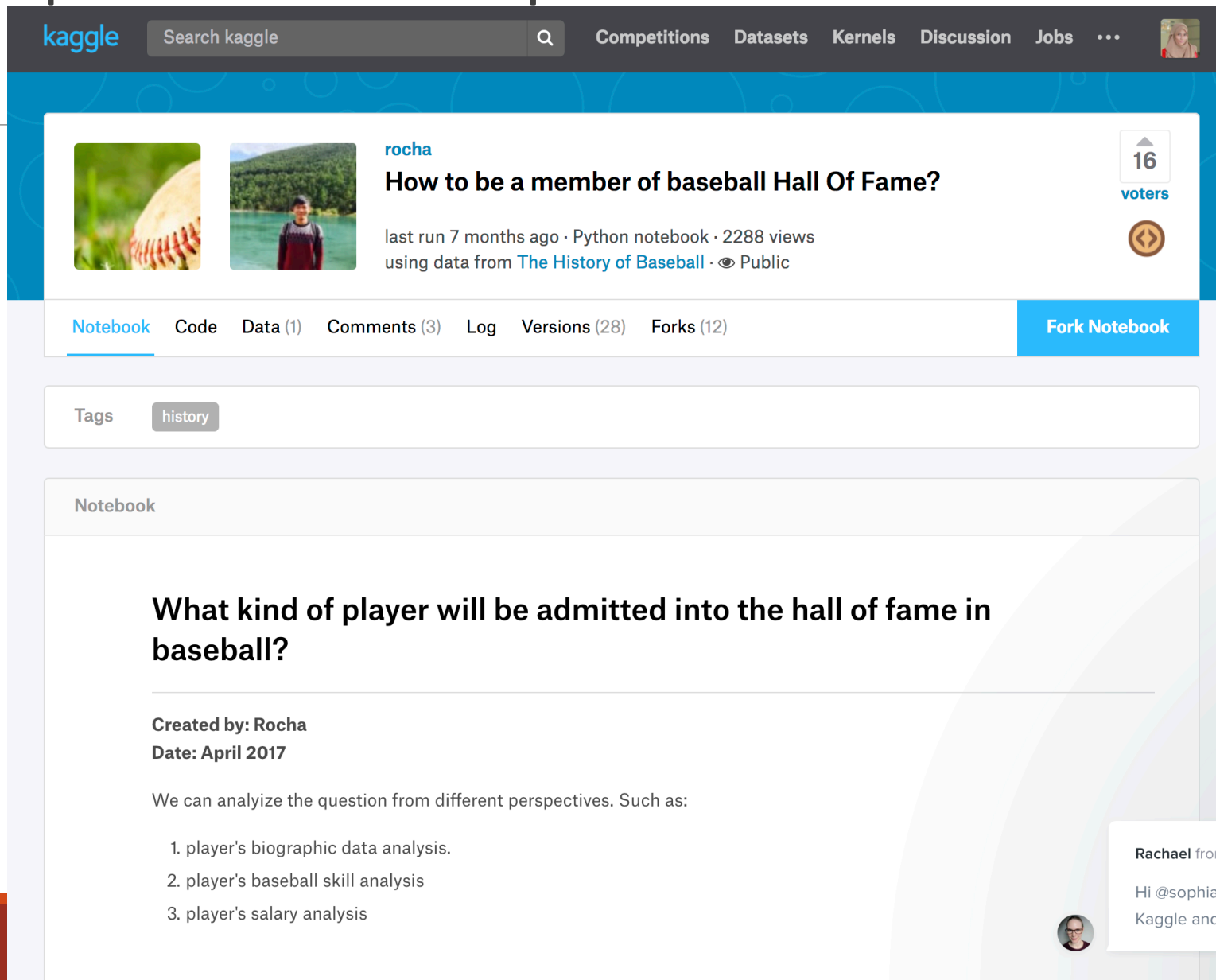
Example of data products: Interactive explorer (web app)



https://demo.bokehplots.com/apps/movies

Example of data products: reproducible report

https://www.kaggle.com/rochachan/how-to-be-a-member-of-baseball-hall-of-fame



Programming to create web-app data products

Python

- Shiny R

Python

- Plotly Dash (https://plot.ly/dash/introduction)

Creating Data Products Using Dash

Install dash (https://plot.ly/dash/installation)

Dash App Layout

https://plot.ly/dash/getting-started#dash-app-layout

Generating HTML with Dash

Dash comprises of 2 parts: layout and interactions

dash_core_components and the dash_html_components

Data Visualization in Dash

Markdown

Core Components

Calling help

Thank you!

sophia@utm.my

ais.utm.my/sophia