Gut Reaction

"Hey! I have a bunch of time series!"

We have a bunch of… things!

"How are these things related?"

https://flic.kr/p/7VUJkB

Pearson Correlation

Covariance of two variables divided by the product of their standard deviations

$$r = \frac{\sum(X - \overline{X})(Y - \overline{Y})}{\sqrt{\sum(X - \overline{X})^2}\sqrt{\sum(Y - \overline{Y})^2}}$$

cloudera

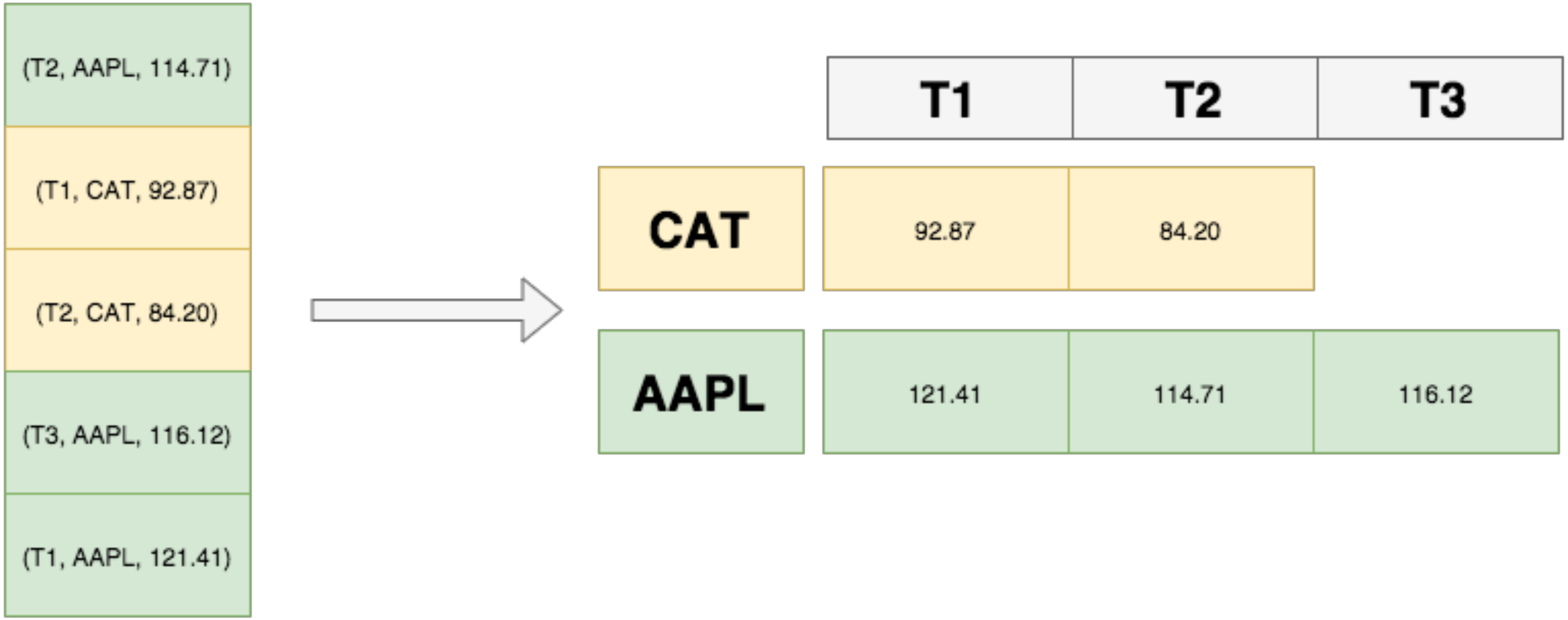
Create a Spark RDD of observations

```
In [33]: wiki_df.head(5)

Out[33]: [Row(timestamp=datetime.datetime(2015, 8, 2, 17, 0), page=u'Morgan_Stanley', views=32.0),
          Row(timestamp=datetime.datetime(2015, 8, 2, 23, 0), page=u'Ford_Motor_Company', views=91.0),
          Row(timestamp=datetime.datetime(2015, 8, 3, 13, 0), page=u'Mattel', views=24.0),
          Row(timestamp=datetime.datetime(2015, 8, 3, 13, 0), page=u'Visa_Inc.', views=36.0),
          Row(timestamp=datetime.datetime(2015, 8, 3, 13, 0), page=u'Yum!_Brands', views=36.0)]
```

```
In [32]: ticker_df.head(5)

Out[32]: [Row(timestamp=datetime.datetime(2015, 9, 1, 6, 0), symbol=u'DOV', price=60.38130930615455),
          Row(timestamp=datetime.datetime(2015, 8, 7, 9, 0), symbol=u'BLL', price=68.8183106049063),
          Row(timestamp=datetime.datetime(2015, 9, 15, 6, 0), symbol=u'MDLZ', price=42.369310917713
         31),
          Row(timestamp=datetime.datetime(2015, 8, 24, 12, 0), symbol=u'EMC', price=23.828349188493 72
         5),
          Row(timestamp=datetime.datetime(2015, 8, 20, 11, 0), symbol=u'HRS', price=80.573756374344 6
         1)]
```

cloudera

A time series is more than a pile of observations.



cloudera

Spark-Timeseries

Spark-Timeseries

- Library for manipulating and analyzing large-scale time series data
- Github: https://github.com/cloudera/spark-timeseries
- Doc: http://cloudera.github.io/spark-timeseries/

cloudera

Observations

Timestamp	Symbol	Price
2015-04-10	AAPL	2.0
2015-04-11	AAPL	3.0
2015-04-10	MSFT	4.5
2015-04-11	MSFT	1.5
2015-04-10	GOOG	6.0

```
schema = StructType([
    StructField('timestamp', TimestampType()), \
    StructField('symbol', StringType()), \
    StructField('price', DoubleType()) \
])
ticker_obs = sqlCtx.createDataFrame(row_rdd,
schema)
```

Time samples

Timestamp	AAPL	MSFT	GOOG
2015-04-10	2.0	4.5	6.0
2015-04-11	3.0	1.5	NaN

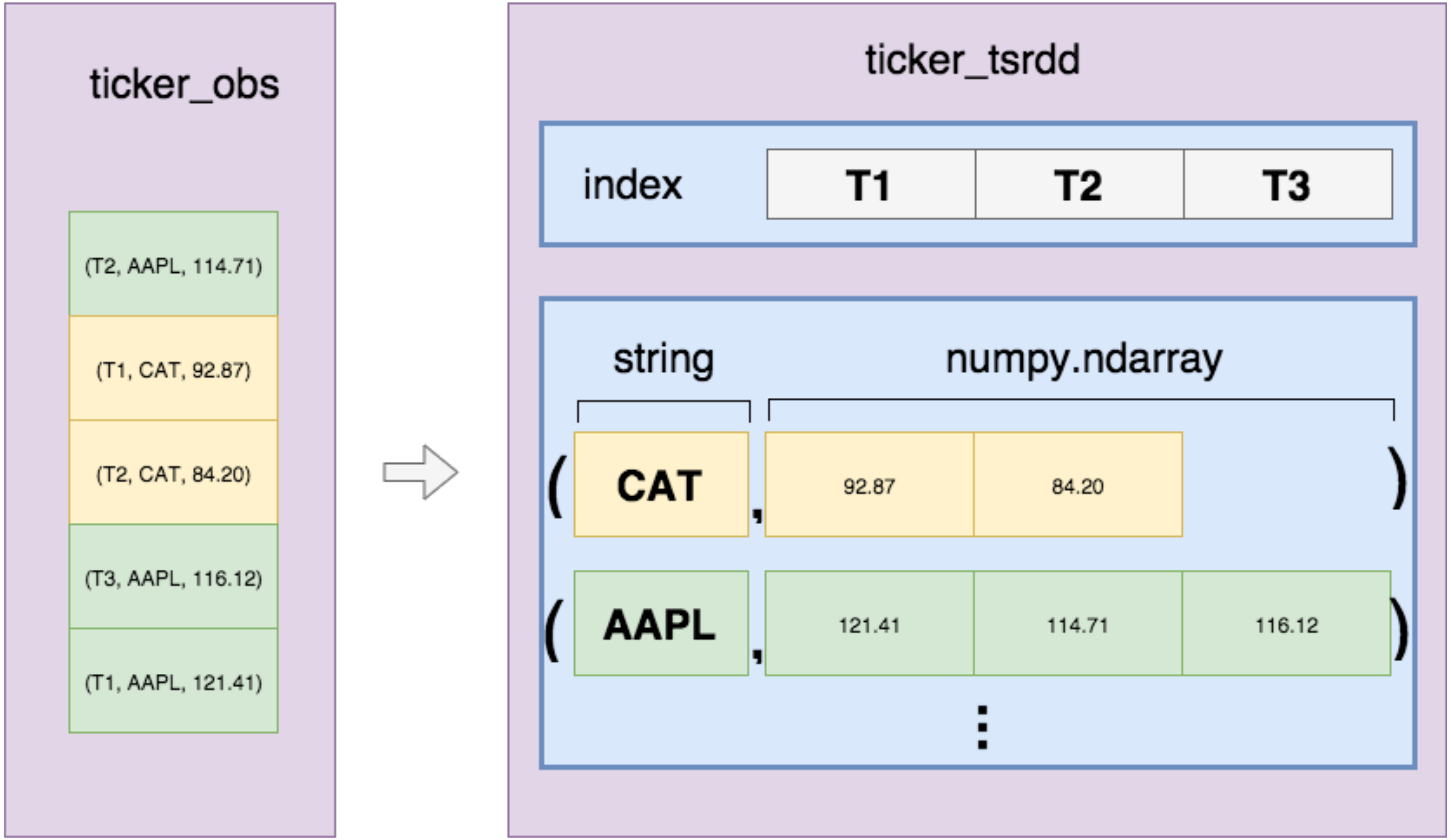
cloudera

Time series

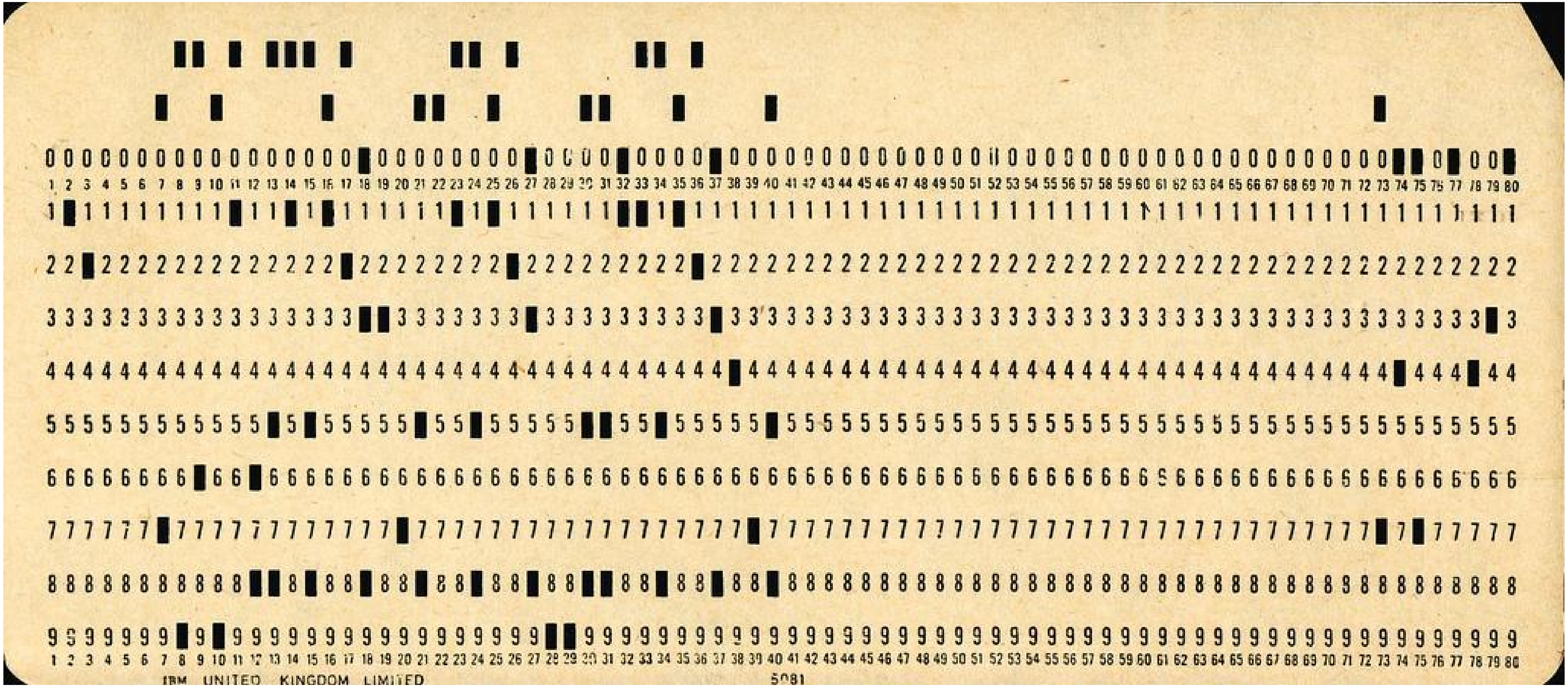
DateTimeIndex: [2015-04-10, 2015-04-11]	
Symbol	**Series**
AAPL	[2.0, 3.0]
MSFT	[4.5, 1.5]
GOOG	[6.0, NaN]

cloudera

Transform observations into time series



Time to code



https://flic.kr/p/8RcFUD

Apple

https://flic.kr/p/8d4XB3

What's the correlation between AAPL and en.wikipedia.org/wiki/Apple_Inc.?

Wiki Page:

Apple Inc.

From Wikipedia, the free encyclopedia

Coordinates: 37.33182°N 122.03118°W

This article is about the technology company. For other companies named "Apple", see Apple (disambiguation).

Apple Inc. (commonly known as **Apple**) is an American multinational technology company headquartered in Cupertino, California, that designs, develops, and sells consumer electronics, computer software, and online services. Its best-known hardware products are the Mac personal computers, the iPod portable media player, the iPhone smartphone, the iPad tablet computer, and the Apple Watch smartwatch. Apple's consumer software includes the OS X and iOS operating systems, the iTunes media player, the Safari web browser, and the iLife and iWork creativity and productivity suites. Its online services include the iTunes Store, the iOS App Store and Mac App Store, and iCloud.

Apple was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne on April 1, 1976, to develop and sell personal computers.[5] It was incorporated as **Apple Computer, Inc.** on January 3, 1977, and was renamed as Apple Inc. on January 9, 2007, to reflect its shifted focus toward consumer electronics. Apple (NASDAQ: AAPL) joined the Dow Jones Industrial Average on March 19, 2015.[6]

Apple is the world's second-largest information technology company by revenue after Samsung Electronics, the world's largest technology company by total assets, and the world's third-largest mobile phone manufacturer. On November 25, 2014, in addition to being the largest publicly traded corporation in the world by market capitalization, Apple became the first U.S. company to be valued at over US$700 billion.[7] As of July 2015, Apple employs 115,000 permanent full-time employees;[4] maintains 453 retail stores in sixteen countries;[1] and operates the online Apple Store and iTunes Store, the latter of which is the world's largest music retailer.

Apple's worldwide annual revenue in 2014 totaled $182 billion for the fiscal year ending in October 2014.[8] The company enjoys a high level of brand loyalty and, according to the 2014 edition of the Interbrand Best Global Brands report, is the world's most valuable brand with a valuation of $118.9 billion.[9] By the end of 2014, the

Apple Inc.

Apple Campus (1 Infinite Loop, Cupertino, California)

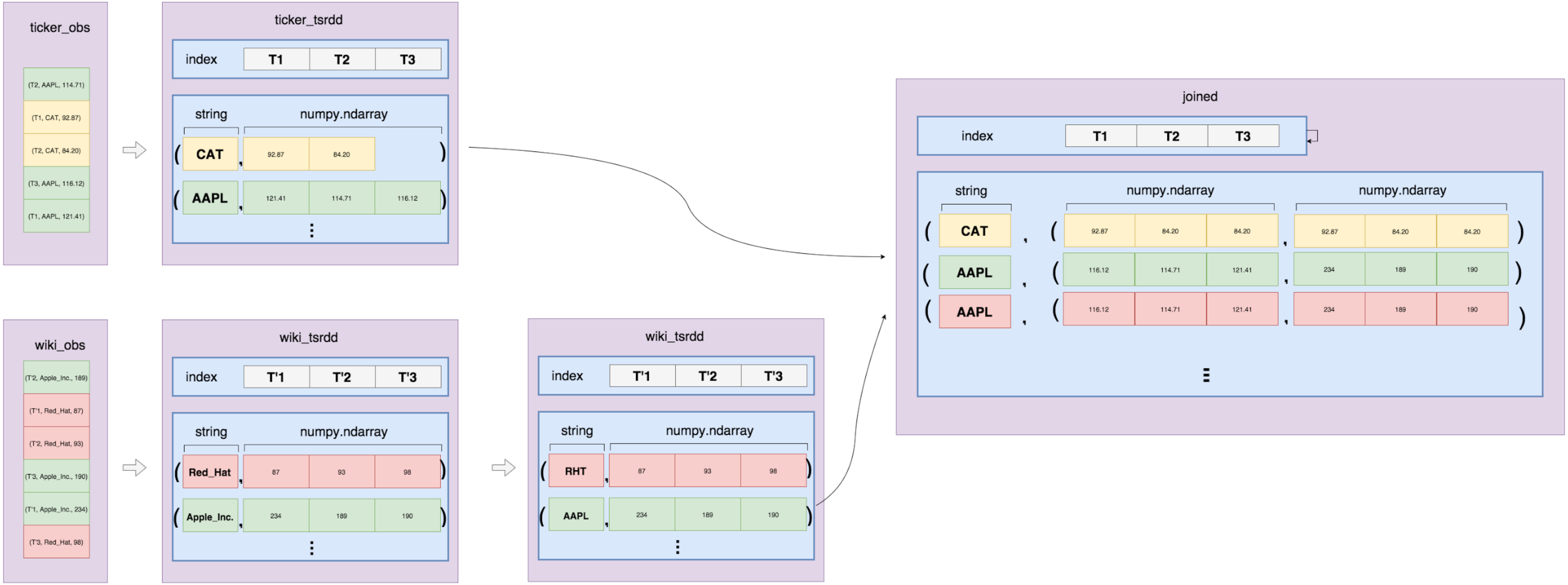
Type	Public
Traded as	NASDAQ: AAPL Dow Jones Industrial Average Component NASDAQ-100 Component S&P 500 Component

cloudera

Data Quality and NaNs

https://flic.kr/p/bnnHaP

Link symbols and pages



Ask focused questions

Volatility

Measure of variety or variation of a time series

https://flic.kr/p/dtWHy

Magic

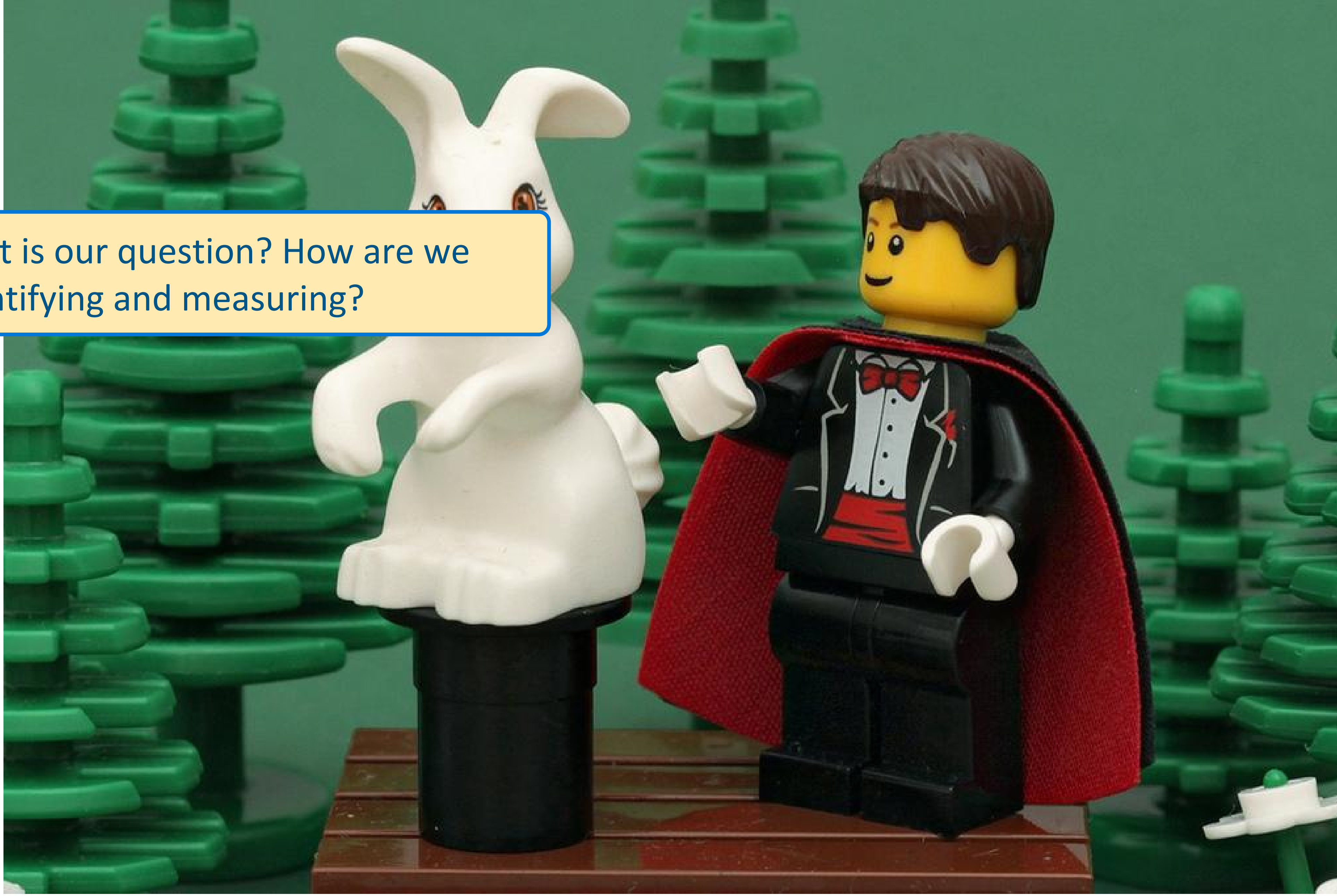
- Magnitude
- Articulation
- Generality
- Interestingness
- Credibility

https://flic.kr/p/a4h8WE

Magic

- Magnitude
- Articulation
- Generality
- Interestingness
- Credibility

What is our question? How are we quantifying and measuring?

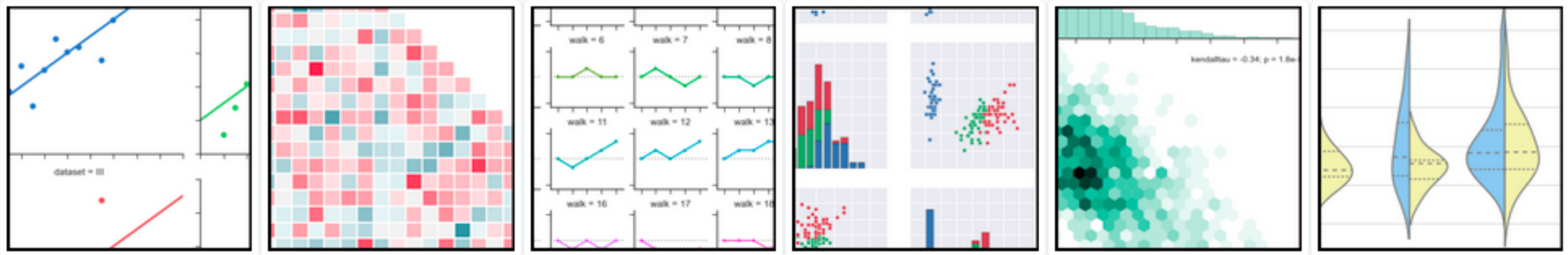


https://flic.kr/p/a4h8WE

Statistical Plotting: Seaborn

seaborn 0.6.0 API Tutorial Gallery Site ▾ Page ▾ Search

Seaborn: statistical data visualization



Seaborn is a Python visualization library based on matplotlib. It provides a high-level interface for drawing attractive statistical graphics.

For a brief introduction to the ideas behind the package, you can read the *introductory notes*. More practical information is on the *installation page*. You may also want to browse the *example gallery* to get a sense for what you can do with seaborn and then check out the *tutorial* and *API reference* to find out how.

To see the code or report a bug, please visit the github repository. General support issues are most at home on stackoverflow, where there is a seaborn tag.

Documentation

- An introduction to seaborn
- What's new in the package
- Installing and getting started
- Example gallery
- API reference
- Seaborn tutorial

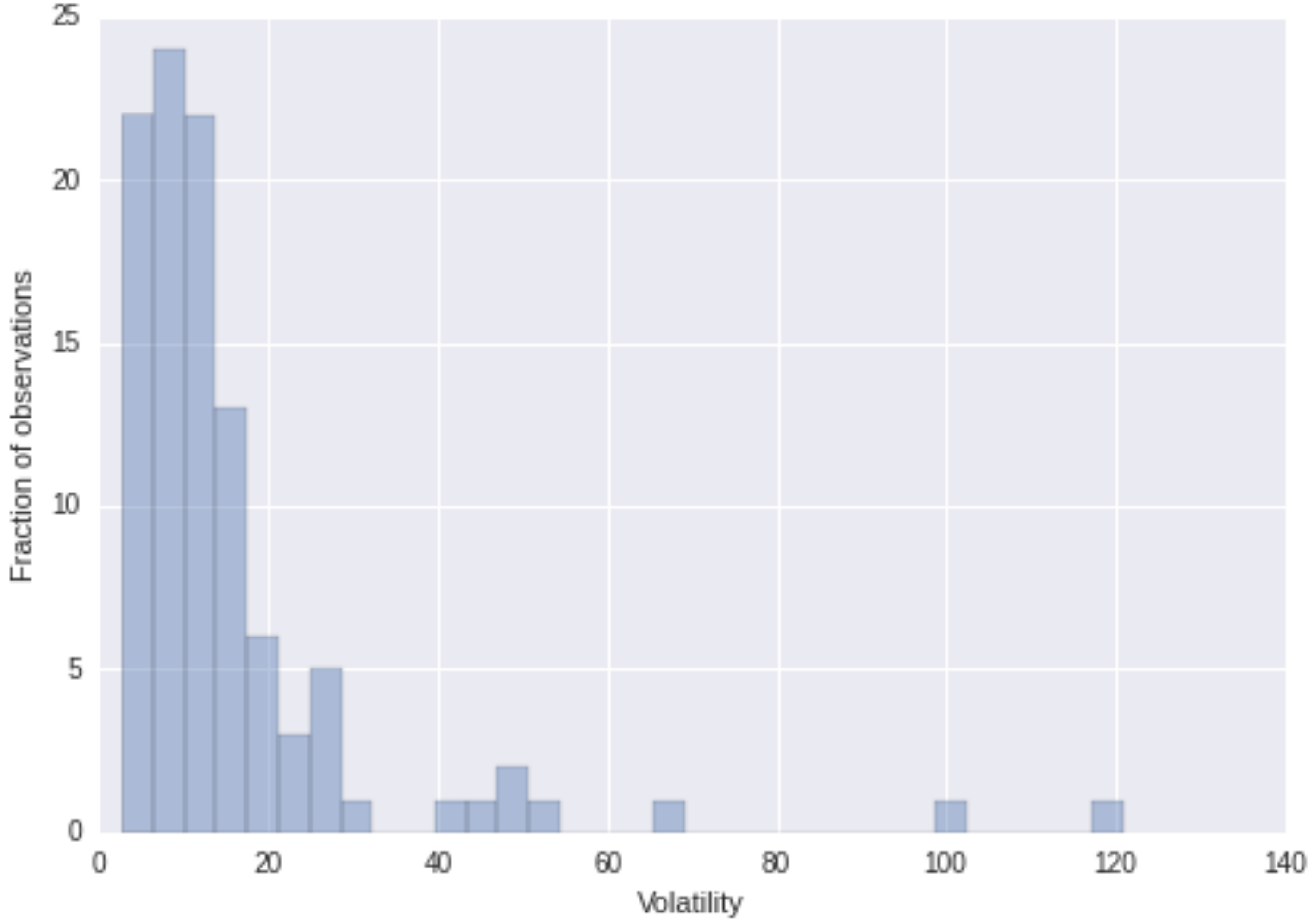
Features

- Style functions: *API* | *Tutorial*
- Color palettes: *API* | *Tutorial*
- Distribution plots: *API* | *Tutorial*
- Regression plots: *API* | *Tutorial*
- Categorical plots: *API* | *Tutorial*
- Axis grid objects: *API* | *Tutorial*

cloudera

http://stanford.edu/~mwaskom/software/seaborn/

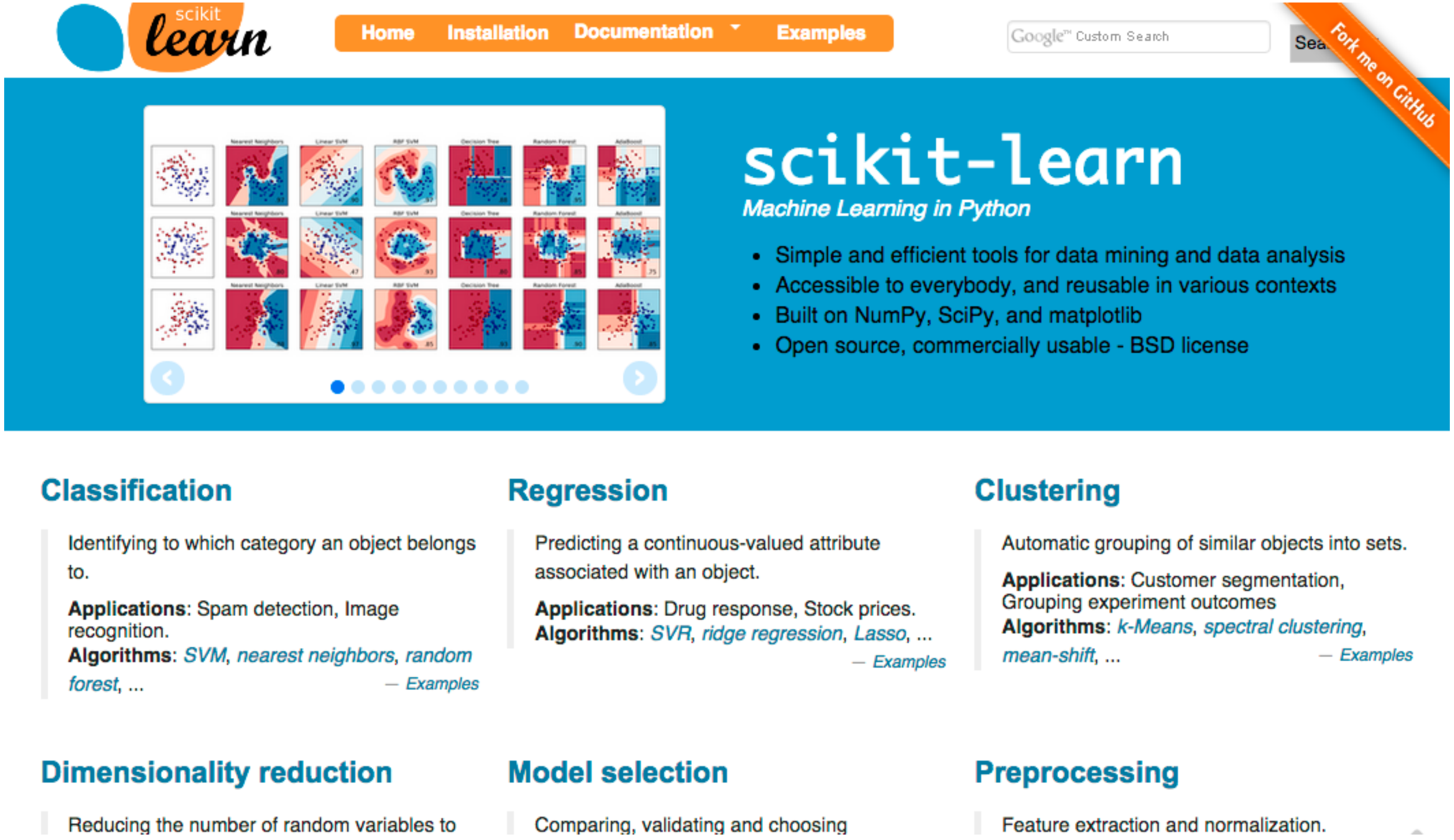
Distribution of Volatility



http://stanford.edu/~mwaskom/software/seaborn/

Regression as a measure of effect

Linear Regression using sklearn



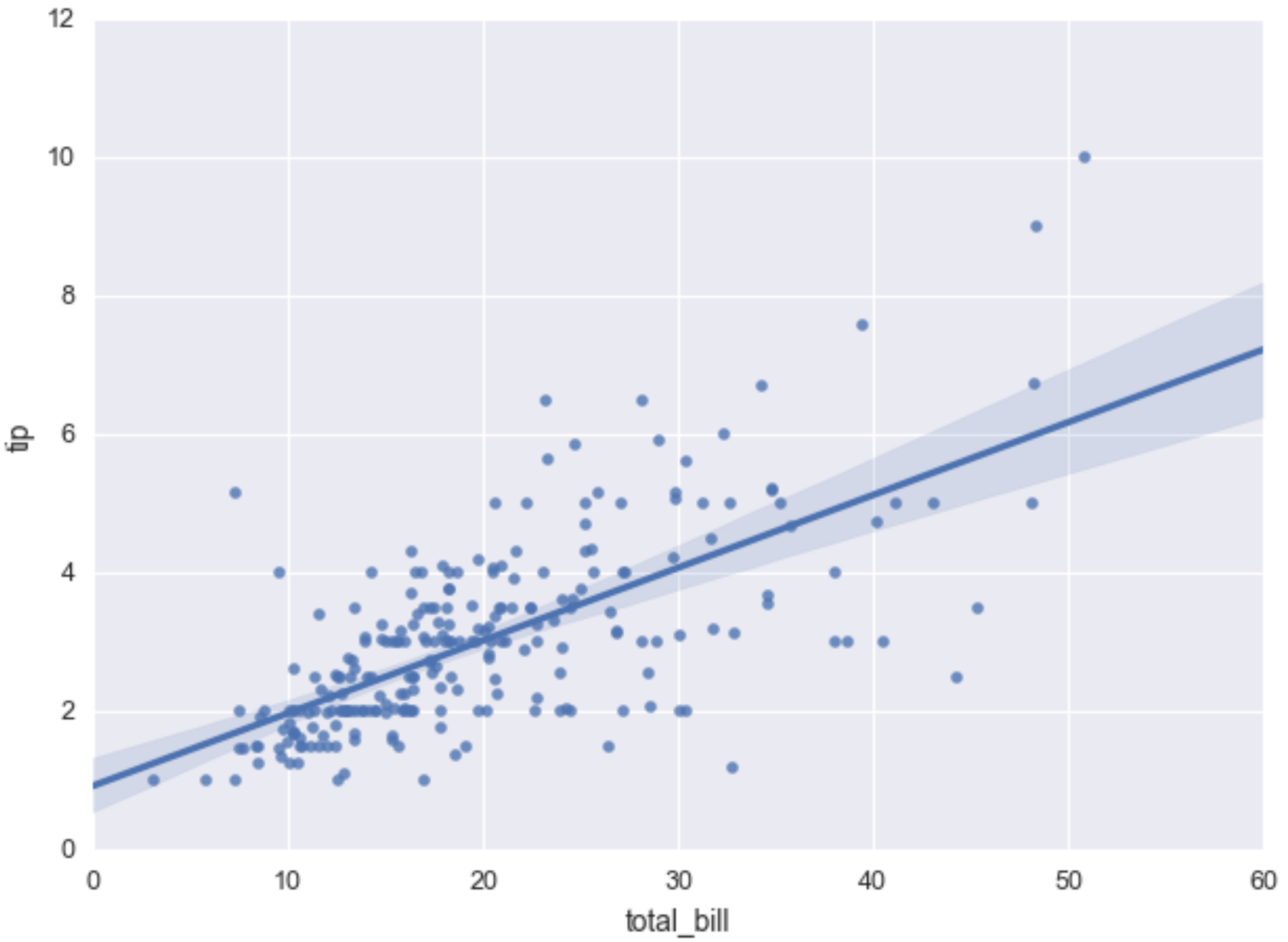
http://scikit-learn.org/stable/documentation.html

Linear Regression using sklearn

```
from sklearn import linear_model

def regress(X, y):
    model = linear_model.LinearRegression()
    model.fit(X, y)
    score = model.score(X, y)
    return (score, model)

lag = 2
lead = 2

joined = regressions = wiki_daily_views.flatMap(get_page_symbol) \
    .join(ticker_daily_vol)

models = joined.mapValues(lambda x: regress(lead_and_lag(lead, lag, x[0]), x[1][lag:-lead]))
models.cache()
models.count()
```

cloudera

Linear Regression using seaborn



Why regression when we have correlation?

https://flic.kr/p/8mNZ6r

If you need to fit a linear model over all of your data

pyspark.mllib.classification module

class `pyspark.mllib.classification.`**`LogisticRegressionModel`**(*weights*, *intercept*, *numFeatures*, *numClasses*) [source]

Classification model trained using Multinomial/Binary Logistic Regression.

Parameters:

- **weights** – Weights computed for every feature.
- **intercept** – Intercept computed for this model. (Only used in Binary Logistic Regression. In Multinomial Logistic Regression, the intercepts will not be a single value, so the intercepts will be part of the weights.)
- **numFeatures** – the dimension of the features.
- **numClasses** – the number of possible outcomes for k classes classification problem in Multinomial Logistic Regression. By default, it is binary logistic regression so numClasses will be set to 2.

```
>>> data = [
...     LabeledPoint(0.0, [0.0, 1.0]),
...     LabeledPoint(1.0, [1.0, 0.0]),
... ]
>>> lrm = LogisticRegressionWithSGD.train(sc.parallelize(data), iterations=10)
>>> lrm.predict([1.0, 0.0])
1
>>> lrm.predict([0.0, 1.0])
0
>>> lrm.predict(sc.parallelize([[1.0, 0.0], [0.0, 1.0]])).collect()
[1, 0]
>>> lrm.clearThreshold()
>>> lrm.predict([0.0, 1.0])
0.279...
```

cloudera

https://flic.kr/p/8mNZ6r

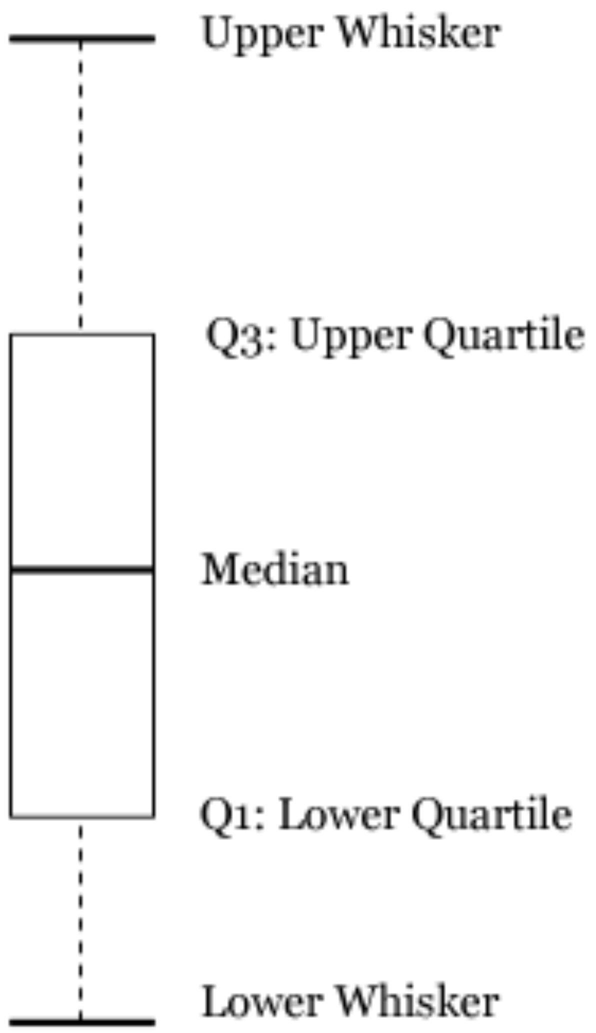
Identifying Outliers

When are “unusual” events occurring?

https://flic.kr/p/bn9tsC

Tukey’s criterion

- Tukey's boxplot criterion for outlier identification
  - Non-parametric: doesn’t need to assume particular probability distribution over daily purchases
  - *Robust to outliers*: focusing on interquartile range means we sidestep effects of anomalies



cloudera

Black Monday

August 24th, 2015

CNN Money | Business | Markets | Tech | Media | Personal Finance | Small Biz | Luxury | stock tickers

Investing Guide

Trading was halted 1,200 times Monday

Recommend 3k

By Matt Egan @mattmegan5

Black Monday vs. the Flash Crash and the SEC Response

NEWS

Black Monday vs. the Flash Crash and the SEC Response

Calder Lamb Sep 23, 2015

Share

TRADER TALK

What happened during the Aug 24 'flash crash'

Bob Pisani | @BobPisani

Friday, 25 Sep 2015 | 3:59 PM ET

CNBC

What happened?

Which stocks had the largest overnight drop
from end of August 23rd to open August 24th?

Which stocks saw the most volatility that day?

Hue: cdh4.srowen.com:8888 srowen/srowen
ipython: cdh1.srowen.com:8880 user11.ipynb
Spark: cdh1.srowen.com:4040 (redirects)
HDFS: /user/srowen/user11

Hue: cdh4.srowen.com:8888 srowen/srowen
ipython: cdh1.srowen.com:8880 user12.ipynb
Spark: cdh1.srowen.com:4041 (redirects)
HDFS: /user/srowen/user12

Hue: cdh4.srowen.com:8888 srowen/srowen
ipython: cdh1.srowen.com:8880 user13.ipynb
Spark: cdh1.srowen.com:4042 (redirects)
HDFS: /user/srowen/user13

Hue: cdh4.srowen.com:8888 srowen/srowen
ipython: cdh1.srowen.com:8880 user14.ipynb
Spark: cdh1.srowen.com:4043 (redirects)
HDFS: /user/srowen/user14

Hue: cdh4.srowen.com:8888 srowen/srowen
ipython: cdh1.srowen.com:8880 user15.ipynb
Spark: cdh1.srowen.com:4044 (redirects)
HDFS: /user/srowen/user15

Hue: cdh4.srowen.com:8888 srowen/srowen
ipython: cdh1.srowen.com:8880 user16.ipynb
Spark: cdh1.srowen.com:4045 (redirects)
HDFS: /user/srowen/user16

Hue: cdh4.srowen.com:8888 srowen/srowen
ipython: cdh1.srowen.com:8880 user17.ipynb
Spark: cdh1.srowen.com:4046 (redirects)
HDFS: /user/srowen/user17

Hue: cdh4.srowen.com:8888 srowen/srowen
ipython: cdh1.srowen.com:8880 user18.ipynb
Spark: cdh1.srowen.com:4047 (redirects)
HDFS: /user/srowen/user18

Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh2.srowen.com:8880 user21.ipynb* *Spark: cdh2.srowen.com:4040 (redirects)* *HDFS: /user/srowen/user21*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh2.srowen.com:8880 user22.ipynb* *Spark: cdh2.srowen.com:4041 (redirects)* *HDFS: /user/srowen/user22*
Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh2.srowen.com:8880 user23.ipynb* *Spark: cdh2.srowen.com:4042 (redirects)* *HDFS: /user/srowen/user23*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh2.srowen.com:8880 user24.ipynb* *Spark: cdh2.srowen.com:4043 (redirects)* *HDFS: /user/srowen/user24*
Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh2.srowen.com:8880 user25.ipynb* *Spark: cdh2.srowen.com:4044 (redirects)* *HDFS: /user/srowen/user25*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh2.srowen.com:8880 user26.ipynb* *Spark: cdh2.srowen.com:4045 (redirects)* *HDFS: /user/srowen/user26*
Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh2.srowen.com:8880 user27.ipynb* *Spark: cdh2.srowen.com:4046 (redirects)* *HDFS: /user/srowen/user27*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh2.srowen.com:8880 user28.ipynb* *Spark: cdh2.srowen.com:4047 (redirects)* *HDFS: /user/srowen/user28*

Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh3.srowen.com:8880 user31.ipynb* *Spark: cdh3.srowen.com:4040 (redirects)* *HDFS: /user/srowen/user31*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh3.srowen.com:8880 user32.ipynb* *Spark: cdh3.srowen.com:4041 (redirects)* *HDFS: /user/srowen/user32*
Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh3.srowen.com:8880 user33.ipynb* *Spark: cdh3.srowen.com:4042 (redirects)* *HDFS: /user/srowen/user33*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh3.srowen.com:8880 user34.ipynb* *Spark: cdh3.srowen.com:4043 (redirects)* *HDFS: /user/srowen/user34*
Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh3.srowen.com:8880 user35.ipynb* *Spark: cdh3.srowen.com:4044 (redirects)* *HDFS: /user/srowen/user35*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh3.srowen.com:8880 user36.ipynb* *Spark: cdh3.srowen.com:4045 (redirects)* *HDFS: /user/srowen/user36*
Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh3.srowen.com:8880 user37.ipynb* *Spark: cdh3.srowen.com:4046 (redirects)* *HDFS: /user/srowen/user37*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh3.srowen.com:8880 user38.ipynb* *Spark: cdh3.srowen.com:4047 (redirects)* *HDFS: /user/srowen/user38*

Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh4.srowen.com:8880 user41.ipynb* *Spark: cdh4.srowen.com:4040 (redirects)* *HDFS: /user/srowen/user41*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh4.srowen.com:8880 user42.ipynb* *Spark: cdh4.srowen.com:4041 (redirects)* *HDFS: /user/srowen/user42*
Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh4.srowen.com:8880 user43.ipynb* *Spark: cdh4.srowen.com:4042 (redirects)* *HDFS: /user/srowen/user43*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh4.srowen.com:8880 user44.ipynb* *Spark: cdh4.srowen.com:4043 (redirects)* *HDFS: /user/srowen/user44*
Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh4.srowen.com:8880 user45.ipynb* *Spark: cdh4.srowen.com:4044 (redirects)* *HDFS: /user/srowen/user45*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh4.srowen.com:8880 user46.ipynb* *Spark: cdh4.srowen.com:4045 (redirects)* *HDFS: /user/srowen/user46*
Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh4.srowen.com:8880 user47.ipynb* *Spark: cdh4.srowen.com:4046 (redirects)* *HDFS: /user/srowen/user47*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh4.srowen.com:8880 user48.ipynb* *Spark: cdh4.srowen.com:4047 (redirects)* *HDFS: /user/srowen/user48*

Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh5.srowen.com:8880 user51.ipynb* *Spark: cdh5.srowen.com:4040 (redirects)* *HDFS: /user/srowen/user51*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh5.srowen.com:8880 user52.ipynb* *Spark: cdh5.srowen.com:4041 (redirects)* *HDFS: /user/srowen/user52*
Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh5.srowen.com:8880 user53.ipynb* *Spark: cdh5.srowen.com:4042 (redirects)* *HDFS: /user/srowen/user53*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh5.srowen.com:8880 user54.ipynb* *Spark: cdh5.srowen.com:4043 (redirects)* *HDFS: /user/srowen/user54*
Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh5.srowen.com:8880 user55.ipynb* *Spark: cdh5.srowen.com:4044 (redirects)* *HDFS: /user/srowen/user55*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh5.srowen.com:8880 user56.ipynb* *Spark: cdh5.srowen.com:4045 (redirects)* *HDFS: /user/srowen/user56*
Hue: cdh4.srowen.com:8888 srowen/srowen *ipython: cdh5.srowen.com:8880 user57.ipynb* *Spark: cdh5.srowen.com:4046 (redirects)* *HDFS: /user/srowen/user57*	*Hue: cdh4.srowen.com:8888 srowen/srowen* *ipython: cdh5.srowen.com:8880 user58.ipynb* *Spark: cdh5.srowen.com:4047 (redirects)* *HDFS: /user/srowen/user58*