Big Data Analysis and Visualization with Python

丁来强

About me

- Splunk Lab @Shanghai
- 10+ years working experiences
- Mainly used C++/Python and JS

Contact me

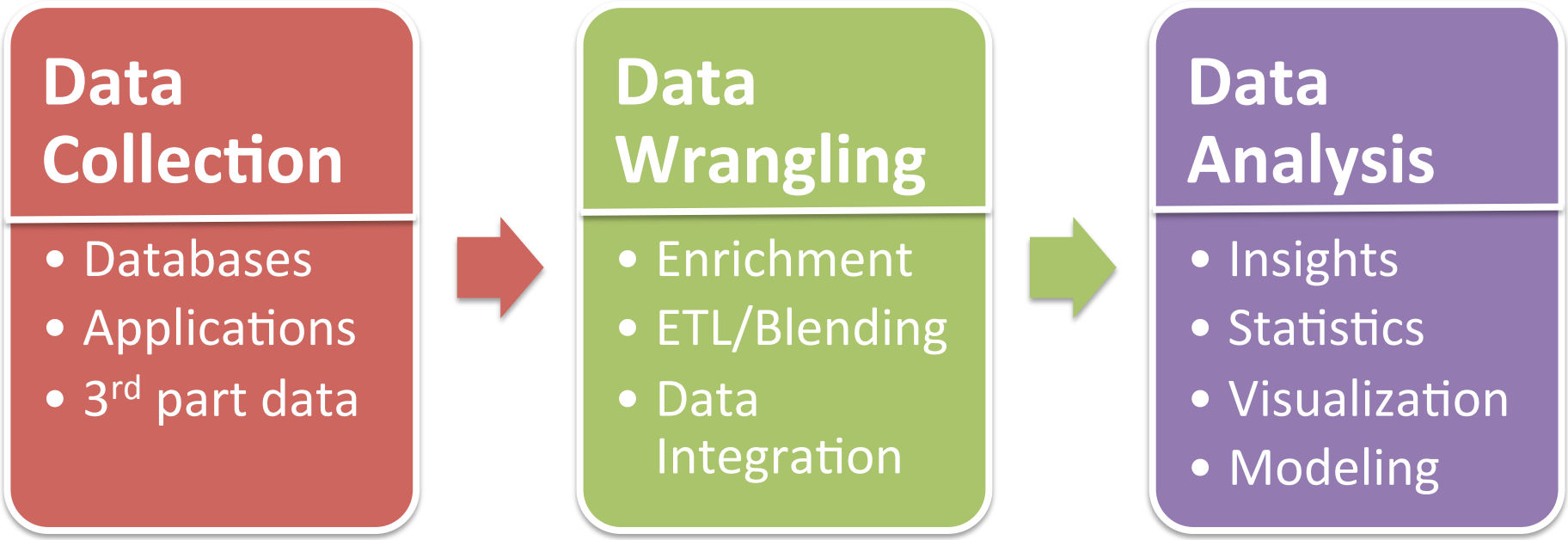
- Email: wjo1212 at 163.com
- Wechat: LaiQiangDing

Agenda

- Data Science ecosystem
- Data Wrangling
- Data Analysis
- Data Visualization
- 3 Real Case Demo
- Bigger Data Consideration
- Spark Data Frame Demo

About Data Science

Data Science Process

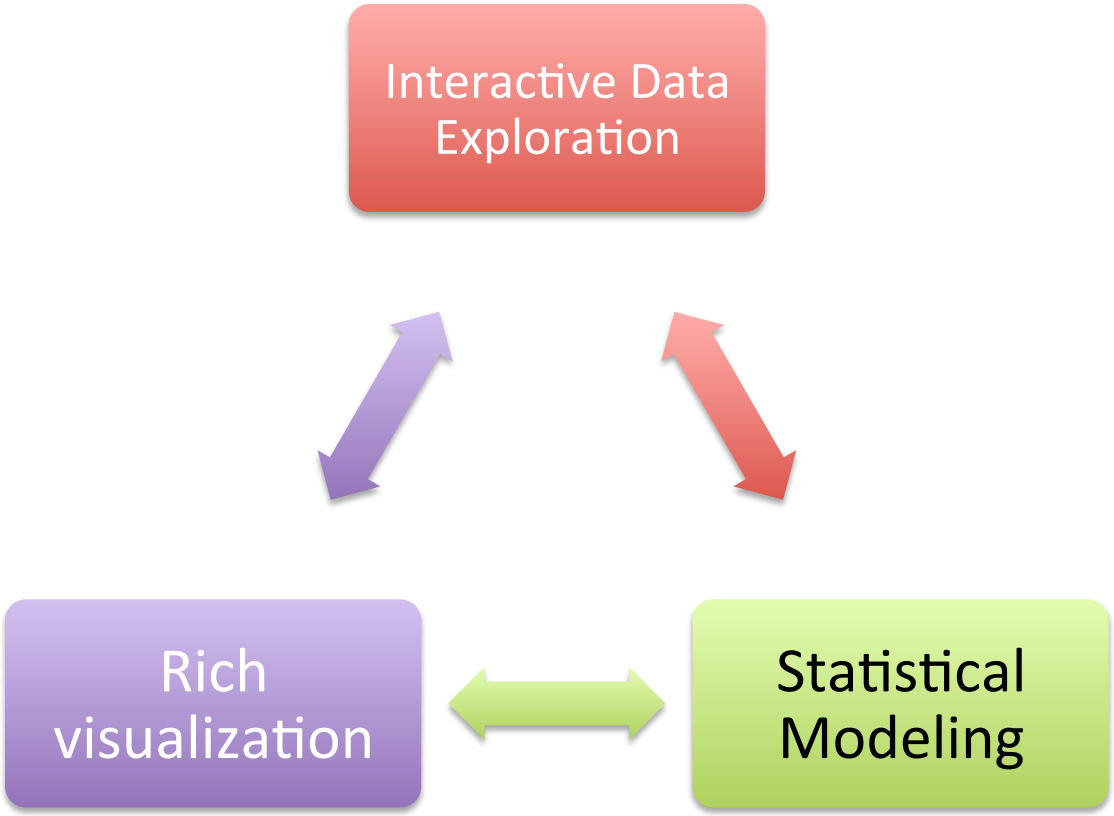


Data Wrangling

Data scientists spend 80% of their time convert data into a usable form.

- Clean data: handle messy or missed data
- Transform and Extract data
- Merge, Join and Reshape data
- Time Series Resampling

Data Analysis



Python vs. R

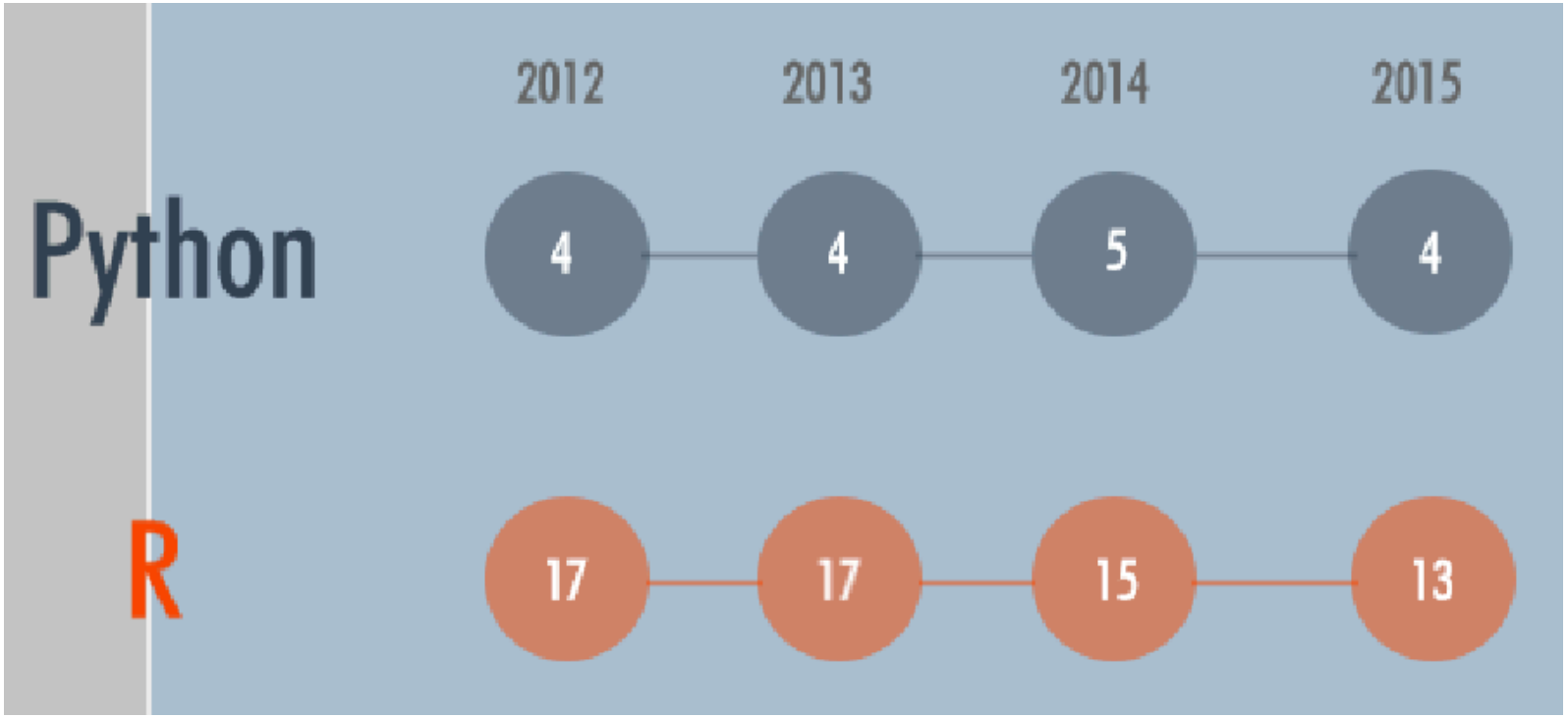
TIOBE Index for Sep. 2015

Sep 2015	Sep 2014	Change	Programming Language
1	2	↑	Java
2	1	↓	C
3	4	↑	C++
4	5	↑	C#
5	**8**	↑	**Python**
6	7	↑	PHP
7	9	↑	JavaScript
8	11	↑	Visual Basic .NET
9	12	↑	Perl
10	3	⇊	Objective-C

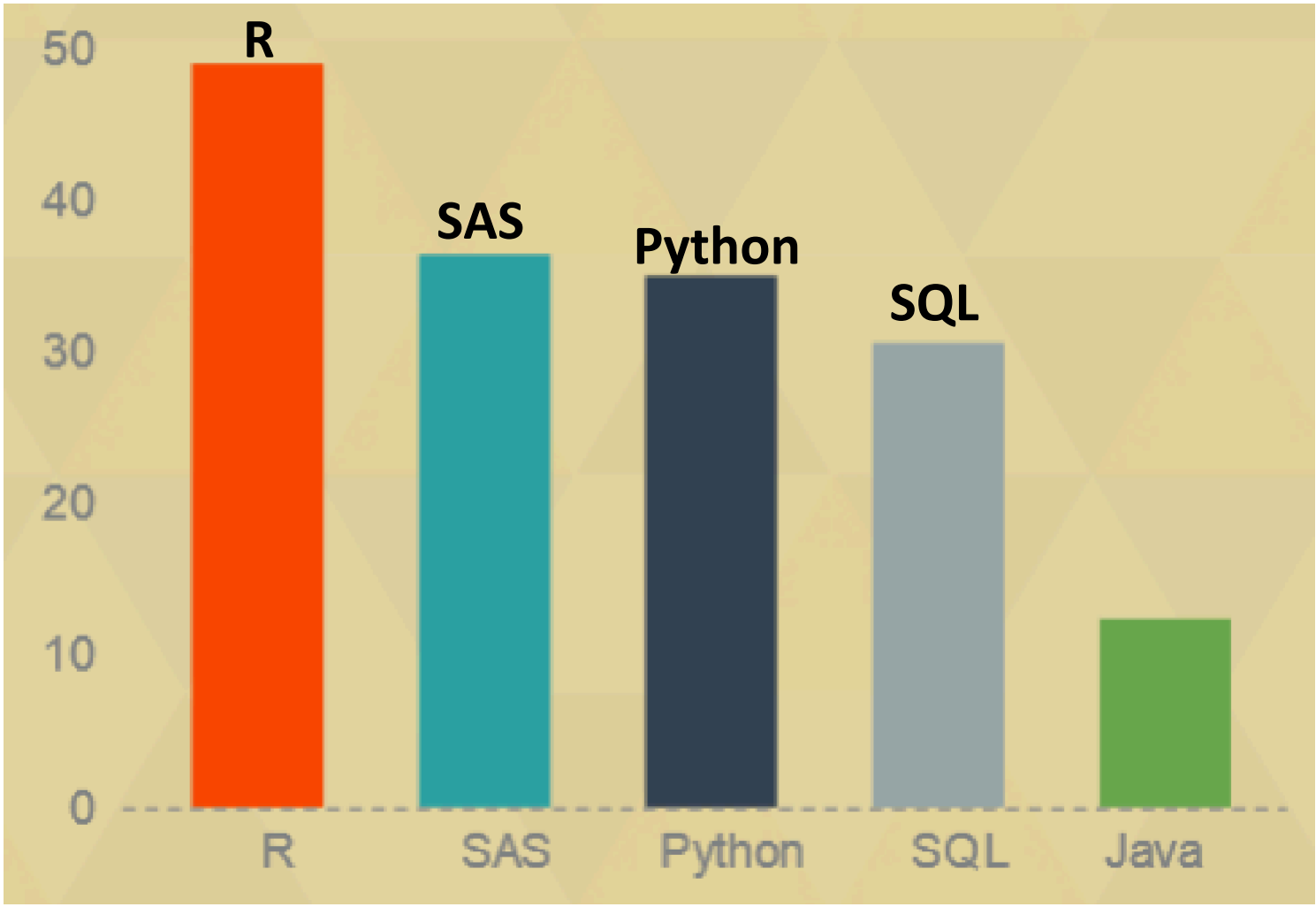
TIOBE Index for Sep. 2015

11	29	⇈	Assembly language
12	13	↑	Ruby
13	15	↑	Delphi/Object Pascal
14	14		Visual Basic
15	17	↑	Pascal
16	18	↑	Swift
17	19	↑	MATLAB
18	20	↑	PL/SQL
19	**21**	↑	**R**
20	31	⇈	COBOL

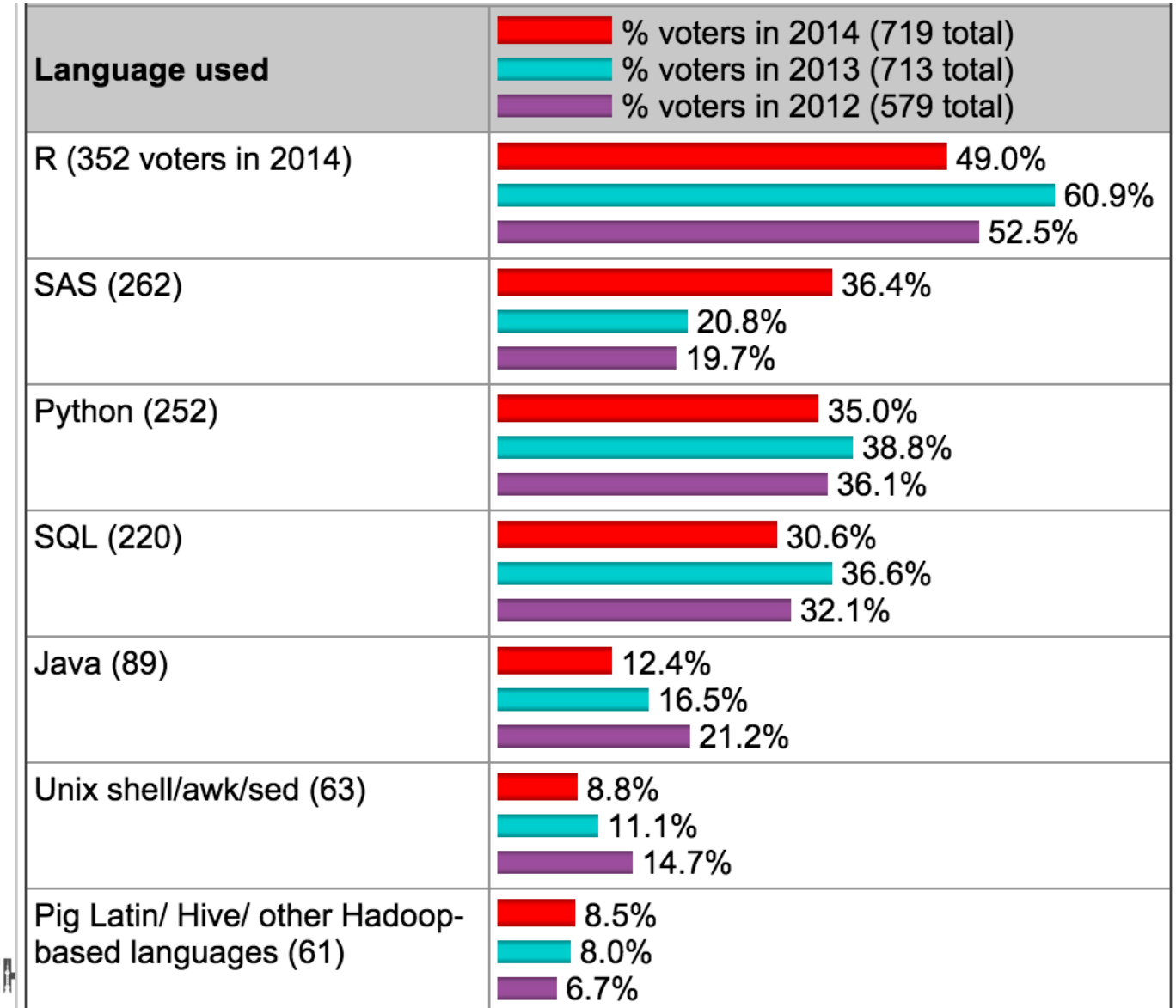
Redmonk ranking (2012~2015)



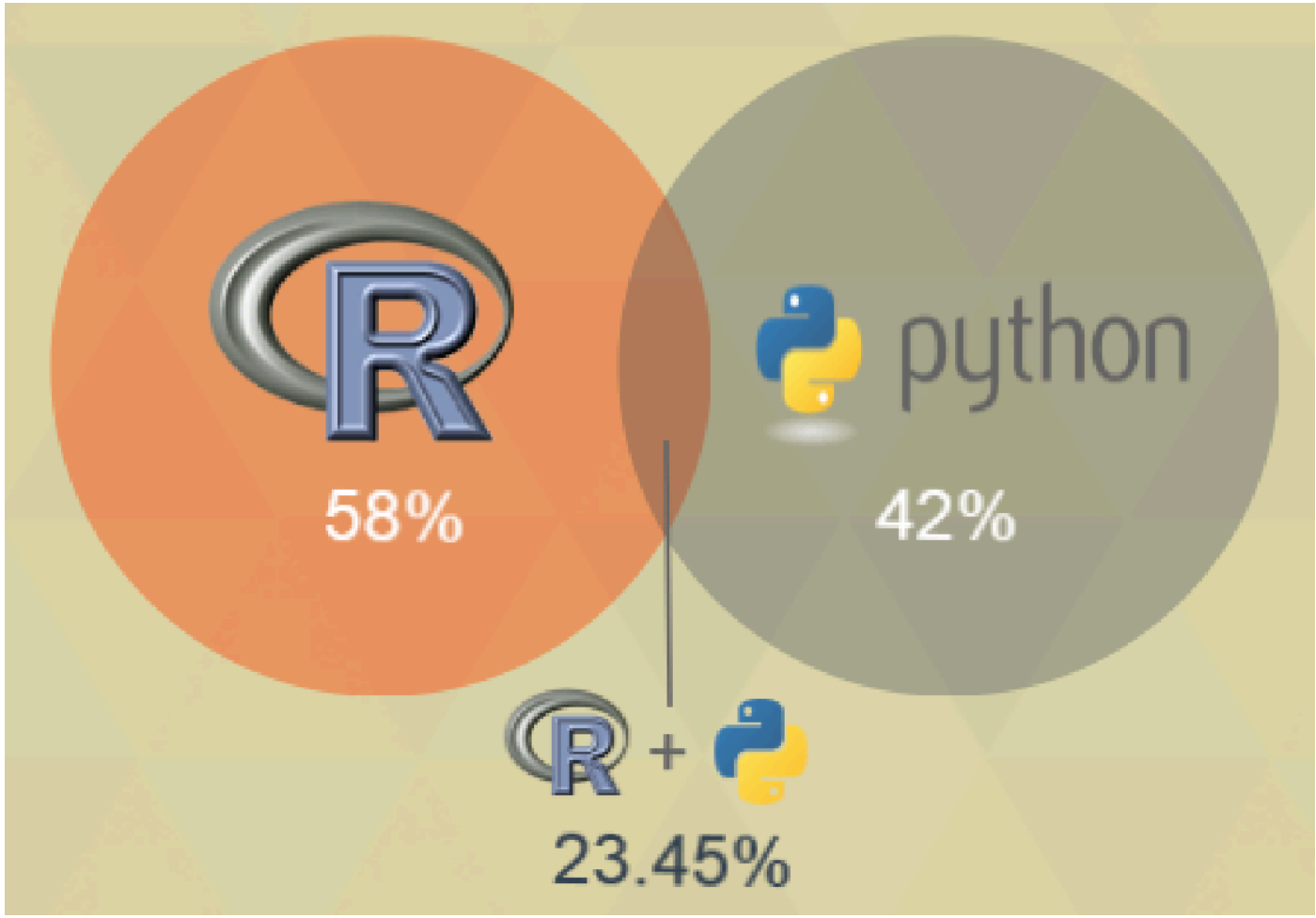
Kdnuggets Polling for data analysis



Kdnuggets Polling for data analysis



Kdnuggets Polling 2014



Pros and Cons

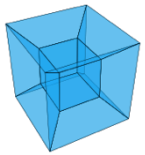
- R + visualization = perfect match
- R, Lingua Franca of Statistics (developed by statisticians)
- R is slow
- Python is multi-purpose language
- Python is challenger for either visualization or essential R packages replacement

More refer to: http://blog.datacamp.com/r-or-python-for-data-analysis/

Python Data Science Ecosystem

PyData Ecosystem

- Fundamental Libs:
  - numpy
  - scipy
- Advanced Libs:
  - pandas
  - sympy
  - SciKit-lean
  - xray
  - Blaze



Blaze

PyData Ecosystem

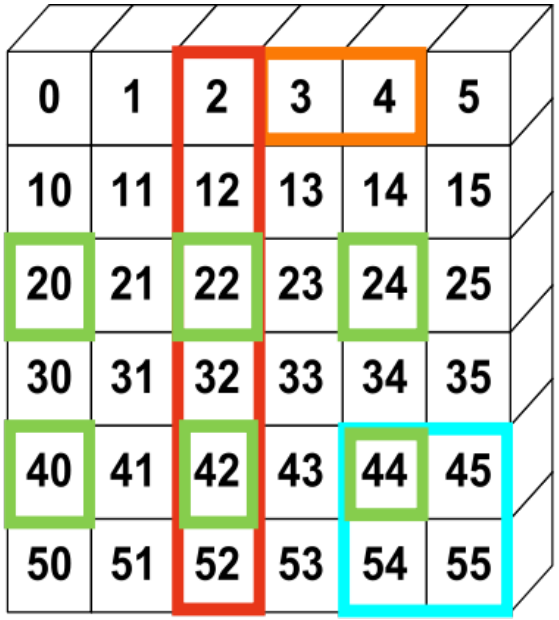
- Visualization
  - matplotlib
  - Searborn
  - bokeh
  - yhat/ggplot
  - vincent, chaco, mayavi…
- IDE, tools:
  - Anaconda: IDE
  - IPython/Notebook

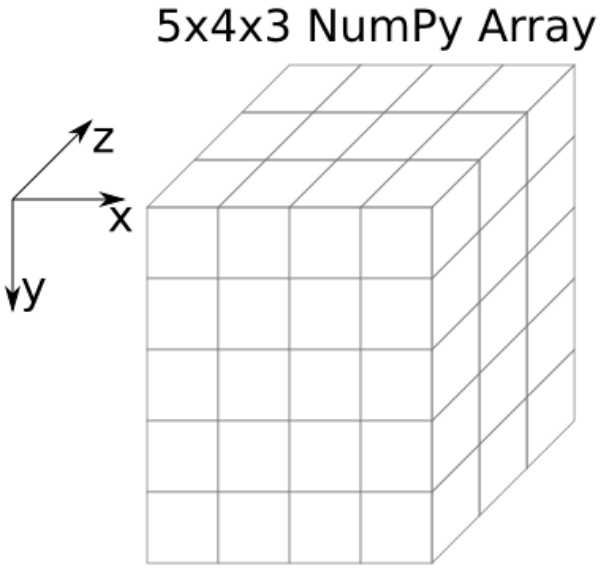
Bokeh

IP[y]: IPython

ANACONDA

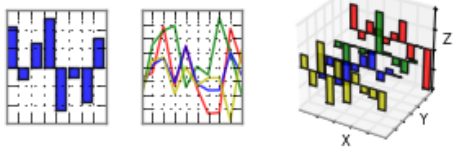
- High performance N-Arrary operation lib





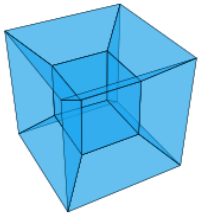
pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$



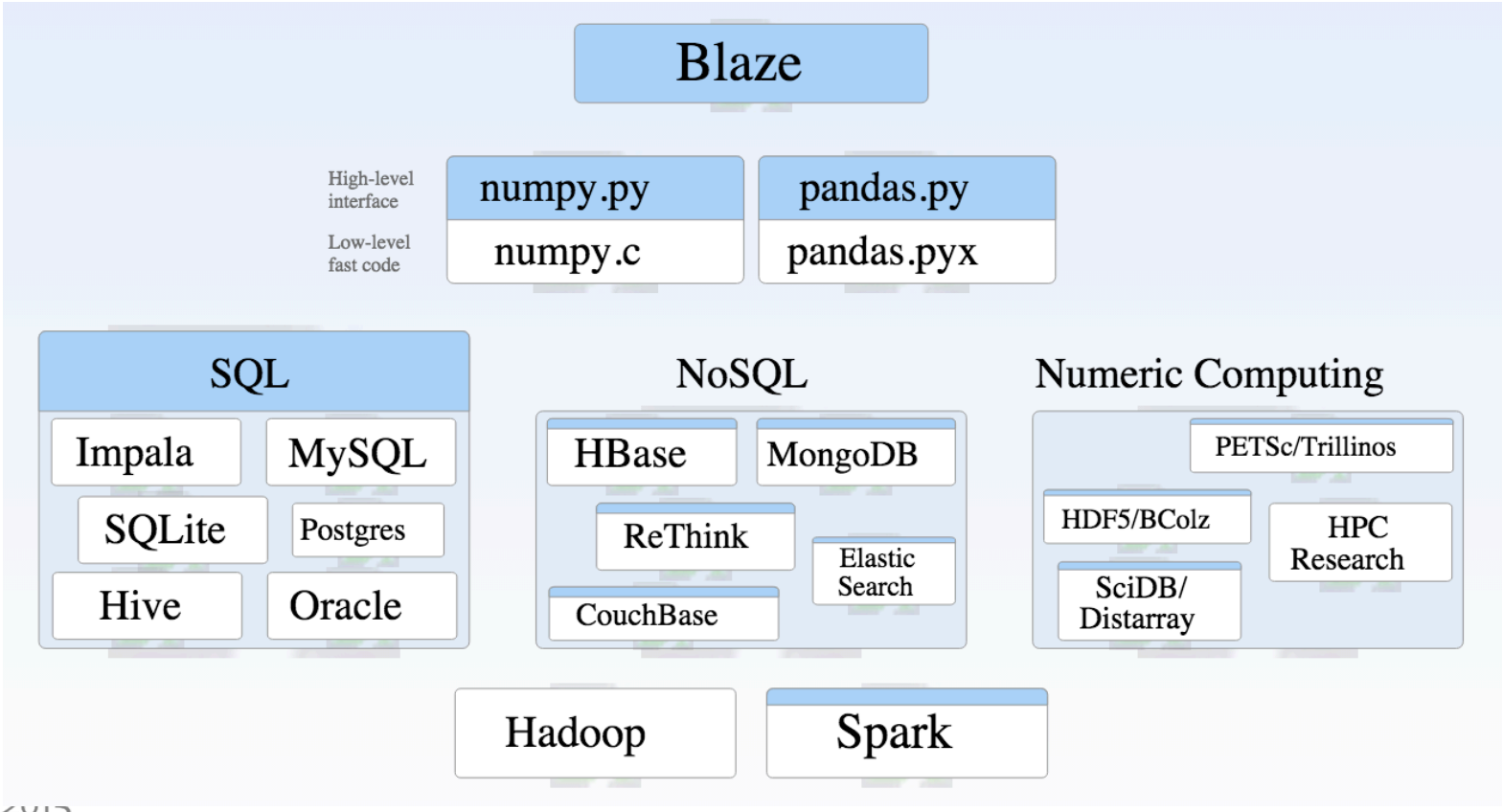
Data Collection → Data Wrangling → Data Analysis

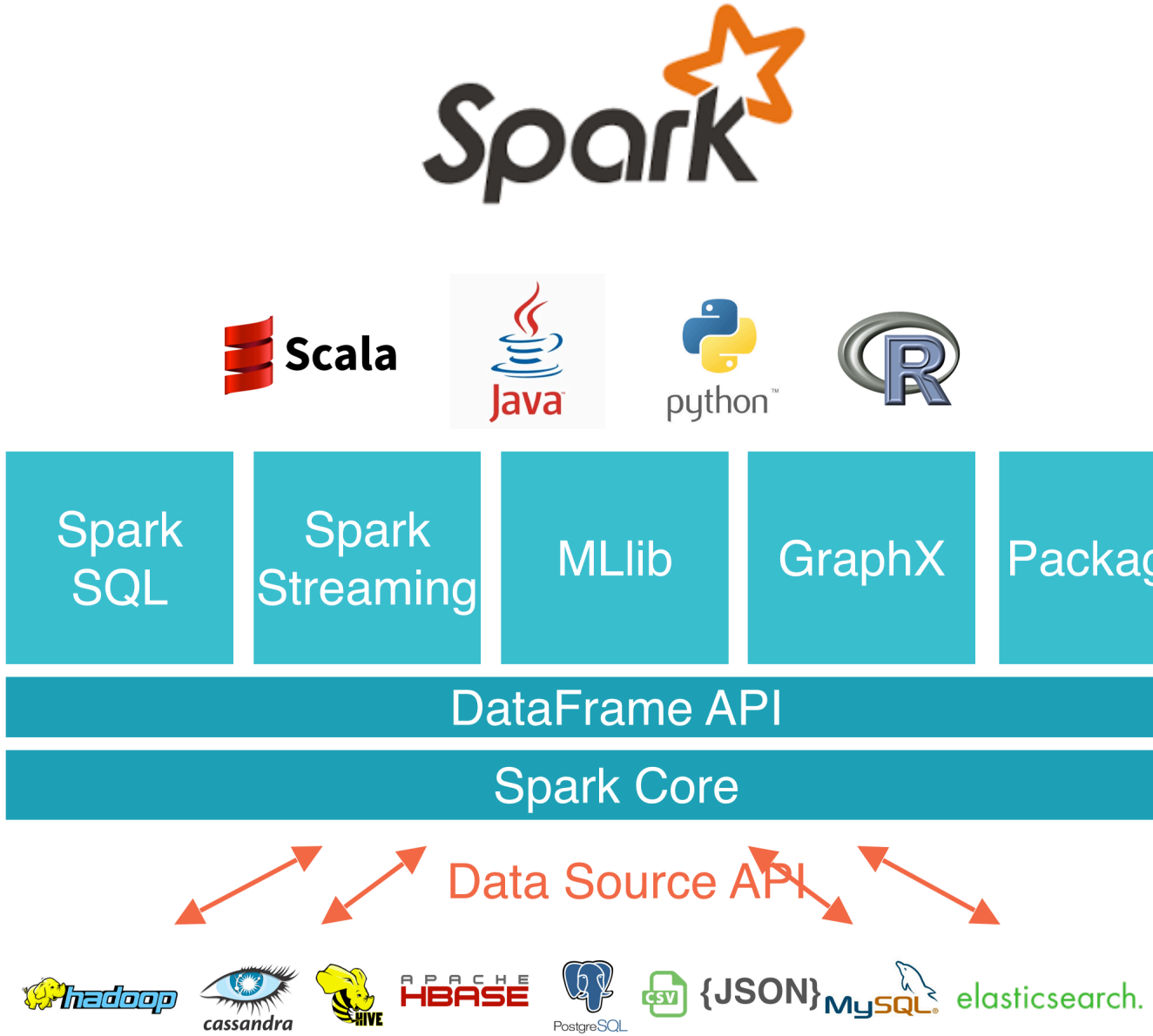
IO	Indexing, Data alignment	Regression
Merging/ Joining	Time Series	Plotting
Sparse Indexing	Group By, Pivoting	Summary Stats



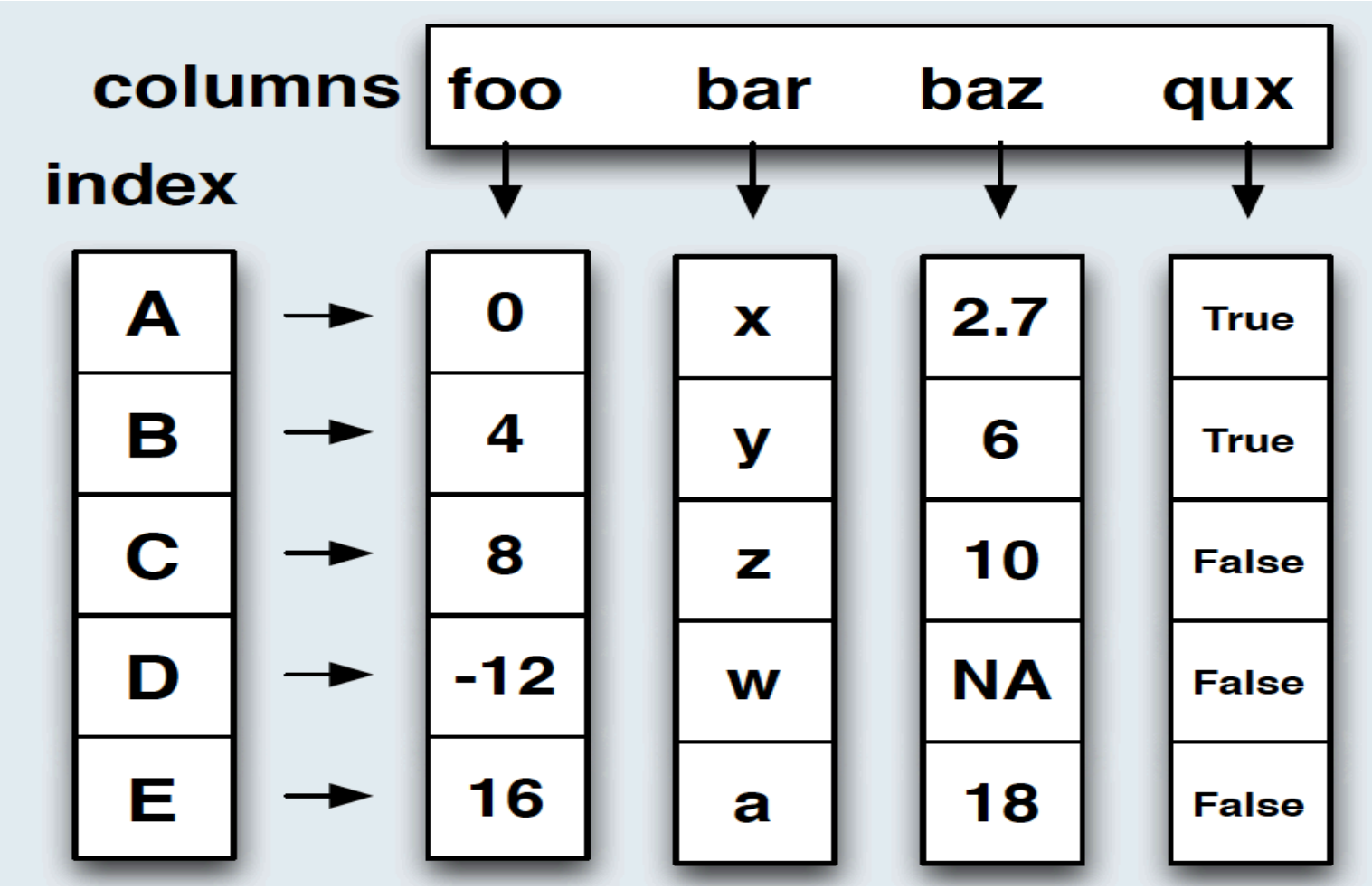
Blaze

- high-level user interface for databases and array computing systems.




Spark
Scala
Java
python
R
Spark SQL
Spark Streaming
MLlib
GraphX
Packages
DataFrame API
Spark Core
Data Source API
hadoop
cassandra
HIVE
APACHE HBASE
PostgreSQL
CSV
{JSON}
MySQL
elasticsearch.

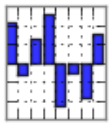
DataFrame

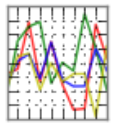

columns
foo
bar
baz
qux
index
A
B
C
D
E
0
4
8
-12
16
x
y
z
w
a
2.7
6
10
NA
18
True
True
False
False
False

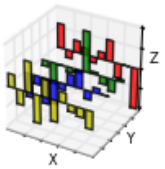
Data Frame

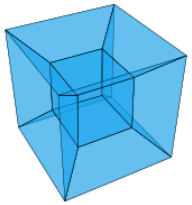
pandas

$$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$$





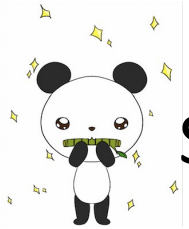




Blaze

Spark Data Frame

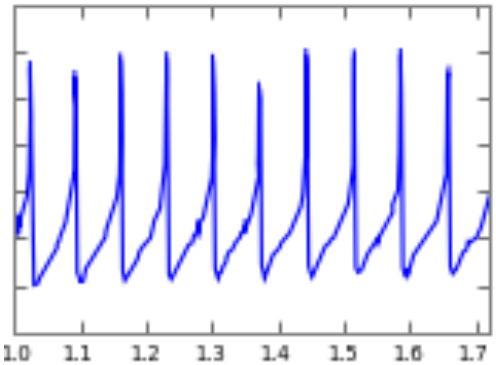
Dask

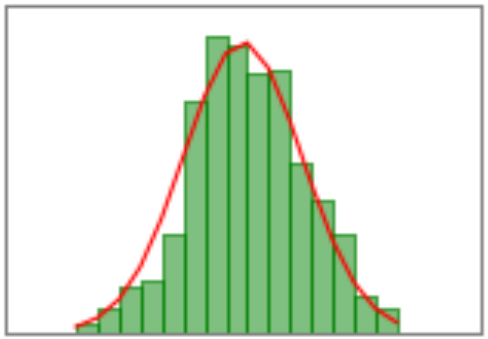


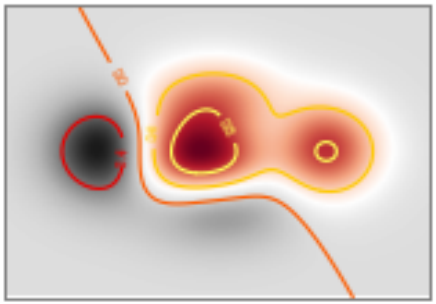
SparkingPandas

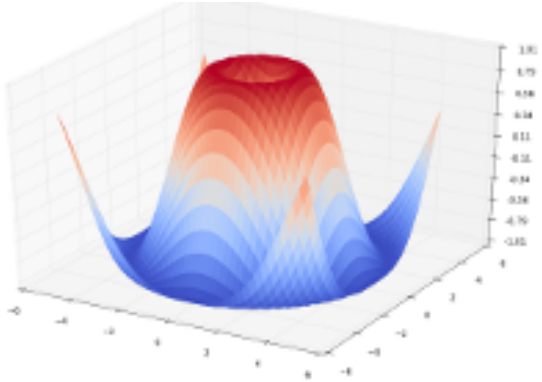
matplotlib

- Powerful Python plotting lib



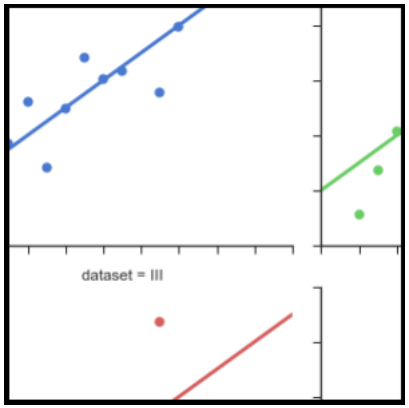


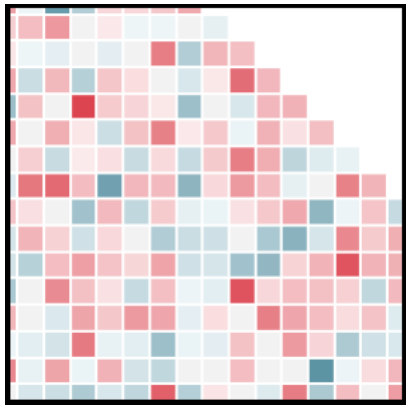


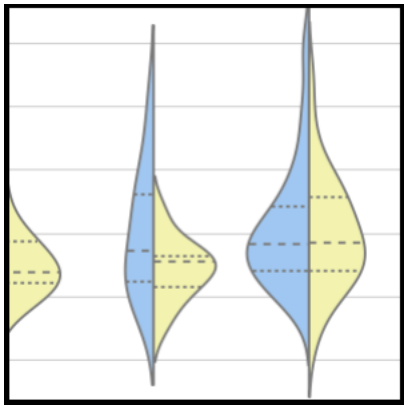


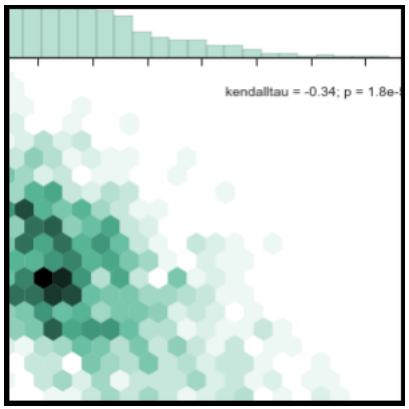
Seaborn

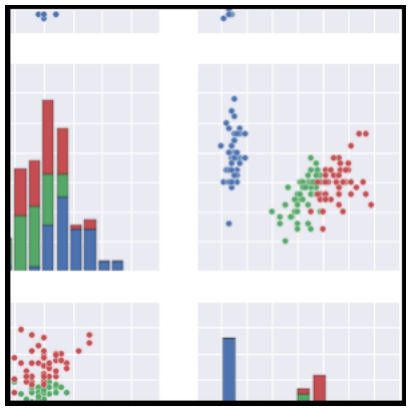
- high-level statistical visualization tool

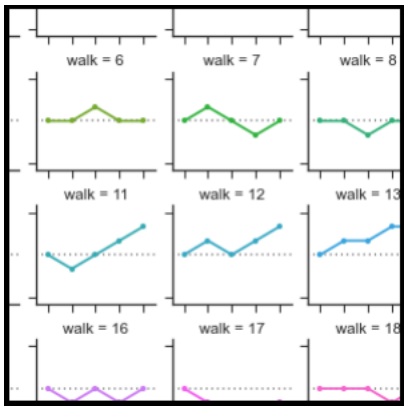






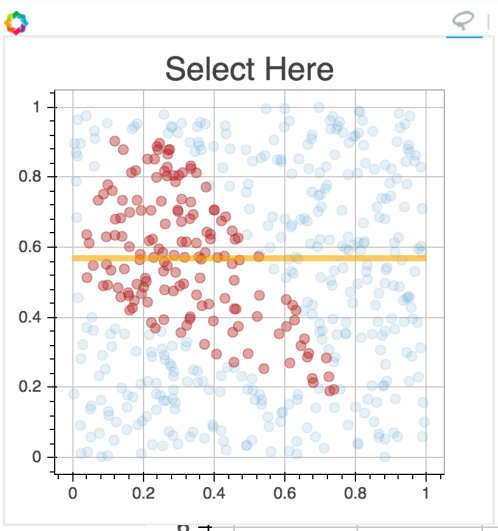


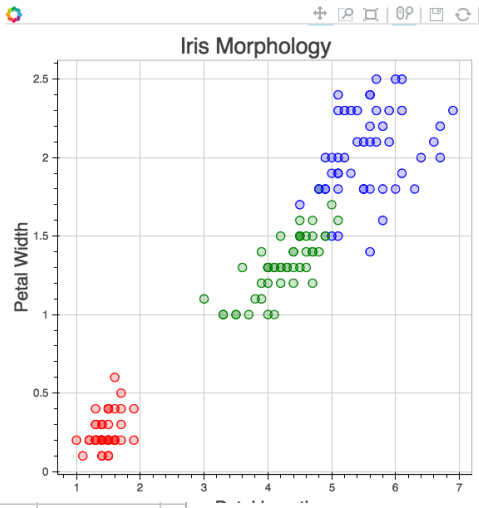


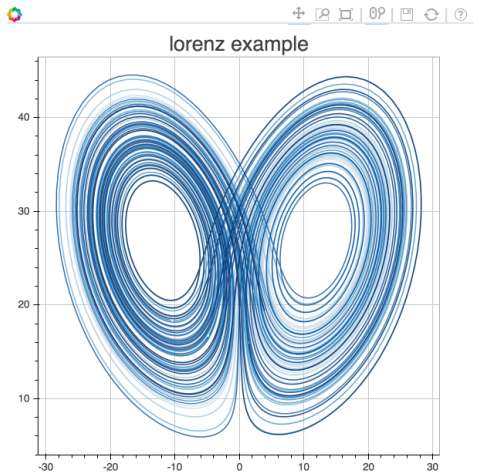


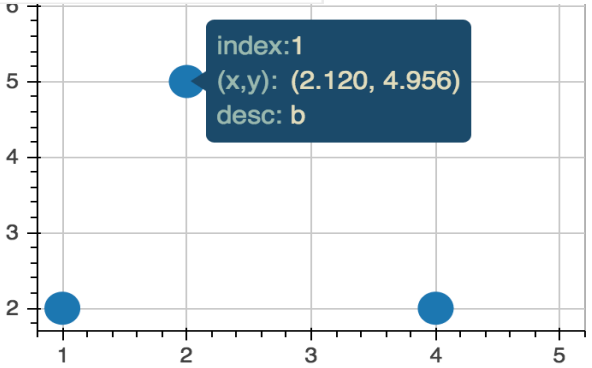
Bokeh

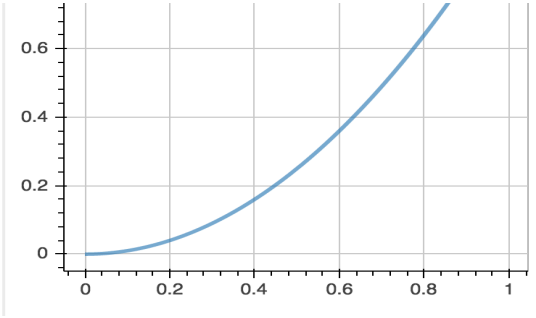
- Web based large data set interactive plotting lib



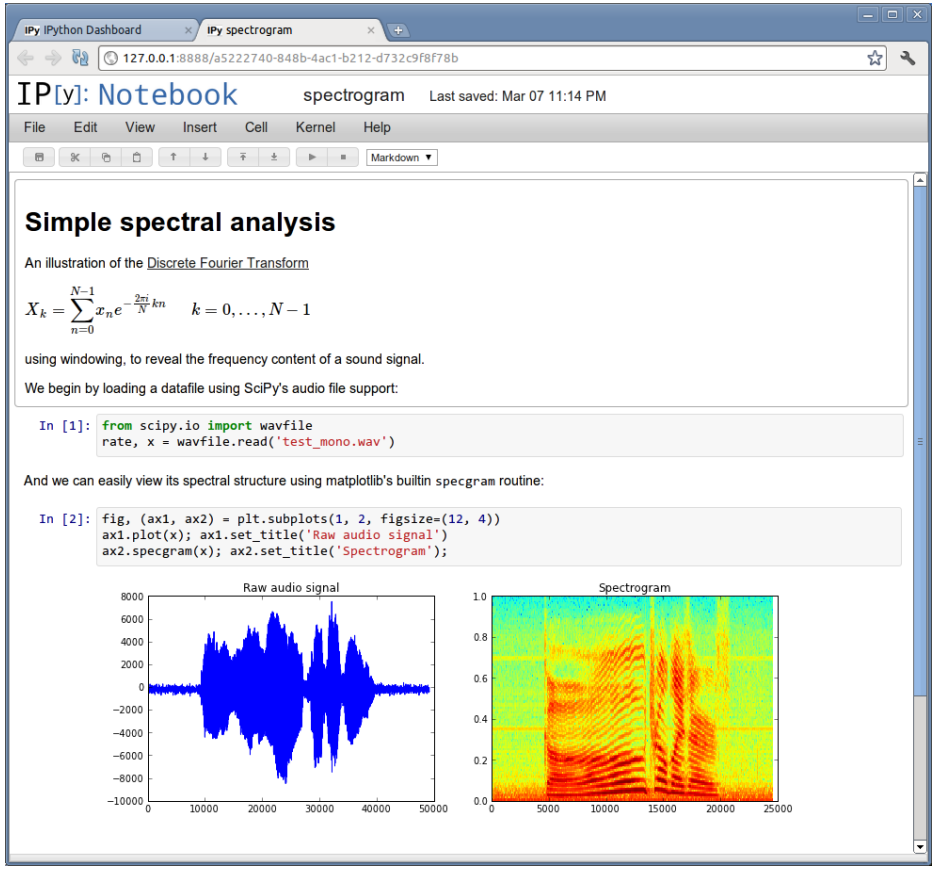


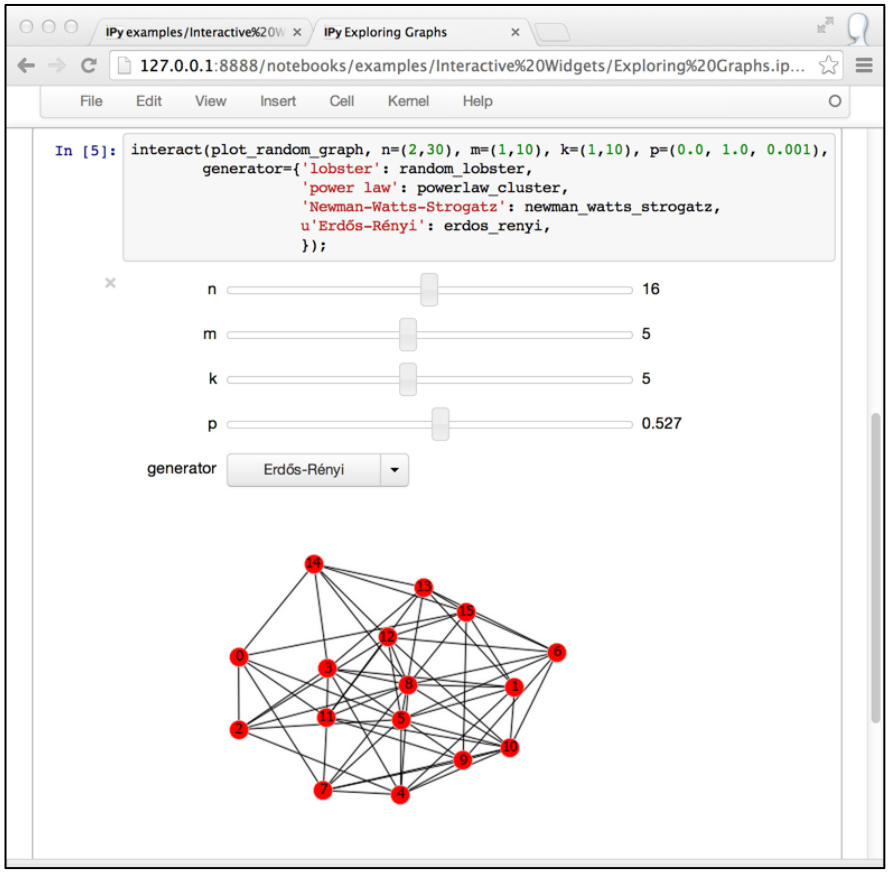






IP[y]: IPython
Interactive Computing

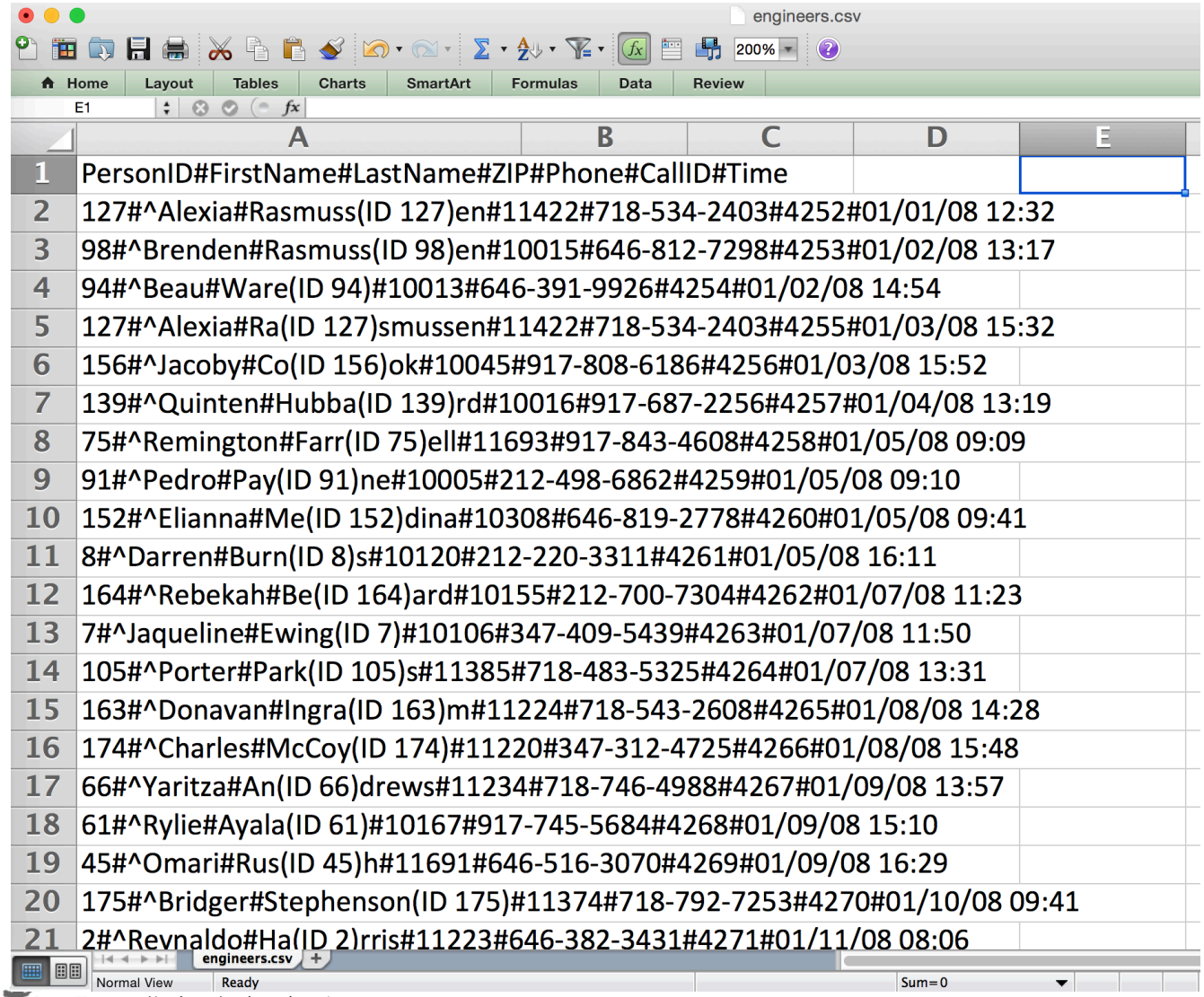




Life's pathetic, go Pythonic! *(wjo1212@163.com)*

Data Wrangling

Start with an employee list



Load it

In [1]:
```
import numpy as np
import pandas as pd
```

In [7]:
```
df = pd.read_csv("./engineers.csv", sep="#")
```

In [8]:
```
df.head()
```

Out[8]:

	PersonID	FirstName	LastName	ZIP	Phone	CallID	Time
0	127	^Alexia	Rasmuss(ID 127)en	11422	718-534-2403	4252	01/01/08 12:32
1	98	^Brenden	Rasmuss(ID 98)en	10015	646-812-7298	4253	01/02/08 13:17
2	94	^Beau	Ware(ID 94)	10013	646-391-9926	4254	01/02/08 14:54
3	127	^Alexia	Ra(ID 127)smussen	11422	718-534-2403	4255	01/03/08 15:32
4	156	^Jacoby	Co(ID 156)ok	10045	917-808-6186	4256	01/03/08 15:52

Messy Names

```
In [1]: import numpy as np
        import pandas as pd
```

```
In [7]: df = pd.read_csv("./engineers.csv", sep="#")
```

```
In [8]: df.head()
```

Out[8]:

	PersonID	FirstName	LastName	ZIP	Phone	CallID	Time
0	127	^Alexia	Rasmuss(ID 127)en	11422	718-534-2403	4252	01/01/08 12:32
1	98	^Brenden	Rasmuss(ID 98)en	10015	646-812-7298	4253	01/02/08 13:17
2	94	^Beau	Ware(ID 94)	10013	646-391-9926	4254	01/02/08 14:54
3	127	^Alexia	Ra(ID 127)smussen	11422	718-534-2403	4255	01/03/08 15:32
4	156	^Jacoby	Co(ID 156)ok	10045	917-808-6186	4256	01/03/08 15:52

Transform it

```
In [20]: df.FirstName = df.FirstName.str.replace(r"\^", "")
         df.LastName = df.LastName.str.replace(r"\(ID \d+\)", "")
         df.head()
```

Out[20]:

	PersonID	FirstName	LastName	ZIP	Phone	CallID	Time
0	127	Alexia	Rasmussen	11422	718-534-2403	4252	2008-01-01 12:32:00
1	98	Brenden	Rasmussen	10015	646-812-7298	4253	2008-01-02 13:17:00
2	94	Beau	Ware	10013	646-391-9926	4254	2008-01-02 14:54:00
3	127	Alexia	Rasmussen	11422	718-534-2403	4255	2008-01-03 15:32:00
4	156	Jacoby	Cook	10045	917-808-6186	4256	2008-01-03 15:52:00

Duplicated People

In [20]:
```
df.FirstName = df.FirstName.str.replace(r"\^", "")
df.LastName = df.LastName.str.replace(r"\(ID \d+\)", "")
df.head()
```

Out[20]:

	PersonID	FirstName	LastName	ZIP	Phone	CallID	Time
0	127	Alexia	Rasmussen	11422	718-534-2403	4252	2008-01-01 12:32:00
1	98	Brenden	Rasmussen	10015	646-812-7298	4253	2008-01-02 13:17:00
2	94	Beau	Ware	10013	646-391-9926	4254	2008-01-02 14:54:00
3	127	Alexia	Rasmussen	11422	718-534-2403	4255	2008-01-03 15:32:00
4	156	Jacoby	Cook	10045	917-808-6186	4256	2008-01-03 15:52:00

Drop it

```
In [51]: df.drop_duplicates('PersonID', inplace=True)
         df.drop('CallID', axis=1, inplace=True)
         df.head()
```

Out[51]:

	PersonID	FirstName	LastName	ZIP	Phone	Time
0	127	Alexia	Rasmussen	11422	718-534-2403	2008-01-01 12:32:00
1	98	Brenden	Rasmussen	10015	646-812-7298	2008-01-02 13:17:00
2	94	Beau	Ware	10013	646-391-9926	2008-01-02 14:54:00
4	156	Jacoby	Cook	10045	917-808-6186	2008-01-03 15:52:00
5	139	Quinten	Hubbard	10016	917-687-2256	2008-01-04 13:19:00

Verify

Any ill-formatted phone number?

	PersonID	FirstName	LastName	ZIP	Phone	Time
0	127	Alexia	Rasmussen	11422	718-534-2403	2008-01-01 12:32:00
1	98	Brenden	Rasmussen	10015	646-812-7298	2008-01-02 13:17:00
2	94	Beau	Ware	10013	646-391-9926	2008-01-02 14:54:00
4	156	Jacoby	Cook	10045	917-808-6186	2008-01-03 15:52:00
5	139	Quinten	Hubbard	10016	917-687-2256	2008-01-04 13:19:00

Verify

```
In [138]: df[~df.Phone.str.match(r"\d{3}-\d{3}-\d{4}")]
```

Out[138]:

	PersonID	FirstName	LastName	ZIP	Phone	CallID	Time

search out all ill-formatted data

Output all people after 2008

```
In [ ]: df = df.set_index('PersonID')
```

```
In [85]: df = df[df.Time.dt.year >= 2008]
         df.sort("Time").to_csv("people_parsed_2008.csv")
```

```
In [87]: !cat ./people_parsed_2008.csv
```

```
PersonID,FirstName,LastName,ZIP,Phone,Time
127,Alexia,Rasmussen,11422,718-534-2403,2008-01-01 12:32:00
98,Brenden,Rasmussen,10015,646-812-7298,2008-01-02 13:17:00
94,Beau,Ware,10013,646-391-9926,
156,Jacoby,Cook,10045,917-808-61
139,Quinten,Hubbard,10016,917-68
75,Remington,Farrell,11693,917-843-4608,2008-01-05 09:09:00
91,Pedro,Payne,10005,212-498-6862,2008-01-05 09:10:00
```

Ready for Analaysis

Put all code together: just one line

```
pd.read_csv("./engineers.csv", sep="#", parse_dates=['Time']) \
  .assign(FirstName=df.FirstName.str.replace(r"\^", "")) \
  .assign(LastName=df.LastName.str.replace(r"\(ID \d+\)", "")) \
  .drop_duplicates('PersonID') \
  .drop('CallID', axis=1) \
  .set_index('PersonID') \
  .sort("Time") \
  .query('Time.dt.year >= 2008') \
  .to_csv("people_parsed_2008.csv")
```

Chained Call

Next, we have three tables

	company	year	employees
0	ICBC	2014	8000
1	Splunk	2014	1000
2	Cisco	2014	30000
3	EMC	2014	20000

	company	year	employees
0	ICBC	2015	10000
1	Splunk	2015	1000
2	Cisco	2015	25000
3	EMC	2015	20000

	company	category
0	ICBC	Bank
1	Splunk	Big Data
2	Cisco	Network
3	EMC	Storage
4	HP	Server

Next, we have three tables

	company	year	employees
0	ICBC	2014	8000
1	Splunk	2014	1000
2	Cisco	2014	30000
3	EMC	2014	20000

	company	year	employees
0	ICBC	2015	10000
1	Splunk	2015	1000
2	Cisco	2015	25000
3	EMC	2015	20000

	company	category
0	ICBC	Bank
1	Splunk	Big Data
2	Cisco	Network
3	EMC	Storage
4	HP	Server

Concatenate and Join them

In [122]:
```
df = pd.concat([df1, df2])
df = df[["company", "year", "employees"]]
pd.merge(df, df3, on='company')
```

Out[122]:

	company	year	employees	category
0	ICBC	2014	8000	Bank
1	ICBC	2015	10000	Bank
2	Splunk	2014	1000	Big Data
3	Splunk	2015	1000	Big Data
4	Cisco	2014	30000	Network
5	Cisco	2015	25000	Network
6	EMC	2014	20000	Storage
7	EMC	2015	20000	Storage

Concatenate and Join them

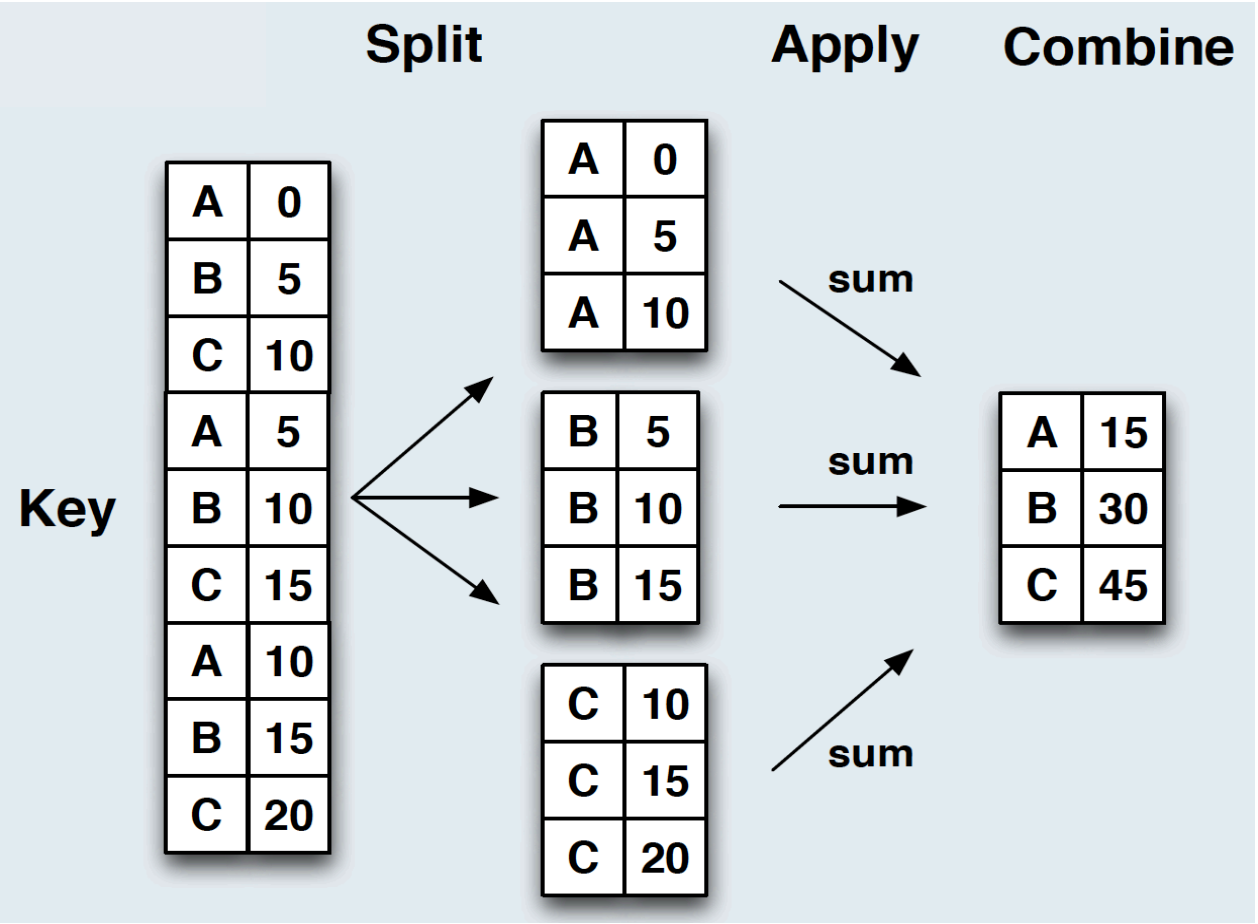
```
In [122]: df = pd.concat([df1, df2])
          df = df[["company", "year", "employees"]]
          pd.merge(df, df3, on='company')
```

Out[122]:

	company	year	employees	category
0	ICBC	2014	8000	Bank
1	ICBC	2015	10000	Bank
2	Splunk	2014	1000	Big Data
3	Splunk	2015	1000	Big Data
4	Cisco	2014	30000	Network
5	Cisco	2015	25000	Network
6	EMC	2014	20000	Storage
7	EMC	2015	20000	Storage

Split-Apply-Combine

- More powerful than Data Frame in Dark, Blaze, or Spark



Powerful Time Series Support

- One key reason for finance usage

Name	Description
D	Calendar day
B	Business day
M	Calendar end of month
BM	Business end of month
MS	Calendar start of month
BMS	Calendar start of month
W-{MON, TUE,...}	Weekly on Monday, Tuesday, ...
Q-{JAN, FEB,...}	Quarterly starting on January, February...
A-{JAN, FEB, ...}	Business year end (December)
H	Hour
T	Minute
s	Second
L, ms	Millisecond
U	Microsecond

Workday?
Vacation?
Finance Year?
Finance Quarter?
Moving Window?
Resample?

Case Study


Csdn.net
登录 | 注册
帮助
论坛牛人
论坛地图
专家问答
论坛焦点
头条回顾
更多>>
下载CSDN 移动客户端
【黄勇】架构探险 略惊异的面
3D游戏终极技术全像素3D！
大波C币等着你 问答9月新活动
程序员6年coding踏出高效狂暴
大家在玩的“极客头条”是什么
书写“程序人生”，好礼
程序员越老越吃香 论坛标兵，
C程序员从校园到职场 程序猿常见面试题
5种优化你的安卓5.0 Lollipop代码的方法
YOU HAVE BEEN
HACKED !
论坛公告
CSDN 移动客户端发布
JetBrains个人版限时7折
高效程序员的狂暴之路
你是“野生程序员”吗?
C语言及程序设计初步
用大数据融合创造大决策智脑
软考套餐限时7折优惠
高校俱乐部全国巡讲讲师招募
我要发帖
版主推荐-技术区
更多>>
[u012005572推荐] 【网络与通信】大板上线公告 [网络协议与配置]
u012005572
WJB08223
论坛标兵
技术区
非技术区

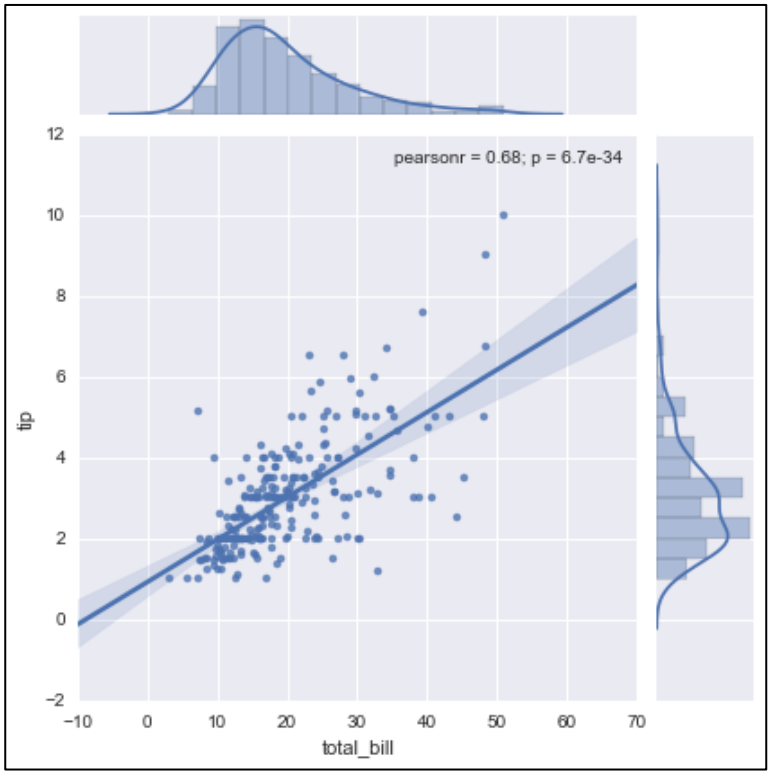

AdultFriendFinder
Hookup, Find Sex or Meet Someone Hot Now
Login
Forgot password?
Join Now!
Home
Browse
Hookup
Dating Forums
Live Chat
Sign Up Now!
Start Hooking Up Tonight!
I am/We are a:
Man
Interested in meeting:
Man
Woman
Couples / Groups
TS/TV/TGs
My birthdate:
Month
Day
Year
Country:
United States
Zip code:
Register Now
The Hottest
Dating, Hookup and Sex
Community

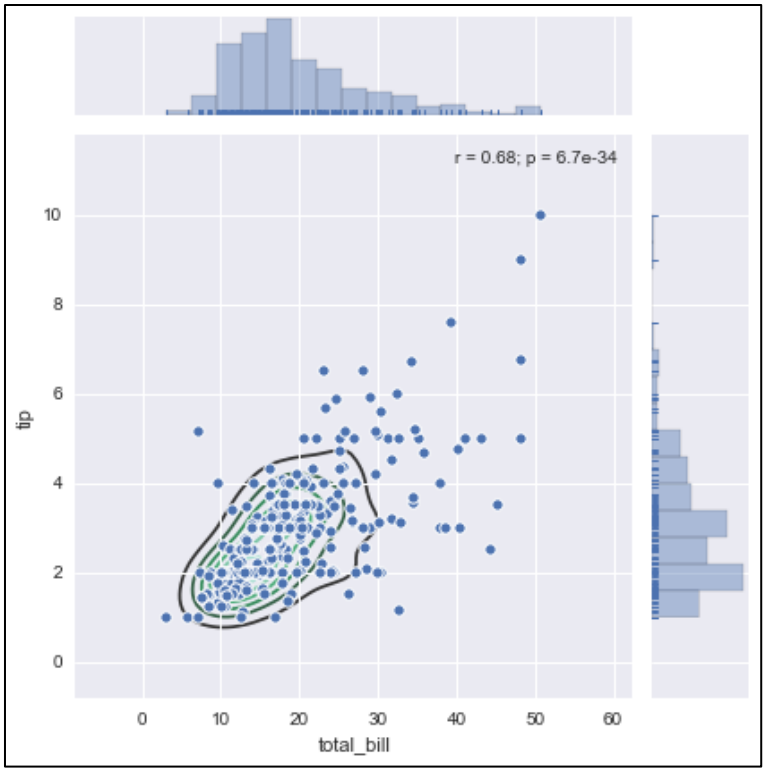

AdultFriendFinder
Hookup, Find Sex or Meet Someone Hot Now
Login
Forgot password?
Join Now!
Home
Browse
Hookup
Dating Forums
Live Chat
Sign Up Now!
Start Hooking Up Tonight!
YOU HAVE BEEN HACKED !
The Hottest
Dating, Hookup and Sex Community

Car Sale Case in China

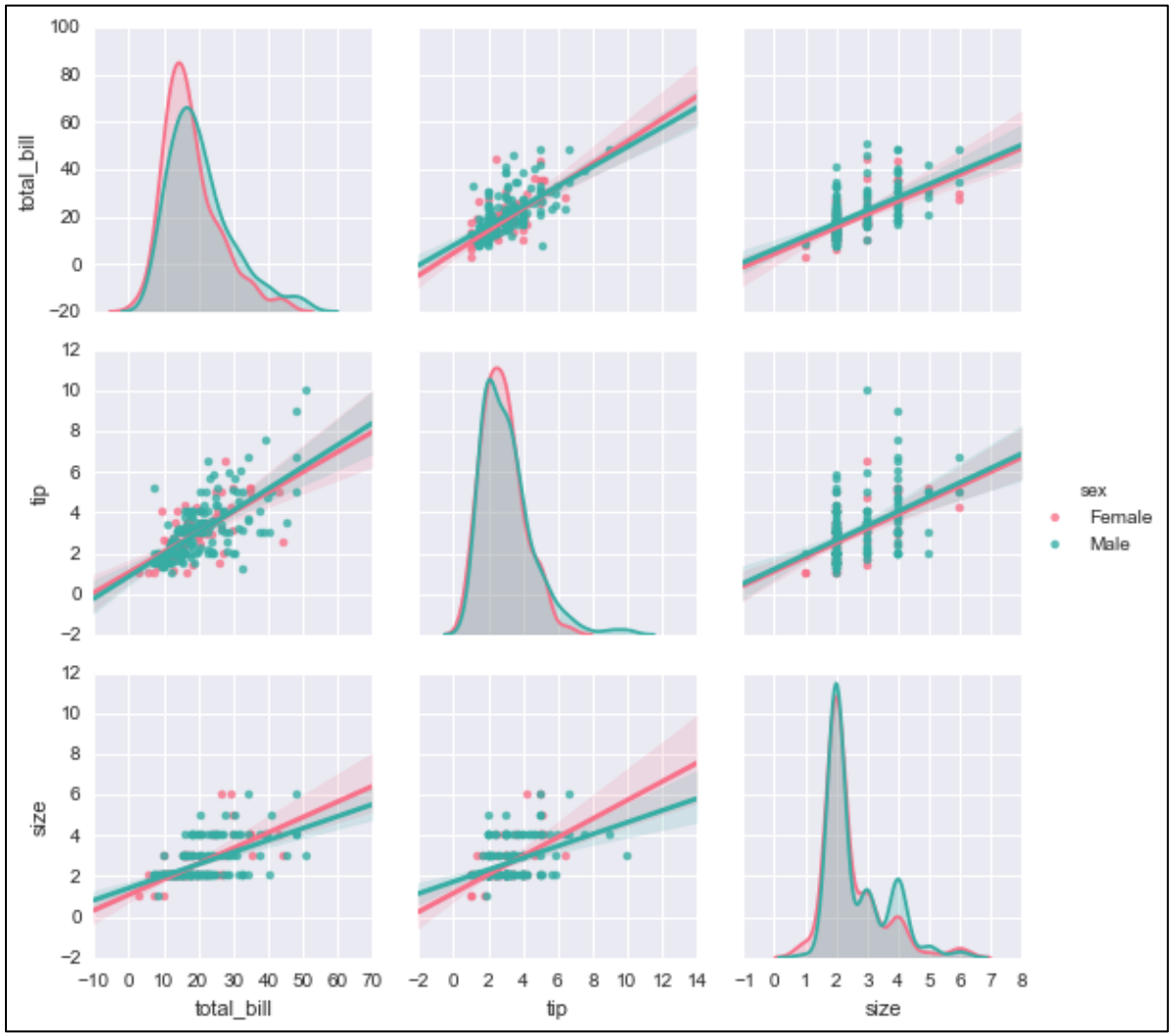
More About Visualization

Seaborn JointPlot

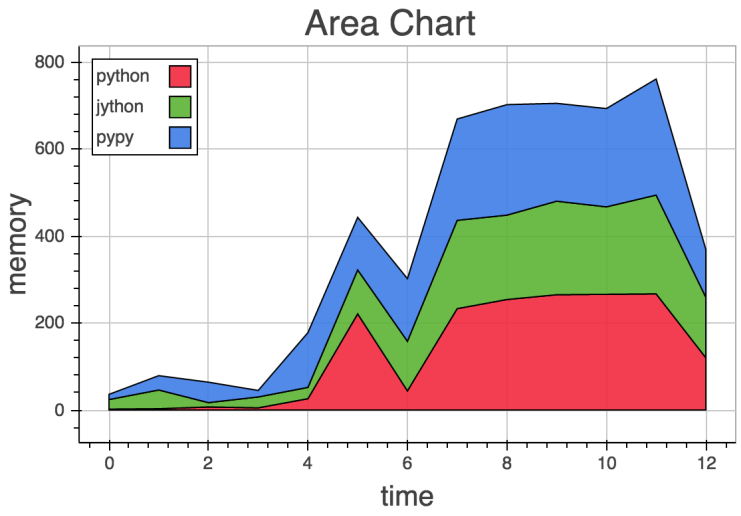


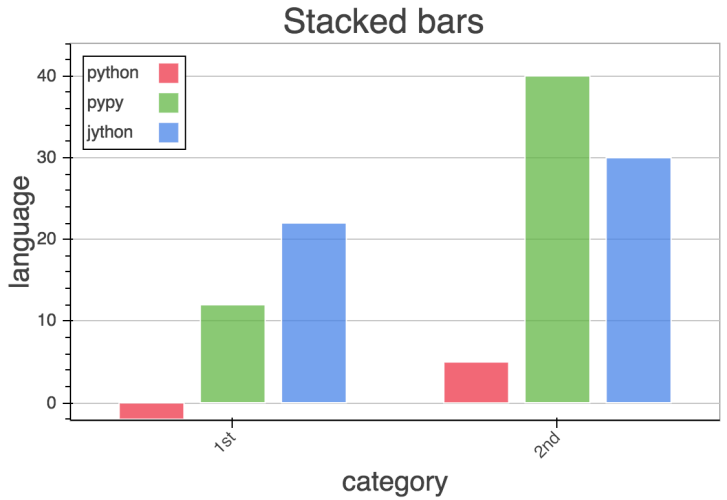


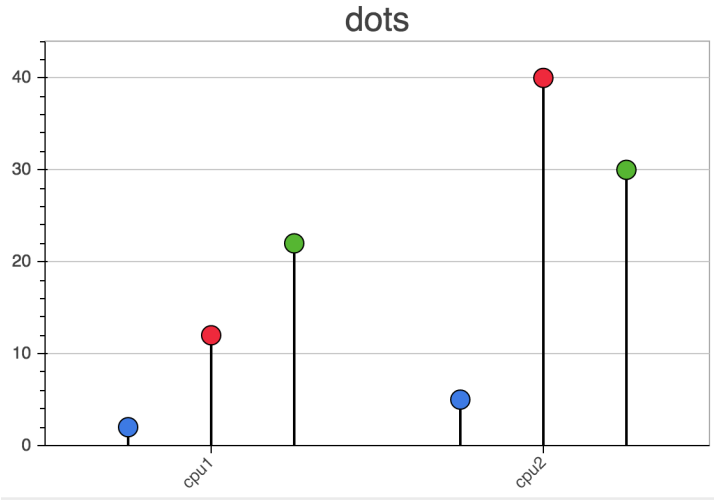
Seaborn PairPlot

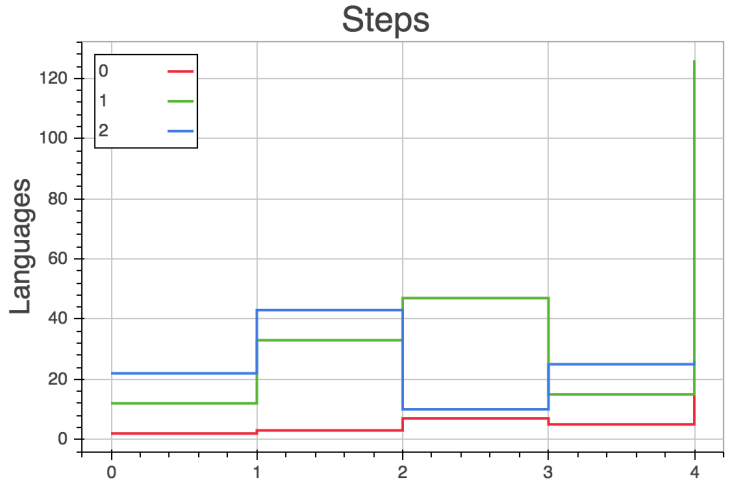


Bokeh Charts

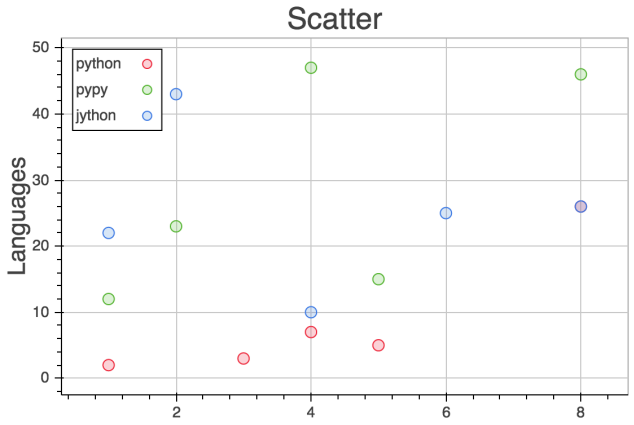


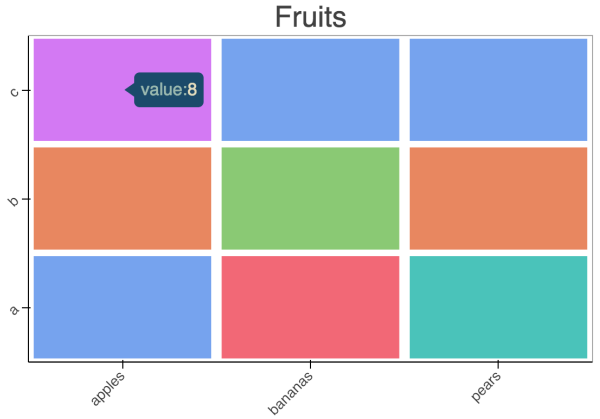


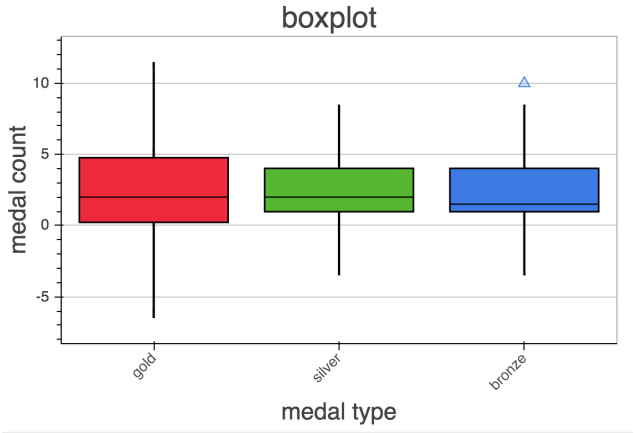


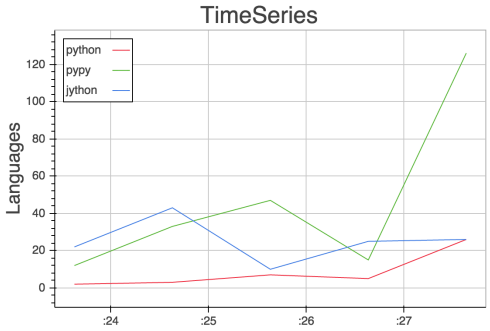


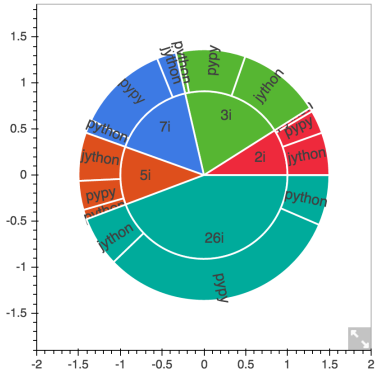
Bokeh Charts





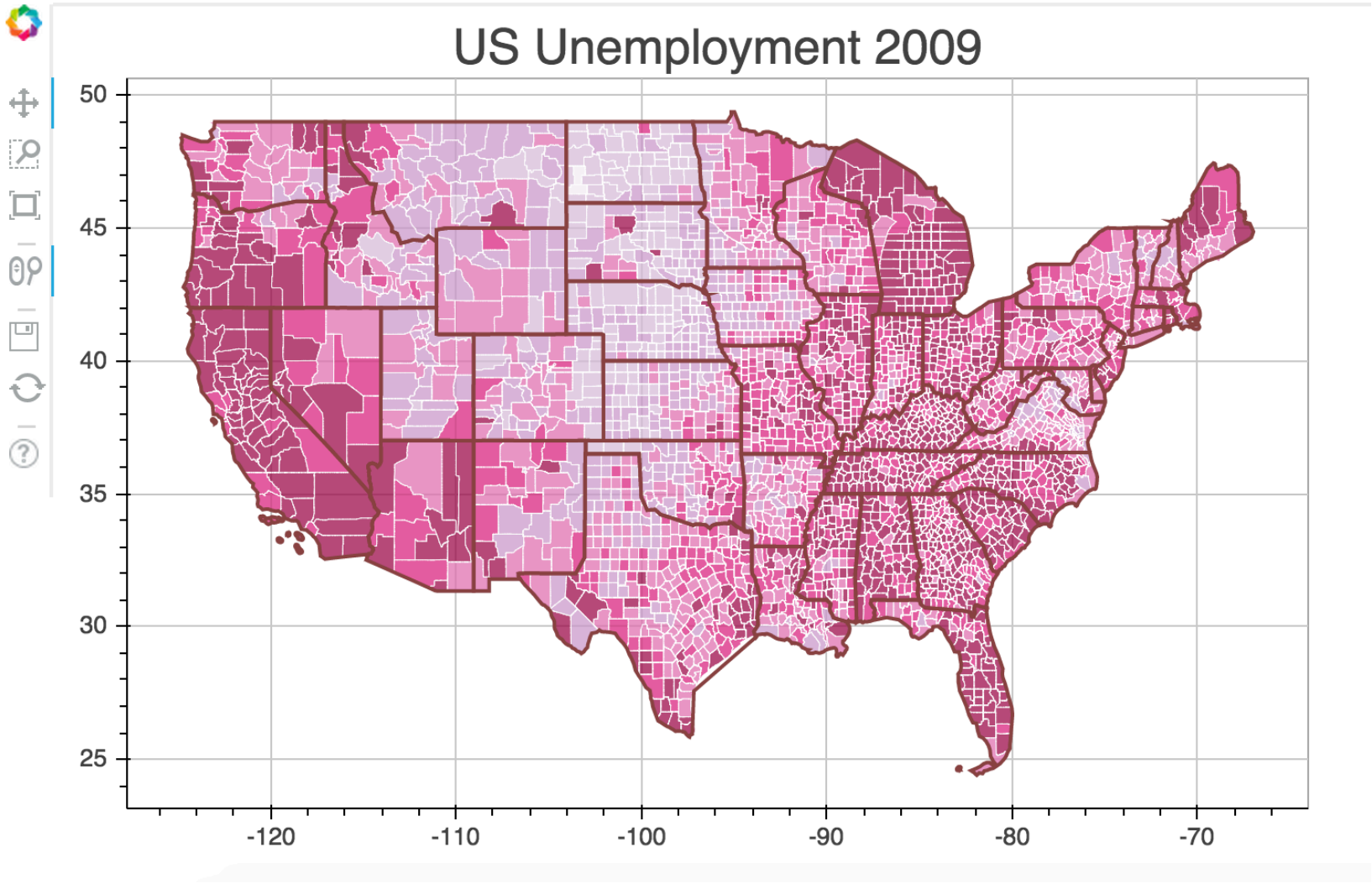




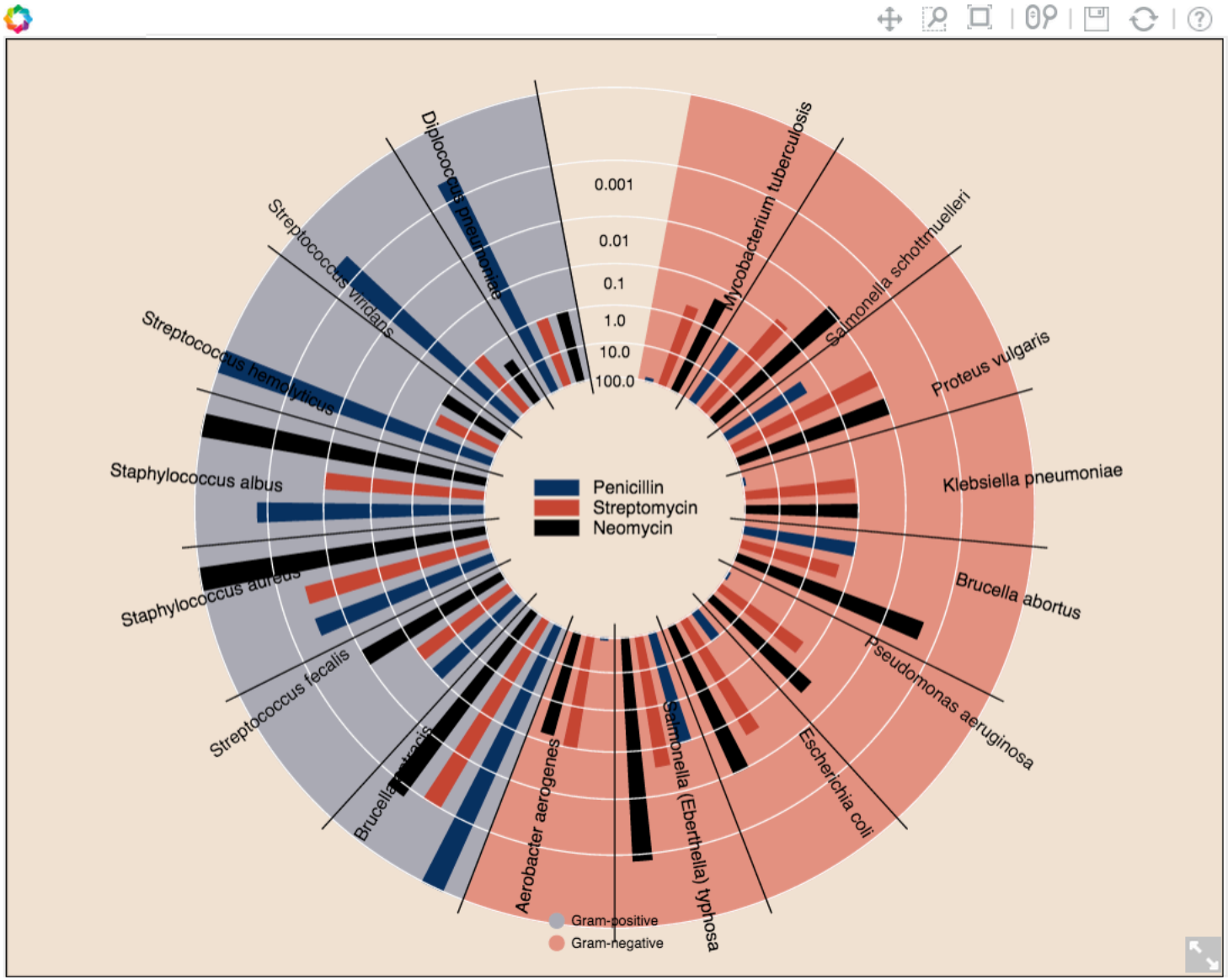


Life's pathetic, go Pythonic! ***(wjo1212@163.com)***

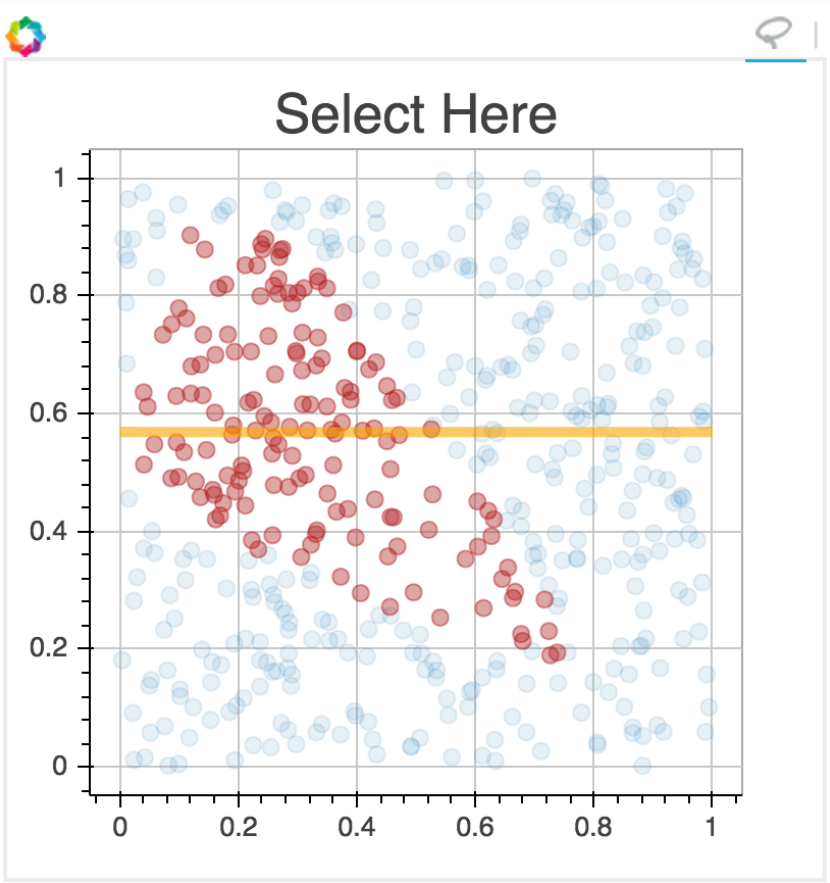
Bokeh Plotting

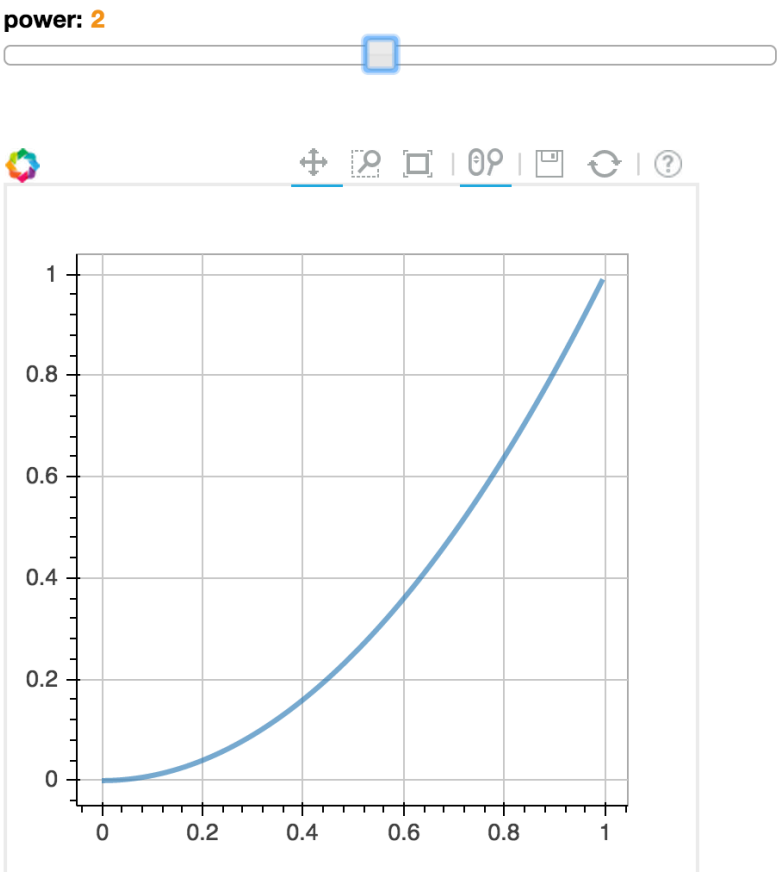


Bokeh Plotting



Bokeh Interaction





Let’s talk about Bigger Data

Bigger Data Handling

- Pandas with Chunk/Multiprocess (like Dask):

```
import pandas as pd
pd.read_csv(..., chumks=10000000)
```

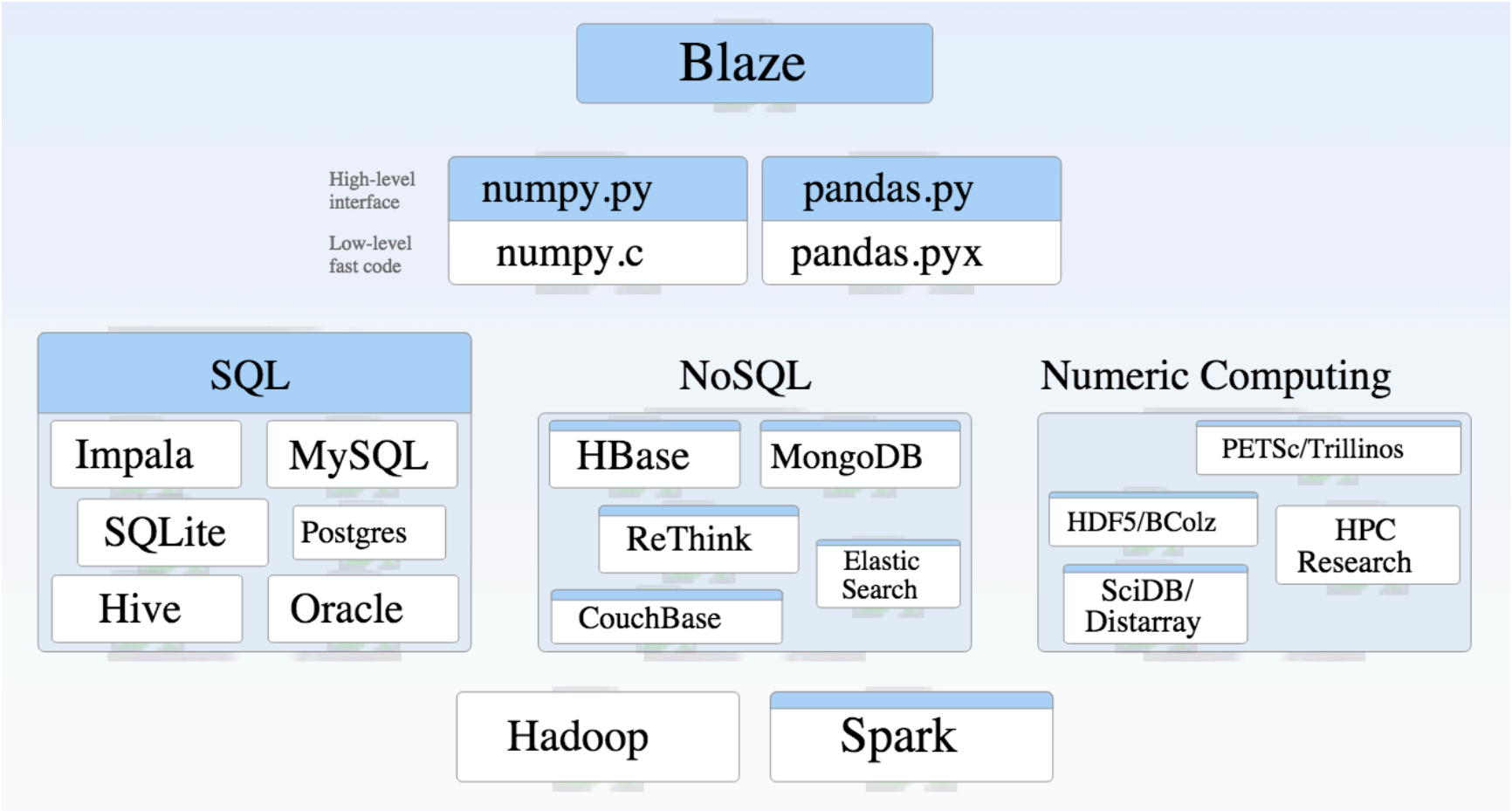
- Blaze with Database:

```
from blaze import Data
df = Data("sqlite:///larg.db::dta")
```

- Spark with distribution and then back to Pandas:

```
sqlContext = SQLContext(sc)
ddf = sqlContext.read.json("...")
df = ddf.toPandas()
```

Blaze


Blaze
High-level interface
Low-level fast code
numpy.py
numpy.c
pandas.py
pandas.pyx
SQL
Impala
MySQL
SQLite
Postgres
Hive
Oracle
NoSQL
HBase
MongoDB
ReThink
Elastic Search
CouchBase
Numeric Computing
PETSc/Trillinos
HDF5/BColz
HPC Research
SciDB/ Distarray
Hadoop
Spark

Blaze computing logic

```
from blaze.compute import compute
from blaze.expr import Symbol
```

```
bank = Symbol('bank',
    'var * {id:int, name:string, balance:int}')

deadbeats = bank[bank.balance < 0].name
```

Blaze with Python

```
L = [[1, 'Alice',   100],
     [2, 'Bob',    -200],
     [3, 'Charlie', 300],
     [4, 'Dennis',  400],
     [5, 'Edith',  -500]]


list(compute(deadbeats, L)
['Bob', 'Edith']
```

Blaze with Pandas

- Re-use the computing logic

```
df = DataFrame([[1, 'Alice',   100],
                [2, 'Bob',    -200],
                [3, 'Charlie', 300],
                [4, 'Dennis',  400],
                [5, 'Edith',  -500]],
               columns=['id', 'name', 'balance'])

list(compute(deadbeats, df)
1       Bob
4     Edith
```

Blaze with Spark

- Re-use the computing logic

```
import pyspark

sc = pyspark.SparkContext('local', 'Blaze-demo')
rdd = odo(sc, L)

compute(deadbeats, rdd).collect()
['Bob', 'Edith']
```

Why Blaze?

- Write once, run anywhere
- Scalable development
  - (start with CSV files, end with Impala/Spark)
- Rapid prototyping
  - (try Postgres, MongoDB, Spark, see what suits you best)
- Cross-backend query optimization

Why not Blaze?

- Interfaces and functions are quite limited
- Performance is not so good in some scenarios
- Still in early stage. Interfaces keep big changes

Spark already shipped Data Frame API!

About Dask

- Another version of Pandas provides similar interfaces but handled on blocked algorithms in parallel on a single machine.
- Interfaces are limited comparing to Pandas.
- Immature and still in early stage

Demo Spark/Pandas Data Frame with visualization

Now you know

- Python data science ecosystem
- Data Wrangling , Analysis and Visualization with
  - Pandas
  - Matplotlib
  - SeaBorn
  - Bokeh
- Bigger Data Consideration:
  - Blaze
  - Spark

Thank you!

人生苦短 python当歌

Life's pathetic, go Pythonic! (wjo1212@163.com)

Contact me

- Email: wjo1212 at 163.com
- Wechat: LaiQiangDing