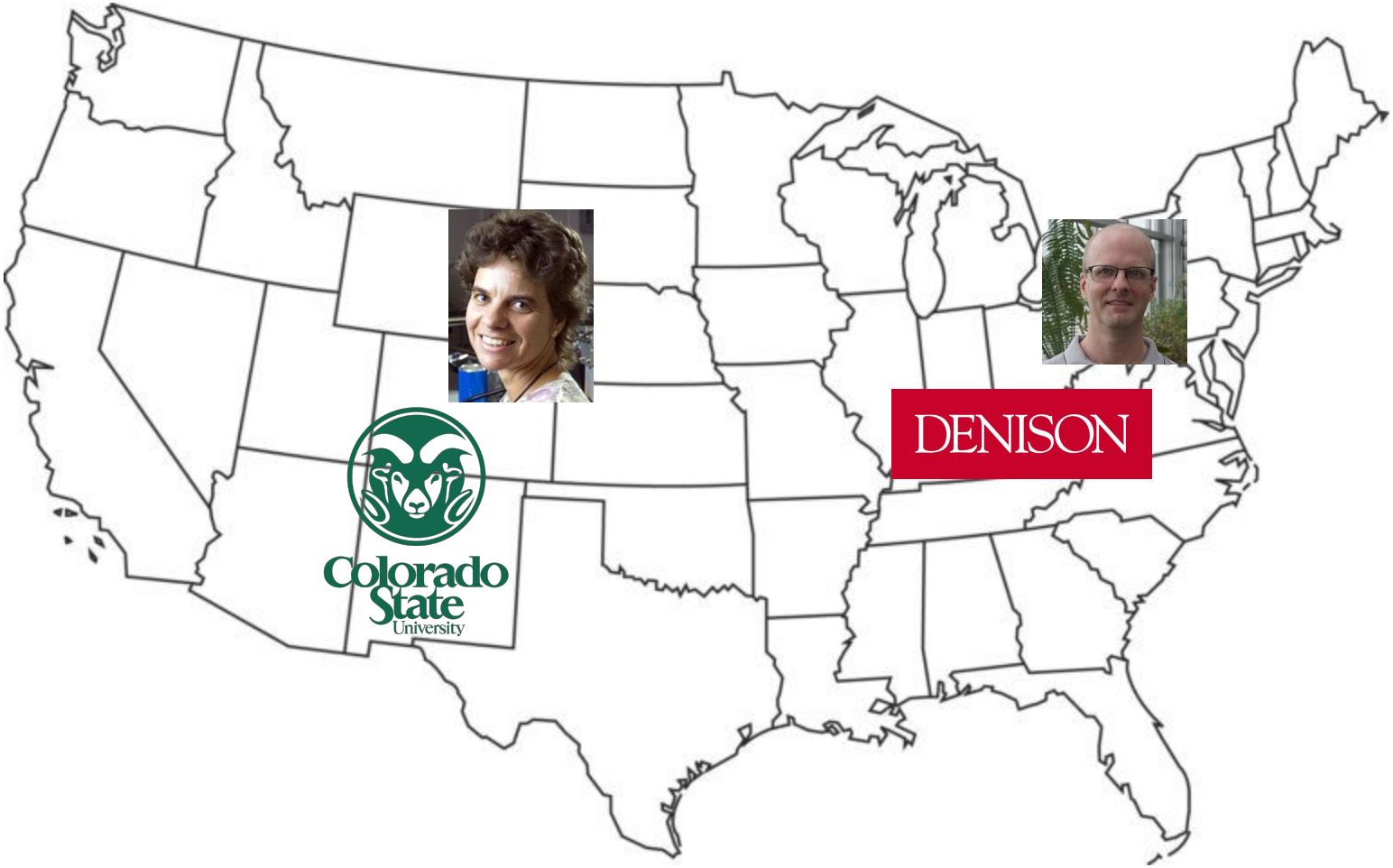
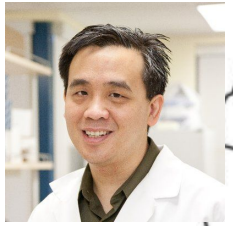
BD2K Summer Workshop 2018

Jacob Pfeil
Graduate Student


Colorado State University
DENISON


RUSSIA
Lake Baikal
Amur
KAZAKHSTAN
Lake Balkhash
MONGOLIA
Harbin
NORTH PACIFIC OCEAN
Urumqi
Turpan Pendi
KYR.
TAJ.
Shenyang
Sea of Japan
JAPAN
N. KOR.
BEIJING
Tianjin
Huang He
S. KOR.
Indian claim
Indus
Lanzhou
Zhengzhou
Yellow Sea
Xi'an
Zhangjiagang
Nanjing
Shanghai
Chinese line of control
Brahmaputra
Mekong
Lhasa
Chengdu
Wuhan
East China Sea
Philippine Sea
NEPAL
Mount Everest
Chongqing
Chang Jiang
Ganges
BHU.
Taipei
INDIA
Guangzhou
Taiwan
Hong Kong S.A.R.
Macau S.A.R.
BANGL.
BURMA
VIET.
LAOS
Hainan Dao
South China Sea
PHIL.
THAI.
0 400 800 km
0 400 800 mi



UCSF

DENISON

UC SANTA CRUZ

BANANA SLUGS

Colorado State University

Molecular Heterogeneity in Patient Population Leads to Differential Treatment Responses

Drug is Toxic and Not Beneficial

Patient Population

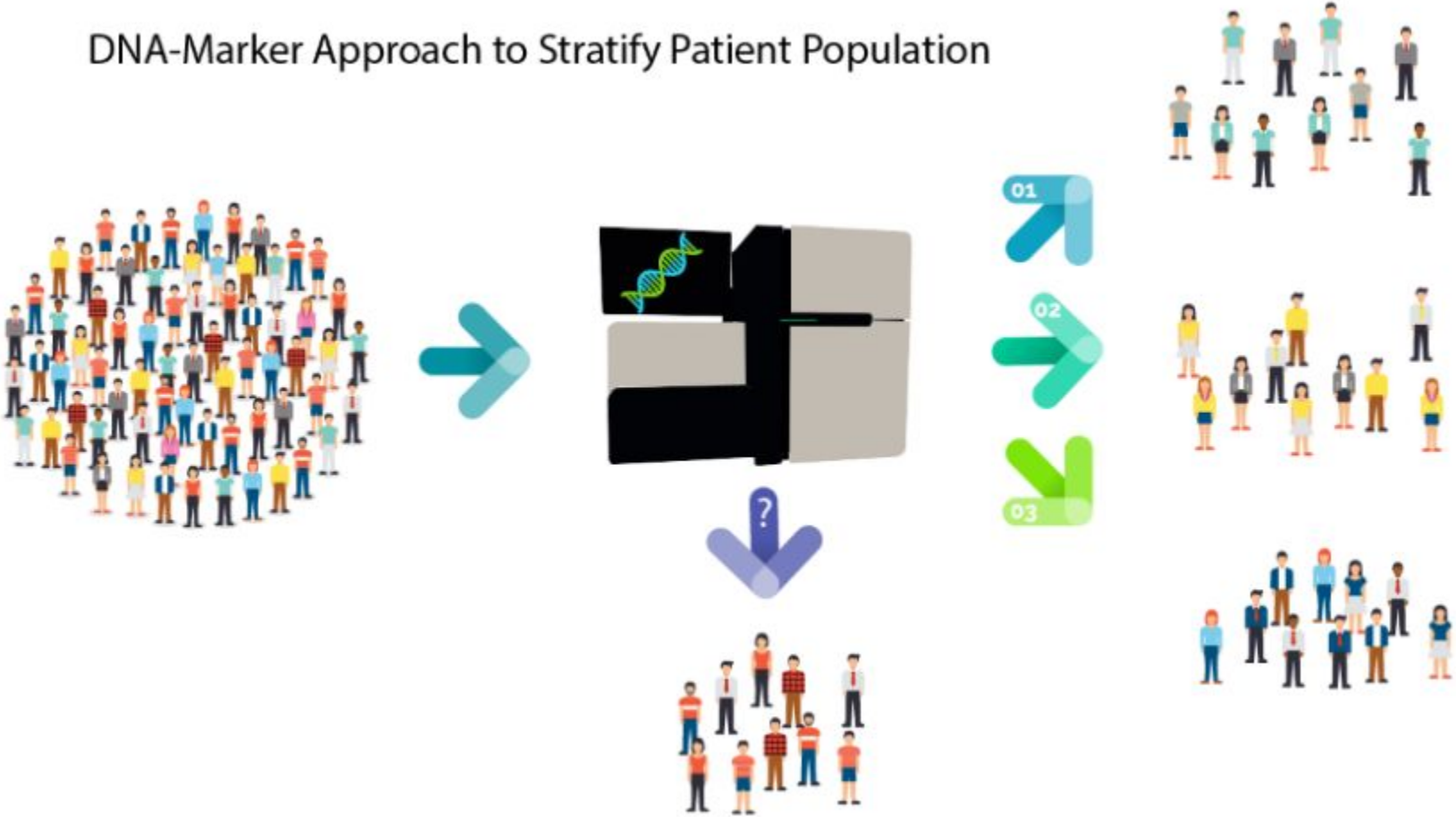
Same Diagnosis, Same Drug

Drug is Not Toxic and Not Beneficial

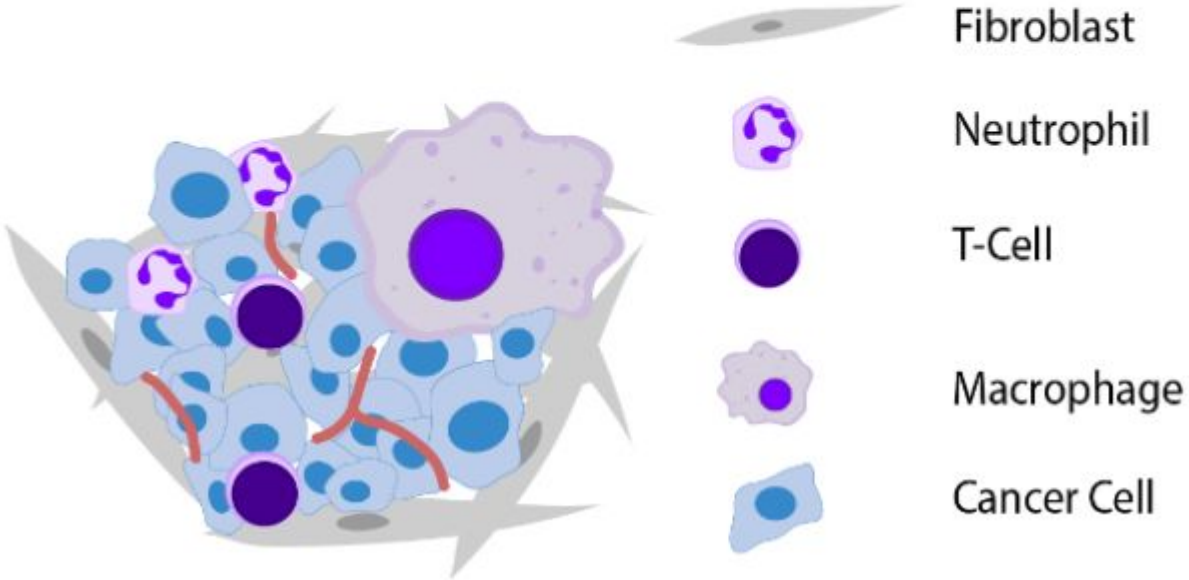
Drug is Toxic but Beneficial

Drug is Not Toxic and Beneficial

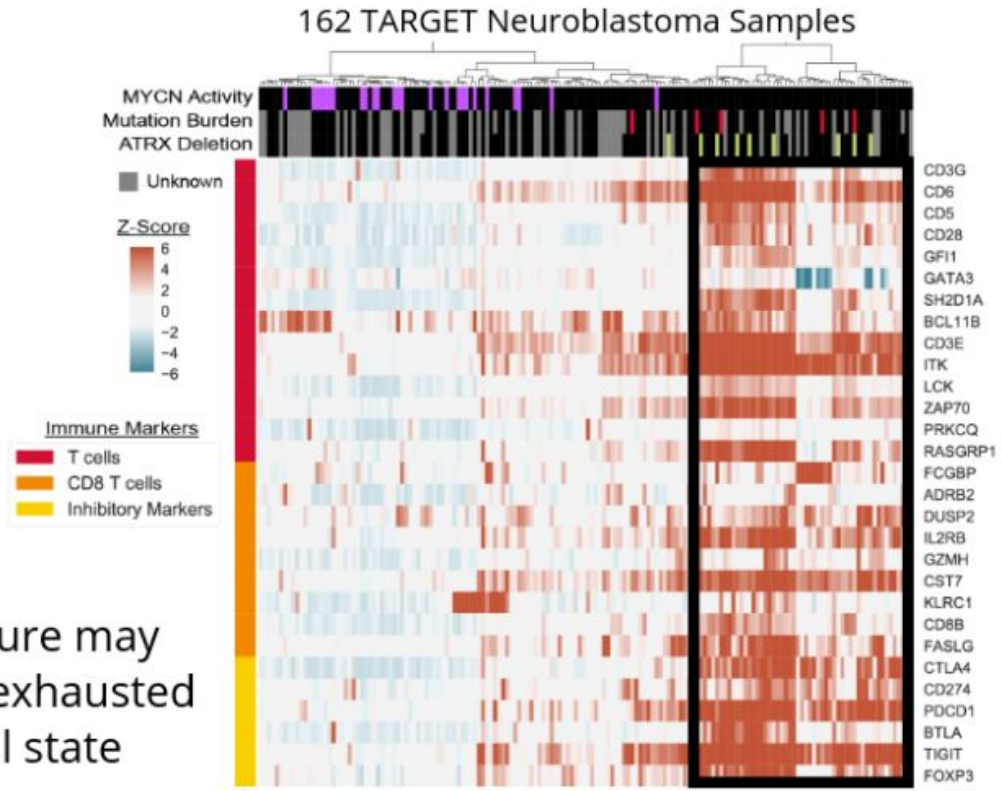
DNA-Marker Approach to Stratify Patient Population



TME Influences Tumor Progression and Response to Therapies



Z-Score Heatmap for T-Cell Profiling Genes and Inhibitory Markers



Signature may predict exhausted T-cell state

Lyons, Y. A., Wu, S. Y., Overwijk, W. W., Baggerly, K. A., & Sood, A. K. (2017). Immune cell profiling in cancer: molecular approaches to cell-specific identification. *npj Precision Oncology*, *1*(1), 26.

TREEHOUSE TEAM

Isabel Bjork
Director

Ann Durbin
Data Coordinator

Ellen Kephart
Software Engineer

Sofie Salama
Research Scientist

David Haussler
Scientific Director

Katrina Learned
Data Coordinator

Rob Currie
CTO

Holly Beale
Lead Computational Biologist

Lauren Sanders
Graduate Student Researcher

Olena Morozova
Scientific Lead

Basic Navigation

Basic Navigation

```
[Workshop-2018] ls
data  images  notebooks  outline.md  README.md  rosalind
```

```
[Workshop-2018] cd notebooks
[notebooks] ls
matplotlib.ipynb  numpy.ipynb  pandas.ipynb  seaborn.ipynb
```

Creating and Removing Directories

```
[Workshop-2018] mkdir tmp
[Workshop-2018] ls
data  images  notebooks  outline.md  README.md  rosalind  tmp
```

```
[Workshop-2018] rm -rf tmp
[Workshop-2018] ls
data  images  notebooks  outline.md  README.md  rosalind
```

CONDA

conda install <package>
conda update <package>
conda remove <package>

```
[~] ls miniconda3
bin              envs     lib64        mkspecs      qml          translations
compiler_compat  etc      libexec      phrasebooks  resources    var
conda-meta       include  LICENSE.txt  pkgs         share        x86_64-conda_cos6-linux-gnu
doc              lib      man          plugins      ssl
```

https://conda.io/miniconda.html

	Windows	Mac OS X	Linux
Python 3.6	64-bit (exe installer) 32-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer) 32-bit (bash installer)
Python 2.7	64-bit (exe installer) 32-bit (exe installer)	64-bit (bash installer)	64-bit (bash installer) 32-bit (bash installer)

Python REPL

```
[~] python
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 18:21:58)
[GCC 7.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

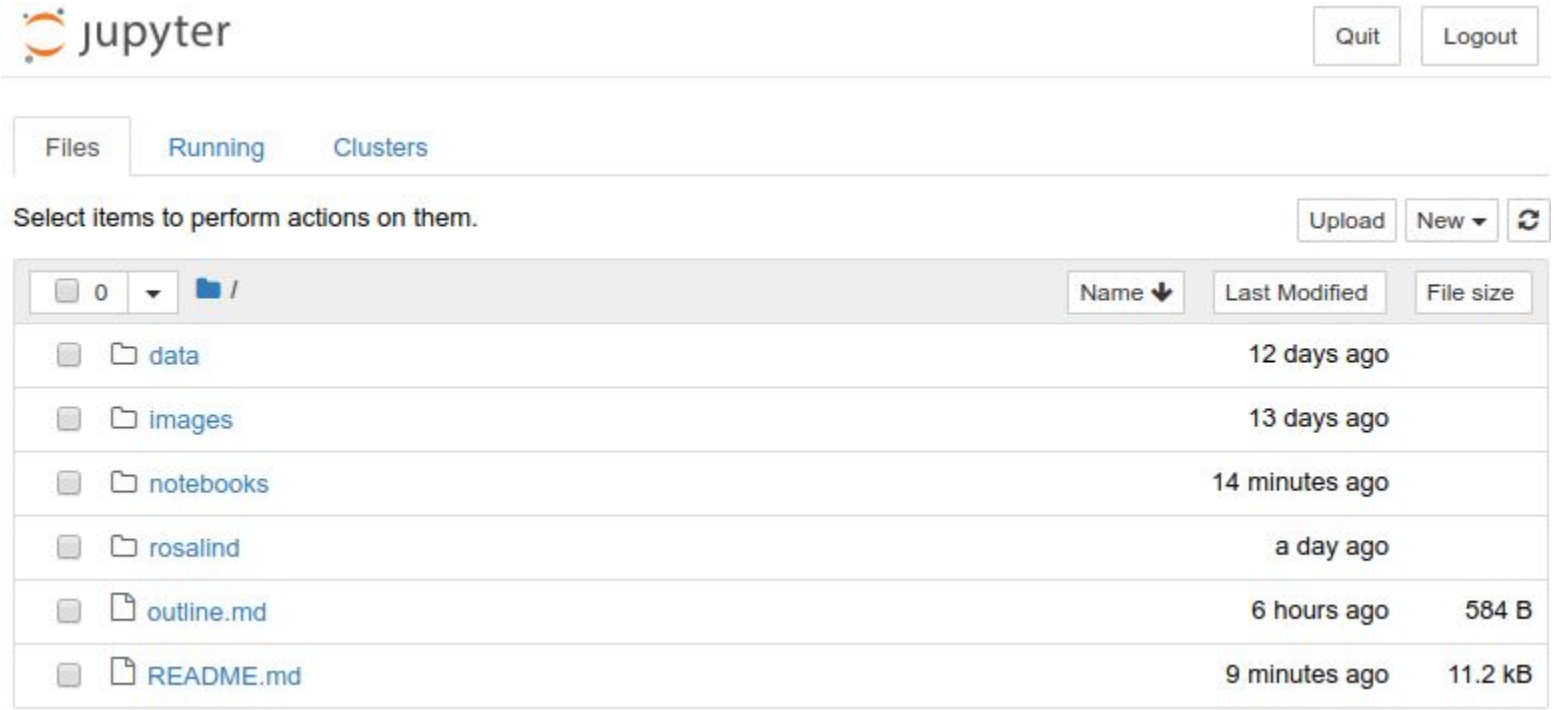
ctrl-D to exit

Interactive Python Basics

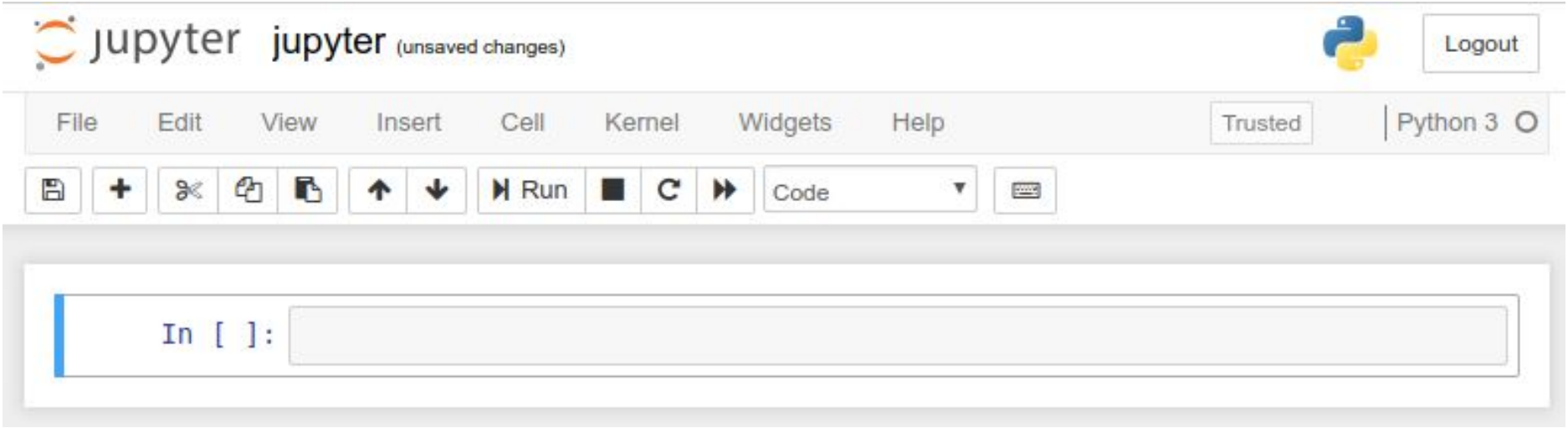
iPython

```
[Workshop-2018] ipython
Python 3.6.5 |Anaconda, Inc.| (default, Mar 29 2018, 18:21:58)
Type 'copyright', 'credits' or 'license' for more information
IPython 6.4.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: ls
data/  images/  notebooks/  outline.md  README.md  rosalind/
```

$ jupyter notebook



Jupyter Notebook



jupyter jupyter (unsaved changes)

Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

```
# Markdown

1. Great for documenting your code
2. Use it like you would a lab notebook
```

Markdown

1. Great for documenting your code
2. Use it like you would a lab notebook

In [ ]:

In [1]:
```
def hello_world():
    print("hello, world!")
```

In [2]:
```
hello_world()
```

```
hello, world!
```

In [ ]:
```
|
```

NumPy: The Scientific Computing Library

NumPy

Library for scientific computing

- N-dimensional array object
- linear algebra functions
- random number generator

Array Operations

```
>>> import numpy as np
>>> a = np.arange(15).reshape(3, 5)
>>> a
array([[ 0,  1,  2,  3,  4],
       [ 5,  6,  7,  8,  9],
       [10, 11, 12, 13, 14]])
```

```
>>> a.shape
(3, 5)
```

```
>>> a.dtype
dtype('int64')
```

```
>>> a = np.array(1,2,3,4)    # WRONG
>>> a = np.array([1,2,3,4])  # RIGHT
```

```
>>> b = np.array([(1.5,2,3), (4,5,6)])
>>> b
array([[ 1.5,  2. ,  3. ],
       [ 4. ,  5. ,  6. ]])
```

```
v1 = np.array([1, 2, 3])
v2 = np.array([4, 5, 6])
v3 = np.array([7, 8, 9])
M = np.vstack([v1, v2, v3])
print M
>>> [[1 2 3]
     [4 5 6]
     [7 8 9]]
```

https://web.stanford.edu/class/cs231a/section/section1.pdf

```
>>> np.zeros( (3,4) )
array([[ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.],
       [ 0.,  0.,  0.,  0.]])
>>> np.ones( (2,3,4), dtype=np.int16 )                # dtype can also be specified
array([[[ 1, 1, 1, 1],
        [ 1, 1, 1, 1],
        [ 1, 1, 1, 1]],
       [[ 1, 1, 1, 1],
        [ 1, 1, 1, 1],
        [ 1, 1, 1, 1]]], dtype=int16)
>>> np.empty( (2,3) )                                 # uninitialized, output may vary
array([[  3.73603959e-262,   6.02658058e-154,   6.55490914e-260],
       [  5.30498948e-313,   3.14673309e-307,   1.00000000e+000]])
```

Useful functions

```
>>> from numpy import pi
>>> np.linspace( 0, 2, 9 )                  # 9 numbers from 0 to 2
array([ 0.  ,  0.25,  0.5 ,  0.75,  1.  ,  1.25,  1.5 ,  1.75,  2.  ])
>>> x = np.linspace( 0, 2*pi, 100 )         # useful to evaluate function at lots of points
>>> f = np.sin(x)
```

```
>>> B = np.arange(3)
>>> B
array([0, 1, 2])
>>> np.exp(B)
array([ 1.        ,  2.71828183,  7.3890561 ])
>>> np.sqrt(B)
array([ 0.        ,  1.        ,  1.41421356])
>>> C = np.array([2., -1., 4.])
>>> np.add(B, C)
array([ 2.,  0.,  6.])
```

Matrix Multiplication

$$A_{n\times m} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \vdots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{bmatrix} \quad \mathbf{a}_i$$

$$B_{m\times p} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1p} \\ b_{21} & b_{22} & \dots & b_{2p} \\ \vdots & \vdots & \vdots & \vdots \\ b_{m1} & b_{m2} & \dots & b_{mp} \end{bmatrix}$$

$$\mathbf{b}_j$$

Product:

$$C_{n\times p} = A_{n\times \boxed{m}} B_{\boxed{m}\times p}$$

$$c_{ij} = \mathbf{a}_i \cdot \mathbf{b}_j = \sum_{k=1}^{m} a_{ik} b_{kj}$$

A and B must have compatible dimensions!

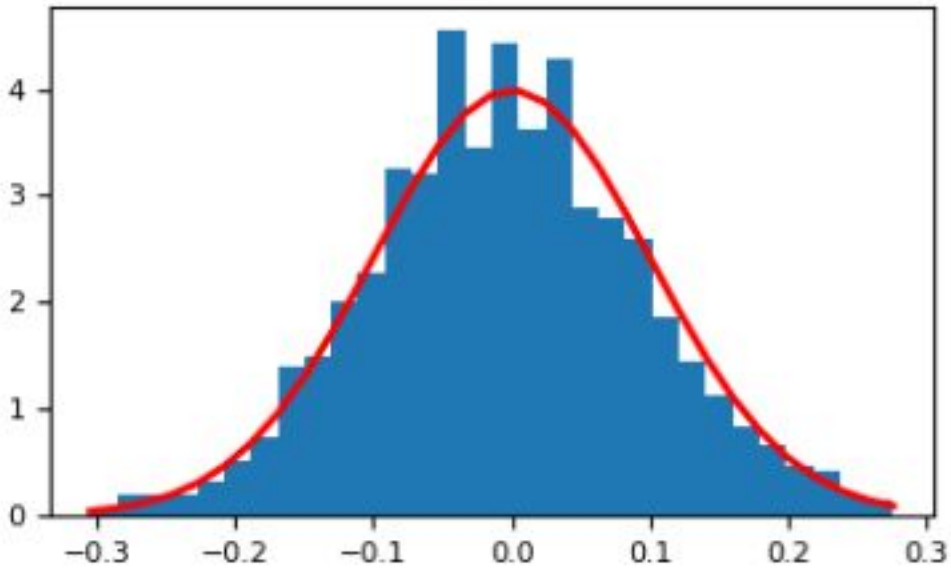
$$A_{n\times n} B_{n\times n} \neq B_{n\times n} A_{n\times n}$$

*Courtesy of last year's slides.

https://web.stanford.edu/class/cs231a/section/section1.pdf

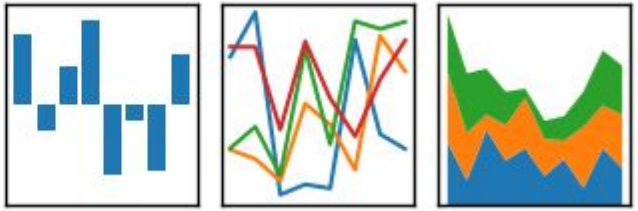
```
>>> A = np.array( [[1,1],
...               [0,1]] )
>>> B = np.array( [[2,0],
...               [3,4]] )
>>> A*B                          # elementwise product
array([[2, 0],
       [0, 4]])
>>> A.dot(B)                     # matrix product
array([[5, 4],
       [3, 4]])
>>> np.dot(A, B)                 # another matrix product
array([[5, 4],
       [3, 4]])
```

```
>>> mu, sigma = 0, 0.1 # mean and standard deviation
>>> s = np.random.normal(mu, sigma, 1000)
```



pandas

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



Implements a dataframe object

- similar to R dataframe
- commonly used for data analysis
- used in many other libraries (seaborn)

Examples were drawn from:
https://www.hackerearth.com/practice/machine-learning/data-manipulation-visualisation-r-python/tutorial-data-manipulation-numpy-pandas-python/tutorial/

How to create a dataframe

```
#create a data frame - dictionary is used here where keys get converted to column names and values to row values.
data = pd.DataFrame({'Country': ['Russia','Colombia','Chile','Equador','Nigeria'],
                     'Rank':[121,40,100,130,11]})
data
```

Columns

Index

	Country	Rank
0	Russia	121
1	Colombia	40
2	Chile	100
3	Equador	130
4	Nigeria	11

How to create a dataframe

```
dates = pd.date_range('20130101',periods=6)
df = pd.DataFrame(np.random.randn(6,4),index=dates,columns=list('ABCD'))
df
```

	A	B	C	D
2013-01-01	1.030816	-1.276989	0.837720	-1.490111
2013-01-02	-1.070215	-0.209129	0.604572	-1.743058
2013-01-03	1.524227	1.863575	1.291378	1.300696
2013-01-04	0.918203	-0.158800	-0.964063	-1.990779
2013-01-05	0.089731	0.114854	-0.585815	0.298772
2013-01-06	0.222260	0.435183	-0.045748	0.049898

How to access items

```
#slice based on date range
df['20130101':'20130104']
```

	A	B	C	D
2013-01-01	1.030816	-1.276989	0.837720	-1.490111
2013-01-02	-1.070215	-0.209129	0.604572	-1.743058
2013-01-03	1.524227	1.863575	1.291378	1.300696
2013-01-04	0.918203	-0.158800	-0.964063	-1.990779

How to access items

```
#slicing based on column names
df.loc[:,['A','B']]
```

	A	B
2013-01-01	1.030816	-1.276989
2013-01-02	-1.070215	-0.209129
2013-01-03	1.524227	1.863575
2013-01-04	0.918203	-0.158800
2013-01-05	0.089731	0.114854
2013-01-06	0.222260	0.435183

How to access items by the index and columns

```
#slicing based on both row index labels and column names
df.loc['20130102':'20130103',['A','B']]
```

	A	B
2013-01-02	-1.070215	-0.209129
2013-01-03	1.524227	1.863575

How to add a new row

```
df = pd.DataFrame(columns=['value'])

for _ in range(5):
    df.loc[len(df), :] = [np.random.normal(0, 1)]

df
```

	value
0	-0.884526
1	0.304444
2	-0.288040
3	0.398078
4	1.006034

How to access items by numerical indices

```
#returns a specific range of rows
df.iloc[2:4, 0:2]
```

	A	B
2013-01-03	1.524227	1.863575
2013-01-04	0.918203	-0.158800

How to access items using a boolean

```
df[df.A > 1]
```

	A	B	C	D
2013-01-01	1.030816	-1.276989	0.837720	-1.490111
2013-01-03	1.524227	1.863575	1.291378	1.300696

How to assign values to a column or row

```
df2['E']=['one', 'one','two','three','four','three']
df2
```

	A	B	C	D	E
2013-01-01	1.030816	-1.276989	0.837720	-1.490111	one
2013-01-02	-1.070215	-0.209129	0.604572	-1.743058	one
2013-01-03	1.524227	1.863575	1.291378	1.300696	two
2013-01-04	0.918203	-0.158800	-0.964063	-1.990779	three
2013-01-05	0.089731	0.114854	-0.585815	0.298772	four
2013-01-06	0.222260	0.435183	-0.045748	0.049898	three

How to subset df by overlap with a list

```
#select rows based on column values
df2[df2['E'].isin(['two','four'])]
```

	A	B	C	D	E
2013-01-03	1.524227	1.863575	1.291378	1.300696	two
2013-01-05	0.089731	0.114854	-0.585815	0.298772	four

How to access items using multiple booleans

```
titles[(titles.year < 1959) & (titles.year > 1955)].sort_values('year').head(2)
```

	title	year
225480	Tischlein, deck dich	1956
64366	Yield to the Night	1956

Wide versus Long Format

name	George	Lisa	Michael
date			
2000-01-03	500.0	450.0	200.0
2000-01-04	450.0	448.0	180.5
2000-01-05	420.0	447.0	177.0
2000-01-06	300.0	344.6	150.0

```
In  : df
Out :
```

	date	person	dollars
0	2000-01-03	Michael	200
1	2000-01-03	George	500
2	2000-01-03	Lisa	450
3	2000-01-04	Michael	180.5
4	2000-01-04	George	450
5	2000-01-04	Lisa	448
6	2000-01-05	Michael	177
7	2000-01-05	George	420
8	2000-01-05	Lisa	447
9	2000-01-06	Michael	150
10	2000-01-06	George	300
11	2000-01-06	Lisa	344.6

In [ ]:

```
df2 = pd.DataFrame(columns=["ID", "Group", "Value"])
for index, row in df.iterrows():
    value = row["Value"]        # May need the .item() method
    df2.loc[len(df2), :] = [index, group, value]
```

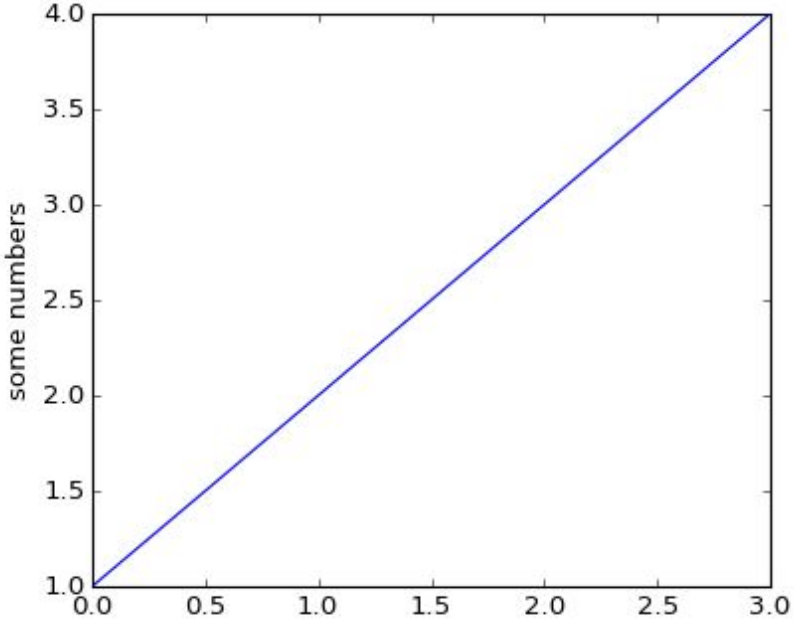
matplotlib

2D plotting library

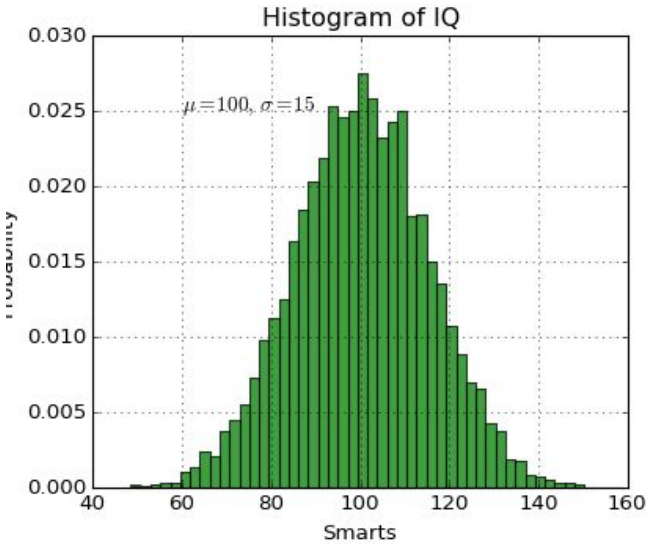
- similar to matlab plotting functions
- flexible environment for making figures
- creates publication quality figures

```
import matplotlib.pyplot as plt
plt.plot([1,2,3,4])
plt.ylabel('some numbers')
plt.show()
```

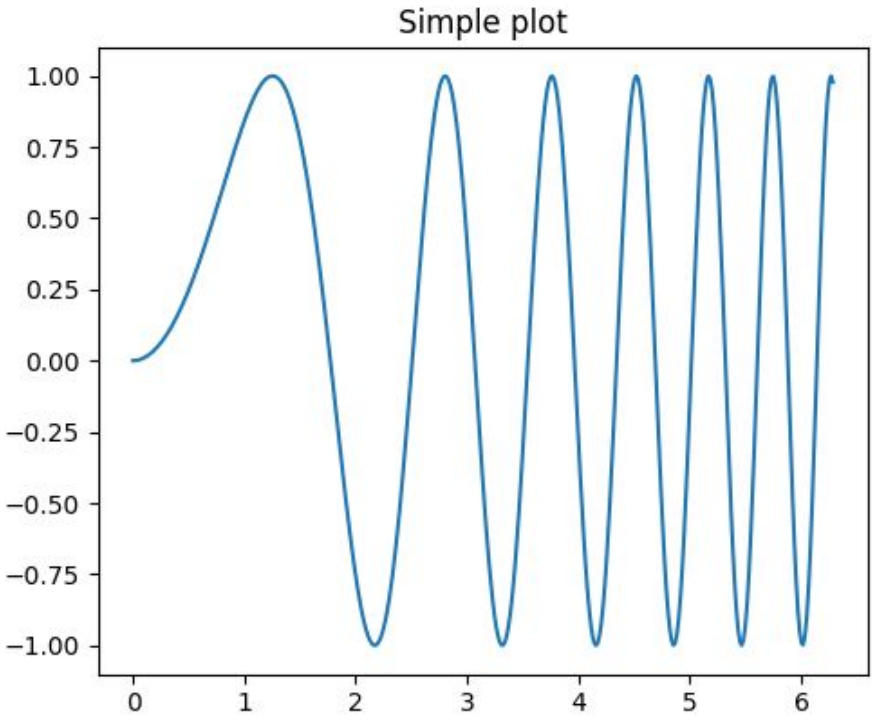
(Source code, png, hires.png, pdf)



```python
import numpy as np
import matplotlib.pyplot as plt

mu, sigma = 100, 15
x = mu + sigma * np.random.randn(10000)

# the histogram of the data
n, bins, patches = plt.hist(x, 50, normed=1, facecolor='g', alpha=0.75)


plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title('Histogram of IQ')
plt.text(60, .025, r'$\mu=100,\ \sigma=15$')
plt.axis([40, 160, 0, 0.03])
plt.grid(True)
plt.show()
```

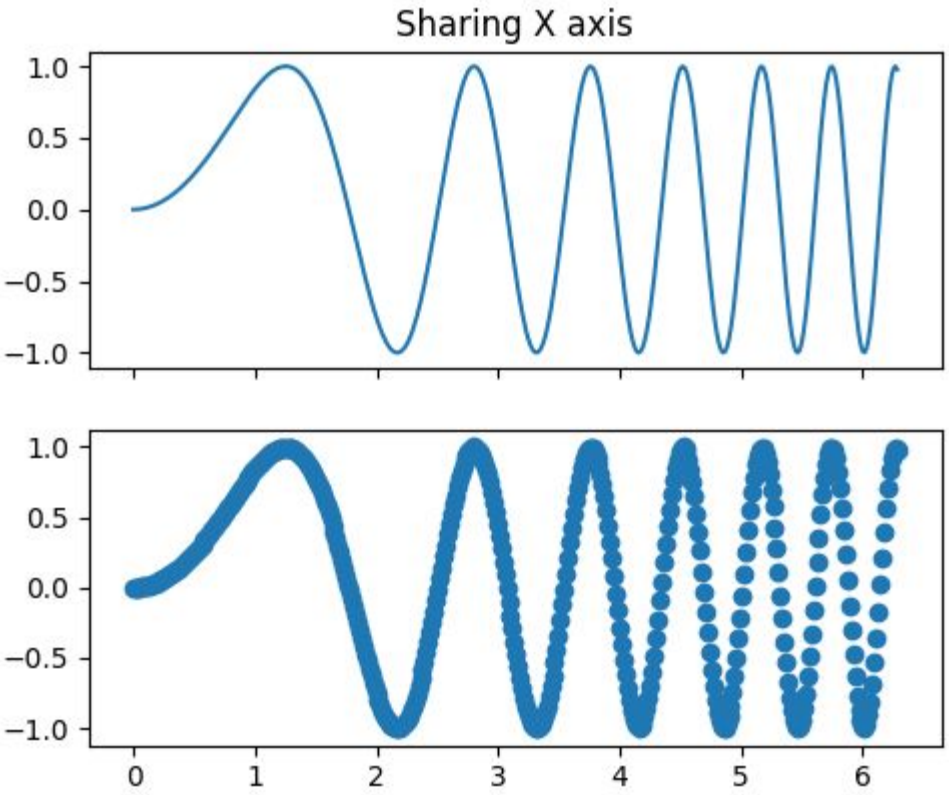


```python
import matplotlib.pyplot as plt
import numpy as np

# Simple data to display in various forms
x = np.linspace(0, 2 * np.pi, 400)
y = np.sin(x ** 2)

plt.close('all')

# Just a figure and one subplot
f, ax = plt.subplots()
ax.plot(x, y)
ax.set_title('Simple plot')
```

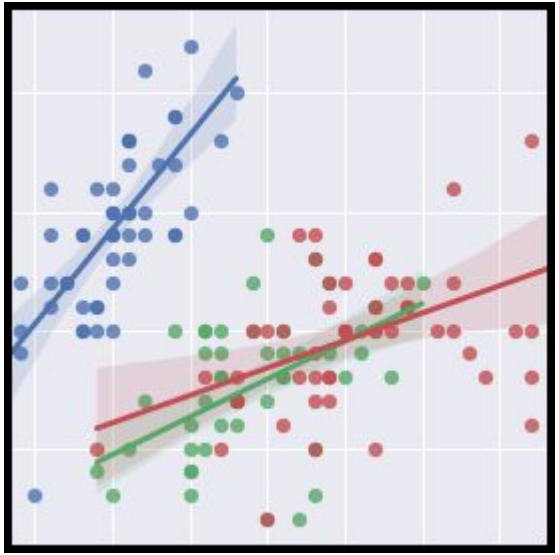


```python
import matplotlib.pyplot as plt
import numpy as np

# Simple data to display in various forms
x = np.linspace(0, 2 * np.pi, 400)
y = np.sin(x ** 2)
```

```python
# Two subplots, the axes array is 1-d
f, axarr = plt.subplots(2, sharex=True)
axarr[0].plot(x, y)
axarr[0].set_title('Sharing X axis')
axarr[1].scatter(x, y)
```



seaborn



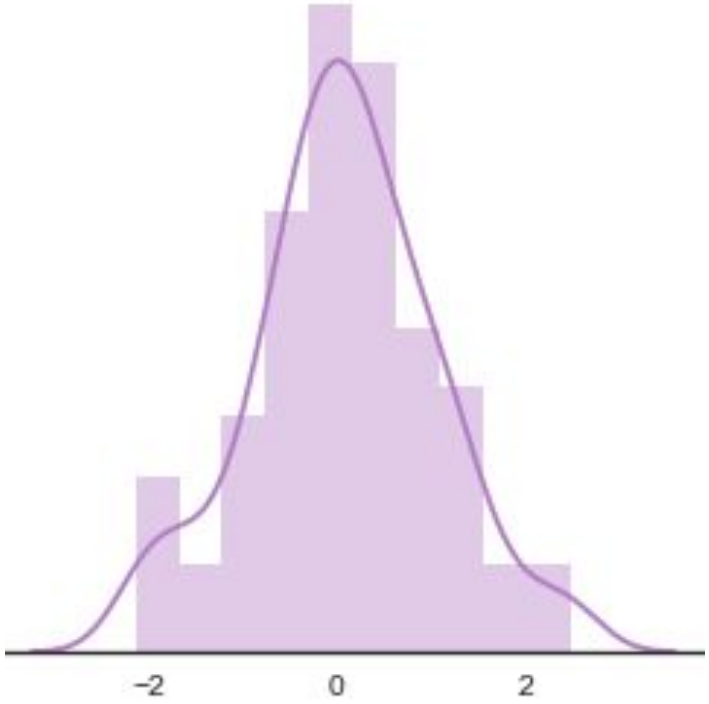
2D plotting library

- built on top of matplotlib
- figures look better with less work
- expanded set of plotting functions

distplot

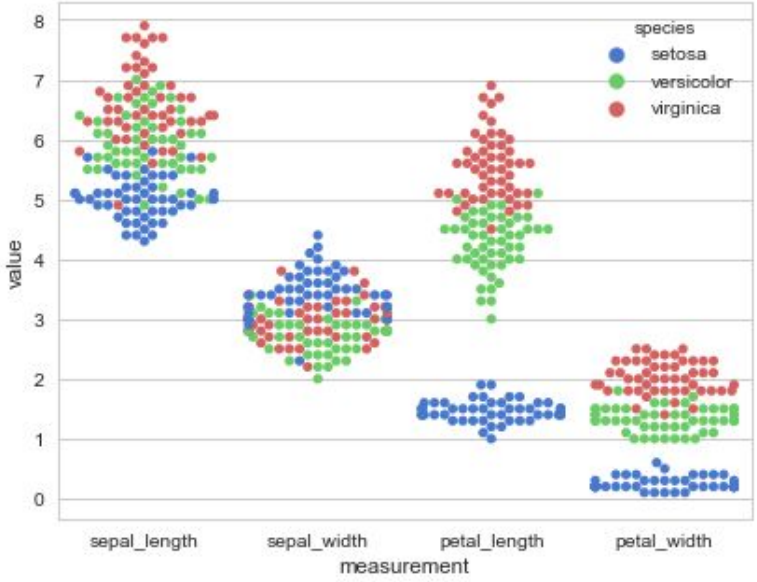
```
# Generate a random univariate dataset
d = rs.normal(size=100)
```

```
# Plot a historgram and kernel density estimate
sns.distplot(d, color="m", ax=axes[1, 1])
```



swarmplot

```
# Draw a categorical scatterplot to show each observation
sns.swarmplot(x="measurement", y="value", hue="species", data=iris)
```



Wide versus Long Format

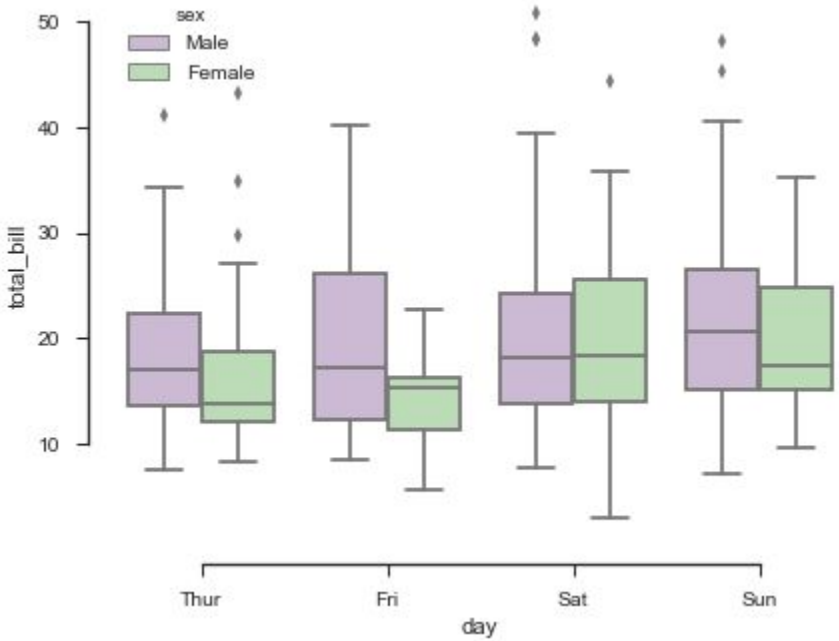
name	George	Lisa	Michael
date			
2000-01-03	500.0	450.0	200.0
2000-01-04	450.0	448.0	180.5
2000-01-05	420.0	447.0	177.0
2000-01-06	300.0	344.6	150.0

```
In  : df
Out :
```

	date	person	dollars
0	2000-01-03	Michael	200
1	2000-01-03	George	500
2	2000-01-03	Lisa	450
3	2000-01-04	Michael	180.5
4	2000-01-04	George	450
5	2000-01-04	Lisa	448
6	2000-01-05	Michael	177
7	2000-01-05	George	420
8	2000-01-05	Lisa	447
9	2000-01-06	Michael	150
10	2000-01-06	George	300
11	2000-01-06	Lisa	344.6

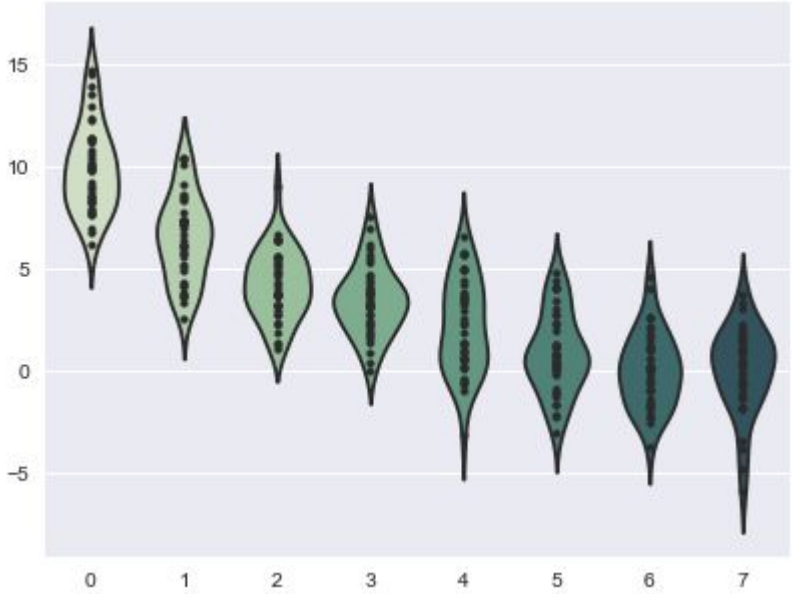
boxplot

```
# Draw a nested boxplot to show bills by day and sex
sns.boxplot(x="day", y="total_bill", hue="sex", data=tips, palette="PRGn")
sns.despine(offset=10, trim=True)
```



violinplot

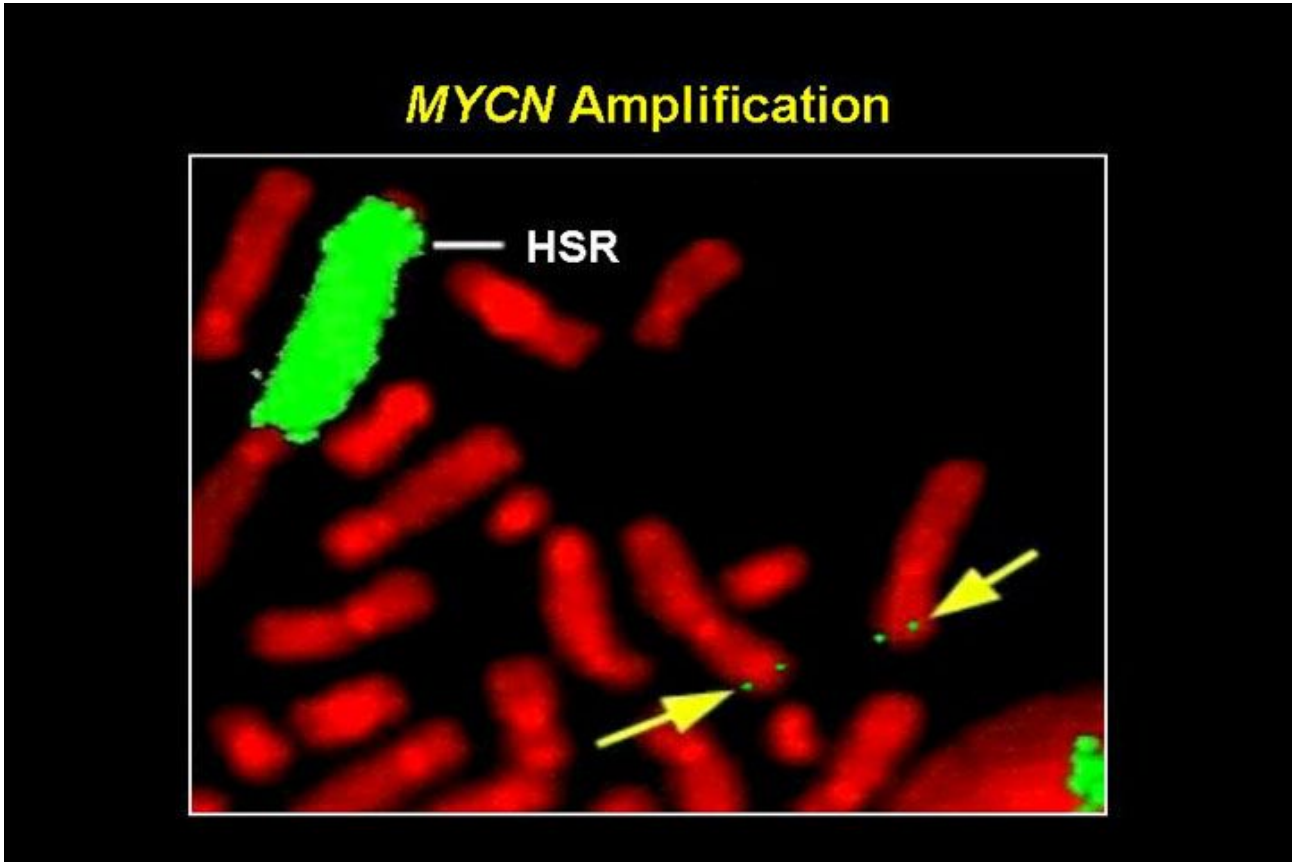
```
# Show each distribution with both violins and points
sns.violinplot(data=d, palette=pal, inner="points")
```



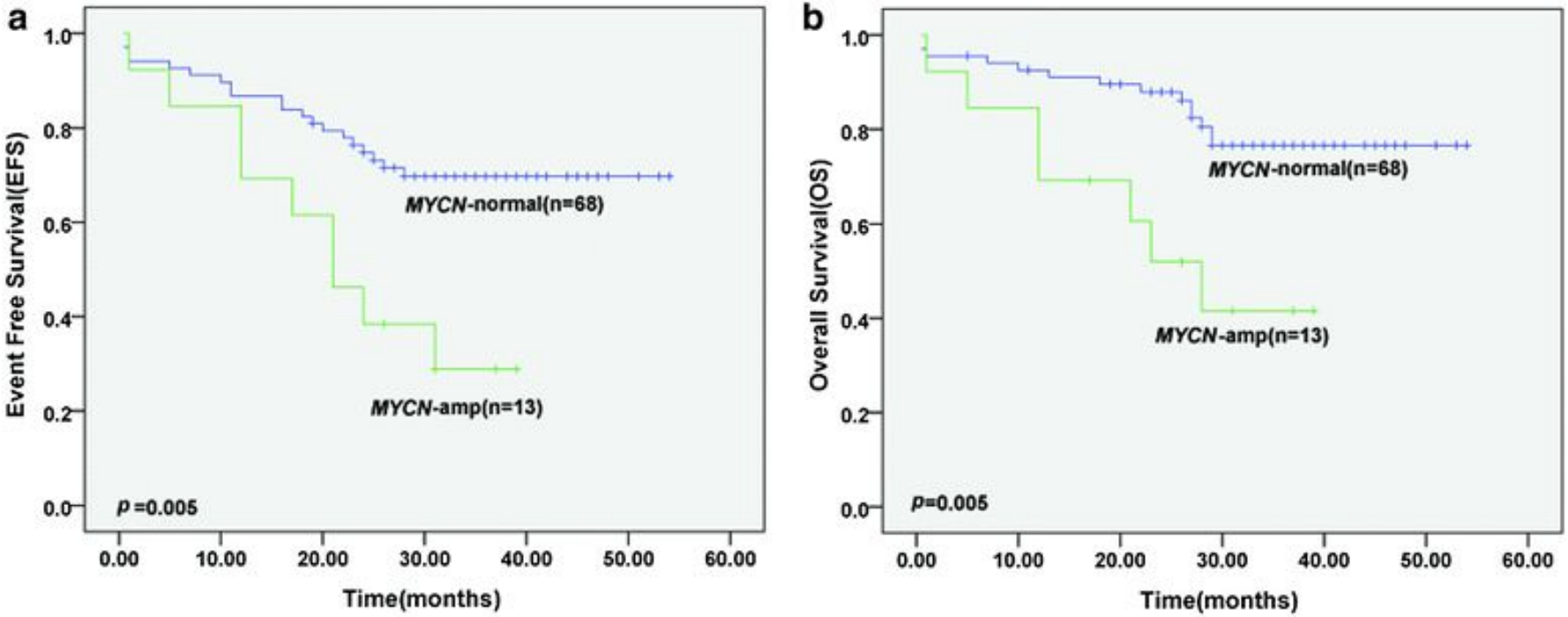
barplot

```
sns.barplot(x, y3, palette="Set3", ax=ax3)
ax3.set_ylabel("Qualitative")
```

MYCN



http://streaming.cineca.it/sestri/courses/cancgen/Maris.htm



https://cancerci.biomedcentral.com/articles/10.1186/s12935-017-0412-z

BD2K Summer Workshop 2018: Day 2

rosalind - [~/code/Workshop-2018/rosalind] - .../rosalind/run.py - PyCharm 2017.2.3

File Edit View Navigate Code Refactor Run Tools VCS Window Help

rosalind 〉 rosalind 〉 run.py 〉

Project

- rosalind ~/code/Workshop-201
  - rosalind
    - bin
    - doc
    - library
    - run.py
  - Dockerfile
  - example.dockerfile
  - README.md
- External Libraries

run.py

```
#! /bin/usr/env python3
import argparse



def main():
    pass


if __name__ == '__main__':
    main()
```

Terminal

```
(base) [rosalind]
```

master

Python Console | Terminal | 6: TODO | Event Log

Dockerfile detection // You may setup Docker deployment run configuration for the following file(s): // /hom... (today 7:00 PM) 1:1 LF UTF-8

```python
#! /bin/usr/env python3
import argparse


def main():
    pass


if __name__ == '__main__':
    main()
```

```python
def example(arg):
    """
    This is an example of how to structure a
    docstring for a function with a single argument.

    :param arg: The argument for the function
    :return: A transformation of the argument
    :rtype: int
    """
    value = code(arg)

    return value
```

Settings

- Appearance & Behavior
- Keymap
- Editor
- Plugins
- Version Control
- Project: rosalind
  - Project Interpreter
  - Project Structure
- Build, Execution, Deployment
- Languages & Frameworks
- Tools
- Vim Emulation

Project: rosalind › Project Interpreter For current project

Project Interpreter: 3.6.2 (~/miniconda3/bin/python3.6)

Package	Version	Latest
Cython	0.28.2	0.28.3
Jinja2	2.10	2.10
MarkupSafe	1.0	1.0
Pillow	5.1.0	5.1.0
PySocks	1.6.8	1.6.8
Pygments	2.2.0	2.2.0
Send2Trash	1.5.0	1.5.0
asn1crypto	0.24.0	0.24.0
backcall	0.1.0	0.1.0
binutils_impl_linux-64	2.28.1	
binutils_linux-64	7.2.0	
blas	1.0	

OK Cancel Apply Help

Settings

- Appearance & Behavior
- Keymap
- Editor
- Plugins
- Version Control
- Project: rosalind
  - Project Interpreter
  - Project Structure
- Build, Execution, Deployment
- Languages & Frameworks
- Tools
- Vim Emulation

Project: rosalind › Project Structure For current project

Mark as: Sources Excluded Templates Resources

- /home/jpfeil/code/Workshop-2018/rosalind
  - .idea
    - inspectionProfiles
  - rosalind
    - bin
    - doc
    - library

+ Add Content Root

/...Workshop-2018/rosalind

Source Folders

rosalind

Exclude files:

Use ; to separate name patterns, * for any number of symbols, ? for one.

OK Cancel Apply Help

```python
#!/usr/bin/env python2.7
import argparse
import bnpy
import os
import logging
import multiprocessing
import numpy as np
import pandas as pd

from collections import defaultdict

from library.utils import mkdir_p, parallel_fit
```

```
[rosalind] cd rosalind/library
[library] ls
__init__.py  util.py
[library]
```

import argparse

```
parser = argparse.ArgumentParser(description='Description of your program')
parser.add_argument('-f','--foo', help='Description for foo argument', required=True)
parser.add_argument('-b','--bar', help='Description for bar argument', required=True)
args = vars(parser.parse_args())
```

```
if args['foo'] == 'Hello':
    # code here

if args['bar'] == 'World':
    # code here
```

https://stackoverflow.com/questions/7427101/simple-argparse-example-wanted-1-argument-3-results

Terminal

```
[hydra] ls
Dockerfile  hydra  README.md
[hydra] cd hydra
[hydra] ls
gene-sets  hydra-out  library  run.py  test
[hydra] ./run.py --expression test/nbl-data.tsv --CPU 7
```

6: TODO 9: Version Control Python Console Terminal

Dockerfile detection // You may setup Docker deployment run configuration for

ROSALIND

Problems

Bioinformatics Stronghold ▾ | List | Tree

Rosalind is a platform for learning bioinformatics and programming through problem solving. Take a tour to get the hang of how Rosalind works.

Last win: rlyluann vs. "Enumerating Gene Orders", 9 minutes ago

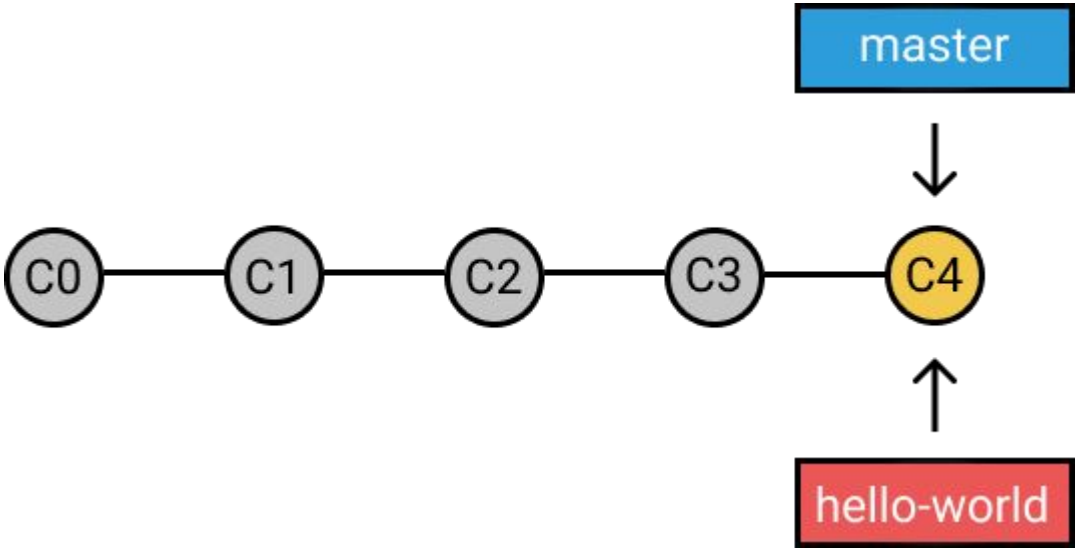
Problems: 285 (total), users: 55565, attempts: 934705, correct: 525016

ID	Title	Solved By	Correct Ratio
DNA	Counting DNA Nucleotides	32486	
RNA	Transcribing DNA into RNA	29038	
REVC	Complementing a Strand of DNA	26301	
FIB	Rabbits and Recurrence Relations	14924	
GC	Computing GC Content	15520	
HAMM	Counting Point Mutations	17560	
IPRB	Mendel's First Law	9916	
PROT	Translating RNA into Protein	13595	
SUBS	Finding a Motif in DNA	13994	
CONS	Consensus and Profile	7841	
FIBD	Mortal Fibonacci Rabbits	6458	
GRPH	Overlap Graphs	6499	
IEV	Calculating Expected Offspring	5889	
LCSM	Finding a Shared Motif	5465	

Version control system

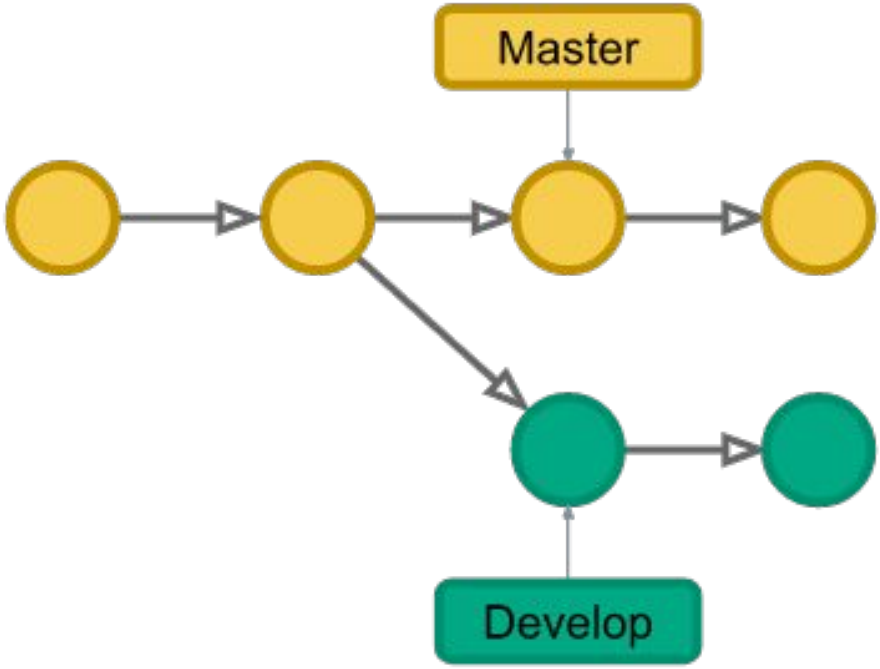
- tracks code changes
- facilitates collaboration on projects
- supports distributed version control platforms like github

git add & git commit -m C4



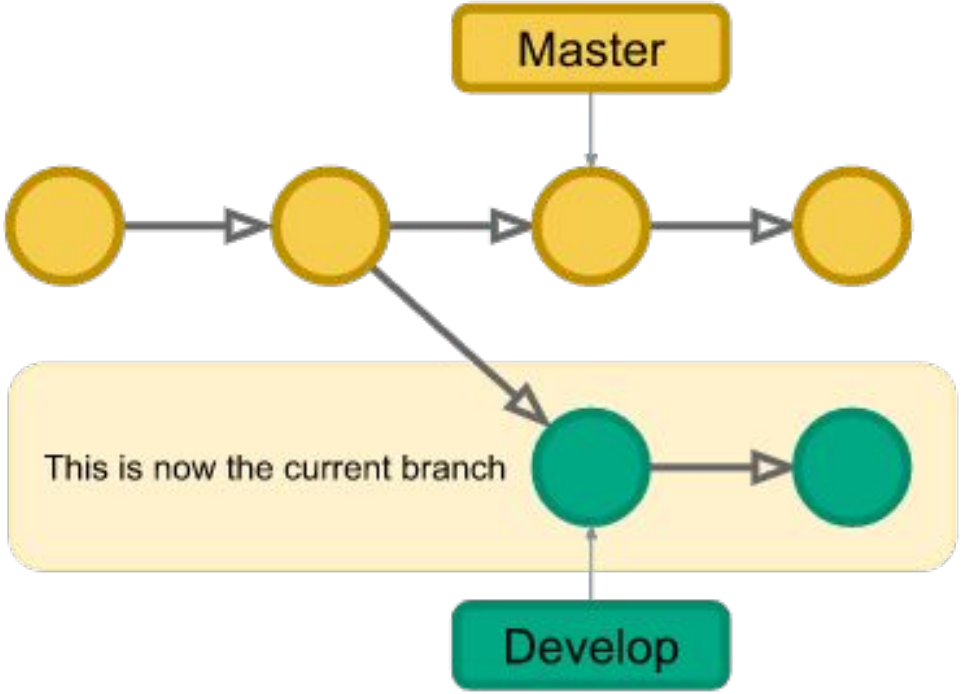
https://dev.to/neshaz/how-to-work-with-git-branches-30c0

git branch develop



https://www.campingcoder.com/2018/02/git-branches-simply-explained/

git checkout develop



https://www.campingcoder.com/2018/02/git-branches-simply-explained/

octo-org / octo-repo Private

Watch 1 | Star 0 | Fork 1

Code | Issues 5 | Pull requests 10 | Projects 3 | Wiki | Insights | Settings

Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base: update-readme ← compare: modifications-to-143v ✓ Able to merge. These branches can be automatically merged.

Before you submit a pull request please review the contributing guidelines for this repository.

Modifications to 143v

Write | Preview

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipbaord.

Styling with Markdown is supported

Create pull request

Reviewers
No reviews—request one

Assignees
No one—assign yourself

Labels
None yet

Projects
None yet

Milestone
No milestone

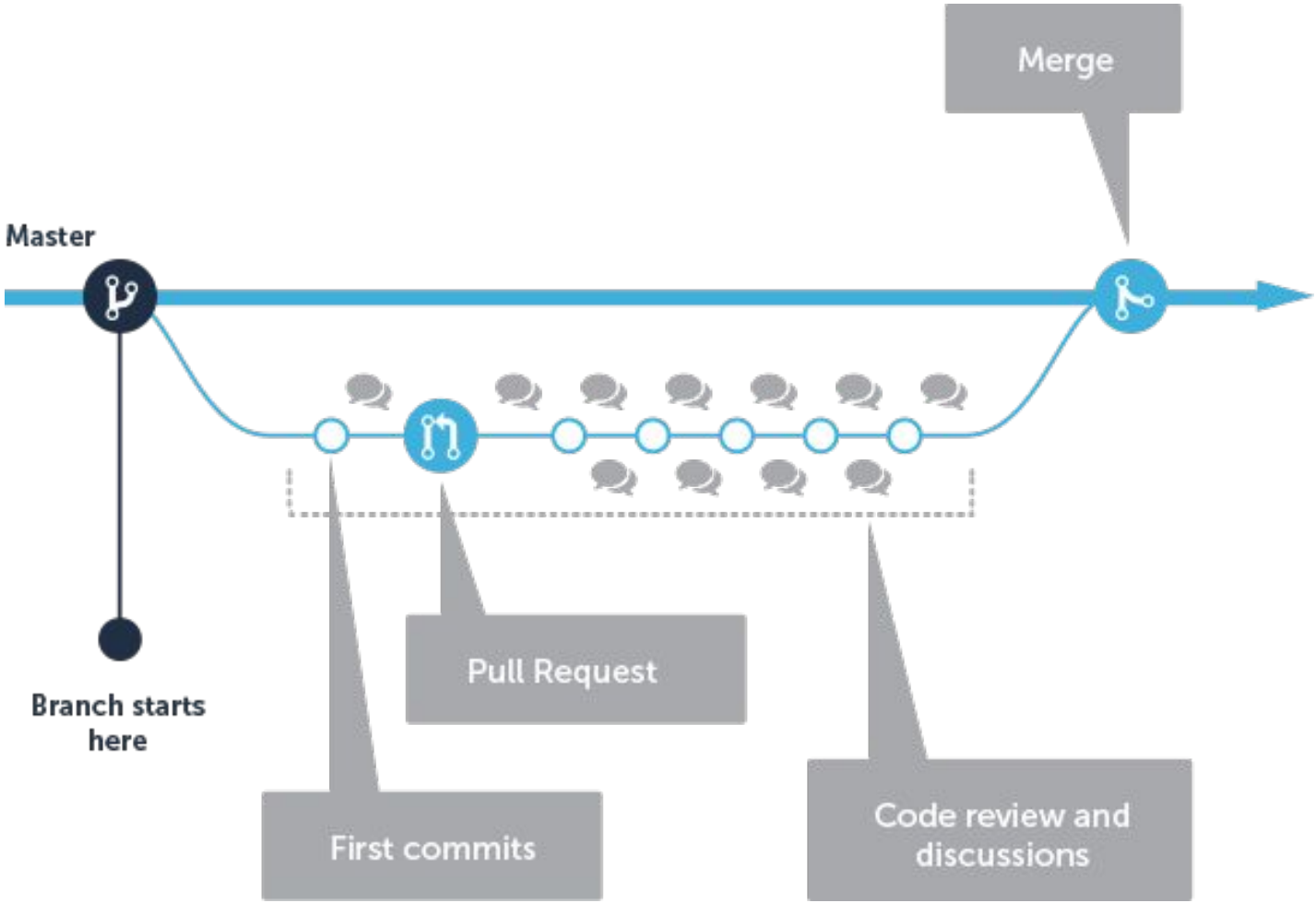
3 commits | 2 files changed | 0 commit comments | 2 contributors

Commits on Aug 06, 2015

bernars	Merge pull request #1 from octo-org/update-readme ···	1509fa5

Commits on Nov 04, 2015

jleaver	Adds Branch 1 document	dbae914
jleaver	Branch 2 test	a97b678



https://www.gitcolony.com/features

jpfeil / **Workshop-2018**

Unwatch 1 | Star 0 | Fork 1

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Insights | Settings

Title

Write | Preview

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

Styling with Markdown is supported

Submit new issue

Assignees

No one—assign yourself

Labels

None yet

Projects

None yet

Milestone

No milestone

Software Container Tool

- packages code and dependencies
- makes it easier to deploy
- kind of like a virtual machine

docker run

```
[~] docker run hello-world
Unable to find image 'hello-world:latest' locally
latest: Pulling from library/hello-world
9bb5a5d4561a: Pull complete
Digest: sha256:f5233545e43561214ca4891fd1157e1c3c563316ed8e237750d59bde73361e77
Status: Downloaded newer image for hello-world:latest

Hello from Docker!
This message shows that your installation appears to be working correctly.

To generate this message, Docker took the following steps:
 1. The Docker client contacted the Docker daemon.
 2. The Docker daemon pulled the "hello-world" image from the Docker Hub.
    (amd64)
 3. The Docker daemon created a new container from that image which runs the
    executable that produces the output you are currently reading.
 4. The Docker daemon streamed that output to the Docker client, which sent it
    to your terminal.

To try something more ambitious, you can run an Ubuntu container with:
 $ docker run -it ubuntu bash

Share images, automate workflows, and more with a free Docker ID:
 https://hub.docker.com/

For more examples and ideas, visit:
 https://docs.docker.com/engine/userguide/
```

docker images

```
[~] docker images
REPOSITORY                    TAG                 IMAGE ID            CREATED             SIZE
jpfeil/mixture                0.3.3               10f4f835f845        4 weeks ago         4.11GB
```

Dockerfile

```
FROM ubuntu:16.04

MAINTAINER Jacob Pfeil, jpfeil@ucsc.edu

# Update and install required software
RUN apt-get update --fix-missing \
    && apt-get install -y python python-matplotlib zlib1g-dev \
                          build-essential make wget libgl1-mesa-glx \
                          libboost-all-dev llvm autotools-dev libicu-dev \
                          g++ parallel \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/*

# Install miniconda
RUN wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh -O ~/miniconda.sh
RUN bash ~/miniconda.sh -b -p $HOME/miniconda

ENV PATH=/root/miniconda/bin:$PATH

RUN conda update -y conda

RUN conda install -y seaborn numpy pandas

RUN pip install pystan

COPY mixture /opt/mixture

ENV TH_MIXTURE_SRC=/opt/mixture

WORKDIR /opt

RUN wget https://github.com/stan-dev/cmdstan/releases/download/v2.17.0/cmdstan-2.17.0.tar.gz -O /opt/cmd
stan-2.17.0.tar.gz \
    && tar xvf /opt/cmdstan-2.17.0.tar.gz \
    && cd /opt/cmdstan-2.17.0 \
    && echo "CC=g++" > make/local \
    && make build -j 4

RUN cd /opt/cmdstan-2.17.0 && parallel make {.} ::: /opt/mixture/models/*.stan

ENV STAN_SRC=/opt/cmdstan-2.17.0

# Data processing occurs at /data
WORKDIR /data

ENTRYPOINT ["python", "/opt/mixture/run.py"]
CMD ["-h"]
```

Dockerfile

```
FROM ubuntu:16.04

MAINTAINER Jacob Pfeil, jpfeil@ucsc.edu
```

Dockerfile

```
# Update and install required software
RUN apt-get update --fix-missing \
    && apt-get install -y python python-matplotlib zlib1g-dev \
                          build-essential make wget libgl1-mesa-glx \
                          libboost-all-dev llvm autotools-dev libicu-dev \
                          g++ parallel \
    && apt-get clean \
    && rm -rf /var/lib/apt/lists/*
```

Dockerfile

```
COPY mixture /opt/mixture

ENV TH_MIXTURE_SRC=/opt/mixture
```

Dockerfile

```
# Data processing occurs at /data
WORKDIR /data

ENTRYPOINT ["python", "/opt/mixture/run.py"]
CMD ["-h"]
```

```
docker build . -t jpfeil/mixture:0.3.3
```

```
docker run -it -v $(pwd):/data/ DockerhubName/rosalind:version
```