



Graph your data

... and be social about it

tinyurl.com/codher1704

Outline for the next hours



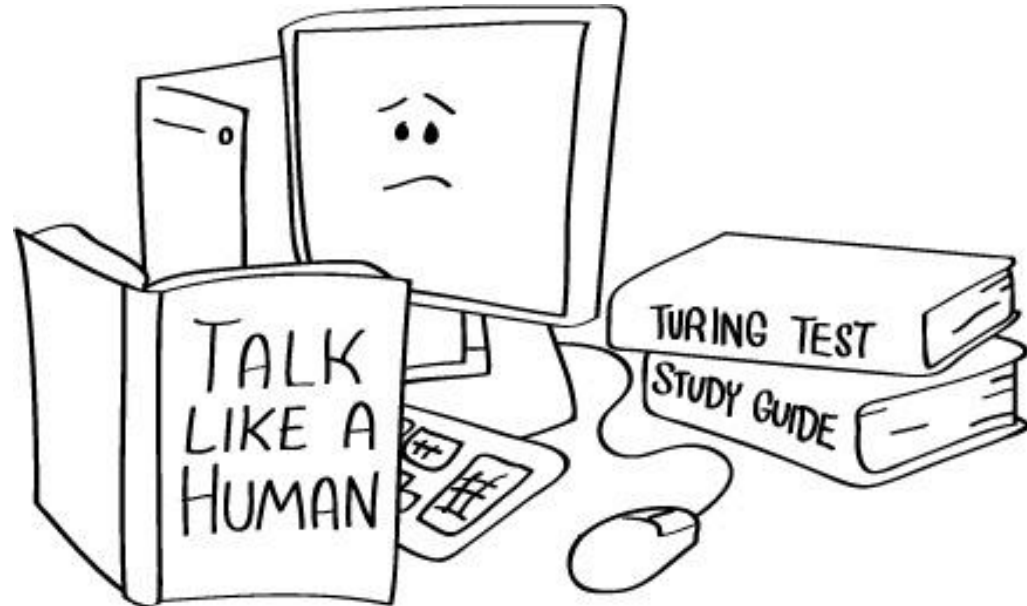
tinyurl.com/codher1704

- 1st Hour: Meet colleagues, intro to Python, make account for online programming
- 2nd Hour: Simple execution of Python, Jupyter notebook, discuss data, collaborative programming
- 3rd Hour: Discuss challenge, implement solution, training datasets from kaggle.com, plot data from dataset

(...and be social about it)



Humans need humans





MARIA HIDALGO
Co-founder



PLAMENA CHERNEVA
Co-founder



CHRISTIANE SANCHEZ
Co-founder



RAUL PINEDA
Program Manager



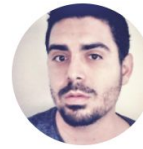
TROELS SCHWARZ-LINNET
Mentor



IVAYLO TAHIRSKI
Mentor



RONI RADEV
Mentor



JORGE RODRIGUEZ
Mentor



NICOLAI GONCALVES
Mentor



LENKA PROCHAZKOVA
Meetups Manager



PAVLE AVRAMOVIC
Mentor



MIKOLAJ MATYASZCZYK
Mentor



JESPER DEVANTIER
Mentor



EMILIS SAPRONAS
Mentor



KASPER MULLESGAARD
Mentor



ARYA SHAHALI
Mentor



CHRIS HJORTH
Mentor



IOANA GROZAV
Mentor



CARSTEN NILSSON
Mentor



ANNE REINHARD
Mentor



HASAN SAMANCI
Mentor



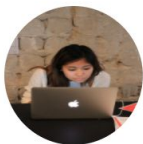
JU GONCALVES
Mentor



DITLEV TOJNER
Mentor



PAUL MORAR
Mentor



LOUISE CHRISTENSEN
Mentor



MARK LEWIS
Mentor



CLARISSA SCHWARZ-LINNET
Community manager

But who are you?

1. Organize tables and form groups of 4.
2. In turn at the table, present yourself and why you are here.
3. Afterwards, 2 people leave for the next table. Here you have to present your colleague for the 2 new people at the table.

Ready?

In-class voting poll

During the next 3 hours, we will use an in-class anonymous voting poll.
I can use this tool to get feedback from all of you at the same time, and it is fun. :)

b.socrative.com

Get Account

Apps

Resources

STUDENT LOGIN

TEACHER LOGIN

- → Student login
- Room name: GHFTL

Student Login

Room Name

GHFTL

JOIN

Questions about our participants

1st question

Type: Short answer

Why are you here?

2nd question

Type: Multiple choice

What is your experience with programming?

- A) None at all
- B) I have tried small scripts before
- C) I know the concepts of variables and functions
- D) I know the concepts of iterators and classes

3rd question

Type: True / False

Is it likely that programming will be part of any of your future work?

4th question

Type: Short answer

Describe how you think programming could be part of your future work, OR describe why not.

Questions/Suggestions during exercise?

- During exercises, there will be mentors available to help you.
- If you have any suggestions or comments during the presentation, please don't hesitate to comment on it. :)

goo.gl/slides/43srww

- Is it going too fast?
- What means “xyz” word?
- Can we again practice “xyz” ?
- Can you elaborate on “xyz” ?



pythonTM

Who, when, why?

- 1991, by Guido van Rossum
- Dutch programmer
- “Hobby programming project”
- Fan of “Monty Python’s flying circus” → The name should be short, unique, and slightly mysterious :)
- Should be easy and intuitive
- Open source
- Understandable in plain English
- Suitable for everyday task
- Allow short development time

Who sponsors it?

- Organisations which benefit from Python
- Many “super master programmers” are hired by high-profile tech companies. They can often devote up to 20% percent of their working time to “pet” projects.
- If you get something for free and you gain something, then pay back to the society with your work, for free.



Price ?

It's free!

And that is the main reasons for its popularity.

Who use it?

- Easy entry into programming for students
- Those who use free tools, or would like to give away free tools
- Visualisation of data, data exploration and data prediction
- If your colleagues or working place use it → You will use it

Programming

Javascript code

Python code

The “language”

- Structure commands
- Different syntax and words between languages
- Each language has its own strength and weakness

```
Vue.component('pizza-item', {
  props: ['type'],
  template: `<li class="list-group-item">
    <input type="button" class="btn btn-default" v-on:click="add" value="+"></input>
    <input type="button" class="btn btn-default" v-on:click="subtract" value="-"></input>
    <span class="badge">{{amount}}</span>
  </li>`,
  data: function () {
    return {
      amount: 0
    }
  },
  methods: {
    add: function () {
      this.amount += 1;
    },
    subtract: function () {
      if (this.amount != 0) {
        this.amount -= 1;
      }
    }
  }
});
```

```
def calc_TPO2(self, R1=None, r1rho_prime=None, dw=None, pA=None, kex=None):
    """Calculation function for the Trott and Palmer (2002) R1rho off-resonance 2-site model.

    @keyword R1: The R1 value.
    @type R1: list of float
    @keyword r1rho_prime: The R1rho value for all states in the absence of exchange.
    @type r1rho_prime: list of float
    @keyword dw: The chemical shift differences in ppm for each spin.
    @type dw: list of float
    @keyword pA: The population of state A.
    @type pA: float
    @keyword kex: The rate of exchange.
    @type kex: float
    @return: The chi-squared value.
    @rtype: float
    """

    # Reshape r1rho_prime to per experiment, spin and frequency.
    self.r1rho_prime_struct1 = multiply.outer( r1rho_prime.reshape(self.NE, self.NM), self.no_nd_ones )

    # Convert dw from ppm to rad/s. Use the out argument, to pass directly to structure.
    multiply( multiply.outer( dw.reshape(1, self.NS), self.nm_no_nd_ones ), self.frq, out=self.dw_struct )

    # Back calculate the R1rho values.
    r1rho_TPO2(r1rho_prime=self.r1rho_prime_struct, omega=self.chemical_shifts, offset=self.offset, pA=pA, dw=self.dw_struct, kex=kex,

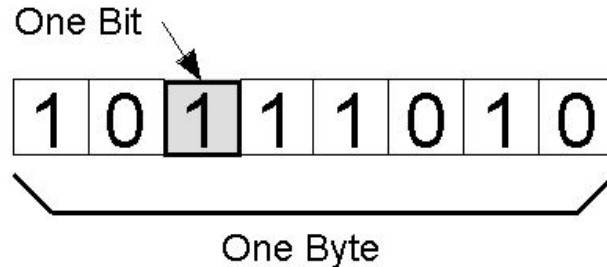
    # Clean the data for all values, which is left over at the end of arrays.
    self.back_calc = self.back_calc+self.disp_struct

    # For all missing data points, set the back-calculated value to the measured values so that it has no effect on the chi-squared val
    if self.has_missing:
        # Replace with values.
        self.back_calc[self.mask_replace_blank_mask] = self.values[self.mask_replace_blank_mask]

    # Return the total chi-squared value.
    return chi2_rnm(self.values, self.back_calc, self.errors)
```

The “code” is read and translated to **machine code** by an “Interpreter”. For javascript code, the interpreter is the “Browser” program. For python code, the interpreter is the python program.

1/0 byte-compiled program.
Unreadable by humans.
All structure and meaning lost.

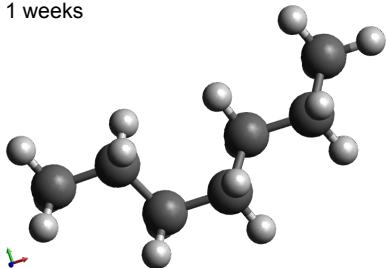


Examples from real life

Molecular statistics

- Calculate all possible positions
- Calculate the energy landscape for all positions
- Determine positions with minimum energy

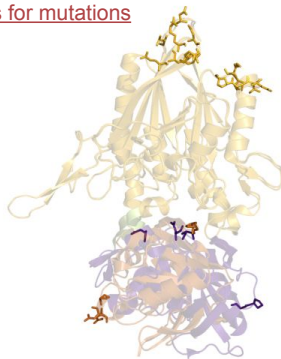
Time: 1 weeks



Protein exploration

- Find all suitable positions for mutations

Time: 2 weeks. Project 10 months.



Analyse nuclear magnetic resonance data

- Make a graphical user interface (GUI) for easy use
- From 100 GB experimental data, perform iterative calculations that will prove that recording time can be done at 25%.

Total time of 2 years.

BIOINFORMATICS APPLICATIONS NOTE Vol. 30 no. 15 2014, pages 2219–2220
doi:10.1093/bioinformatics/btu166

Structural bioinformatics

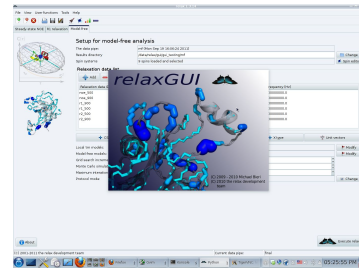
Advance Access publication April 9, 2014

relax: the analysis of biomolecular kinetics and thermodynamics using NMR relaxation dispersion data

Sébastien Morin^{1,2}, Troels E. Linnert³, Mathilde Lescanne⁴, Paul Schanda⁴, Gary S. Thompson⁵, Martin Tollinger⁶, Kaare Teilum³, Stéphane Gagné¹, Dominique Marion⁴, Christian Griesinger⁷, Martin Blackledge⁴ and Edward J. d'Auvergne^{4,7,*}

¹PROTEO, Université Laval, Québec G1V 0A6, Canada; ²International AIDS Society HQ, CH-1202 Geneva, Switzerland; ³Department of Biology, University of Copenhagen, DK-2200 Denmark; ⁴Institut de Biologie Structurale, Grenoble F-38027, France; ⁵Astbury Centre for Structural Molecular Biology, University of Leeds, Leeds LS2 9JT, UK; ⁶Institute of Organic Chemistry & CMBI, University of Innsbruck, A-6020, Austria and ⁷NMR-based Structural Biology, Max Planck Institute for Biophysical Chemistry, D-37077 Göttingen, Germany

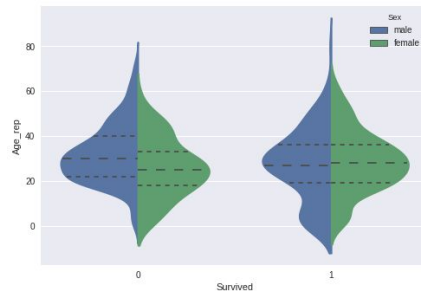
Associate Editor: Anna Tramontano



Predict data

- Find training dataset at kaggle.com

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



Access to a Python programming environment

From home, we suggest <https://www.continuum.io/downloads> **This will not be covered here.**

We need:

- Easy access to Python environment, with no time used on download and installation
- Workable on all participants computers
- All extra packages working for all participants
- Should be free
- Allow for working together on the same project, and programming together.

cloud.sagemath.com

- Created by Mathematics Professor William A. Stein, University of Washington
- To facilitate teaching in programming
- Free access, **but no internet connection and slow computer (shared public kernel instance)**
- \$7 per month, internet access for API programming and faster computers. [Pricing is listed here.](#)

Optimized for Chrome browser. Please install it.

Tea break

Please install chrome browser and
make account before/during break

cloud.sagemath.com

I will check

goo.gl/slides/c2xmnw



Small exercises to run and write a Python program

Where are we?

1st Hour: Meet colleagues, intro to Python, make account for online programming

2nd Hour: Simple execution of Python, Jupyter notebook, discuss data, collaborative programming

For the next hour, we will

- Learn to start Python through a Terminal, and execute small commands “inside” Python
- Write a small text file with commands, and let Python interpret/execute the commands from the file
- Create a Jupyter notebook, and execute Python commands in this
- In groups, discuss your latest data exploration problem and mix with other participants
- Prepare to start a project in sagemath, and invite a colleague to the project for collaborative work

If you have any suggestions or comments during the presentation, please don't hesitate to comment on it. :)

goo.gl/slides/c2xmnw

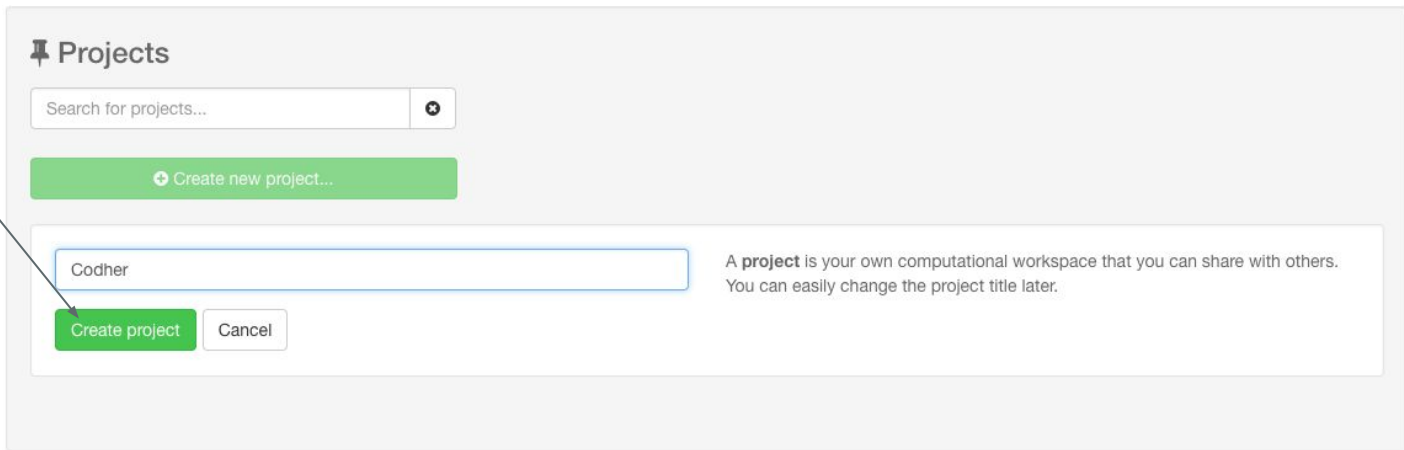
Python through a Terminal

- 1) Create a project

Name: Codher

Click: Create project

A project should contain all files related together in a project. Datafiles, images, etc.



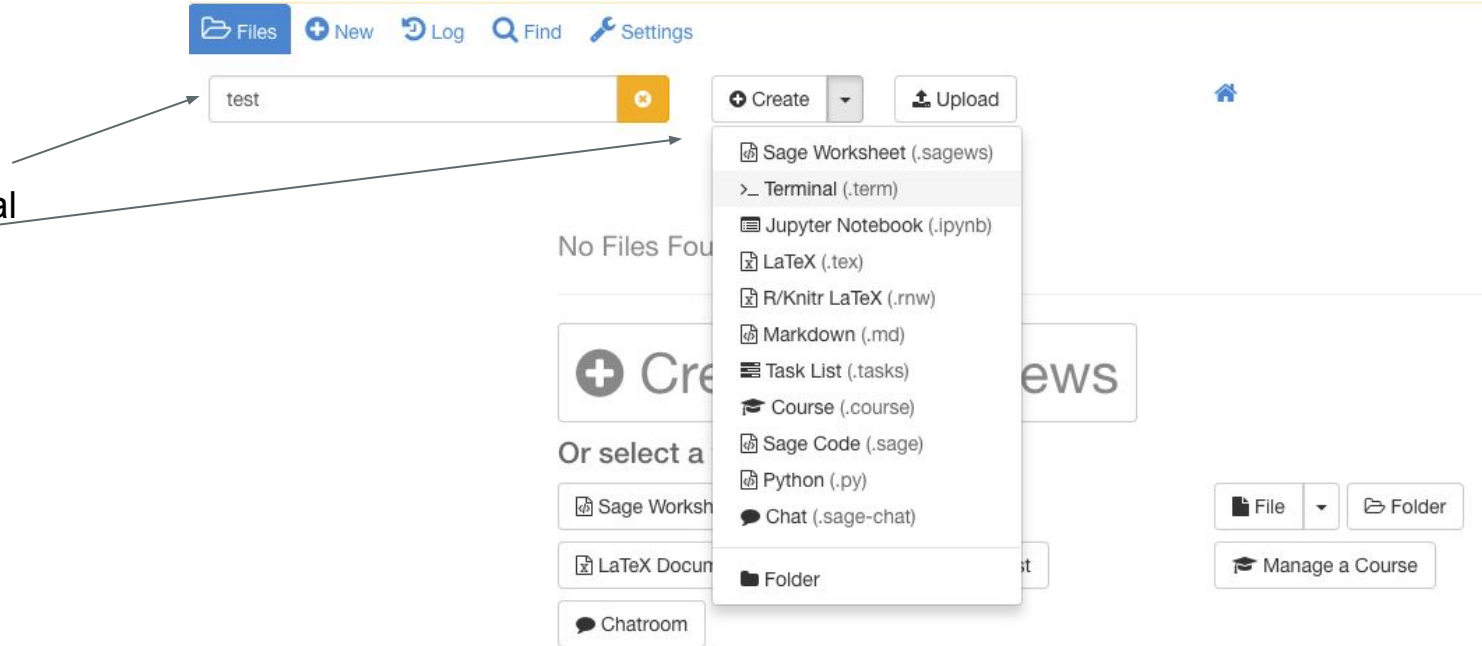
The screenshot shows a web interface titled "Projects" with a pushpin icon. It features a search bar labeled "Search for projects..." with a magnifying glass icon. Below the search bar is a green button labeled "+ Create new project...". A modal dialog is open, showing a text input field with the value "Codher". Below the input field are two buttons: "Create project" (green) and "Cancel" (white). To the right of the modal, there is a text box explaining: "A **project** is your own computational workspace that you can share with others. You can easily change the project title later."

Python through a Terminal

- 2) Create a Terminal

Filename: test

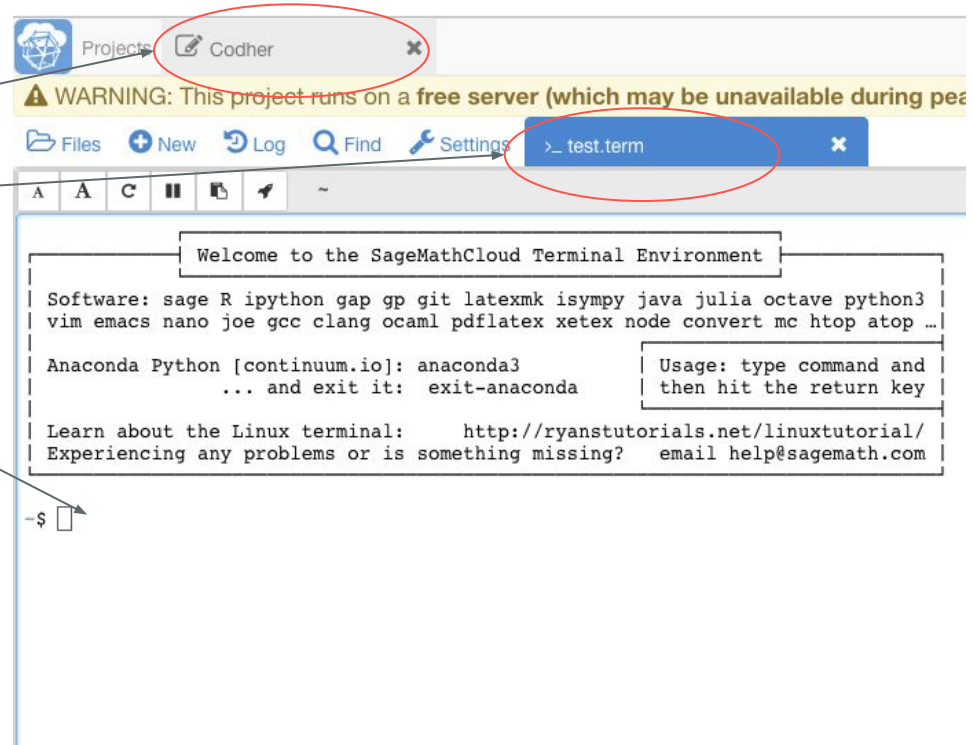
Click: Create Terminal



Python through a Terminal

- 3) Inspect Sagemath setup

Current project
Current file
The terminal



Python through a Terminal

- 4) Try commands in terminal and python

To see the current time
\$ date

To see all files in project
\$ ls

To read the manual for any terminal command,
use the **man** command
\$ man ls
(Click “q” to quit the manual)

Read the manual for python
\$ man python

Clear any text in terminal
\$ clear

Start python
\$ python

Ctrl+d to exit →

```
-$ python
Python 2.7.10 (default, Oct 14 2015, 16:09:02)
[GCC 5.2.1 20151010] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> 2*2
4
>>> print("hello")
hello
>>> a = "Hello World"
>>> print(a)
Hello World
>>> type(a)
<type 'str'>
>>> b = 10
>>> type(b)
<type 'int'>
>>> c = 10.1
>>> type(c)
<type 'float'>
>>> help
Type help() for interactive help, or help(object) for help about object.
>>> help()
```

print() is a built-in function to print text to the screen

a is here defined to a variable, which is a string (text)
The print() command, to print the screen to screen.

type() is a built-in function to show which type the variable is.

b is here defined to a variable, which is an integer (a whole number)

c is here defined to a variable, which is a float (a Real number)

Get help in python

Python through a Terminal

- 5) Inspect the help

```
help> keywords
```

```
Here is a list of the Python keywords. Enter any keyword to get more help.
```

and	elif	if	print
as	else	import	raise
assert	except	in	return
break	exec	is	try
class	finally	lambda	while
continue	for	not	with
def	from	or	yield
del	global	pass	

```
help> print
```

Press "q" to quit

```
help> quit
```

You are now leaving help and returning to the Python interpreter. If you want to ask for help on a particular object directly from the interpreter, you can type "help(object)". Executing "help('string')" has the same effect as typing a particular string at the help> prompt.

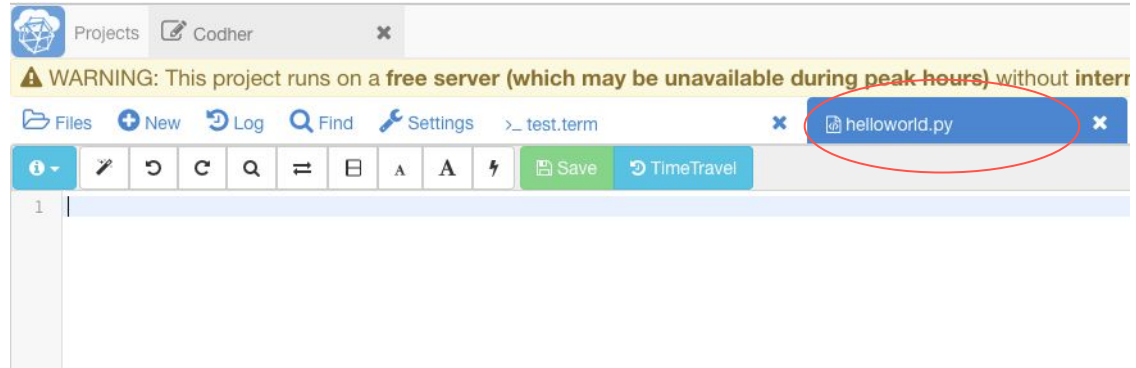
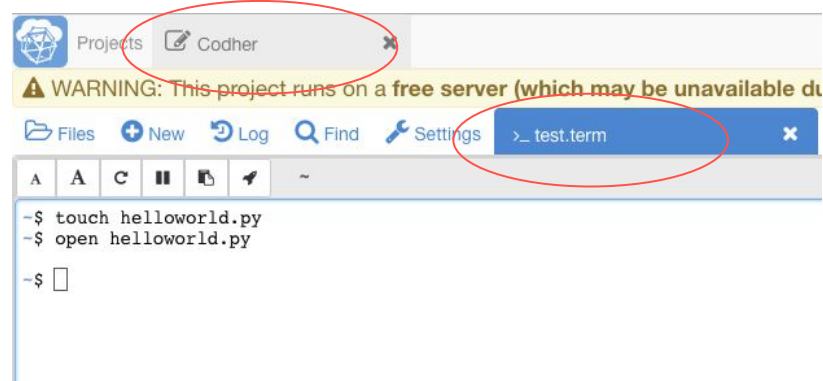
```
>>> █
```

Python through a text file

- 1) Create a text file

Create a new empty text file
\$ touch helloworld.py

Open the file for text edit
\$ open helloworld.py



Python through a text file

- 2) Insert this is in the text field

```
# Let us define a variable with a string
a = "Hello World!"
```

```
# Let us print the string, and expand the use of the print command
print("Let us say all loud: %s"%(a) )
```

```
# Let us make a function
```

```
def calc_area(x, y):
    area = x * y
    return area
```

```
side_a = 10.0
```

```
side_b = 5.0
```

```
temp_area = calc_area(side_a, side_b)
```

```
print("The area was calculated from %f, %f and was in total: %f"% (side_a, side_b, temp_area))
```



- Save with the green button
- Then click on the terminal tab

Python through a text file

- 3) Let Python interpret the text file

Execute the python code file
\$ python helloworld.py

Tip: If you start writing a filename, you
can you just write:
\$ python he (and then click Tab button)

Change in the code,
from: %f
to: %2.3f

Save, and run again

```
~$ python helloworld.py
Let us say all loud: Hello World!
The area was calculated from 10.000000, 5.000000 and was in total: 50.000000
```

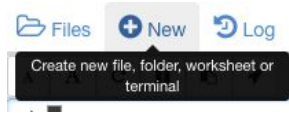
```
print("The area was calculated from %2.3f, %2.3f and was in total: %2.3f"% (side_a, side_b, temp_area))
```

```
~$ python helloworld.py
Let us say all loud: Hello World!
The area was calculated from 10.000000, 5.000000 and was in total: 50.000000
~$ python helloworld.py
Let us say all loud: Hello World!
The area was calculated from 10.000, 5.000 and was in total: 50.000
~$ █
```

Python in a Jupyter notebook

- 1) Create a notebook

- Click new



- Filename: mybook
- Click: Jupyter Notebook

Name your file, folder or paste in a link

mybook

Select the type

Sage Worksheet

Jupyter Notebook

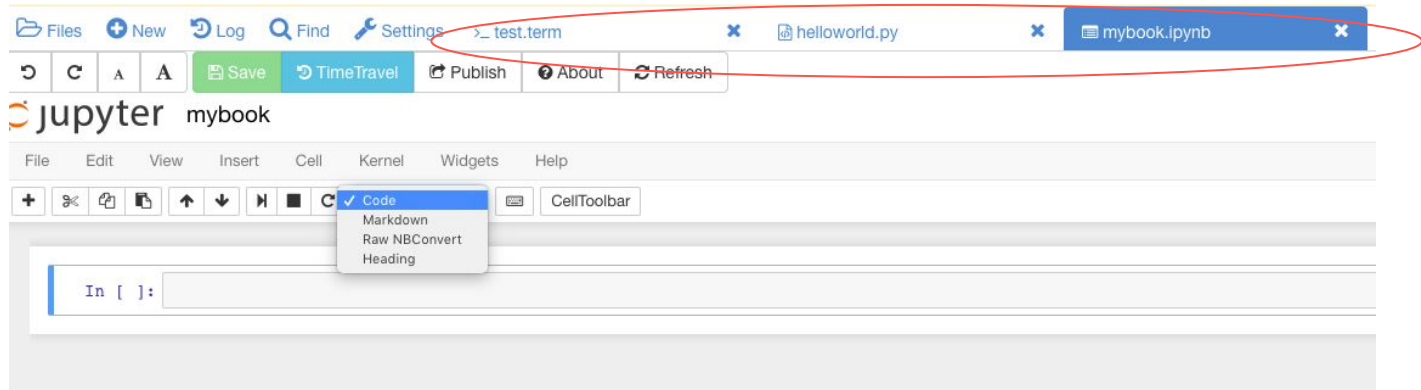
LaTeX Document

Terminal

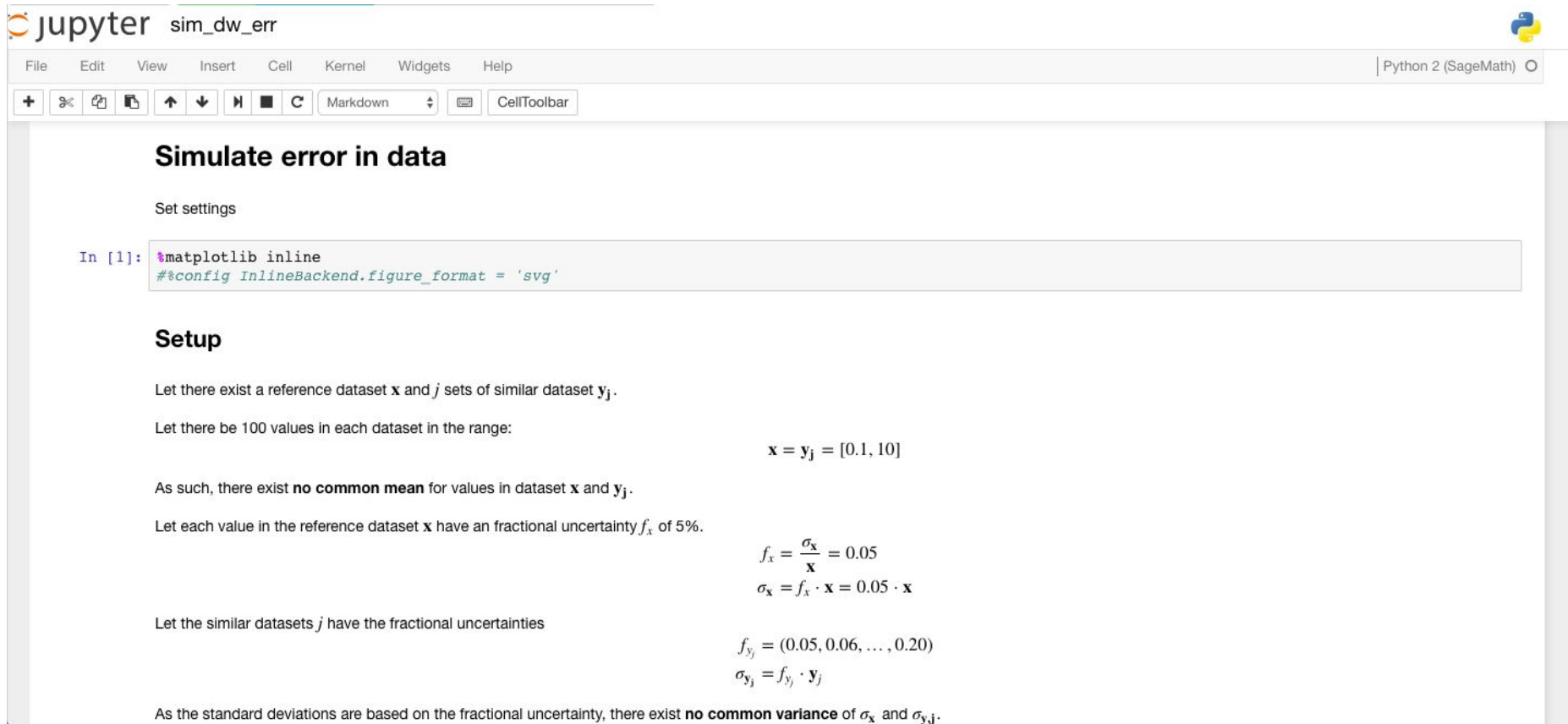
Task Log

Jupyter Notebook

Create an interactive notebook for using Python, Julia, R and more.



Real-life Jupyter notebook example



jupyter sim_dw_err

File Edit View Insert Cell Kernel Widgets Help Python 2 (SageMath)

+ % Save Run Cell Toolbar

Simulate error in data

Set settings

```
In [1]: %matplotlib inline
        # %config InlineBackend.figure_format = 'svg'
```

Setup

Let there exist a reference dataset \mathbf{x} and j sets of similar dataset \mathbf{y}_j .

Let there be 100 values in each dataset in the range:

$$\mathbf{x} = \mathbf{y}_j = [0.1, 10]$$

As such, there exist **no common mean** for values in dataset \mathbf{x} and \mathbf{y}_j .

Let each value in the reference dataset \mathbf{x} have an fractional uncertainty f_x of 5%.

$$f_x = \frac{\sigma_x}{\mathbf{x}} = 0.05$$
$$\sigma_x = f_x \cdot \mathbf{x} = 0.05 \cdot \mathbf{x}$$

Let the similar datasets j have the fractional uncertainties

$$f_{y_j} = (0.05, 0.06, \dots, 0.20)$$
$$\sigma_{y_j} = f_{y_j} \cdot \mathbf{y}_j$$

As the standard deviations are based on the fractional uncertainty, there exist **no common variance** of σ_x and $\sigma_{y,j}$.

Draw datapoints from normal distributions

For each value x_i, y_i in \mathbf{x}, \mathbf{y}_j , draw new values u_i, v_i from the normal distributions $\mathcal{N}(x_i, \sigma_{x,i}), \mathcal{N}(y_i, \sigma_{y,i})$ to the vector of values \mathbf{u}, \mathbf{v}_j .

Examine data

For each iteration of j , let the difference vector \mathbf{d}_j between \mathbf{u} and \mathbf{v}_j be:

$$\mathbf{d}_j = \mathbf{u} - \mathbf{v}_j$$

Let an estimation of the **bias/accuracy** be the average of the differences

$$\mu_j = \frac{\sum \mathbf{d}_j}{n}$$

Let an estimation of the **precision** be the standard deviation of the differences

$$\sigma_{d,j} = \sqrt{\frac{\sum (\mathbf{d}_{i,j} - \mu_j)^2}{n - 1}}$$

Let an another estimation of the **precision** be the root mean square of the differences. Note that we here divide by n , since we have not used one degree on freedom to find the average.

$$\text{RMSD}_{d,j} = \sqrt{\frac{\sum (\mathbf{d}_{i,j})^2}{n}}$$

Run script

In [2]: `!reset -f`

Run script

```
In [2]: %reset -f
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.mlab as mlab

# Determine the x values
n_points = 100
x_vals = np.linspace(0.1, 10, num=n_points, endpoint=True)
#x_vals = np.linspace(0.1, 100, num=n_points, endpoint=True)
# Overall random number
shift_ran = 100

# Determine reference uncertainty and new values uncertainty
frac_unc_pct_x = 5
frac_unc_pct_y_arr = np.asarray(range(frac_unc_pct_x, 21), dtype=float)

# Create u values
frac_unc_x = frac_unc_pct_x/100.
stds_x = frac_unc_x * x_vals
seeds_x = range(shift_ran, n_points+shift_ran)

u = []
for i, seed_x in enumerate(seeds_x):
    x_val = x_vals[i]
    std_x = stds_x[i]

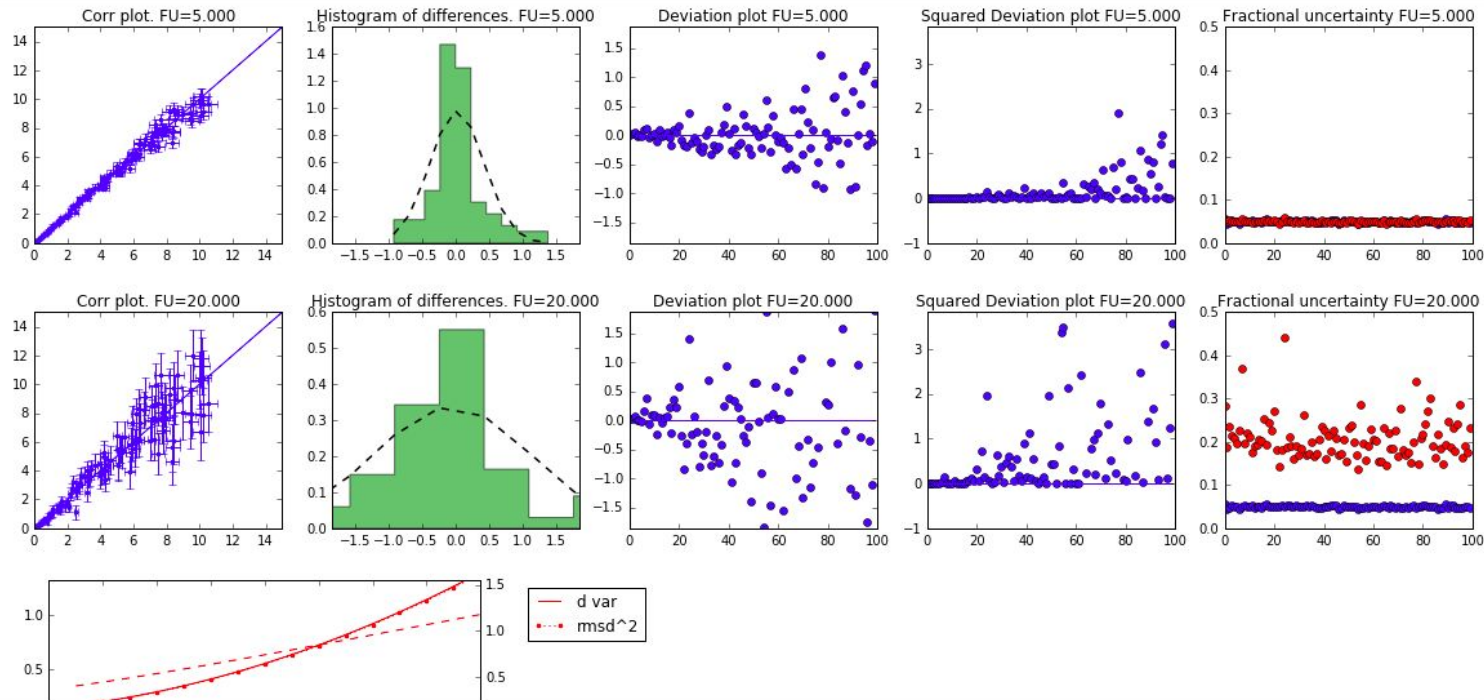
    # Set the seed number, and draw for x
    np.random.seed(seed_x)
    u.append(float(np.random.normal(loc=x_val, scale=std_x, size=1)))
```

```
# Plot histogram of differences
nbins, bins, patches = axarr[1].hist(d_j[j], normed=True, histtype='stepfilled')
plt.setp(patches, 'facecolor', 'g', 'alpha', 0.55)
# add a line showing the expected distribution
prob = mlab.normpdf(bins, d_avg_j[j], d_std_j[j])
axarr[1].plot(bins, prob, 'k--', linewidth=1.5)
axarr[1].set_title("Histogram of differences. FU=%1.3f"%frac_unc_pct_y_arr[j])
if x_bin_lim == None:
    x_bin_lim = [-2*abs(bins.min()), 2*abs(bins.min())]
axarr[1].set_xlim(x_bin_lim)

# Make a deviation plot
axarr[2].plot(range(n_points), d_j[j], linestyle='', marker='o', fillstyle='full')
axarr[2].plot(range(n_points), np.zeros_like(d_j[j]), linestyle='-', marker='', fillstyle='full', color="b")
axarr[2].set_title("Deviation plot FU=%1.3f"%frac_unc_pct_y_arr[j])
if y_lim_dev == None:
    y_lim_dev = [-2*abs(d_j[j].min()), 2*abs(d_j[j].min())]
axarr[2].set_ylim(y_lim_dev)

# Make a squared deviation plot
axarr[3].plot(range(n_points), np.square(d_j[j]), linestyle='', marker='o', fillstyle='full')
axarr[3].plot(range(n_points), np.zeros_like(d_j[j]), linestyle='-', marker='', fillstyle='full', color="b")
axarr[3].set_title("Squared Deviation plot FU=%1.3f"%frac_unc_pct_y_arr[j])
if y_lim_var == None:
    y_lim_var = [-1, 2*abs(np.square(d_j[j]).max())]
axarr[3].set_ylim(y_lim_var)

# Make fractional uncertainty graph
axarr[4].plot(range(n_points), u_f, linestyle='', marker='o', fillstyle='full', color="b")
axarr[4].plot(range(n_points), v_f_j[j], linestyle='', marker='o', fillstyle='full', color="r")
axarr[4].set_title("Fractional uncertainty FU=%1.3f"%frac_unc_pct_y_arr[j])
axarr[4].set_ylim([0.00, 0.50])
```



Python in a Jupyter notebook

- 2) Fill into notebook

Change cell to markdown

My first notebook $\sum \frac{1}{N}$

Then Shift+Enter



[Link to markdown cheatsheet](#)

Let us define a variable with a string

a = "Hello World!"

Let us print the string

print("Let us say all loud: %s"%(a))

Then Shift+Enter

```
In [2]: # Let us define a variable with a string
a = "Hello World!"
# Let us print the string
print("Let us say all loud: %s"%(a) )

Let us say all loud: Hello World!
```

Let us make a function

def calc_area(x, y):

area = x * y

return area

side_a = 10.0

side_b = 5.0

temp_area = calc_area(side_a, side_b)

print("The area was calculated from %2.1f, %2.1f and was in total: %2.1f"%(side_a, side_b, temp_area))

```
In [5]: # Let us make a function
def calc_area(x, y):
    area = x * y
    return area

side_a = 10.0
side_b = 5.0
temp_area = calc_area(side_a, side_b)
print("The area was calculated from %2.1f, %2.1f and was in total: %2.1f"%(side_a, side_b, temp_area))

The area was calculated from 10.0, 5.0 and was in total: 50.0
```

Group discussion about data

- On a piece of paper, make rough small sketch of the last graph you made. For example Excel, MATLAB or similar.
- Think about, what was the necessary steps for preparing the data? (Think in programming steps)

Then

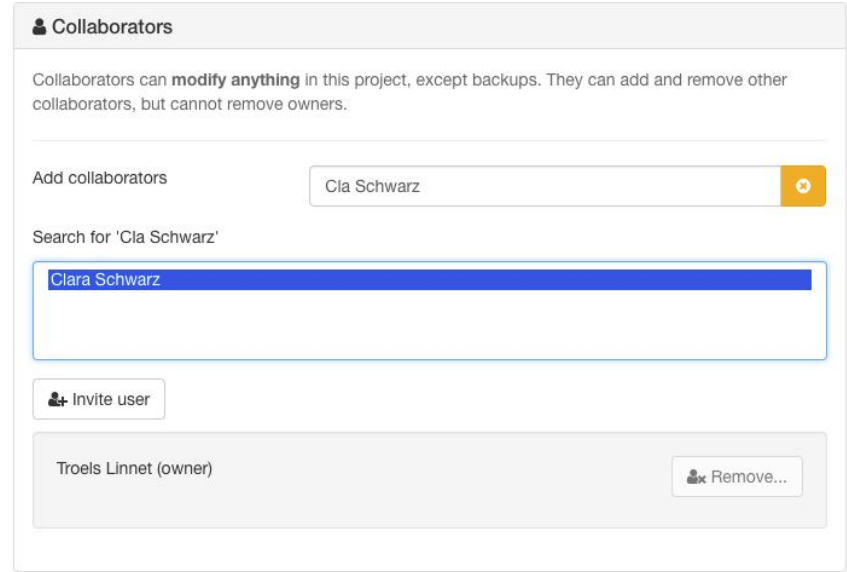
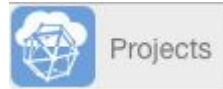
- Everyone get up, and mix in the room
- Present your graph for a new person
- Then mix
- Now, start presenting the graph idea from the previous person, before you present your own again
- Then mix. Now present the last 2 persons graph idea, before your own.



Make a new project and invite colleague

Form groups of 2. One do:

- Click projects
- Project title: Collaborate
- Click: “Create new project”
- Click settings
- To the right in pane “Collaborators” search by the name of your collaborator
- Search, and then click person, then invite
- Click new
- Create a new Jupyter notebook
- Now the other find the project, and open the notebook.
- You can both edit and see each other’s work.



Tea break for 10 min.

Please make sure the collaborative project is working before break

I will check

goo.gl/slides/c2xmnw



How to plot in Python

Where are we?

2nd Hour: Simple execution of Python, Jupyter notebook, discuss data, collaborative programming

3rd Hour: Discuss challenge, implement solution, training datasets from kaggle.com, plot data from dataset

For the next hour, we will

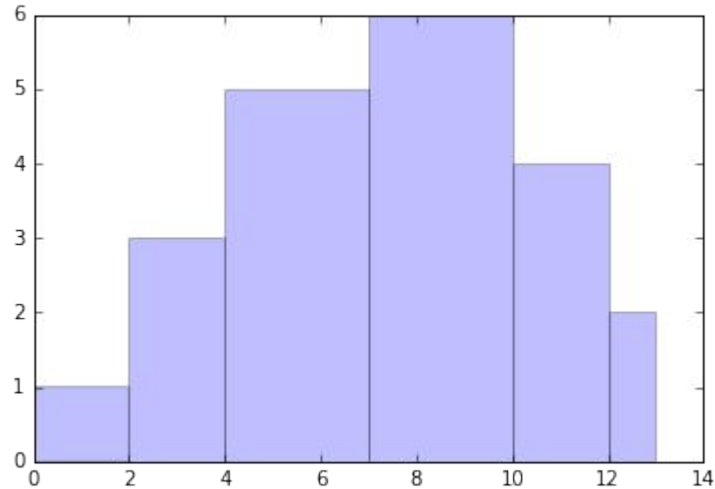
- Discuss how to handle data in programming
- Show some possible solutions to the same problem
- Try it out
- Introduction to kaggle.com
- Plot data from a large dataset

If you have any suggestions or comments during the presentation, please don't hesitate to comment on it. :)

goo.gl/slides/c2xmnw

List of grades in chemistry

grades = [0, 7, 10, 12, 4, 2, 4, 7, 10, 12, 7, 4, 2, 10, 7, 4, 7, 2, 10, 7, 4]



By using words, discuss with your colleague how you would by “paper and pen” draw this graph manually.

The manual way to solve the problem

```
In [24]: grades = [0, 7, 10, 12, 4, 2, 4, 7, 10, 12, 7, 4, 2, 10, 7, 4, 7, 2, 10, 7, 4]
print("Nr of grades are: %i" %(len(grades)) )

uniq_grades = list(set(grades))
print("The uniq grades are: %s" %(uniq_grades) )

count = []
for grade in uniq_grades:
    nr = grades.count(grade)
    count.append(nr)
print("The nr of each uniq grade is: %s" %(count))
```

```
Nr of grades are: 21
The uniq grades are: [0, 2, 4, 7, 10, 12]
The nr of each uniq grade is: [1, 3, 5, 6, 4, 2]
```

```
grades = [0, 7, 10, 12, 4, 2, 4, 7, 10, 12, 7, 4, 2, 10, 7, 4, 7, 2, 10, 7, 4]
print("Nr of grades are: %i" %(len(grades)) )
```

```
uniq_grades = list(set(grades))
print("The uniq grades are: %s" %(uniq_grades) )
```

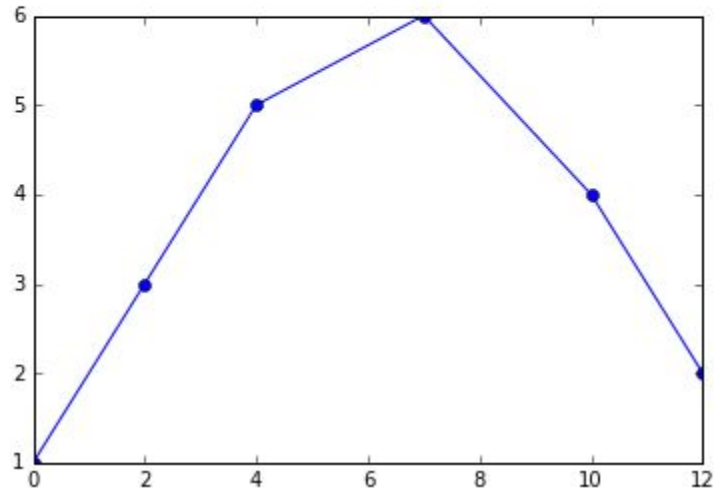
```
count = []
for grade in uniq_grades:
    nr = grades.count(grade)
    count.append(nr)
print("The nr of each uniq grade is: %s" %(count))
```

The manual way to solve the problem

```
In [27]: import matplotlib.pyplot as plt  
%matplotlib inline
```

```
plt.plot(uniq_grades, count, "o-")
```

```
Out[27]: [<matplotlib.lines.Line2D at 0x7fd0d0120950>]
```



Nr of grades are: 21

The uniq grades are: [0, 2, 4, 7, 10, 12]

The nr of each uniq grade is: [1, 3, 5, 6, 4, 2]

```
import matplotlib.pyplot as plt  
%matplotlib inline
```

```
plt.plot(uniq_grades, count, "o-")
```

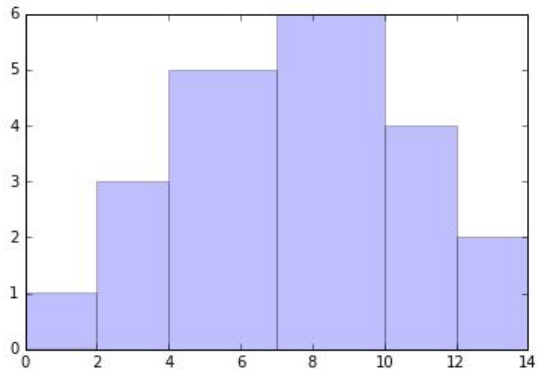
Using the tools available

```
import matplotlib.pyplot as plt
%matplotlib inline

grades = [0, 7, 10, 12, 4, 2, 4, 7, 10, 12, 7, 4, 2, 10, 7, 4, 7, 2, 10, 7, 4]
bins = [0, 2, 4, 7, 10, 12, 14]
n, bins, patches = plt.hist(grades, bins, normed=0, alpha=0.25, label="Grades")
print("The bins are: %s"%(bins))
print("The nr of bins are: %s"%(n))
```

The bins are: [0 2 4 7 10 12 14]

The nr of bins are: [1. 3. 5. 6. 4. 2.]



Nr of grades are: 21

The uniq grades are: [0, 2, 4, 7, 10, 12]

The nr of each grade is: [1, 3, 5, 6, 4, 2]

```
import matplotlib.pyplot as plt
%matplotlib inline

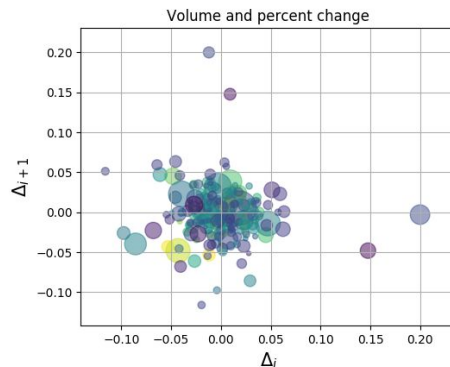
grades = [0, 7, 10, 12, 4, 2, 4, 7, 10, 12, 7, 4, 2, 10, 7, 4, 7, 2, 10, 7, 4]
bins = [0, 2, 4, 7, 10, 12, 14]
n, bins, patches = plt.hist(grades, bins, normed=0, alpha=0.25,
label="Grades")
print("The bins are: %s"%(bins))
print("The nr of bins are: %s"%(n))
```

Try it out!

To get help, try also

- `help(set)`
- `help(list)`
- `help(count)`
- `help(plt.plot)`
- `help(plt.hist)`

http://matplotlib.org/examples/pylab_examples/scatter_demo2.html



```
"""
Demo of scatter plot with varying marker colors and sizes.
"""
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.cbook as cbook

# Load a numpy record array from yahoo csv data with fields date,
# open, close, volume, adj_close from the mpl-data/example directory.
# The record array stores python datetime.date as an object array in
# the date column
datafile = cbook.get_sample_data('goog.npy')
try:
    # Python3 cannot load python2 .npy files with datetime(object) arrays
    # unless the encoding is set to bytes. However this option was
    # not added until numpy 1.10 so this example will only work with
    # python 2 or with numpy 1.10 and later
    price_data = np.load(datafile, encoding='bytes').view(np.recarray)
except TypeError:
    price_data = np.load(datafile).view(np.recarray)
price_data = price_data[-250:] # get the most recent 250 trading days

delta1 = np.diff(price_data.adj_close)/price_data.adj_close[:-1]

# Marker size in units of points^2
volume = (15 * price_data.volume[:-2] / price_data.volume[0])**2
close = 0.003 * price_data.close[:-2] / 0.003 * price_data.open[:-2]
```

Let us fix any questions until now!

b.socrative.com

Get Account

Apps

Resources

STUDENT LOGIN

TEACHER LOGIN

- → Student login
- Room name: GHFTL

Student Login

Room Name

GHFTL

JOIN

When errors arrive

```
a = "I am a string"  
b = 10  
print(a/b)
```

```
-----  
TypeError                                Traceback (most recent call last)  
<ipython-input-42-0c82e7e68443> in <module>()  
      1 a = "I am a string"  
      2 b = 10  
----> 3 print(a/b)
```

TypeError: unsupported operand type(s) for /: 'str' and 'int'

```
print(c)
```

```
-----  
NameError                                Traceback (most recent call last)  
<ipython-input-43-5315f3e3adca> in <module>()  
----> 1 print(c)
```

NameError: name 'c' is not defined



Tea break for 5 min.

I will check

goo.gl/slides/c2xmnw



Kaggle.com

 Search kaggle 

Competitions Datasets Kernels Discussion Jobs [Sign Up](#) [Log In](#)

Your Home for Data Science

Kaggle helps you learn, work, and play

[Create an account](#) or [Host a competition](#)

Competitions ›

Climb the world's most elite machine learning leaderboards

[Want to host a competition?](#)

Datasets ›

Explore and analyze a collection of high quality public datasets

Kernels ›

Run code in the cloud and receive community feedback on your work

Learn to do data science

Survive the Titanic and get started with Python or R

Study benchmark models and code shared on Kernels

Read interviews with top data scientists on No Free Hunch

Teach machine learning with the free InClass platform

Iris

Classify iris plants into three species in this classic dataset

Powered by UCI Machine Learning

[Description](#) [Scripts](#) [Forum](#) [Download data \(16MB\)](#) [New script](#)

Scripts

Python Data Visualizations

Rotating 3D Scatter Plot

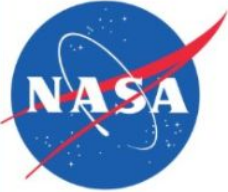
General data analysis

Mapping with sum(row)

Bayes classification


Analysis of Iris Data in R


kaggle.com/nasa/datasets



NASA

Washington, District of Columbia, United States
Joined 6 months ago
<https://www.nasa.gov/>







Organization

Home **Datasets** People

7 datasets


15




NASA Astronauts, 1959-Present **Featured**

Which American astronaut has spent the most time in space?
NASA · updated a month ago

175 downloads
0 comments


6



Trappist-1 Solar System **Featured**

Data from the recently announced 7 exoplanet system
NASA · updated 2 months ago

175 downloads
2 comments


10



Possible Asteroid Impacts with Earth **Featured**

Name, orbit, and year range for impacts predicted by Sentry system
NASA · updated 2 months ago

357 downloads
2 comments

NASA Astronauts, 1959-Present

Which American astronaut has spent the most time in space?

15



by NASA · last updated a month ago

[Overview](#)

[Kernels](#)

[Discussion](#)

[Activity](#)

[Download \(20 KB\)](#)

[New Notebook](#)

[New Script](#)

Context

The term "astronaut" derives from the Greek words meaning "space sailor" and refers to all who have been launched as crew members aboard NASA spacecraft bound for orbit and beyond.

Content

The National Aeronautics and Space Administration (NASA) selected the first group of astronauts in 1959. From 500 candidates with the required jet aircraft flight experience and engineering training in addition to a height below 5 feet 11 inches, seven military men became the nation's first astronauts. The second and third groups chosen included civilians with extensive flying experience. By 1964, requirements had changed, and emphasis was placed on academic qualifications; in 1965, six scientist astronauts were selected from a group of 400 applicants who had a doctorate or equivalent experience in the natural sciences, medicine, or engineering. The group named in 1978 was the first of space shuttle flight crews and fourteen groups have been selected since then with a mix of pilots and mission specialists.

There are currently 50 active astronauts and 35 management astronauts in the program; 196 astronauts have retired or resigned and 49 are deceased (as of April 2013).

Acknowledgements

This dataset was published by the National Aeronautics and Space Administration as the "Astronaut Fact Book" (April 2013 edition). Active astronauts' mission names and flight statistics were updated from the NASA website.

Inspiration

Which American astronaut has spent the most time in space? What university has produced the most astronauts? What subject did the most astronauts major in at college? Have most astronauts served in the military? Which branch? What rank did they achieve?

The cool thing? You can get inspiration from other users. Extremely valuable!!!



by NASA · last updated a month ago

Overview **Kernels** Discussion Activity Download (20 KB) New Notebook **New Script**

6 kernels Sort by Hotness

All Mine All Languages All Output Types

1		Exploring Astronauts! run a month ago by Juan M. Escalante			R
1		Basic Pandas Walkthrough. run a month ago by ArunDas			Py
0		An Exploration of Space Explorers run a month ago by Bryce Freshcorn			Py
0		To the moon run a month ago by Fadi Ramahi			R
0		Notebookf3118d5319 run a month ago by Suezicq			Py
0		Notebook171f61e1e4 run a month ago by Fadi Ramahi			R

No more kernels to show ~_(ツ)_/~

Hi All,

I am a recent graduate from UT San Antonio. My research interest is in Deep Learning and Cloud Computing. This notebook is not complete. I will be adding in more when I get time.

These are just some sample snippets to explore the dataset.

Thanks for sharing this dataset @NASA !

```
In [1]: # This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load in

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in
the input directory

from subprocess import check_output
print(check_output(["ls", "../input"]).decode("utf8"))

# Any results you write to the current directory are saved as output.

astronauts.csv
```

```
In [2]: # Load the dataset.
data = pd.read_csv('../input/astronauts.csv')
```

```
In [3]: # View first 5 rows
data.head()
```

Out[3]:

	Name	Year	Group	Status	Birth Date	Birth Place	Gender	Alma Mater	Undergraduate Major	Graduate Major	Military Rank
0	Joseph M. Acaba	2004.0	19.0	Active	5/17/1967	Inglewood, CA	Male	University of California-Santa Barbara; Univer...	Geology	Geology	NaN
								Montana State			

Let's play! Get the data and plot it...

<https://github.com/Codher/graphing-data>

8 commits 1 branch 0 releases 3 contributors

Branch: master New pull request

Create new file Upload files Find file Clone or download

tinnet Added astronauts dataset from <https://www.kaggle.com/nasa/astronaut-ye...> Latest commit fd9c331 22 minutes ago

README.md Restructured README file 3 days ago

astronauts.csv Added astronauts dataset from <https://www.kaggle.com/nasa/astronaut-ye...> 21 minutes ago

Code Issues 0 Pull requests 0 Projects 0 Wiki Pulse Graphs Settings

Branch: master graphing-data / astronauts.csv Find file Copy path

tinnet Added astronauts dataset from <https://www.kaggle.com/nasa/astronaut-ye...> fd9c331 23 minutes ago

1 contributor

Executable File 359 lines (358 sloc) 79.7 KB

Search this file...

	Name	Year	Group	Status	Birth Date	Birth Place	
1	Joseph M. Acaba	2004	19	Active	5/17/1967	Inglewood, CA	M
2	Loren W. Acton			Retired	3/7/1936	Lewiston, MT	M
3	James C. Adamson	1984	10	Retired	3/3/1946	Warsaw, NY	M
4	Thomas D. Akers	1987	12	Retired	5/20/1951	St. Louis, MO	M
5	Buzz Aldrin	1963	3	Retired	1/20/1930	Montclair, NJ	M
6	Andrew M. Allen	1987	12	Retired	8/4/1955	Philadelphia, PA	Male
7	Joseph P. Allen	1967	6	Retired	6/27/1937	Crawfordsville, IN	Male
8	Scott D. Altman	1995	15	Retired	8/15/1959	Lincoln, IL	Male
9	William A. Anders	1963	3	Retired	10/17/1933	Hong Kong	Male

Raw Blame History

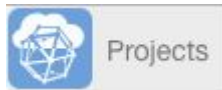
Open Link in New Tab
Open Link in New Window
Open Link in Incognito Window
Save Link As...
Copy Link Address
AdBlock
Inspect
New TextWrangler Document with Selection
Add to iTunes as a Spoken Track
Add to Evernote

- Right-click Raw
- Save Link As...
- Save to desktop or similar

Upload to sagemath

To re-practice

- Make a new project “NASA”
- Invite your colleague to the project
- Upload the datafile to the project by clicking New
- Drag **astronauts.csv** file into the box



→ **Drop files** to upload
(or click)

- Click Files
- Check that **astronauts.csv** is there
- Make a new Jupyter notebook called “plotdata”



plotdata



- 📄 Sage Worksheet (.sagews)
- >_ Terminal (.term)
- 📓 Jupyter Notebook (.ipynb)

Get basic info of data

```
# linear algebra
import numpy as np
# data processing, CSV file I/O (e.g. pd.read_csv)
import pandas as pd
```

```
# Load the dataset.
data = pd.read_csv('astronauts.csv')
```

```
# Let us see which type the data variable is
print("The type of data is: %s"%(type(data)) )
print(data.info())
```

```
In [1]: # linear algebra
import numpy as np
# data processing, CSV file I/O (e.g. pd.read_csv)
import pandas as pd

# Load the dataset.
data = pd.read_csv('astronauts.csv')
```

```
In [5]: # Let us see which type the data variable is
print("The type of data is: %s"%(type(data)) )
print(data.info())
```

```
The type of data is: <class 'pandas.core.frame.DataFrame'>
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 357 entries, 0 to 356
Data columns (total 19 columns):
Name                357 non-null object
Year                330 non-null float64
Group              330 non-null float64
Status             357 non-null object
Birth Date         357 non-null object
Birth Place        357 non-null object
Gender             357 non-null object
Alma Mater         356 non-null object
Undergraduate Major 335 non-null object
Graduate Major     298 non-null object
Military Rank      207 non-null object
Military Branch    211 non-null object
Space Flights      357 non-null int64
Space Flight (hr)  357 non-null int64
Space Walks        357 non-null int64
Space Walks (hr)   357 non-null float64
Missions           334 non-null object
Death Date         52 non-null object
Death Mission      16 non-null object
dtypes: float64(3), int64(3), object(13)
memory usage: 53.1+ KB
None
```

Get basic info of data

```
# Show the first entries in data
data.head()
```

	Name	Year	Group	Status	Birth Date	Birth Place	Gender	Alma Mater	Undergraduate Major	Graduate Major	Military Rank	Military Branch	Space Flights	Space Flight (hr)	Space Walks	Space Walks (hr)	Missions	Death Date	Death Mission
0	Joseph M. Acaba	2004.0	19.0	Active	5/17/1967	Inglewood, CA	Male	University of California-Santa Barbara; Univer...	Geology	Geology	NaN	NaN	2	3307	2	13.0	STS-119 (Discovery), ISS-31/32 (Soyuz)	NaN	NaN
1	Loren W. Acton	NaN	NaN	Retired	3/7/1936	Lewiston, MT	Male	Montana State University; University of Colorado	Engineering Physics	Solar Physics	NaN	NaN	1	190	0	0.0	STS 51-F (Challenger)	NaN	NaN
2	James C. Adamson	1984.0	10.0	Retired	3/3/1946	Warsaw, NY	Male	US Military Academy; Princeton University	Engineering	Aerospace Engineering	Colonel	US Army (Retired)	2	334	0	0.0	STS-28 (Columbia), STS-43 (Atlantis)	NaN	NaN
3	Thomas D. Akers	1987.0	12.0	Retired	5/20/1951	St. Louis, MO	Male	University of Missouri-Rolla	Applied Mathematics	Applied Mathematics	Colonel	US Air Force (Retired)	4	814	4	29.0	STS-41 (Discovery), STS-49 (Endeavor), STS-61 ...	NaN	NaN
4	Buzz Aldrin	1963.0	3.0	Retired	1/20/1930	Montclair, NJ	Male	US Military Academy; MIT	Mechanical Engineering	Astronautics	Colonel	US Air Force (Retired)	2	289	2	8.0	Gemini 12, Apollo 11	NaN	NaN

```
#show the last entries
data.tail()
```

	Name	Year	Group	Status	Birth Date	Birth Place	Gender	Alma Mater	Undergraduate Major	Graduate Major	Military Rank	Military Branch	Space Flights	Space Flight (hr)	Space Walks	Space Walks (hr)	Missions	Death Date	Death Mission
352	David A. Wolf	1990.0	13.0	Retired	8/23/1956	Indianapolis, IN	Male	Purdue University; Indiana University	Electrical Engineering	Medicine	NaN	NaN	3	4044	7	41.0	STS-58 (Columbia). STS-86/89 (Atlantis/Endeavo...	NaN	NaN
353	Neil W. Woodward III	1998.0	17.0	Retired	7/26/1962	Chicago, IL	Male	MIT; University of Texas-Austin; George Washin...	Physics	Physics; Business Management	Commander	US Navy	0	0	0	0.0	NaN	NaN	NaN
354	Alfred M. Worden	1966.0	5.0	Retired	2/7/1932	Jackson, MI	Male	US Military Academy; University of Michigan	Military Science	Aeronautical & Astronautical Engineering	Colonel	US Air Force (Retired)	1	295	1	0.5	Apollo 15	NaN	NaN
355	John W. Young	1962.0	2.0	Retired	9/24/1930	San Francisco, CA	Male	Georgia Institute of Technology	Aeronautical Engineering	NaN	Captain	US Navy (Retired)	6	835	3	20.0	Gemini 3, Gemini 10, Apollo 10, Apollo 16, STS...	NaN	NaN
356	George D. Zamka	1998.0	17.0	Retired	6/29/1962	Jersey City, NJ	Male	US Naval Academy; Florida Institute of Technology	Mathematics	Engineering Management	Colonel	US Marine Corps (Retired)	2	692	0	0.0	STS-120 (Discovery), STS-130 (Endeavor)	NaN	NaN

How many of each gender?

```
print(sum(data['Gender'] == "Male"))  
print(sum(data['Gender'] == "Female"))
```

307

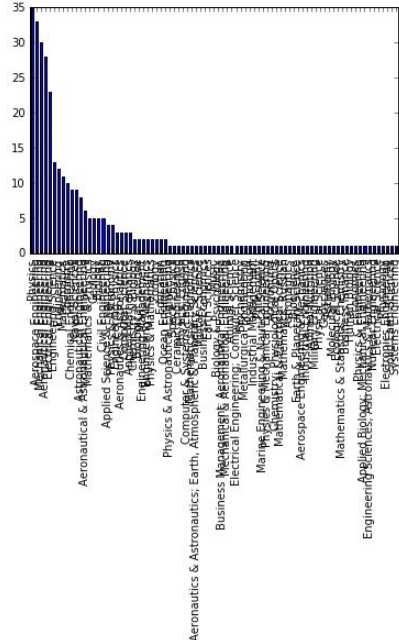
50

```
print(sum(data['Gender'] == "Male"))  
print(sum(data['Gender'] == "Female"))
```

pandas has built-in plotting facility

```
var = data['Undergraduate Major'].value_counts()
print("Type of value_counts() is: %s"%(type(var)))
var.plot(kind='bar')
#print(var)
```

```
Type of value_counts() is: <class 'pandas.core.series.Series'>
<matplotlib.axes._subplots.AxesSubplot at 0x7f1cc37dbfd0>
```

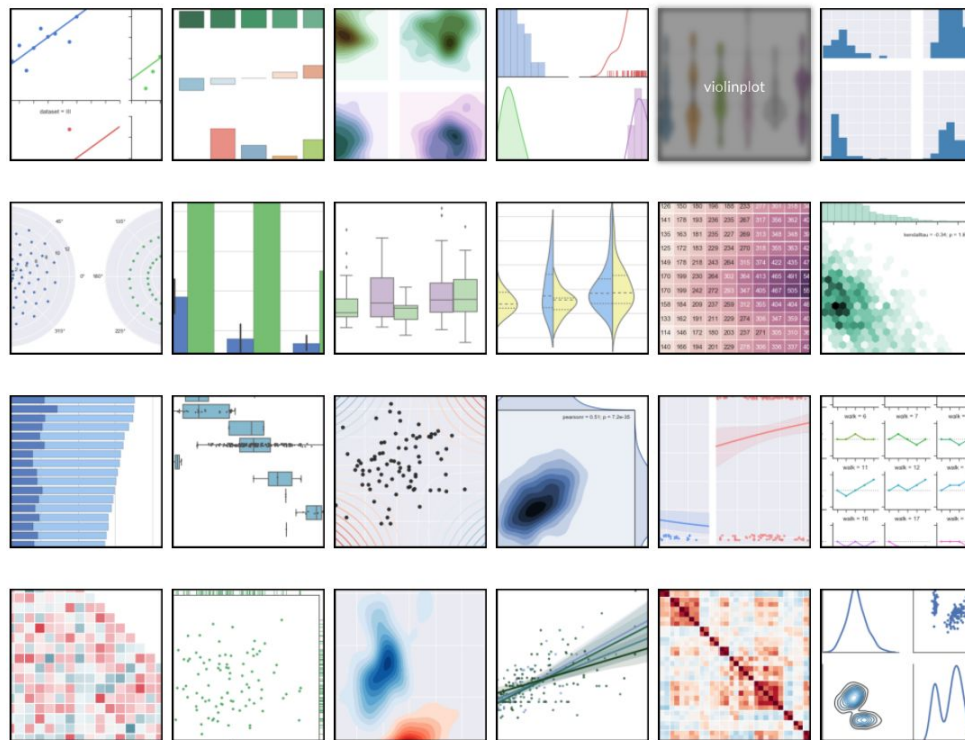


Physics	35
Aerospace Engineering	33
Mechanical Engineering	30
Aeronautical Engineering	28
Electrical Engineering	23
Engineering Science	13
Engineering	12
Mathematics	11
Chemistry	10
Chemical Engineering	9
Naval Sciences	9
Astronautical Engineering	8
Aeronautical & Astronautical Engineering	6
Mathematics & Physics	5
Geology	5
Biology	5
Civil Engineering	5

```
var = data['Undergraduate Major'].value_counts()
print("Type of value_counts() is: %s"%(type(var)))
var.plot(kind='bar')
print(var)
```

<http://seaborn.pydata.org/examples>

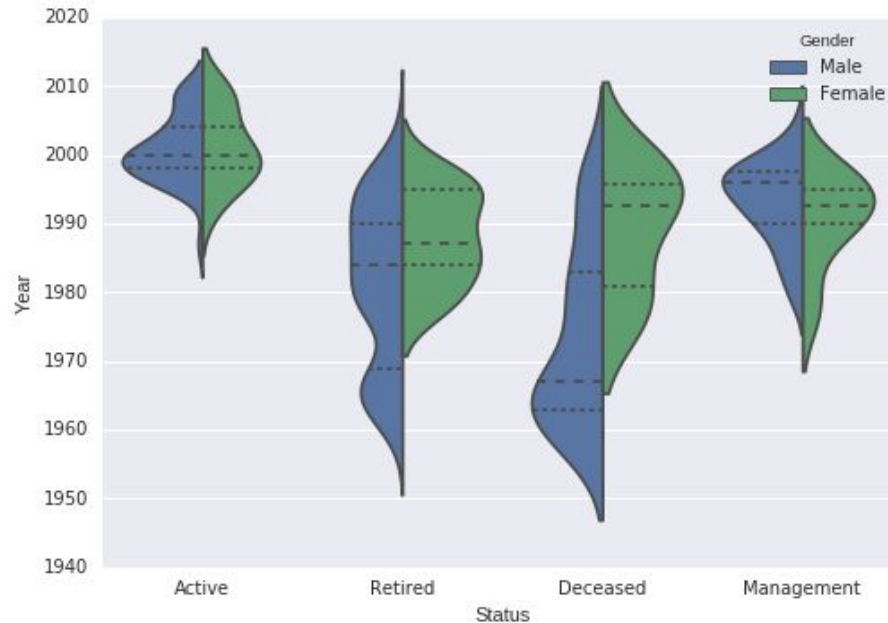
Example gallery



The violinplot

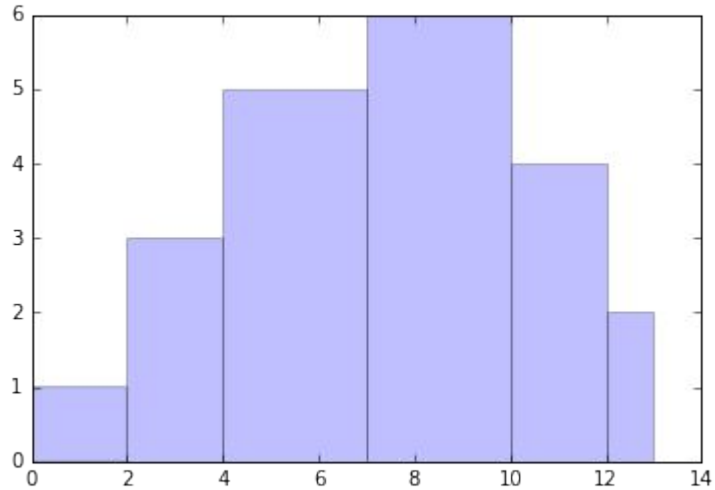
```
import seaborn as sns
sns.violinplot(x="Status", y="Year", hue="Gender", data=data, split=True, inner="quartile")
```

<matplotlib.axes._subplots.AxesSubplot at 0x7f1cc2bc8310>



```
import seaborn as sns
sns.violinplot(x="Status", y="Year", hue="Gender", data=data, split=True, inner="quartile")
```

Try yourself



Before plotting, try by using words, how you would by “paper and pen” draw a graph you are interested in.

Be social - share your experience

#codher #programming #iamcool #kaggle #yetanotherdayinscience





MARIA HIDALGO
Co-founder



PLAMENA CHERNEVA
Co-founder



CHRISTIANE SANCHEZ
Co-founder



RAUL PINEDA
Program Manager



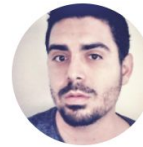
TROELS SCHWARZ-LINNET
Mentor



IVAYLO TAHIRSKI
Mentor



RONI RADEV
Mentor



JORGE RODRIGUEZ
Mentor



NICOLAI GONCALVES
Mentor



LENKA PROCHAZKOVA
Meetups Manager



PAVLE AVRAMOVIC
Mentor



MIKOLAJ MATYASZCZYK
Mentor



JESPER DEVANTIER
Mentor



EMILIS SAPRONAS
Mentor



KASPER MULLESGAARD
Mentor



ARYA SHAHALI
Mentor



CHRIS HJORTH
Mentor



IOANA GROZAV
Mentor



CARSTEN NILSSON
Mentor



ANNE REINHARD
Mentor



HASAN SAMANCI
Mentor



JU GONCALVES
Mentor



DITLEV TOJNER
Mentor



PAUL MORAR
Mentor



LOUISE CHRISTENSEN
Mentor




MARK LEWIS
Mentor



CLARISSA SCHWARZ-LINNET
Community manager






Happy graphing


facebook.com/pg/codher/events/



Codher

Q


Troels
Startside
Find venner







Codher

@codher

Startside

Om

Begivenheder

Billeder

Videoer

Anmeldelser

Synes godt om

Opslag

Opret en side

Synes godt om

Følger ▼

Del

...

Kontakt os

Besked

Kommende begivenheder

Del begivenheder

APR. 19

Diversity in Space Exploration/Intro to program...

I dag 16:00 · Lenka er interesseret

IT University of Copenhagen
Copenhagen

Få billetter

✓ Deltager ▼

APR. 20

Seminar: Declare your UI, don't code it! with Tr...

I morgen 18:00 · Lenka er interesseret

Tradeshift Copenhagen Office
Copenhagen

Få billetter

★ Interesseret

APR. 25

Codher Pop-up Programming vol. 10

ti. 18:00 · 5 gæster

Copenhagen School of Entrepr...
Frederiksberg

Få billetter

★ Interesseret

Tidligere begivenheder

MAR. 21

Codher Pop-up Programming vol. 9

ti. 18:00 · Lenka Prochazkova deltog

Copenhagen School of Entrepr...
Frederiksberg

MAR. 18

Programming Your First Pizza

lo. 9:00 · Clarissa Schwarz-Linnert deltog

Pentia A/S
Copenhagen

FEB. 28

Codher Pop-up Programming vol.8

ti. 18:00 · Lenka Prochazkova deltog

Copenhagen School of Entrepr...
Frederiksberg

FEB. 4

Getting Started with Front End Development

lo. 9:00 · Lenka Prochazkova deltog

Pentia A/S
Copenhagen

JAN. 24

Codher Pop-up Programming vol.7

lo. 18:00 · Lenka Prochazkova deltog

Copenhagen School of Entrepr...

Evaluation time :)

b.socrative.com

Get Account

Apps

Resources

STUDENT LOGIN

TEACHER LOGIN

- → Student login
- Room name: **GHFTL**

Student Login

Room Name

GHFTL

JOIN