Data Science and Interactive Visualization

Analysis and Visualization of Big Data

Franziska Peter and Josep Perelló

Contents - Creating and editing graphics

Session II

Topic 1: Scientific Plots in Practice. Static plots for scientific publication. Python plotting libraries.

Sessions VI

Data Science and Interactive Visualizations. Introduction Pandas. Finishing git. Selecting Databases. First plots. What is a dashboard? Streamlit.

Sessions XI, XII

More interactive plots for web apps. Displaying Geographical Data. Dashboards.

Evaluation

Gradual and incremental set of tasks (in class and through Campus Virtual)

Task 1: Data Management Plan Forensics, in group (Tues 9, JPerelló): 10%

Task 2: Sharing code in Github, individual (Wed 10, FPeter): 10%

Task 3: Write an abstract (Mon 15, JPerelló): 10%

Task 4: Create a dashboard (Thu 18, FPeter): 30%

Task 5: Oral presentation, in group (Fri 19, JPerelló + FPeter): 40%

To set a group between 2 and 4. You will work together during the course.

Outline

1. *Pandas Intro: Theory and Practice*
2. *The rest of git*
3. *Sift your favourite data bases + Form Teams*
4. *Plotting some basic plots from the database*
5. *Dashboards and Streamlit*

Some references

1. https://pandas.pydata.org/

2. Book on Git: https://git-scm.com/book/en/v2

3. http://www.transparenciacatalunya.cat, https://datos.gob.es

4. Few, Stephen. Information dashboard design: displaying data for at-a-glance monitoring. Burlingame, CA : Analytics Press, 2013.

5. Cairo, Alberto. The truthful art : data, charts, and maps for communication. Berkeley : New Riders, 2016.

6. Yau, Nathan. Visualize this: the flowing data guide to design, visualization, and statistics. Indianapolis, Wiley, 2011.

Pandas

= open source data analysis and manipulation tool

Introductory jupyter notebook:
Master_Visualizations_2021/
Topic_3_DataScience_and_Interactive_Visualizations/
Pandas in a nutshell.ipynb

First: Theory
Then: Solve 4 Tasks

Back to git

1. *Small recap from yesterday*
2. *More handy stuff on github*
3. *Feedback on your submissions*

Back to git

Some comments to your shared work:

1. To make your code executable on any machine, use relative paths instead of absolute paths

```
import os
here = os.path.dirname(__file__)
filename = os.path.join(here, "some_file.csv")
```

Also:

* **store all files necessary for execution in the git repository**, so the code can really be run from other machines

* **avoid using spaces or special characters in filenames**. They make it harder to use them in programs on different operating systems

* instead, **use** _ (hello_world.py)

* python programs should only contain small letters by the norm https://www.python.org/dev/peps/pep-0008/

* **NEVER** use python library module names as filenames, e.g. plotly.py

Back to git

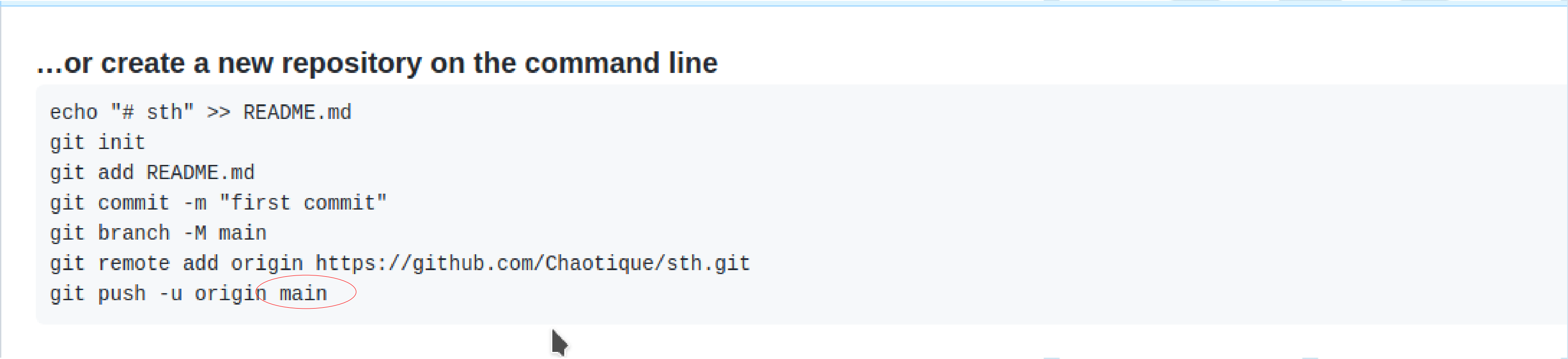
Some comments to your shared work:
2. Use pprint to output DataFrames (especially outside of jupyter)

```
import pandas as pd
from pprint import pprint

df = pd.DataFrame()
df["Country"]  = ["UK",     "France", "Spain"]
df["Capital"] = ["London", "Paris", "Madrid"]
df["Inhabitants Country in Mio"] = [67.22, 67.39,
47.35]
pprint(df)
```

Back to git

Some comments to your shared work:
3. always use master as your main branch. Git explanation is misleading (when creating a new remote repo)



…or create a new repository on the command line

```
echo "# sth" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/Chaotique/sth.git
git push -u origin main
```

While some solve git problems...

… the others can try to get an animation code running:

Master_Visualizations_2021/
Topic_3_DataScience_and_Interactive_Visualizations/
matplotlib_animation/
plot_kuramoto_dots.py

** try to get code running*
** play around with model parameters*
** try to include model parameters in filename (see Monday notebook)*

Dashboard

Collection of interactive figures and diagrams on some specific topic

Dashboards are often required to:

- update themselves automatically
- be self-explanatory (depending on user community)
- be accessible online

Dashboard

Collection of interactive figures and diagrams on some specific topic

My favourite Dashboard is unfortunately in German..
https://www.zeit.de/wissen/corona-karte-deutschland-aktuelle-zahlen-landkreise

They also explain how the dashboard evolved and which programs they use. Unfortunately in German as well…
https://blog.zeit.de/fragen/2021/02/26/was-hinter-dem-corona-dashboard-auf-zeit-online-steckt/

Your favourite Databases

? *What is the topic of the data base?*

? *Which columns are contained?*

?* *Which plots could we think of?*

We need two teams, SO..

* meet in your small groups and discuss which database(s) you prefer

* we'll see if that magically matches well, or if some would rather hop to an other group/ database

* we should end up with two (or three) more or less equal groups

We need two teams, and in each team we need:

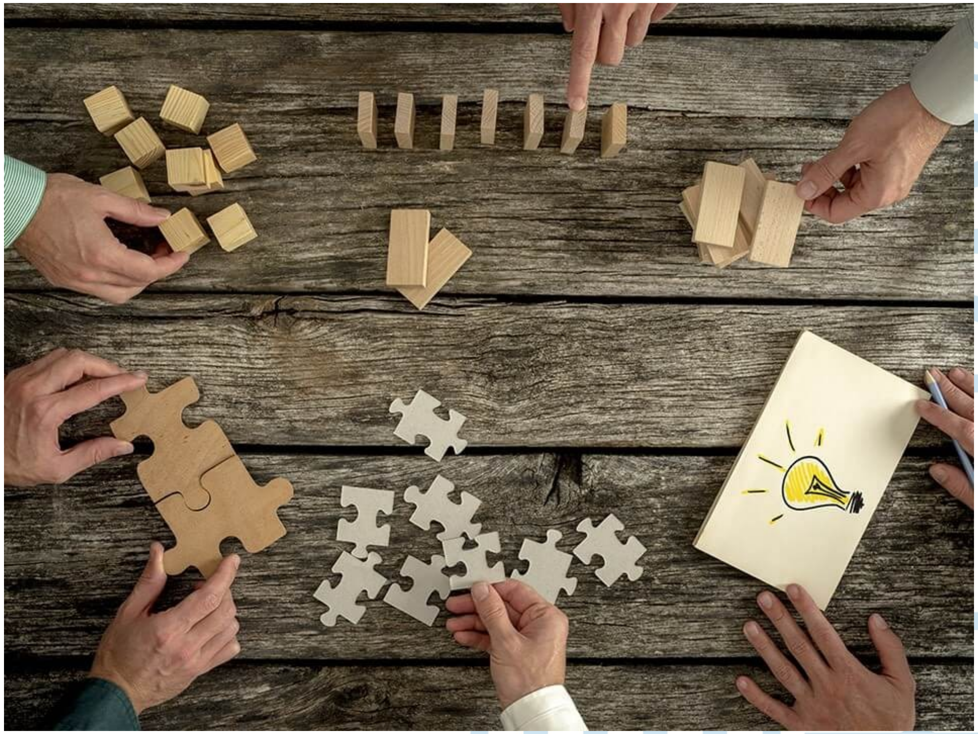
* an editorial board (2):
- coordinates what will end up on the page

* task coordinators (2):
- coordinates who does what when

* a representative (1):
- communicates issues and progress

* a documenter (1):
- writes down work progress

* git responsible (2):
- make sure all code is bundled in one repo

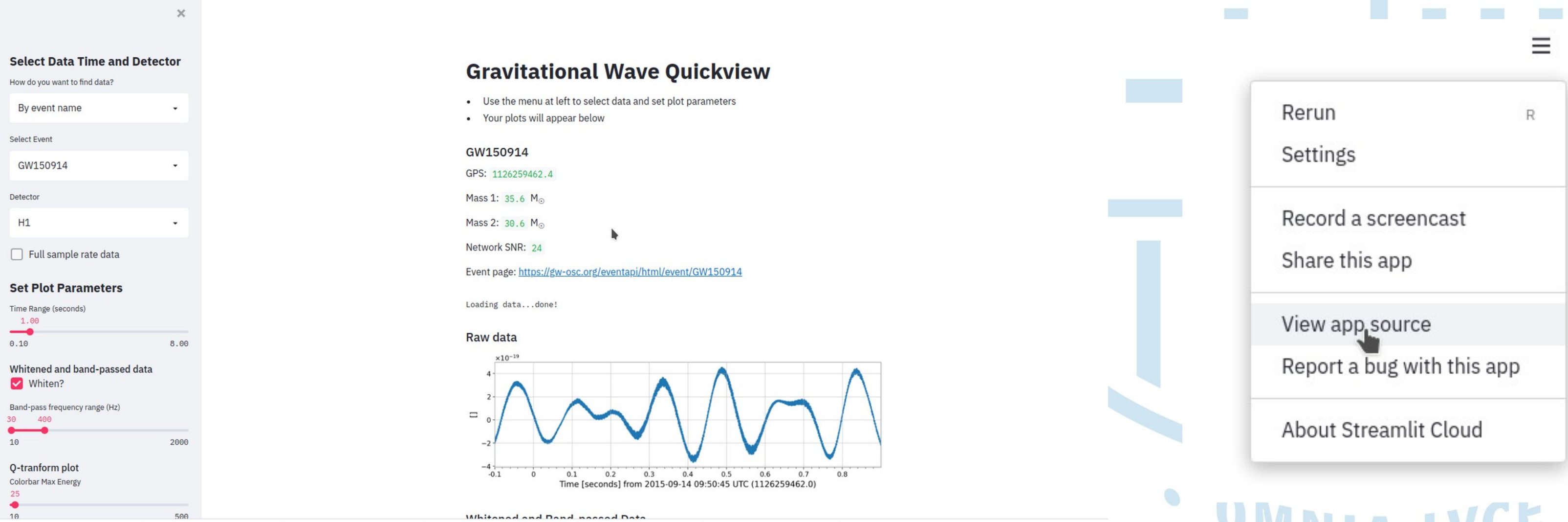


… and of course, all of you are skillful programmers.

Streamlit

= one and maybe the simplest way to create a dashboard with python

Example:
https://share.streamlit.io/jkanner/streamlit-dataview/master/app.py/+/



Select Data Time and Detector

How do you want to find data?
By event name

Select Event
GW150914

Detector
H1

□ Full sample rate data

Set Plot Parameters

Time Range (seconds)
1.00
0.10 8.00

Whitened and band-passed data
☑ Whiten?

Band-pass frequency range (Hz)
30 400
10 2000

Q-tranform plot
Colorbar Max Energy
25
10 500

Gravitational Wave Quickview

- Use the menu at left to select data and set plot parameters
- Your plots will appear below

GW150914

GPS: 1126259462.4

Mass 1: 35.6 $M_\odot$

Mass 2: 30.6 $M_\odot$

Network SNR: 24

Event page: https://gw-osc.org/eventapi/html/event/GW150914

```
Loading data...done!
```

Raw data

Whitened and Band-passed Data

Streamlit

Installation:

`pip install streamlit` or `conda install streamlit`

Execution:

```
streamlit run some_app.py
```

Quick tutorial:

```
streamlit hello
```

Gallery:

https://streamlit.io/gallery

Documentation:

https://docs.streamlit.io/library/api-reference

Streamlit / basic commands

```
import streamlit as st

# Text
st.write("Hello World!") # understands markdown

# Interactive widgets
st.selectbox("Choose!",("Option1", "Option2"))
st.number_input("How old are you?", value=18, step=1)

# Sidebar
st.sidebar.selectbox("Choose!",("Option1", "Option2"))

# Plots
from bokeh.plotting import figure, show
p = figure()
p.line([1, 2, 3], [4, 5, 6])
st.bokeh_chart(p)
```