ADS ML – Week 2

Reproducible Research & Exploratory Visualization with Matplotlib

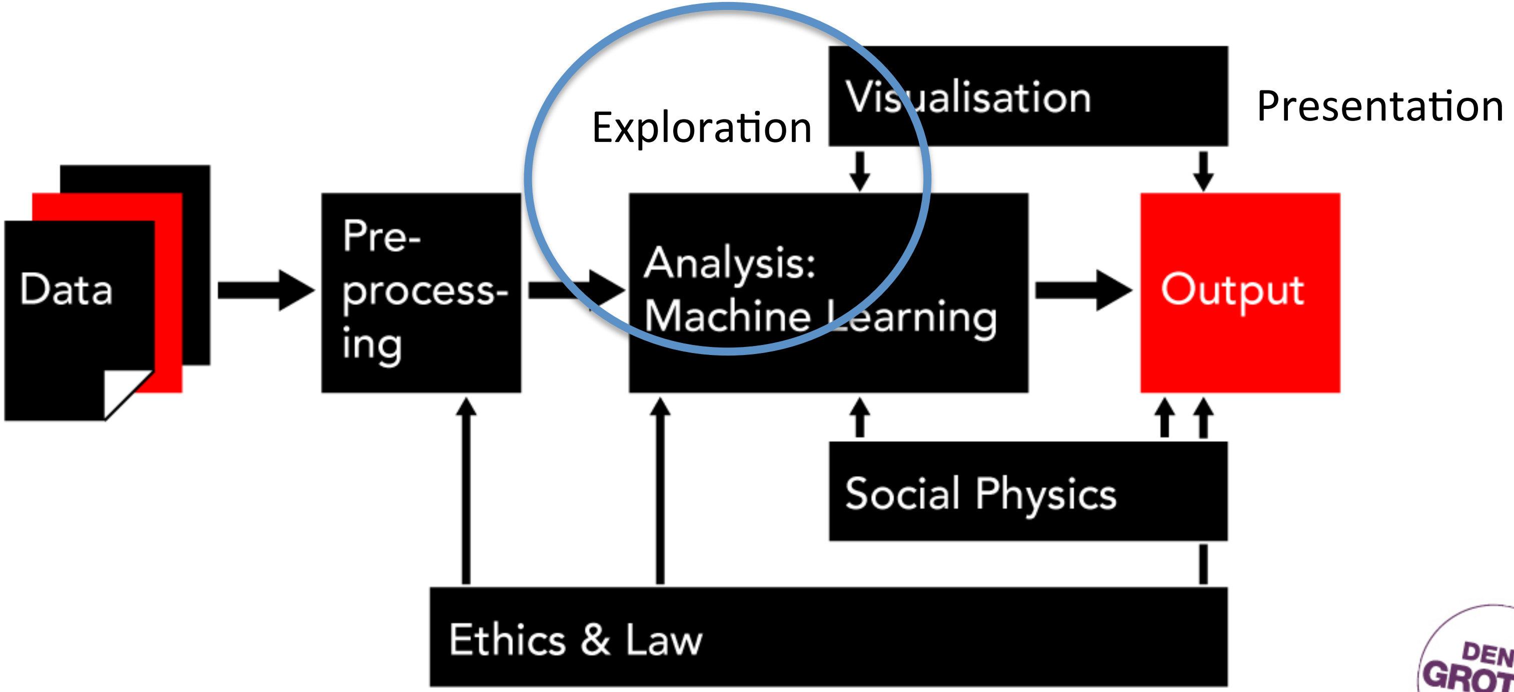
Goals: At the end of this lesson …

- You have created your personal portfolio.
- You know how to apply the rules of reproducible research to your iPython notebooks.
- You can create all parts for a scientific plot in Python with Matplotlib.
- You can create/recreate simple scatter, histogram, line plots and heatmaps.

Contribution to the learning objectives

- **create** a visualization from any data set that is not misleading and that clearly shows clustering, outliers and trends,
- **motivate** every design choice in a created visualization,
- **motivate** the next step in a data analysis based on a given visualization,
- **present** data analysis and visualizations as part of reproducible research,
- **apply** narrative techniques in visualizations,
- **create** engaging visualizations that allow for data exploration and story progression.

What are we doing?



Reproducible Research

Warning:

heavy text slides ahead

Reproducible Research

Reproducible research means that someone else could, in principle, redo **all** your actions. Reproducible research is all about **transparancy**.

It has nothing to do with **correctness or reliability**. Your research can be reproducible, yet completely false, unreliable and drawing the wrong conclusions. So why is it important then?

Reproducible Research

Reproducible research is important because, even though you can never guarantee that your work is correct, you at least ensure that **others have full insight** into what you have done.

Your lecturers can assess your work, and the company that wants to use your result can **build further on your work**.

In this minor we advise you to use a quite specific workflow to create reproducible research.

Reproducible Research Rules

- Use iPython Notebooks to combine your report with source code and its output.
- Use a version control system (Git is recommended).
- Include all data used in the data analysis in its rawest form and include a codebook when possible.
- You will NOT change the raw data by hand but in a Notebook.
- You will NOT use a GUI based program to edit or visualize data and include it as reproducible research. You can use such tools for data exploration or to produce more esthetic visualizations based on reproducible visualizations.
- Intermediate results can only be used if there is a Notebook creating them.

Based on:
https://github.com/rdpeng/courses/blob/master/05_ReproducibleResearch/lectures/Checklist.pdf

Assignment: Setting up your portfolio

Set up your personal **private** digital portfolio at:

https://git.fhict.nl/ or (private) https://github.com/ or …

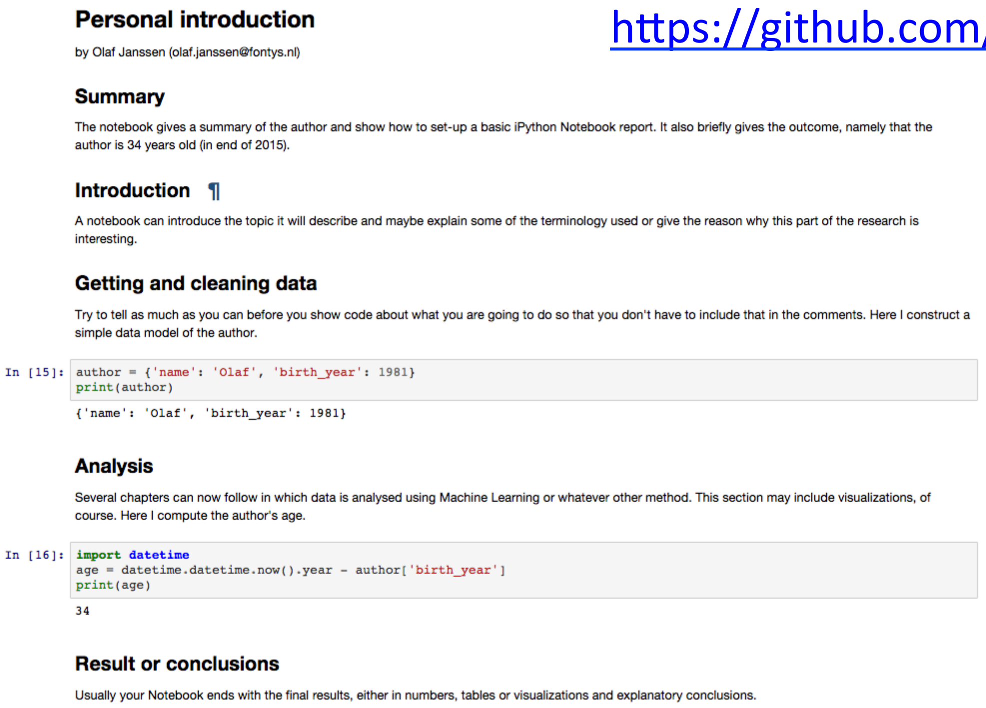
New at Git?

https://www.udacity.com/course/how-to-use-git-and-github--ud775 (lesson 2) or

https://www.youtube.com/watch?v=Y9XZQO1n_7c

Reproducible iPython Notebooks

https://github.com/olafjanssen/ads-dv



Personal introduction

by Olaf Janssen (olaf.janssen@fontys.nl)

Summary

The notebook gives a summary of the author and show how to set-up a basic iPython Notebook report. It also briefly gives the outcome, namely that the author is 34 years old (in end of 2015).

Introduction ¶

A notebook can introduce the topic it will describe and maybe explain some of the terminology used or give the reason why this part of the research is interesting.

Getting and cleaning data

Try to tell as much as you can before you show code about what you are going to do so that you don't have to include that in the comments. Here I construct a simple data model of the author.

```
In [15]: author = {'name': 'Olaf', 'birth_year': 1981}
         print(author)

{'name': 'Olaf', 'birth_year': 1981}
```

Analysis

Several chapters can now follow in which data is analysed using Machine Learning or whatever other method. This section may include visualizations, of course. Here I compute the author's age.

```
In [16]: import datetime
         age = datetime.datetime.now().year - author['birth_year']
         print(age)

34
```

Result or conclusions

Usually your Notebook ends with the final results, either in numbers, tables or visualizations and explanatory conclusions.

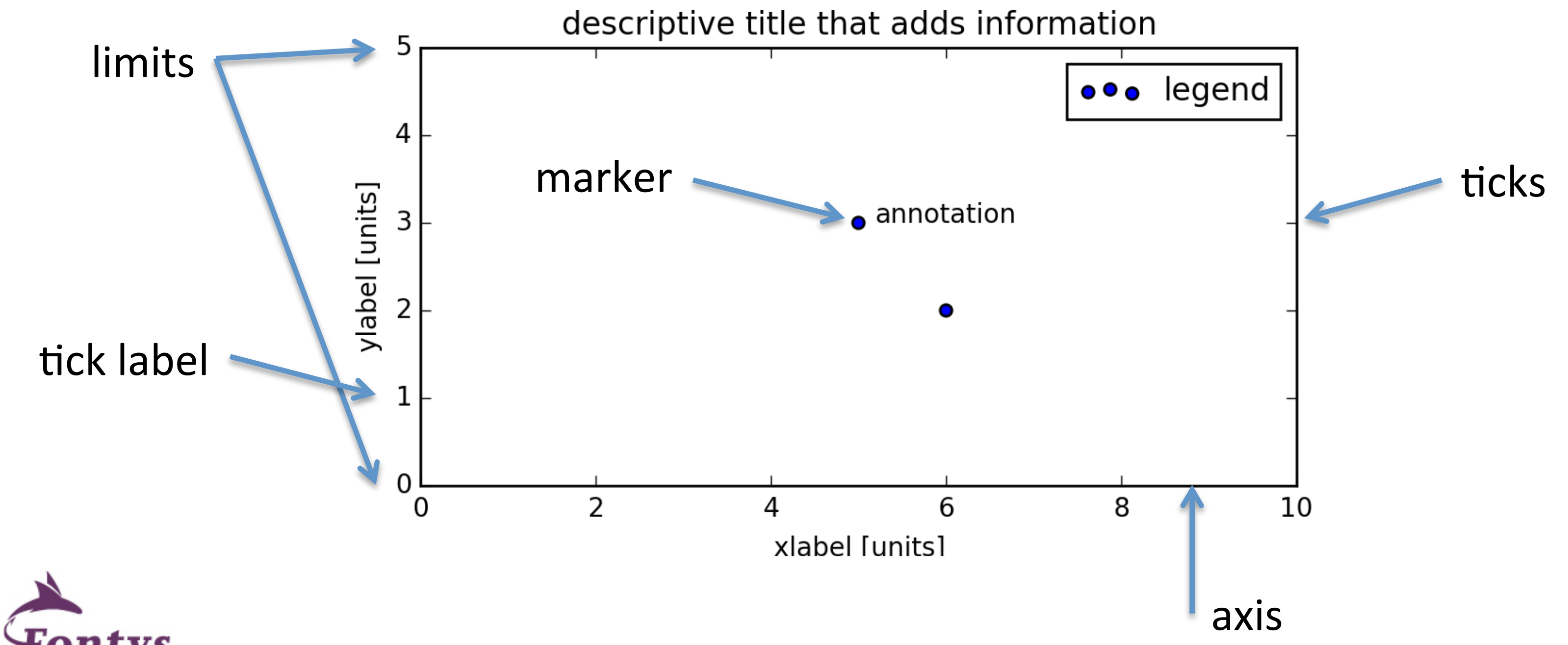
Fontys

DENK GROTER

Plotting with Matplotlib & All the Parts of a Chart

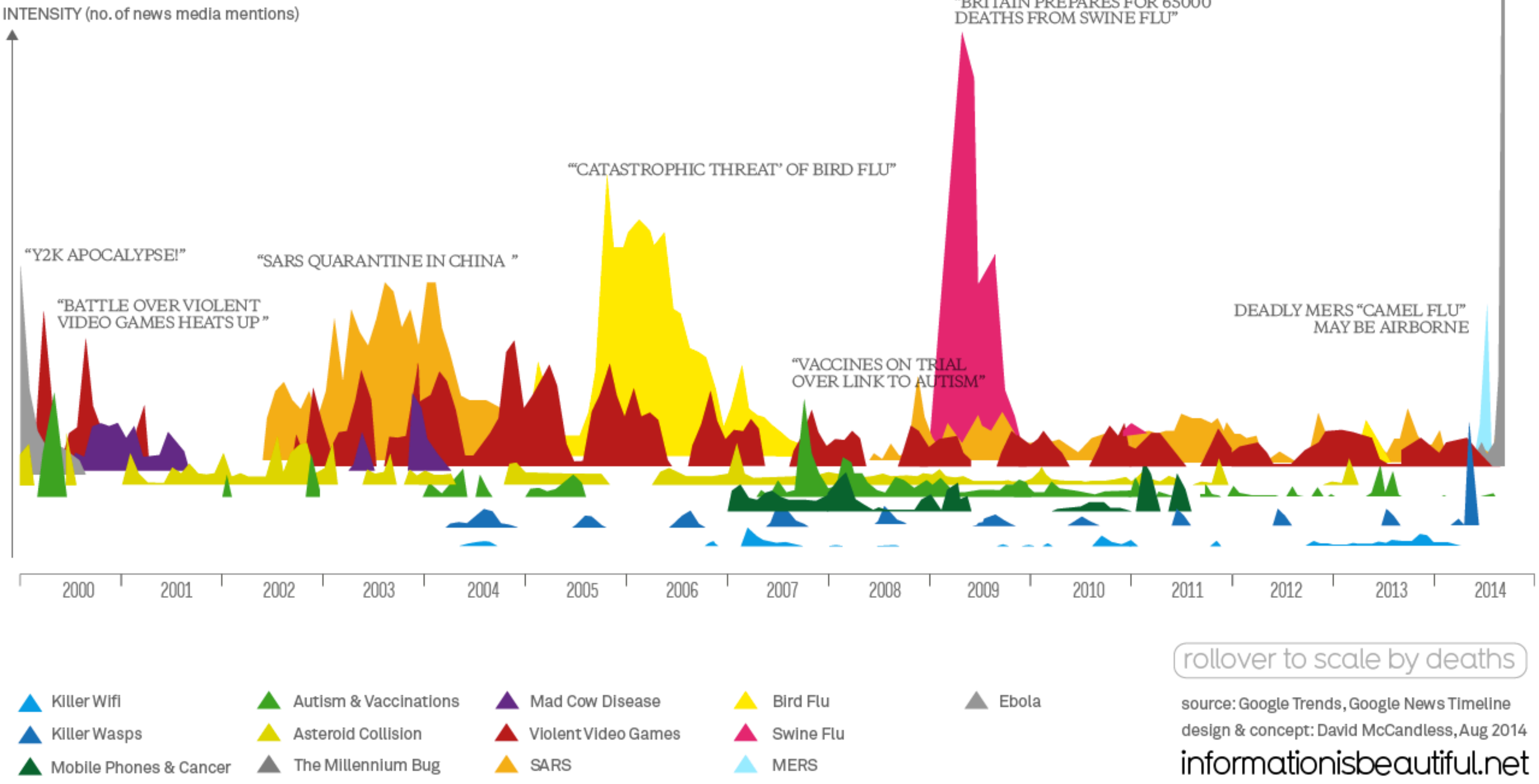
Parts of a Chart

Common parts of a graph that you must consider (if functional)



Mountains out of Molehills

A timeline of media-inflamed fears



source: Google Trends, Google News Timeline

design & concept: David McCandless, Aug 2014

informationisbeautiful.net

Chart Types

Scatter Plots,

Line Chart,

Bar Chart / Histogram,

Heat map

Do you know what they are?

More on other types and when to use what in the next few lessons.

Plotting with Matplotlib

- Matplotlib allows for plotting in Python using an easy API. It does not generate the most beautiful plots out-of-the-box.
- Still the most popular one.
- Alternatives are: Bokeh, Vincent, and ggplot.

- http://matplotlib.org/

Assignments 2-5

- Practice with scatter and line charts, histograms, and heat maps.
- Use the Notebooks in the Git repo as template and add your name and work to the Notebook.
- Add everything to you digital portfolio.

Before week 4:

Watch lectures **1a** en **2a** of the Udacity MOOC **Data Visualization and D3.js** on Visualization Fundamentals and Design Principles.