

Architektura Systemów Bioinformatycznych

Wykład 8.

Od “robienia wykresów” do wizualizacji danych
Częsty problem, wiele rozwiązań

Bartek Wilczyński

30.11.2015

Wizualizacja danych

- Co to znaczy wizualizacja danych
- Komercyjne pakiety – głównie z zeszłego wykładu
- Pionierzy opensource: gnuplot i latex
- Ważne pomysły:
 - E. Tufte - grafika elegancka i wierna danym
 - L. Wilkinson - gramatyka grafiki
- Nowa fala pakietów w ostatnim dziesięcioleciu:
 - Ggplot2, Shiny dla R'a
 - Matplotlib, Seaborn, Bokeh dla Python'a
 - D3.js dla java scriptu

Komercyjne pakiety do grafiki

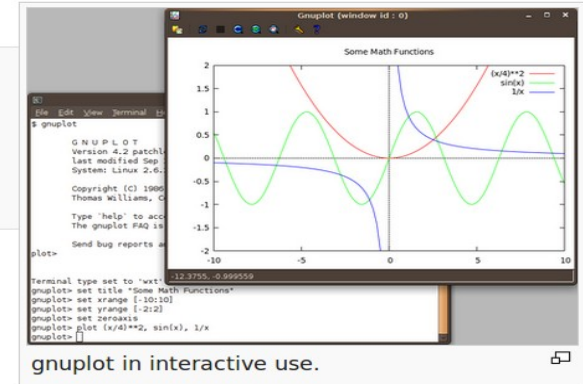
- W zasadzie powtórzenie z zeszłego tygodnia:
 - Excel, Excel, Excel, Excel,
 - Matlab, Mathematica,
 - SPSS, STATISTICA
 - Dalej już specjalizowane oprogramowanie, typu CAD, CAM, itp.
 - Także dla biznesu cały arsenał specjalistycznych narzędzi typu OLAP, często jako “wtyczki” do Excel'a

GNUplot – jedno z pierwszych narzędzi, wcale nie GNU, ale opensource

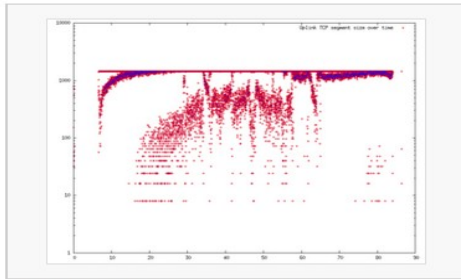
The gnuplot core code is programmed in C. Modular subsystems for output via Qt, wxWidgets, and LaTeX/TikZ/ConTeXt are written in C++ and lua.

The code below creates the graph to the right.

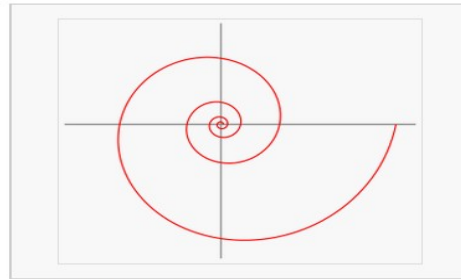
```
set title "Some math functions"
set xrange [-10:10]
set yrange [-2:2]
set zeroaxis
plot (x/4)**2, sin(x), 1/x
```



gnuplot in interactive use.



A [scatter plot](#) of samples from a text file.



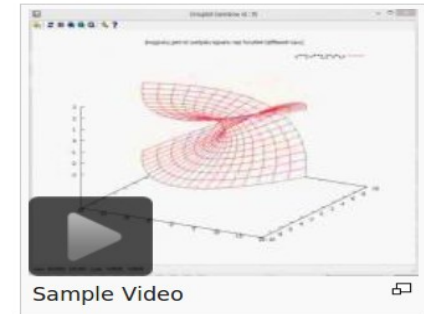
A logarithmic spiral.

The name of this program was originally chosen to avoid conflicts with a program called "newplot", and was originally a compromise between "llamaplot" and "nplot".^[4]

Distribution terms [\[edit \]](#)

Despite gnuplot's name, it is not named after, part of or related to the [GNU Project](#), nor does it use the [GNU General Public License](#). It was named as part of a compromise by the original authors, punning on *gnu* (the animal) and *newplot*.^[5]

Official source code to gnuplot is freely redistributable, but modified versions thereof are not. The gnuplot license instead recommends distribution of patches against official releases, optionally accompanied by officially released source code. Binaries may be distributed along with the unmodified source code and any patches applied thereto. Contact information must be supplied with derived works for [technical support](#) for the modified



Sample Video

TikZ ist kein zeichenprogramm, pgf, LaTeX

[Home](#) > [TikZ](#) > [Examples](#)


TikZ and PGF examples

Welcome to the PGF and TikZ examples gallery.

Browse by: [Features](#) | [Tags](#) | [Technical areas](#) | [Non-technical areas](#) | [Authors](#)

Navigation

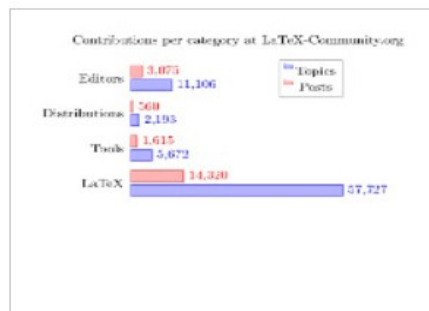
- [Gallery main page](#)
- [About the gallery](#)
- [Contribute](#)
- [Show all examples](#)

 [Subscribe to the TikZ examples RSS feed](#)

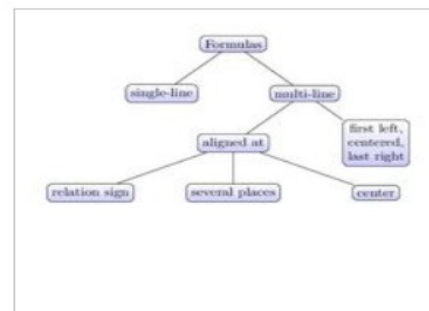
Recently added examples



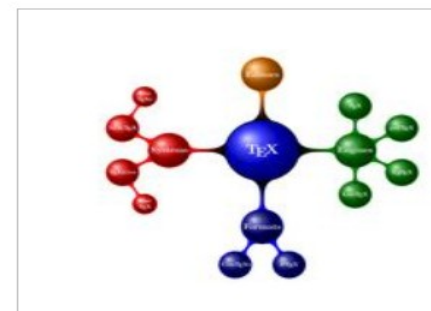
A mindmap showing TeX projects supported by DANTE e.V.
[\[PDF\]](#) [\[TEX\]](#) [\[Open in writeLaTeX\]](#)



Bar chart
[\[PDF\]](#) [\[TEX\]](#) [\[Open in writeLaTeX\]](#)



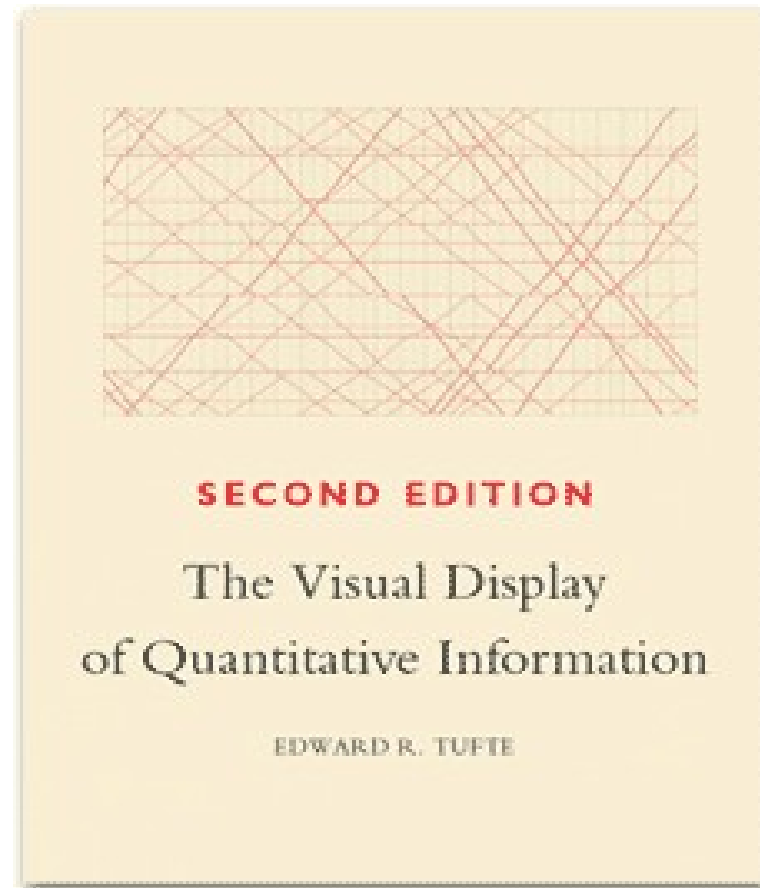
A simple Tree
[\[PDF\]](#) [\[TEX\]](#) [\[Open in writeLaTeX\]](#)



Mindmap
[\[PDF\]](#) [\[TEX\]](#) [\[Open in writeLaTeX\]](#)

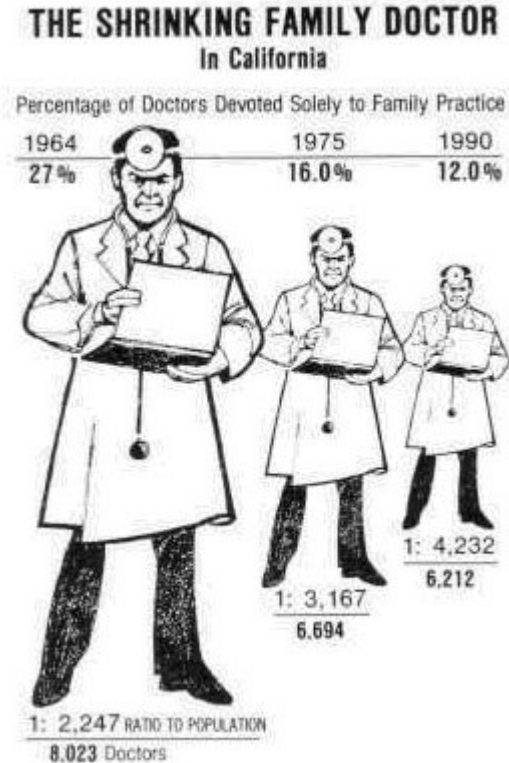
Edward R. Tufte – wizualizacja musi być estetyczna i nie kłamać

- Opublikował ogromny best-seller, *The Visual Display of Quantitative Information* w 1993 (ponad 20 “dodruków”, dwa wydania)
- Nie znosił generycznych wykresów z Excela i powerpointa (np. Pie chart)
- Spopularyzował “lie factor”



Lie Factor

- Współczynnik tego jak bardzo proporcje na wykresie różnią się od rzeczywistych proporcji w danych
- Tutaj np. Mamy rzeczywisty spadek z 8.023 na 6.212, gdy infografika przedstawia mniej więcej 6-krotne zmniejszenie “lekarza”



Carte Figurative des pertes successives en hommes de l'Armée Française dans la campagne de Russie 1812-1813.

Dressée par M. MINARD, Inspecteur Général des Ponts et Chaussées en retraite. Paris, le 20 Novembre 1869.

Les nombres d'hommes perdus sont représentés par les largeurs des zones colorées à raison d'un millimètre pour dix mille hommes; ils sont de plus écrits en travers des zones. Le rouge désigne les hommes qui ont péri en Russie, le noir ceux qui en sortirent. Les renseignements qui ont servi à dresser la carte ont été puisés dans les ouvrages de M. M. Chiers, de Séjourné, de Fezensac, de Chambray et le journal inédit de Jacob, pharmacien de l'Armée depuis le 28 Octobre. Pour mieux faire juger à l'œil la diminution de l'armée, j'ai supposé que les corps du Prince Jérôme et du Maréchal Davout qui avaient été détachés sur Minsk et Mohilew n'avaient rejoint vers Orscha ou Witebsk, avaient toujours marché avec l'armée.

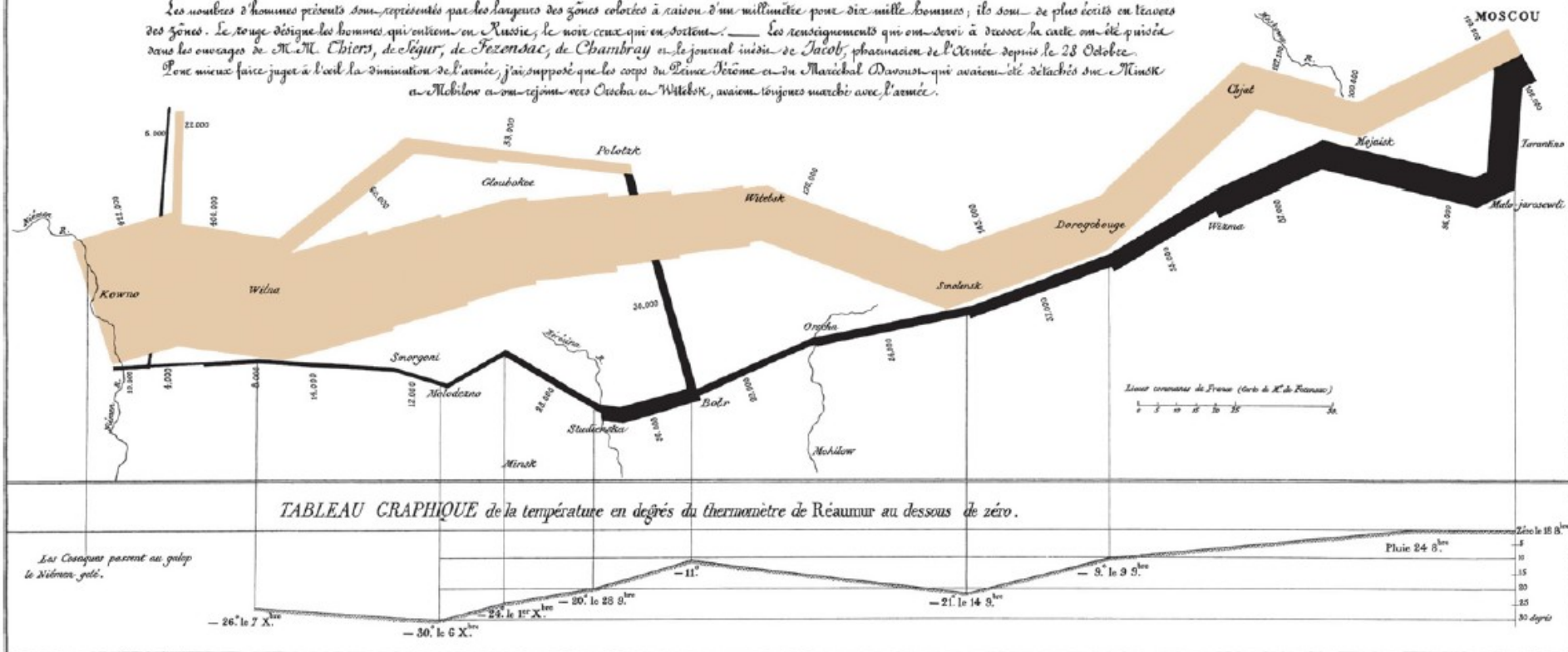
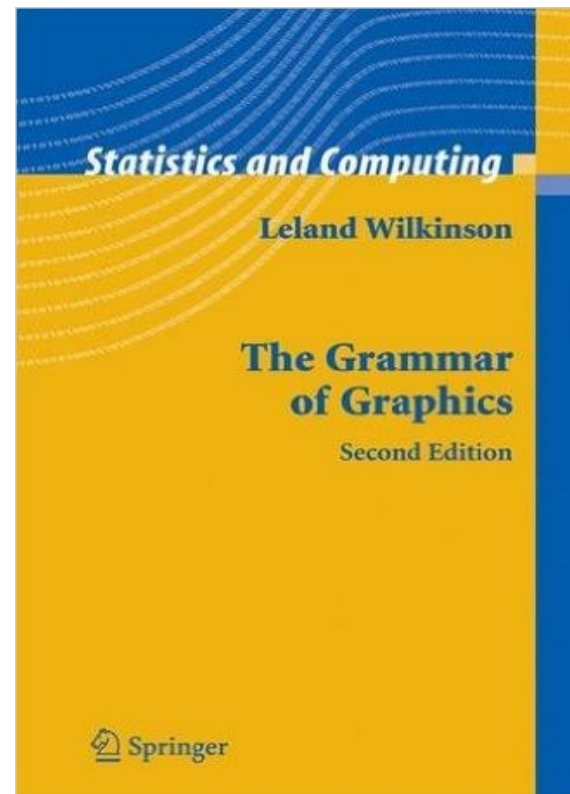


Figure 11. “Carte figurative des pertes successives en hommes de l’Armée Français dans la campagne de Russe 1812–1813” by Charles Joseph Minard. Public domain image from <http://en.wikipedia.org/wiki/File:Minard.png>.

L. Wilkinson – the Grammar of graphics

- Tutaj nieco inna historia,
- Pracownik uniwersytetu i firm komercyjnych (m.in. SPSS)
- Autor książki the grammar of graphics (1999), i wzorowanego na niej systemu nVizN
- Rozpoznawalny tylko w kręgach specjalistów



```

ELEMENT: point(position(birth*death), size(0), label(country))
ELEMENT: contour(position(
    smooth.density.kernel.epanechnikov.joint(birth*death)),
    color.hue())
GUIDE: form.line(position((0,0),(30,30)), label("Zero Population Growth"))
GUIDE: axis(dim(1), label("Birth Rate"))
GUIDE: axis(dim(2), label("Death Rate"))

```

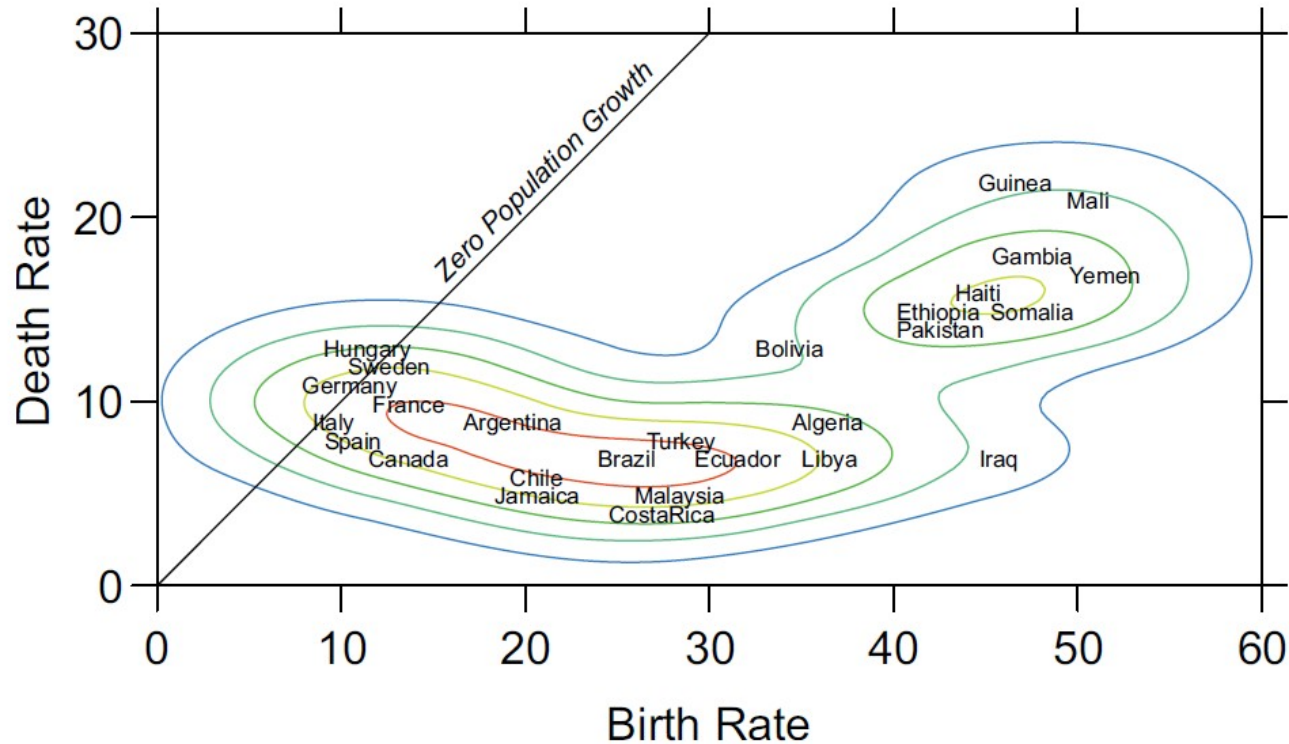
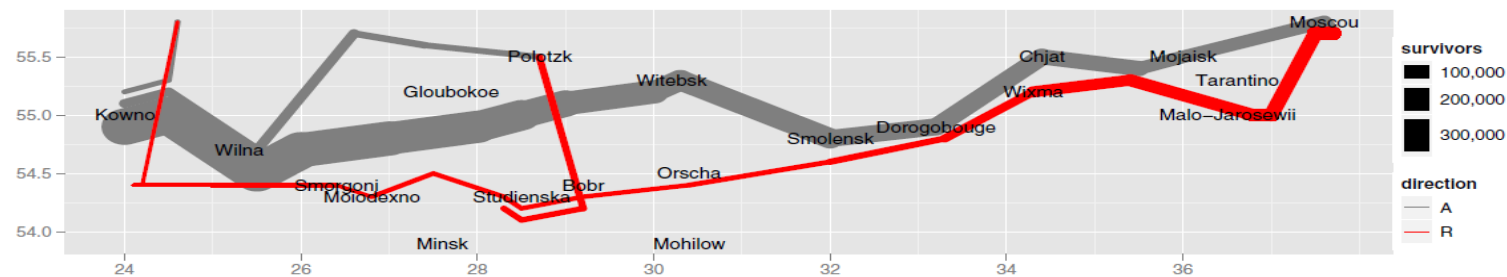
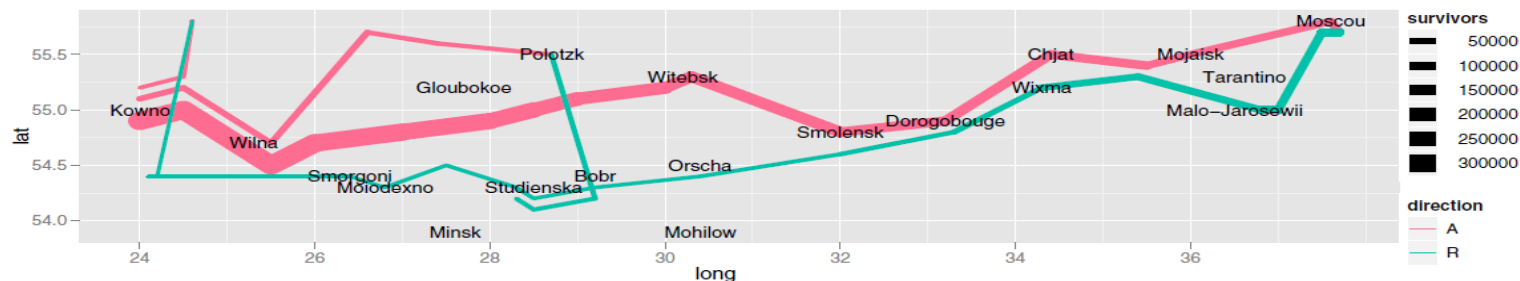
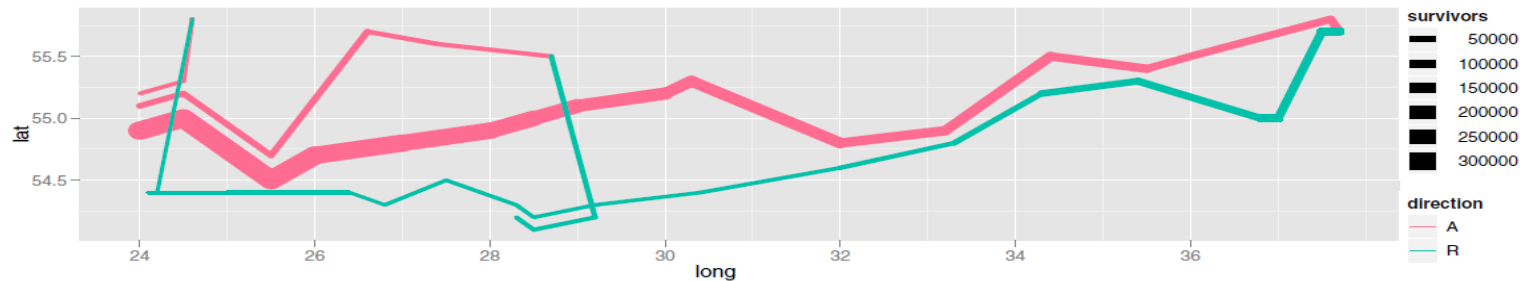


Figure 1.1 Plot of death rates against birth rates for selected countries Wilkinson, 1999

Ggplot2 – przełomowa biblioteka dla R



A Layered Grammar of Graphics

Hadley WICKHAM

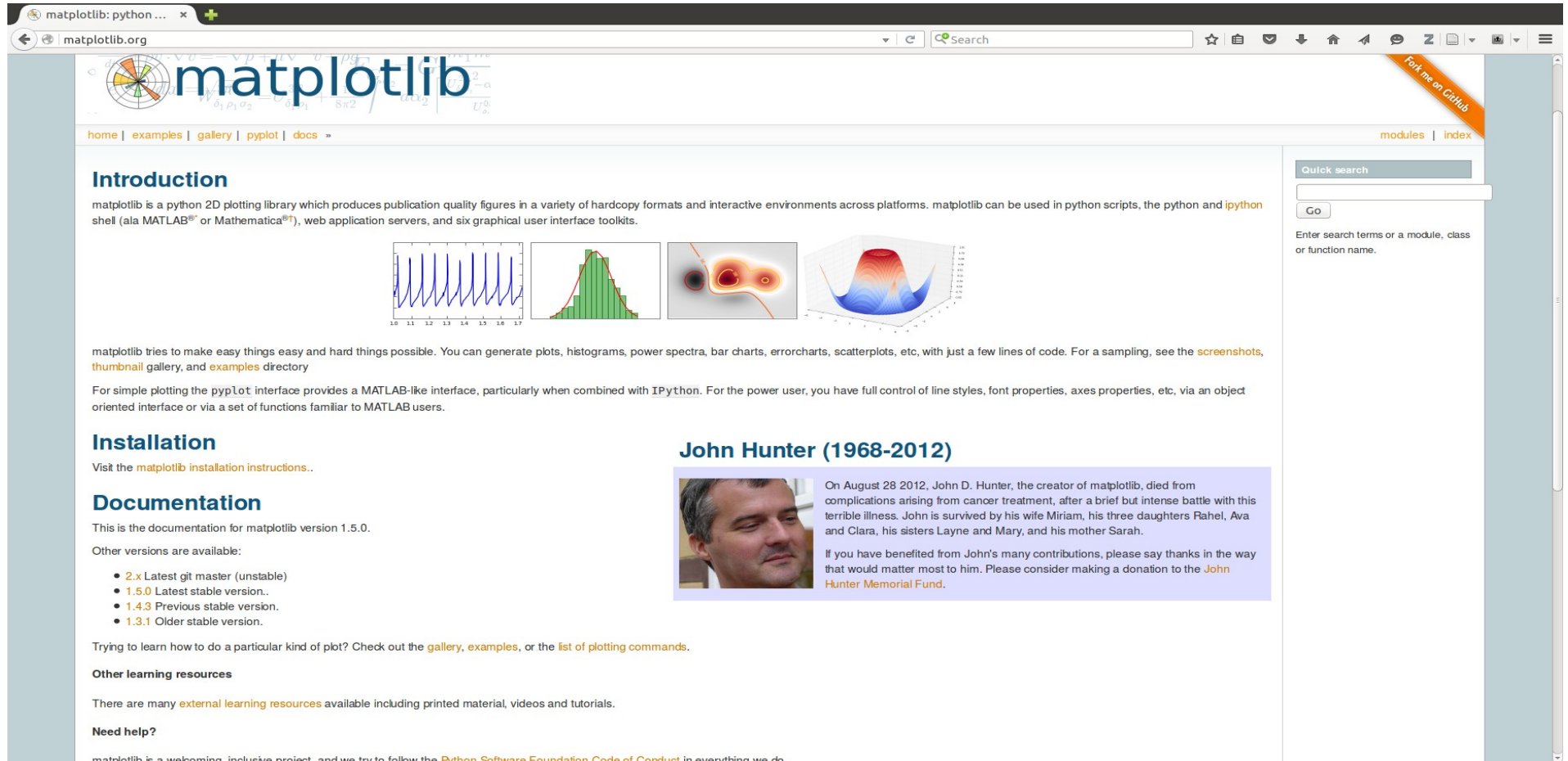
A grammar of graphics is a tool that enables us to concisely describe the components of a graphic. Such a grammar allows us to move beyond named graphics (e.g., the “scatterplot”) and gain insight into the deep structure that underlies statistical graphics. This article builds on Wilkinson, Anand, and Grossman (2005), describing extensions and refinements developed while building an open source implementation of the grammar of graphics for R, `ggplot2`.

The topics in this article include an introduction to the grammar by working through the process of creating a plot, and discussing the components that we need. The grammar is then presented formally and compared to Wilkinson’s grammar, highlighting the hierarchy of defaults, and the implications of embedding a graphical grammar into a programming language. The power of the grammar is illustrated with a selection of examples that explore different components and their interactions, in more detail. The article concludes by discussing some perceptual issues, and thinking about how we can build on the grammar to learn how to create graphical “poems.”

Supplemental materials are available online.

Key Words: Grammar of graphics; Statistical graphics.

Matplotlib – od matlaba dla python'a do standardu samego dla siebie



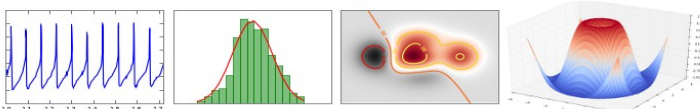
matplotlib: python ... x +

matplotlib.org

home | examples | gallery | pyplot | docs >

Introduction

matplotlib is a python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. matplotlib can be used in python scripts, the python and `ipython` shell (ala MATLAB[®] or Mathematica[®]), web application servers, and six graphical user interface toolkits.



matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc, with just a few lines of code. For a sampling, see the [screenshots](#), [thumbnail](#) gallery, and [examples](#) directory

For simple plotting the `pyplot` interface provides a MATLAB-like interface, particularly when combined with `IPython`. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

Installation

Visit the [matplotlib installation instructions](#).

Documentation

This is the documentation for matplotlib version 1.5.0.

Other versions are available:

- [2.x](#) Latest git master (unstable)
- [1.5.0](#) Latest stable version..
- [1.4.3](#) Previous stable version.
- [1.3.1](#) Older stable version.

Trying to learn how to do a particular kind of plot? Check out the [gallery](#), [examples](#), or the [list of plotting commands](#).


Other learning resources

There are many [external learning resources](#) available including printed material, videos and tutorials.

Need help?

matplotlib is a welcoming, inclusive project, and we try to follow the [Python Software Foundation Code of Conduct](#) in everything we do.

John Hunter (1968-2012)



On August 28 2012, John D. Hunter, the creator of matplotlib, died from complications arising from cancer treatment, after a brief but intense battle with this terrible illness. John is survived by his wife Miriam, his three daughters Rahel, Ava and Clara, his sisters Layne and Mary, and his mother Sarah.

If you have benefited from John's many contributions, please say thanks in the way that would matter most to him. Please consider making a donation to the [John Hunter Memorial Fund](#).

modules | index

Quick search

Go

Enter search terms or a module, class or function name.

Font me on GitHub

PANDAs – python data library



The screenshot shows a Jupyter Notebook interface with a sidebar on the left containing a navigation menu. The main area displays a code cell with the following content:

```
In [6]: dates = pd.date_range('20130101', periods=6)

In [7]: dates
Out[7]:
DatetimeIndex(['2013-01-01', '2013-01-02', '2013-01-03', '2013-01-04',
              '2013-01-05', '2013-01-06'],
              dtype='datetime64[ns]', freq='D')

In [8]: df = pd.DataFrame(np.random.randn(6,4), index=dates, columns=list('ABCD'))

In [9]: df
Out[9]:
```

| | A | B | C | D |
|------------|-----------|-----------|-----------|-----------|
| 2013-01-01 | 0.469112 | -0.282863 | -1.509059 | -1.135632 |
| 2013-01-02 | 1.212112 | -0.173215 | 0.119209 | -1.044236 |
| 2013-01-03 | -0.861849 | -2.104569 | -0.494929 | 1.071804 |
| 2013-01-04 | 0.721555 | -0.706771 | -1.039575 | 0.271860 |
| 2013-01-05 | -0.424972 | 0.567020 | 0.276232 | -1.087401 |
| 2013-01-06 | -0.673690 | 0.113648 | -1.478427 | 0.524988 |

Creating a `DataFrame` by passing a dict of objects that can be converted to series-like.

```
In [10]: df2 = pd.DataFrame({'A': 1.,
.....:                      'B': pd.Timestamp('20130102'),
.....:                      'C': pd.Series(1, index=list(range(4)), dtype='float32'),
.....:                      'D': np.array([3] * 4, dtype='int32'),
.....:                      'E': pd.Categorical(["test", "train", "test", "train"]),
.....:                      'F': 'foo' })

In [11]: df2
Out[11]:
```

| | A | B | C | D | E | F |
|---|---|------------|---|---|-------|-----|
| 0 | 1 | 2013-01-02 | 1 | 3 | test | foo |
| 1 | 1 | 2013-01-02 | 1 | 3 | train | foo |
| 2 | 1 | 2013-01-02 | 1 | 3 | test | foo |
| 3 | 1 | 2013-01-02 | 1 | 3 | train | foo |

Having specific `dtypes`

The sidebar on the left contains a navigation menu with the following items:

- HDF5
- Excel
- Gotchas
- Tutorials
- Cookbook
- Intro to Data Structures
- Essential Basic Functionality
- Working with Text Data
- Options and Settings
- Indexing and Selecting Data
- Multindex / Advanced Indexing
- Computational tools
- Working with missing data
- Group By: split-apply-combine
- Merge, join, and concatenate
- Reshaping and Pivot Tables
- Time Series / Date functionality
- Time Deltas
- Categorical Data
- Visualization
- Style
- IO Tools (Text, CSV, HDF5, ...)
- Remote Data Access
- Enhancing Performance
- Sparse data structures
- Caveats and Gotchas
- rpy2 / R interface
- pandas Ecosystem
- Comparison with R / R libraries
- Comparison with SQL
- Comparison with SAS
- API Reference
- Internals
- Release Notes

Search

Matplotlib + PANDAs = Seaborn

Hexbin plot with ma... x +

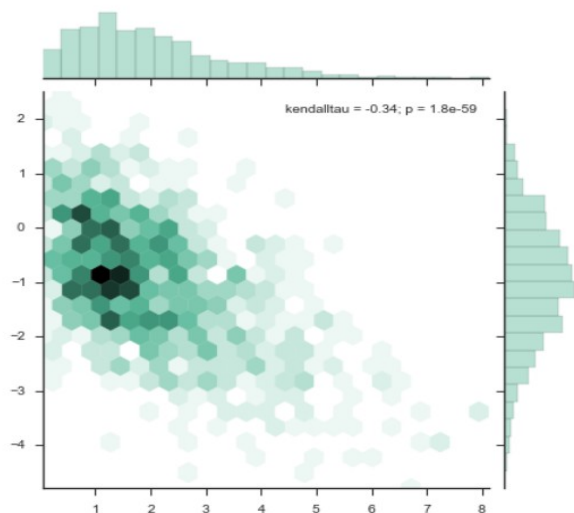
stanford.edu/~mwaskom/software/seaborn/examples/hexbin_marginals.html

Search

seaborn 0.6.0 API Tutorial Gallery Site Page

Search

Hexbin plot with marginal distributions



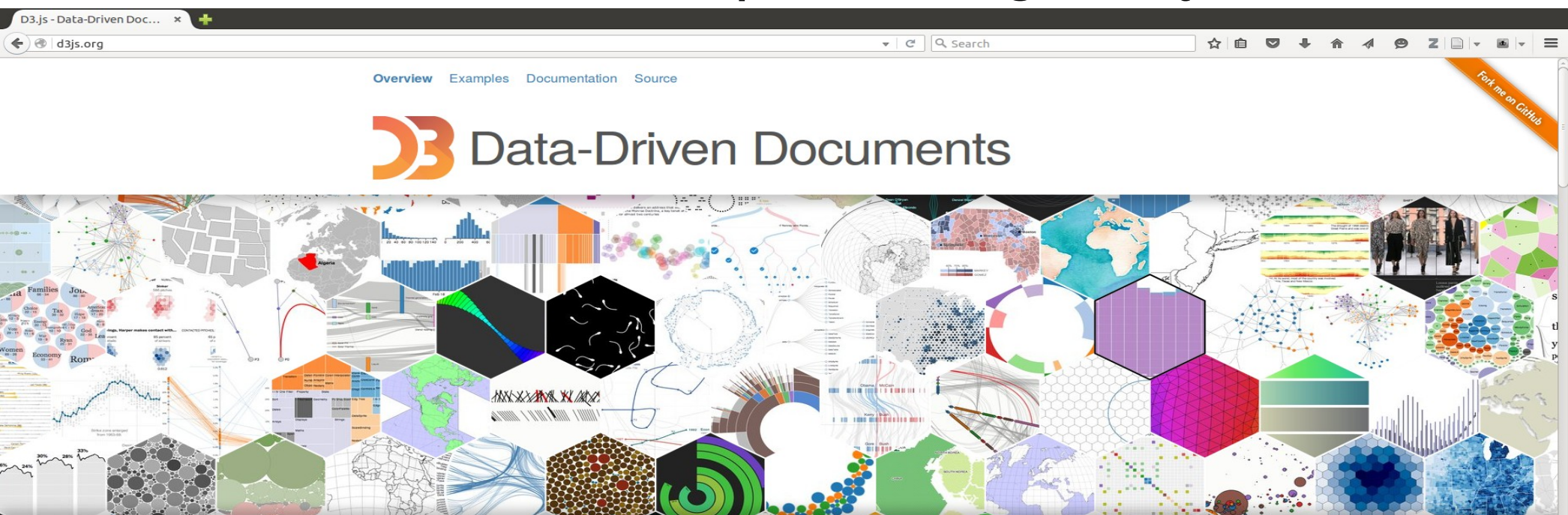
Python source code: [\[download source: hexbin_marginals.py\]](#)

```
import numpy as np
from scipy.stats import kendalltau
import seaborn as sns
sns.set(style="ticks")

rs = np.random.RandomState(11)
x = rs.gamma(2, size=1000)
y = -.5 * x + rs.normal(size=1000)

sns.jointplot(x, y, kind="hex", stat_func=kendalltau, color="#4CB391")
```

D3s – Mike Bostock przenosi grafikę do www



D3.js is a JavaScript library for manipulating documents based on data. **D3** helps you bring data to life using HTML, SVG, and CSS. D3's emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components and a data-driven approach to DOM manipulation.

[See more examples.](#)

Download the latest version (3.5.10) here:

- [d3.zip](#)

Or, to link directly to the latest release, copy this snippet:

```
<script src="//d3js.org/d3.v3.min.js" charset="utf-8"></script>
```


Bokeh – biblioteka przenosząca matplotlib do www

Welcome to Bokeh ... x

bokeh.pydata.org/en/latest/

Bokeh 0.10.0 Gallery Site Page

Quickstart
Installation
Gallery
User Guide
Tutorials
Contributing
Frequently Asked Questions
Release Notes and Roadmap
Reference Guide
Developer Guide
Source

Google™ Custom Search

Welcome to Bokeh

Bokeh is a Python interactive visualization library that targets modern web browsers for presentation. Its goal is to provide elegant, concise construction of novel graphics in the style of D3.js, but also deliver this capability with high-performance interactivity over very large or streaming datasets. Bokeh can help anyone who would like to quickly and easily create interactive plots, dashboards, and data applications.

For more information about the goals and direction of the project, please see the [Technical Vision](#).

To get started quickly, follow the [Quickstart](#).

To see examples of how you might use Bokeh with your own data, check out the [Gallery](#).


For questions and technical assistance, come join the [Bokeh mailing list](#).

Visit the [GitHub source repository](#).

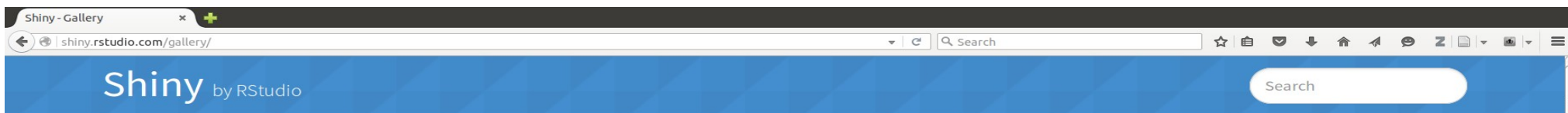
Be sure to follow us on Twitter [@bokehplots](#), as well as on [Vine](#), and [Youtube!](#)

Fork me on GitHub

Contents



Shiny – ggplot2 też jest już w sieci



OVERVIEW

TUTORIAL

ARTICLES

GALLERY

REFERENCE

DEPLOY

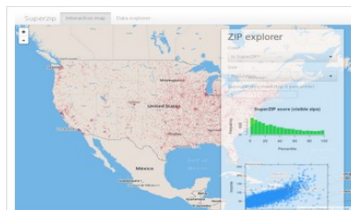
HELP

Gallery

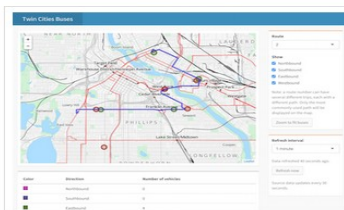
This gallery contains useful examples to learn from. Visit the [Shiny User Showcase](#) to see an inspiring set of sophisticated apps.

Interactive visualizations

Shiny is designed for fully interactive visualization, using JavaScript libraries like [d3](#), [Leaflet](#), and [Google Charts](#).



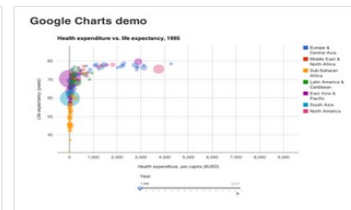
SuperZip example



Bus dashboard



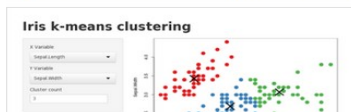
Movie explorer



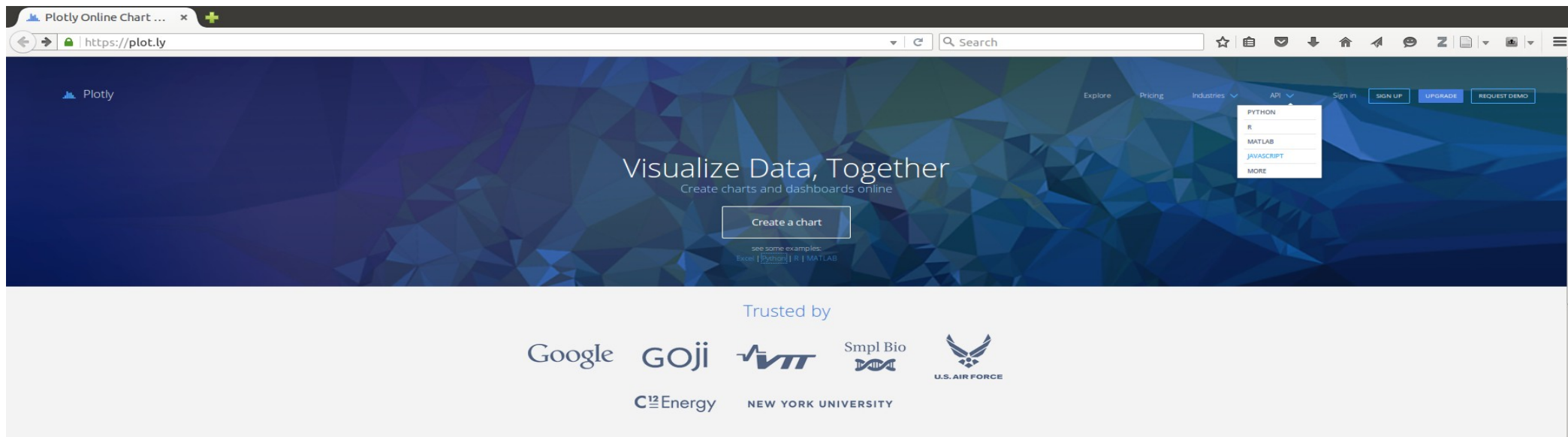
Google Charts

Start simple

If you're new to Shiny, these simple but complete applications are designed for you to study.



Plot.ly – komercyjny ale open Source



The screenshot shows the Plot.ly website homepage. The browser address bar displays "https://plot.ly". The main navigation menu includes "Explore", "Pricing", "Industries", and "API". A dropdown menu for "API" is open, listing "PYTHON", "R", "MATLAB", "JAVASCRIPT", and "MORE". The main heading reads "Visualize Data, Together" with the subtext "Create charts and dashboards online". A prominent "Create a chart" button is centered. Below it, a link says "see some examples: Excel | Python | R | MATLAB". The "Trusted by" section features logos for Google, GOji, VTT, Smpl Bio, U.S. AIR FORCE, C³Energy, and NEW YORK UNIVERSITY.

Plotly Online Chart ... x

https://plot.ly

Plotly

Explore Pricing Industries API Sign in SIGN UP UPGRADE REQUEST DEMO

PYTHON
R
MATLAB
JAVASCRIPT
MORE

Visualize Data, Together

Create charts and dashboards online

Create a chart

see some examples:
Excel | Python | R | MATLAB

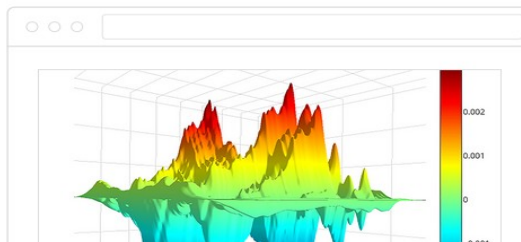
Trusted by

Google GOji VTT Smpl Bio U.S. AIR FORCE
C³Energy NEW YORK UNIVERSITY

Charts that stand out

Create a chart

see some examples:
Excel | Python | R | MATLAB



Podsumowanie

- Mamy wpływowych autorów: E.R. Tufte, L. Wilkinson, dających idee do implementacji
- Mamy serię doktorantów, którzy zajmowali się czymś innym niż doktoratem:
 - Hadley Wickham (ggplot)
 - Wes McKinney (pandas)
 - John Hunter (matplotlib)
 - Mike Bostock (d3.js)
- I firm, które płacą za ich pracę: Rstudio, Continuum Analytics, Cloudera i rozwój narzędzi

A propos estetyki – palety kolorów Brewer

The screenshot displays the ColorBrewer 2.0 web application interface. The browser address bar shows "colorbrewer2.org". The main interface includes a control panel on the left and a map on the right. The control panel is set to "Number of data classes: 3" and "Nature of your data: sequential". Under "Pick a color scheme:", the "Multi-hue" section shows a 3-class sequential palette with three color swatches: light blue (#e5f5f9), medium green (#99d8c9), and dark green (#2ca25f). The "Single hue" section shows a grayscale palette. The "Only show:" section has "colorblind safe", "print friendly", and "photocopy safe" options. The "Context:" section has "roads", "cities", and "borders" options, with "borders" selected. The "Background:" section has "solid color" and "terrain" options, with "solid color" selected. The "3-class BuGn" section shows the selected color palette with hex codes: #e5f5f9, #99d8c9, and #2ca25f. The map on the right shows a map of Poland with a 3-class sequential color scheme applied to its administrative boundaries. The map is titled "COLORBREWER 2.0 color advice for cartography".