

Tools

Last updated: 18. Juni 2017

Jörg Cassens




Data and Process Visualization
SoSe 2017







Inhaltsverzeichnis

1 Applications	2
2 Programming	6
3 Tutorial	16

Work in Progress

- This list of tools is work in progress
- In the “Application” section, you will usually see a screenshot of the application in question since they are mostly GUI tools
- In the “Programming” section, you will most often have links to programming examples
- Known missing tools to be added include:
 - Visio and similar process visualization tools
 - yED  www.yworks.com/products/yed
 - GeoGebra  www.geogebra.org
 - graphviz/dot  www.graphviz.org

Work in Progress

- Missing tools added with update 18 June
 - Shiny  shiny.rstudio.com
 - plotly  plot.ly
 - Bokeh  bokeh.github.io
 - HoloViews  holoviews.org
 - Jupyter and Zeppelin Notebooks

1 Applications

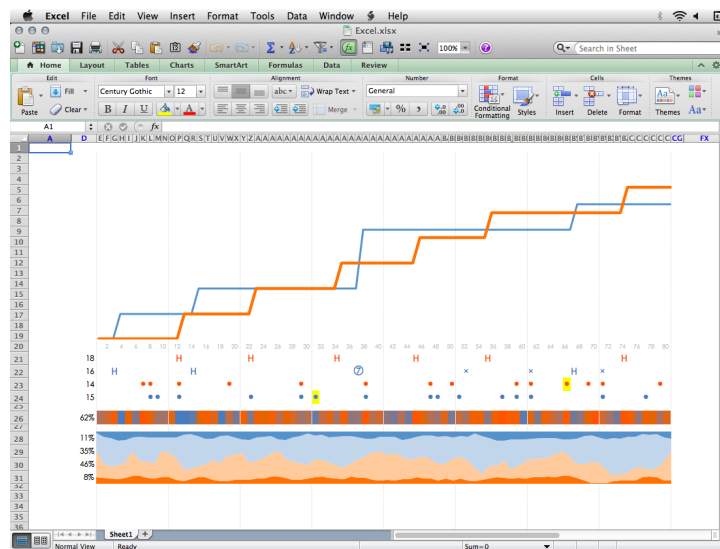
Spreadsheets

- Spreadsheet software, universal and has been around for decades
- A lot of data is made available as an Excel spreadsheet
- Easy to highlight columns and make a few charts, so you can get a quick idea of what your data looks like
- Not necessarily fit for thorough analysis or graphics made for publication
 - Limited by the amount of data it can handle at once
 - Unless you know Visual Basic for Applications (VBA) it can be a chore to reproduce charts for different data-sets
- Basically the same applies to LibreOffice or OpenOffice

MS Excel

- Within the data visualization world, Excel's charting capabilities are somewhat derided largely down to the terrible default settings and the range of bad-practice charting functions it enables
 - 3D cone charts, anyone?
- However, Excel does allow you to do much more than you would expect and, when fully exploited, it can prove to be quite a valuable ally
- With experience and know-how, you can control and refine many chart properties and you will find that most of your basic charting requirements are met, certainly those that you might associate more with a pragmatic or analytical tone

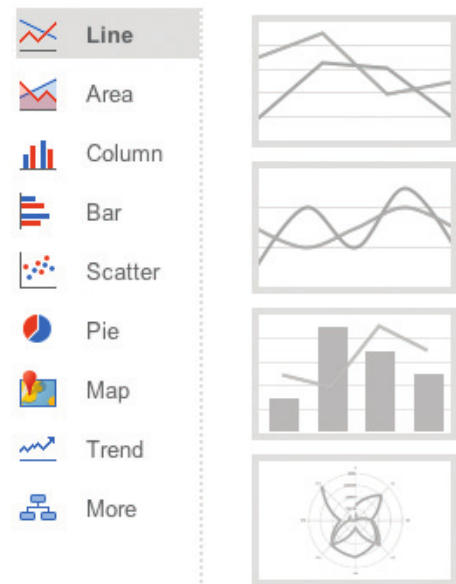
MS Excel Screenshot



Source: Kirk (2012)

Google Spreadsheets

- Essentially Google's version of Microsoft Excel, but it's simpler and online
- Online feature is the main plus because you can quickly access your data across different machines and devices
 - You can collaborate via built-in chat and real-time editing
 - You can also import HTML and XML files from the web using the importHTML and importXML functions, respectively



Source: Yau (2013)

Tableau Software

- Tableau Software is often the go-to analysis software
- If you want to dig deeper into your data than you can in Excel, without programming, this is a good place to look
- The program is visually-based, and you can easily interact with your data as you find interesting spots to look at
- The downside is that the software is pricey (with special pricing for students and nonprofits)
- For Windows and Mac OS X version
- Tableau Public is free to use and enables you to put together dashboards with a variety of charts and publish online
- As the name suggests, you must make your data public and upload it to Tableau servers

Tableau Screenshot

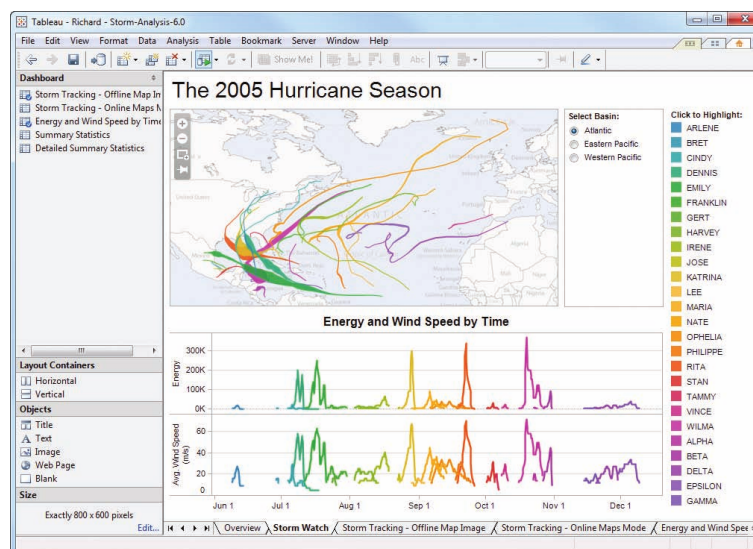


tableau.com

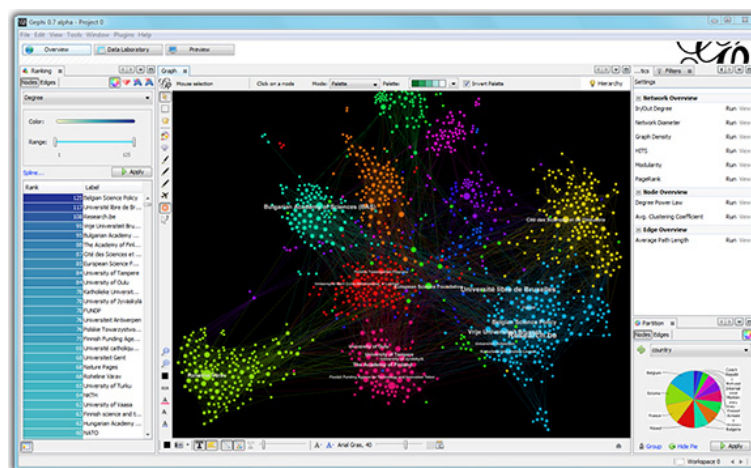
Tableau Usage

- Tableau is particularly valuable when it comes to the important stage of data familiarization
- When you want to quickly discover the properties, the shapes and quality of your data, Tableau is a great solution
- It also enables you to create embeddable interactive visualizations and, like Excel, lets you export charts as images for use in other applications

Graphs and Treemaps

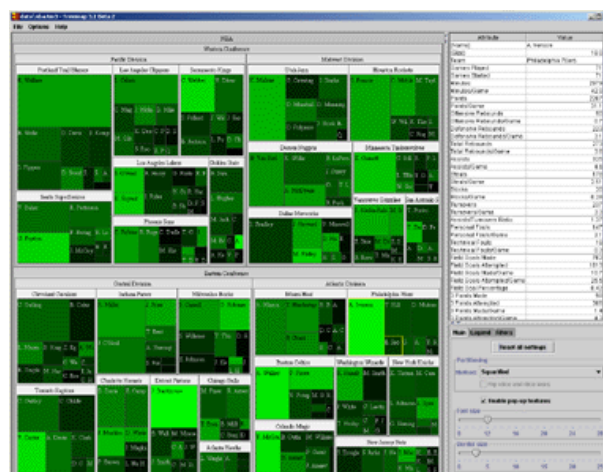
- Gephi
 - Open-source graphing software that enables you to interactively explore networks and hierarchy
- Treemap
 - A number of ways to make treemaps, but the interactive software by the University of Maryland Human-Computer Interaction Lab is the original and is free to use
 - Treemaps (developed by Ben Shneiderman in 1991) are useful for exploring hierarchical data in a small space.
 - The Hive Group also develops and maintains a commercial version for businesses

Gephi Screenshot



gephi.org

Treemap Screenshot

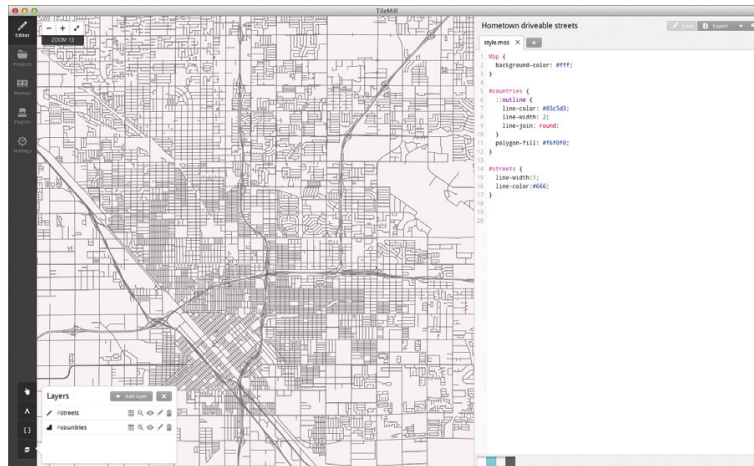


www.cs.umd.edu/hcil/treemap

Maps

- TileMill
 - TileMill, originally by mapping platform MapBox, is open source desktop software available for Windows, OS X, and Linux
 - Utilizes shapefiles, a file format that describes geospatial data, such as polygons, lines, and points
- indiemapper
 - indiemapper is a free to use online service provided by cartography group Axis Maps
 - Like TileMill, it enables you to create custom maps and map your own data, but it runs in the browser rather than as a desktop client
 - It's straightforward to use, and there are plenty of examples to help you begin.

TileMill



tilemill-project.github.io/tilemill

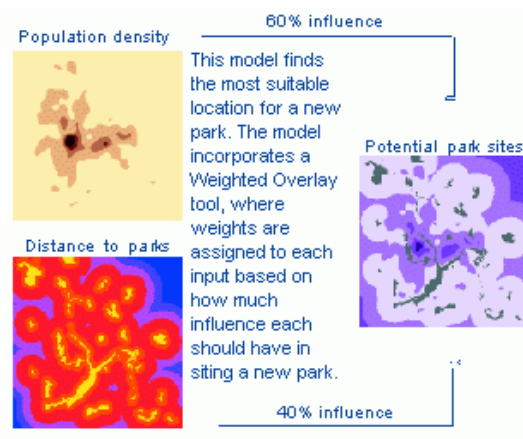
indiemapper



indiemapper.io

ArcGIS

- ArcGIS is the primary commercial mapping software
- It's a feature-rich platform that enables you to do just about anything with maps
- For most though, the basic subset of features is enough, so to avoid the hefty cost of the software, it's probably best to try the free options first, and if those aren't enough, try ArcGIS
- See more at: arcgis.com



arcgis.com

2 Programming

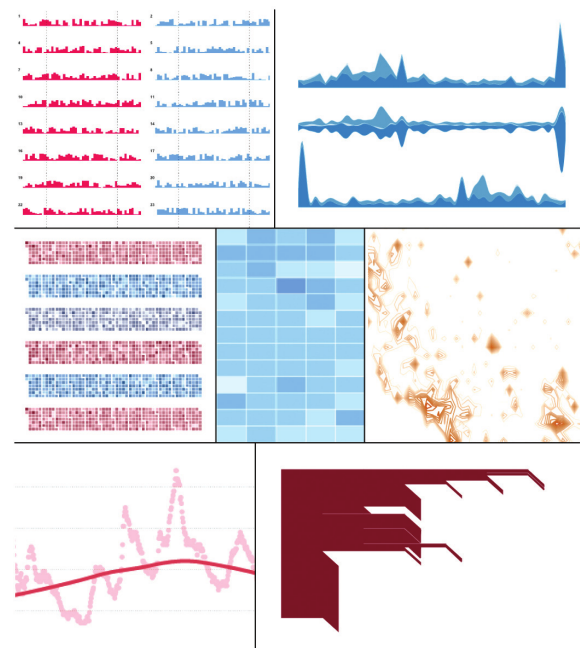
Trade-Off

- Out-of-the-box software gets you up and running in a short amount of time
- The trade-off is that you're using software that's generalized in some way so that more people can use it with their own data
- Also, if you want a new feature or method, you need to wait for someone else to implement it for you
- On the other hand, you can visualize data to your specific needs and gain flexibility when you use programming frameworks
- It's also grows easier to reproduce your work and apply it to other datasets as you build up your library and learn new things

R

- R is a language and environment for statistical computing and graphics
- It was originally used mostly by statisticians but it has expanded its audience in recent years
- There are plotting functions that enable you to make graphics with just a few lines of code, and often, one line can do the trick

- r-project.org



Source: Yau (2013)

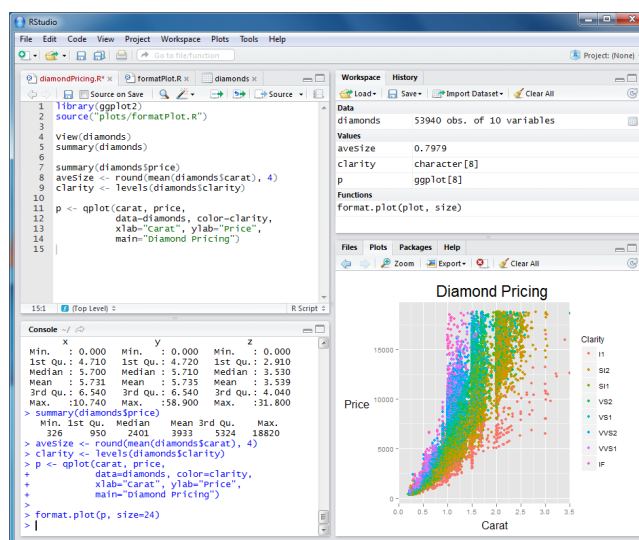
R Features

- R's is open source and many packages expand on the base distribution, which makes statistical graphics (and analysis) more straightforward, such as:
 - **ggplot2**: A plotting system based on the Leland Wilkinson's grammar of graphics, which is a framework for statistical visualization.
 - **network**: Create network graphs with nodes and edges
 - **ggmaps**: Visualization of spatial data on top of maps from Google Maps, OpenStreetMap, and others
 - * uses ggplot2
 - **animation**: Build a gallery of images and string them together for an animation
 - **portfolio**: Visualize hierarchical data with a treemap
- Just a small sample, you can view and install packages easily via the package manager place to start
- Examples: www.r-bloggers.com/7-visualizations-you-should-learn-in-r/

RStudio

- RStudio is an integrated development environment (IDE) for R
- Available as Free Software (AGPL) and as a commercial application
- Available as a Desktop Application and a Browser-accessible Server
- Integrated R help and documentation

RStudio Screenshot

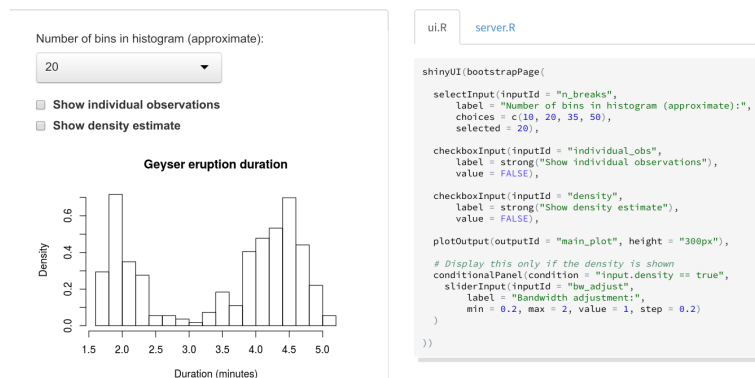


Source: programming.net/download-and-install-rstudio

Shiny

- A web application framework for R
- Turn your analyses into interactive web applications
- No HTML, CSS, or JavaScript knowledge required
- shiny.rstudio.com

Shiny Screenshot



Source: shiny.rstudio.com

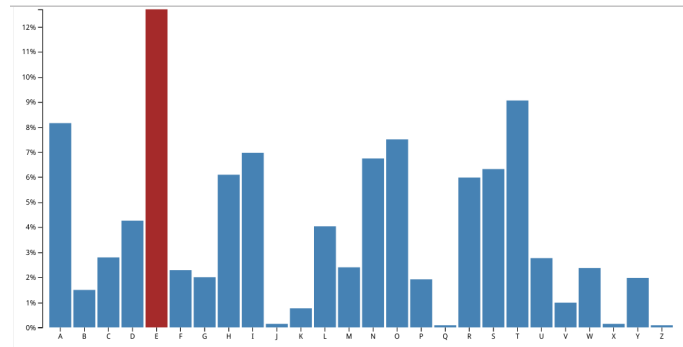
HTML5

- Not long ago, you couldn't do much visualization-wise that was native in the browser
- You had to use Flash and ActionScript
- But when Apple mobile devices didn't have Flash on them, there was a quick rush forward toward JavaScript and HTML
- The former is used to manipulate the latter, in addition to Scalable Vector Graphics (SVG)
- Cascading Style Sheets (CSS) are used to specify color, size, and other aesthetic features
- Whereas support in various browsers was inconsistent before, functionality is available now in modern browsers, such as Firefox, Safari, and Google Chrome to make interactive visualization online

HTML5 Frameworks

- Data-Driven Documents (D3)
 - One of the most, if not the most, popular JavaScript library for visualization
 - Lots of examples and a growing community
 - Advantage: powerful
 - Disadvantage: powerful
 - d3js.org
- Raphaël
 - It's not as data-centric as D3, but it's lightweight and makes drawing vector graphics in the browser straightforward
 - dmitrybaranovskiy.github.io/raphael
- InfoVis Toolkit
 - Interactive Data Visualizations
 - Includes visualization types such as bar charts, pie charts, tree maps, amongst others
 - thejit.org
- In addition, specialized libraries

D3.js Example



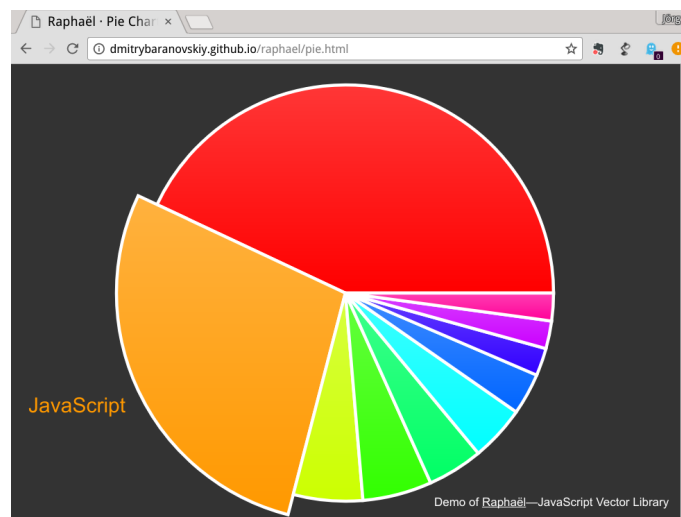
This simple bar chart is constructed from a TSV file storing the frequency of letters in the English language. The chart employs [conventional margins](#) and a number of D3 features:

[Open](#)

- [d3-dsv](#) - parse tab-separated values
- [d3-format](#) - number formatting
- [d3-scale](#) - position encodings
- [d3-array](#) - data processing
- [d3-axis](#) - axes

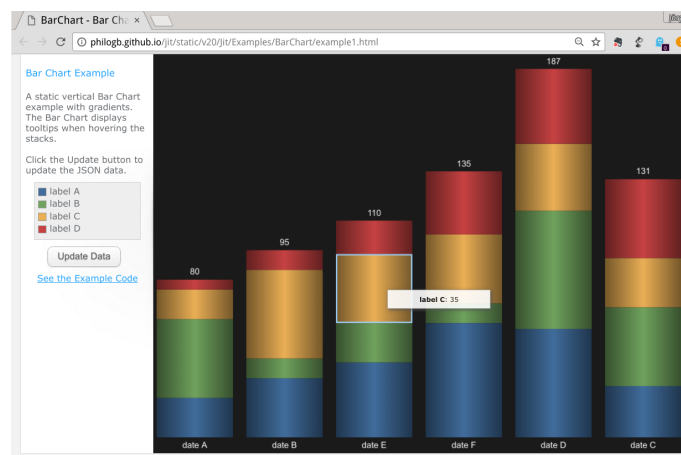
bost.ocks.org/mike/bar

Raphaël.js Example



dmitrybaranovskiy.github.io/raphael/pie.html

InfoVis Toolkit Example



philogb.github.io/jit/static/v20/Jit/Examples/BarChart/example1.html

Plotly

- Both a cloud-based visualization and business intelligence solution and a set of frameworks for visualization
- Plotly Cloud offers free accounts, but data in public repositories
- plot.ly/products/cloud
- Different graphing libraries:
 - JavaScript plot.ly/javascript
 - Python plot.ly/python
 - R plot.ly/r
 - Matlab plot.ly/matlab

Plotly JavaScript Code Example

```
y: [0.6, 0.7, 0.3, 0.6, 0.6, 0.5, 0.7, 0.9, 0.5, 0.8, 0.7, 0.2],
x: x,
name: 'radishes',
marker: {color: '#FF4136'},
type: 'box'
});

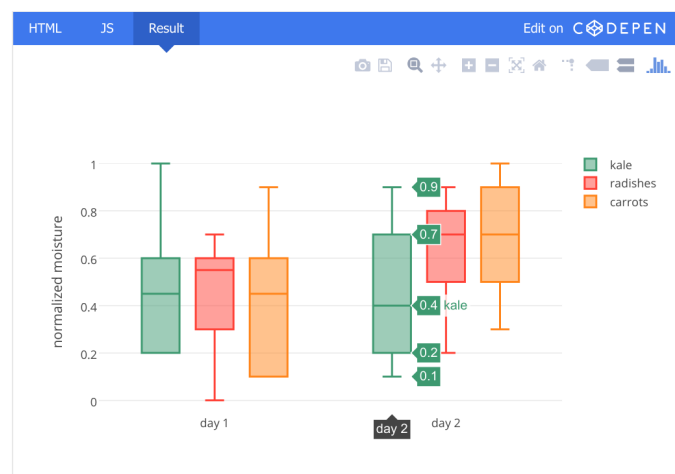
var trace3 = {
  y: [0.1, 0.3, 0.1, 0.9, 0.6, 0.6, 0.9, 1.0, 0.3, 0.6, 0.8, 0.5],
  x: x,
  name: 'carrots',
  marker: {color: '#FF851B'},
  type: 'box'
};

var data = [trace1, trace2, trace3];

var layout = {
  yaxis: {
    title: 'normalized moisture',
    zeroline: false
  },
  boxmode: 'group'
};
```

plot.ly/javascript/box-plots

Plotly JavaScript Chart Example



plot.ly/javascript/box-plots

Plotly Python Code Example

```

colorbar = dict(
    autotick = False,
    tickprefix = '$',
    title = 'GDP<br>Billions US$',
) ]

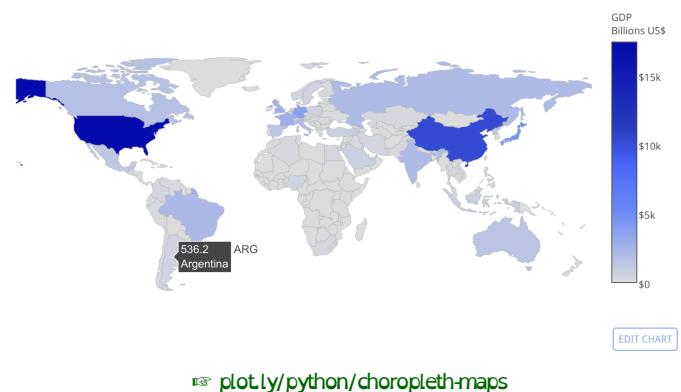
layout = dict(
    title = '2014 Global GDP<br>Source:\n'
    <a href="https://www.cia.gov/library/publications/the-world-factbook/fields/2195.htm"
    1">\n
        CIA World Factbook</a>',
    geo = dict(
        showframe = False,
        showcoastlines = False,
        projection = dict(
            type = 'Mercator'
        )
    )
)

fig = dict( data=data, layout=layout )
nu inIntf fig validate=False filename='d3_world_map' )

```

plot.ly/python/choropleth-maps

Plotly Python Chart Example



Bokeh

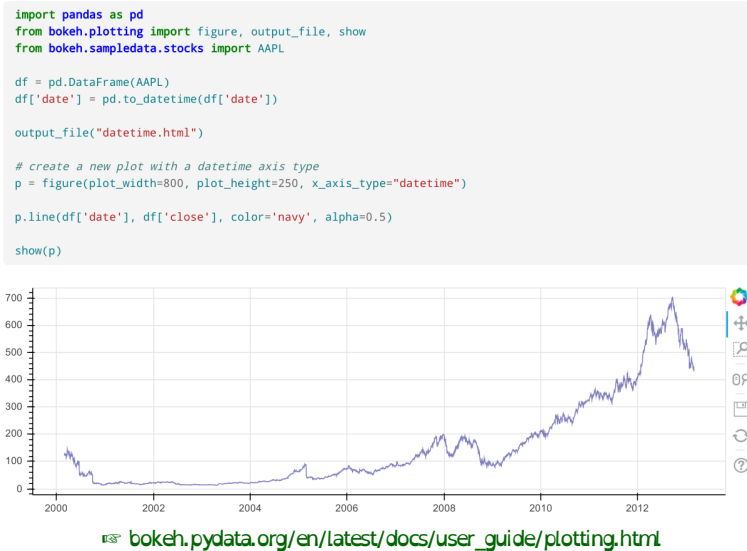
- Platform for interactive visualization and data applications in modern web browsers
- Affords concise construction of versatile graphics
- Can provide interactivity over large or streaming data sets
- Several sub-projects and bindings
 - Core library: BokehJS client library, Python bindings, and Bokeh Server
 - rBokeh: R bindings for BokehJS
 - bokeh-scala: Scala bindings for BokehJS
 - datashader: A graphics pipeline and visual query system for creating meaningful visual representations from large data sets
 - HoloViews: A high-level declarative interface to Bokeh for exploring and interacting with data
- bokeh.github.io

Bokeh Core

- Bokeh is a Python interactive visualization library
- Bokeh exposes two interface levels to users:
 - a low-level `bokeh.models` interface that provides the most flexibility to application developer
 - an higher-level `bokeh.plotting` interface centered around composing visual glyphs
- Workflow
 - Prepare some data
 - Tell Bokeh where to generate output
 - Call `figure()` to create a plot with some overall options like title, tools and axes labels
 - Add renderers for our data, with visual customizations like colors, legends and widths to the plot

- Ask Bokeh to show() or save() the results.
- Bokeh also comes with an optional server component
- bokeh.github.io

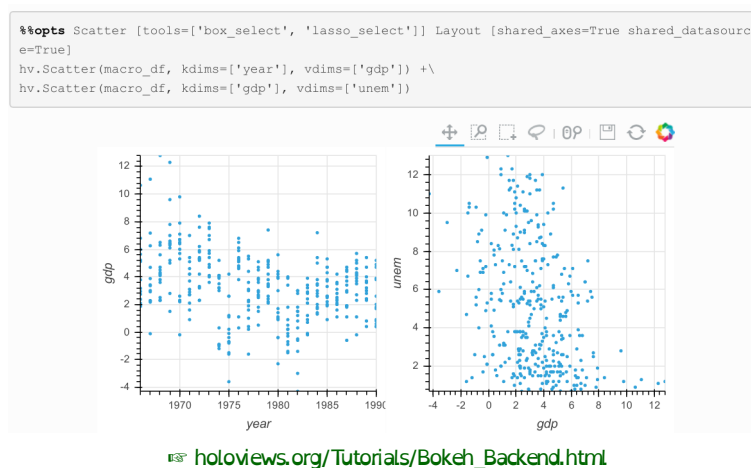
Bokeh Example





HoloViews

- Python library for analyzing and visualizing scientific or engineering data
- Instead of specifying every step for each plot, HoloViews lets you store your data in an annotated format that is instantly visualizable, with immediate access to both the numeric data and its visualization
- A HoloViews object is just a thin wrapper around your data, with the data always being accessible in its native numerical format, but with the data displaying itself automatically
- The actual rendering is done using a separate library
- All of the HoloViews objects can be used without any plotting library available, so that you can easily create, save, load, and manipulate HoloViews objects from within your own programs for later analysis
- holoviews.org


HoloViews with Bokeh Example



Notebooks

- Not strictly visualization as we have seen before, the concept of scientific Notebooks is all about integration of text, code, data and visualization
- Other examples that work with Bokeh and/or HoloViews are
 - Jupyter Notebooks
 - * born out of the IPython Project in 2014 as it evolved to support interactive data science and scientific computing across all programming languages
 - * Supporting kernels (Backends) e.g. Python, R, Julia
 - *  jupyter.org
 - Apache Zeppelin Notebooks
 - * web-based notebook that enables interactive data analytics
 - * Currently many interpreters (Backends) such as Apache Spark, Python, JDBC, Markdown and Shell
 - *  zeppelin.apache.org
- Another example is the Wolfram Computable Document Format (CDF)

Processing

- Originally designed for artists, Processing is an open source programming language that uses a sketchbook metaphor to write code
- Also a good place to start because a few lines of code can get you far, with lots of examples, libraries, books, and a large and helpful community that make Processing inviting
-  processing.org

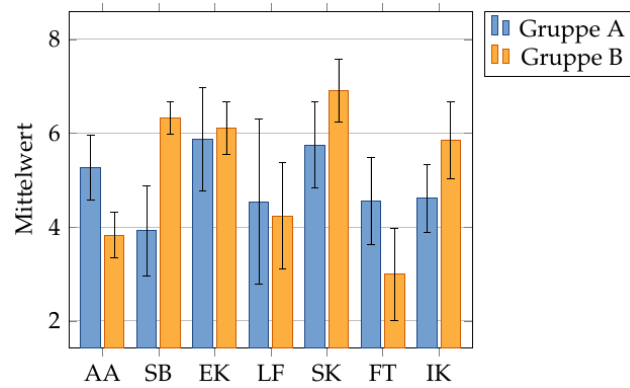
Processing Example



 processing.org/examples/piechart.html

TikZ/pgf

- TikZ and the underlying macro collection pgf is the go-to graphics framework for producing graphs in pdf-based versions of \TeX
- pgfplots draws high-quality function plots in normal or logarithmic scaling with directly in \TeX
- The user supplies axis labels, legend entries and the plot coordinates for one or more plots and pgfplots applies axis scaling, computes any logarithms and axis ticks and draws the plots
- It supports line plots, scatter plots, piecewise constant plots, bar plots, area plots, mesh- and surface plots, patch plots, contour plots, quiver plots, histogram plots, box plots, polar axes, ternary diagrams, smith charts and some more
- The pgfplotstable package reads tab-separated numerical tables and generates code for pretty-printed tables

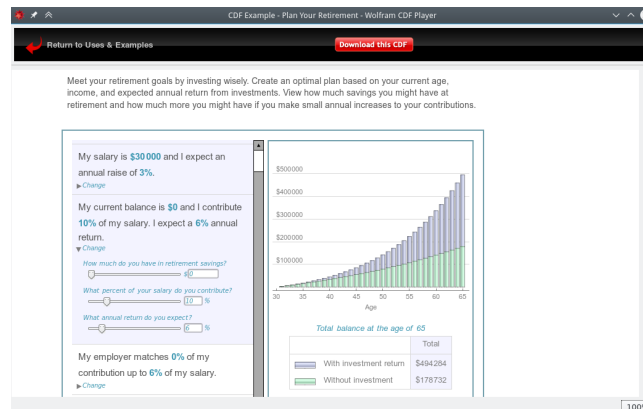


Source: mi.krwi.de/templates/thesis-template

Computer Algebra Systems

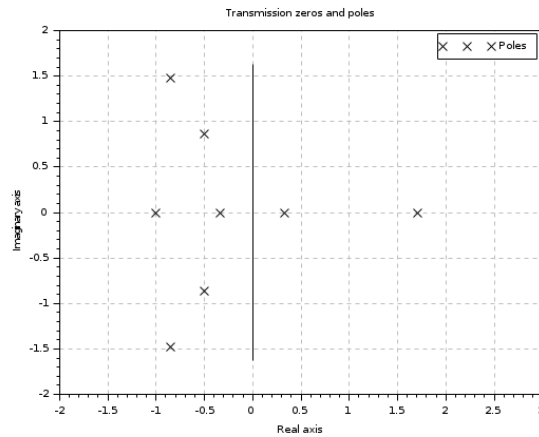
- Computer Algebra Systems (CAS) are capable of manipulating mathematical expressions symbolically
 - Mathematica www.wolfram.com/mathematica
 - * Proprietary commercial software
 - Maxima maxima.sourceforge.net
 - * Free and open source software
 - Scilab www.scilab.org
 - * Free and open source software
- In addition to a specified programming language, they often come with visualization capabilities
- The advantage is that those visualizations are tied into logical manipulations of mathematical expressions

Mathematica (CDF) Example



Source: www.wolfram.com/cdf/uses-examples/investment-statements.html

Scilab Example



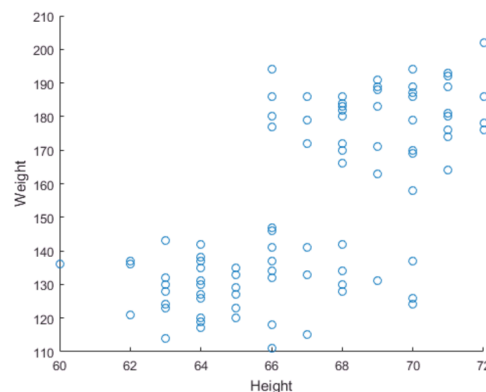
Source: help.scilab.org/docs/6.0.0/en_US/Graphics.html

Numerical Computing Environments

- In contrast to CAS, numerical computing environments focus on solving mathematical problems numerically instead of symbolically
 - Matlab www.mathworks.com/products/matlab.html
 - * Proprietary commercial software
 - Octave gnu.org/software/octave
 - * Free and open source software
- In addition to a specified programming language, they often come with visualization capabilities
- The advantage is that those visualizations are tied into the powerful numerical solvers

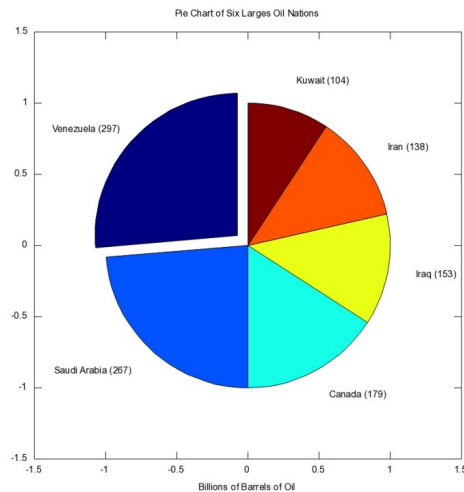
Matlab Example

```
load patients Height Weight Systolic % load data
scatter(Height,Weight) % scatter plot of Weight vs. Height
xlabel('Height')
ylabel('Weight')
```



Source: www.mathworks.com/help/matlab/examples/creating-2-d-plots.html

Octave Example



Source: mathblog.com/plotting-and-graphics-in-octave
Can make use of gnuplot

3 Tutorial

Assignment 6.1: Experiences

- Basically, this is a short overview including the systems that have been discussed last week
- Do you have any experience using any of the systems and frameworks outlined?
 - What kind of experience?
 - Advantages and disadvantages
- Did I forget anything?
- With the overview given, do you have preferences for looking into these systems?

Assignment 6.2: Homework

- Try out some tools
 - Either personal liking. . .
 - . . . or what is most interesting for the course
- Using data sets
 - The small training data sets given
 - Data sets you have
 - Other data sets that are freely available
- Share your experience with us in two weeks

References

Literatur

- Kirk, A. (2012). *Data Visualization – A Successful Design Process*. PACKT Publishing, Birmingham.
- Yau, N. (2013). *Data Points – Visualization that means something*. Wiley.