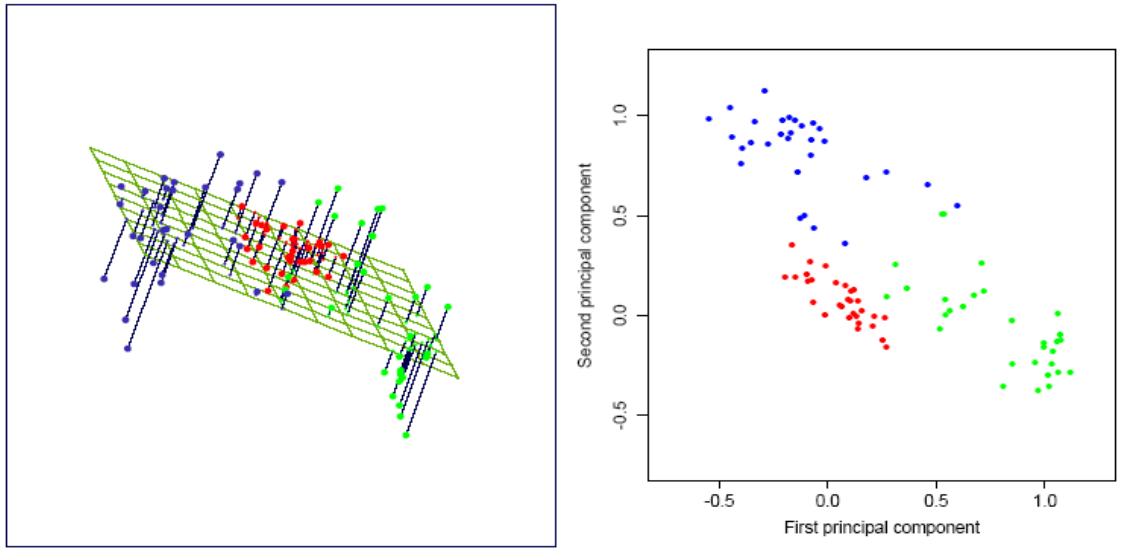
Visualization for Data Science

Visualization

- Of data; process; results
- Motivation
  - For data-driven hypotheses human interaction is necessary
    - Humans can quickly analyze complex systems
    - Humans are good at pattern recognition
    - Humans are flexible
  - Exploratory Data Analysis
  - And communication! http://www.gapminder.org/

PCA often used to Visualize/Explore High D data

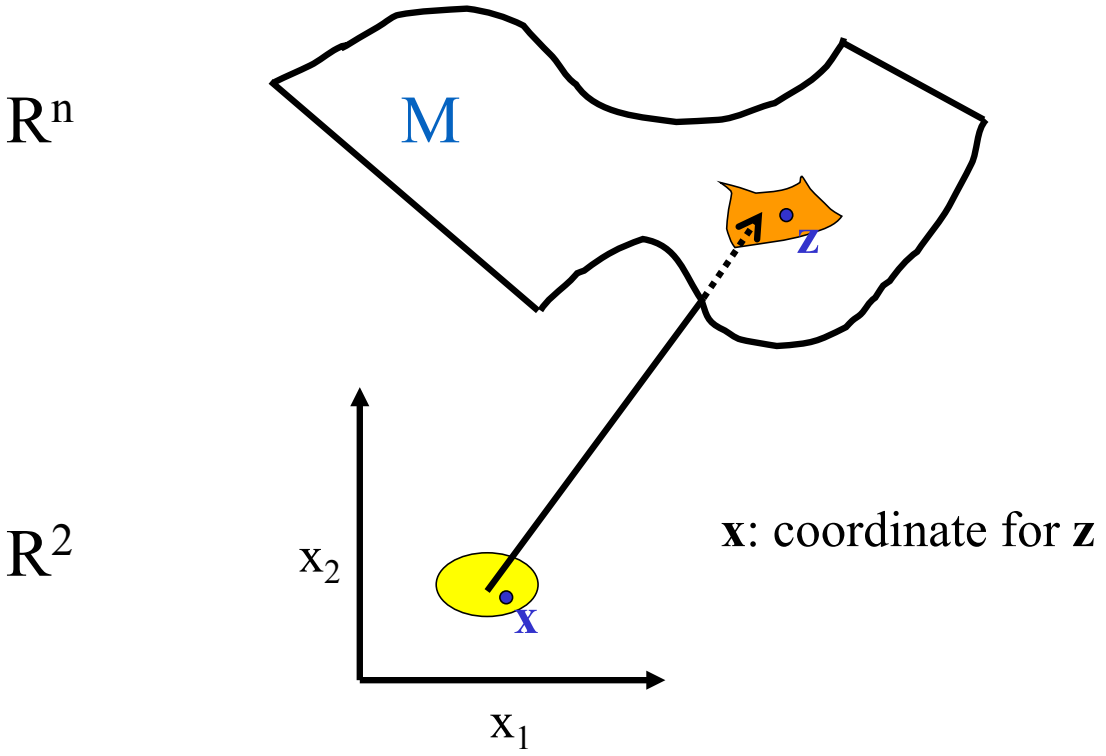
- 

- "Half-Sphere Example, HTF Fig 14.21
- Often one projects along multiple pairs of Principal Components.

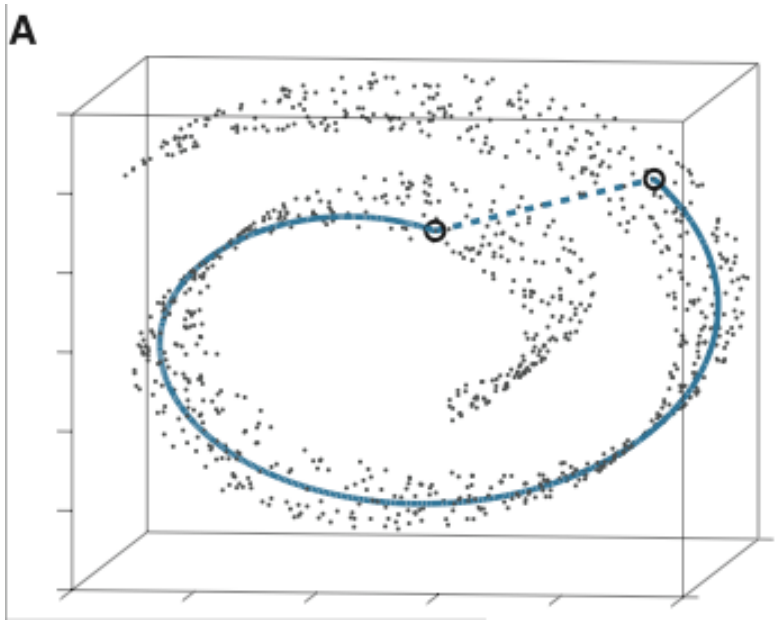
Manifold and Dimensionality Reduction

- Manifold: generalized "subspace" in R^n ($n >> 1$)
- Points in a *local* region on a manifold can be indexed by a subset of R^k
  - The value of k is usually small
  - Thus map n-dim space into local k-dim coordinates.
  - Neural approaches include SOM and GTM

Example of a Manifold



Example: Manifold in Swiss Roll



T-SNE

- https://distill.pub/2016/misread-tsne/

Modern Web-Based Visualization

- Interactive, often Javascript based
- http://d3js.org/
- **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.
- Gallery at: https://github.com/mbostock/d3/wiki/Gallery
- Webinar and ebook: http://it-ebooks.info/book/1265/
  - **D3** allows you to bind arbitrary data to a Document Object Model (DOM), and then apply data-driven transformations to the document. For example, you can use D3 to generate an HTML table from an array of numbers. Or, use the same data to create an interactive *Scalable Vector Graphics* (*SVG*) bar chart with smooth transitions and interaction.
- http://nvd3.org/
  - Simpler than D3.js

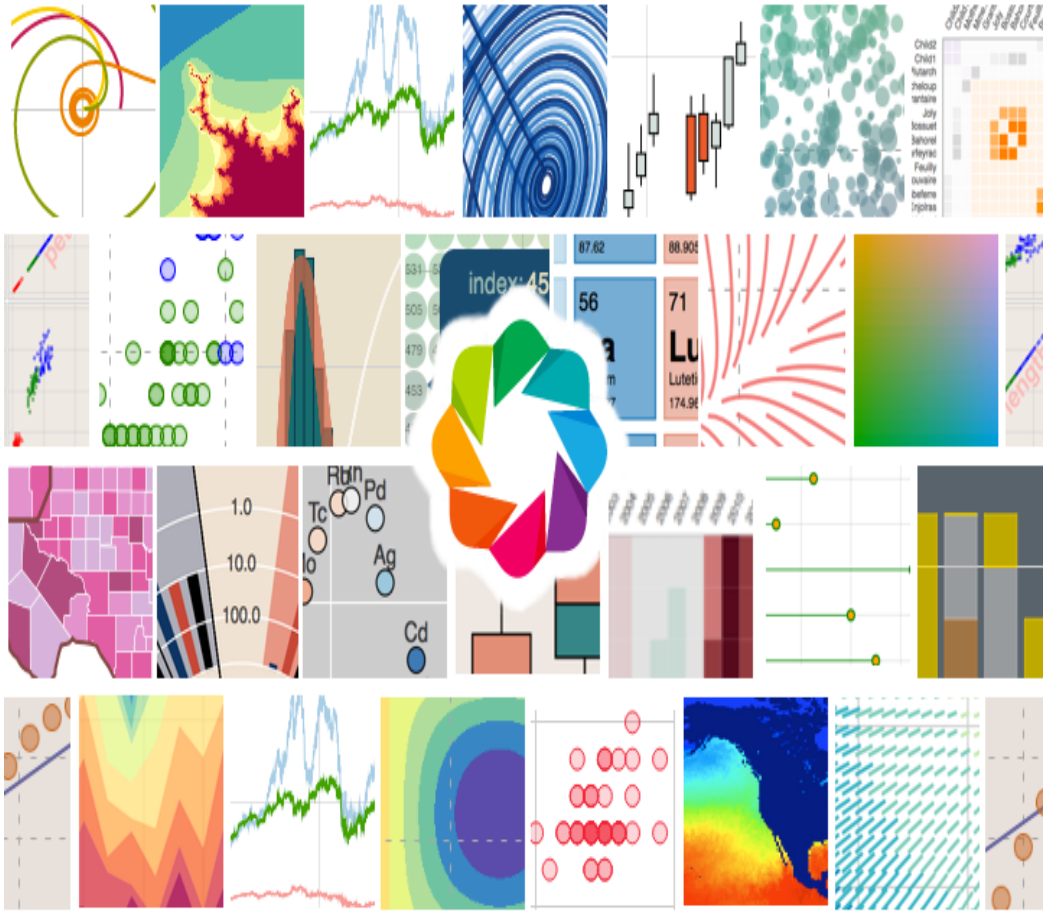
R/Python interactive visualizations

- Ggplot2 in R / matplotlib in python are **static**
- Dynamic (Python): http://bokeh.pydata.org/en/latest/
  - Gallery shows source code
- Dynamic (R): Shiny from Rstudio
- http://shiny.rstudio.com/
  - Again gallery shows code: (server.R, ui.R)

- Also see
  - Google charts https://developers.google.com/chart/?hl=en
  - Google bubble chart is similar to Gapminder video: https://www.youtube.com/watch?v=jbkSRLYSojo
  - http://setosa.io/ (e.g Simpson's paradox, PCA visuals etc)
  - http://www.highcharts.com/

Bokeh

- **Beautiful Python Visualizations: An Interview with Bryan Van de Ven, Bokeh Core Developer**
  - http://www.kdnuggets.com/2017/08/interview-bryan-van-de-ven-bokeh.html
- Resources pointed to:
- GitHub: https://github.com/Bokeh/bokeh
  Documentation: http://bokeh.pydata.org/en/latest
  Example Apps: https://demo.bokehplots.com
  Tutorials: http://nbviewer.jupyter.org/github/bokeh/bokeh-notebooks/blob/master/index.ipynb
  Mailing List: https://groups.google.com/a/continuum.io/forum/#!forum/bokeh
  Gitter Chat: https://gitter.im/bokeh/bokeh



Visualizing Very Large Data Sets

- NY Taxi cab data:

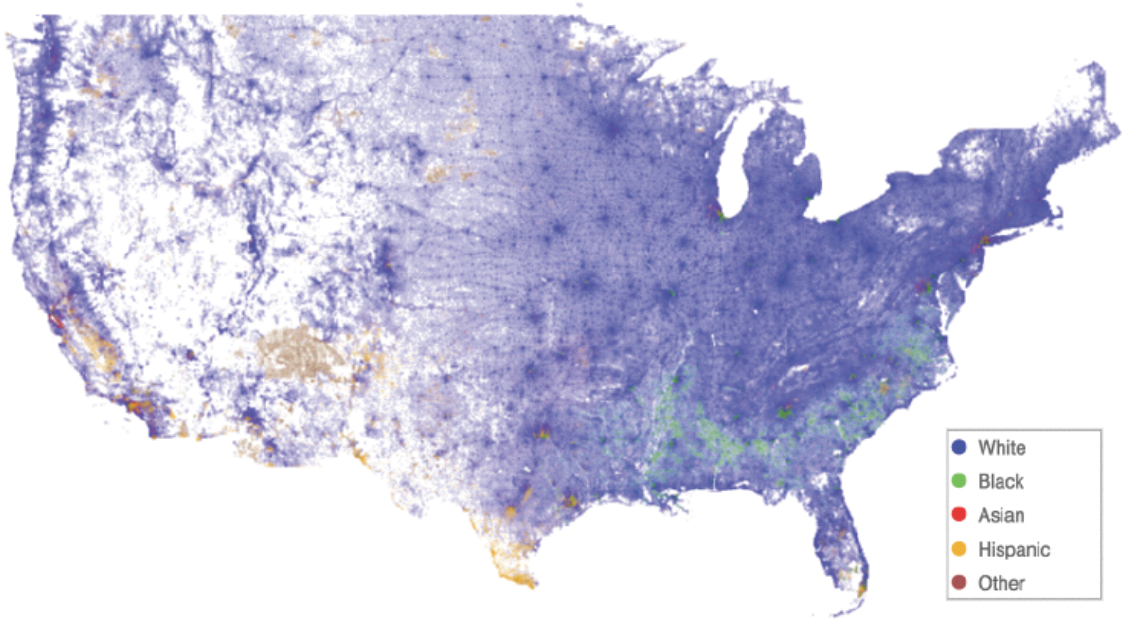
Because Datashader decouples the data-processing from the visualization, it can handle arbitrarily large data

E.g. Open Street Map data:

- About 3 billion GPS coordinates
- https://blog.openstreetmap.org/2012/04/01/bulk-gps-point-data/.
- This image was rendered in one minute on a standard MacBook with 16 GB RAM
- Renders in 7 seconds on a 128GB Amazon EC2 instance

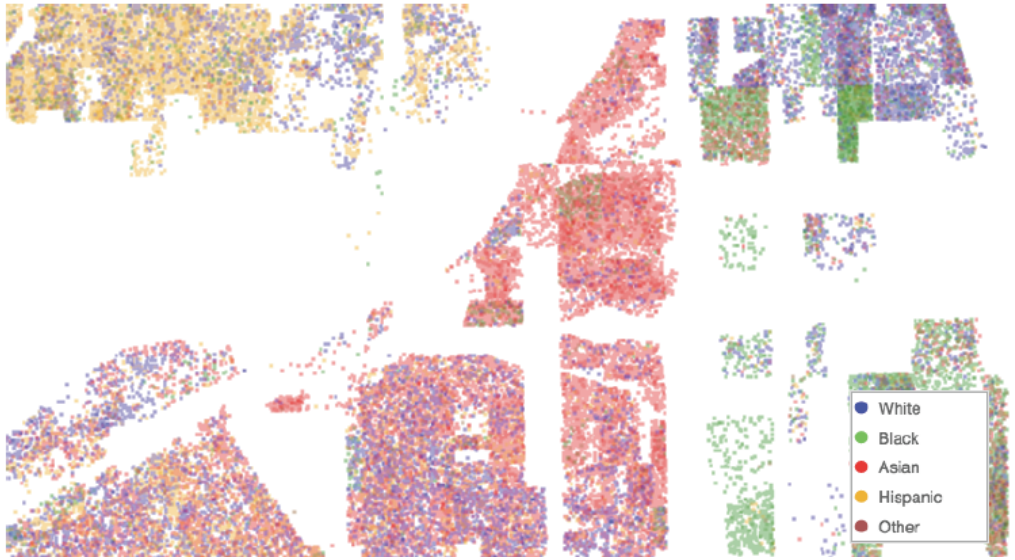
Courtesy: Jim Bednar, Continuum. Also see http://www.slideshare.net/continuumio/visualizing-a-billion-points-w-bokeh-datashader

Categorical data: 2010 US Census



- One point per person
- 300 million total
- Categorized by race
- Datashading shows faithful distribution per pixel

Categorical data: Chinatown Census



- At neighborhood level, the full racial distribution is clear
- Size of dots increases automatically for visibility