Data visualization

Yosuke Tanigawa
2018/2/1 BIODS 215

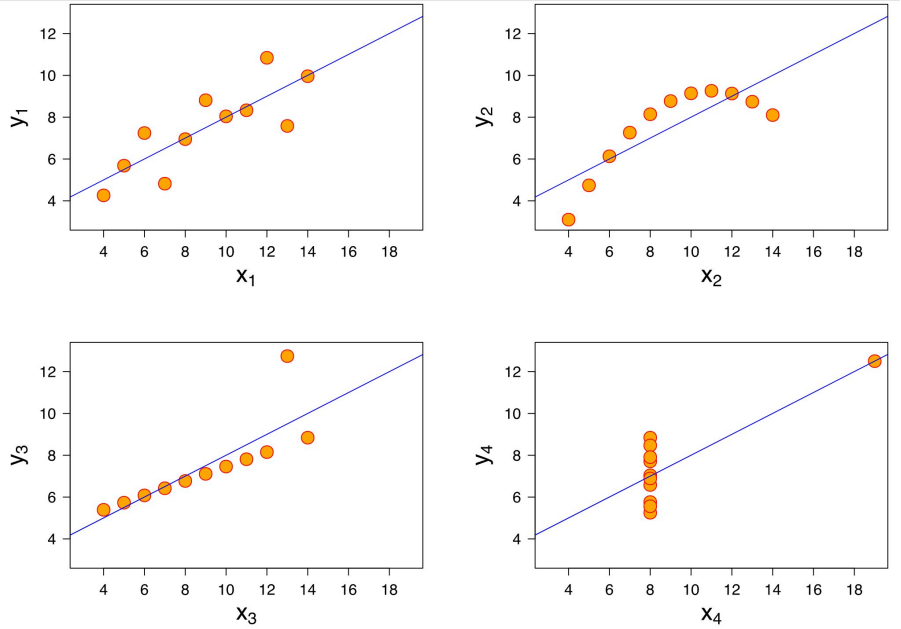
Announcement

- Homework 1:
  - due 2/6/2018 (Tue.)

- Class project:
  - Milestone due 2/20/2018 (Tue.)
  - please start working on the data

Why we'd like to visualize the data ??

Anscombe's quartet ---- Can we understand the data from descriptive statistics?

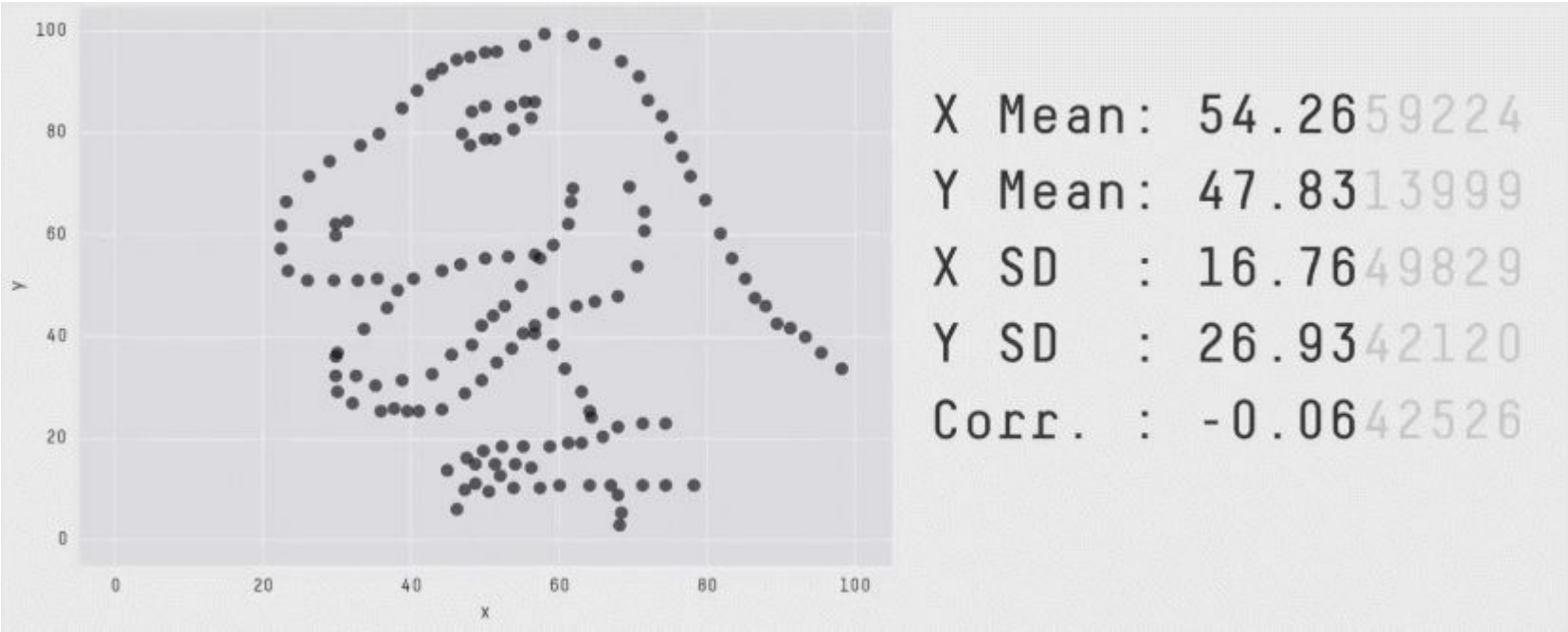


Data [edit]

For all four datasets:

Property	Value	Accuracy
Mean of *x*	9	exact
Sample variance of *x*	11	exact
Mean of *y*	7.50	to 2 decimal places
Sample variance of *y*	4.125	plus/minus 0.003
Correlation between *x* and *y*	0.816	to 3 decimal places
Linear regression line	$y = 3.00 + 0.500x$	to 2 and 3 decimal places, respectively
Coefficient of determination of the linear regression	0.67	to 2 decimal places

datasauRus



J. Matejka & G. Fitzmaurice. “Same Stats, Different Graphs: Generating Datasets with Varied Appearance and Identical Statistics through Simulated Annealing”. 2017.

Today’s topics

- Types of plots and relevant topics
- How to cope with high-dimensional data?
- Interactive plotting
- Resource: useful software/packages

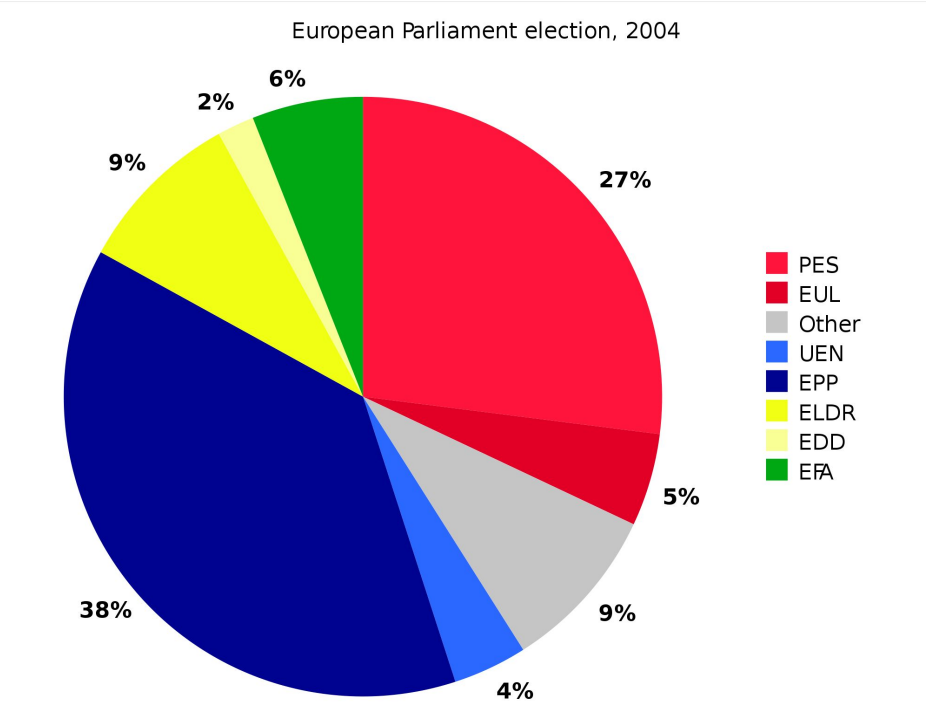
Today’s topics

- Types of plots and relevant topics
- How to cope with high-dimensional data?
- Interactive plotting
- Resource: useful software/packages

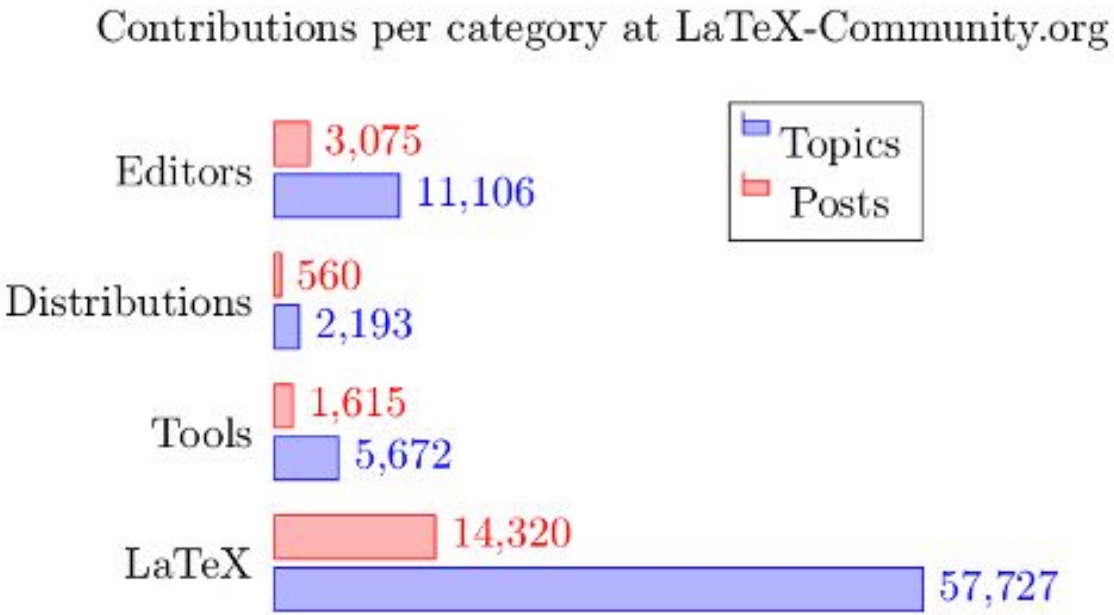
Types of plots and relevant topics

- Pie chart
- Bar plot
- Line plot
  - Transformation of scales (log, etc.)
- Histogram
  - Density estimation
- Box plot, Violin plot
- Scatter plot
  - Correlation (Pearson’s, Spearman’s)
  - Mutual information
- Heatmap, Dendrogram
  - Hierarchical Clustering
- Circos plot
  - Pairwise Interactions
- Network/Graph
  - Trees and Forests
  - Directed Acyclic Graphs
- Specialized data browser
  - Maps
  - Genome browser

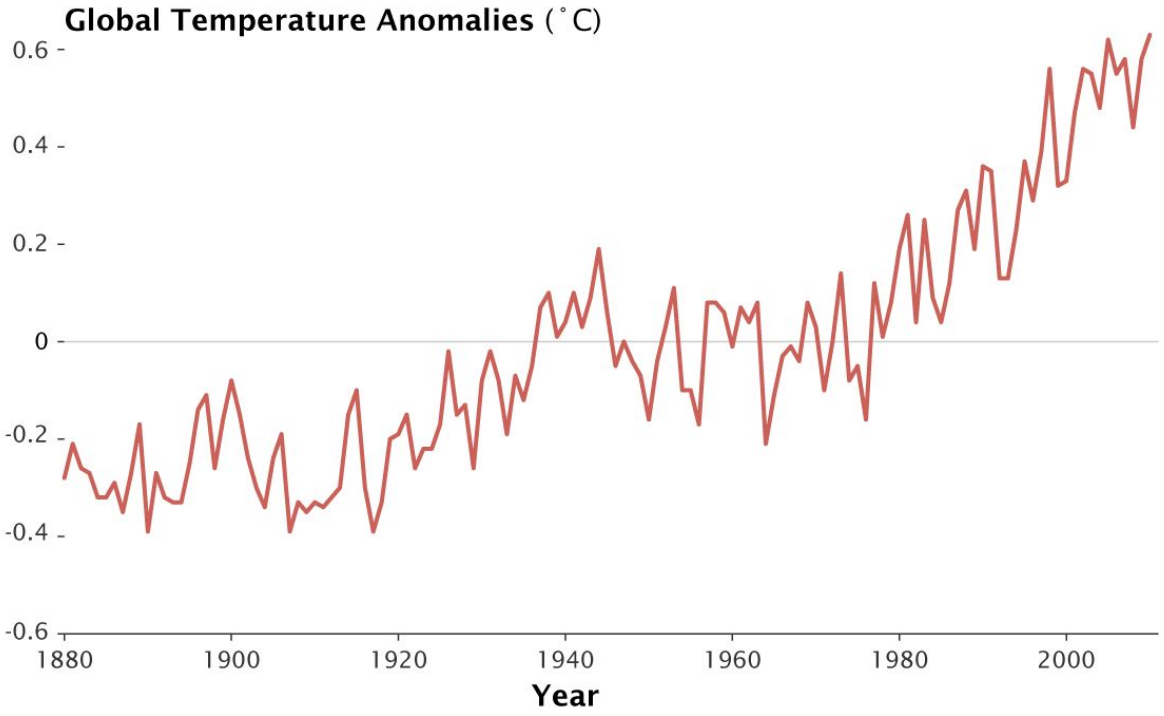
Pie chart: fractions



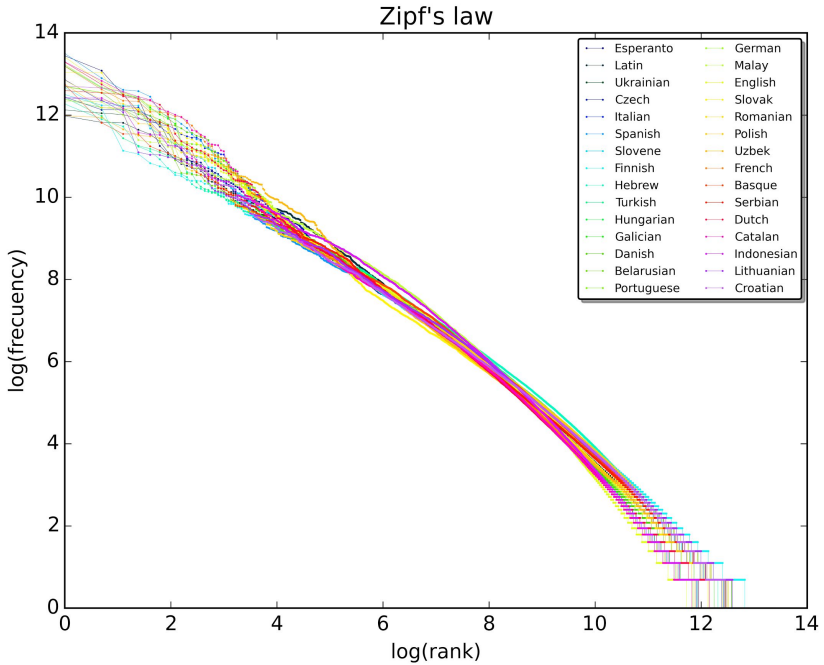
Bar plot: graphical comparison of quantities



Line plot: best for series data (e.x. time series)



Transformation of axis (semi-log, log-log plot, etc.)

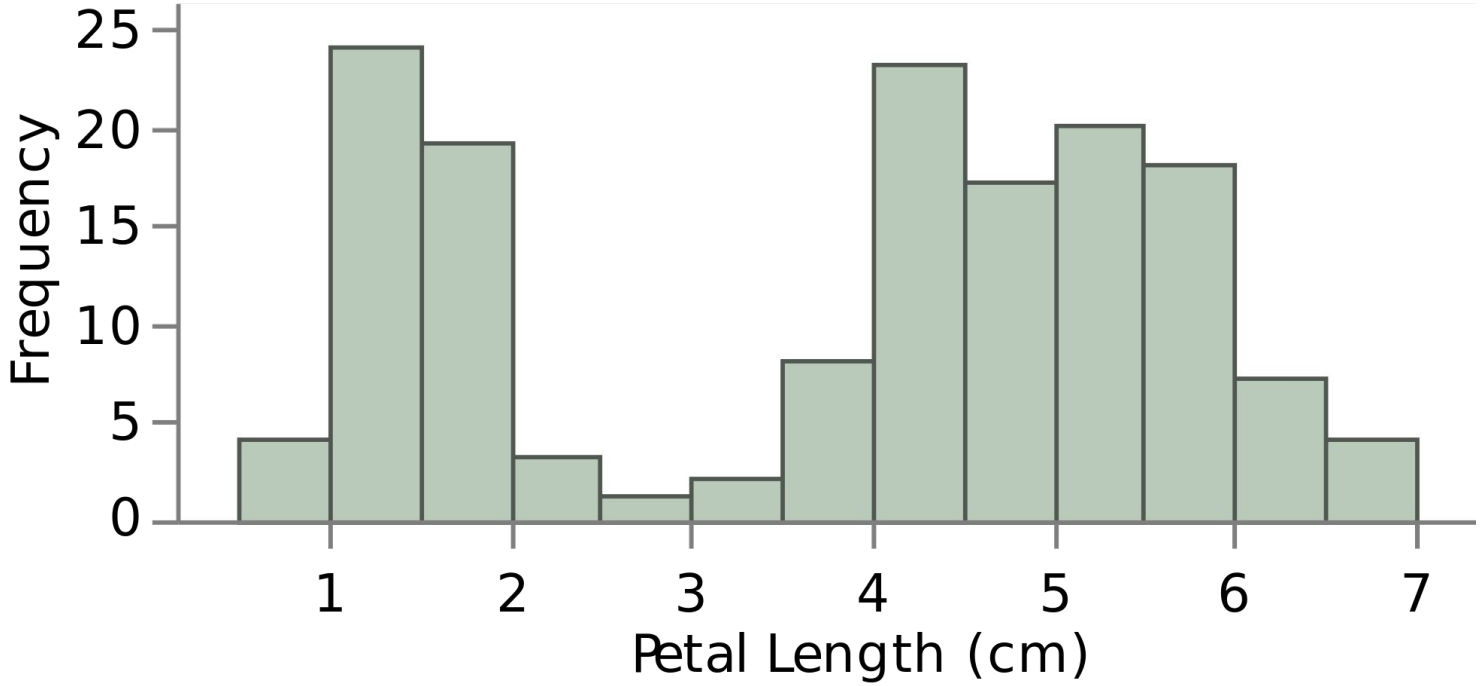


Semi-log: $y = A \exp(x)$
-> $\log(y) = x + \log(A)$

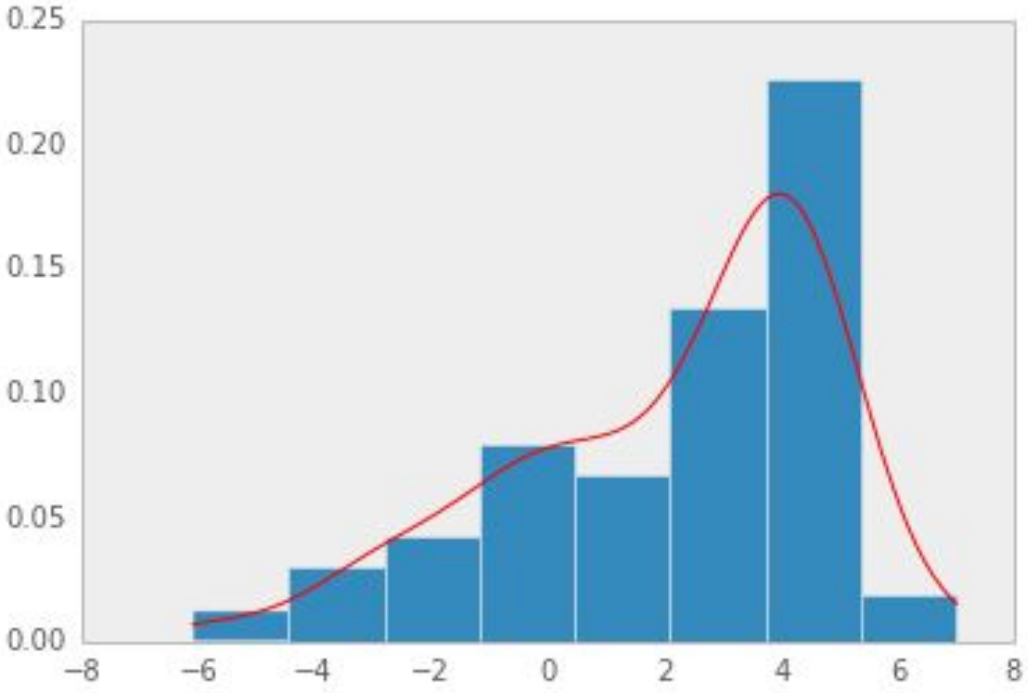
Log-log: $\exp(y) = A \exp(x)$
-> $y = x + \log(A)$

Goodness of fit:
Rank vs. quantile plot

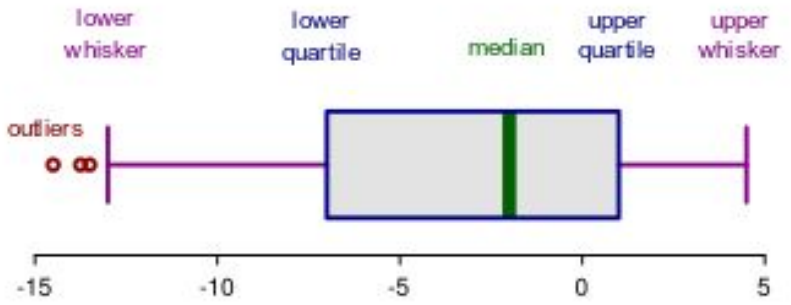
Histogram: distribution of 1D data

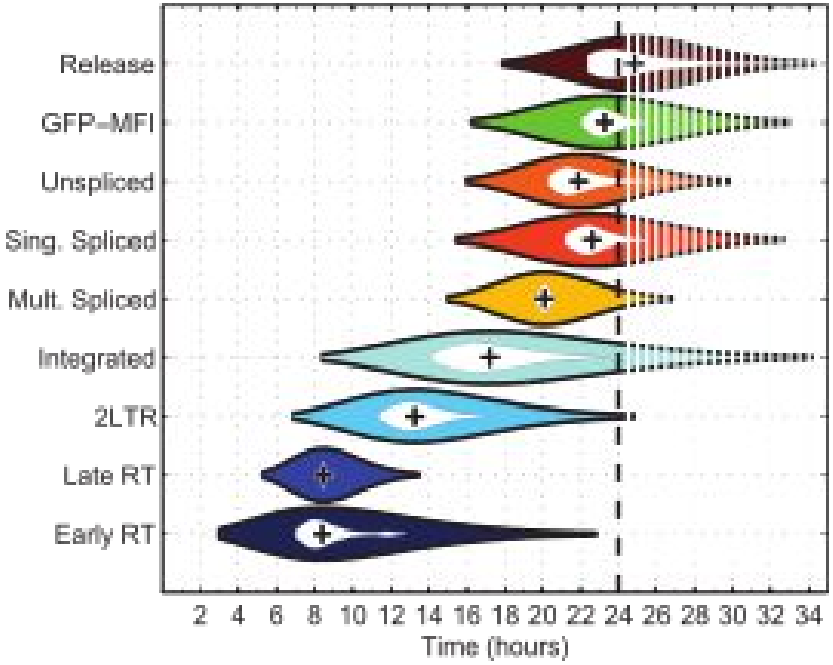


Density estimation: infer the population distribution



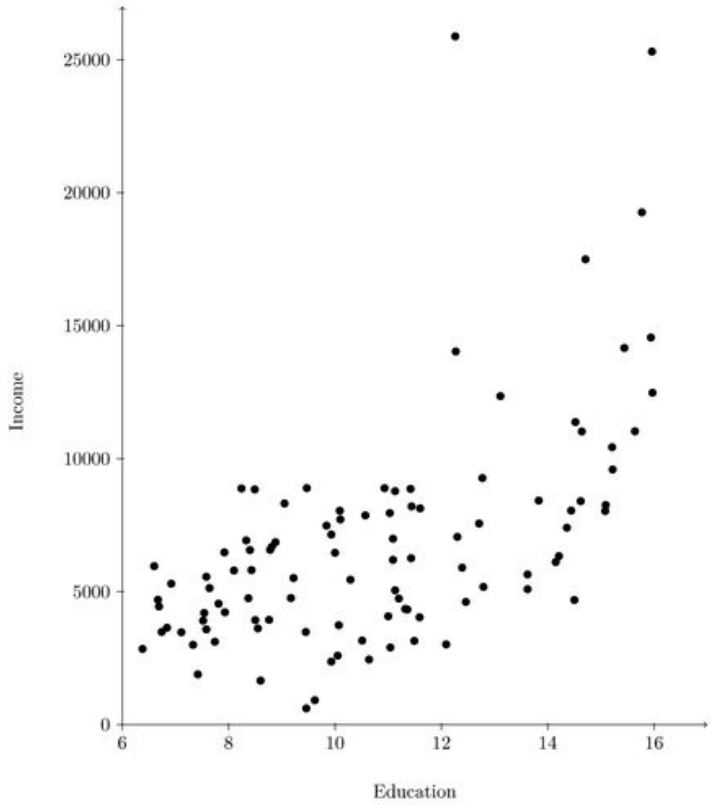
Box plot, Violin plot: distribution of 1D data





Useful for comparisons of multiple conditions

Scatter plot: distribution of 2D data



Outlier?

Correlation between two variables?

Can we fit a liner model?

Correlation

Pearson's correlation

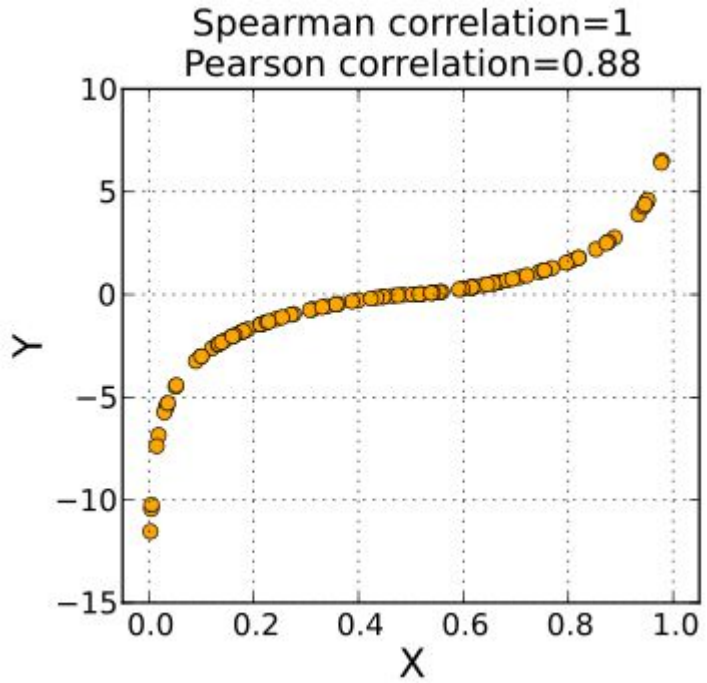
$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y}$$

where:

- cov is the covariance
- σ_X is the standard deviation of X
- σ_Y is the standard deviation of Y

Spearman's correlation (non-parametric)

→ Pearson's correlation of ranks



Mutual information (Easy to compute for discrete distributions)

Mutual Information

Definition

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\,p(y)} \right),$$

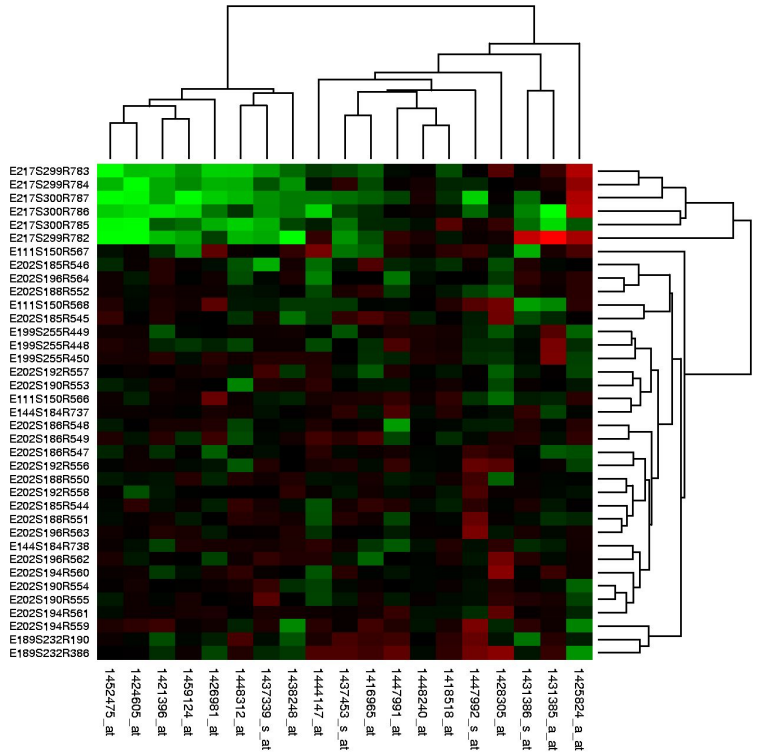
Property *I(X; Y)* >= 0

$$\begin{aligned} I(X;Y) &\equiv \mathrm{H}(X) - \mathrm{H}(X|Y) \\ &\equiv \mathrm{H}(Y) - \mathrm{H}(Y|X) \\ &\equiv \mathrm{H}(X) + \mathrm{H}(Y) - \mathrm{H}(X,Y) \\ &\equiv \mathrm{H}(X,Y) - \mathrm{H}(X|Y) - \mathrm{H}(Y|X) \end{aligned}$$

H(.) is entropy

Heatmap: a representation of matrix data

- Commonly used for
  - correlation matrix
  - Microarray gene expression data
  - Visualization of interaction of two variables
- Each cell represents intensity of the value
- Recommended to use blue-yellow color gradient (color blindness)

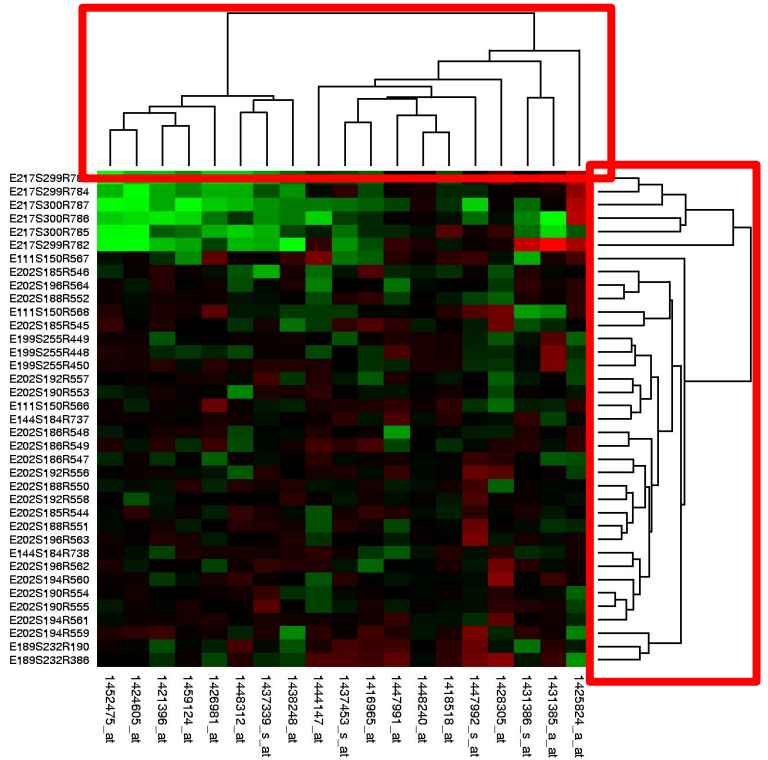


Hierarchical clustering and dendrogram

Hierarchical clustering

1. Treat each data point as one cluster
2. Compute the pairwise distance between all clusters
3. Merge two clusters with the smallest distance
4. Go to step 2 and repeat

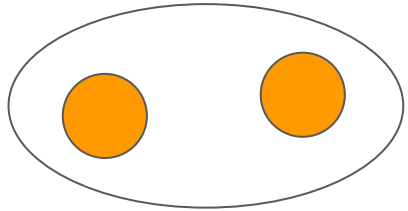
Dendrogram is a binary tree that represents merge operation



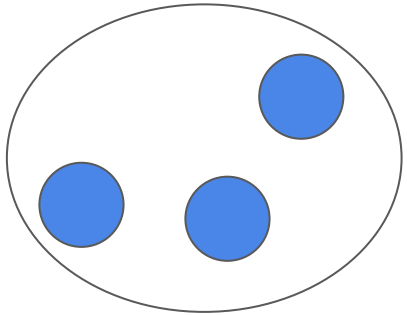
Distance between clusters

How do we define distance between clusters?

Let’s say there are two clusters A = {a_i} and B = {b_j}

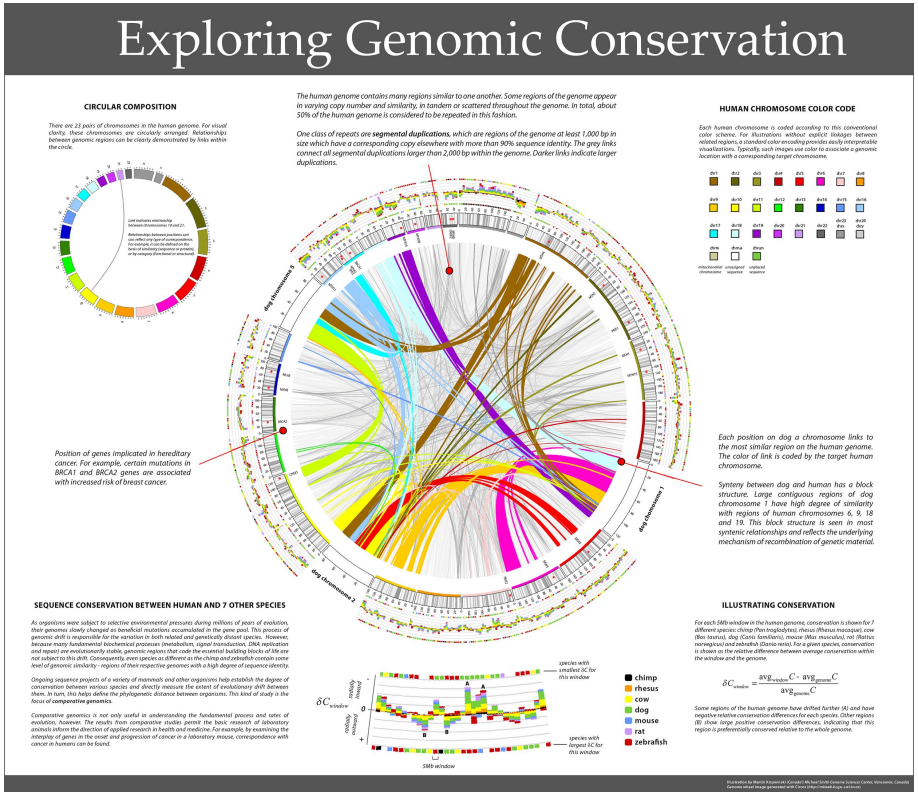


Method	Definition of d(A, B)
Single linkage	min_{i, j} d(a_i, b_j)
Complete linkage	max_{i, j} d(a_i, b_j)
UPGMA*1	E_{i, j} d(a_i, b_j)



*1: UPGMA (Unweighted Pair Group Method with Arithmetic Mean)

Circos plot: visualization of pairwise interaction

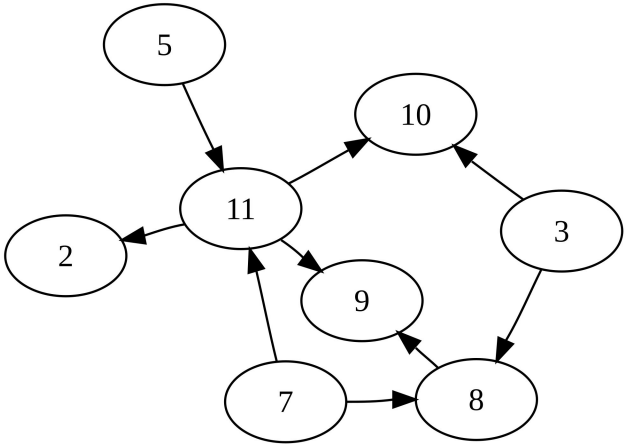


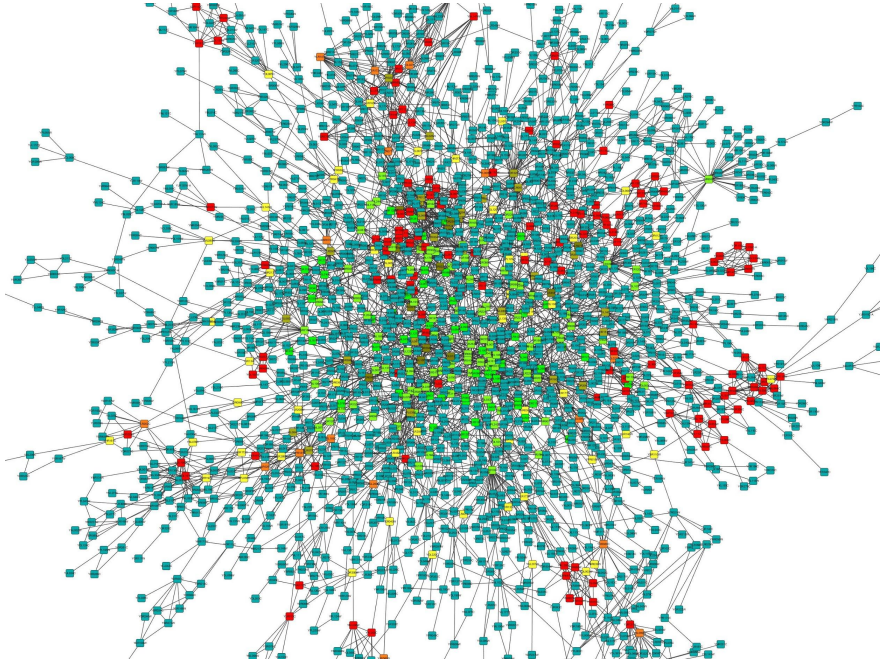
Visualization of sparse pairwise interaction

Once can add bar plot, etc.

http://circos.ca/guide/genomic/img/circos-conservation.png

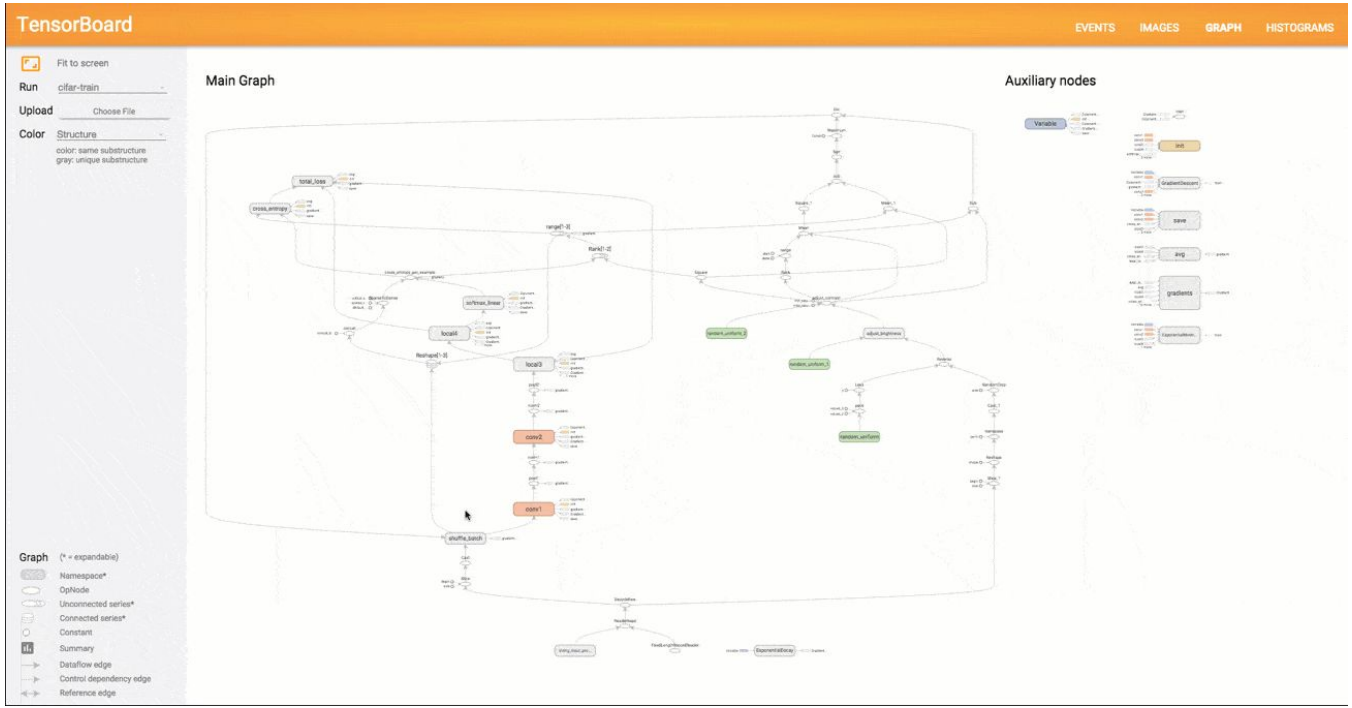
Trees, forest, DAGs, networks



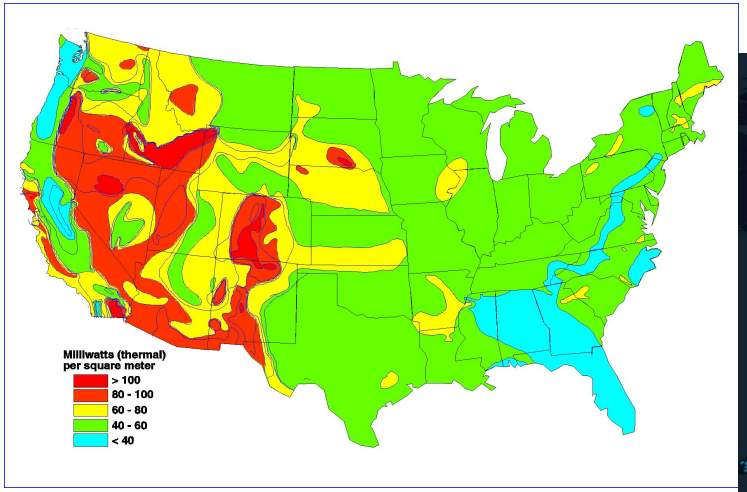


Represent node and edges

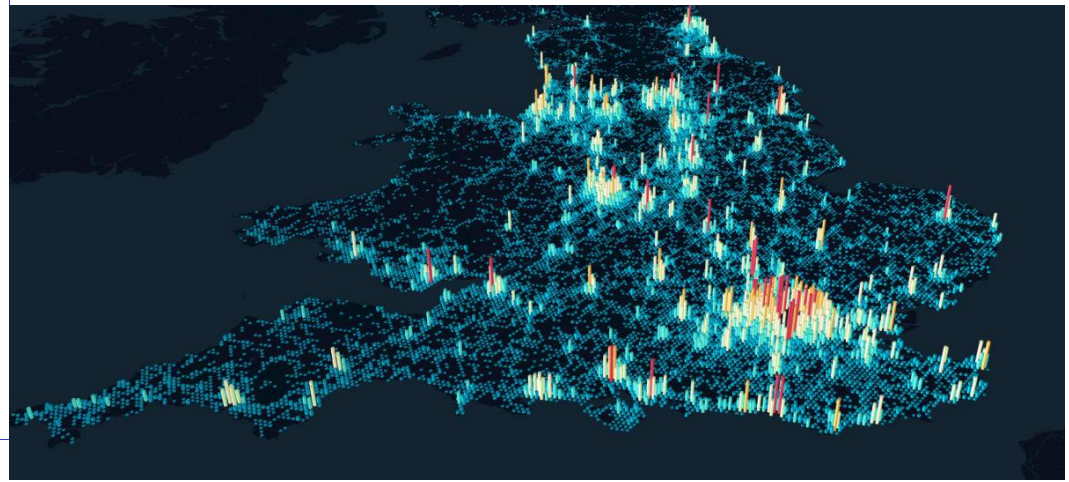
Tensorboard: visualization of deep learning model



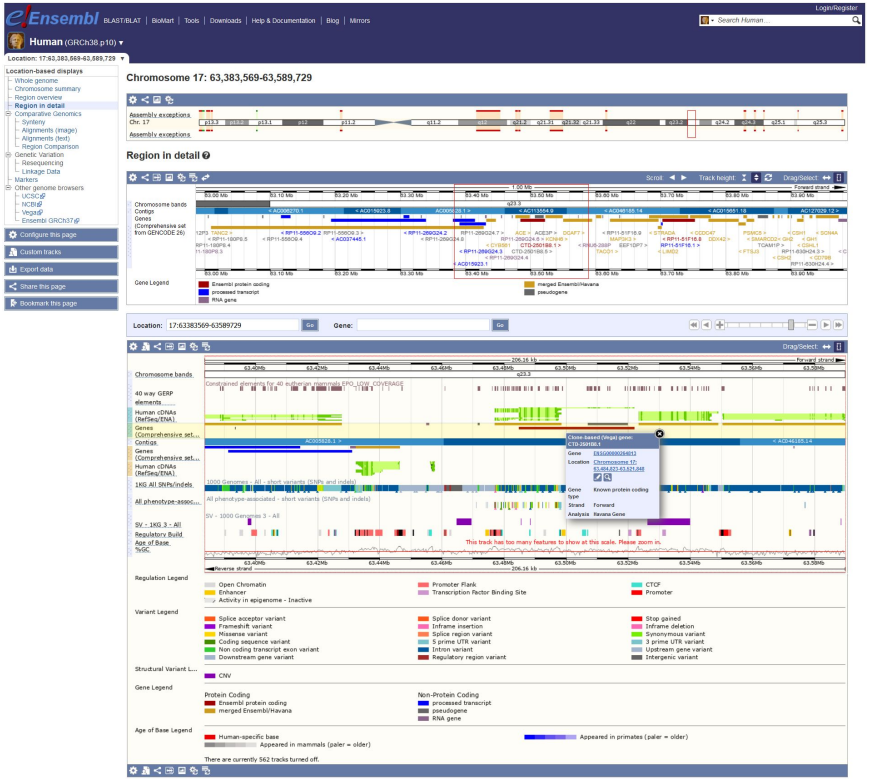
Specialized plot -- Map



DECK.GL (by Uber)



Specialized plot -- Genome browser



Types of plots and relevant topics

- Pie chart
- Bar plot
- Line plot
  - Transformation of scales (log, etc.)
- Histogram
  - Density estimation
- Box plot, Violin plot
- Scatter plot
  - Correlation (Pearson’s, Spearman’s)
  - Mutual information
- Heatmap, Dendrogram
  - Hierarchical Clustering
- Circos plot
  - Pairwise Interactions
- Network/Graph
  - Trees and Forests
  - Directed Acyclic Graphs
- Specialized data browser
  - Maps
  - Genome browser

Today’s topics

- Types of plots and relevant topics
- How to cope with high-dimensional data?
- Interactive plotting
- Resource: useful software/packages

How to cope with high-dimensional data?

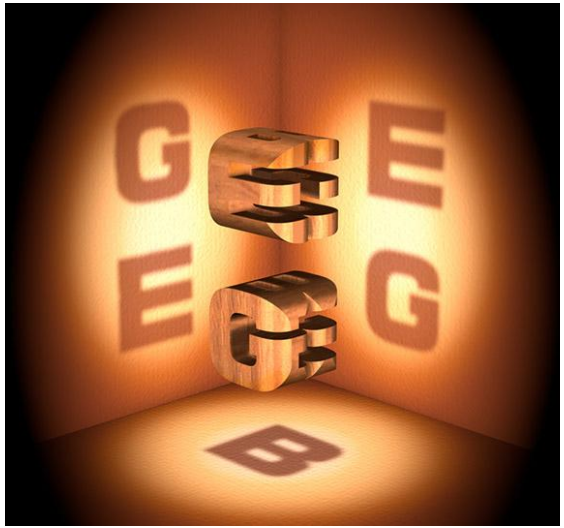
- Marginal distributions
  - Corner plot
- Dimension reduction
  - Principal component analysis (PCA)
  - t-distributed stochastic neighbor embedding (t-SNE)
  - Multidimensional scaling (MDS)
- Clustering
  - Distance/Similarity metric
  - Hierarchical clustering
  - K-means
  - Kernel methods

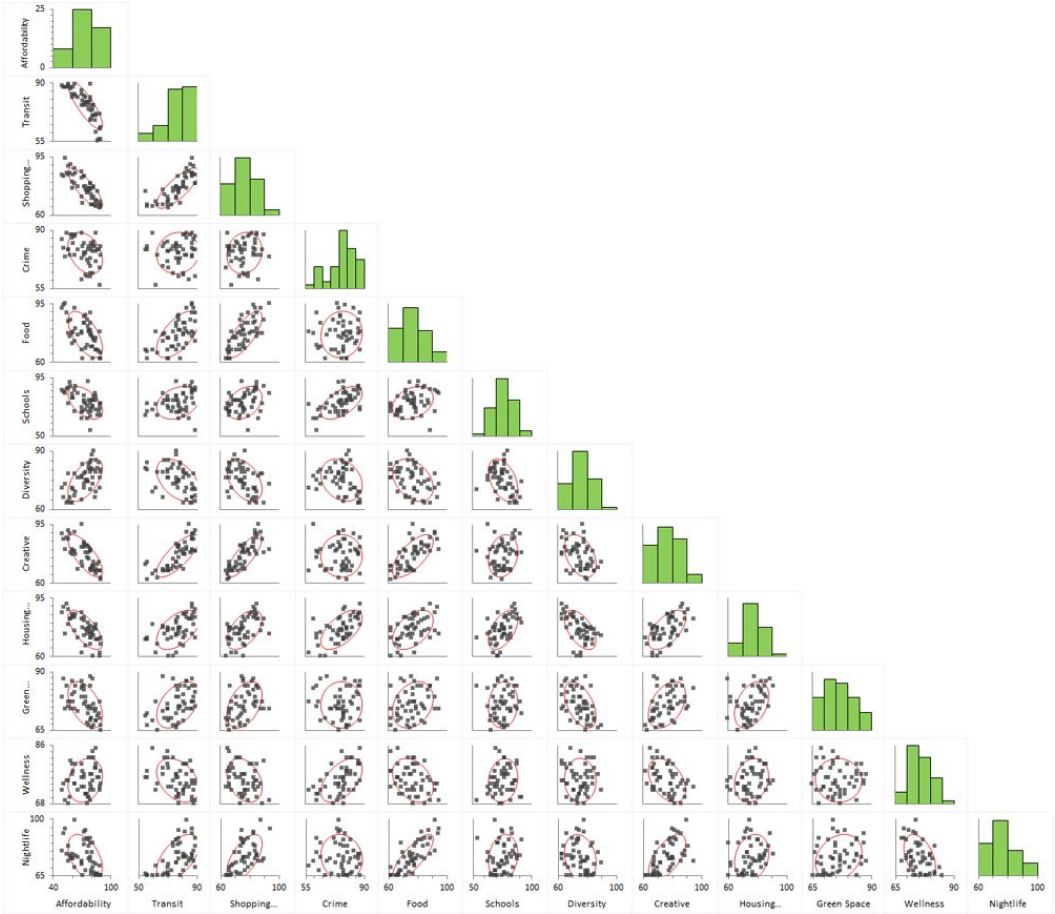
How to cope with high-dimensional data?

- Marginal distributions
  - Corner plot
- Dimension reduction
  - Principal component analysis (PCA)
  - t-distributed stochastic neighbor embedding (t-SNE)
  - Multidimensional scaling (MDS)
- Clustering
  - Distance/Similarity metric
  - Hierarchical clustering
  - K-means
  - Kernel methods

Marginal distribution

- Marginal distribution
- Corner plot





Douglas Hofstadter. Gödel, Escher, Bach: an Eternal Golden Braid. 1979

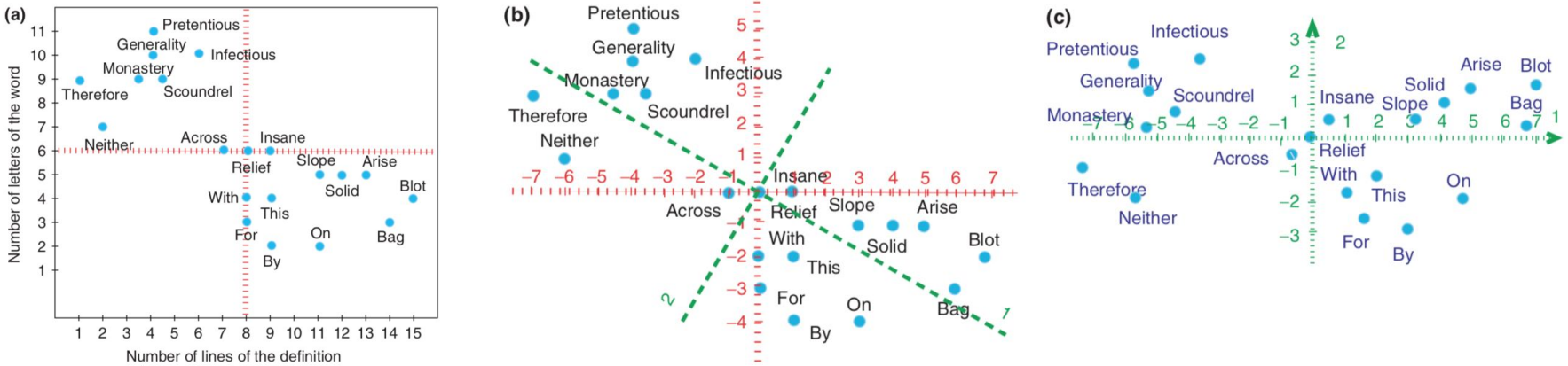
How to cope with high-dimensional data?

- Marginal distributions
  - Corner plot
- Dimension reduction
  - Principal component analysis (PCA)
  - t-distributed stochastic neighbor embedding (t-SNE)
  - Multidimensional scaling (MDS)
- Clustering
  - Distance/Similarity metric
  - Hierarchical clustering
  - K-means
  - Kernel methods

Dimension reduction

- Principal component analysis (PCA)
  - Preserves linear structure of the data
- Multidimensional scaling (MDS)
  - Preserves distance structure
- t-distributed stochastic neighbor embedding (t-SNE)
  - Preserves local structure

Principal Component Analysis (PCA)

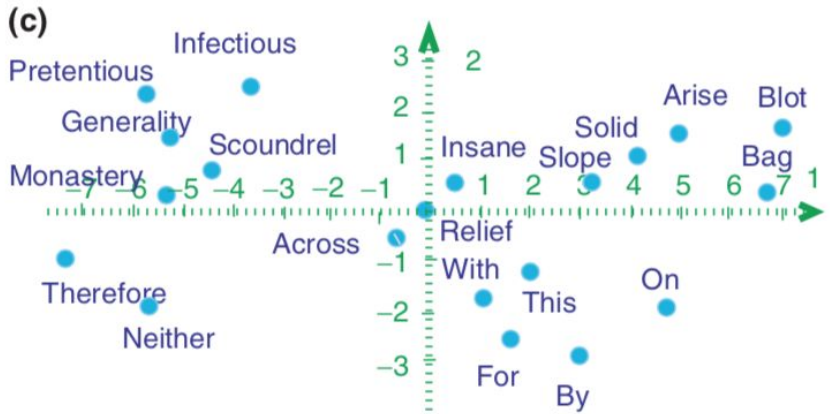


For each step, find an axis that maximizes the variance (along that axis)

H. Abdi, L. Williams. “Principal Component Analysis”. 2010.

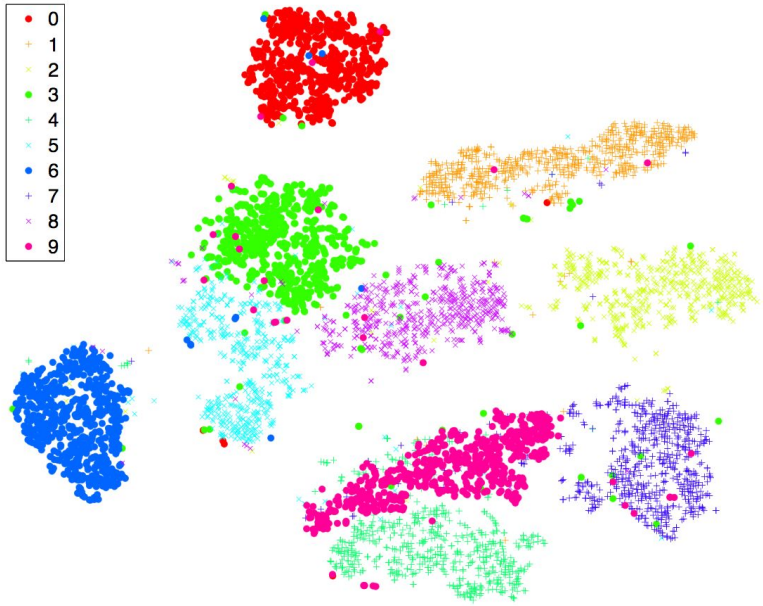
Interpretation of PCA

- Components are orthogonal to each other
- Each component is a linear combination of the original basis
- Also, component is a linear combination of training data points
  - One can characterize components by driving data points (contribution score)

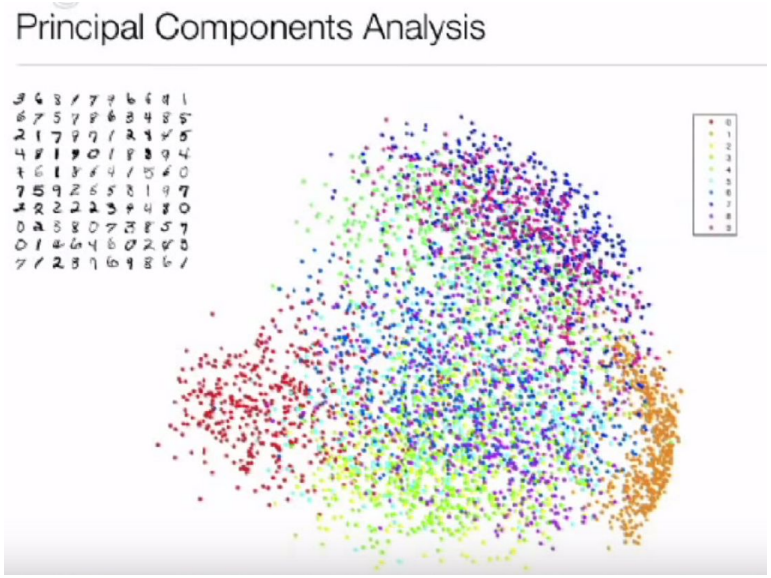


H. Abdi, L. Williams. “Principal Component Analysis”. 2010.

t-SNE: preserves local structure

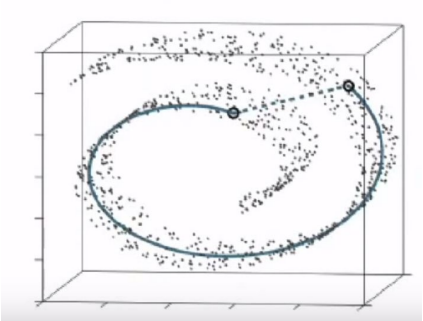


(a) Visualization by t-SNE.



L. Maaten & G. Hinton. JMLR 2008

t-SNE: preserves local structure



Local distance is reliable even in high dimension

1. Define similarities between i and j in the original space

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \qquad p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)},$$

2. Model the similarity in low-dimensional space with Student's t-distribution (Cauchy dist, heavy tail)

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq i} (1 + \|\mathbf{y}_i - \mathbf{y}_k\|^2)^{-1}}$$

3. Find {q_i} that minimizes KL divergence

$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

L. Maaten & G. Hinton. JMLR 2008

How to cope with high-dimensional data?

- Marginal distributions
  - Corner plot
- Dimension reduction
  - Principal component analysis (PCA)
  - t-distributed stochastic neighbor embedding (t-SNE)
  - Multidimensional scaling (MDS)
- Clustering
  - Distance/Similarity metric
  - Hierarchical clustering
  - K-means
  - Kernel methods

How to cope with high-dimensional data?

- Marginal distributions
  - Corner plot
- Dimension reduction
  - Principal component analysis (PCA)
  - t-distributed stochastic neighbor embedding (t-SNE)
  - Multidimensional scaling (MDS)
- Clustering
  - Distance/Similarity metric
  - Hierarchical clustering
  - K-means
  - Kernel methods

Clustering

- Hierarchical clustering
- K-means
  - Iterative algorithm to find local optima
  - Randomized start performs better (kmeans++)
- Gaussian Mixture
  - Soft clustering version of k-means
  - EM (expectation-maximization) algorithm
- Similarity/Distance metric
  - Kernel methods

K-means clustering

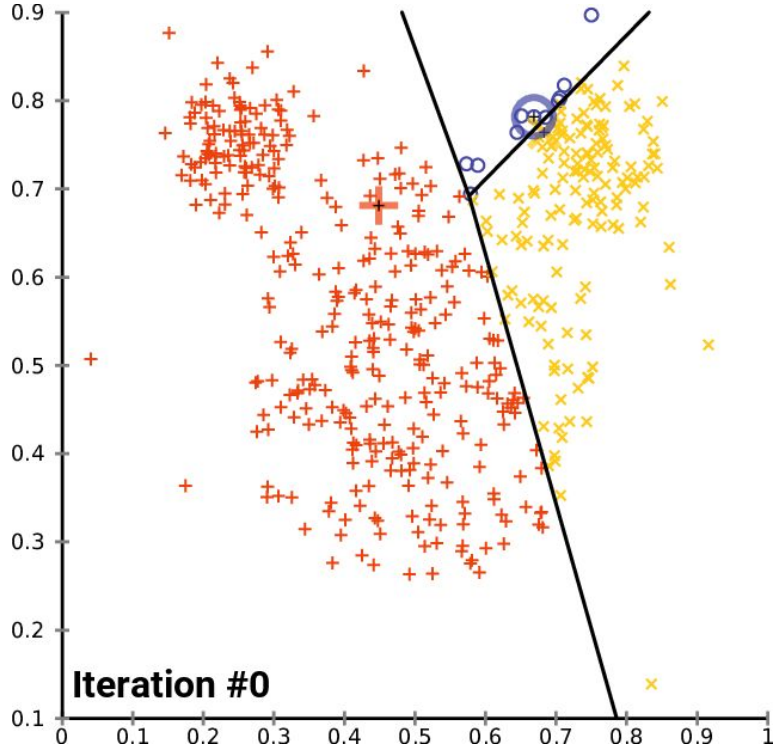
Given a data point and number of clusters K

1. Randomly initialize cluster centrers
2. Assign data points to the nearest cluster
3. Update the cluster centrers by taking the average of data points in the cluster
4. Go to step 2 and repeat

Acceleration of K-means:

Idea: use upper/lower bounds of distance

C. Elkan ICML 2003. G. Hamerly 2010.



K-means clustering: how to pick the initial centers?

- No guarantee for the global optima
- Careful random initialization of cluster centers is recommended (kmeans++)
  - D. Arthur and S. Vassilvitskii. SODA. 2007

2.2 The k-means++ algorithm

We propose a specific way of choosing centers for the `k-means` algorithm. In particular, let $D(x)$ denote the shortest distance from a data point to the closest center we have already chosen. Then, we define the following algorithm, which we call `k-means++`.

1a. Take one center c_1, chosen uniformly at random from $\mathcal{X}$.
1b. Take a new center c_i, choosing $x \in \mathcal{X}$ with probability $\frac{D(x)^2}{\sum_{x \in \mathcal{X}} D(x)^2}$.
1c. Repeat Step 1b. until we have taken k centers altogether.
2-4. Proceed as with the standard `k-means` algorithm.

We call the weighting used in Step 1b simply "D^2 weighting".

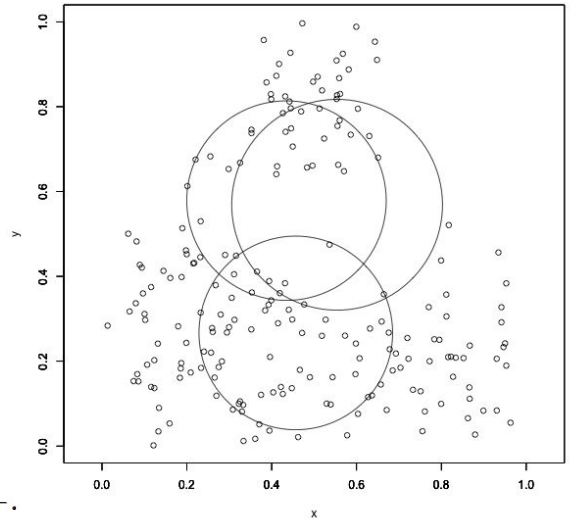
Theorem 3.1. *If $\mathcal{C}$ is constructed with `k-means++`, then the corresponding potential function ϕ satisfies, $E[\phi] \leq 8(\ln k + 2)\phi_{\mathrm{OPT}}$.*

Gaussian mixture models

Model: data points are generated from mixture of Gaussian distribution

Solution: EM algorithm

- E-step: compute responsibilities $\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}.$



- M-step: re-estimate model parameters

$$\boldsymbol{\mu}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n$$

$$\boldsymbol{\Sigma}_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}}$$

$$\pi_k^{\text{new}} = \frac{N_k}{N}$$

This is a soft clustering version of k-means

Distance/Similarity metric

- Distance / Similarity measure can dramatically change the results of clustering/dimension reduction

Sequence 1	A	T	G	C	A
Sequence 2	A	G	C	A	T

Mismatch: 4

Sequence 1	A	T	G	C	A
Sequence 2	A	G	C	A	T

Biologically, there is one insertion

- Proper design of distance/similarity metric requires domain knowledge

How to find good distance/similarity metric?

- Use domain knowledge
  - Feature engineering
- Kernel trick
  - Similarity measure <--> inner product
  - One can think about mapping to high dimensional space
  - You don’t need to explicitly compute the mapping, but can have similarity measures
- Representation learning with Neural nets

Today’s topics

- Types of plots and relevant topics
- How to cope with high-dimensional data?
- Interactive plotting
- Resource: useful software/packages

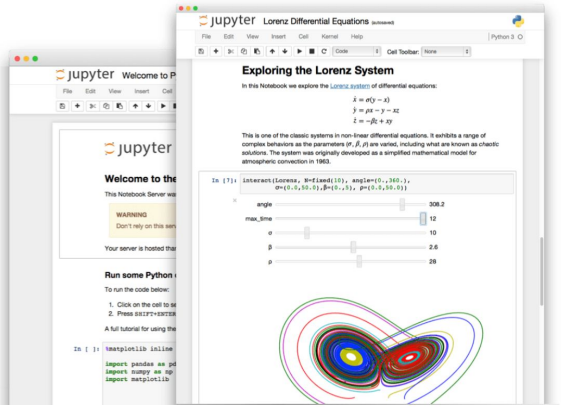
Interactive plotting

- Interactive plotting can accelerate exploratory data analysis
- Demo1: Plotly @ Jupyter notebook
  - https://github.com/biods215/biods215.github.io/blob/master/lecture_material/Visualization/2018/Plotly_on_Jupyter_example.html
- Demo2: Python Dash App
  - https://plot.ly/products/dash/
- Demo3: Global Biobank Engine
  - https://gbe.stanford.edu

Today’s topics

- Types of plots and relevant topics
- How to cope with high-dimensional data?
- Interactive plotting
- Resource: useful software/packages

Jupyter notebook: tool for reproducible research



The Jupyter Notebook

The Jupyter Notebook is an open-source web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. Uses include: data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more.

- Useful packages:
  - R kernel
  - Bash kernel
  - Jupyter_contrib_nbextensions
- Colaboratory
  - Google Docs + Jupyter

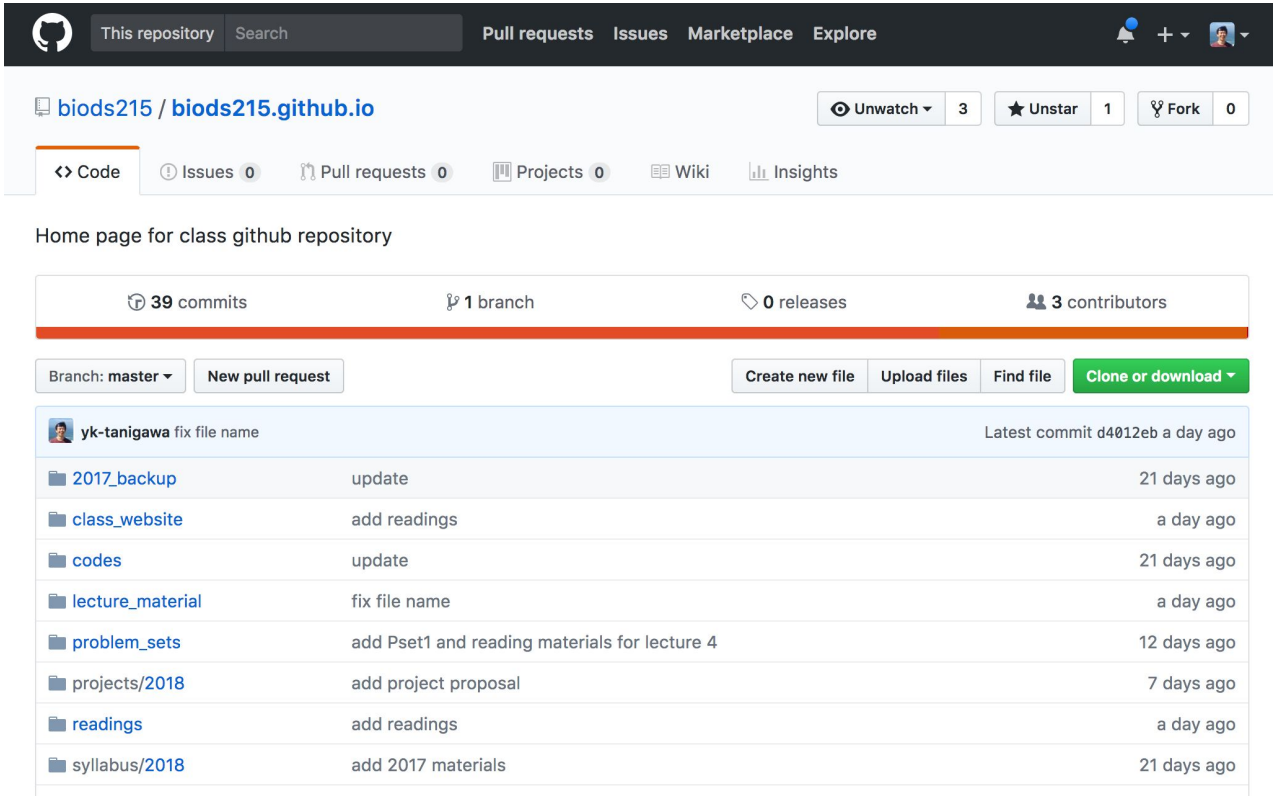
</> Language of choice

Share notebooks

Interactive output

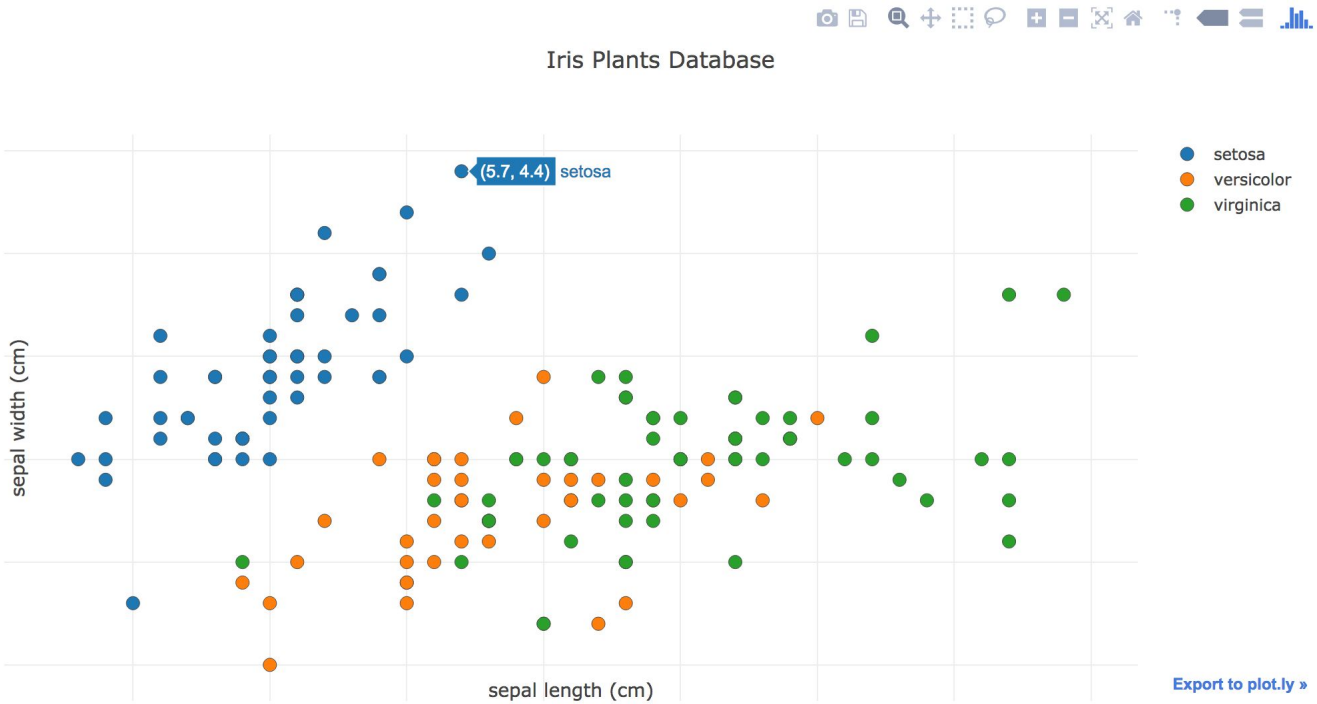
Spark Big data integration

git + GitHub: version control system



Version control is important for reproducible research

Plotly: interactive plotting API



Interactive plotting

Python/R/JS APIs

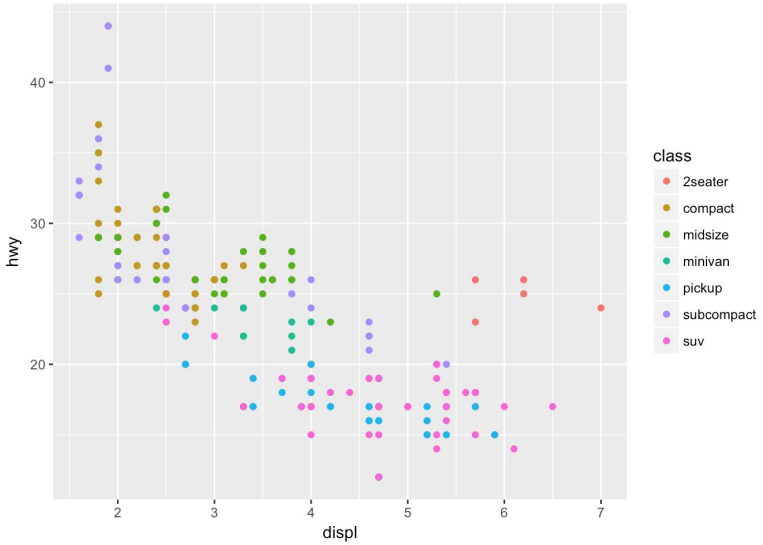
Dash App is nice

ggplot2/matplotlib/seaborn: plot library for R/Python

ggplot2 part of the tidyverse

```
library(ggplot2)

ggplot(mpg, aes(displ, hwy, colour = class)) +
  geom_point()
```



matplotlib

home | examples | tutorials | pyplot | docs »

Introduction

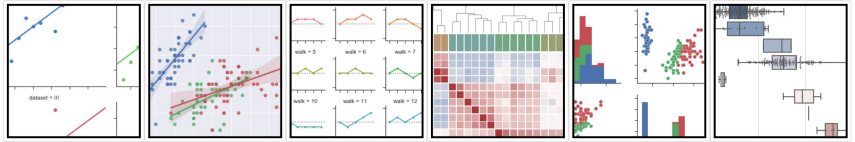
Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web application servers, and four graphical 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 examples, see the sample plots and thumbnail gallery.

For simple plotting the `pyplot` module 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.

seaborn 0.8.1 Gallery Tutorial API Site Page Search

seaborn: statistical data visualization



Today’s contents

- Descriptive statistics alone can mislead us.
- Types of plots and relevant topics
- How to cope with high-dimensional data?
- Interactive plotting
- Resource: useful software/packages

Questions?