Data Mining

CS57300
Purdue University

Bruno Ribeiro

February 1st, 2018

Exploratory Data Analysis & Feature Construction

- How to explore a dataset
  - Understanding the variables (values, ranges, and empirical distribution)
  - Finding easy relationships between variables
- Building features from data
  - How to better represent the data for various data mining tasks

Data exploration and visualization

Visualization

- Human eye/brain have evolved powerful methods to detect structure in nature

- Display data in ways that exploit human pattern recognition abilities

- Limitation: Can be difficult to apply if data size (number of dimensions or instances) is large

Exploratory data analysis

- Data analysis approach that employs a number of (mostly graphical) techniques to:
  - Maximize insight into data
  - Uncover underlying structure
  - Identify important variables
  - Detect outliers and anomalies
  - Test underlying modeling assumptions
  - Develop parsimonious models
  - **Generate hypotheses from data**

Visualizing/summarizing data

- Low-dimensional data
  - Summarizing data with simple statistics
  - Plotting raw data (1D, 2D, 3D)
- Higher-dimensional data
  - Principal component analysis
  - Multidimensional scaling

Data summarization

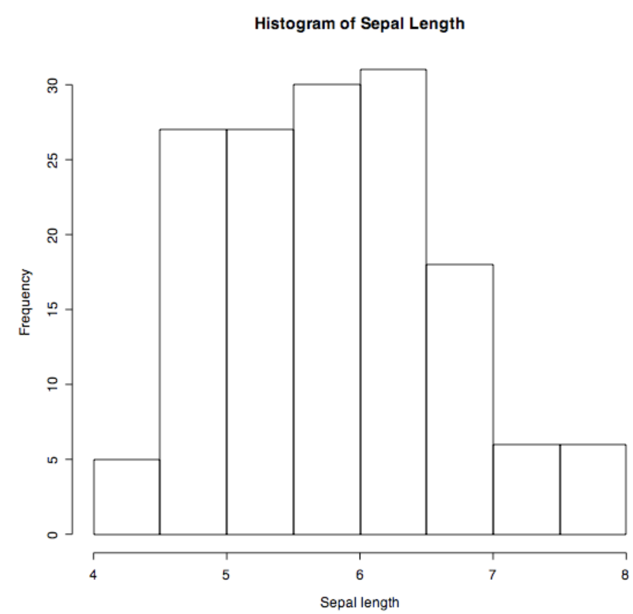
- Measures of location
  - Mean: $\hat{\mu} = \frac{1}{n}\sum_{i=1}^{n} x(i)$
  - Median: value with 50% of points above and below
  - Quartile: value with 25% (75%) points above and below
  - Mode: most common value

Data summarization

- Measures of dispersion or variability
  - Variance: $\hat{\sigma}_k^2 = \frac{1}{n}\sum_{i=1}^{n}(x(i)-\mu)^2$
  - Standard deviation: $\hat{\sigma}_k = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(x(i)-\mu)^2}$
  - Range: difference between max and min point
  - Interquartile range: difference between 1st and 3rd Q
  - Skew: $\frac{\sum_{i=1}^{n}(x(i)-\hat{\mu})^3}{\left(\sum_{i=1}^{n}(x(i)-\hat{\mu})^2\right)^{\frac{3}{2}}}$

Histograms (1D)

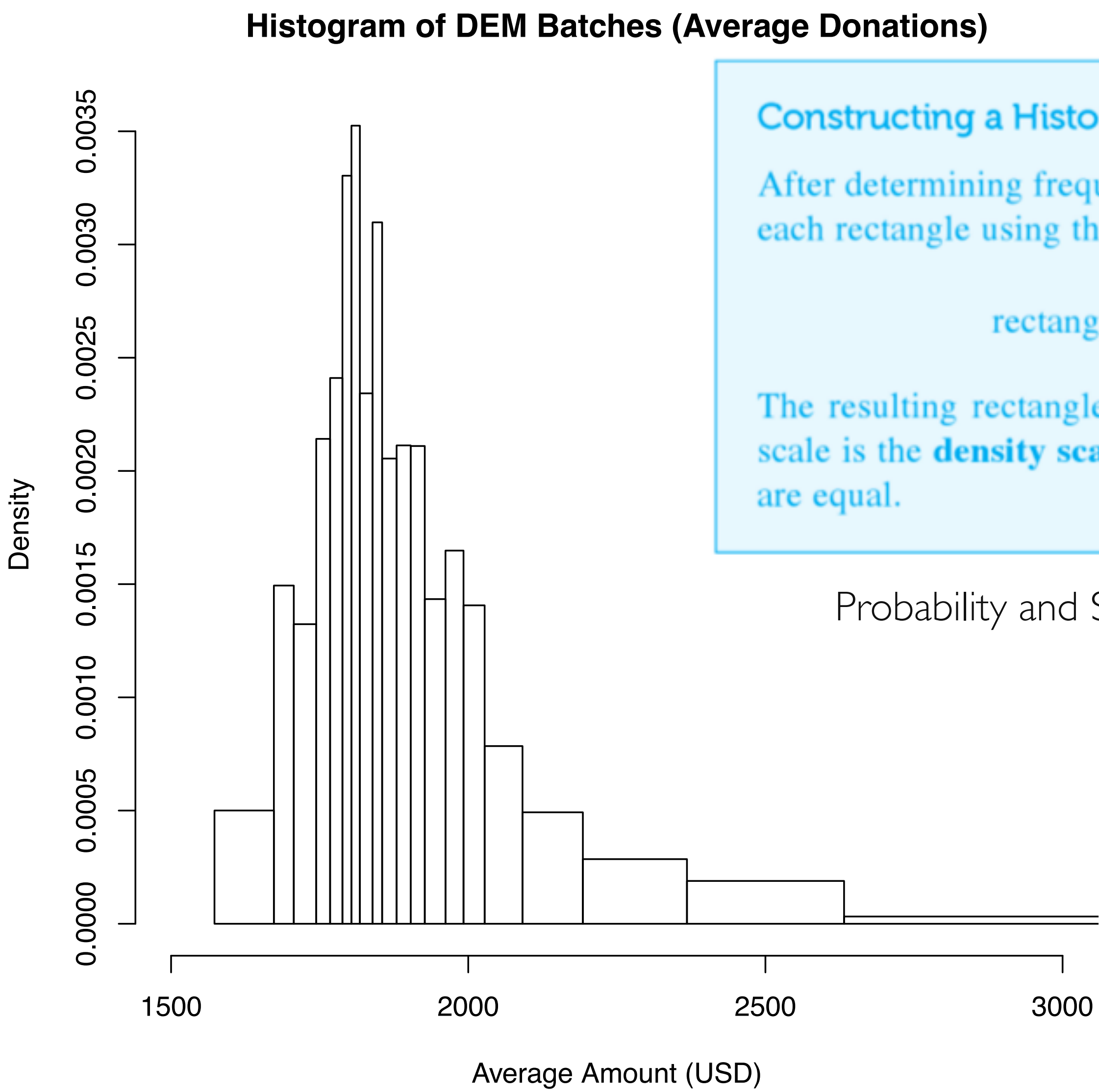
- Most common plot for univariate data
- Split data range into equal-sized bins, count number of data points that fall into each bin
- Graphically shows:
    - Center (location)
    - Spread (scale)
    - Skew
    - Outliers
    - Multiple modes



Problem with Standard Histograms

- Test the following on R:
  - hist(c(0,0,0,2,3),breaks=c(0,1-1e-2,2,3))
  - hist(c(0,0,0,2,3),breaks=c(0,0.5,1-1e-2,2,3))
  - hist(c(0,0,0,2,3),breaks=c(0,0.5,1-1e-2,1.5,2,2.5,3))
- Standard histograms give inconsistent results for continuous data
  - Visualization highly dependent on binning

Quantile Histogram (better)



Constructing a Histogram for Continuous Data: Unequal Class Widths

After determining frequencies and relative frequencies, calculate the height of each rectangle using the formula

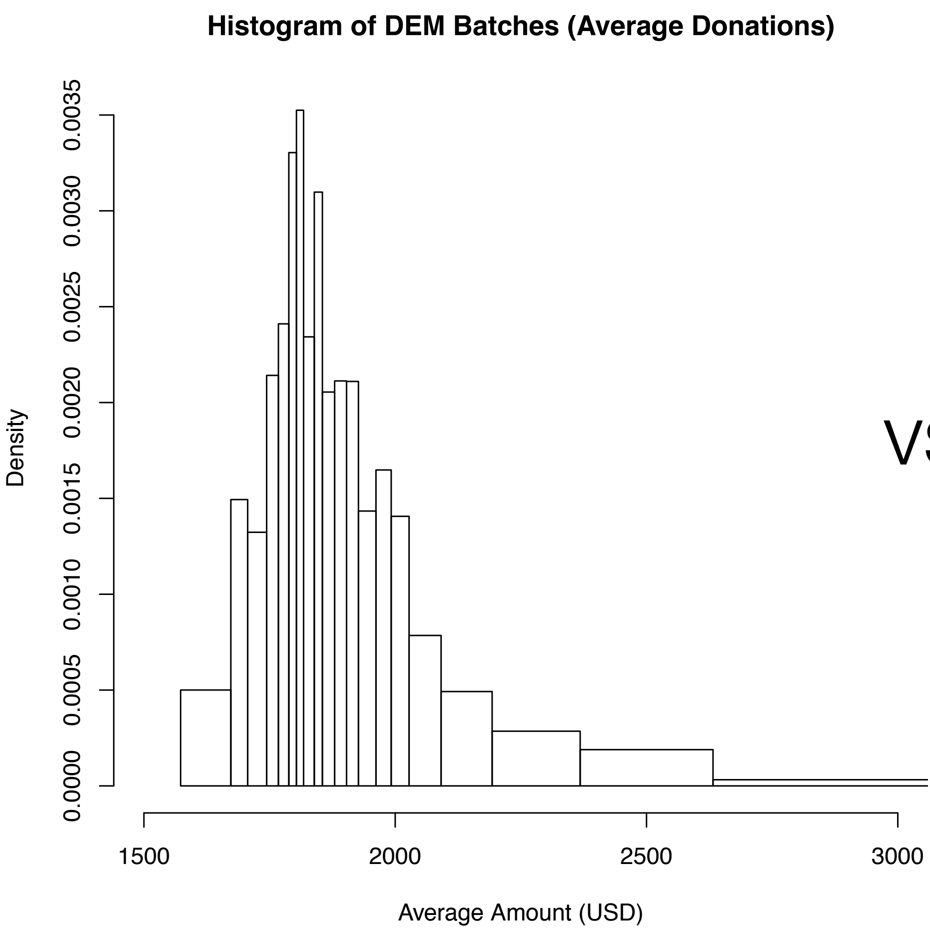
$$\text{rectangle height} = \frac{\text{relative frequency of the class}}{\text{class width}}$$

The resulting rectangle heights are usually called *densities*, and the vertical scale is the **density scale.** This prescription will also work when class widths are equal.

Probability and Statistics for Engineering and the Sciences, Jay Devore, 9th Ed, Brooks Cole

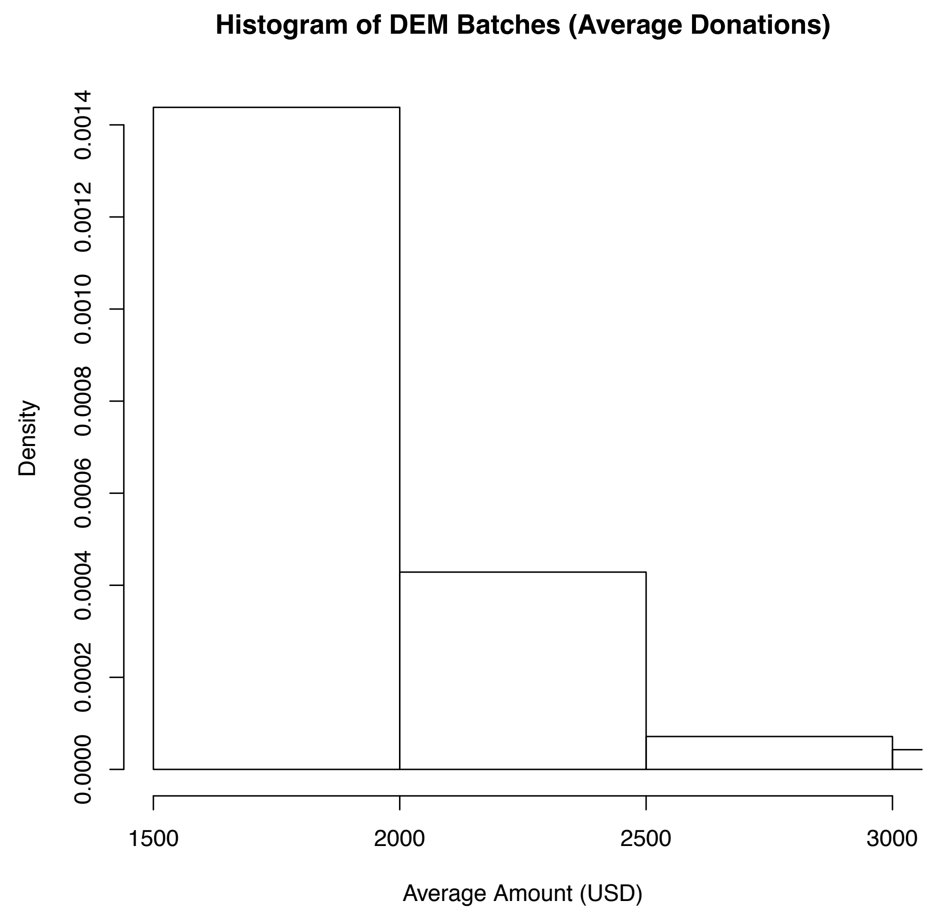
Each bin represents approximately the same number of data points

Quantile Histogram (better)



Quantile histogram

VS



Standard histogram

R code (of example)

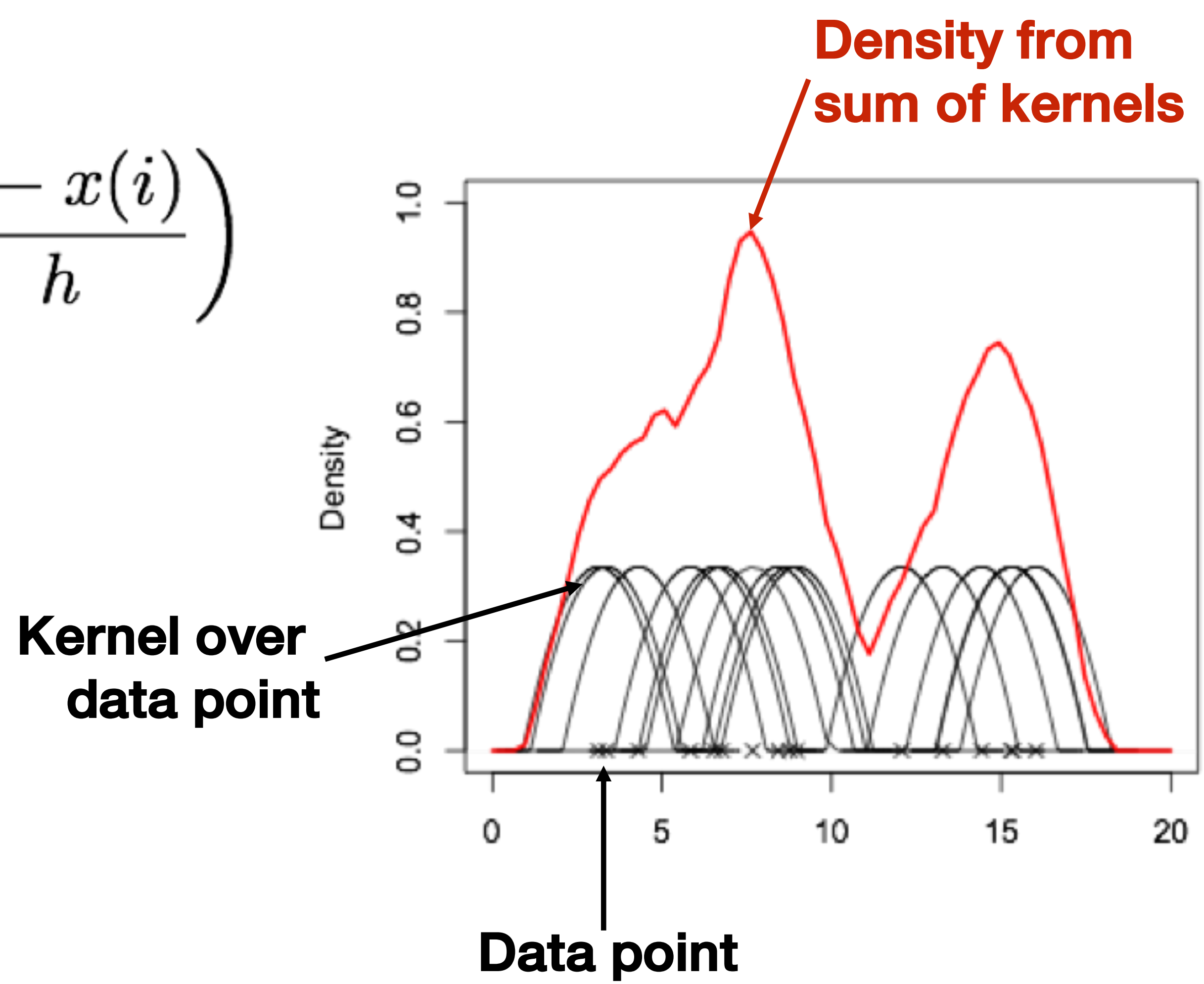
```
dem = read.csv("DEM_donations_2012.csv",header=TRUE)
gop = read.csv("GOP_donations_2012.csv",header=TRUE)

demdonations = as.matrix(dem$donation[dem$donation > 0])
gopdonations = as.matrix(gop$donation[gop$donation > 0])

N = round(nrow(demdonations)/200)

# randomly split DEMs into N groups of average length nrow(demdonations)/N
demsplit = as.list(split(demdonations, sample(1:N, nrow(demdonations), replace=T)))

# find average of each bin
dem_bin_averages = unlist(lapply(demsplit,mean))

# find 20 quantiles of each the batch averages
quantiles_dem = quantile(dem_bin_averages,probs = seq(0, 1, 1/20))

# plot histogram
hist(dem_bin_averages,main="Histogram of DEM Batches (Average Donations)",freq=FALSE,breaks=quantiles_dem,xlim=c(1500,3000))


N = round(nrow(gopdonations)/200)

# randomly split GOP into N groups of average length nrow(demdonations)/N
gopsplit = split(gopdonations, sample(1:N, nrow(gopdonations), replace=T))

# find average of each bin
gop_bin_averages = unlist(lapply(gopsplit,mean))

# find 20 quantiles of each the batch averages
quantiles_gop = quantile(gop_bin_averages,probs = seq(0, 1, 1/20))

# plot histogram
hist(gop_bin_averages,main="Histogram of GOP Batches (Average Donations)",freq=FALSE,breaks=quantiles_gop,xlim=c(1500,3000))
```

Alternative to Histograms: Density plots

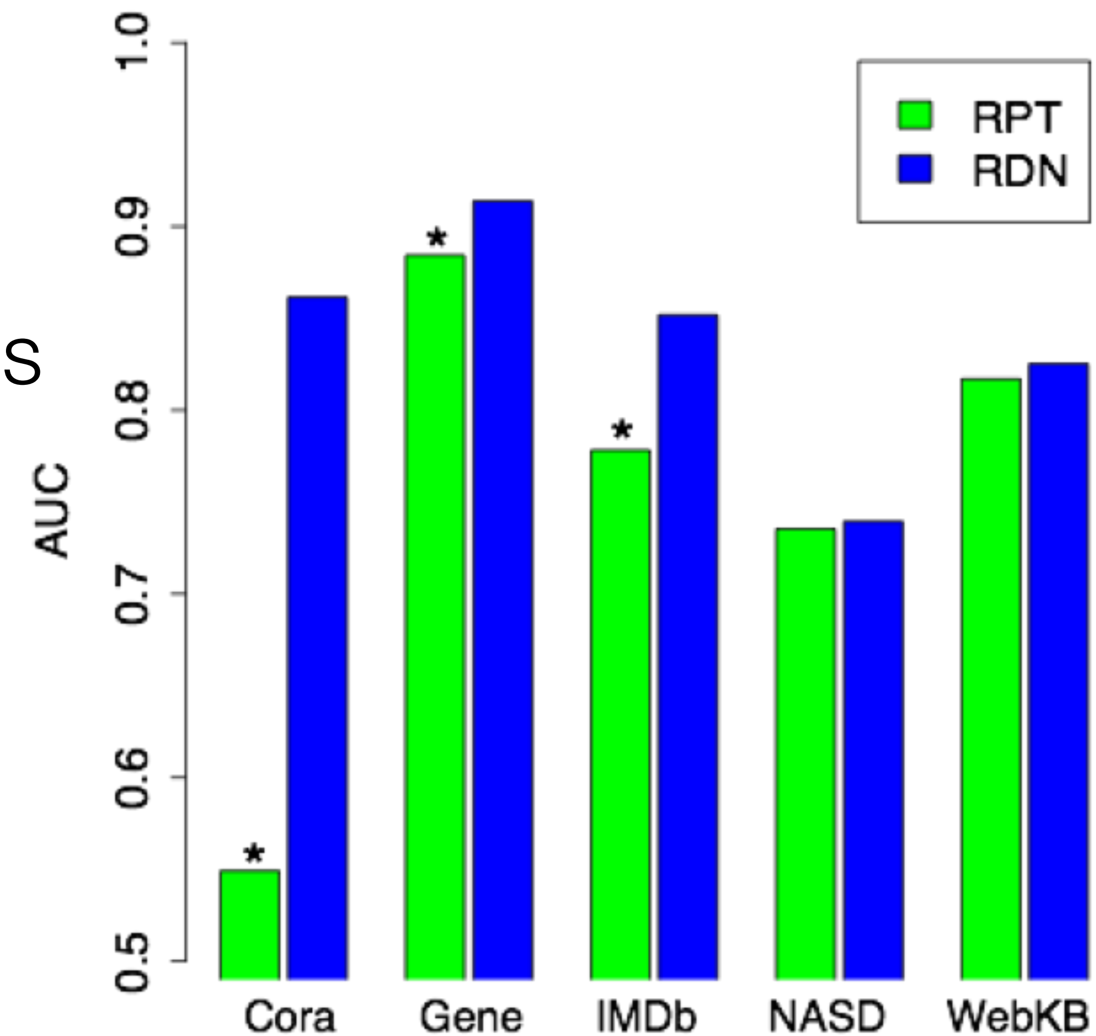
- Estimated density is:

$$\hat{f}(x) = \frac{1}{n} \sum_{i=1}^{n} K\left(\frac{x - x(i)}{h}\right)$$

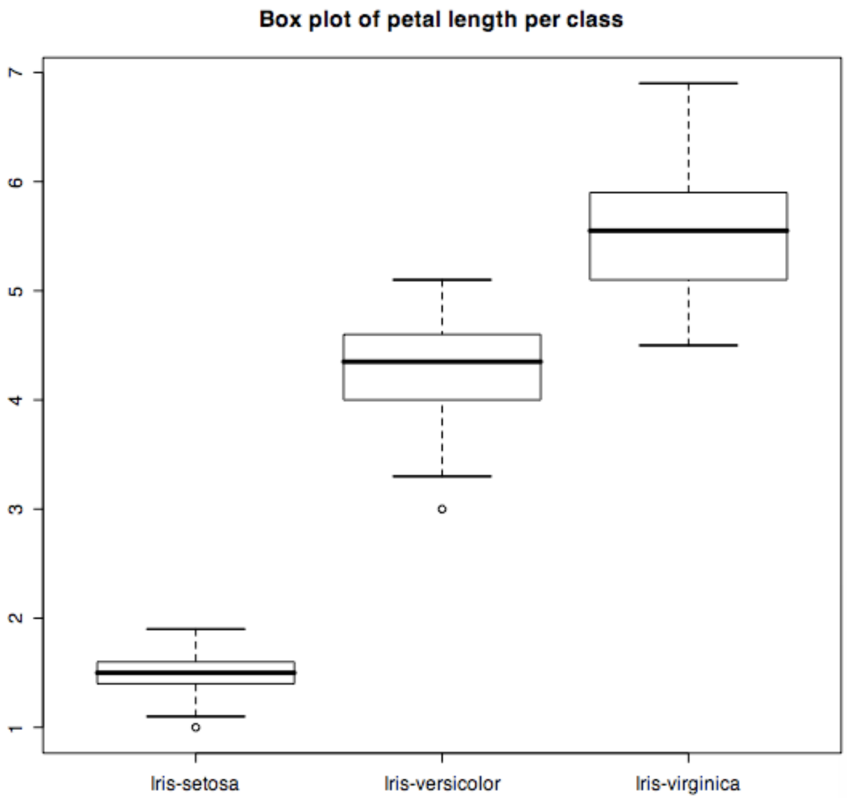
- Two parameters:
  - Kernel function K (e.g., Gaussian, Epanechnikov)
  - Bandwidth h



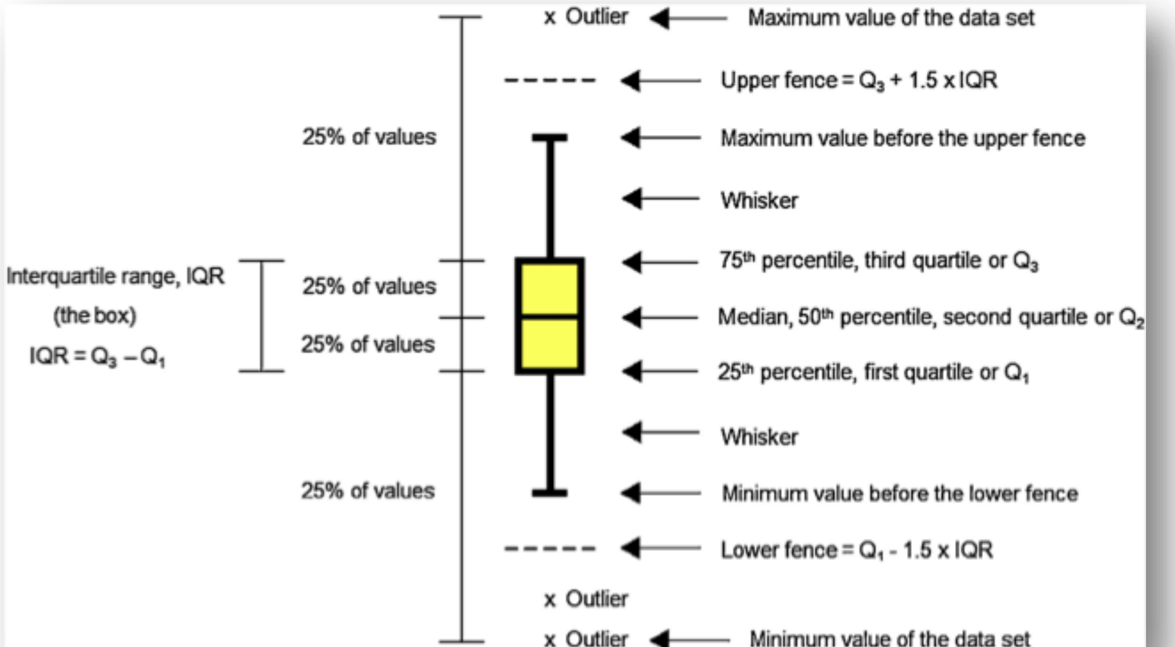
Bar plots



Box plot (2D)



- Display relationship between discrete and continuous variables
- For each discrete value X, calculate quartiles and range of associated Y values

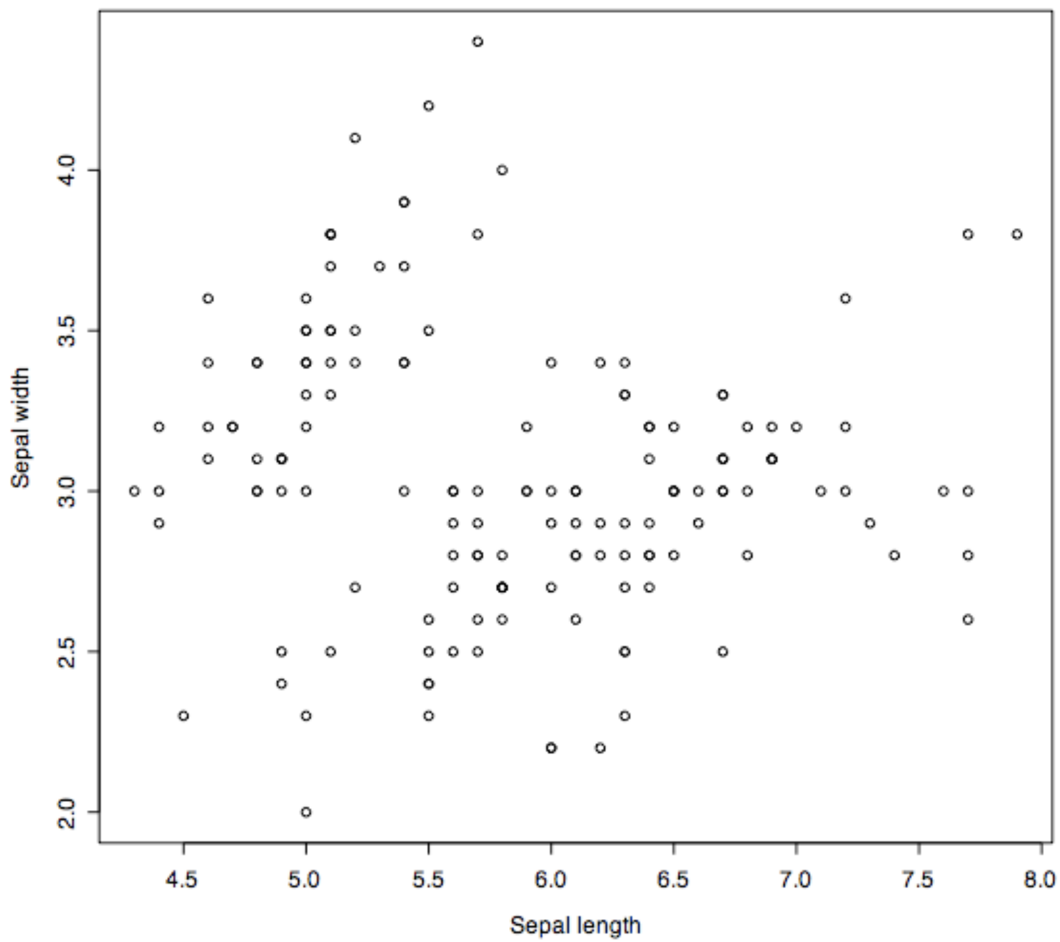


Ack: João Elias Vidueira Ferreira

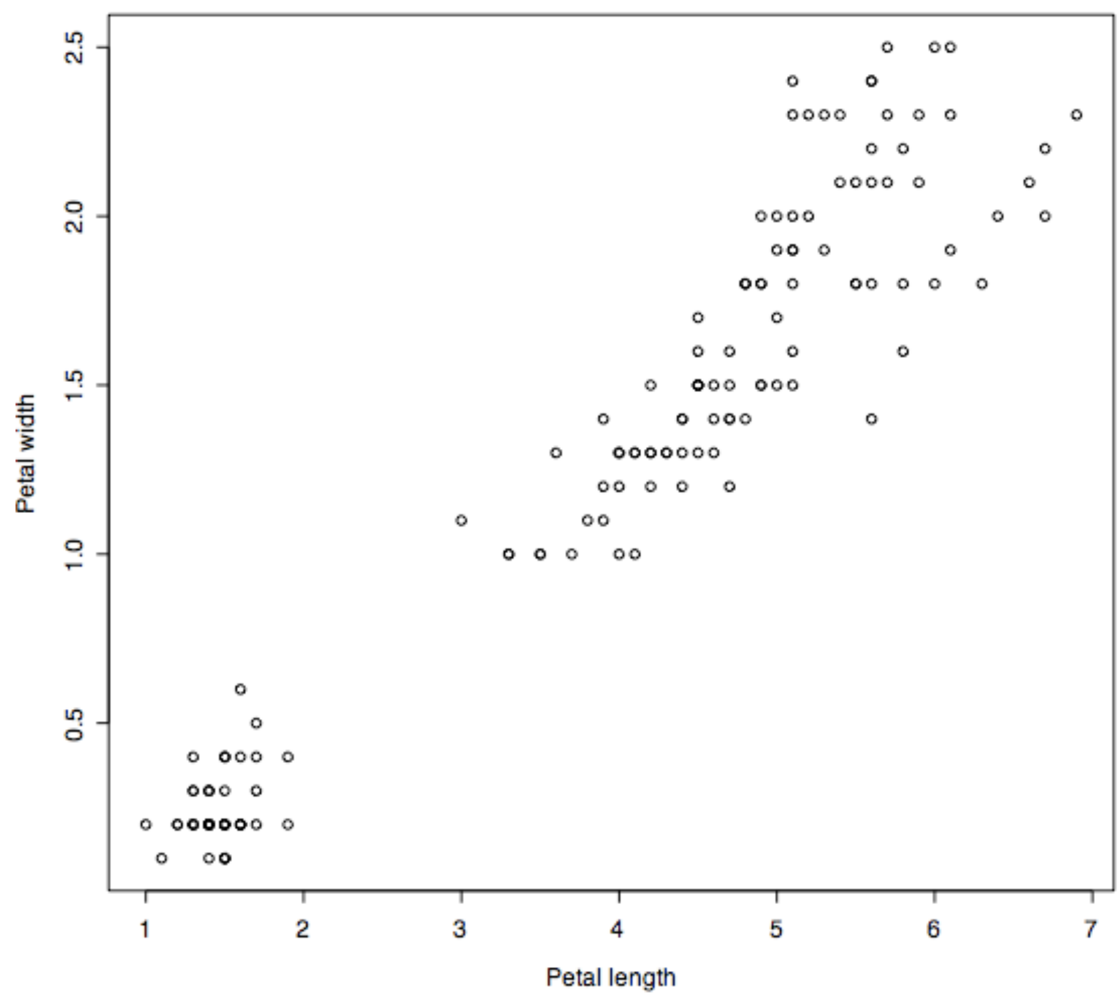
Scatter plot (2D)

- Most common plot for bivariate data
  - Horizontal X axis: the suspected **independent** variable
  - Vertical Y axis: the suspected **dependent** variable
- Graphically shows:
  - If X and Y are related
  - Linear or non-linear relationship
  - If the variation in Y depends on X
  - Outliers

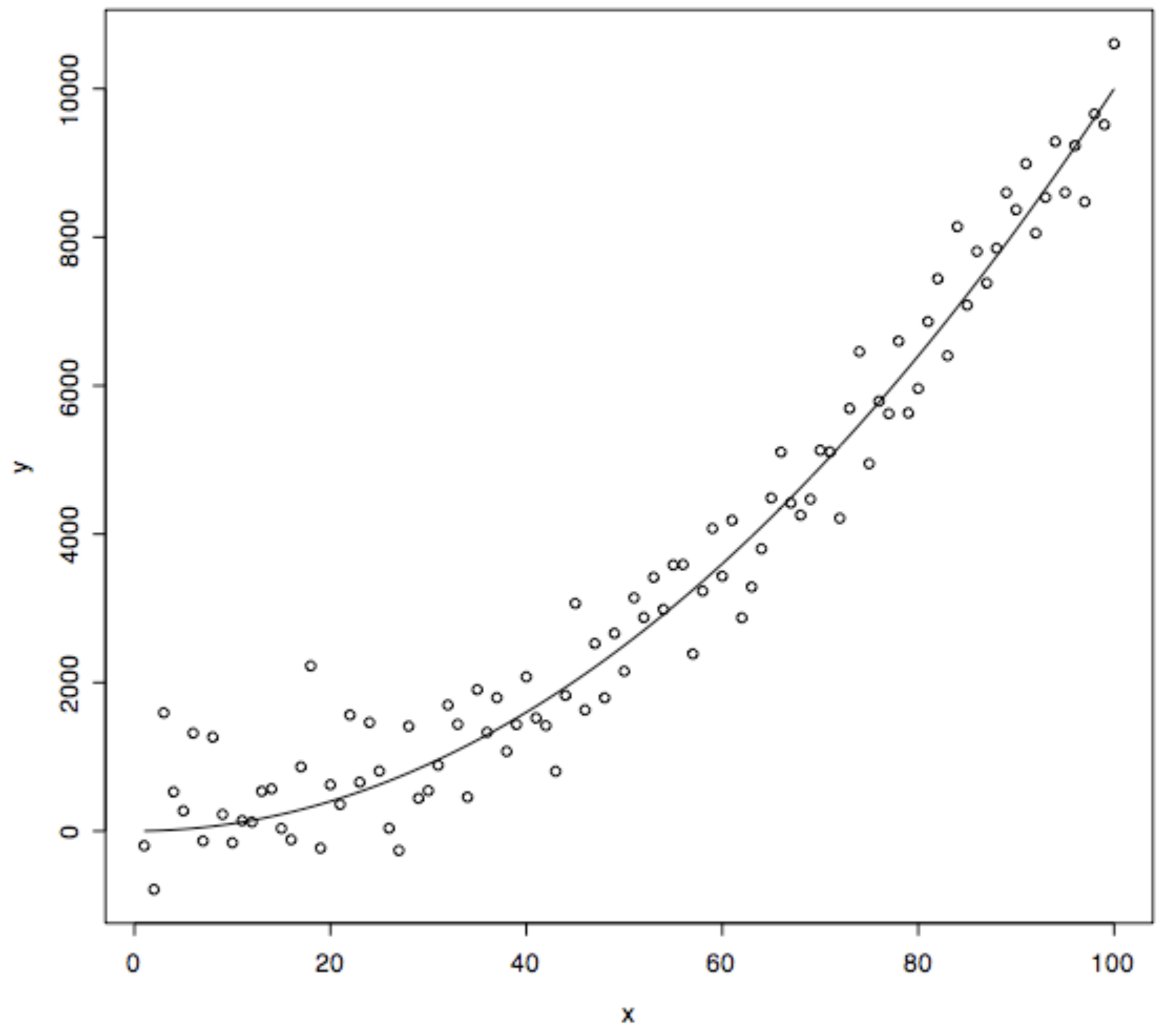
No relationship



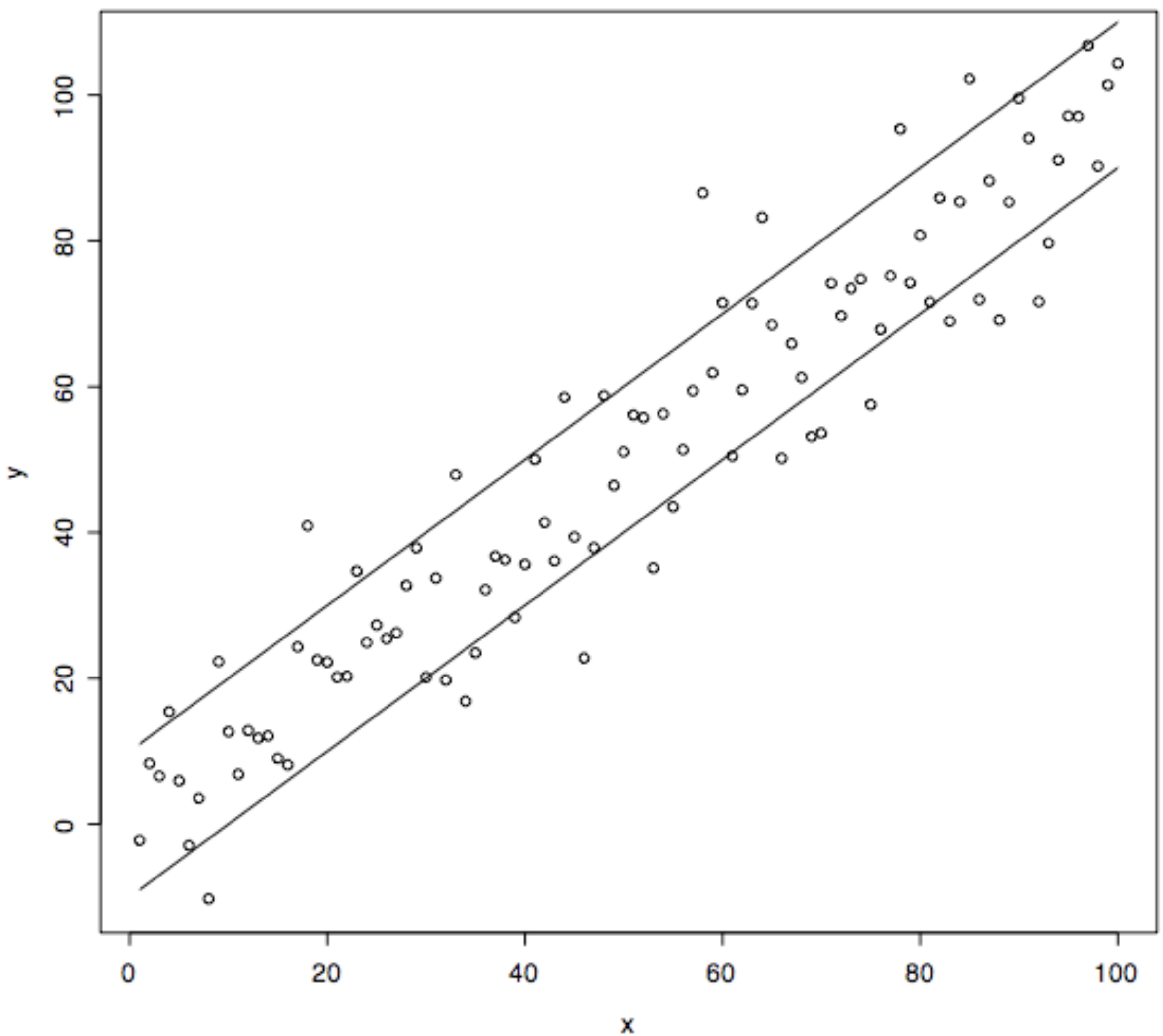
Linear relationship



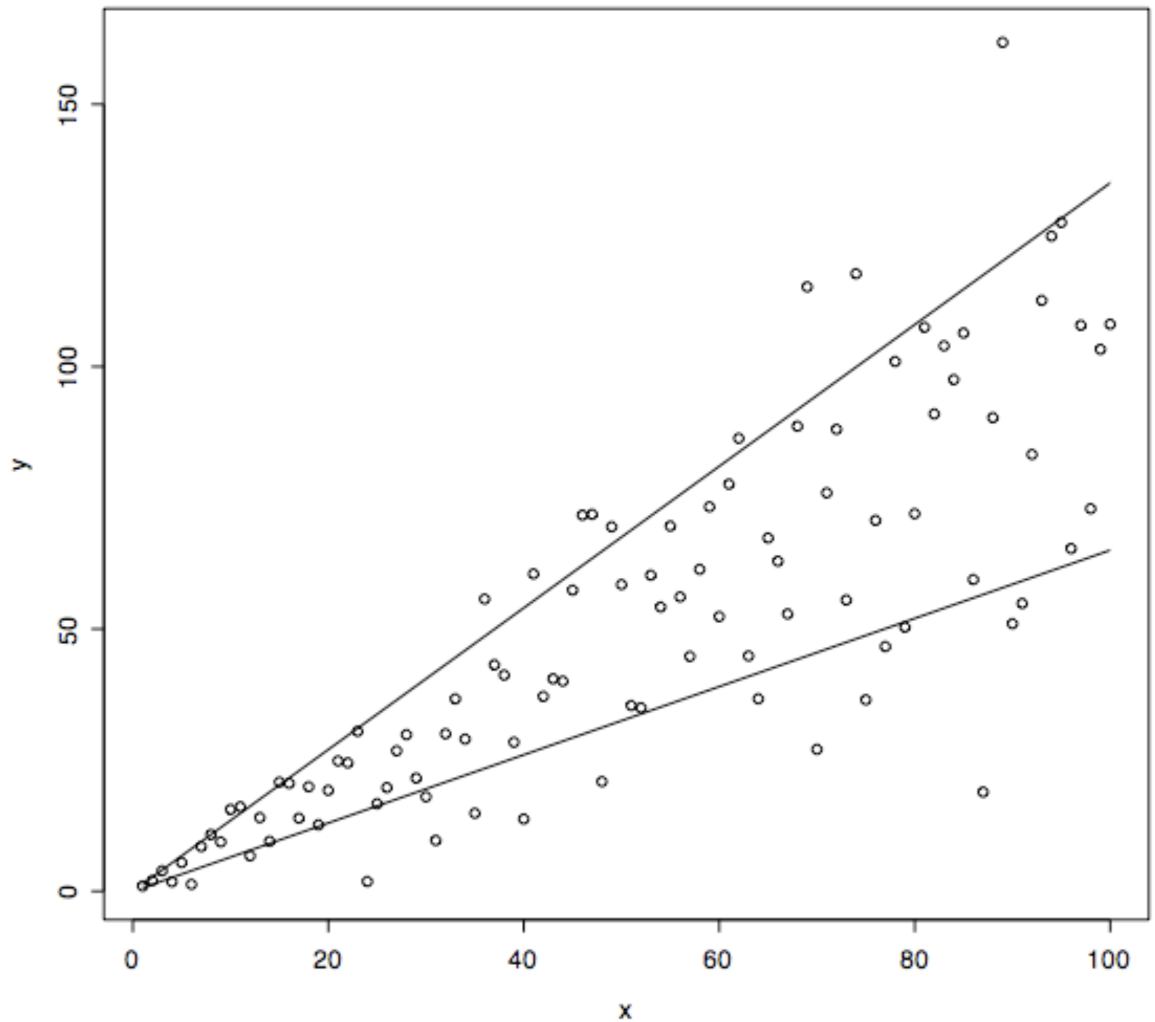
Non-linear relationship



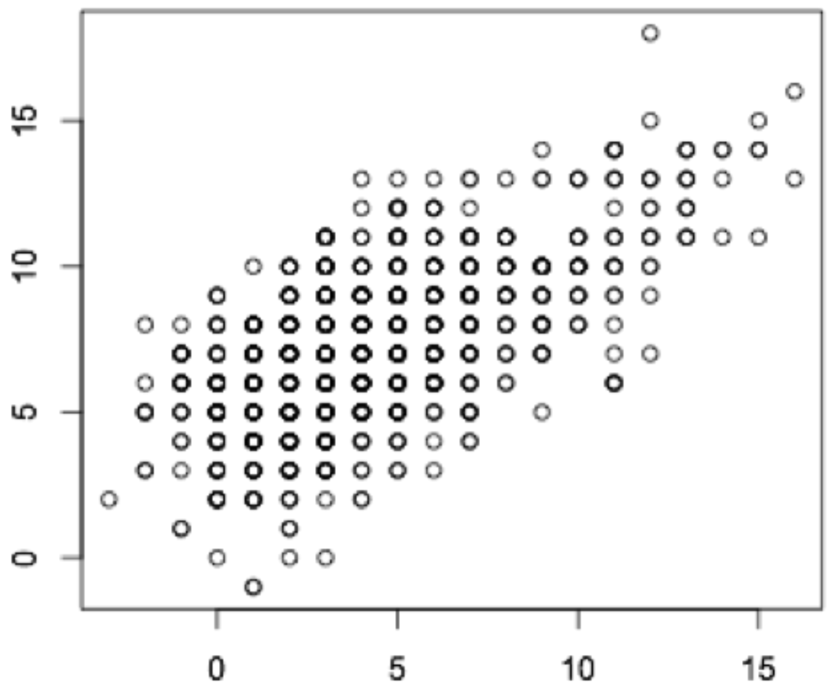
Homoskedastic (equal variance)



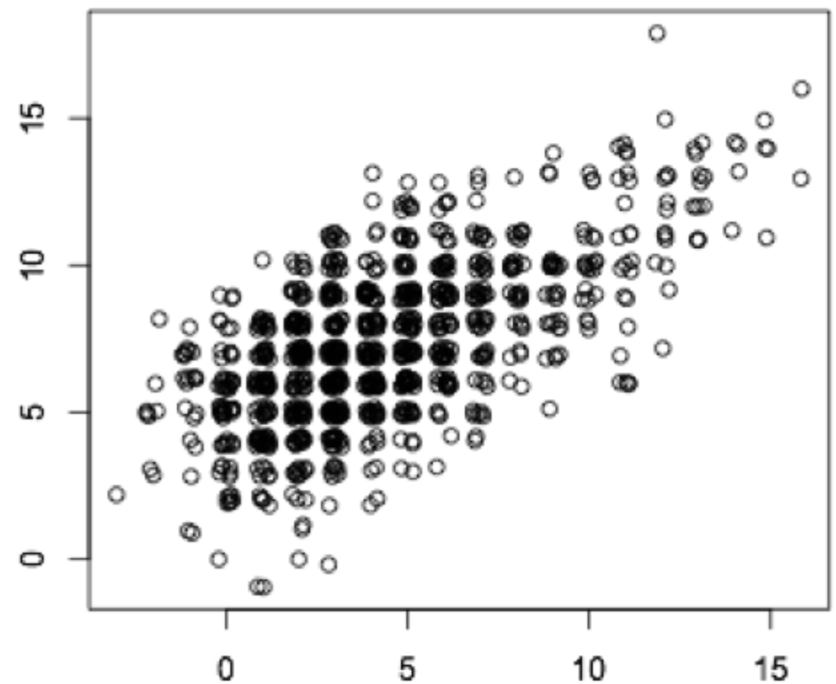
Heteroskedastic (unequal variance)



Scatterplot limitations



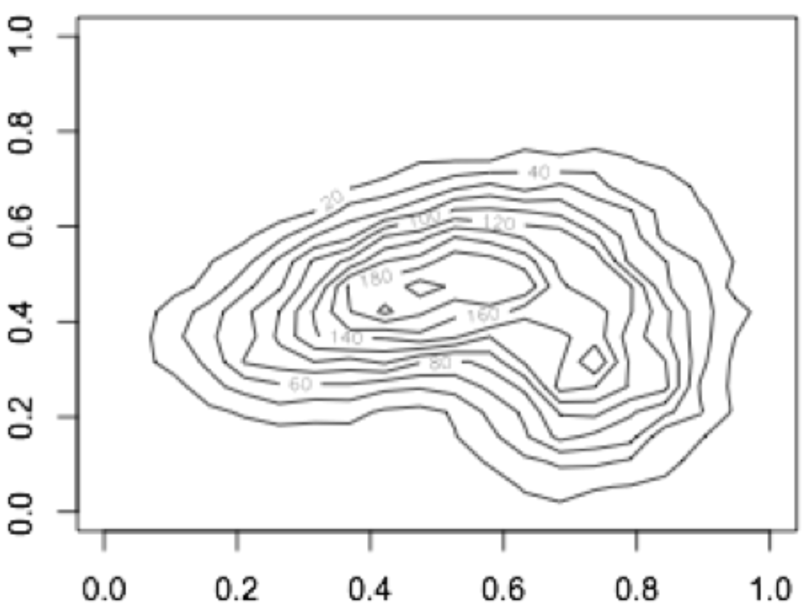
Overprinting



Solution: Jitter points

Contour plot (3D)

- Represents a 3D surface by plotting constant z slices (contours) in a 2D format



- Can overcome some limitations of 2D scatterplot (e.g., when there is too much data to discern relationship)

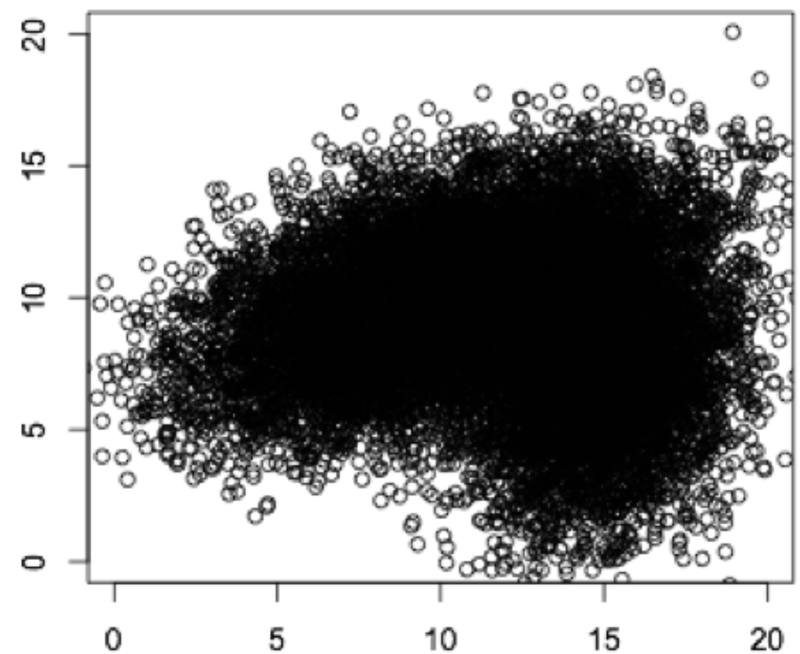
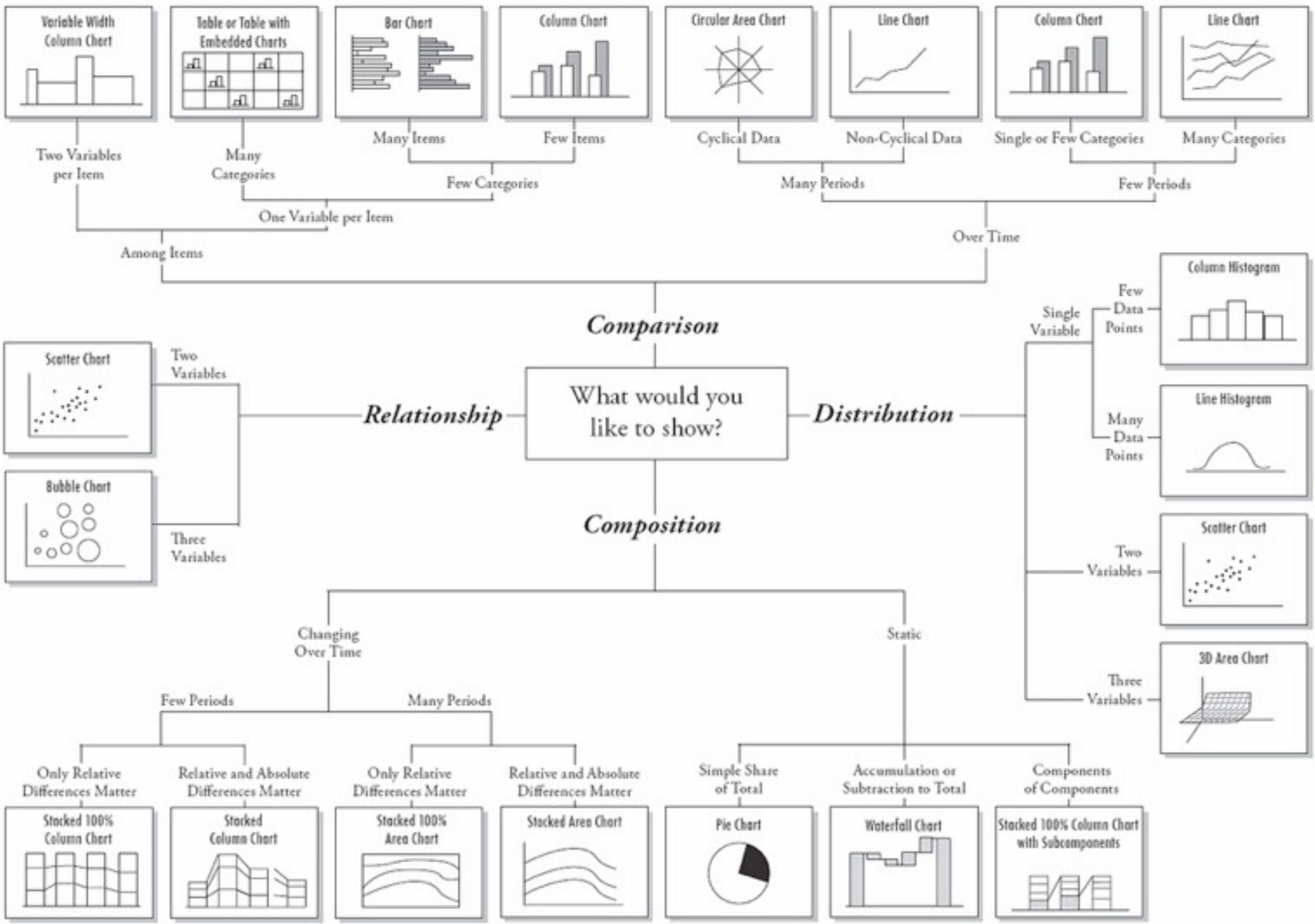


Chart Suggestions—A Thought-Starter



http://extremepresentation.typepad.com/blog/2006/09/choosing_a_good.html

© 2006 A. Abela — a.v.abela@gmail.co

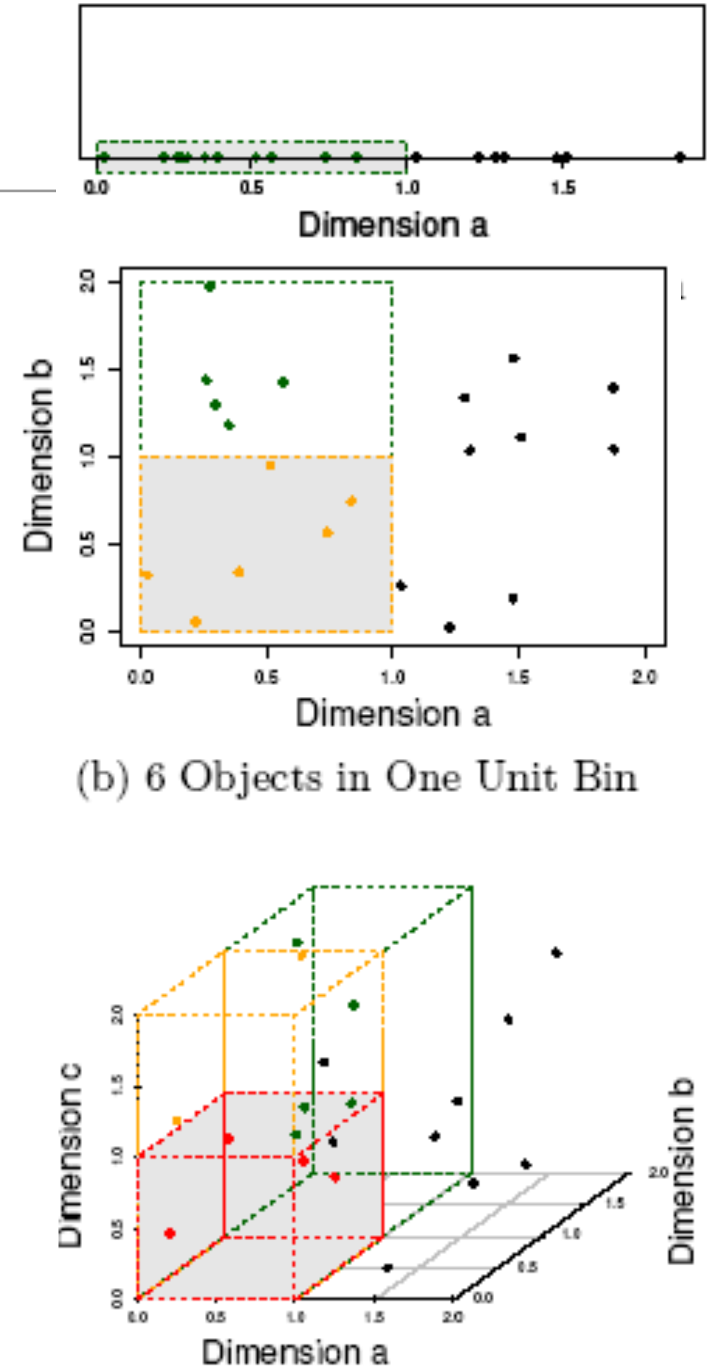
Constructing Features from Data

Whitening (Normalization)

- For numerical features (not categorical)
  - It is common to whiten each feature by subtracting its mean and dividing by its variance
- For regularization, this helps all the features be penalized in the same units, that is, we are assuming they have the same variance σ^2

The Curse of Dimensionality

- Data in only one dimension is relatively packed
- Adding a dimension “stretches” the points across that dimension, making them further apart
- Adding more dimensions will make the points further apart—high dimensional data is extremely sparse
- Distance measure becomes meaningless



(b) 6 Objects in One Unit Bin

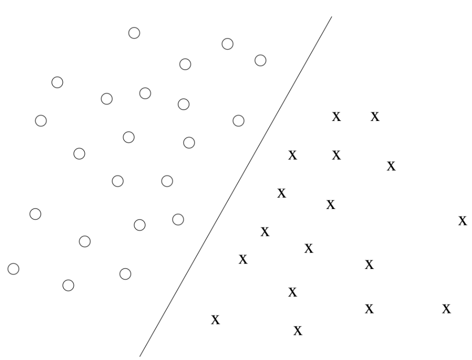
(c) 4 Objects in One Unit Bin

(ack Evimaria Terzi, Parsons et al.)

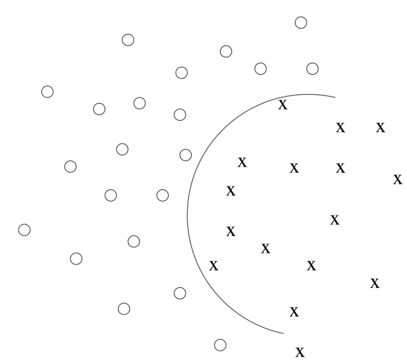
Kernels

- A kernel K is a form of similarity function
    - K(u,v) > 0 is the similarity between vectors u, v ∈ X

- Mercer's theorem: For every continuous symmetric positive semi-definite kernel K there is a feature vector function ϕ such that
    - K(u,v) = ϕ(u) ϕ(v)



Linear separator in the feature ϕ-space



Non-linear separator in the original $\mathbf{x}$-space

Fig ack Tommi Jaakkola

Some Common Kernels

- Polynomials of degree up to d

$$K(u,v) = (u^T v + c)^d$$

- Gaussian/Radial kernels (polynomials of all orders –projected space has infinite dimension)

$$K(u,v) = \exp\left(-\frac{\|u-v\|^2}{2\sigma^2}\right)$$

- Sigmoid

$$K(u,v) = \tanh\left(au^T v + c\right)$$

Other Forms of Dimensionality Reduction

- Dataset **X** consisting of **n** points in a **d**-dimensional space
- Data point $\mathbf{x_i} \in \mathbf{R^d}$ (**d**-dimensional real vector):

$$\mathbf{x_i = [x_{i1}, x_{i2}, \ldots, x_{id}]}$$

- Dimensionality reduction methods:
  - **Feature selection:** choose a subset of the features
  - **Feature extraction:** create new features by combining new ones

Dimensionality reduction

- Dimensionality reduction methods:
  - **Feature selection:** choose a subset of the features
  - **Feature extraction:** create new features by combining new ones
- Both methods map vector $\mathbf{x_i} \in \mathbf{R^d}$, to vector $\mathbf{y_i} \in \mathbf{R^k}$, **(k<<d)**
- $\mathbf{F : R^d \rightarrow R^k}$

Random Projections

- It is also possible to learn models & classifiers over random projections of the data
- Johnson-Lindenstrauss Lemma
  - A given a set $S \in R^n$, if we perform an orthogonal projection of those points onto a random *d*-dimensional subspace, then $d = O(\gamma^{-2} \log |S|)$ is sufficient so that with high probability all pairwise distances are preserved up to $1 \pm \gamma$

Finding random projections

- Vectors $\mathbf{x_i} \in \mathbf{R^d}$, are projected onto a **k**-dimensional space (**k<<d**)
- Random projections can be represented by linear transformation matrix **R**
  - $\mathbf{y_i = R\ x_i}$
- What is the matrix **R**?

Finding matrix R

- Elements $\mathbf{R_{i,j}}$ can be Gaussian distributed
- Achlioptas* has shown that the Gaussian distribution can be replaced by

$$R(i,j) = \begin{cases} +1 \text{ with prob } \dfrac{1}{6} \\ 0 \text{ with prob } \dfrac{2}{3} \\ -1 \text{ with prob } \dfrac{1}{6} \end{cases}$$

- All zero mean, unit variance distributions for $\mathbf{R_{i,j}}$ would give a mapping that satisfies the Johnson-Lindenstrauss lemma