STAT 215A Fall 2017 Week 6

Rebecca Barter
09/29/2017

Lo(w)ess

Independent component analysis (ICA)

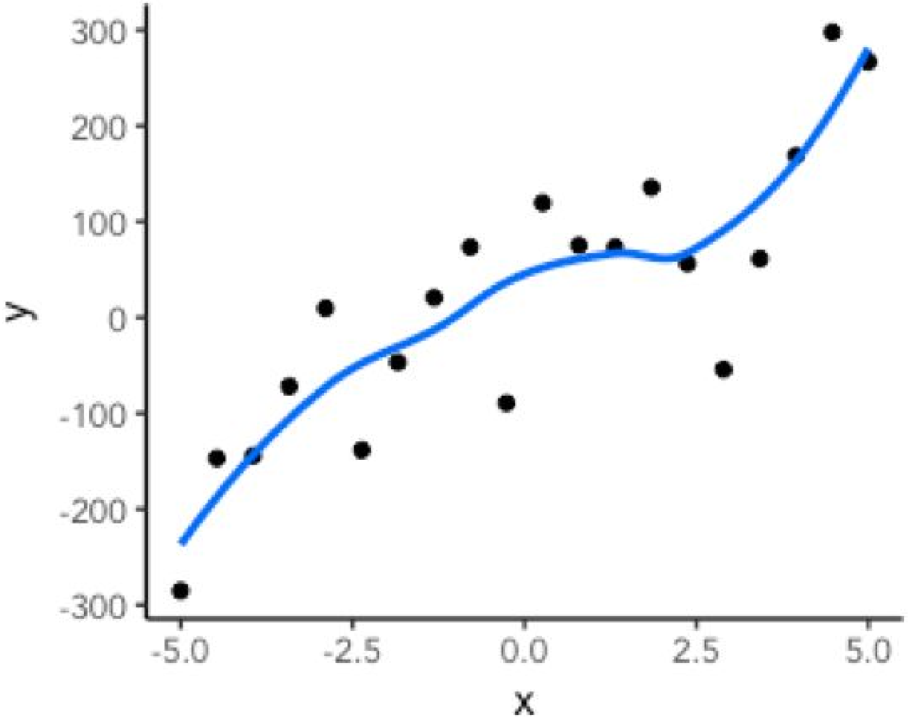
Multidimensional scaling

Making better figures

Loess

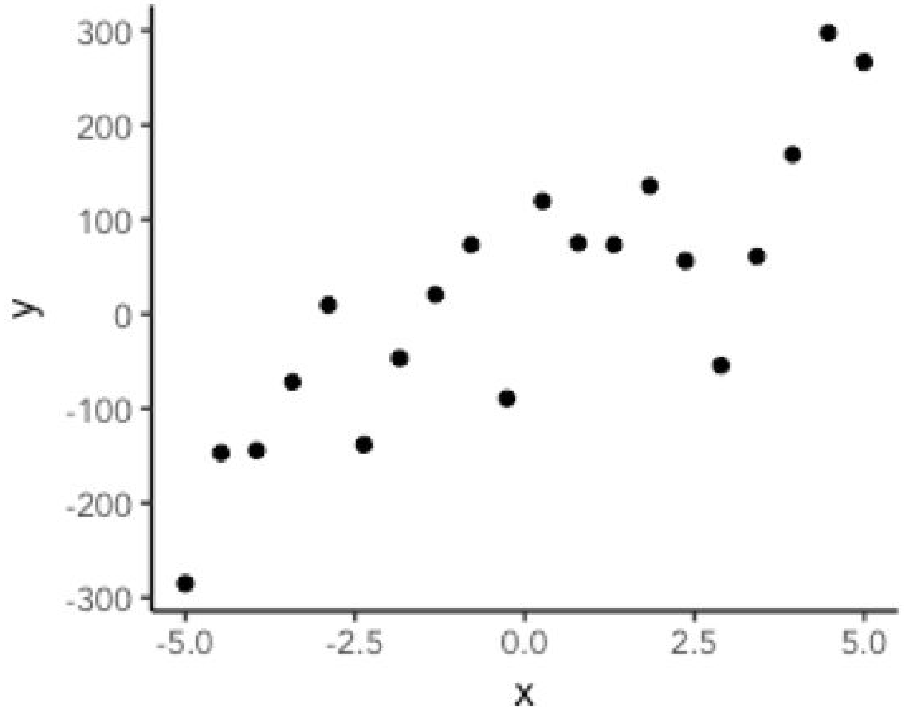
Lo(w)ess

Locally weighted scatterplot smoothing



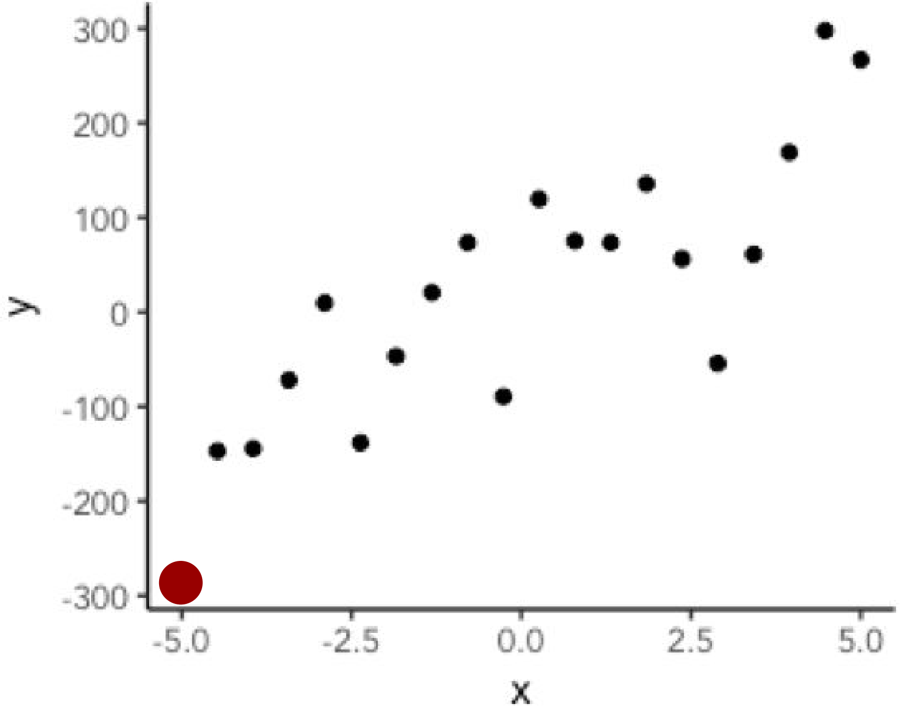
Lo(w)ess

Locally weighted scatterplot smoothing



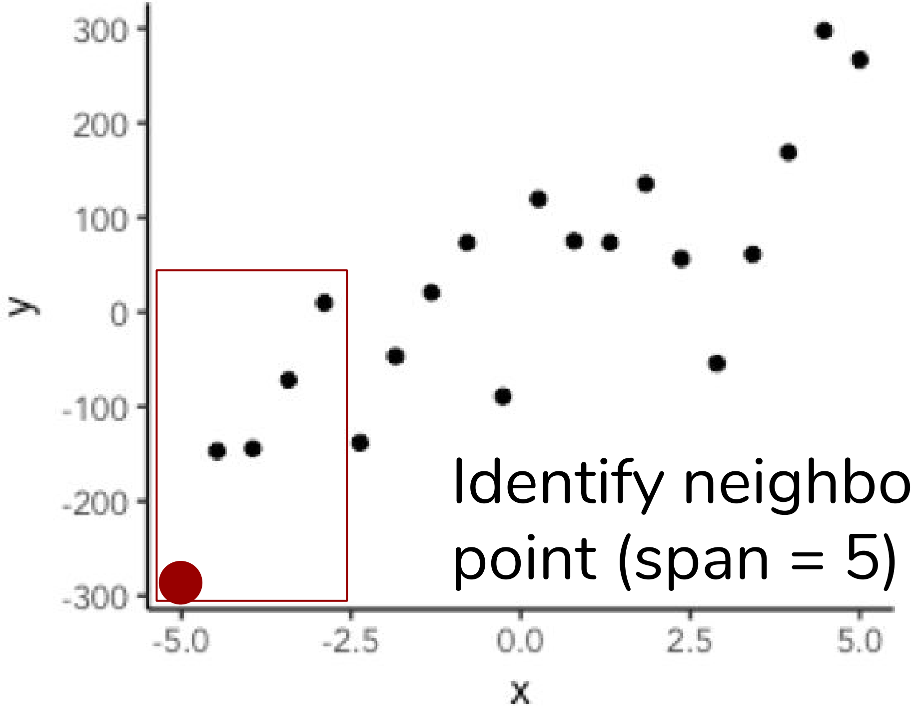
Lo(w)ess

Locally weighted scatterplot smoothing



Lo(w)ess

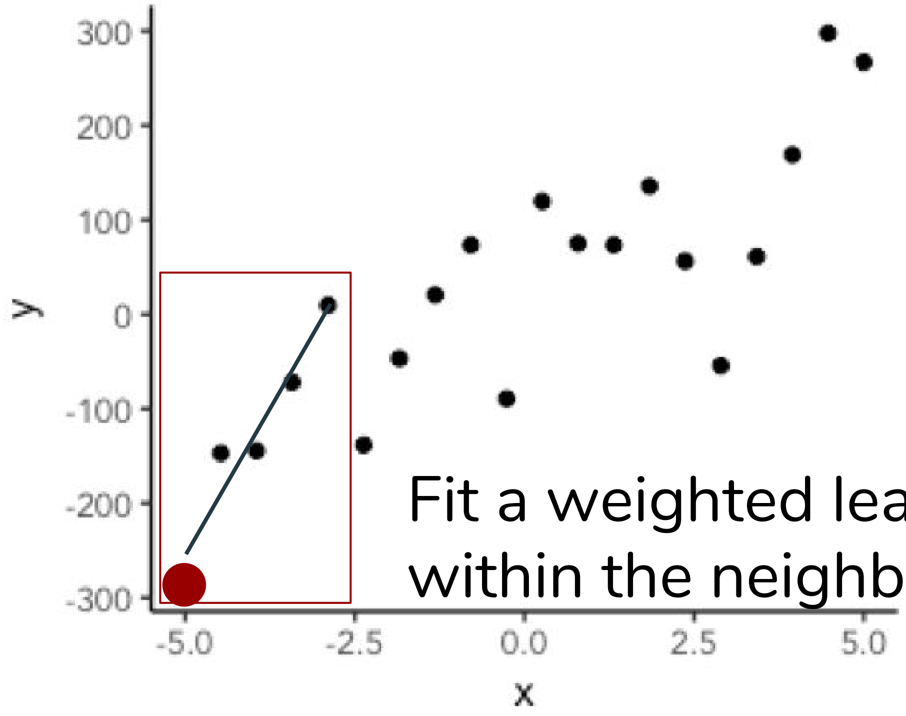
Locally weighted scatterplot smoothing



Identify neighborhood of first point (span = 5)

Lo(w)ess

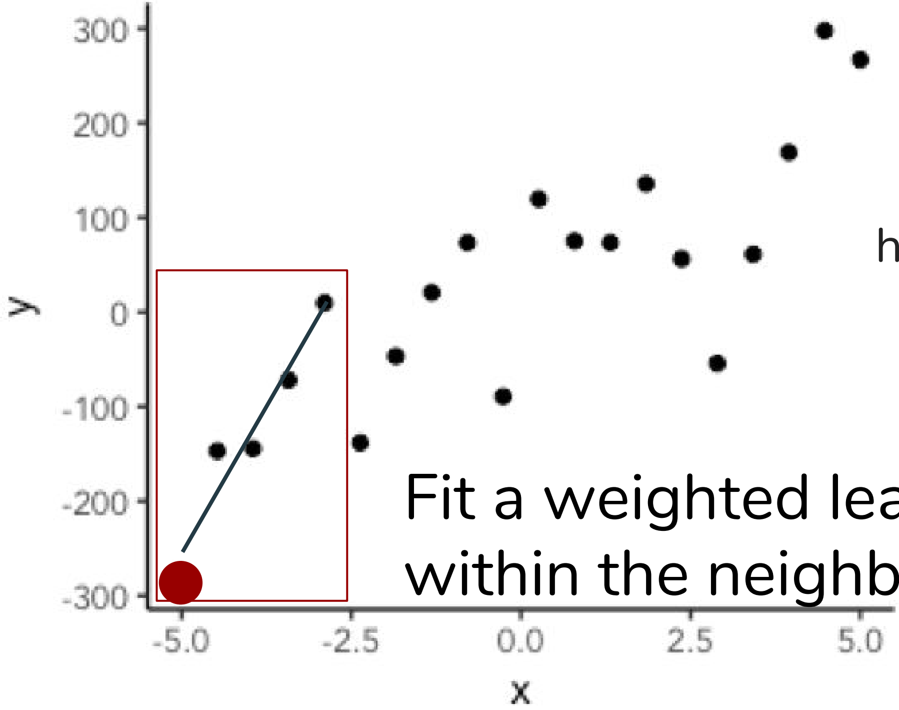
Locally weighted scatterplot smoothing



Fit a weighted least squares line within the neighborhood

Lo(w)ess

Locally weighted scatterplot smoothing

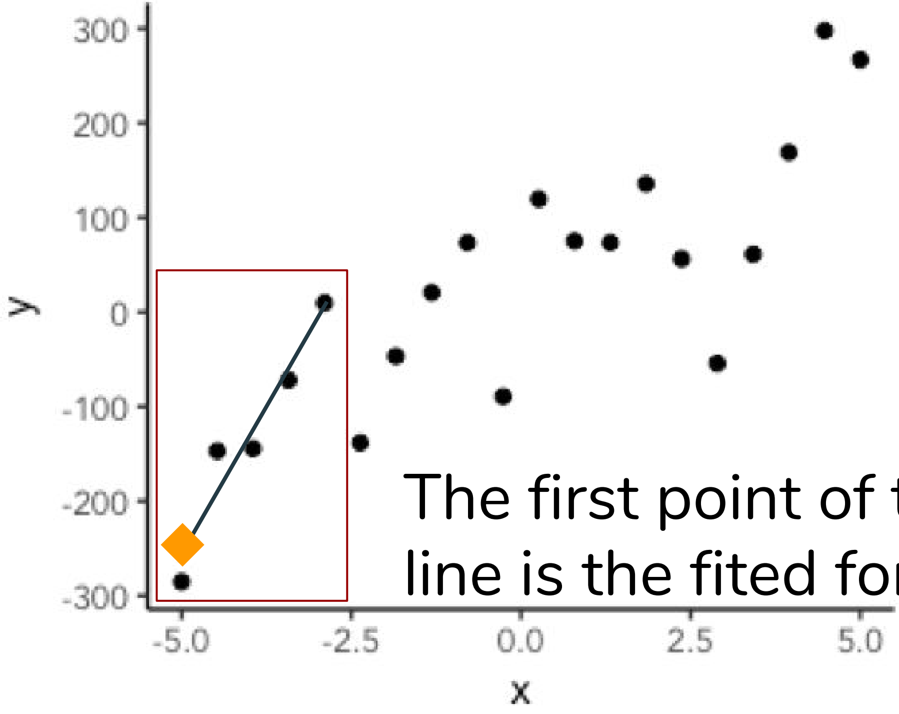


higher weight given to points near the focal point

Fit a weighted least squares line within the neighborhood

Lo(w)ess

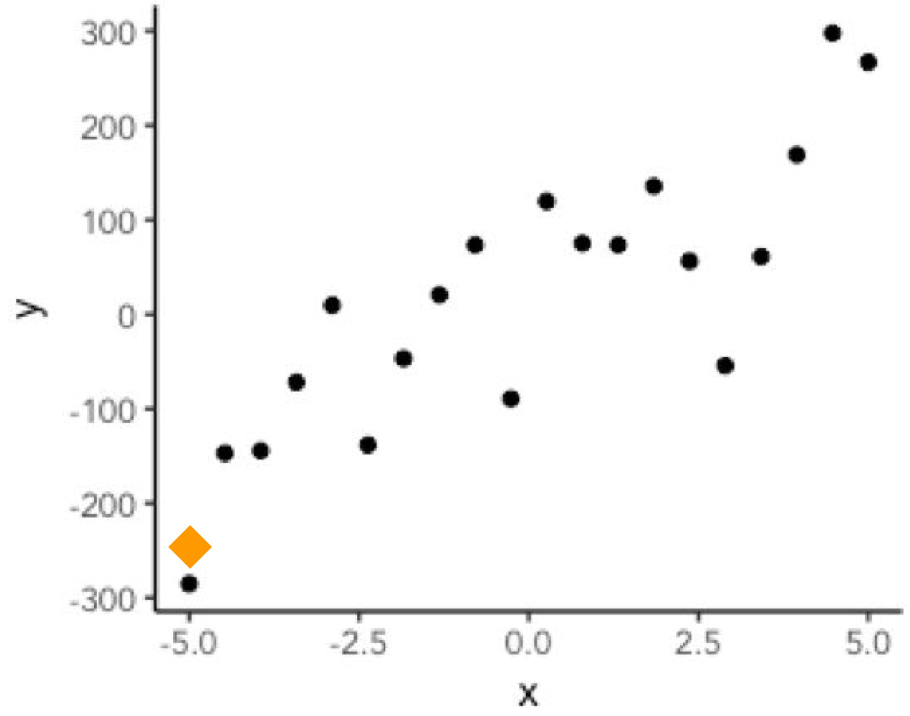
Locally weighted scatterplot smoothing



The first point of the regression line is the fited for the focal point

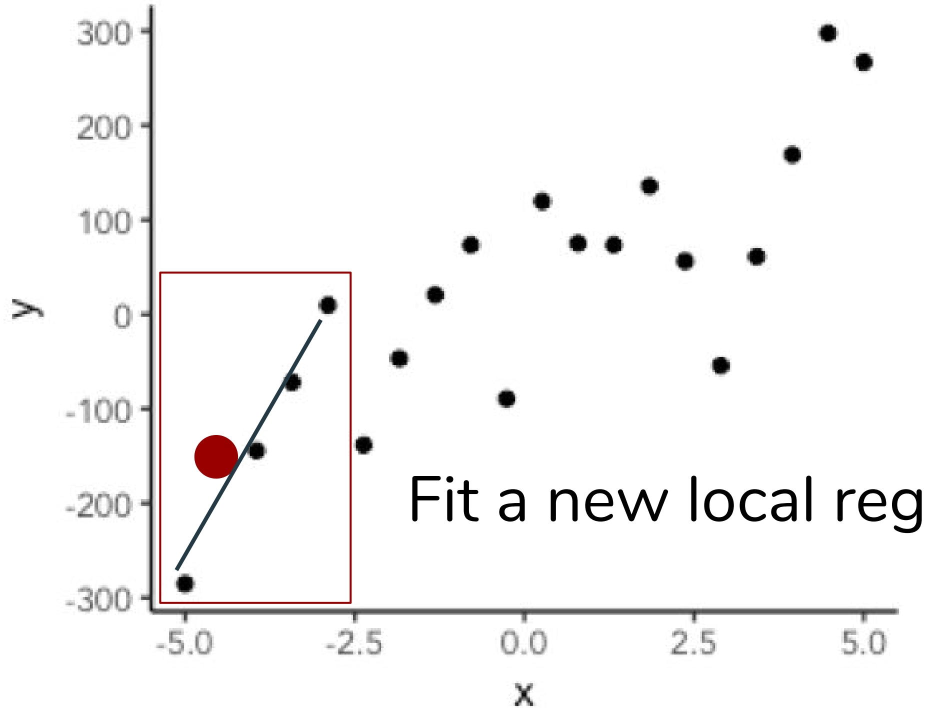
Lo(w)ess

Locally weighted scatterplot smoothing



Lo(w)ess

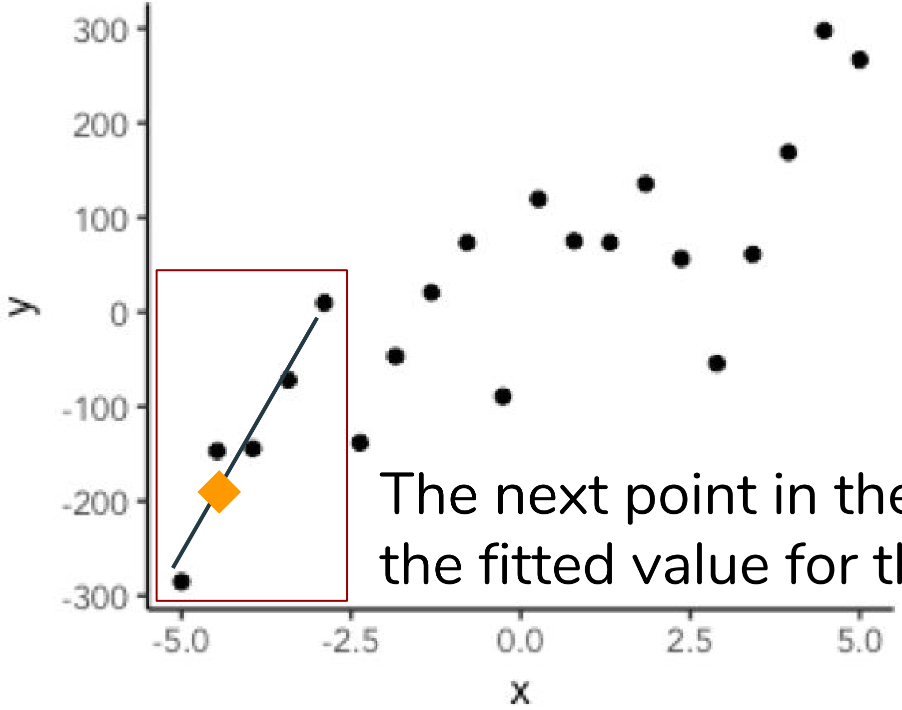
Locally weighted scatterplot smoothing



Fit a new local regression line

Lo(w)ess

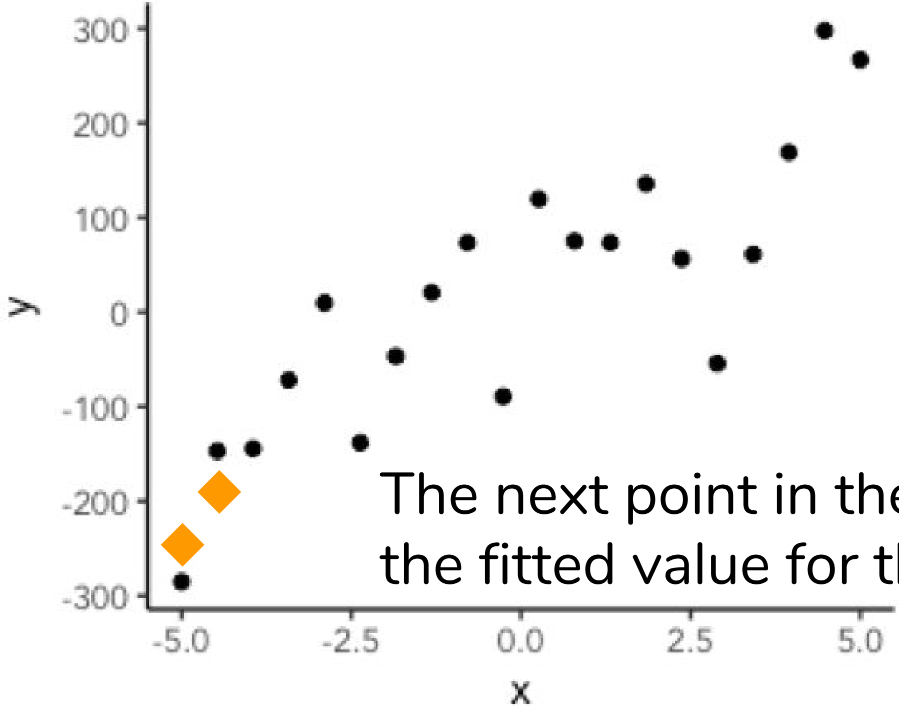
Locally weighted scatterplot smoothing



The next point in the loess curve is the fitted value for the focal point

Lo(w)ess

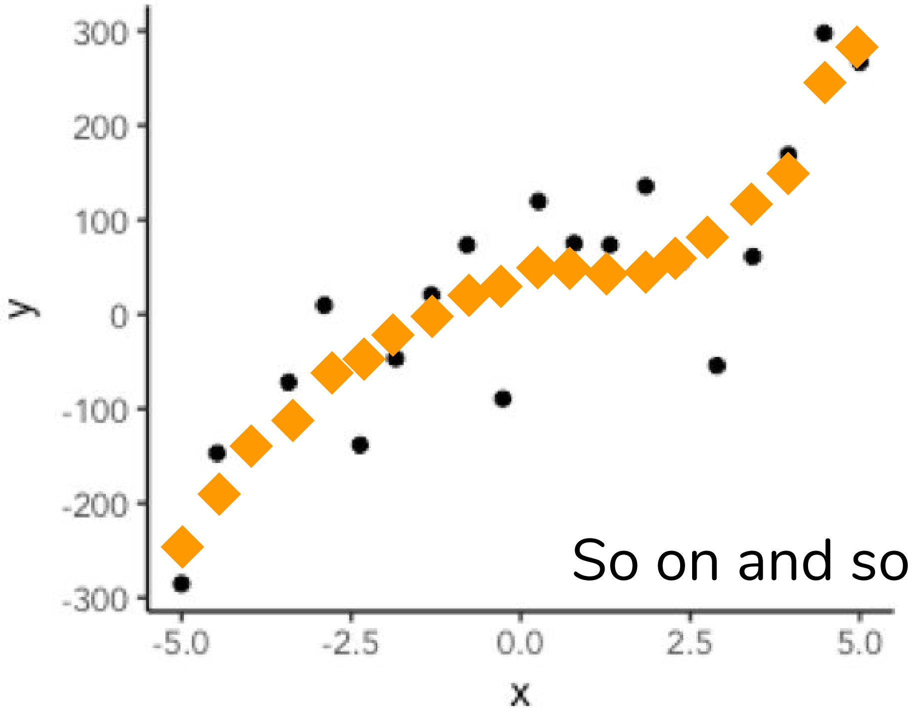
Locally weighted scatterplot smoothing



The next point in the loess curve is the fitted value for the focal point

Lo(w)ess

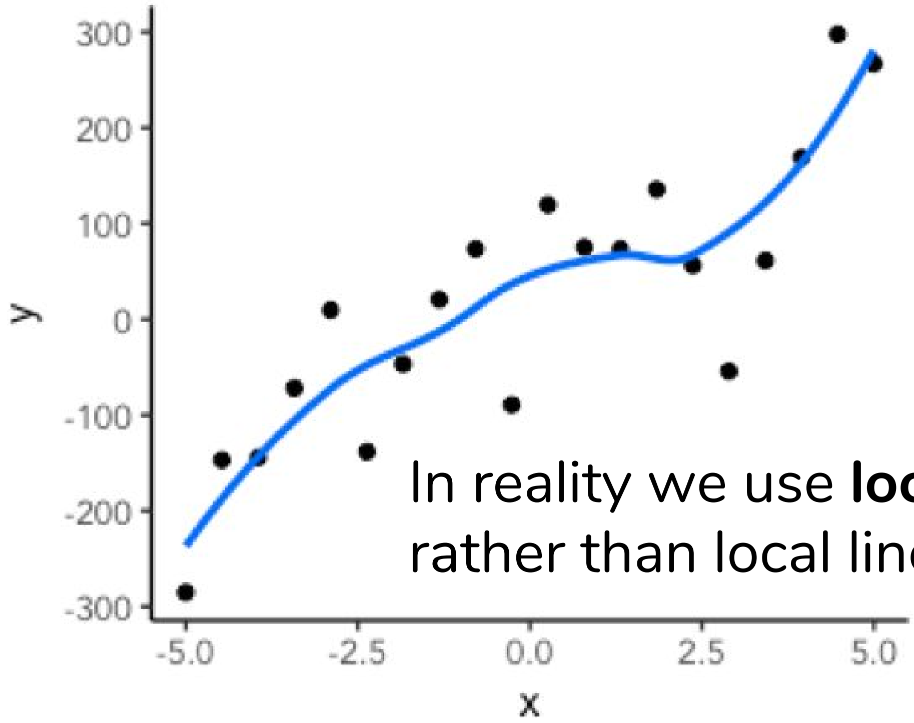
Locally weighted scatterplot smoothing



So on and so forth...

Lo(w)ess

Locally weighted scatterplot smoothing



In reality we use **local polynomial fits** rather than local linear fits

Lo(w)ess

Locally weighted scatterplot smoothing

Parameters to choose:

- The **span/bandwidth**: size of the neighborhood
- The **degree** of the local polynomial (in our example, the degree was 1)
- The **weights** for the weighted least squares

Independent Component Analysis

Independent Component Analysis

Goal: separate the underlying speech signals each corresponding to an individual at a cocktail party

Image source: http://tarynwilliford.com/wp-content/uploads/2010/12/RetroCocktailParty.jpg

Independent Component Analysis

Goal of PCA:

- **Compress the data** so that each dimension contains as much information as possible

Goal of ICA:

- **Identify the independent parts** that make up the data

Independent Component Analysis

Goal of PCA:

- Compress the data so that each dimension contains as much information as possible
- **Maximize variability** (second moment)

Goal of ICA:

- Identify the independent parts that make up the data
- **Maximize kurtosis** (fourth moment)

Independent Component Analysis

Goal of PCA:

- compress the data so that each dimension contains as much information as possible
- maximize variability (second moment)
- Each component is **orthogonal**

Goal of ICA:

- identify the independent parts that make up the data
- maximize kurtosis (fourth moment)
- Each component is **statistically independent**, but not necessarily orthogonal

Independent Component Analysis

Goal of PCA:

- compress the data so that each dimension contains as much information as possible
- maximize variability (second moment)
- Each component is orthogonal
- PCA **removes correlations**, but not higher order dependence

Goal of ICA:

- identify the independent parts that make up the data
- maximize kurtosis (fourth moment)
- Each component is statistically independent, but not necessarily orthogonal
- ICA removes correlations and **higher order dependence**

Independent Component Analysis

Goal of PCA:

- compress the data so that each dimension contains as much information as possible
- maximize variability (second moment)
- Each component is orthogonal
- PCA removes correlations, but not higher order dependence
- The **first few components** are the most important

Goal of ICA:

- identify the independent parts that make up the data
- maximize kurtosis (fourth moment)
- Each component is statistically independent, but not necessarily orthogonal
- ICA removes correlations and higher order dependence
- **All components** are equally important

Independent Component Analysis

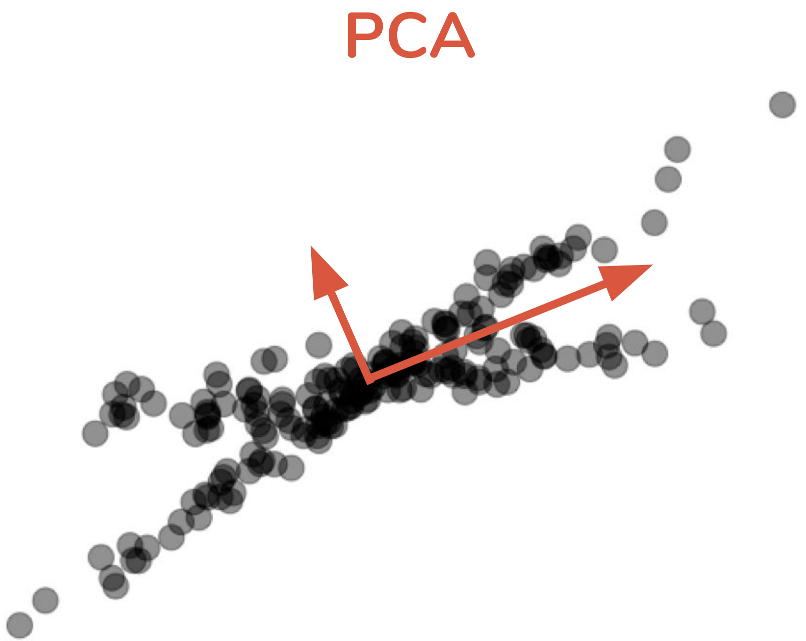
Goal of PCA:

- compress the data so that each dimension contains as much information as possible
- maximize variability (second moment)
- Each component is orthogonal
- PCA removes correlations, but not higher order dependence
- The **first few components** are the most important

Goal of ICA:

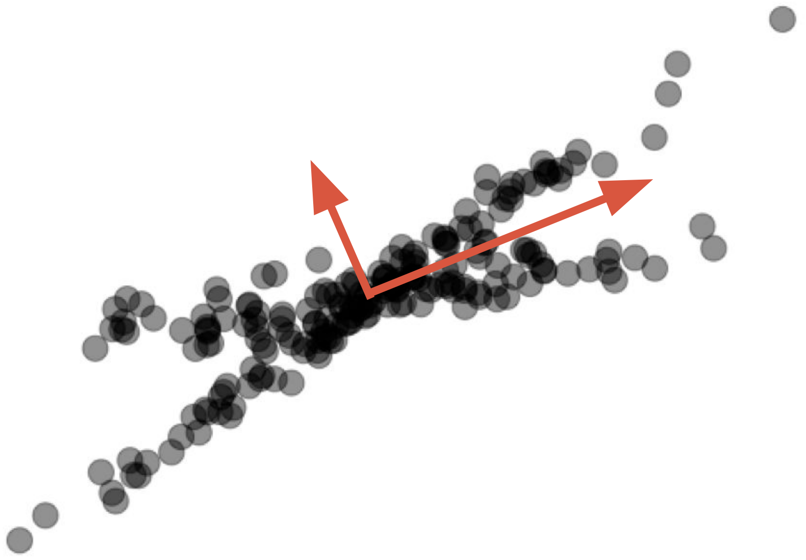
- identify the independent parts that make up the data
- maximize kurtosis (fourth moment)
- Each component is statistically independent, but not necessarily orthogonal
- ICA removes correlations and higher order dependence
- **All components** are equally important

Independent Component Analysis

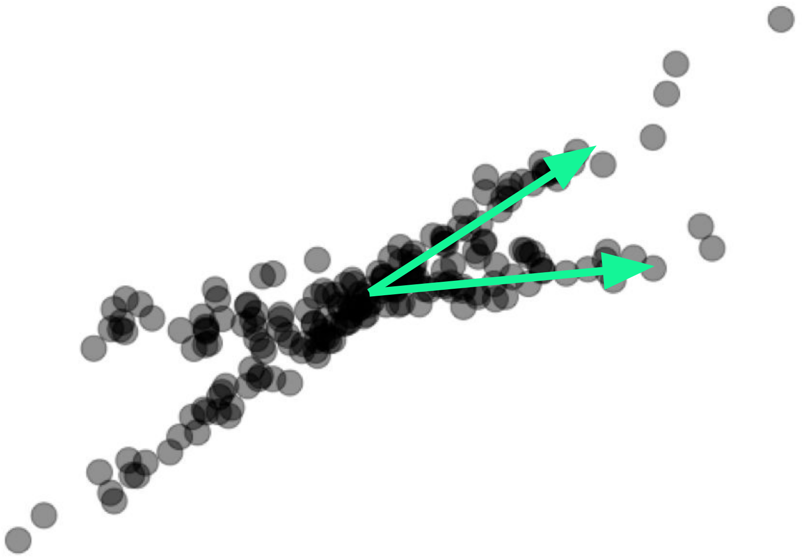


Independent Component Analysis

PCA



ICA



Multi Dimensional Scaling

Multidimensional scaling

Suppose you knew the **distances** between all major US cities

Could you figure out the coordinate locations of each city?

Given a distance matrix, MDS tries to recover low-dimensional coordinates such that the distances are preserved...

Multidimensional scaling (see mds.R)

X	Boston	NYC	DC	Miami	Chicago	Seattle	San.Francisco	Los.Angles	Denver
Boston	0	206	429	1504	963	2976	3095	2979	1949
NYC	206	0	223	1308	802	2815	2934	2786	1771
DC	429	223	0	1075	671	2684	2799	2631	1616
Miami	1504	1308	1075	0	1329	2373	3053	2687	2037
Chicago	963	802	671	1329	0	2013	2142	2054	996
Seattle	2976	2815	2684	2373	2013	0	808	1131	1307
San Francisco	3095	2934	2799	3053	2142	808	0	379	1235
Los Angles	2979	2786	2631	2687	2054	1131	379	0	1059
Denver	1949	1771	1616	2037	996	1307	1235	1059	0

Multidimensional scaling (see mds.R)

Multidimensional scaling

MDS moves objects around in the space defined (e.g. R^2) and check how well the distances between the objects can be reproduced by the new configuration.

It wants to minimize a goodness-of-fit measure called *stress*

$$stress = \sqrt{\frac{\sum \left(d_{ij} - \hat{d}_{ij}\right)^2}{\sum d_{ij}^{\,2}}}$$

Multidimensional scaling

Goal: to visualize (in low dimensions) the similarity between individual data points in a high dimensional dataset.

Input: a dissimilarity matrix

Output: a coordinate matrix whose configuration minimizes a loss function called *strain* (stress?)

Multidimensional scaling

Steps of a Classical MDS algorithm:

Classical MDS uses the fact that the coordinate matrix can be derived by eigenvalue decomposition from $B = XX'$. And the matrix B can be computed from proximity matrix D by using double centering.[2]

1. Set up the squared proximity matrix $D^{(2)} = [d_{ij}^2]$
2. Apply double centering: $B = -\frac{1}{2} J D^{(2)} J$ using the centering matrix $J = I - \frac{1}{n} 11'$, where n is the number of objects.
3. Determine the m largest eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ and corresponding eigenvectors $e_1, e_2, \ldots, e_m$ of B (where m is the number of dimensions desired for the output).
4. Now, $X = E_m \Lambda_m^{1/2}$, where E_m is the matrix of m eigenvectors and Λ_m is the diagonal matrix of m eigenvalues of B.

Classical MDS assumes Euclidean distances. So this is not applicable for direct dissimilarity ratings.

Multidimensional scaling

Double multiplication by this matrix simply subtracts the mean from each row and column

Steps of a Classical MDS algorithm:

Classical MDS uses the fact that the coordinate matrix can be derived by eigenvalue decomposition from $B = XX'$. And the matrix B can be computed from proximity matrix D by using double centering.[2]

1. Set up the squared proximity matrix $D^{(2)} = [d_{ij}^2]$
2. Apply double centering: $B = -\frac{1}{2} J D^{(2)} J$ using the centering matrix $J = I - \frac{1}{n} 11'$, where n is the number of objects.
3. Determine the m largest eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ and corresponding eigenvectors $e_1, e_2, \ldots, e_m$ of B (where m is the number of dimensions desired for the output).
4. Now, $X = E_m \Lambda_m^{1/2}$, where E_m is the matrix of m eigenvectors and Λ_m is the diagonal matrix of m eigenvalues of B.

Classical MDS assumes Euclidean distances. So this is not applicable for direct dissimilarity ratings.