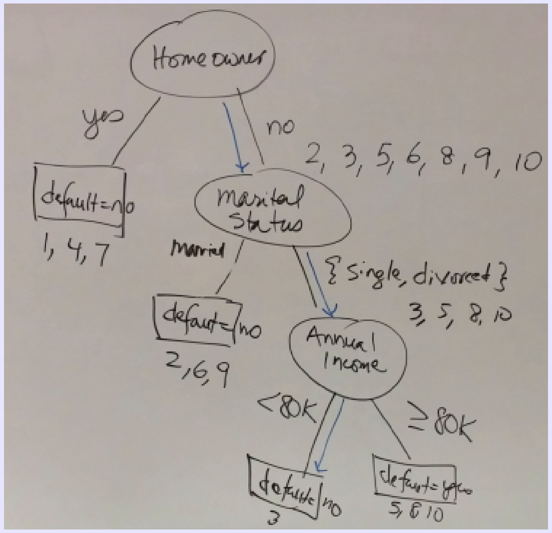
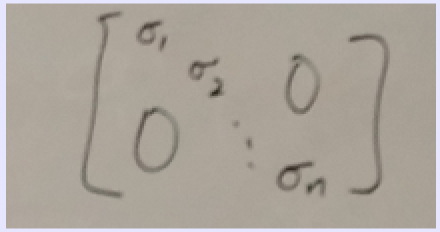
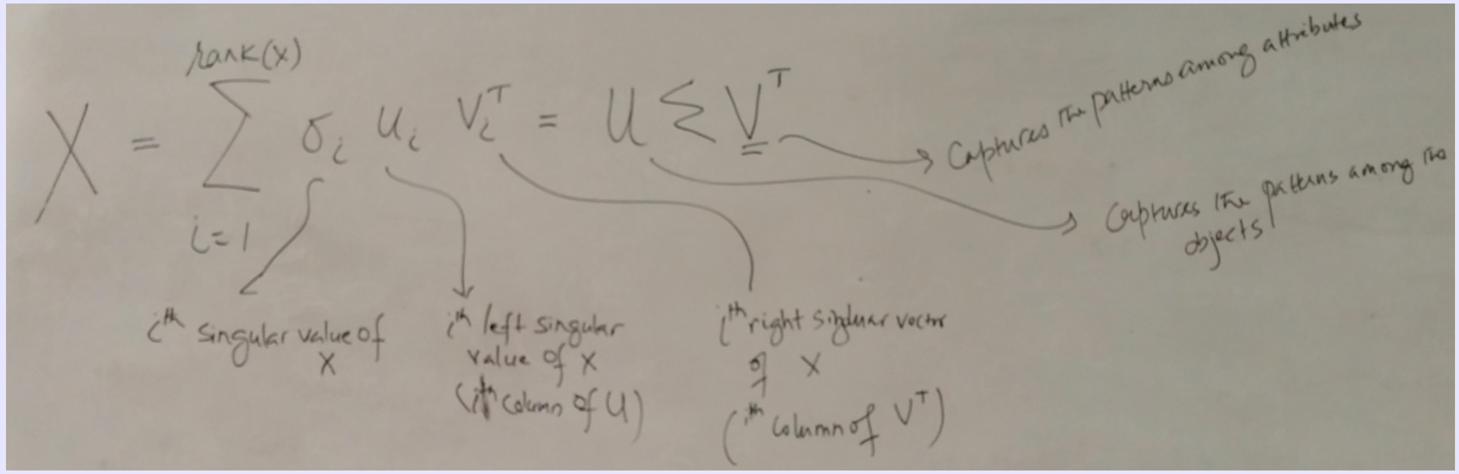
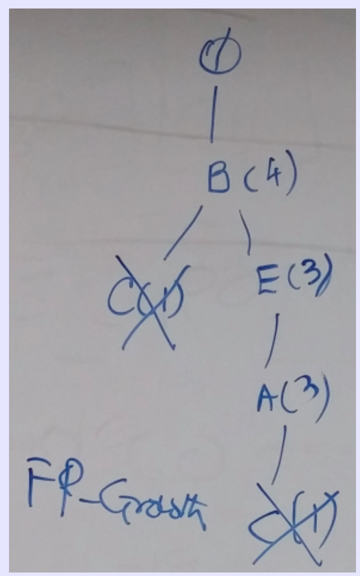
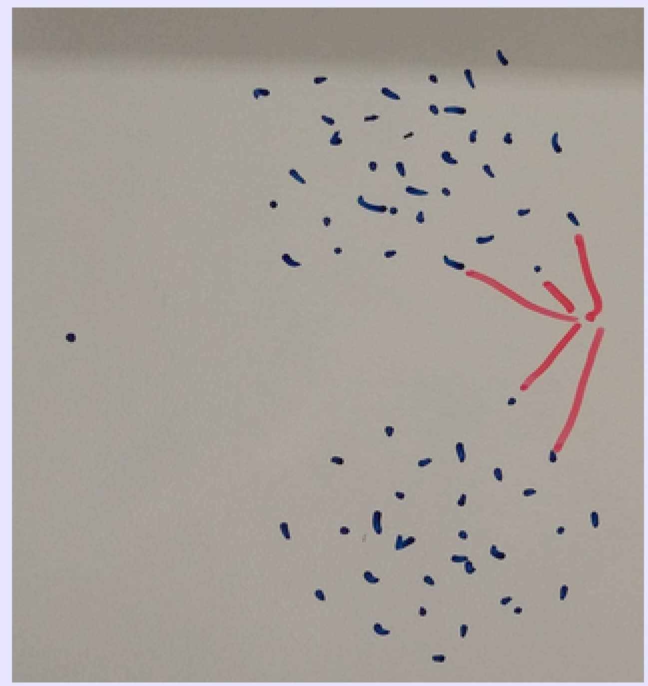










CS 422: Data Mining
Vijay K. Gurbani, Ph.D.,
Illinois Institute of Technology

Distance measures
Data transformation:
Standardization and scaling
Binarization and Discretization

Distance measures

- Given points in a *n*-dimension space, how do we compute the distance between two points?
  - Why? We may want to know whether two points are clustered close together for some purpose.
- Distance measures:
  - Euclidean
  - Manhattan
  - Minkowski (general)
  - Mahalanobis distance

Distance measures: Manhattan

- Also known as
  - “taxi cab” distance.
  - The L_1 norm.

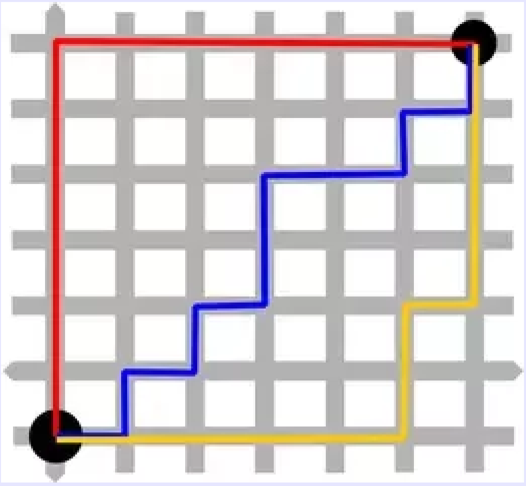
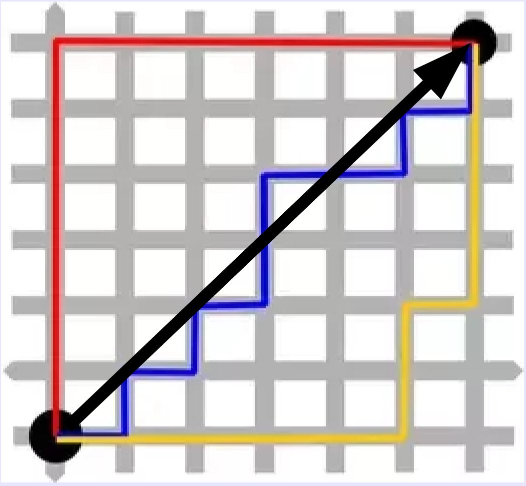


Image source: https://qph.fs.quoracdn.net/main-qimg-8d64c8344fc8364e46b9712e2c51dca4

Distance measures: Euclidean

- Also known as
  - The L_2 norm.

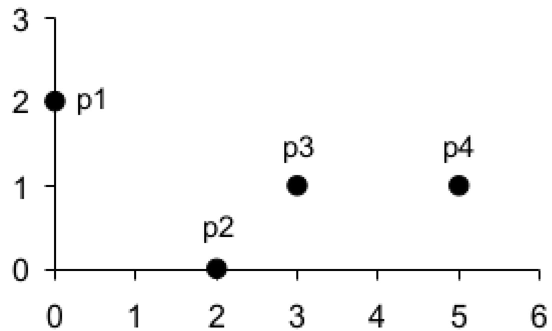


  - Defined as:

$$d(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{k=1}^{n} (x_k - y_k)^2}$$

where n is the number of dimensions (attributes) and x_k and y_k are, respectively, the k^{th} attributes (components) or data objects $\mathbf{x}$ and $\mathbf{y}$.

Distance measures: Euclidean



point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

Distance Matrix

Distance measures: Minkowski

- Minkowski distance is a generalization.

$$d(\mathbf{x}, \mathbf{y}) = \left(\sum_{k=1}^{n} |x_k - y_k|^r \right)^{1/r}$$

Where r is a parameter, n is the number of dimensions (attributes) and x_k and y_k are, respectively, the k^{th} attributes (components) or data objects $\boldsymbol{x}$ and $\boldsymbol{y}$.

  - If r = 1, degenerates to Manhattan distance.
  - If r = 2, degenerates to Euclidean distance.
  - If r = ∞, degenerates to Supremum distance (L_{max}).
- Do not confuse *r* with *n*!

Distance measures: Minkowski

point	x	y
p1	0	2
p2	2	0
p3	3	1
p4	5	1

L1	p1	p2	p3	p4
p1	0	4	4	6
p2	4	0	2	4
p3	4	2	0	2
p4	6	4	2	0

L2	p1	p2	p3	p4
p1	0	2.828	3.162	5.099
p2	2.828	0	1.414	3.162
p3	3.162	1.414	0	2
p4	5.099	3.162	2	0

L_∞	p1	p2	p3	p4
p1	0	2	3	5
p2	2	0	1	3
p3	3	1	0	2
p4	5	3	2	0

Distance Matrix

Code: dist.r

Distance measures: Mahalanobis

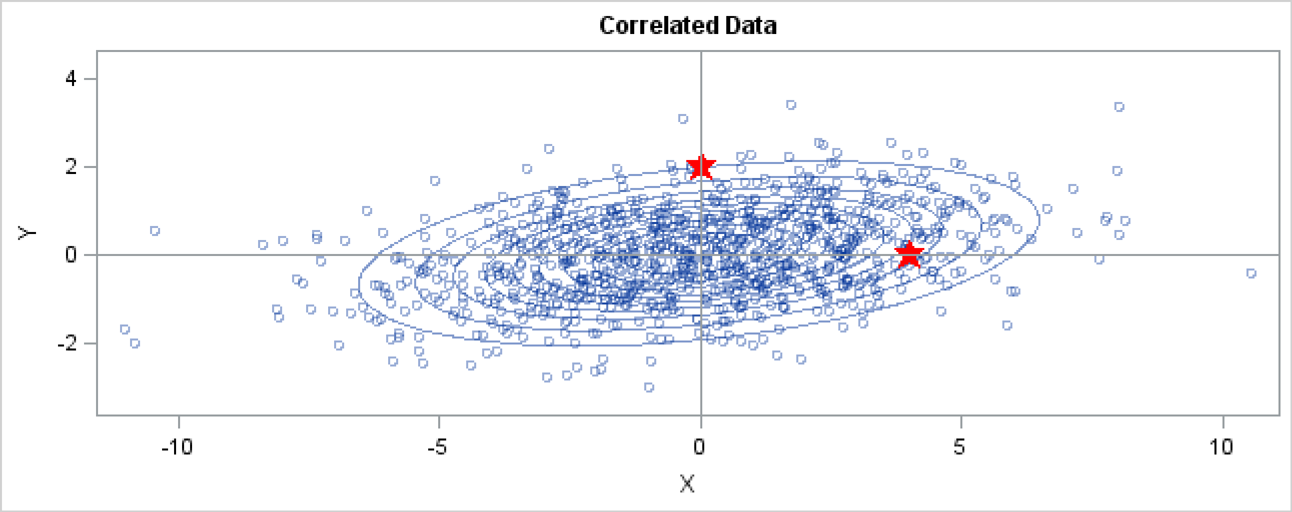


Image Source: https://blogs.sas.com/content/iml/files/2012/02/mahal.png

- Which of the red stars is closer to the origin?
- Pedagogically, it is the distance between a point and a distribution.

Data Transformation

- Example: Binarization (Tan, Ch. 2)
  - Also called “one-hot encoding”

Table 2.5. Conversion of a categorical attribute to three binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3
awful	0	0	0	0
poor	1	0	0	1
OK	2	0	1	0
good	3	0	1	1
great	4	1	0	0

Table 2.6. Conversion of a categorical attribute to five asymmetric binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3	x_4	x_5
awful	0	1	0	0	0	0
poor	1	0	1	0	0	0
OK	2	0	0	1	0	0
good	3	0	0	0	1	0
great	4	0	0	0	0	1

Data Transformation

- Example: Binarization (Tan, Ch. 2)
  - Also called “one-hot encoding”

Table 2.5. Conversion of a categorical attribute to three binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3
awful	0	0	0	0
poor	1	0	0	1
OK	2	0	1	0
good	3	0	1	1
great	4	1	0	0

Table 2.6. Conversion of a categorical attribute to five asymmetric binary attributes.

Categorical Value	Integer Value	x_1	x_2	x_3	x_4	x_5
awful	0	1	0	0	0	0
poor	1	0	1	0	0	0
OK	2	0	0	1	0	0
good	3	0	0	0	1	0
great	4	0	0	0	0	1

Data Transformation

- Example: Discretization

  {10, 20, 30, 1, 22, 25, 2, 18, 15}

- Step 1: Sort:

  {1, 2, 10, 15, 18, 20, 22, 25, 30}

- Step 2: Create split points

  {1, 2, 10, 15, 18, 20, 22, 25, 30}

- Step 3: Map split values to discrete categorical variables; e.g.: {1, 2, 10} → “Small”, ...

Standardization and scaling

- Standardization vs. normalization:
  - Standardization: Transforms data with mean = 0 and std. dev = 1. (Z-score.)
  - Normalization: Scales a variable to have values between 0 and 1.
  - These terms are often used interchangably.

Standardization and scaling

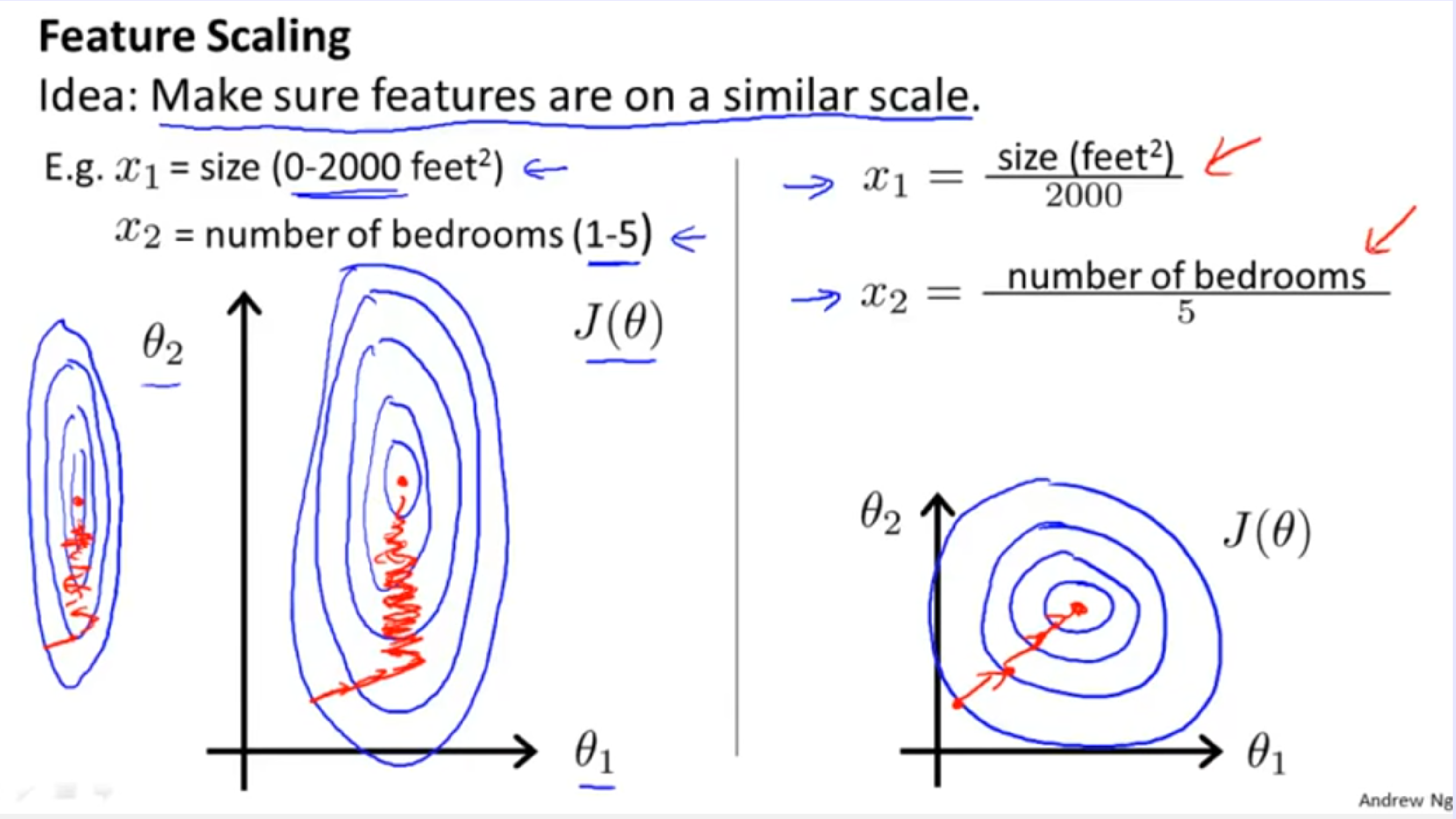
- Caution: when computing distances, you want to standardize if the scales differ significantly.
- E.g.: multiple features, each varying in units:
  - Age: [0-110] years.
  - Height: [18-107] in.
  - Weight: [7.5-400] lbs.
  - Head circum.: [13.5-58.4] in.

Image source: https://cdn-images-1.medium.com/max/800/1*EyPd0sQxEXtTDSJgu72JNQ.jpeg

Standardization and scaling

- What happens if we don’t scale.
- Nothing drastic, but some algorithms will be slow in converging, especially if they use Euclidean distances.
  - These algorithms will only take in the magnitude of features while ignoring the units.
  - Features with high magnitudes will dominate the distance calculations.

Standardization and scaling



Graphic source: Andrew Ng

Standardization and scaling

- Standardization is essentially “feature scaling”.
- Feature scaling strategies:
  - Replace values by *z*-score.

    $$x' = \frac{x - \bar{x}}{\sigma}$$

    Redistributes features with μ = 0, σ = 1.
  - Mean normalization.
  - Min-max scaling.
  - Unit vector.

Standardization and scaling

- Standardization is essentially “feature scaling”.
- Feature scaling strategies:
  - Replace values by *z*-score.

    $$x' = \frac{x - \bar{x}}{\sigma}$$ Redistributes features with μ = 0, σ = 1.
  - Mean normalization.

    $$x' = \frac{x - \text{mean}(x)}{\max(x) - \min(x)}$$ Redistributes with range [-1,1], μ = 0.
  - Min-max scaling.
  - Unit vector.

Standardization and scaling

- Standardization is essentially “feature scaling”.
- Feature scaling strategies:
  - Replace values by *z*-score.

    $$x' = \frac{x - \bar{x}}{\sigma}$$ Redistributes features with μ = 0, σ = 1.

  - Mean normalization.

    $$x' = \frac{x - \text{mean}(x)}{\max(x) - \min(x)}$$ Redistributes with range [-1,1], μ = 0.

  - Min-max scaling.

    $$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$ Redistributes with range [0,1].

  - Unit vector.

Standardization and scaling

- Standardization is essentially "feature scaling".
- Feature scaling strategies:
  - Replace values by *z*-score (standardization).

    $$x' = \frac{x - \bar{x}}{\sigma}$$

    Redistributes features with μ = 0, σ = 1.
  - Mean normalization.

    $$x' = \frac{x - \text{mean}(x)}{\max(x) - \min(x)}$$

    Redistributes with range [-1,1], μ = 0.
  - Min-max scaling.

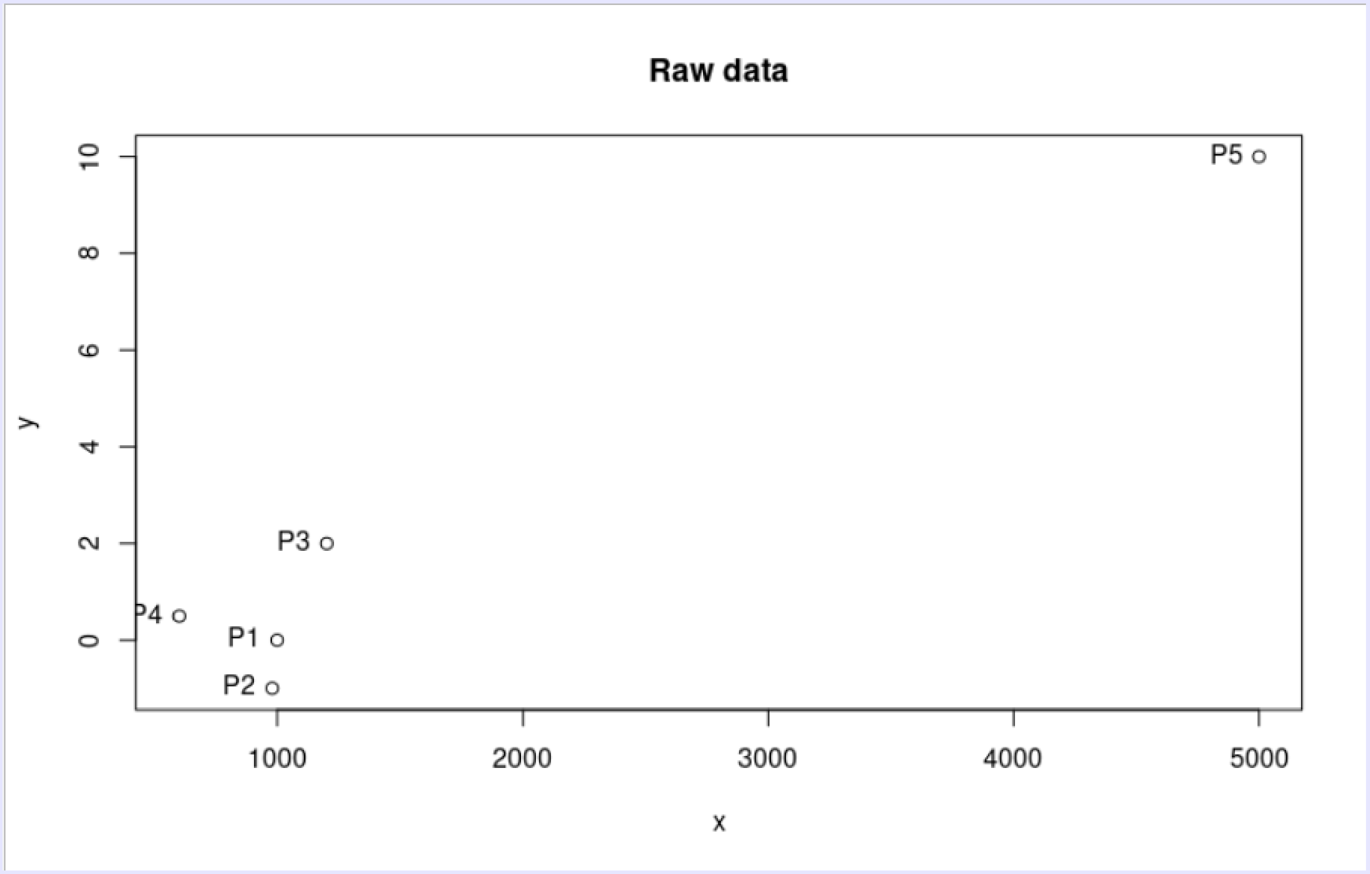
    $$x' = \frac{x - \min(x)}{\max(x) - \min(x)}$$

    Redistributes with range [0,1].
  - Unit vector.

    $$x' = \frac{x}{||x||}$$

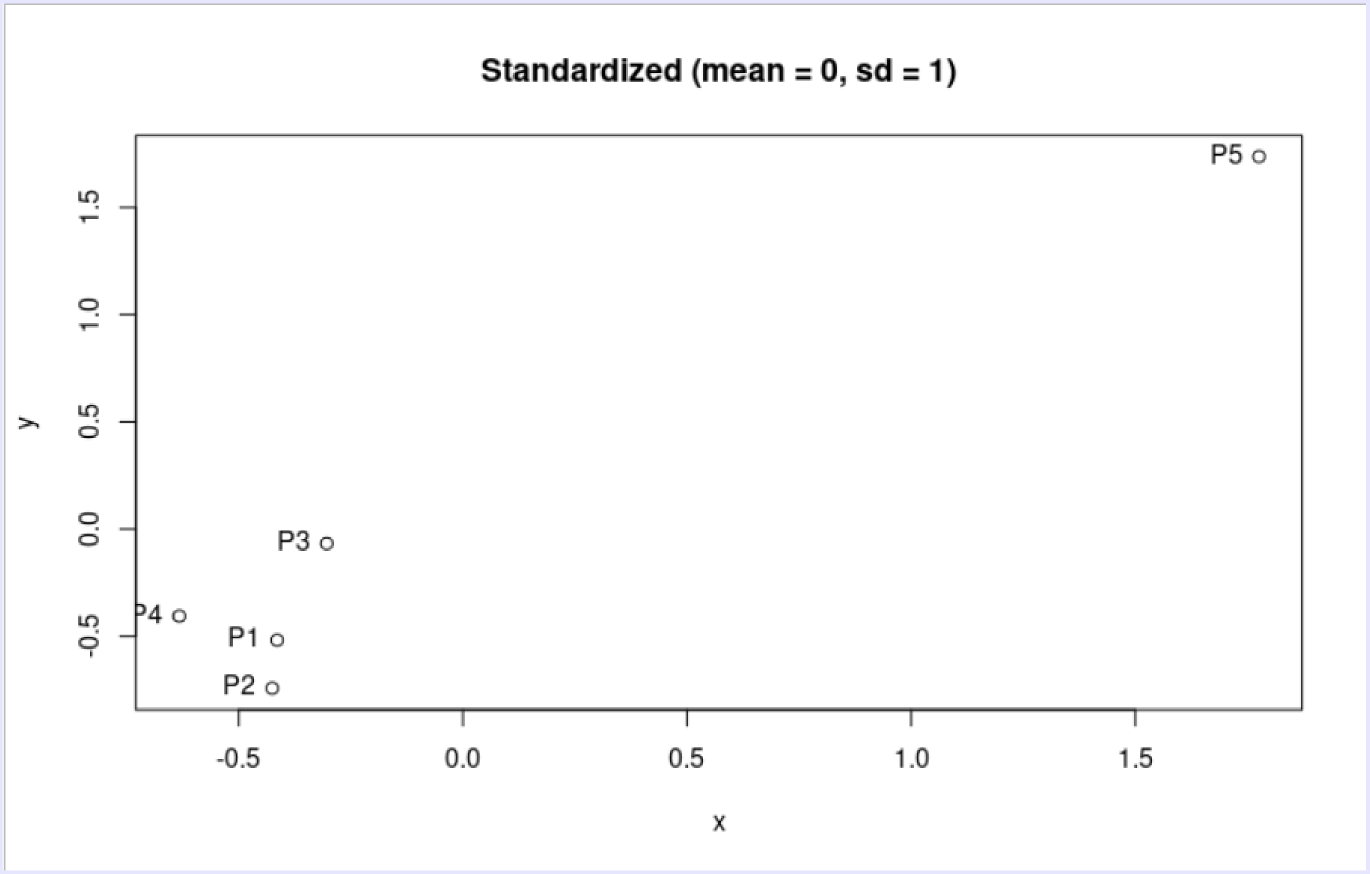
Code: scaling.r

Standardization and scaling



P1(1000, 0), P3(1201, 2), P4(600, 0.5)
D(P1, P3) > D(P1, P4)?
    201 not > 400

Standardization and scaling



P1(1000, 0), P3(1201, 2), P4(600, 0.5)
D(P1, P3) > D(P1, P4)?
Standardized D(P1,P3) > D(P1,P4)
0.464 > 0.246

Standardization and scaling

- When do we scale, and when do we not scale?
  - Scale if all attributes are numeric and using neural networks.
  - If some attributes are nominal or categorical, one-hot encode them.
    - Is one-hot encoding enough or should we scale?
  - Should we scale when using decision trees?
- Scale the dataset and then split? Or split and then scale?