

COMP7990

Exam Details

- Date: 9-12-2022
- Time: 19:00 - 22:00
- Venue: SHSH
- Format: closed-book. No note, no paper, no equations are given.
- **Calculator is allowed.**
- **Steps carry marks.**
 - Except when we give you a table to fill the value.
- **Rulers, Eraser, Pen, Pencils, Calculator are useful**



Exam Content

- Part A (70%)
 - Statistic (~30%)
 - Visualization (~10%)
 - Data Mining (~30%)
- Part B (30%)
 - Database (~20%)
 - Security and Privacy (~10%)

Tips

- Don't be late. No enter after 30 minutes.
- Sleep well before the exam
- Have just enough food before the exam
- Remember the algorithms/computational procedures
- Do revision on your quiz again
- Calculator list
<https://www.hkeaa.edu.hk/DocLibrary/IPE/cal/CAL2019.pdf>
- For calculator that is not from the list, you should consult the lecturer in advanced.

Included and Not Included

- Exam covers topics in the lecture notes only. Labs are excluded.
- Notes stated with [Option] / [Extra] are excluded.

- e.g. SVM: Solving the Optimization Problem (option)

- The optimization problem is

$$\begin{aligned} \min & \frac{\|\mathbf{w}\|^2}{2} \\ \text{subject to } & y_i(\mathbf{w}^T \mathbf{x}_i + b) \geq 1, i = 1, \dots, m \end{aligned}$$

- Introducing Lagrange Multipliers α_i , one for each constraint, leads to the primal Lagrangian:

$$\begin{aligned} \min L_p &= \frac{\|\mathbf{w}\|^2}{2} + \sum_{i=1}^m \alpha_i (1 - y_i(\mathbf{w}^T \mathbf{x}_i + b)) \\ \text{subject to } & \alpha_i \geq 0, i = 1, \dots, m \end{aligned}$$

- Equation on [formula sheet.pdf](#) needs to remember. If equations needed are not on the formula sheet, it will be provided to you in the exam paper.

Kernel as high dimensional feature mapping (optional)

- Consider two samples $\mathbf{x}: [x_1, x_2]$ and $\mathbf{z}: [z_1, z_2]$
- Let us assume there is a kernel function k that takes inputs \mathbf{x} and \mathbf{z}

$$\begin{aligned} k(\mathbf{x}, \mathbf{z}) &= (\mathbf{x}\mathbf{z})^2 \\ &= (x_1z_1 + x_2z_2)^2 \\ &= x_1^2z_1^2 + x_2^2z_2^2 + 2x_1x_2z_1z_2 \\ &= (x_1^2, \sqrt{2}x_1x_2, x_2^2)(z_1^2, \sqrt{2}z_1z_2, z_2^2) \\ &= \phi(\mathbf{x})\phi(\mathbf{z}) \end{aligned}$$

$\phi(\mathbf{x})\phi(\mathbf{z})$ is computed efficiently in original input space.

- This kernel function k implicitly defines a mapping ϕ to a higher dimensional space
- $\phi(\mathbf{x}) = [x_1^2, \sqrt{2}x_1x_2, x_2^2]$
- Note that we do not have to define/compute this mapping. Simple defining the kernel is a certain way to give a higher dimensional mapping ϕ

The background of the slide features a dynamic, abstract design composed of several overlapping circles in various colors. These colors include shades of red, teal, green, yellow, and orange. The circles overlap in a way that creates a sense of depth and movement across the entire frame.

Review

Review for data mining

- How to Handle Noisy Data
- Normalization
 - min-max Normalization
 - Decimal Scaling
- Supervised and Unsupervised Learning
- Classification and Regression
- Linear model (one variable and multiple variables)
- Perceptron
- Multi-Layer Perceptron
- SVM
- KNN
- Clustering

How to Handle Noisy Data

- **Binning**
 - First sort data and partition into bins
 - Smooth by bin mean/median/boundaries
- Regression
 - Smooth by fitting the data into regression functions
- Clustering
 - Detect and remove outliers

Simple Discretization Methods: Binning

- Equal-width (distance) Partitioning
 - Divides the range into N intervals of equal size: uniform grid
 - Suppose \min and \max are the lowest and highest values of the attribute, the width of intervals should be: $w = (\max - \min)/N$
 - The most straight-forward method
 - Outliers may dominate presentation
 - Skewed data is not handled well
- Equal-depth (frequency) partitioning
 - Divides the range into N intervals, each containing approximately same the number of samples
 - Skewed data is also handled well

Normalization

- For distance-based methods, normalization helps to prevent that attributes with large ranges out-weight attributes with small ranges
- Scale the attribute values to a small specified range
- Normalization Methods
 - Normalization by decimal scaling
 - Min-Max normalization (normalized by range)

Normalization: min-max Normalization

- Min-max normalization
 - Performs a linear transformation on the original data.
 - Suppose \min, \max are the minimum and maximum values of an attribute and we want to normalize the attribute value to $[\min_{new}, \max_{new}]$, min-max normalization maps a value x_i to x'_i by

$$x'_i = \frac{(x_i - \min)}{\max - \min} (\max_{new} - \min_{new}) + \min_{new}$$

- E.g., suppose that the minimum and maximum values for the feature income are \$12,000 and \$98,000. We would like to map income to the range [0.0, 1.0]. By min-max normalization, what is the mapped value for \$73,600?

$$\frac{(73,600 - 12,000)}{98,000 - 12,000} (1.0 - 0.0) + 0.0 = 0.716$$

Normalization: Decimal Scaling

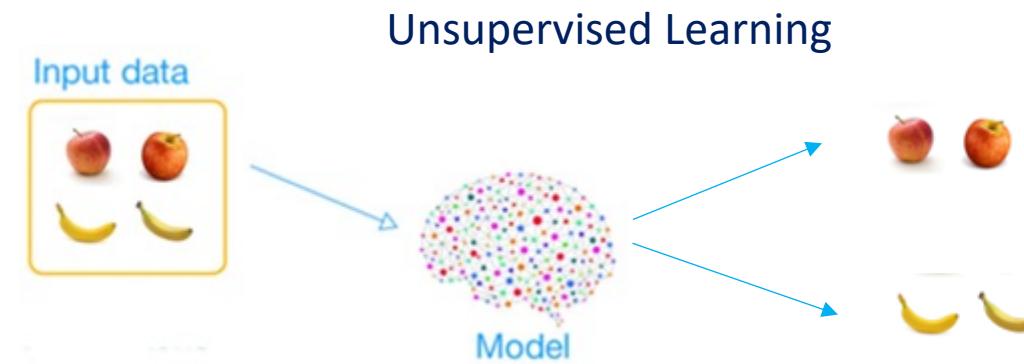
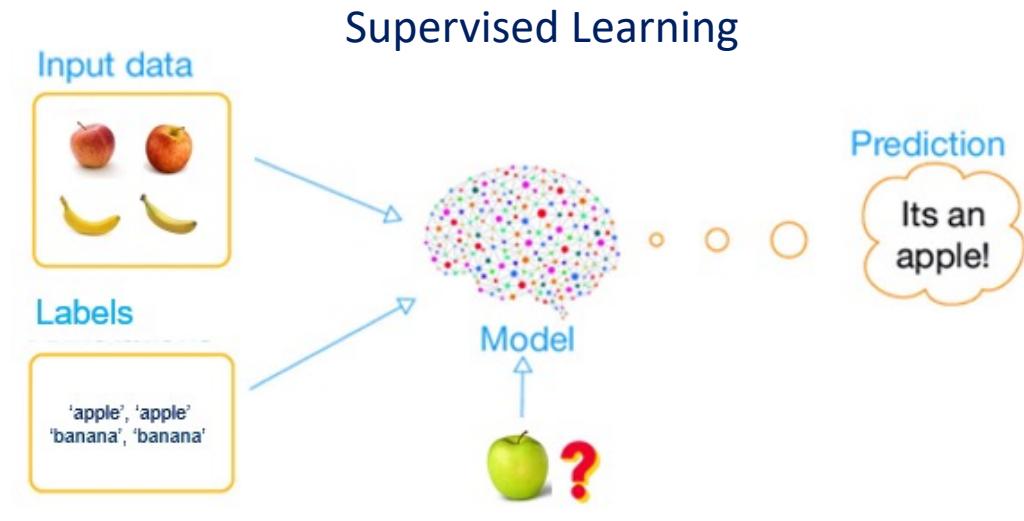
- Decimal Scaling
 - The values of an attribute are normalized by moving the decimal point.
 - The number of decimal points moved depends on the maximum absolute value of the attribute.
 - Decimal scaling maps a value x_i to x'_i by

$$x'_i = \frac{x_i}{10^j} \quad \text{--->} \quad j \text{ is the smallest integer such that } \max(|x_i|) < 1$$

- E.g., suppose that the recorded values of an attribute range from -986 to 917. The maximum absolute value is 986. To normalize by decimal scaling, we therefore divide each value by 1,000 (i.e., $j = 3$) so that -986 normalizes to -0.986 and 917 normalizes to 0.917.

Different Types of Learning Tasks

- Supervised Learning
 - Data with labels
- Unsupervised Learning
 - Data without labels



An Example of Classification Problem

- Learn to recognize apple or banana

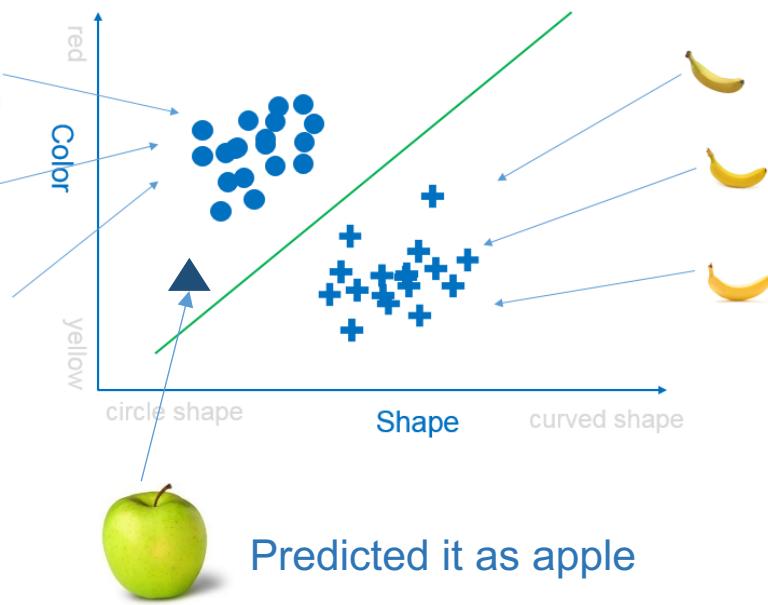
Training Data with Class Label

Label	
apple	
apple	
banana	
banana	
...	...

Feature Engineering →

color(x_1)	shape (x_2)	y
0.8	0.1	1
0.75	0.12	1
0.12	0.6	0
0.2	0.8	0
...
...

Model Building →

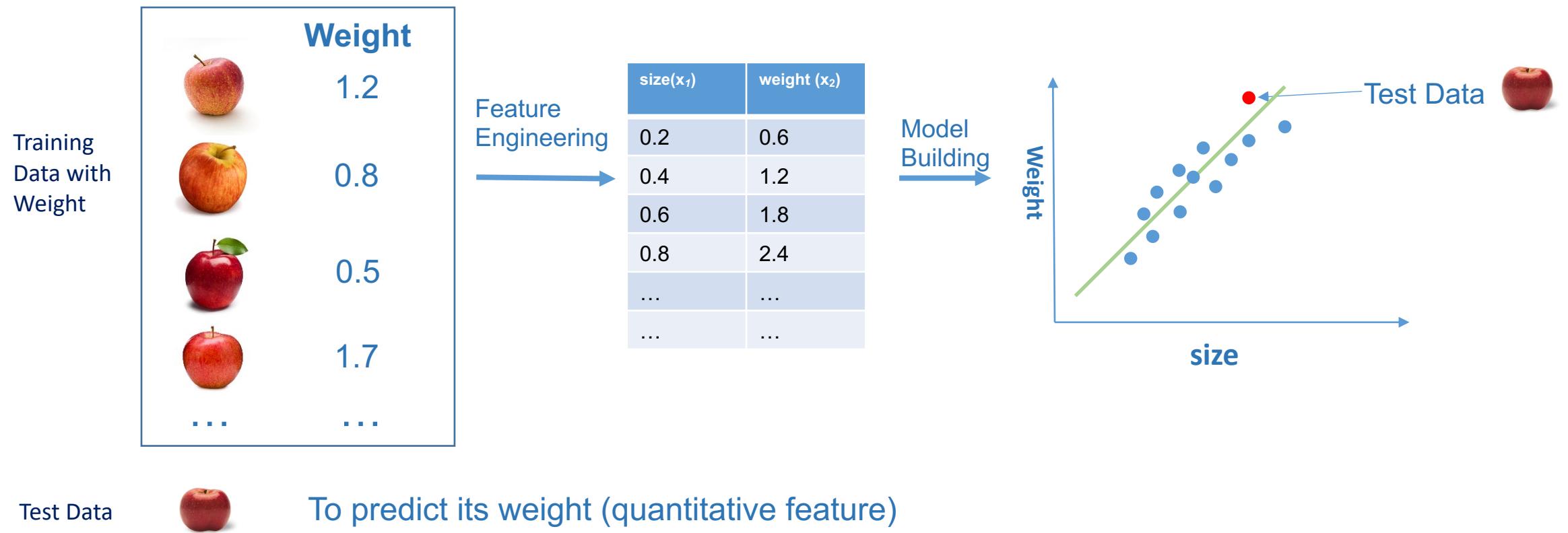


Test Data

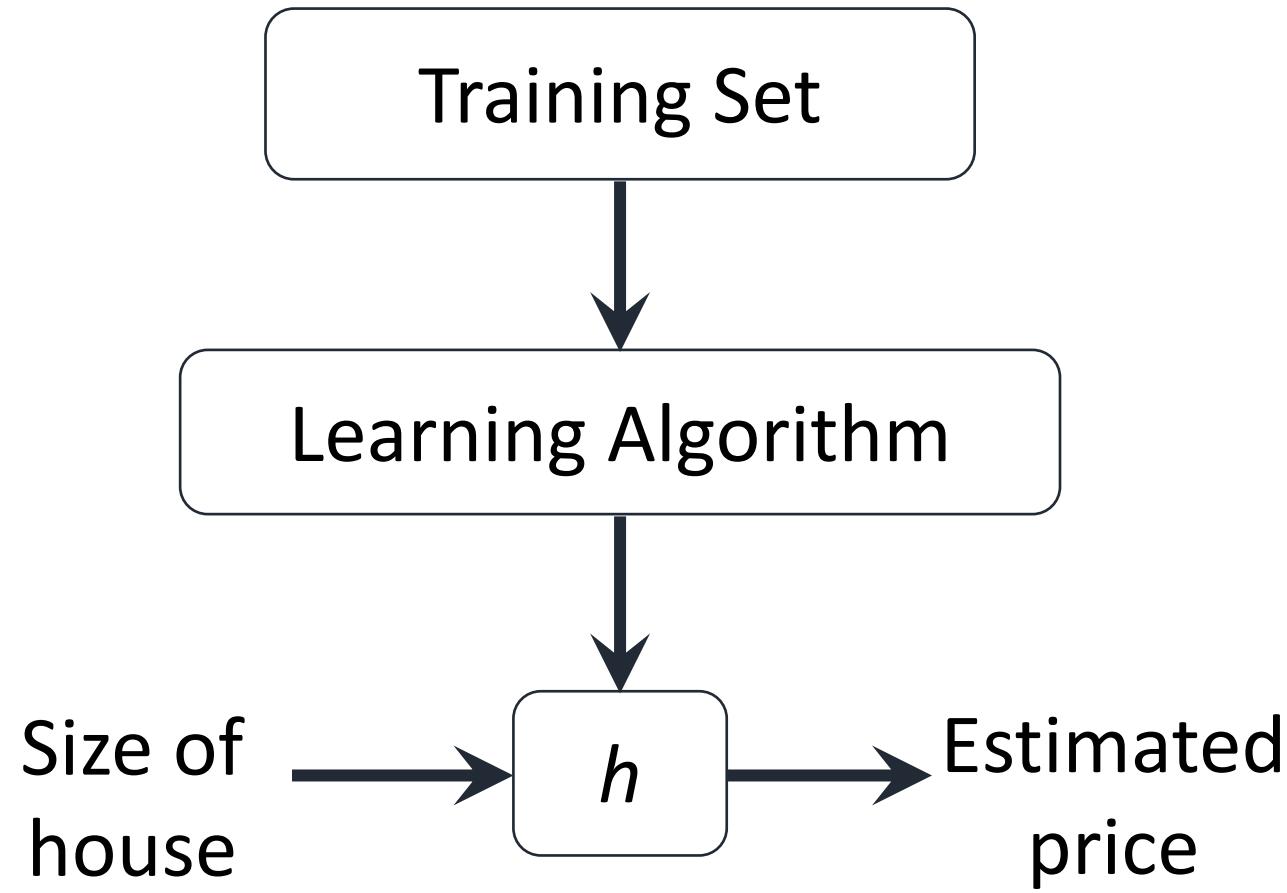


To predict it is apple or banana

An Example of Regression Problem

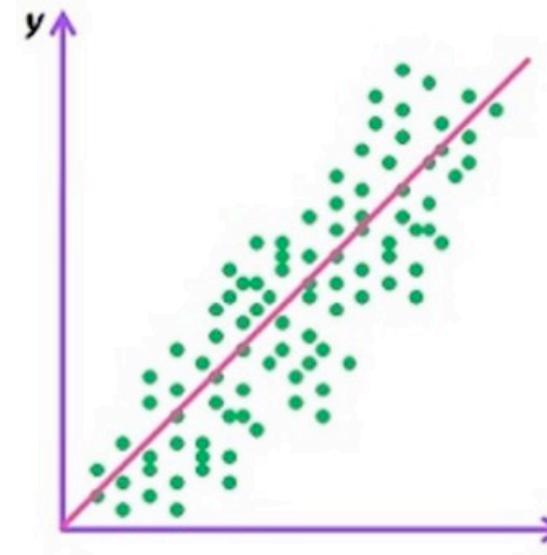


Linear model representation



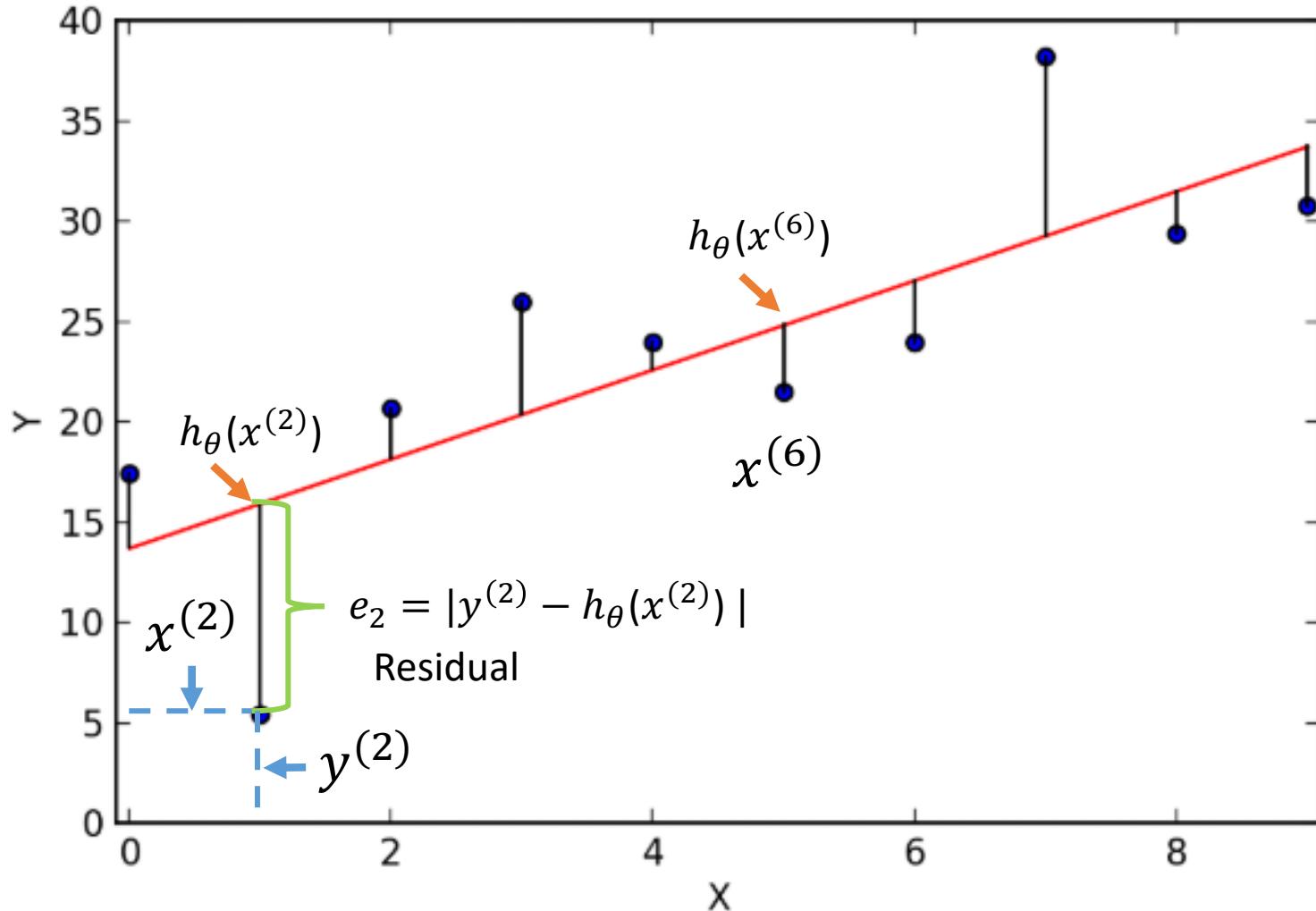
How do we represent h ?

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$



Linear regression with one variable.
Univariate linear regression.

Residual



Linear Regression one variable

Hypothesis: $h_{\theta}(x) = \theta_0 + \theta_1 x$

Parameters: θ_0, θ_1

Cost Function: $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$

Goal: $\underset{\theta_0, \theta_1}{\text{minimize}} J(\theta_0, \theta_1)$

Derive the Solution for Linear Regression on One-Dimensional Data

$$\theta_1 = \frac{\sum(x^{(i)} - \bar{x})(y^{(i)} - \bar{y})}{\sum(x^{(i)} - \bar{x})^2}$$

$$\theta_0 = \bar{y} - \theta_1 * \bar{x}$$

Linear Regression multiple variables

Hypothesis: $h_{\theta}(x) = \theta^T x = \theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$

Parameters: $\theta_0, \theta_1, \dots, \theta_n$

Cost function:

$$J(\theta_0, \theta_1, \dots, \theta_n) = \frac{1}{2m} \sum_{i=1}^m (h_{\theta}(x^{(i)}) - y^{(i)})^2$$

Gradient descent:

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \dots, \theta_n)$$

}

(simultaneously update for every $j = 0, \dots, n$)

Normal equation

Gradient Descent:

Method to solve for θ numerically.

Normal equation:

Method to solve for θ analytically.

Gradient descent:

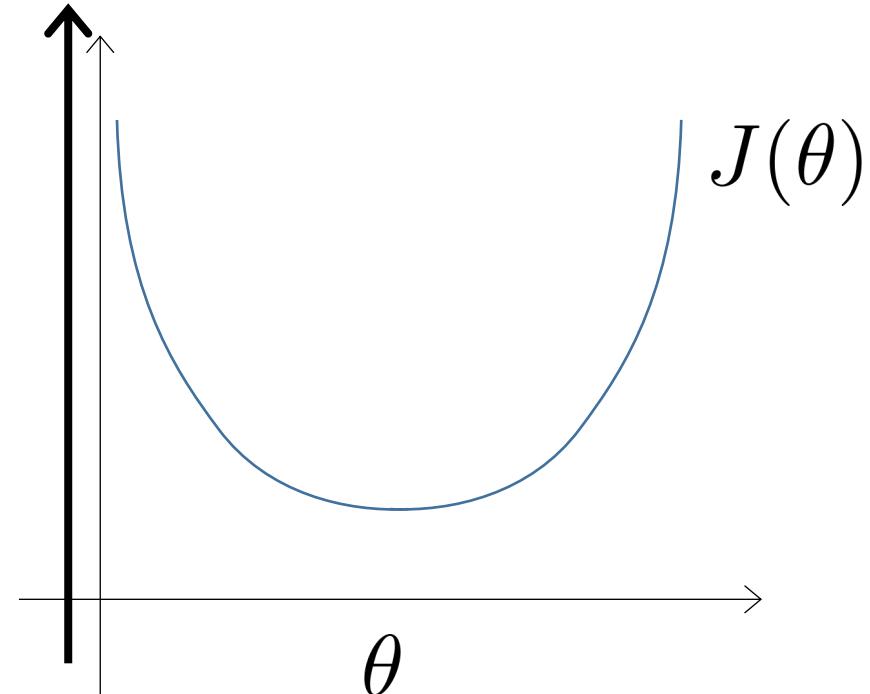
Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \dots, \theta_n)$$

}



$$\theta = (X^T X)^{-1} X^T y$$



Gradient descent and normal equation

m training cases, n variables.

Gradient Descent

- Need to choose α .
- Needs many iterations.
- Works well even when n is large.

Normal Equation

- No need to choose α .
- Don't need to iterate.
- Need to compute $(X^T X)^{-1}$
- Slow if n is very large.

Introduction to Perceptron Algorithm

- Problem Definition
 - Given a training dataset $\{\mathbf{x}_i, y_i\}_{i=1}^m$, where \mathbf{x}_i is a n dimensional input feature vector, $y_i \in \{-1, 1\}$ is the corresponding class label.
 - Objective: Train a linear classifier that can separate positive and negative samples.
 - **Training:** Learn a linear classification function $f(\mathbf{x}) = \sum_{j=1}^n w_j x_j + b$ from training data
 - $y = 1$ if $f(\mathbf{x}) = \sum_{j=1}^n w_j x_j + b > 0$
 - $y = -1$ if $f(\mathbf{x}) = \sum_{j=1}^n w_j x_j + b < 0$
 - w_j, b are the model parameters that we need to learn from training data
 - **Prediction:** for any new input \mathbf{x} , predict its class label as $y = \text{sign}(f(\mathbf{x}))$
- Simplify the notation
 - By introducing an artificial feature $x_0 = 1$, $\mathbf{x} \rightarrow [x_0, x_1, \dots, x_n]$, $f(\mathbf{x})$ can be rewritten in vector form as $f(\mathbf{x}) = \sum_{j=1}^n w_j x_j + b = \sum_{j=1}^n w_j x_j + b x_0 = \sum_{j=0}^n w_j x_j = \mathbf{w}^T \mathbf{x}$

Introduction to Perceptron Algorithm

Initialize $\mathbf{w} = \mathbf{0}$

Repeat

if $y_i(\mathbf{w}^T \mathbf{x}_i) \leq 0$ then

$\mathbf{w} \leftarrow \mathbf{w} + y_i \mathbf{x}_i$

end if

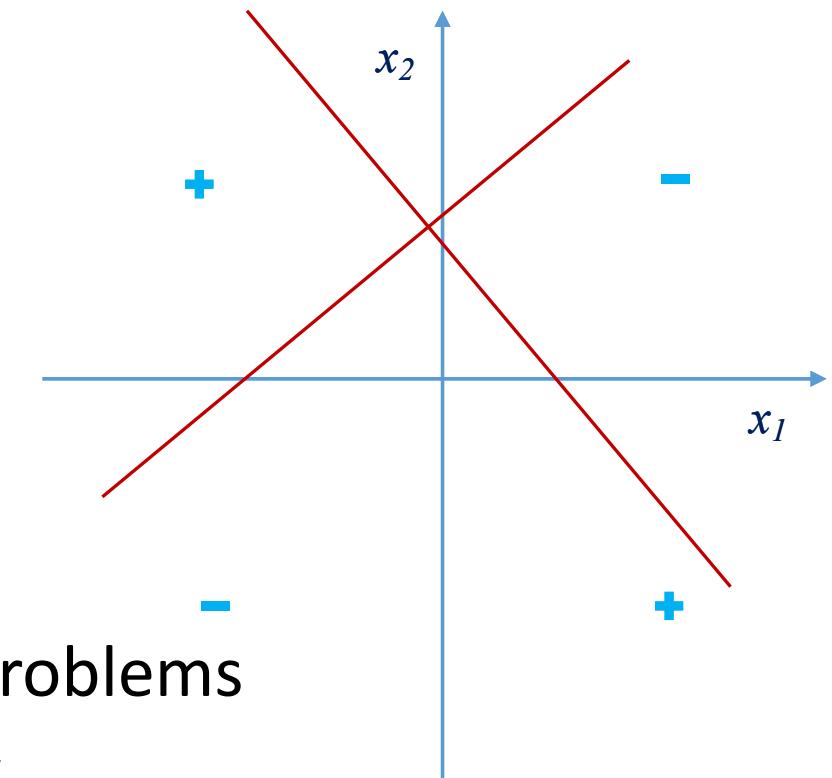
$y_i(\mathbf{w}^T \mathbf{x}_i) \leq 0$ means the training data point \mathbf{x}_i is misclassified.

- true label $y_i = 1$, but the prediction $\text{sign}(\mathbf{w}^T \mathbf{x}) = -1$, or
- true label $y_i = -1$, but the prediction $\text{sign}(\mathbf{w}^T \mathbf{x}) = 1$

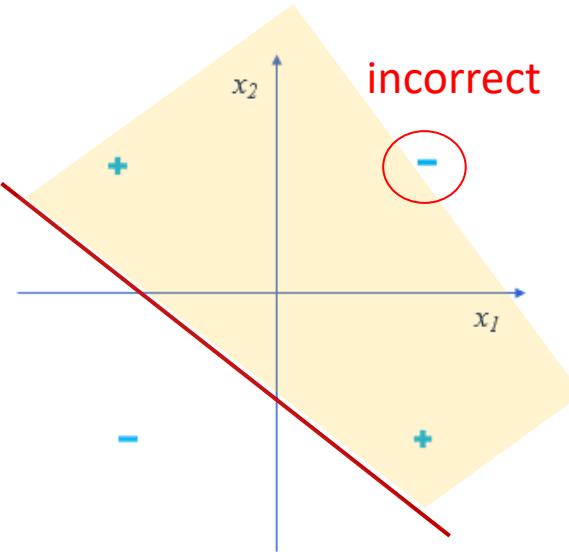
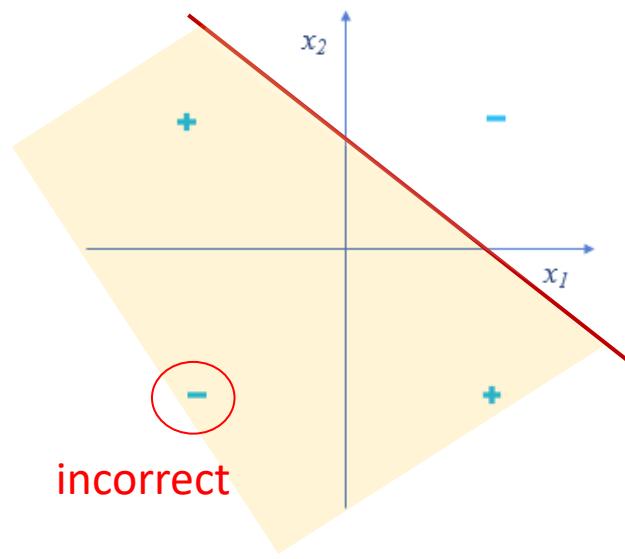
Perceptron: Nonlinear separable problems

x_1	x_2	y
1	-1	1
1	1	-1
-1	1	1
-1	-1	-1

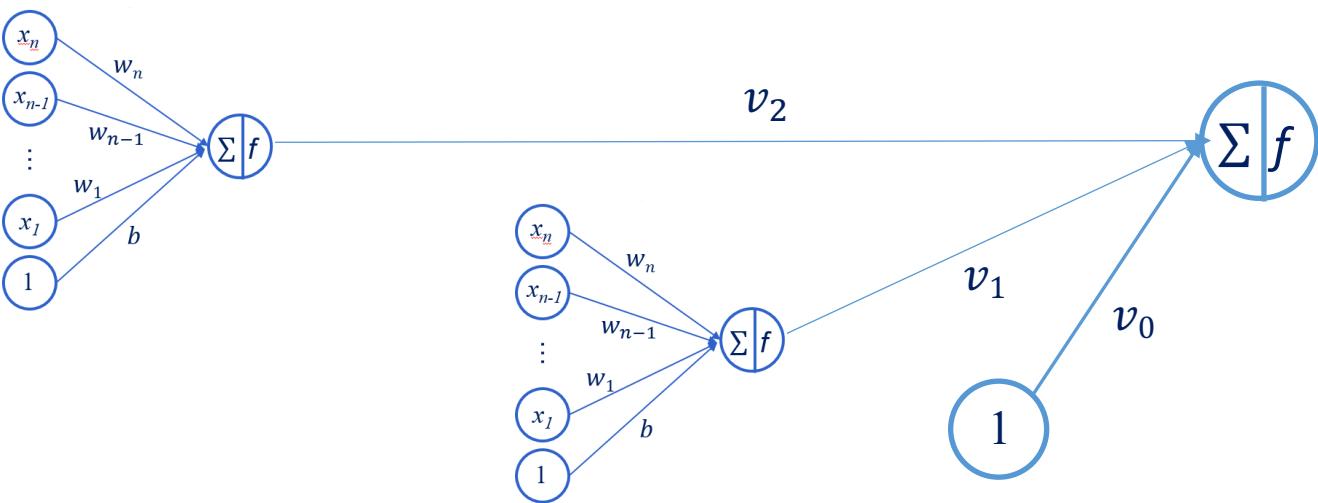
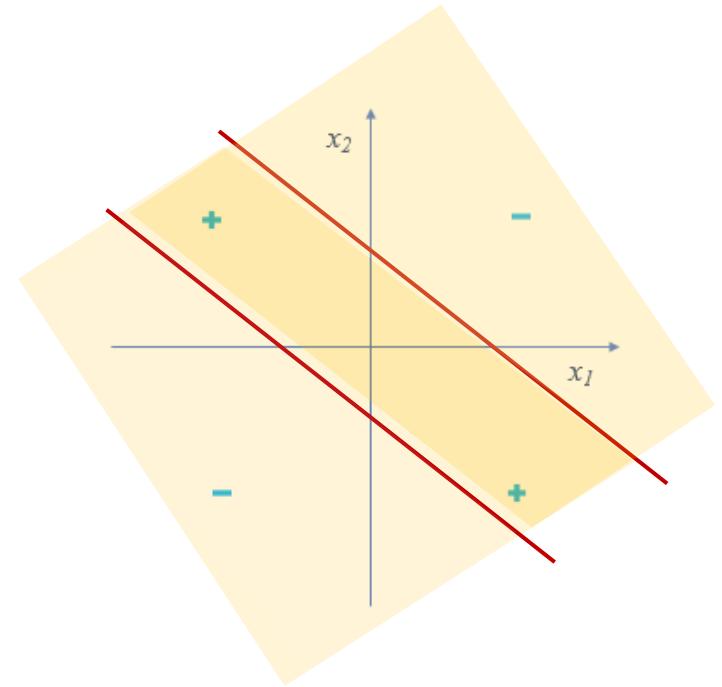
Perceptron:



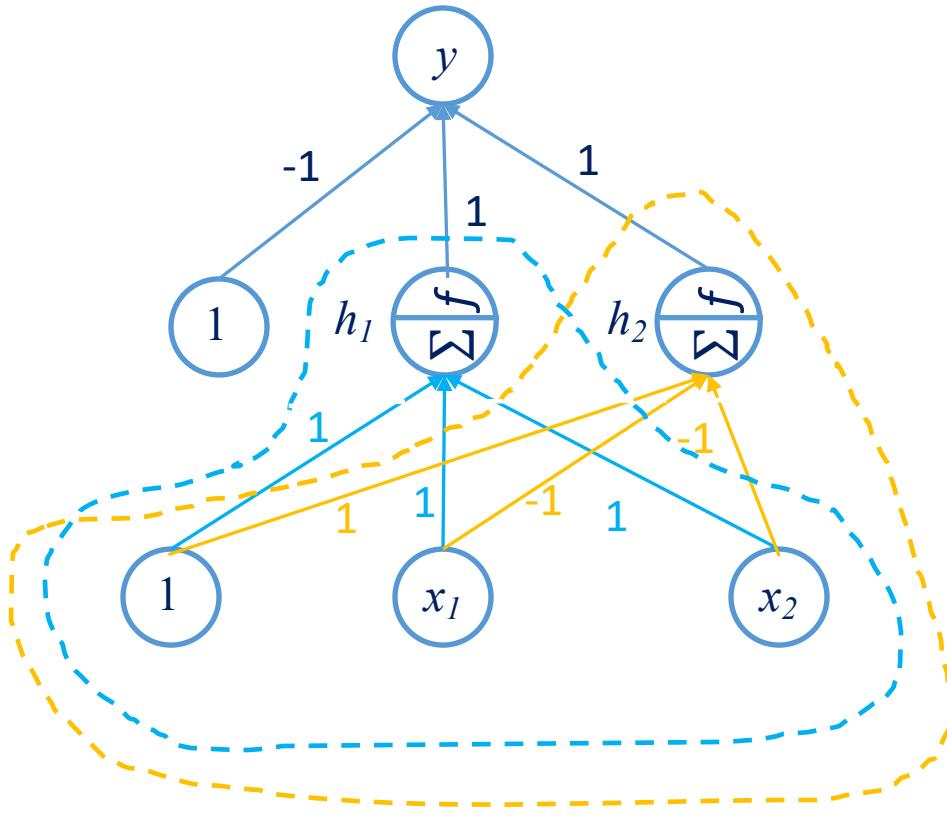
- Perceptron can not solve nonlinear separable problems
- Nonlinear separable problems can be solved by
 - Using multiple layers perceptron (**Artificial Neural Networks**)
 - Making it linearly separable using kernel (**Support Vector Machine**)



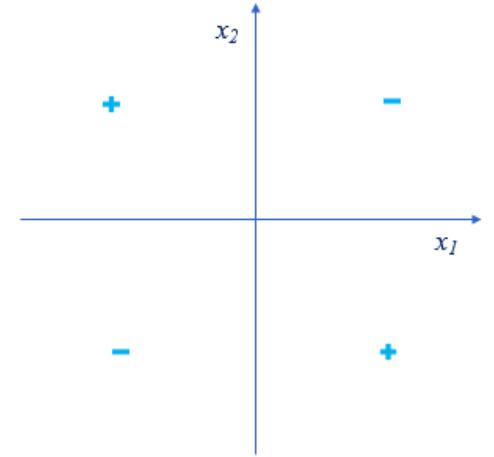
Compose
them together



Multi-Layer Perceptron for Nonlinear Classification



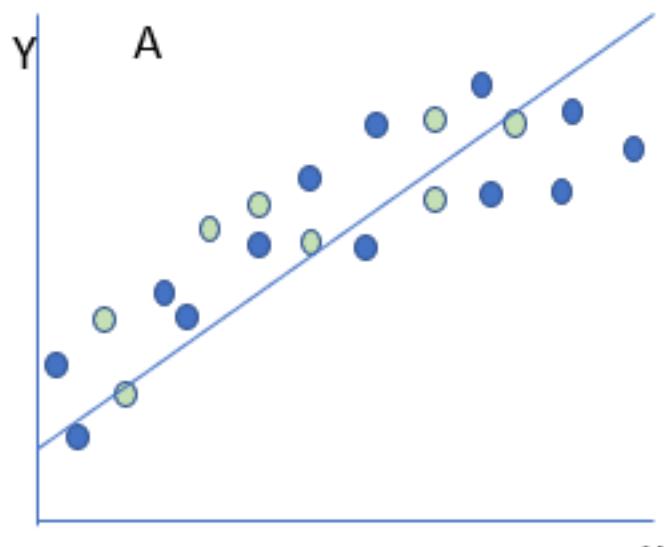
x_1	x_2	y
1	-1	1
1	1	-1
-1	1	1
-1	-1	-1



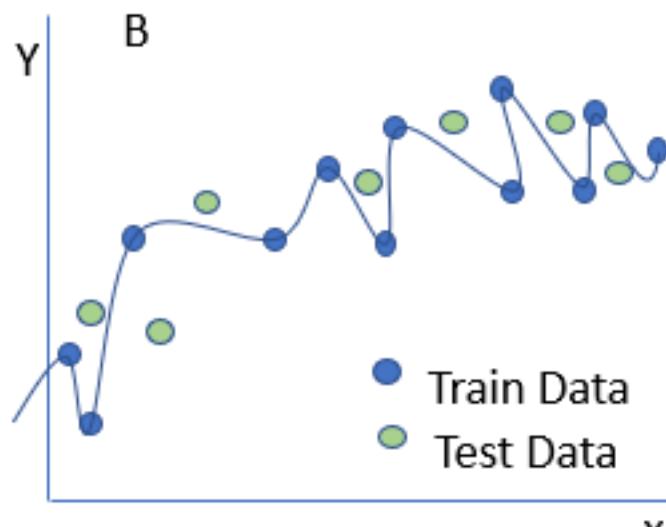
This multi-layer perceptron can solve this nonlinear classification problem.

How to learn the model parameter (i.e., weights on the edge) from data?

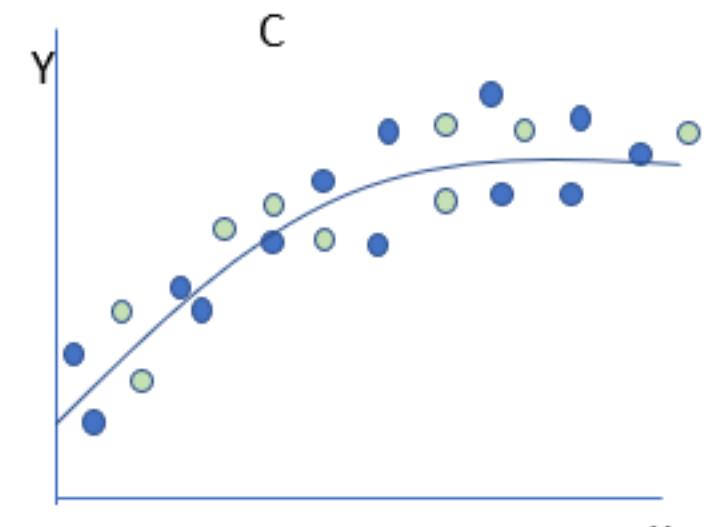
Underfitting/Overfitting and Model Complexity



Underfitting



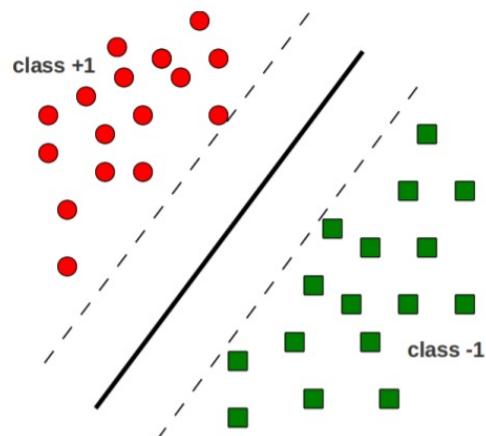
Overfitting



Well fitting

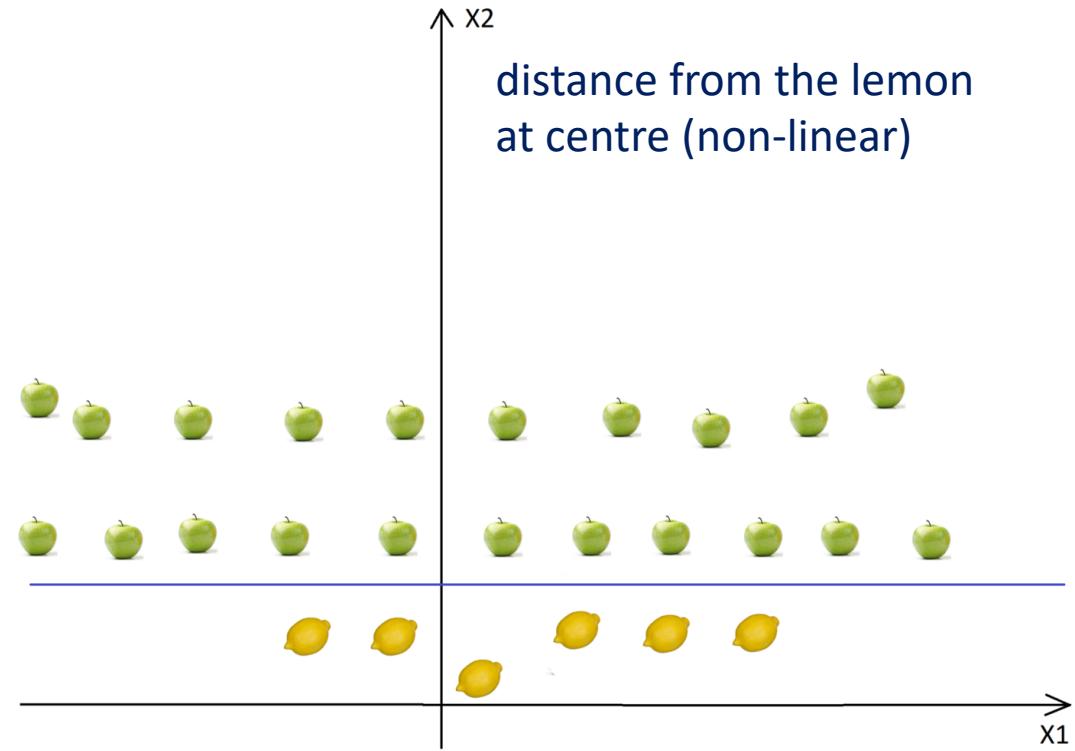
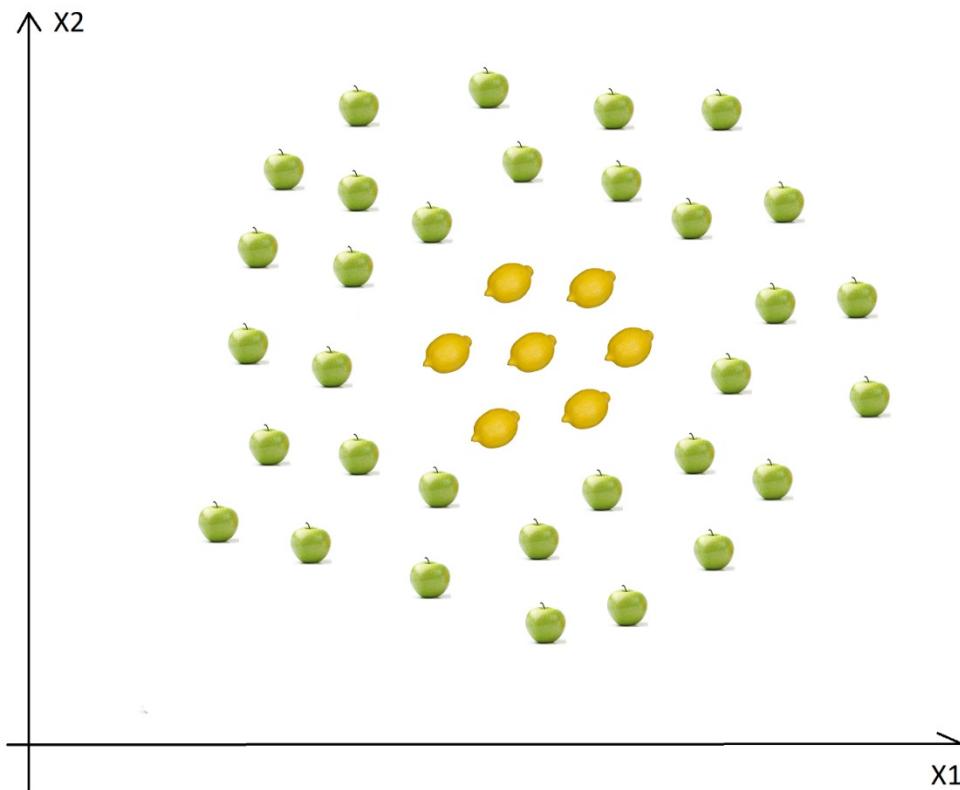
Support Vector Machine (SVM)

- Probably the most popular/influential classification algorithm
- Backed by solid theoretical groundings
- A hyperplane based classifier (like the Perceptron)
- Additionally uses the maximum margin Principle
 - Finds the hyperplane with maximum separation margin on the training data



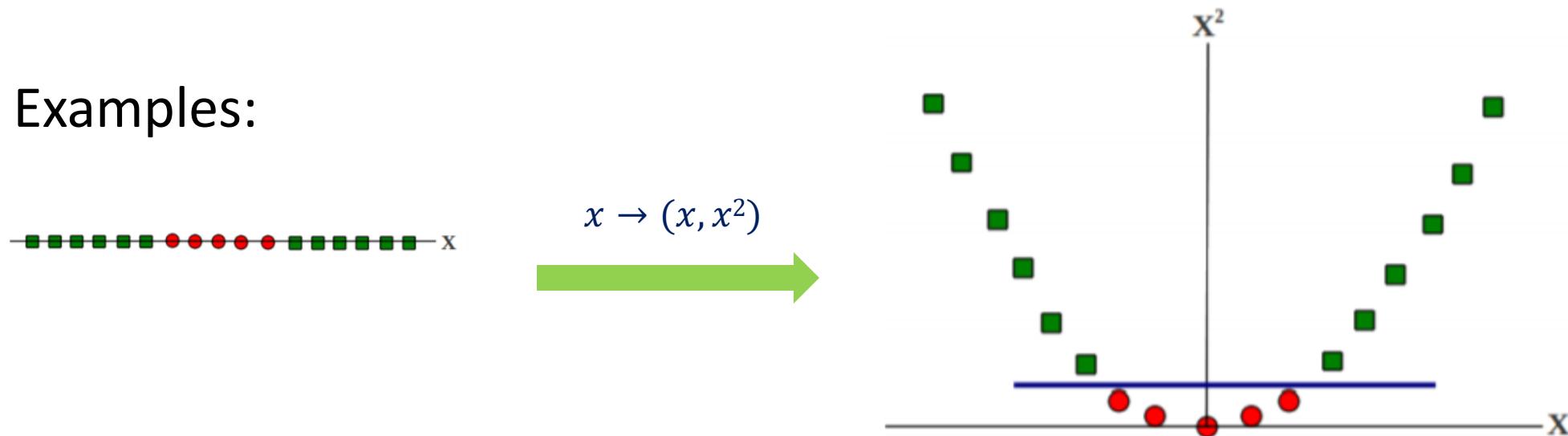
Kernel SVM for Nonlinear Classification

- Key idea: Projecting the input to a high dimensional feature space so that non-linear classification problem becomes linearly separable again!

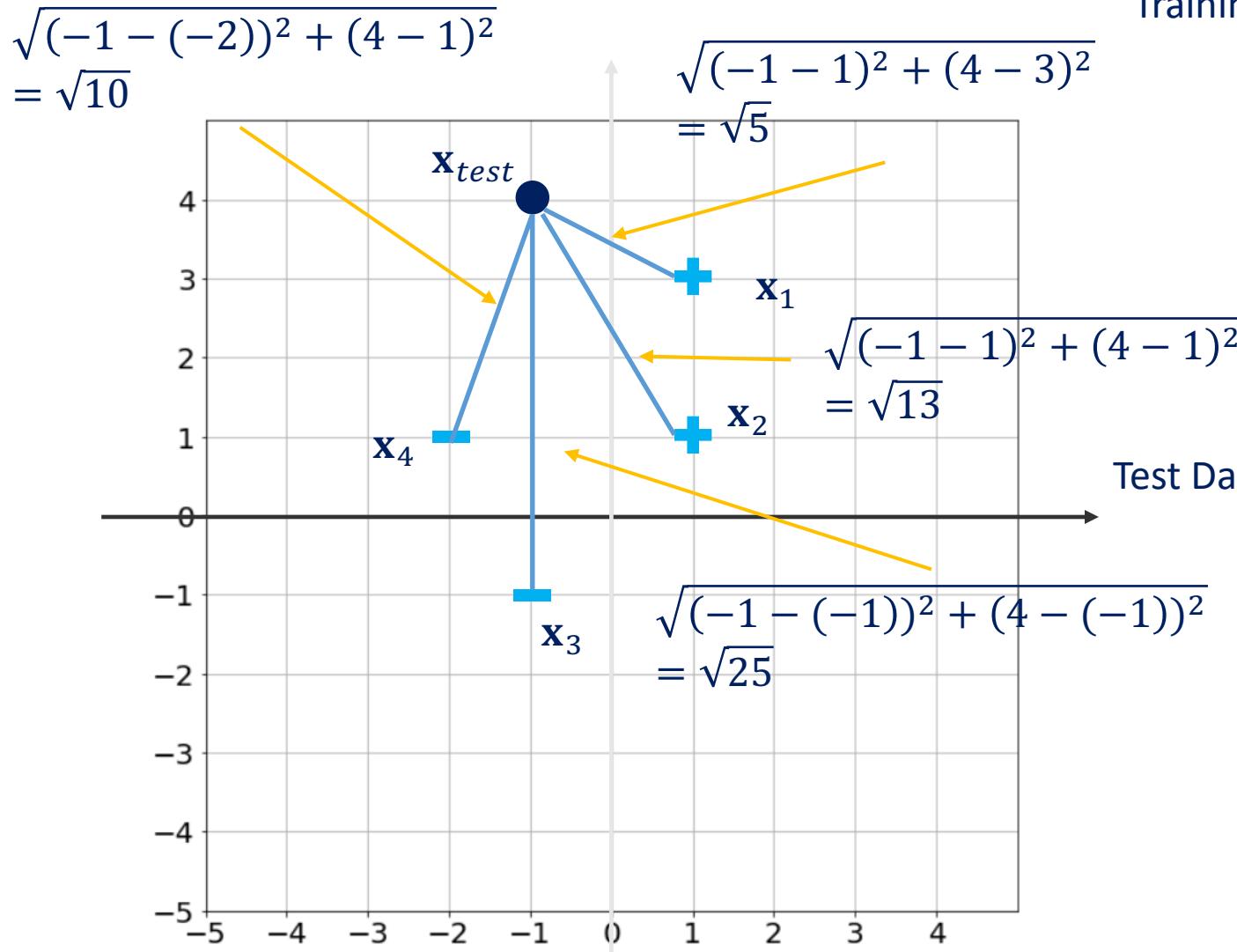


Kernels

- Kernels: Make linear models work in nonlinear settings
 - By mapping data to higher dimensions where it exhibits linear patterns.
 - Apply the linear model in the new input space
 - Mapping means changing the feature representation
- Examples:



k -Nearest Neighbors Example ($k=3$)



x_1	x_2	<i>label</i>
1	3	1
1	1	1
-1	-1	-1
-2	1	-1

x_1	x_2	<i>label</i>
-1	4	1

	x_{test}
x_1	$\sqrt{5}$
x_2	$\sqrt{13}$
x_3	$\sqrt{25}$
x_4	$\sqrt{10}$

Sort

	x_{test}
x_1	$\sqrt{5}$
x_4	$\sqrt{10}$
x_2	$\sqrt{13}$
x_3	$\sqrt{25}$

32

K-means Algorithm

- **Input:** Samples $\{\mathbf{x}_i\}_{i=1}^m$, parameter K (i.e., number of clusters)
- **Initialize:** K cluster centers (means) $\mathbf{c}_1, \dots, \mathbf{c}_k$. Several initialization options:
 - Randomly initialized anywhere in the input space
 - Randomly choose K samples from the data as the cluster centers
- **Iterate:**
 - Assign each sample \mathbf{x}_i to its closest cluster center
$$k = \arg \min_k \|\mathbf{x}_i - \mathbf{c}_k\|$$
 - Re-compute the cluster center \mathbf{c}_k for every new cluster
$$\mathbf{c}_k = \frac{1}{|C_k|} \sum_{\mathbf{x}_i \in C_k} \mathbf{x}_i$$
 - Repeat while not converged
 - Converge criteria: Cluster centers do not change anymore

C_k is the set of samples in cluster k
 $|C_k|$ denotes the number of samples in C_k

Hierarchical Clustering: (Dis)similarity Between Clusters

- **Single Linkage**

- Smallest distances between samples, where each one is taken from one of the two groups

- **Complete Linkage**

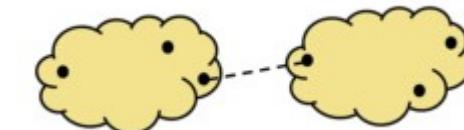
- Largest distances between samples, where each one is taken from one of the two groups

- **Average linkage**

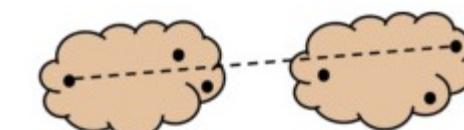
- Average distance between all samples in one cluster to all points in another cluster

- **Centroid linkage**

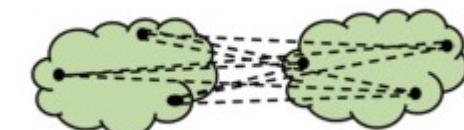
- Distance between their **centroids**.



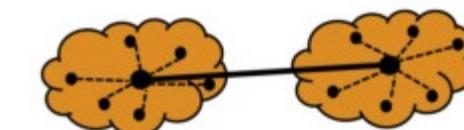
Single Linkage



Complete Linkage



Average Linkage



Centroid Linkage

Example: Hierarchical Clustering (with Centroid Linkage)



x_1	x_2
0	0
1	2
2	1
4	1
5	0
5	3

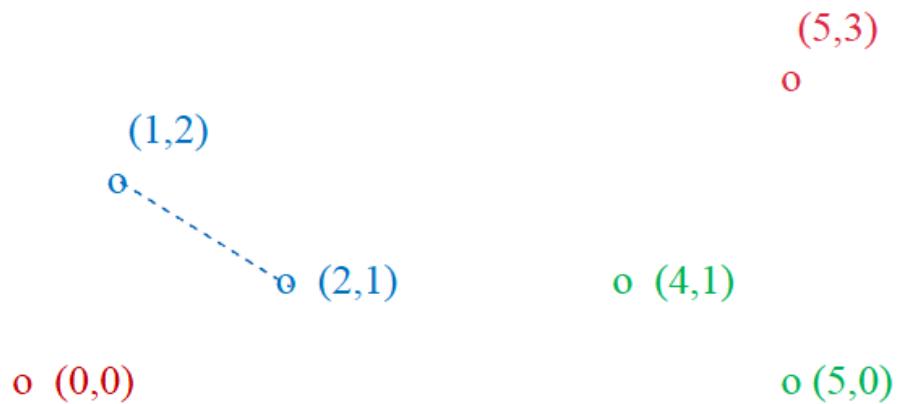
Data:

o ... data point

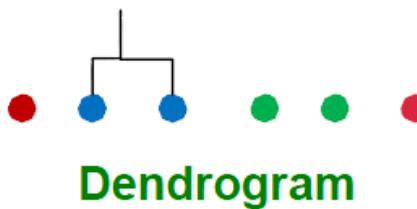


Dendrogram

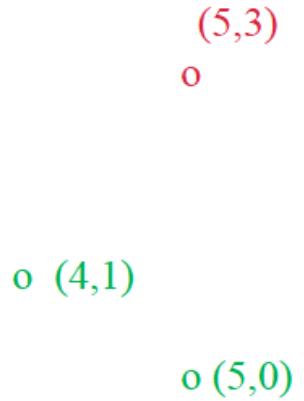
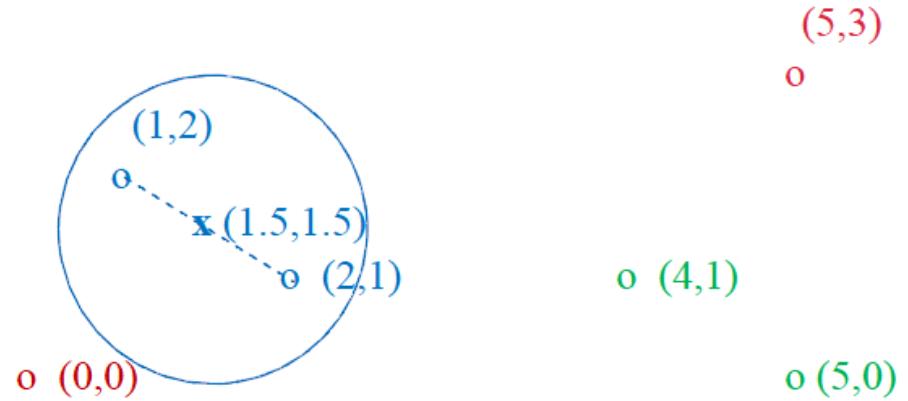
Example: Hierarchical Clustering (with centroid linkage) Step 1



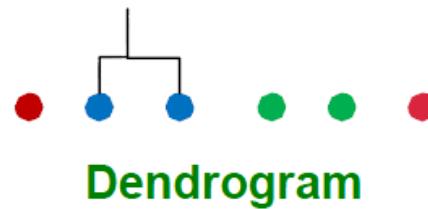
Data:
o ... data point



Example: Hierarchical Clustering (with centroid linkage) Step 2

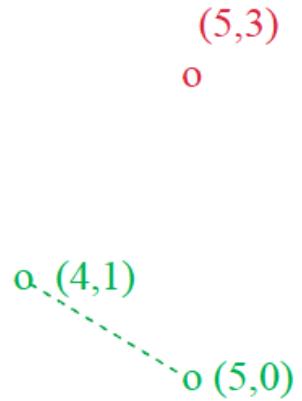
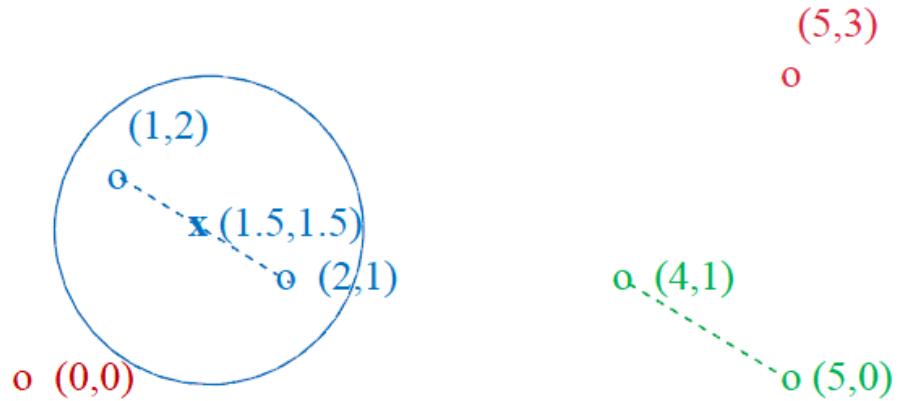


Data:
o ... data point
x ... centroid

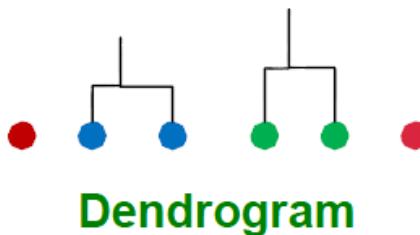


Dendrogram

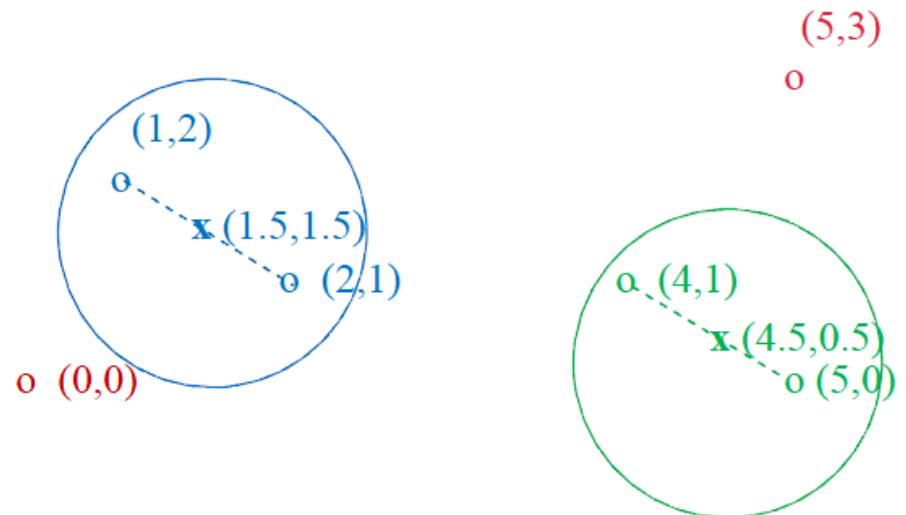
Example: Hierarchical Clustering (with centroid linkage) Step 3



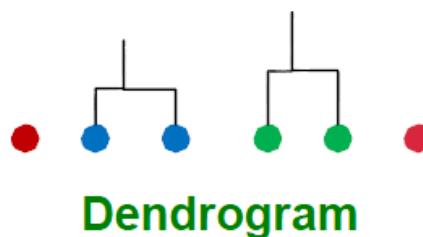
Data:
o ... data point
x ... centroid



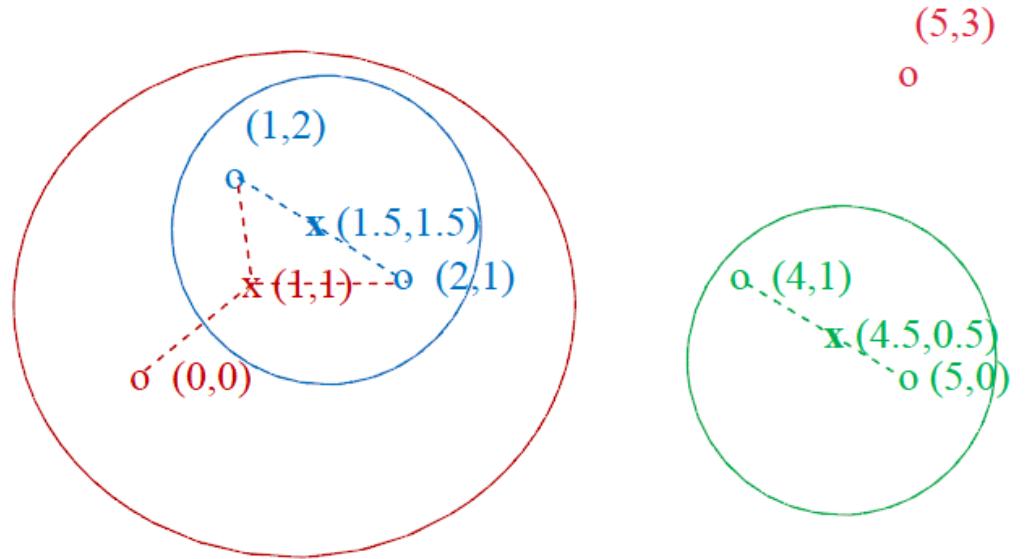
Example: Hierarchical Clustering (with centroid linkage) Step 4



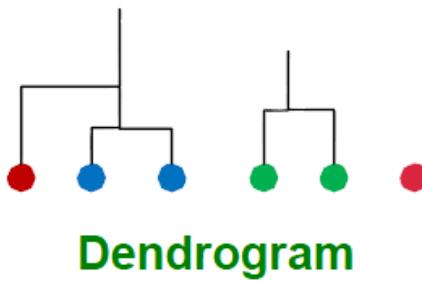
Data:
o ... data point
x ... centroid



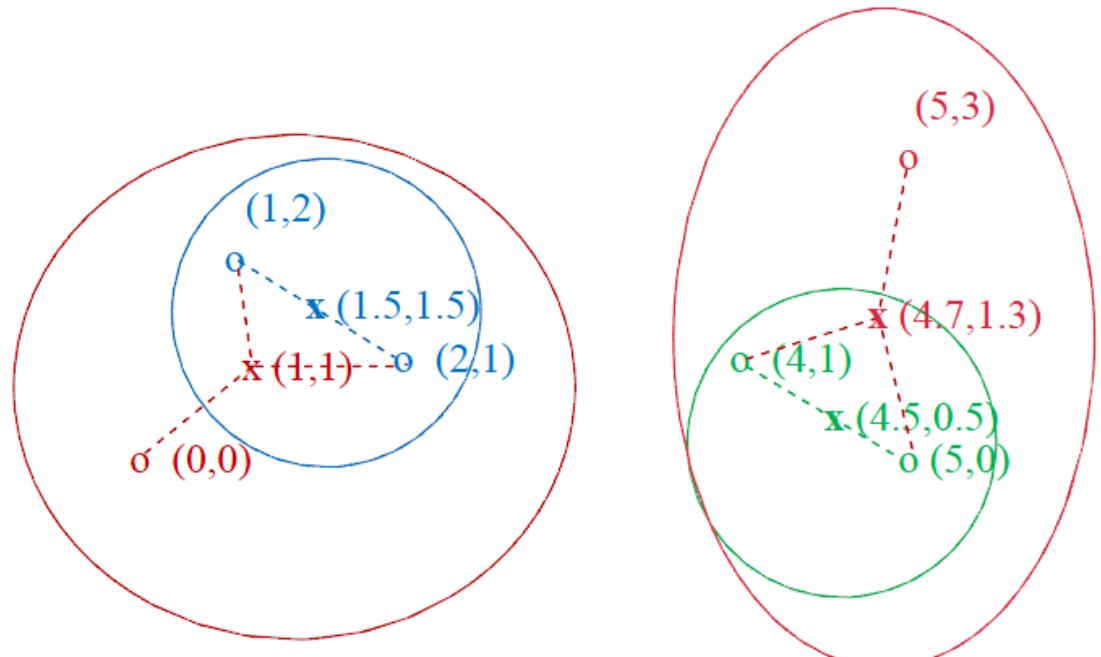
Example: Hierarchical Clustering (with centroid linkage) Step 5



Data:
o ... data point
x ... centroid

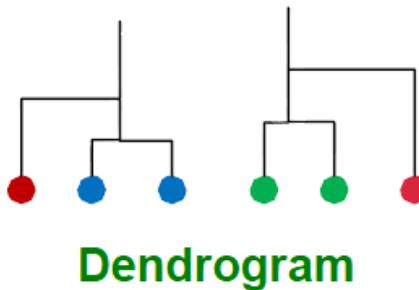


Example: Hierarchical Clustering (with centroid linkage) Step 6



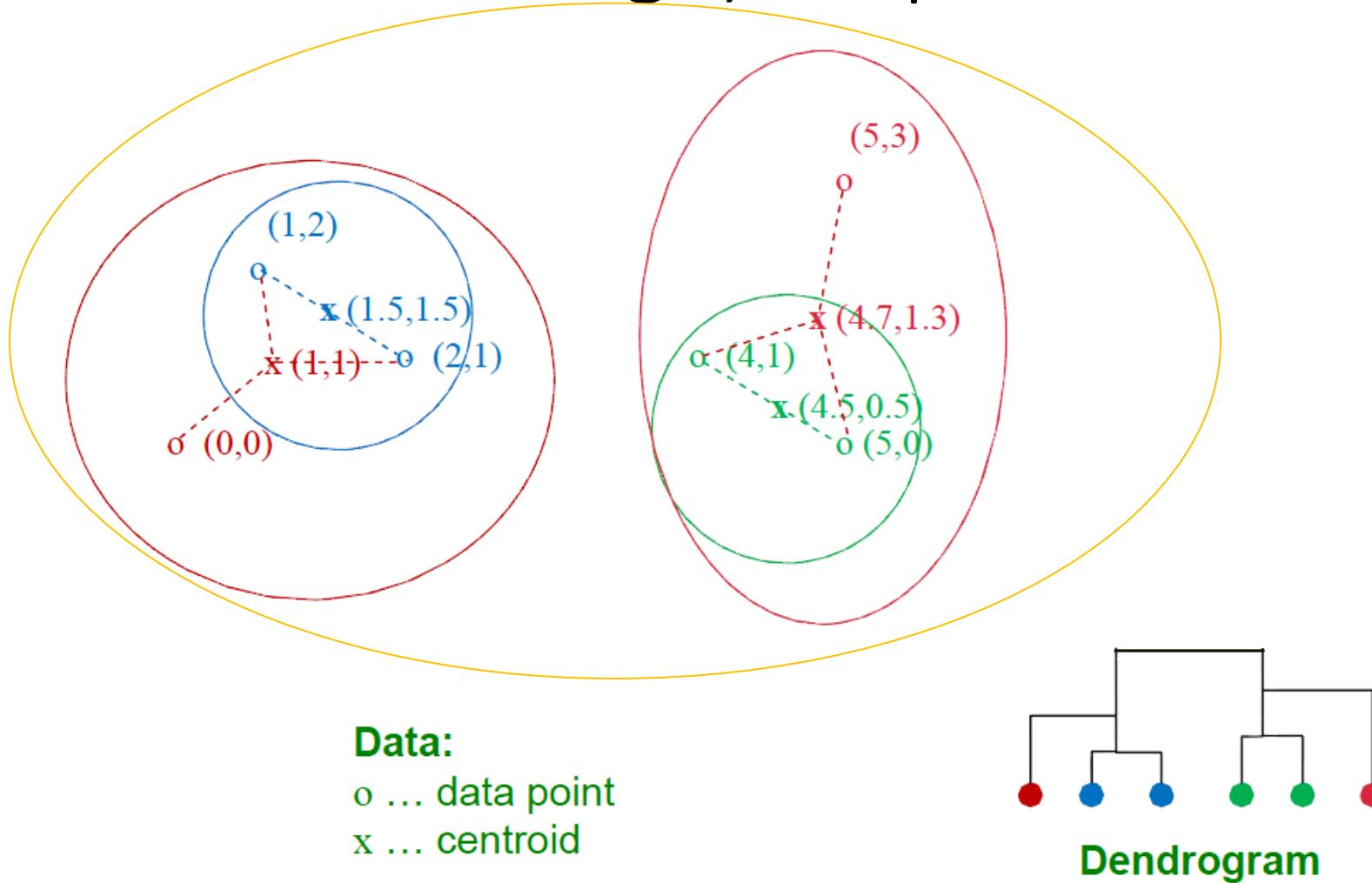
Data:

o ... data point
x ... centroid



Dendrogram

Example: Hierarchical Clustering (with centroid linkage) Step 7

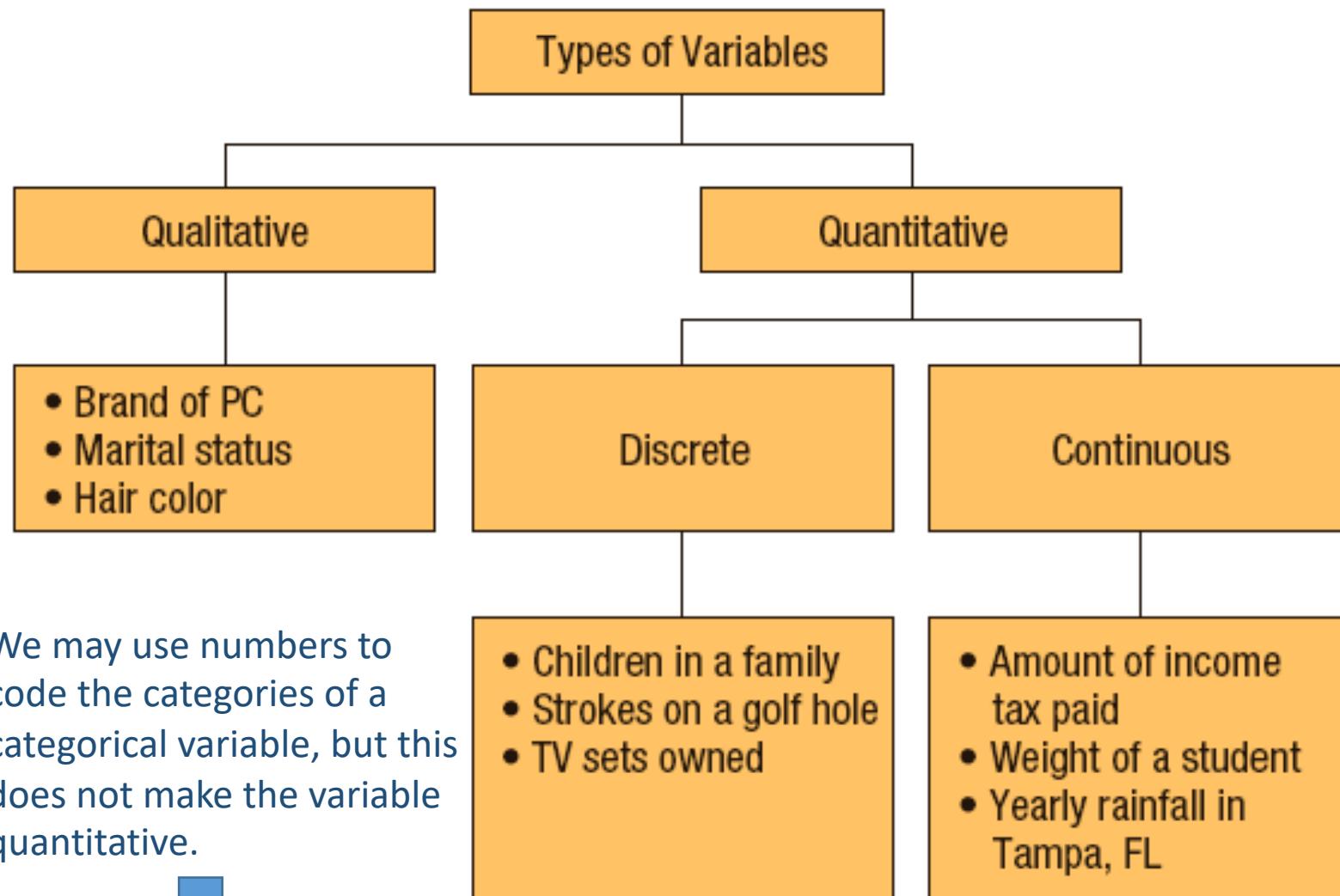


Statistics

Revision Guide

- Summary Statistics
 - Center: mean, median
 - Spread: standard deviation, range, IQR
 - Five number summary
 - Shape: symmetric, skewed, bell-shaped
 - Outliers, resistance
- Statistics
 - Standard error
 - Confidence interval

Variables – Categorical and Quantitative Variables



We may use numbers to code the categories of a categorical variable, but this does not make the variable quantitative.

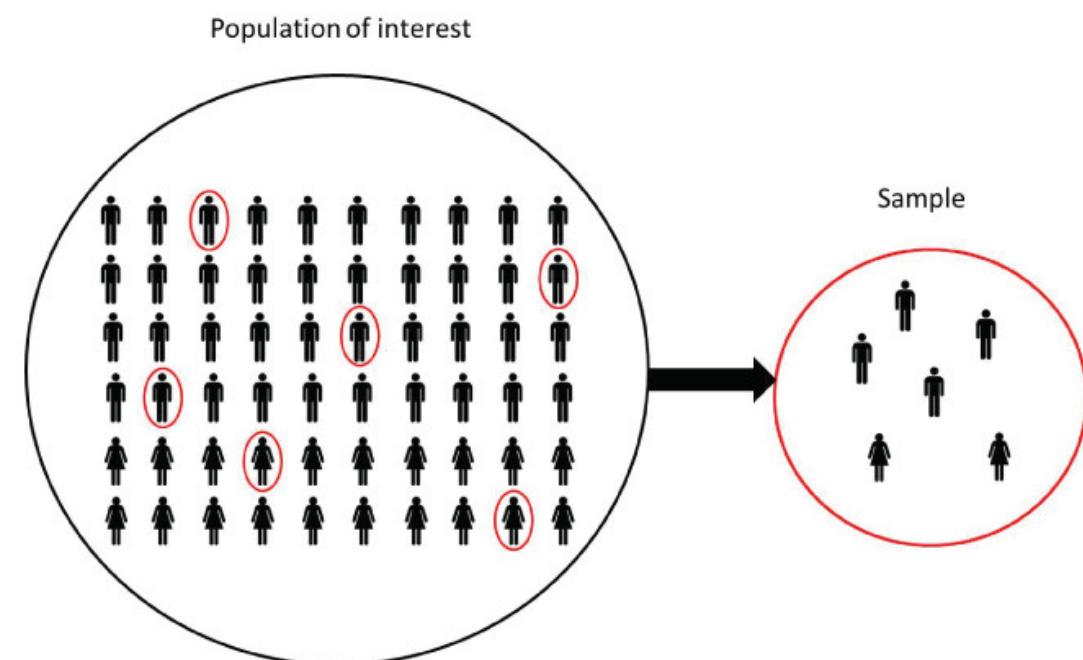


For example, "gender" is categorical even if we choose to record the results as 1 for male and 2 for female.

- A **qualitative / categorical** variable divides the cases into **groups**, placing each case into exactly one of two or more categories.
- A **quantitative /numerical** variable measures or records a numerical quantity for each case. Numerical operations like adding and averaging make sense for quantitative variables.

Random Sampling

- How do we get a sample that looks like the population?
- The key is **random sampling!!!**
- A random sample will resemble the population!
- Random sampling avoids sampling bias!

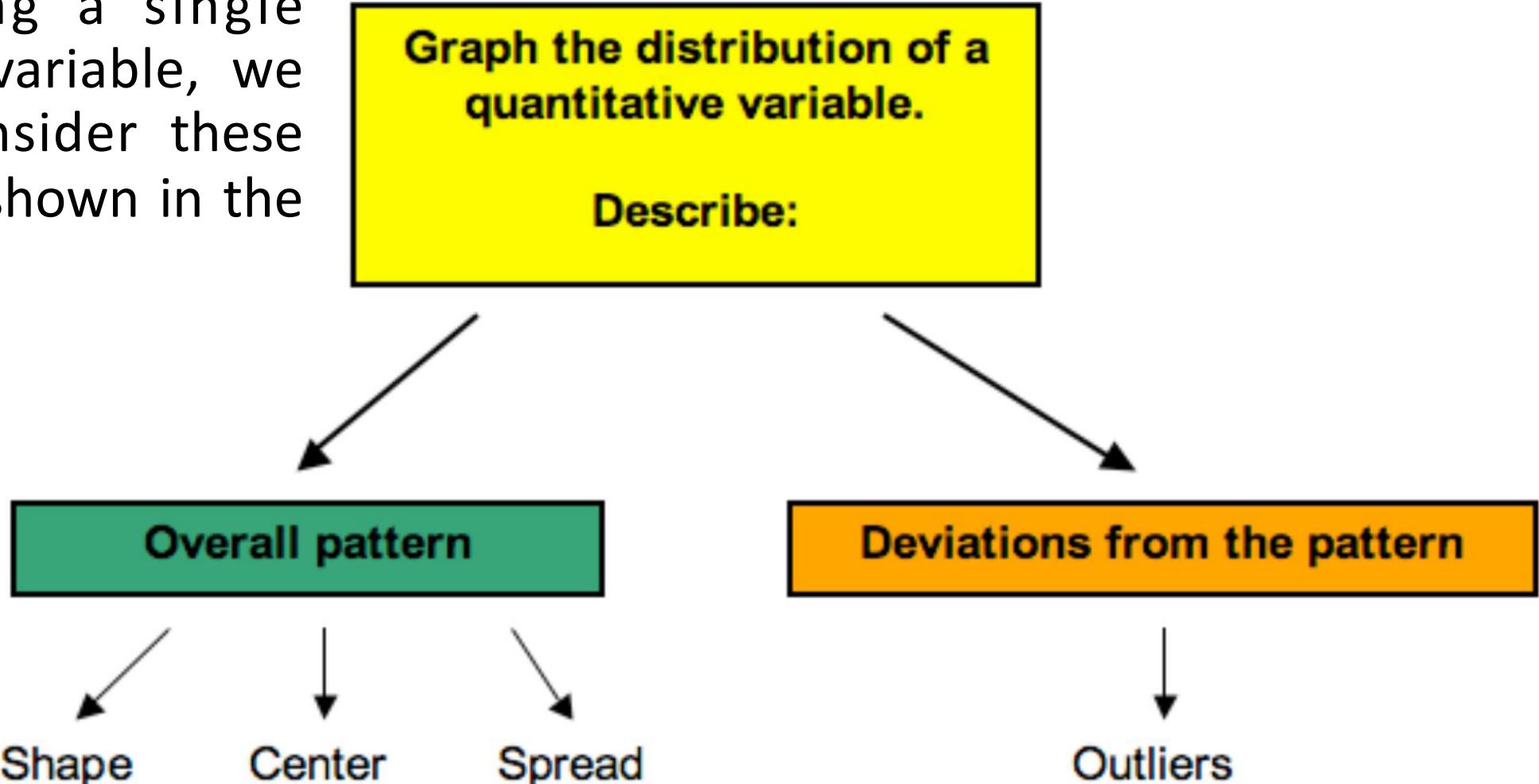


Example of Biased Sampling

- CFQ
 - Only students who extremely like this course or dislike this course will go to do the CFQ.
 - The CFQ result is heavily biased to two ends!
- Students can visit the Course Feedback Questionnaire System (CFQ) at <https://cfq-student.hkbu.edu.hk> to provide feedback. The evaluation period will end on Saturday, 3rd December 2022, 23:59:59.

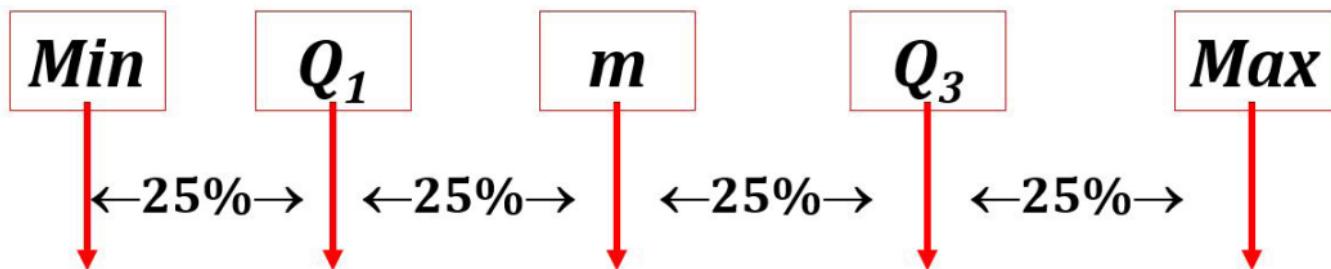
Distribution

- In describing a single quantitative variable, we generally consider these questions as shown in the right graph.



Other measures of location and spread

- Maximum = the largest value
- Minimum = the smallest value
- Mode: The value that has the highest frequency.
- Quartiles:
 - ❖ Q_1 = median of the values below m
 - ❖ Q_3 = median of the values above m
- Five Number Summary



- Range = Max - Min; Is the range resistant to outliers? No!
- Interquartile Range (IQR) = $Q_3 - Q_1$; Is the IQR resistant to outliers? Yes!

Standard Deviation

The standard deviation for a quantitative variable measures the spread of the data in a sample.

- The standard deviation of a **sample** is denoted **s**, and measures how spread out the data are from the sample mean \bar{x} .

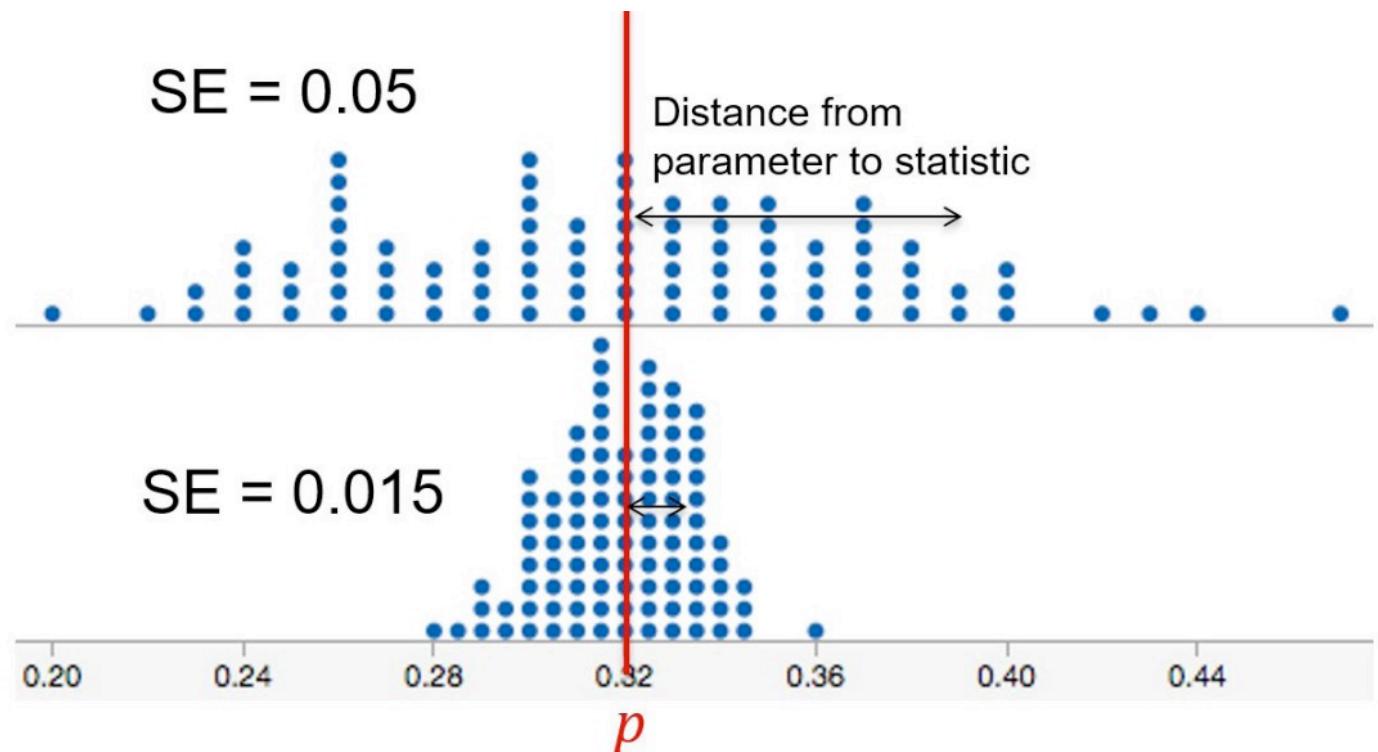
$$s = \sqrt{\frac{\sum_i (x_i - \bar{x})^2}{n - 1}}$$

- The standard deviation of a **population** is denoted **σ** , which is the Greek letter “sigma” and measures how spread out the data are from the population mean μ .

$$\sigma = \sqrt{\frac{\sum_i (x_i - \mu)^2}{n}}$$

Standard Error

- The standard error of a statistics, denoted SE, is the standard deviation of the sample statistic
- The standard error can be calculated as the standard deviation of the sampling distribution
 - ❖ The standard error measures how much the statistic varies from sample to sample
 - The more the statistic varies from sample to sample, the higher the standard error
 - Lower SE means statistic closer to true parameter value
 - ❖ SE measures 'typical' distance between parameter and statistic

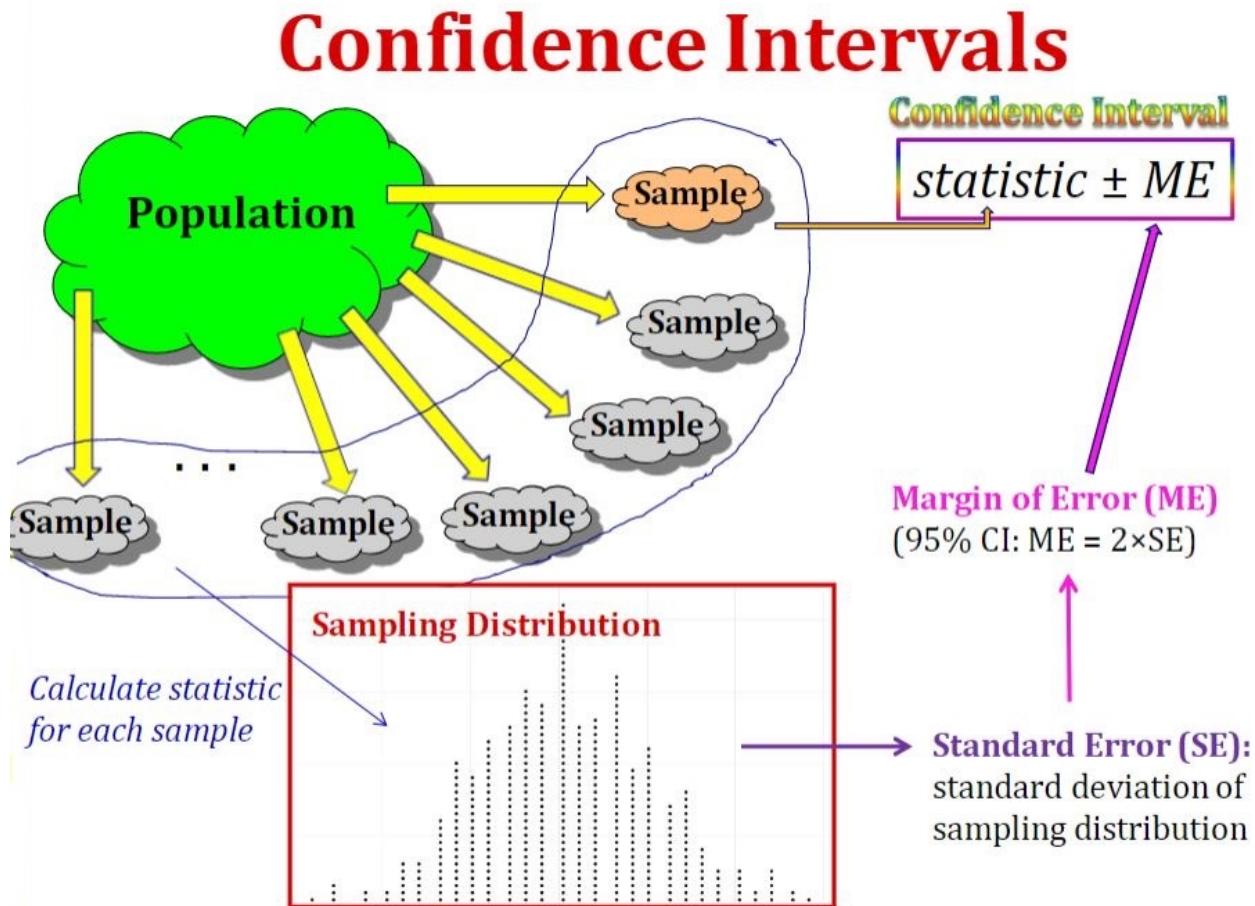


Confidence Intervals

- Confidence Interval
 - ❖ A Confidence Interval is a range of values we are sure our true value lies in.
 - ❖ The success rate (proportion of all samples falls into the confidence interval) is known as the confidence level.
- Calculation of 95% Confidence Interval Using the Standard Error
 - ❖ If we can estimate the standard error and the sampling distribution is relatively symmetric and bell-shaped, a 95% confidence interval can be estimated using

Statistic \pm 2 * SE

95% confidence interval



Construction of confidence interval (an interval estimate) for a sample of size at least 30:

- Step1: Compute the sample mean \bar{x} of the random sample. Suppose we know the population standard deviation σ . Compute the standard error: $\sigma_{\bar{x}} = \frac{\sigma}{\sqrt{n}}$

where n is the sample size

- Step2: Construct the 95% confidence interval

$$[\bar{x} - 2\sigma_{\bar{x}}, \bar{x} + 2\sigma_{\bar{x}}]$$

Here 2 is chosen according to the confidence level 95%

- If σ is unknown, then we use s to estimate σ

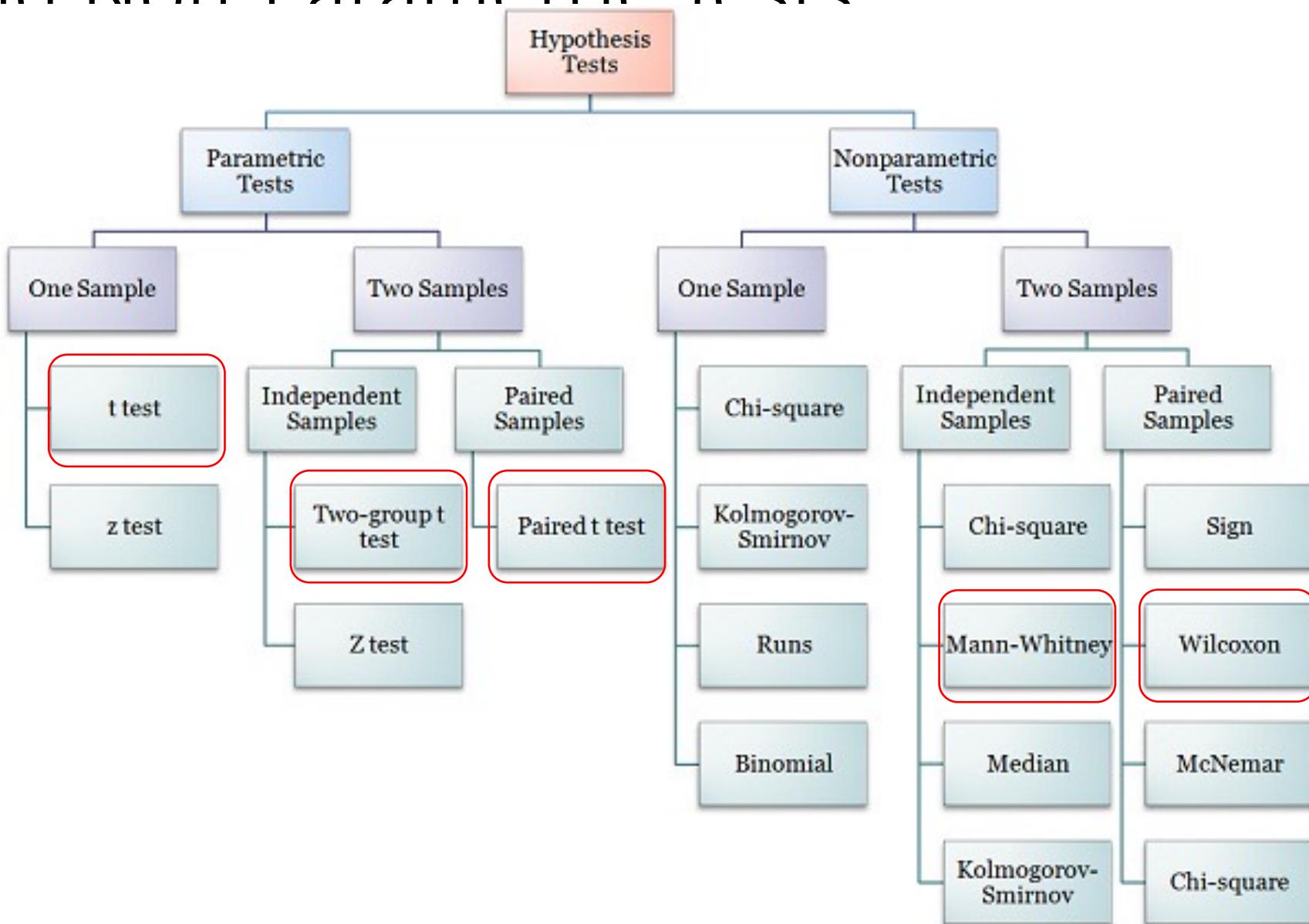
$$s_{\bar{x}} = \frac{s}{\sqrt{n}}$$

- Then the 95% confidence interval is

$$[\bar{x} - 2s_{\bar{x}}, \bar{x} + 2s_{\bar{x}}]$$

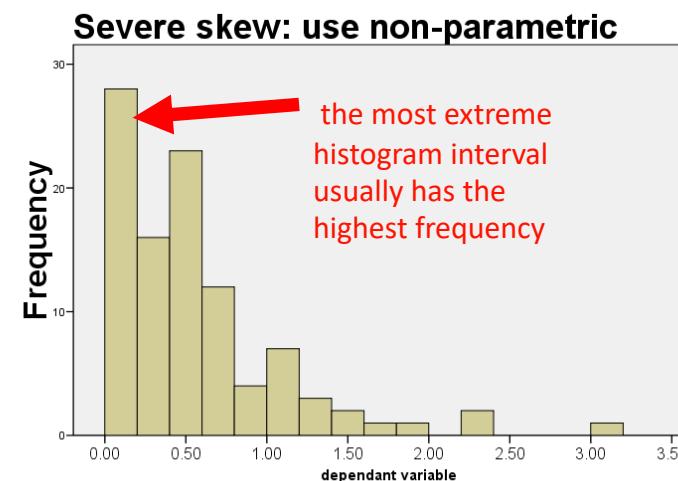
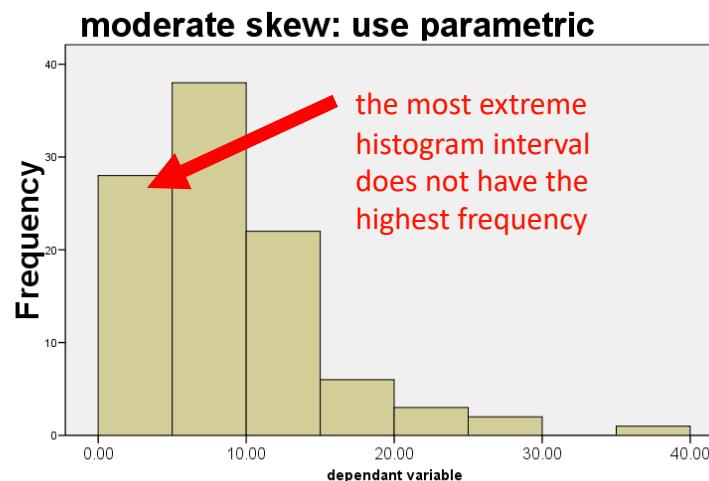
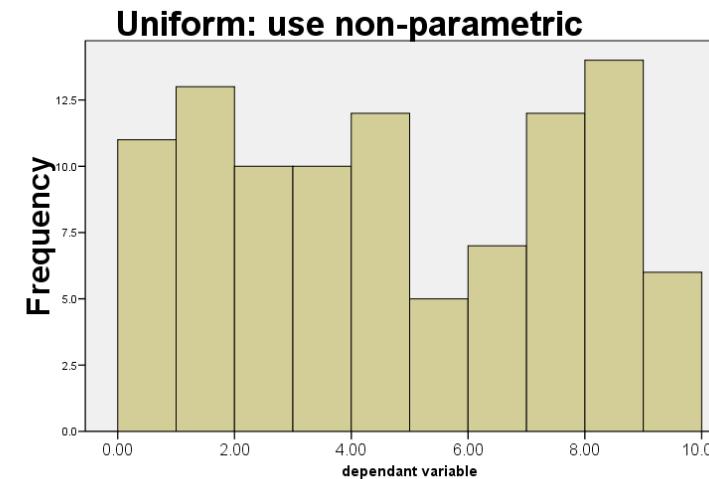
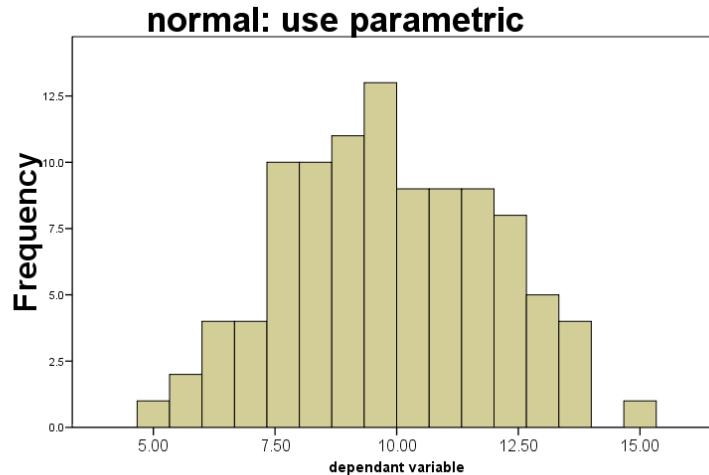
Parametric and Non-Parametric Tests

- **Parametric Tests:** assume the distribution of sample data (i.e. normality)
- **Non-Parametric Tests:** do not assume data are drawn from any particular distribution



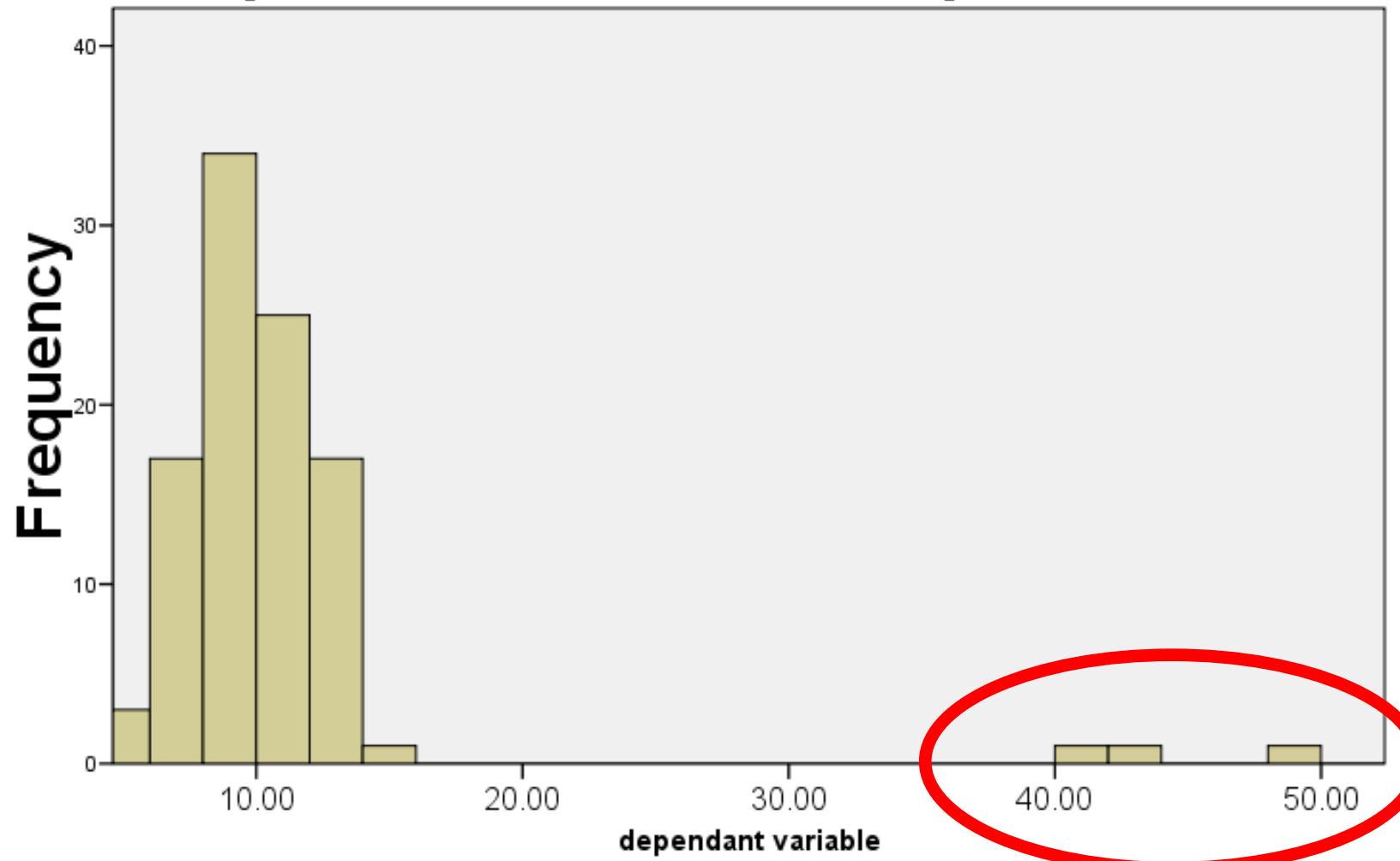
T-test assumption 1 - normality

Assumption 1: The sampling distribution is normally distributed



T-test assumption 2-No extreme scores

normal plus outliers: use non-parametric



There should not include ***extreme scores*** or ***outliers***, because these have a disproportionate influence on the mean and the variance

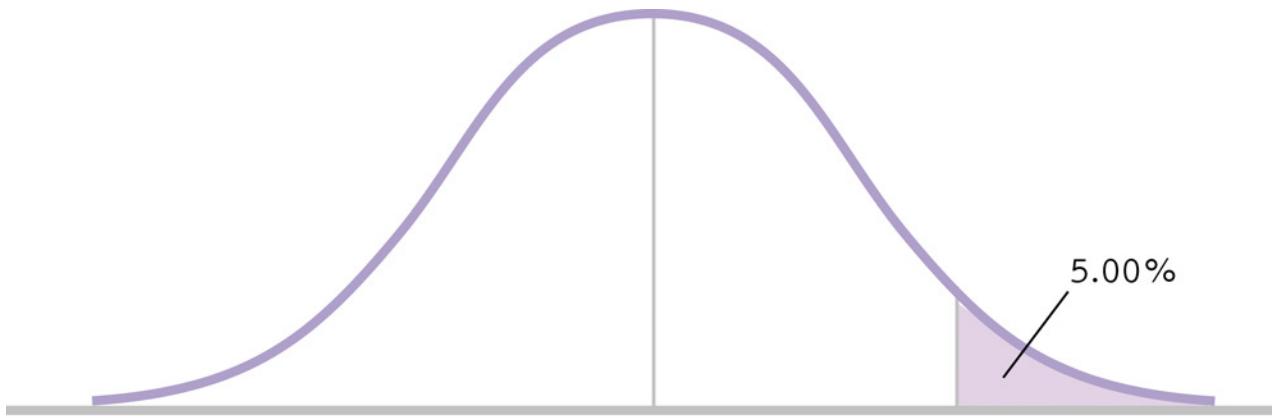
Types of *T-test*

- Single sample t
 - ❖ One sample, compared with known population mean
 - ❖ Goal: Is current sample different from population?
- Independent samples t
 - ❖ Different (independent) samples of participants
 - ❖ Are our samples from different populations?
- Paired/Dependent Samples t
 - ❖ Same or related (dependent) samples
 - ❖ Are our samples from different populations?

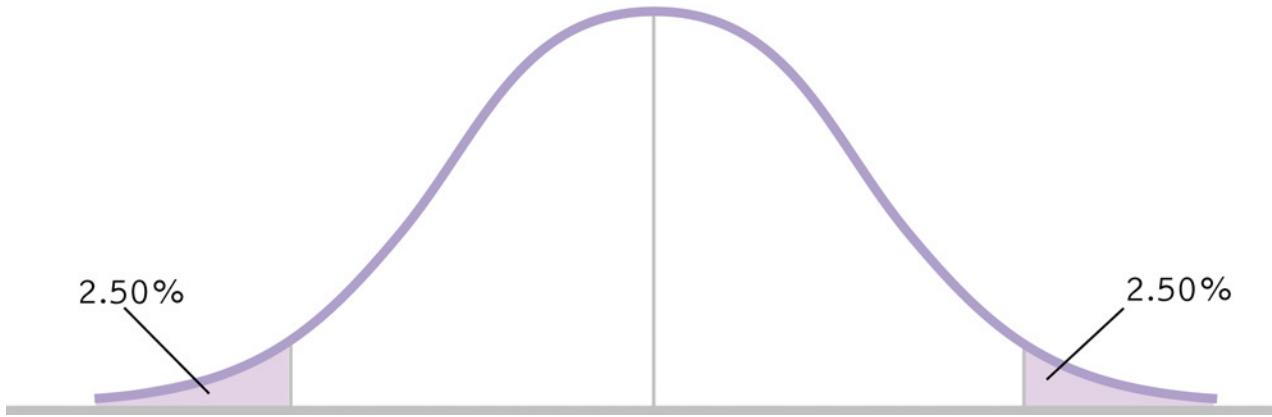
General Form of t-test

	Single sample, Paired/Dependent Samples t	Independent Samples t (same variance)
Statistic	$t = \frac{\bar{x} - \mu}{S_{\bar{x}}}$	$t = \frac{\bar{x} - \bar{y} - (\mu_x - \mu_y)}{s_{Difference}} = \frac{\bar{x} - \bar{y}}{\sqrt{s_{Pooled}^2 \left(\frac{1}{m} + \frac{1}{n} \right)}}$ $s_{Pooled}^2 = \frac{df_x s_x^2 + df_y s_y^2}{df_{Total}} = \left(\frac{df_x}{df_{Total}} \right) s_x^2 + \left(\frac{df_y}{df_{Total}} \right) s_y^2$
Critical value	$t_{\alpha, n-1}$	$t_{\alpha, m+n-2}$

One Tailed vs. Two Tailed Tests



$H_0: A=B$
 $H_a: A>B$
 $H_a: A<B$



$H_0: A=B$
 $H_a: A \neq B$

From two tailed test, you can only draw the conclusion that $A \neq B$, instead of $A>B$ or $A<B$ due to significant level.

Class Activity: Six Steps for Hypothesis Testing

1. Identify
2. State the hypotheses
3. Characteristics of the comparison distribution
4. Critical values
5. Calculate
6. Decide

One-Way ANOVA F-Test Hypotheses

- Tests the equality of 2 or more population means (μ)

$$H_0: \mu_1 = \mu_2 = \mu_3 = \dots = \mu_p$$

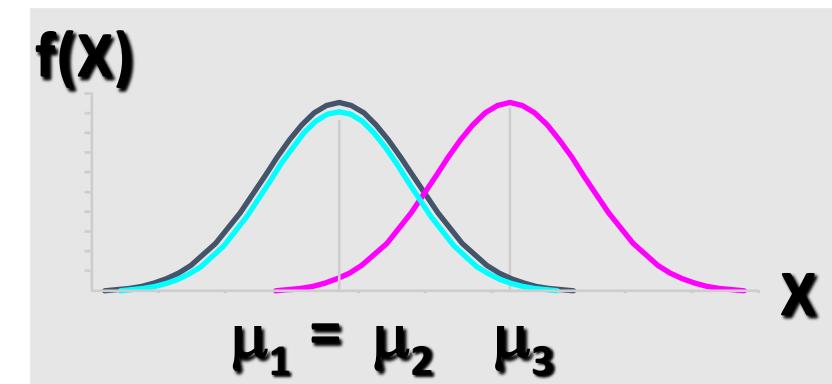
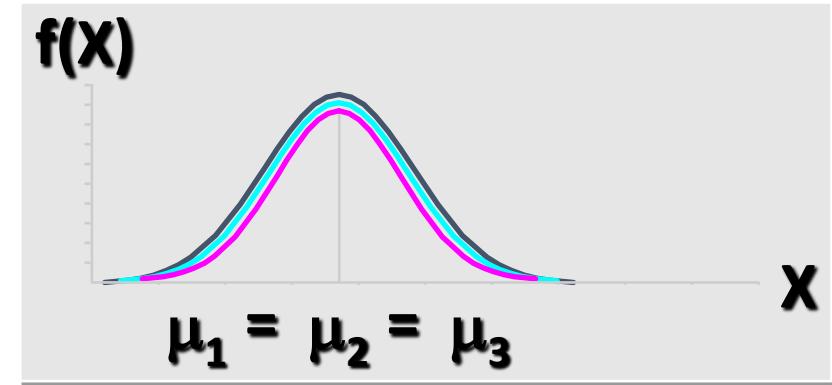
- All population means are equal
- No treatment effect

$$H_a: \text{Not All } \mu_j \text{ Are Equal}$$

- At least 1 population mean is Different
- Treatment Effect
- NOT $\mu_1 = \mu_2 = \dots = \mu_p$
- Or $\mu_i \neq \mu_j$ for some i, j .

- Treatment variation between groups could be significantly greater than the in group variation

- Variation measures are obtained by ‘Partitioning’ total variation



One-Way ANOVA F-Test: Test Statistic

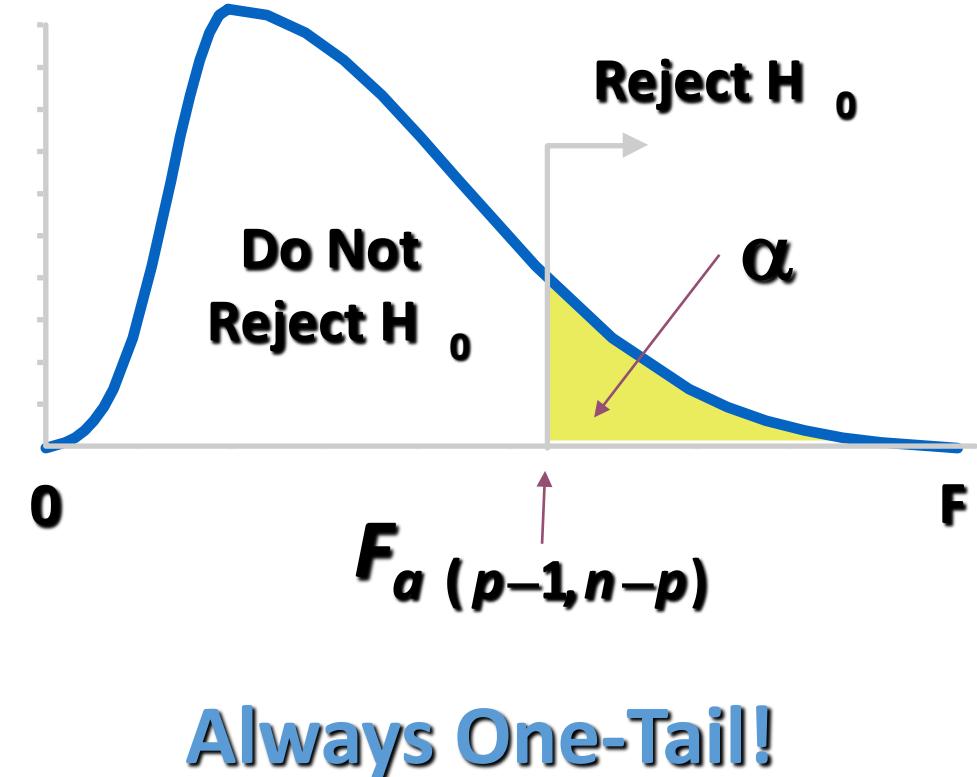
1. Test Statistic

$$\triangleright F = MST / MSE = \frac{SST / (p - 1)}{SSE / (n - p)}$$

- ❖ MST : mean square for treatment
- ❖ MSE : mean square for error

2. Degrees of Freedom

- $\nu_1 = p - 1$
- $\nu_2 = n - p$
 - ❖ p = Populations, Groups, or Levels
 - ❖ n = Total Sample Size



Post Hoc Tests

- If the t-test is significant, you have a difference in population means.
You know where.
- If the F-test is significant, you have a difference in population means.
But you don't know where.
- With 3 means, could be $A=B>C$ or $A>B>C$ or $A>B=C$.
 - ❖ We need a test to tell which means are different.

Tukey HSD(Honestly Significant Difference)

The sizes of three groups should be the same

➤ Step1: compute all possible absolute differences between means

$$\bar{Y}_1 = 24.93$$

$$\bar{Y}_1 - \bar{Y}_2 = 24.93 - 22.61 = 2.32$$

$$\bar{Y}_2 = 22.61$$



$$\bar{Y}_1 - \bar{Y}_3 = 24.93 - 20.59 = 4.34$$

$$\bar{Y}_3 = 20.59$$

$$\bar{Y}_2 - \bar{Y}_3 = 22.61 - 20.59 = 2.02$$

➤ Step2: find the critical value q

➤ Step3: compute HSD

$$HSD = \frac{\bar{Y}_i - \bar{Y}_j}{\sqrt{\frac{MSE}{n}}} = \frac{\bar{Y}_i - \bar{Y}_j}{\sqrt{\frac{0.9211}{5}}} = \frac{\bar{Y}_i - \bar{Y}_j}{0.43} \quad (\bar{Y}_i \text{ is always larger than } \bar{Y}_j)$$

➤ Step4: compare absolute differences with HSD

HSD(G1 to G2): $5.39 > 3.77$ **different each other**

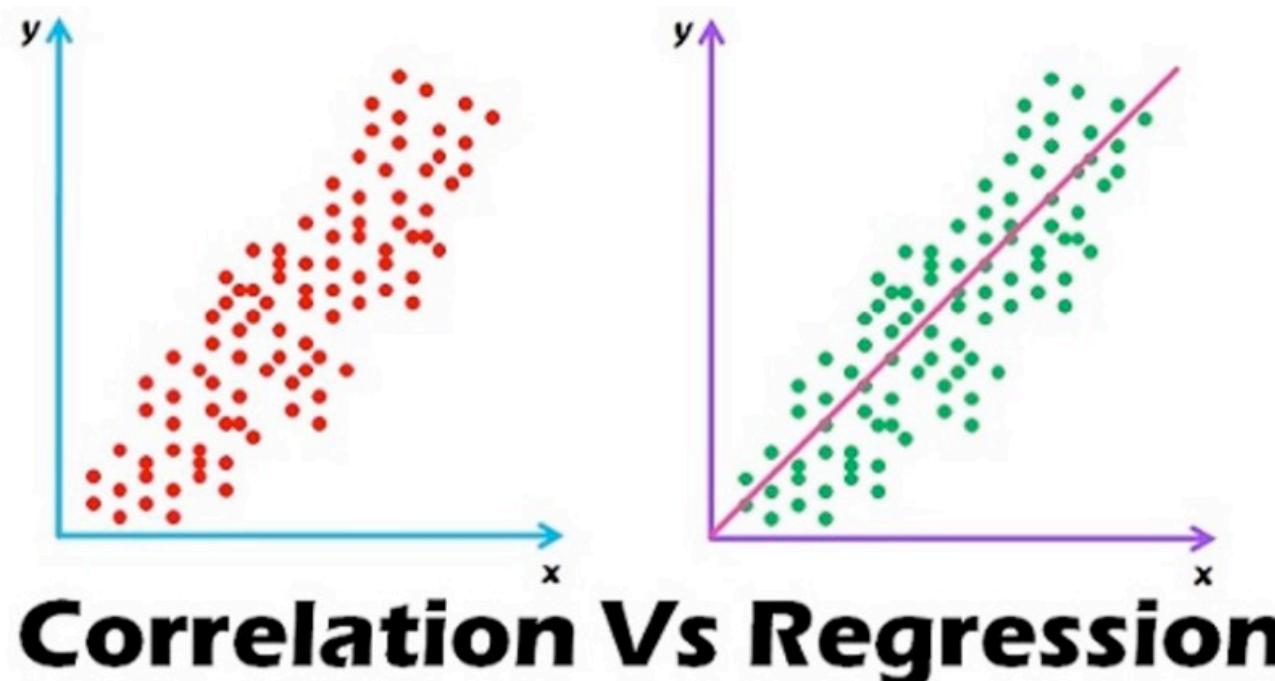
HSD(G1 to G3): $10.09 > 3.77$ **different each other**

HSD(G2 to G3): $4.70 > 3.77$ **different each other**

Critical Values for the Tukey Q Test									
Error df	Number of Groups (Treatments)								
	2	3	4	5	6	7	8	9	10
1	17.97	26.98	32.82	37.08	40.41	43.12	45.40	47.36	49.07
2	6.08	8.33	9.80	10.88	11.74	12.44	13.03	13.54	13.99
3	4.50	5.91	6.82	7.50	8.04	8.48	8.85	9.18	9.46
4	3.93	5.04	5.76	6.29	6.71	7.05	7.35	7.60	7.83
5	3.64	4.60	5.22	5.67	6.03	6.33	6.58	6.80	6.99
6	3.46	4.34	4.90	5.30	5.63	5.90	6.12	6.32	6.49
7	3.34	4.16	4.68	5.06	5.36	5.61	5.82	6.00	6.16
8	3.26	4.04	4.53	4.89	5.17	5.40	5.60	5.77	5.92
9	3.20	3.95	4.41	4.76	5.02	5.24	5.43	5.59	5.74
10	3.15	3.88	4.33	4.65	4.91	5.12	5.30	5.46	5.60
11	3.11	3.82	4.26	4.57	4.82	5.03	5.20	5.35	5.49
12	3.08	3.77	4.20	4.51	4.75	4.95	5.12	5.27	5.39
13	3.06	3.73	4.15	4.45	4.69	4.88	5.05	5.19	5.32
14	3.03	3.70	4.11	4.41	4.64	4.83	4.99	5.13	5.25
15	3.01	3.67	4.08	4.37	4.59	4.78	4.94	5.08	5.20
16	3.00	3.65	4.05	4.33	4.56	4.74	4.90	5.03	5.15
17	2.98	3.63	4.02	4.30	4.52	4.70	4.86	4.99	5.11
18	2.97	3.61	4.00	4.28	4.49	4.67	4.82	4.96	5.07
19	2.96	3.59	3.98	4.25	4.47	4.65	4.79	4.92	5.04
20	2.95	3.58	3.96	4.23	4.45	4.62	4.77	4.90	5.01
24	2.92	3.53	3.90	4.17	4.37	4.54	4.68	4.81	4.92
30	2.89	3.49	3.85	4.10	4.30	4.46	4.60	4.72	4.82
40	2.86	3.44	3.79	4.04	4.23	4.39	4.52	4.63	4.73
60	2.83	3.40	3.74	3.98	4.16	4.31	4.44	4.55	4.65
120	2.80	3.36	3.68	3.92	4.10	4.24	4.36	4.47	4.56
∞	2.77	3.31	3.63	3.86	4.03	4.17	4.29	4.39	4.47

Linear regression and Correlation

- Correlation coefficient: measure of the strength and direction of linear relationship between two quantitative variables
- Linear regression fits a model predicting a quantitative response (dependent) variable (y) based on a quantitative explanatory (independent) variable (x)



Pearson Correlation Coefficient

- The correlation coefficient (r) is a numeric measure of the strength and direction of a linear relationship between two quantitative variables

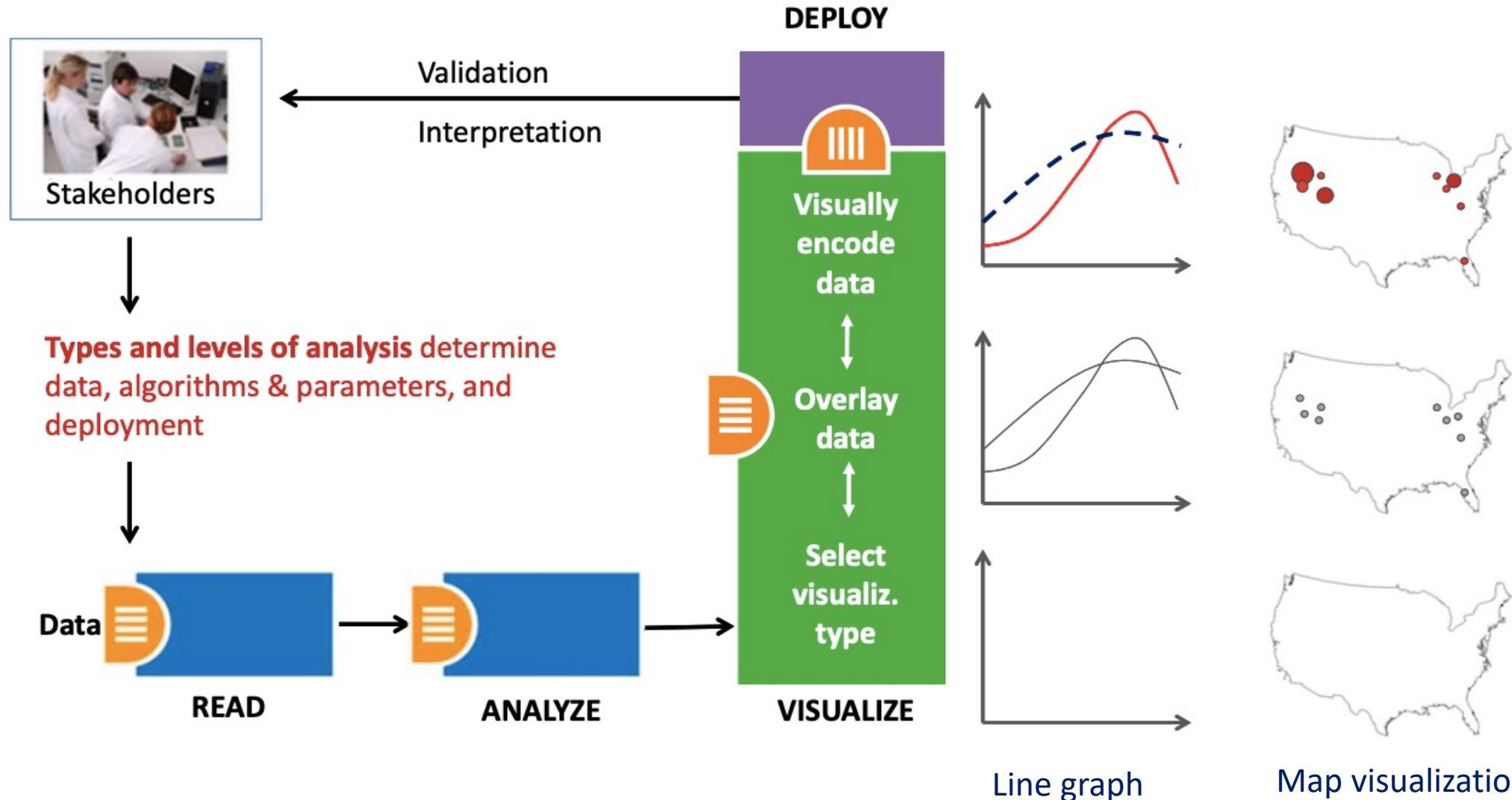
$$r_{XY} = \frac{\sum_{i=1}^n (X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum_{i=1}^n (X_i - \bar{X})^2} \sqrt{\sum_{i=1}^n (Y_i - \bar{Y})^2}}$$

Revision Guide

- **Design of Visualization**
 - Workflow of data visualization
 - Color, Size, Text, Titles, Labels
 - Choosing the right chart

Needs-Driven Workflow

Börner (2014) – Visual Insights



Shneiderman's Mantra

- User-Interface Interaction
 - Immediate interaction not only allows direct manipulation of the visual objects displayed but also allows users to select what to be displayed (Card et al., 1999)
 - Shneiderman (1996) summarizes six types of interface functionality
 - Overview
 - Zoom
 - Filtering
 - Details on demand
 - Relate
 - History
 - **“Overview first, zoom and filter, then details-on-demand.”**

Color Blind Friendly - Accessibility



Left: Normal vision, Right: Deutanopia

1. Avoiding **problematic** color combinations, e.g., red & green / green & brown / green & blue...
2. Selecting color-blind-friendly **palettes**
3. Using different **textures** and patterns to highlight important information—not just color
4. Using **symbols and icons** to supplement color-coded messages, warnings and alerts
5. Using highly **contrasted** color combinations
6. Adopting **minimalistic** design to help avoid unnecessary confusion.

Choosing the Right Chart

- When a single number is important
- How two or more numbers are alike or different
- How we are better or worse than a benchmark
- When there are parts of a whole
- How things change over time

Displaying a Single Number

- Simple text with a single large number

- Icon Array

- Donut or Pie Graph

- Bar

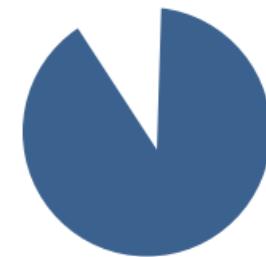
Big Number

23%

Icon Array



Pie/Donut

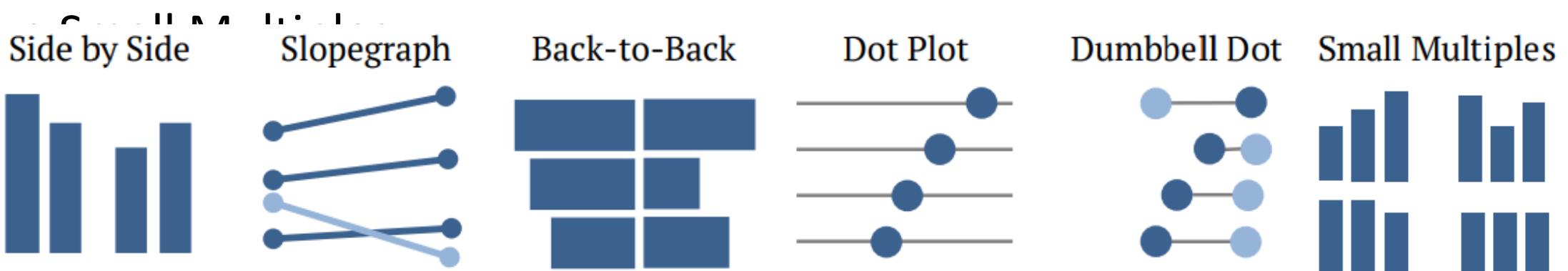


Bar/Column



Visualizing Comparisons

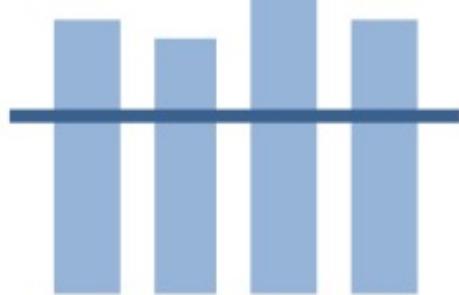
- Side by Side Column or Bar Charts
- Slopegraph
- Back-To-Back Bars
- Dot Plot
- Dumbbell Dot



Displaying Relative Performance

- Benchmark Line
- Combo chart
- Bullet Chart
- Indicator Dots

Benchmark Line



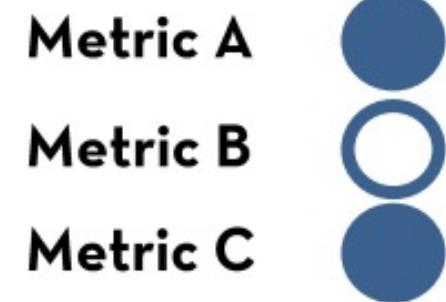
Combo



Bullet Chart



Indicator Dots



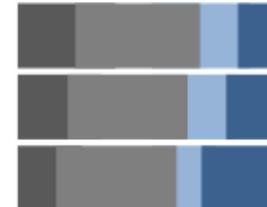
Visualize The Parts of a Whole

- Pie/Donut
- Stacked Bar
- Histogram
- Tree Map
- Map

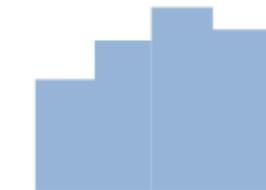
Pie/Donut



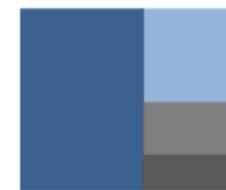
Stacked Bar



Histogram



Tree Map

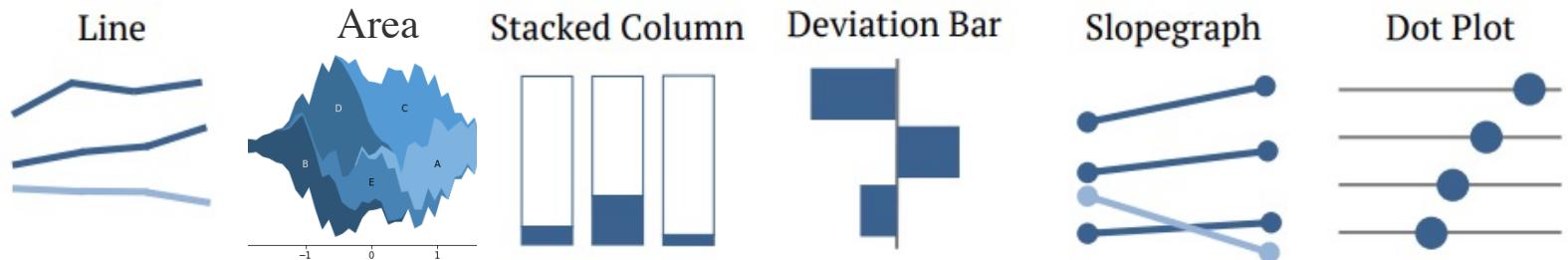


Map



Visualize How Things Changed Over Time

- Line Graph
- Area Graph
- Stacked Column
- Deviation Bar
- Slopegraph
- Dot Plot



Database

Revision Guide

Understand *Foreign Key* and *Primary Key*

Familiar with CREATE table

- Common Data Type: Char/varchar/integer/decimal/date
- NOT NULL
- CONSTRAINT
- PRIMARY KEY/FOREIGN KEY

Familiar with

- INSERT data
 - INSERT INTO....
- UPDATE data
 - UPDATE ... SET
- DELETE data
 - DELETE FROM ...

Revision Guide

- Simple Select Syntax
 - AND/OR
 - LIKE, IN
 - ORDER BY
 - COUNT/MIN/MAX/SUM

SQL for Data Retrieval: AND/OR examples

```
SELECT          empId, empName  
FROM            Employee  
WHERE           empId < 7 OR empId > 12;
```

5	Dan
19	Sheldon

<u>empId</u>	empName	salaryCode	deptId
5	Dan	3	9
9	Penny	4	9
19	Sheldon	5	10

```
SELECT          empId, empName  
FROM            Employee  
WHERE           deptId = 9 AND salaryCode <= 3;
```

5	Dan
---	-----

SQL for Data Retrieval: Wildcard Searches

- The SQL **LIKE** keyword allows for searches on partial data values
- LIKE** can be paired with wildcards to find rows that partially match a string value
 - The multiple character wildcard character is a percent sign (%)
 - The single character wildcard character is an underscore (_)

```
SELECT  
FROM  
WHERE empId  
Employee  
empName LIKE 'Da%';
```

18

29

29

```
SELECT  
FROM  
WHERE empId  
Employee  
phone LIKE '3411_ _ _ _';
```

Employee

<u>emplId</u>	empName	phone
18	Dan	69981245
29	Danny	34111001

SQL for Data Retrieval: Sorting the Result

- Querying results may be sorted using the ORDER BY clause
 - Ascending vs. descending sorts

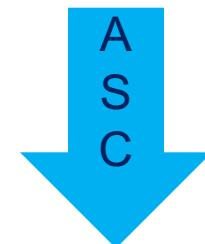
```
SELECT      *  
FROM        Employee  
ORDER BY    empName;
```

```
SELECT      *  
FROM        Employee  
ORDER BY    empName ASC;
```

```
SELECT      *  
FROM        Employee  
ORDER BY    empName DESC;
```

<u>empld</u>	empName
1	Dan
2	Penny
3	Sheldon

<u>empld</u>	empName
3	Sheldon
2	Penny
1	Dan



SQL for Data Retrieval: Built-in Function Examples

```
SELECT  
FROM
```

```
COUNT(*)  
Employee;
```

3

```
SELECT  
MIN(hours) AS minimumHours,  
MAX(hours) AS maximumHours,  
AVG(hours) AS averageHours  
FROM  
Project  
WHERE  
ProjectId > 7;
```

Employee

emplId	empName
10	Dan
15	Penny
29	Sheldon

minimumHours	maximumHours	averageHours
2	8	5

GROUP BY and HAVING

- Familiar with the syntax of GROUP BY
- Make sure GROUP BY appear after WHERE, HAVING after GROUP BY
- Knowing the different between HAVING and WHERE

SUB-QUERY

- Most problem that can be solved by SUB-QUERY can also be solved by joining table.
- Use join table instead!

JOIN TABLE

- Understand how the simple inner join works
- How to cross reference two or more tables
- Knowing the differences between inner join, left outer join, right outer join, full outer join.

Join ... ON Example

SELECT
FROM
WHERE

```
empName, deptName
Employee e INNER JOIN Department d
ON e.deptId = d.deptId
d.deptName NOT LIKE 'Account%';
```

Employee

<u>empld</u>	empName	deptId
1	Dan	102
2	Penny	101
3	Sheldon	



empName	deptName
Dan	Customer Service
Penny	Account Service

Department

<u>deptId</u>	deptName
101	Account Service
102	Customer Service



Dan	Customer Service
-----	------------------

Quick Question

- Suppose deptID in Employee table now becomes a foreign key. i.e., must have a value and that value must appear in the table Department
 - Which of the following table joining are equivalent?
 - Employee inner join Department
 - Employee left outer join Department
 - Department left outer join Employee
 - Employee right outer join Department
 - Employee full outer join Department

Department

<u>deptId</u>	deptName
101	Research
102	Customer Service
103	Marketing

Employee

<u>empld</u>	empName	<i>deptId</i>
1	Dan	102
2	Penny	101
3	Sheldon	103

Security and Privacy

Revision Guide

- Knowing the difference between public key encryption and private key encryption
- Understanding the logic behind SQL injection
- Familiar with the four different types of indirection questioning techniques
- Able to identify EI/QI/SD/NSD
- Understand the basic concept of K-Anonymity.
 - Be able to identify a given data set is or isn't K-Anonymous.