INMAS

Linear Regression

- *Learning Types*
- *Linear Regression*

James Balamuta

INMAS Statistical Methods Workshop Fall 2021

Lecture Objectives

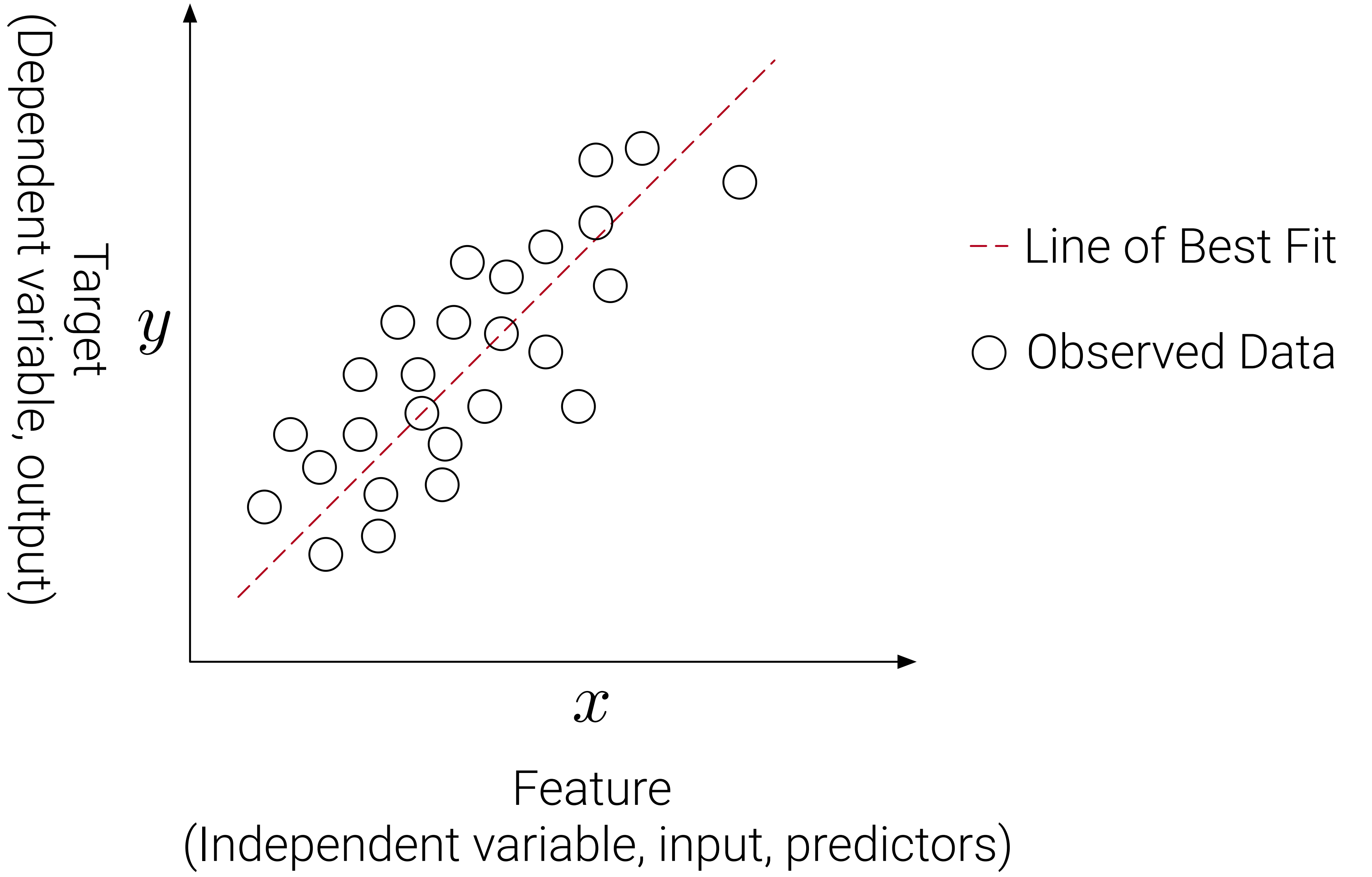
- *Emphasize* differences between Supervised Learning and Unsupervised Learning
- *Discuss* linear regression.

Learning Types

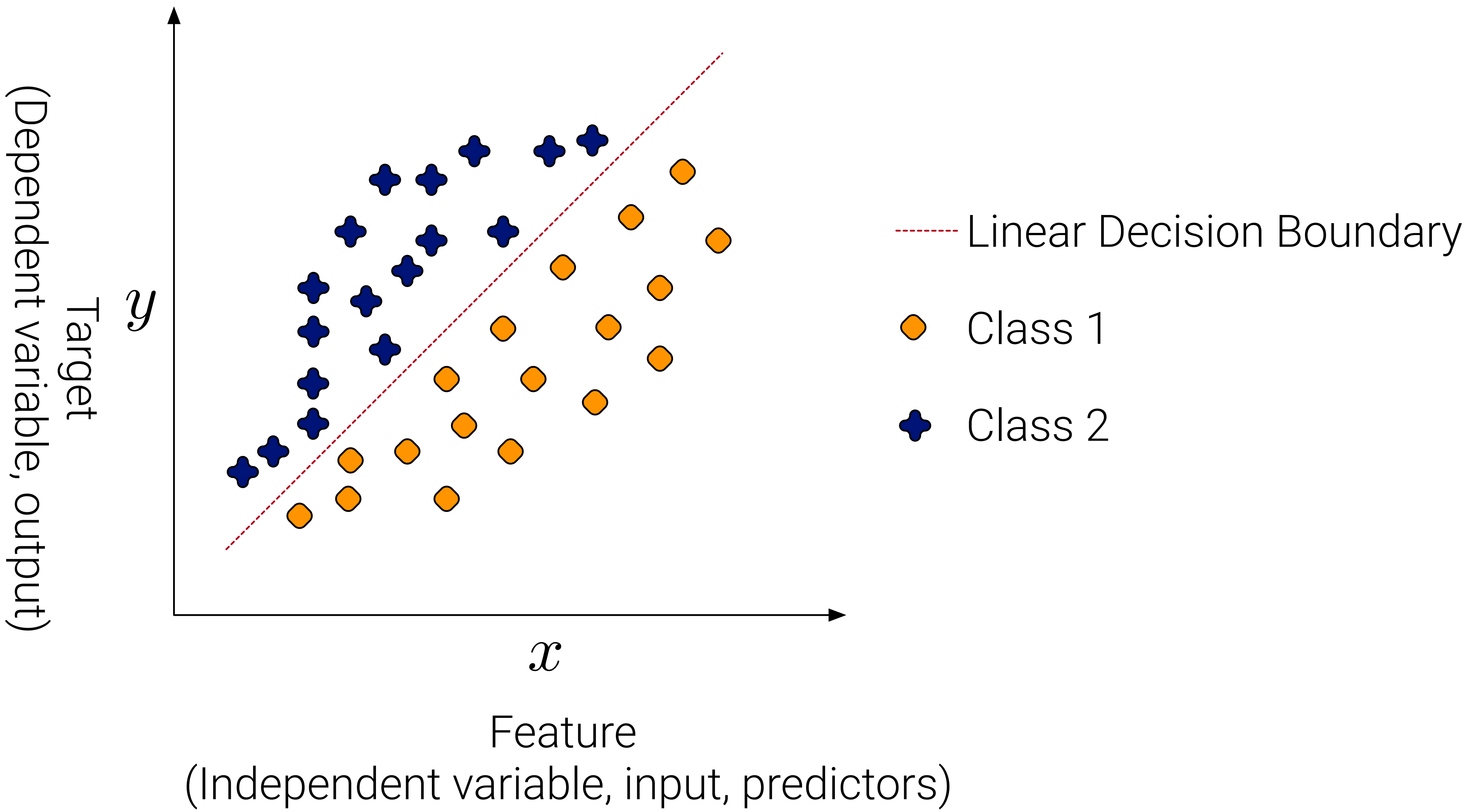
Learning Types

- **Supervised Learning** with Labeled Data. (Today)
  - Methods: Regression or classification
  - Objective: To predict a response or outcome.
- **Unsupervised Learning** with Unlabeled Data.
  - Methods: Clustering, Principle Component Analysis (PCA), autoencoders, generative adversarial networks (GANs)
  - Objective: Identify patterns in the data or understand how data was created.
- Best distinction between the two:
  - **Is there a response variable Y?**

Supervised Learning: Regression

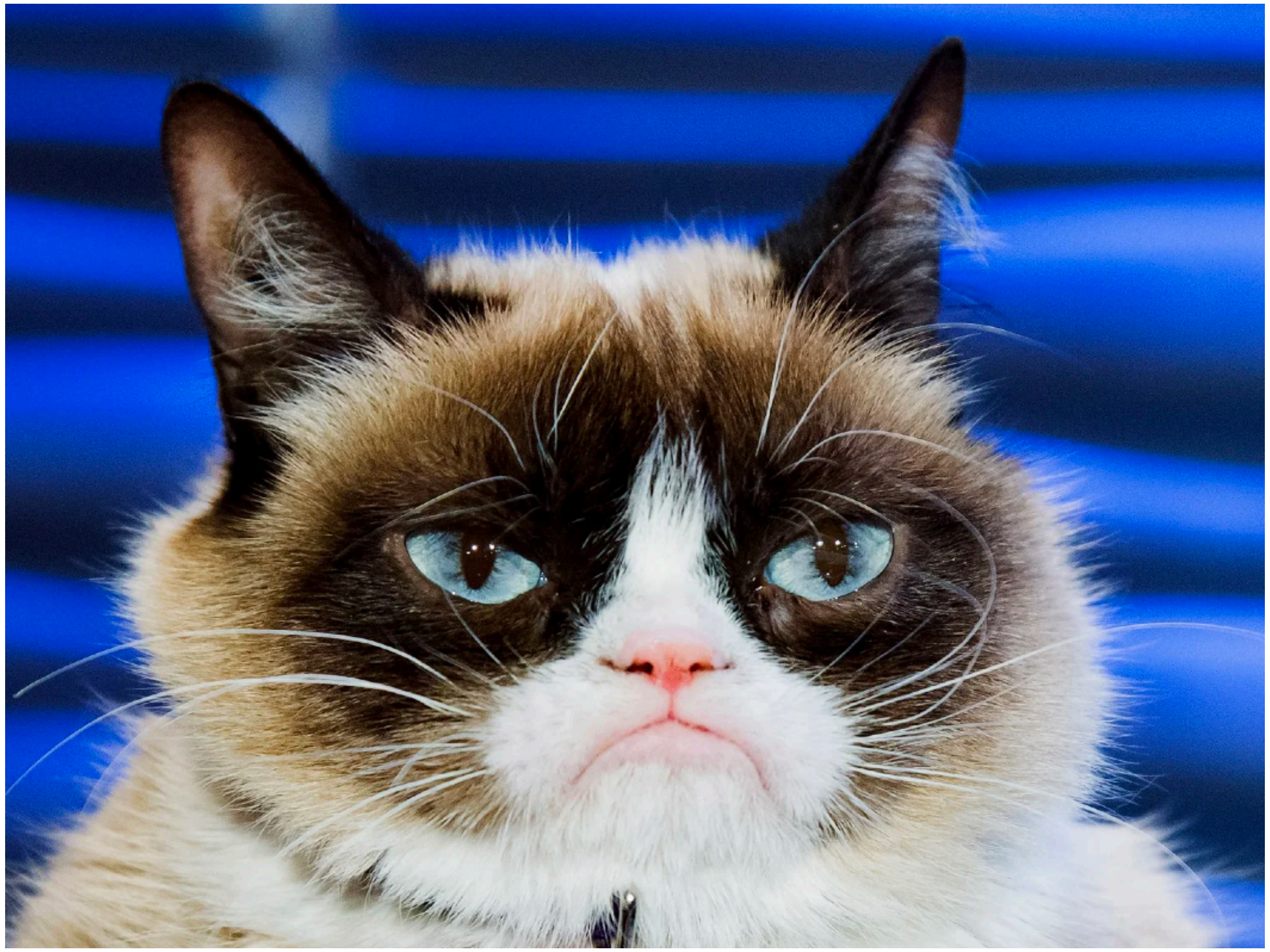


Supervised Learning: Classification



Is Dog?

380



506

Image Source

Image dimensions: (380, 506, **3**)

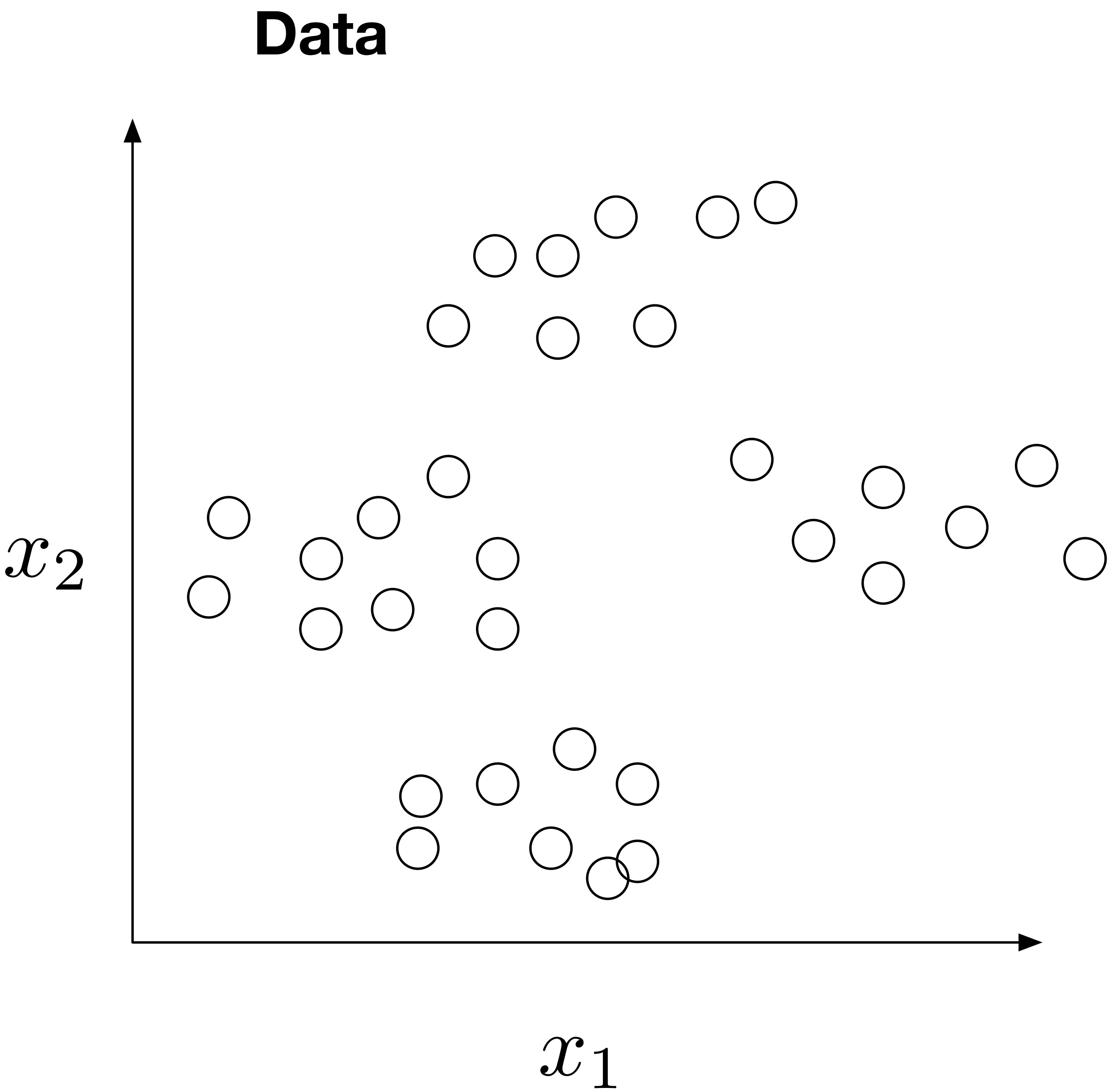
Is Dog?

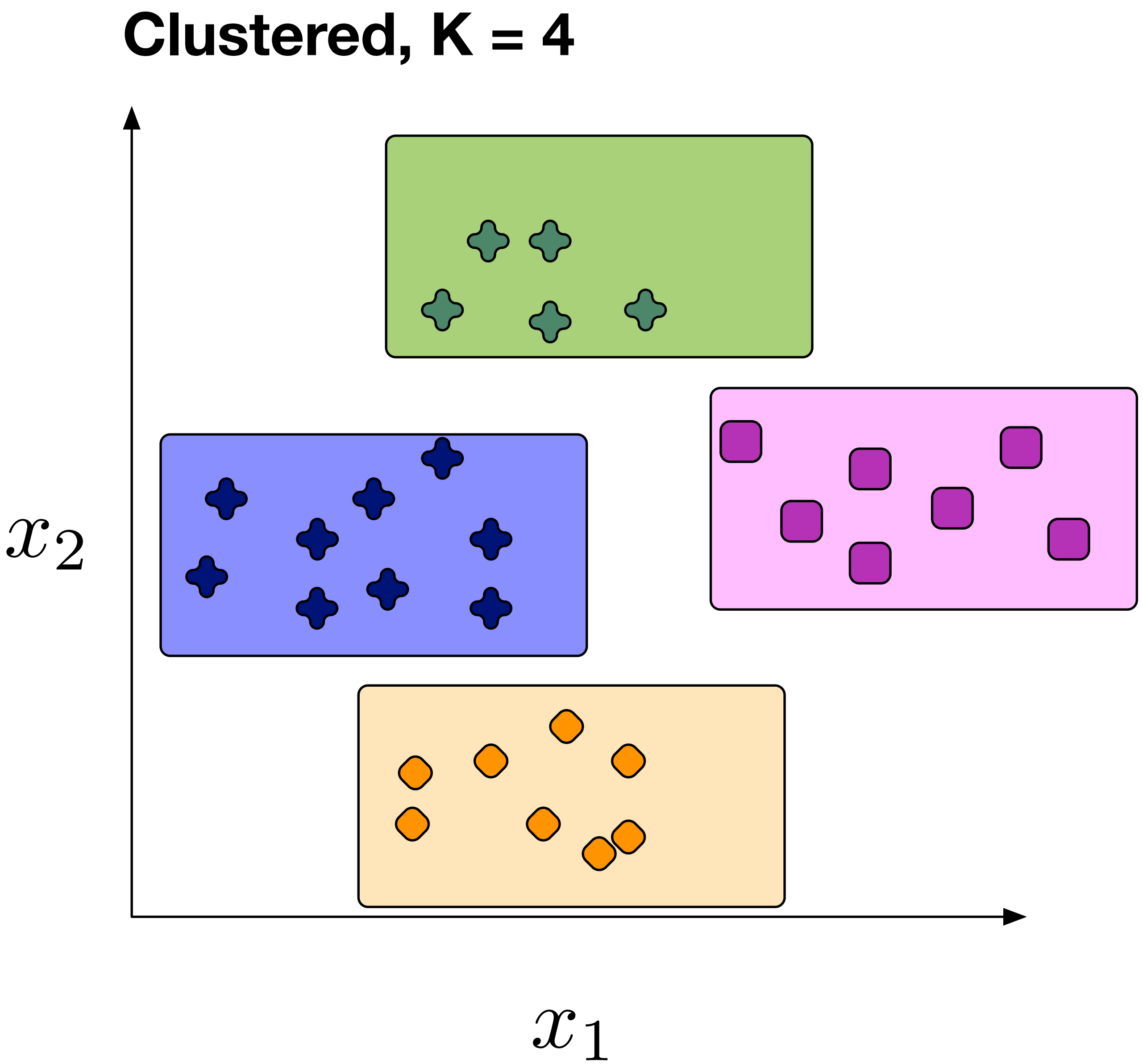
965

800 Image Source

Image dimensions: (965, 800, **3**)

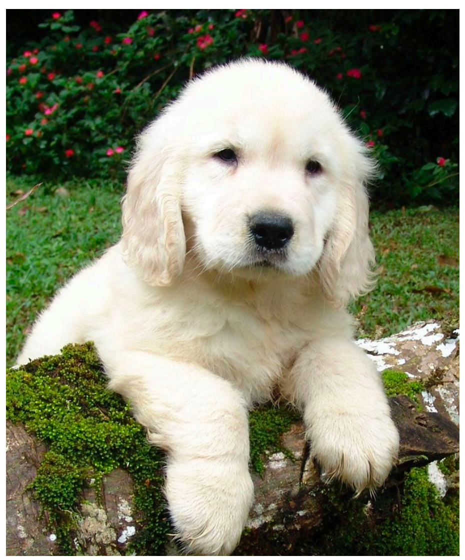
Unsupervised Learning: Clustering





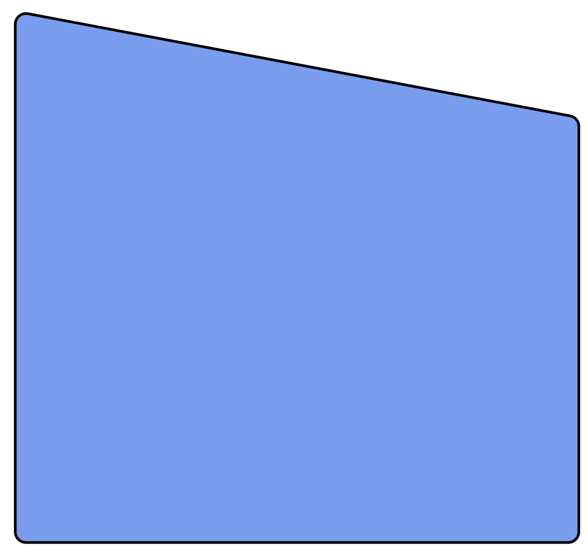
Clustering **is not** Classification

x



y

Dog



$$P(Y = \text{Dog})$$

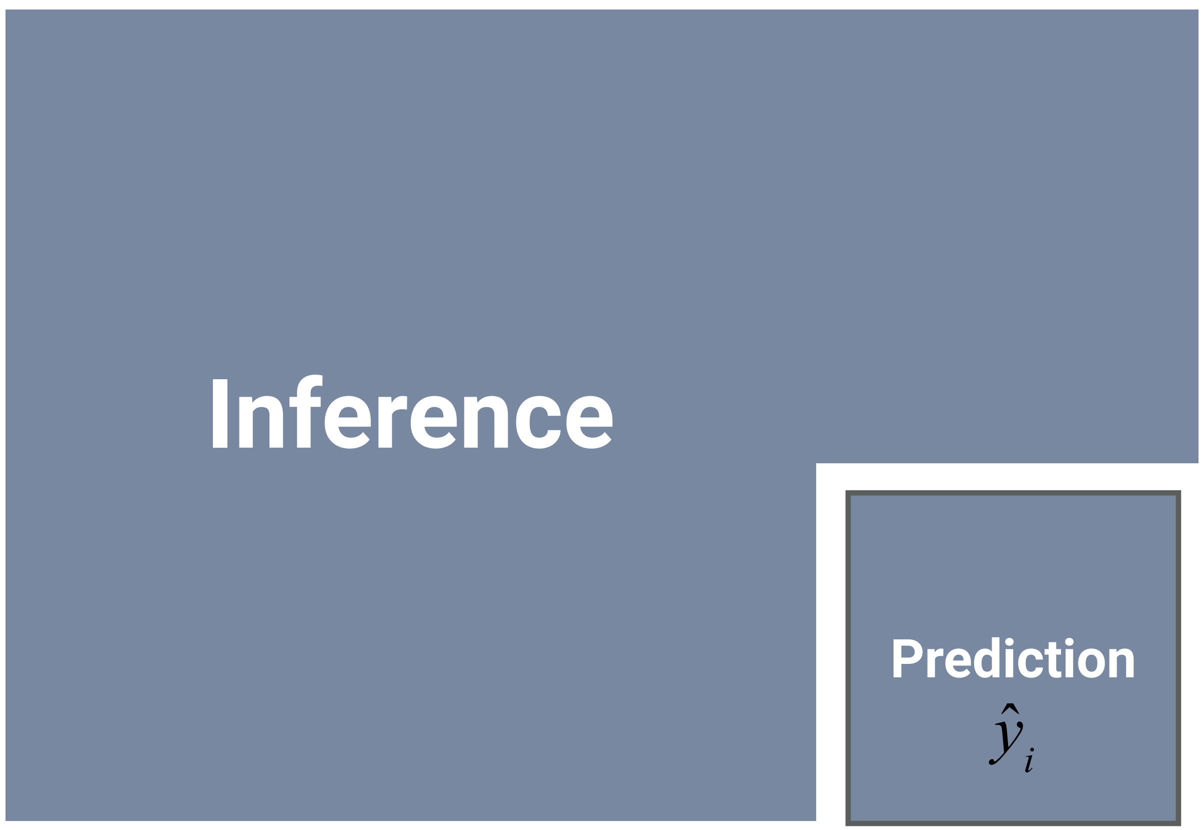
Data **Model** **Output**

Linear Regression

Viewpoints

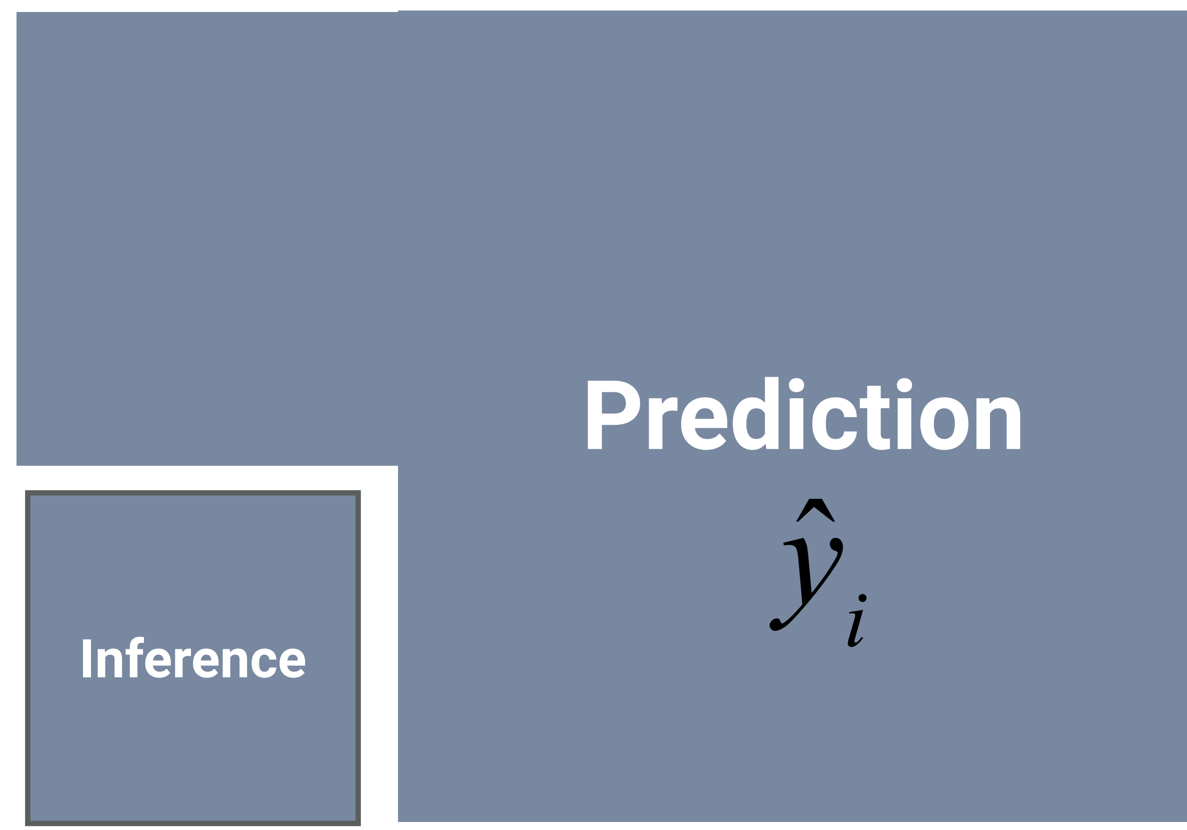
... inference or predictions ...

Statistician



How good are ...
... the predictors selected?
... the diagnostic plots?

Machine Learning



How good are ...
... future predictions?

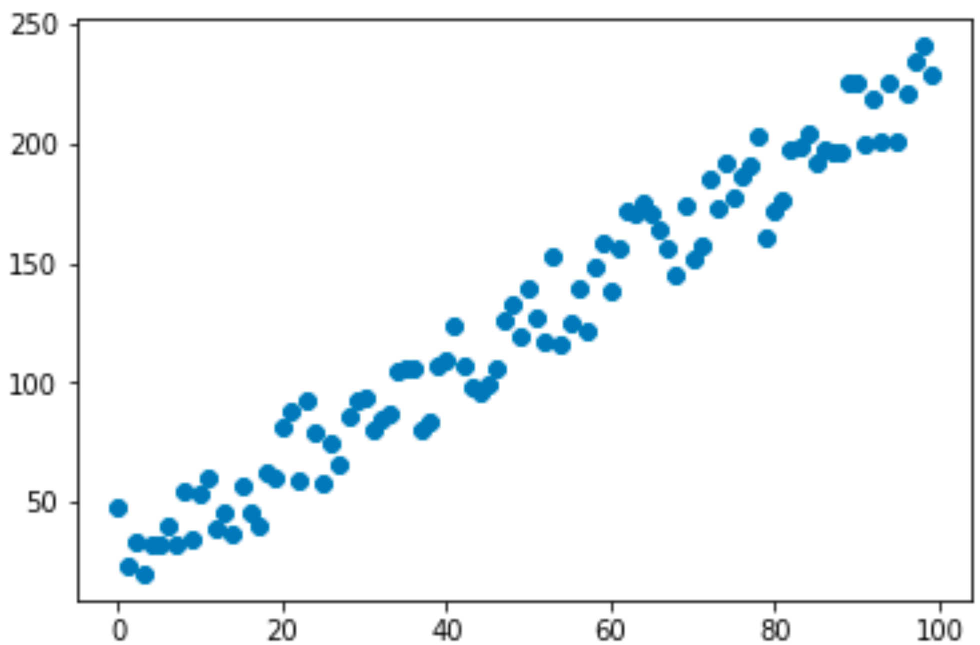
$$Y = f(X) + \varepsilon$$

True Response

Unknown Relationship

Unlearnable Noise

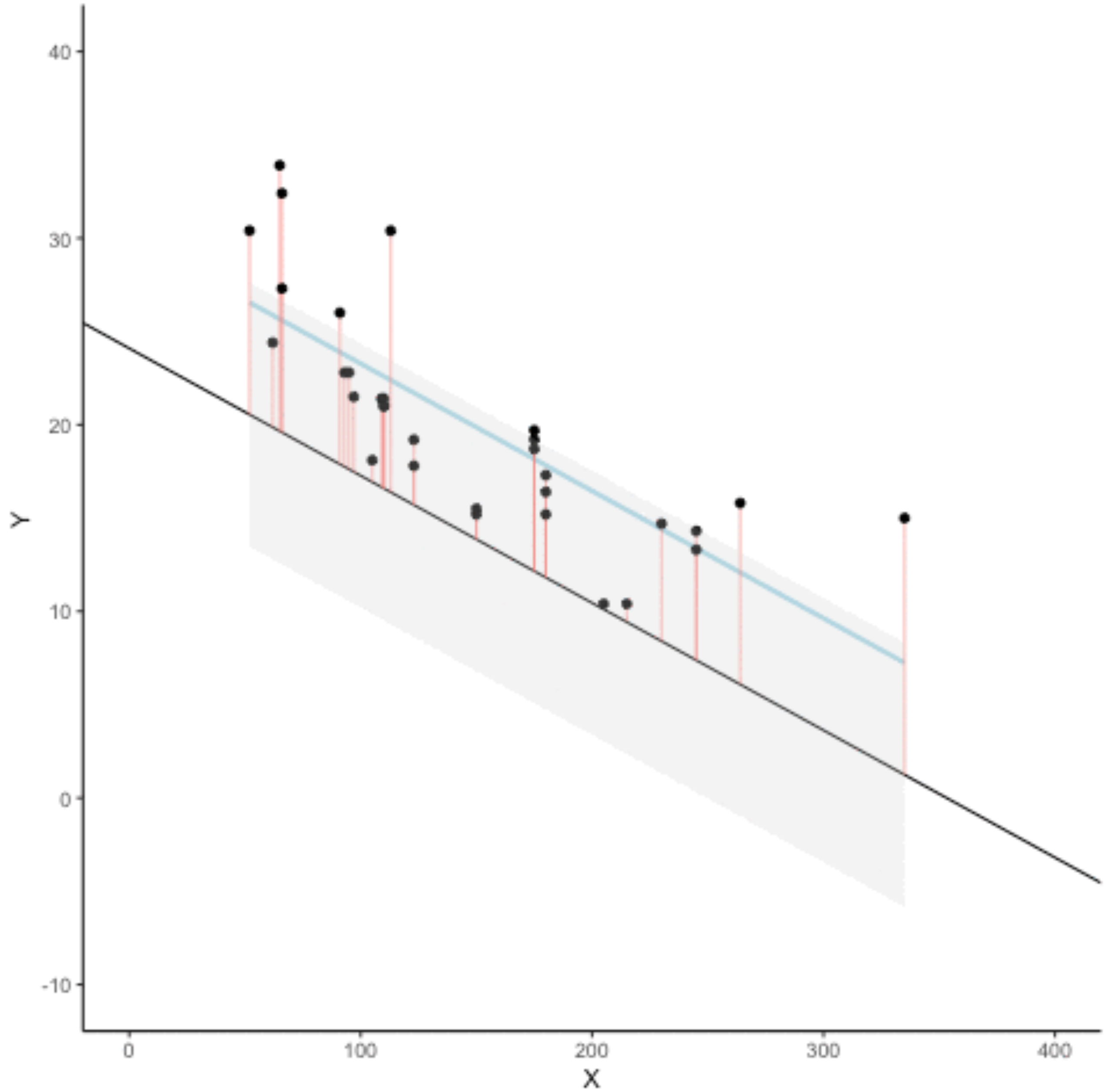
Simulating Data




```
import numpy as np
import matplotlib.pyplot as plt

# Set parameters
theta_0 = 3
theta_1 = 2
n = 100

# Generate design matrix
X = np.arange(0, n, 1)

# Create relationship
Y = theta_1*X + theta_0

# Add error
error = 45*np.random.rand(n)
Y = Y + error

# Show graph
plt.scatter(X, Y)
plt.show()
```

Simple Linear Regression

Line of best fit between two variables



Source

---

*   The **blue** line represents the optimal line of best fit.

**   The **black** line represents the current line of fit.

*** The **red** lines represent distance from points. The goal is to *minimize* these values.

Simple Linear Regression

Mathematical formulation

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$

$$\underbrace{\begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}_{n\times 1}}_{\text{Responses}} = \underbrace{\begin{pmatrix} 1 & x_{1,1} \\ \vdots & \vdots \\ 1 & x_{n,1} \end{pmatrix}_{n\times 2}}_{\text{Design Matrix}} \underbrace{\begin{pmatrix} \beta_0 \\ \beta_1 \end{pmatrix}_{2\times 1}}_{\text{Parameters}} + \underbrace{\begin{pmatrix} \varepsilon_1 \\ \vdots \\ \varepsilon_n \end{pmatrix}_{n\times 1}}_{\text{Error}}$$

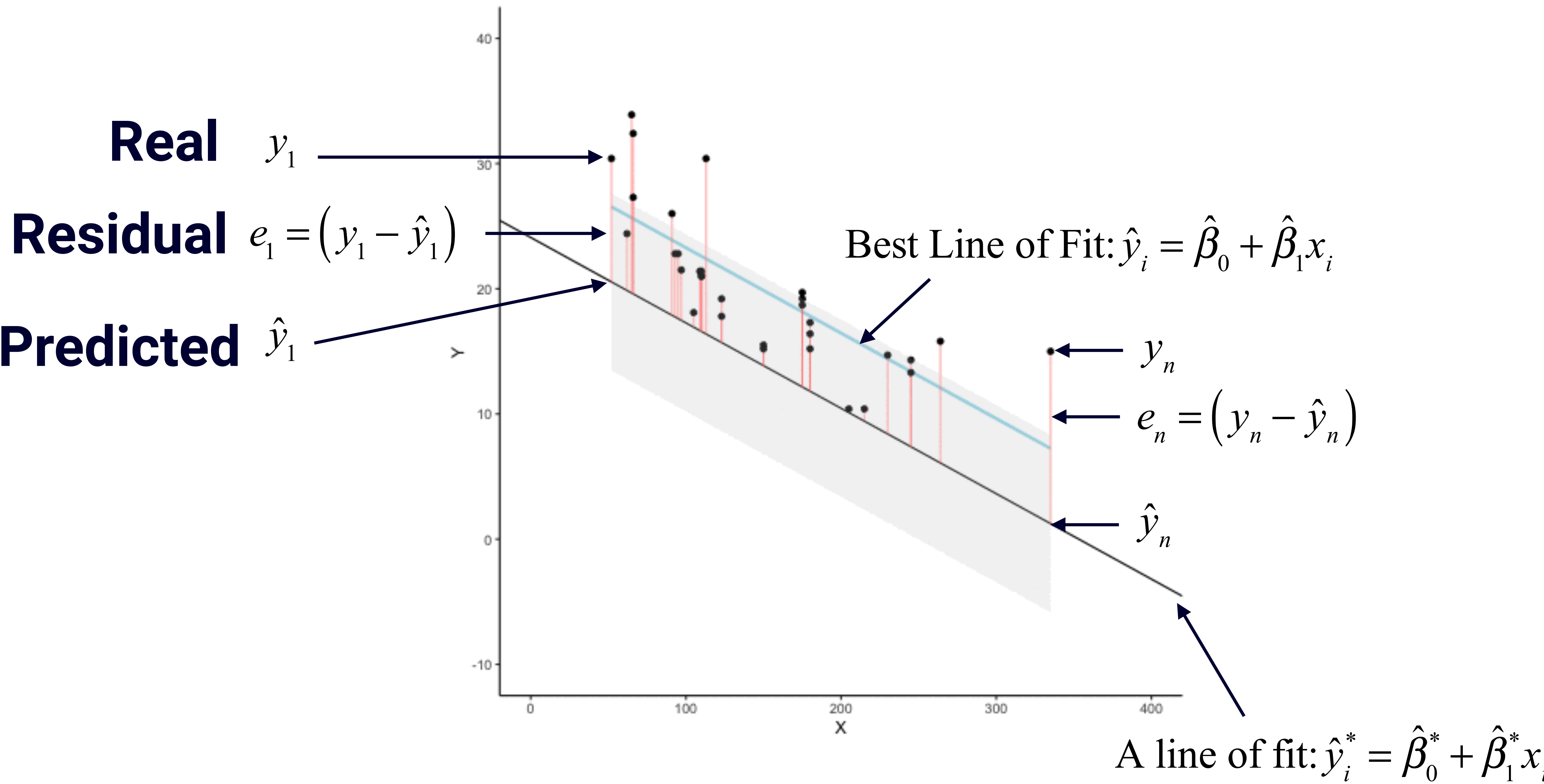
$$Y_{n\times 1} = X_{n\times 2}\beta_{2\times 1} + \varepsilon_{n\times 1}$$

---

n is the number of observations, there are 2 variables, ***X*** provides the design matrix for the variables, ***y*** is response vector, β is the parameter or coefficient vector and $\boldsymbol{\varepsilon}$ is the random error vector

SLR

Components



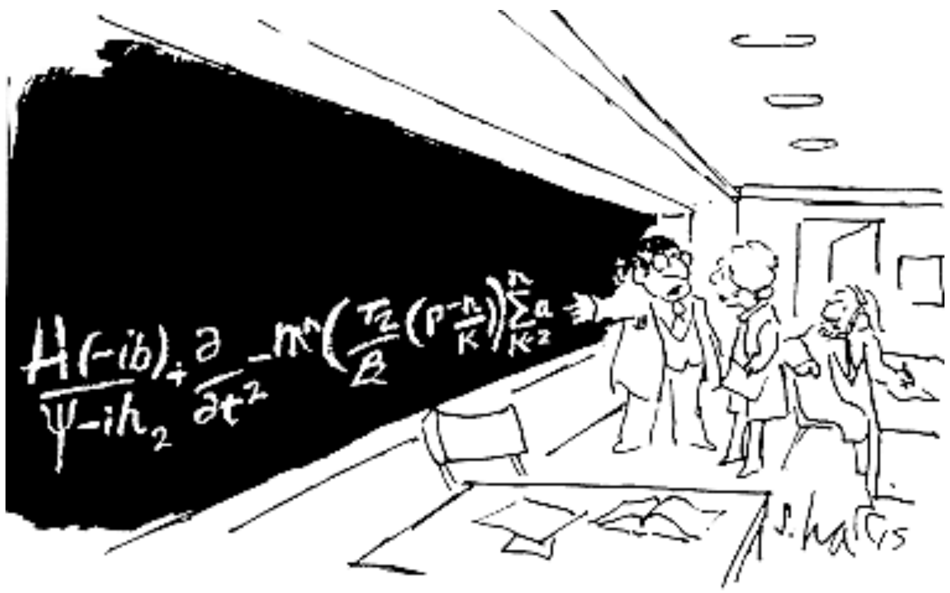
Definition:
A *closed-form* or *analytical expression* is a formula that provides an answer to a mathematical statement involving infinity that is finite.

The **Quadratic Formula** given by

$$0 = ax^2 + bx + c$$

has a solution of

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$



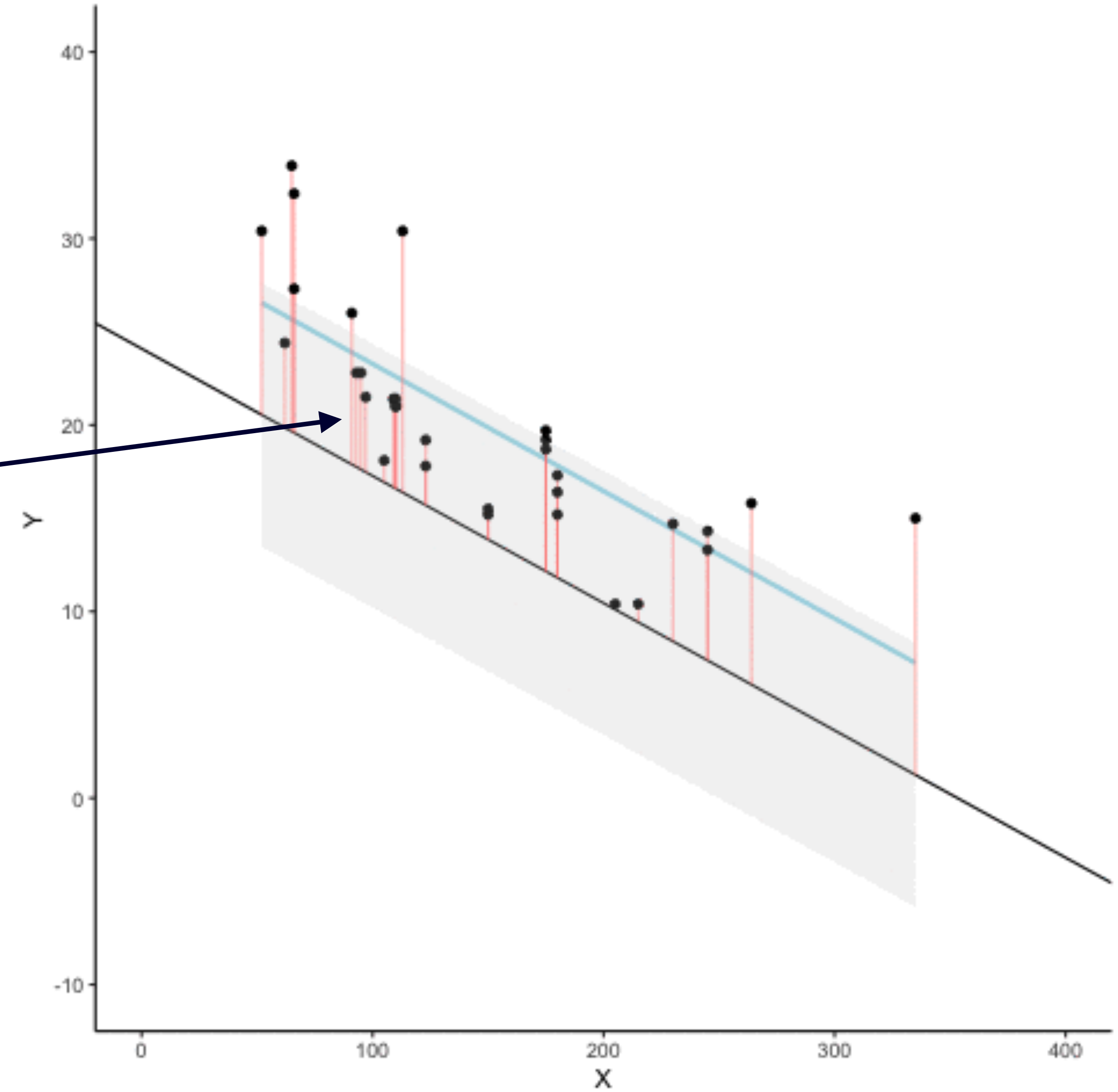
"But this *is* the simplified version for the general public."

Sidney Harris

Estimating Parameters

Minimize the <u>R</u>esidual <u>S</u>um <u>S</u>quared (red lines)

$$\hat{\beta} = \underset{\beta_0, \beta_1 \in \mathbb{R}}{\arg\min} \sum_{i=1}^{n} \left(y_i - \underbrace{\left(\beta_0 + \beta_1 x_i \right)}_{\hat{y}_i} \right)^2$$



Analytical Solutions

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} = \frac{S_{xy}}{S_{xx}}$$

$$\bar{x} = \frac{1}{n} \sum_{i=1}^{n} x_i \qquad \bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$$

Through the Mean

The closed-form solution for the intercept is given as:

$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

As a result, we note that **every linear regression line** must go through means of x and y

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \qquad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

Normal **Equation**

$$\hat{\boldsymbol{\beta}} = \left(\boldsymbol{X}^\top \boldsymbol{X}\right)^{-1} \boldsymbol{X}^\top \boldsymbol{y}$$

- For more than two parameters, the solution to linear regression is given by the normal equation.
- This is a closed-form solution that works well as it is a well-studied problem.

Extra

Aside

Notations

Training Set $\mathcal{D} = \left\{ \langle \boldsymbol{x}^{(i)}, y^{(i)} \rangle, i = 1, \ldots, n \right\}$

Unknown Function $f(\boldsymbol{x}) = y$

Hypothesis $h(\boldsymbol{x}) = \hat{y}$

$h : \mathbb{R}^m \rightarrow \mathcal{Y}, \mathcal{Y} = \{1, \ldots, k\}$

Classification

$h : \mathbb{R}^m \rightarrow \mathbb{R}$

Regression

Aside

Regression Hypothesis

Simple Linear Regression

$$
\begin{aligned}
h(\boldsymbol{x}) &= \beta_0 + \beta_1 x_1 \\
&= \beta_0 x_0 + \beta_1 x_1, \text{where } x_0 = 1 \\
&= \sum_{i=0}^{p} \beta_i x_i \\
&= \boldsymbol{\beta}^T \boldsymbol{x}
\end{aligned}
$$

Training

$$\mathcal{D} = \left\{ \langle \boldsymbol{x}^{(i)}, y^{(i)} \rangle, i = 1, \ldots, n \right\}$$

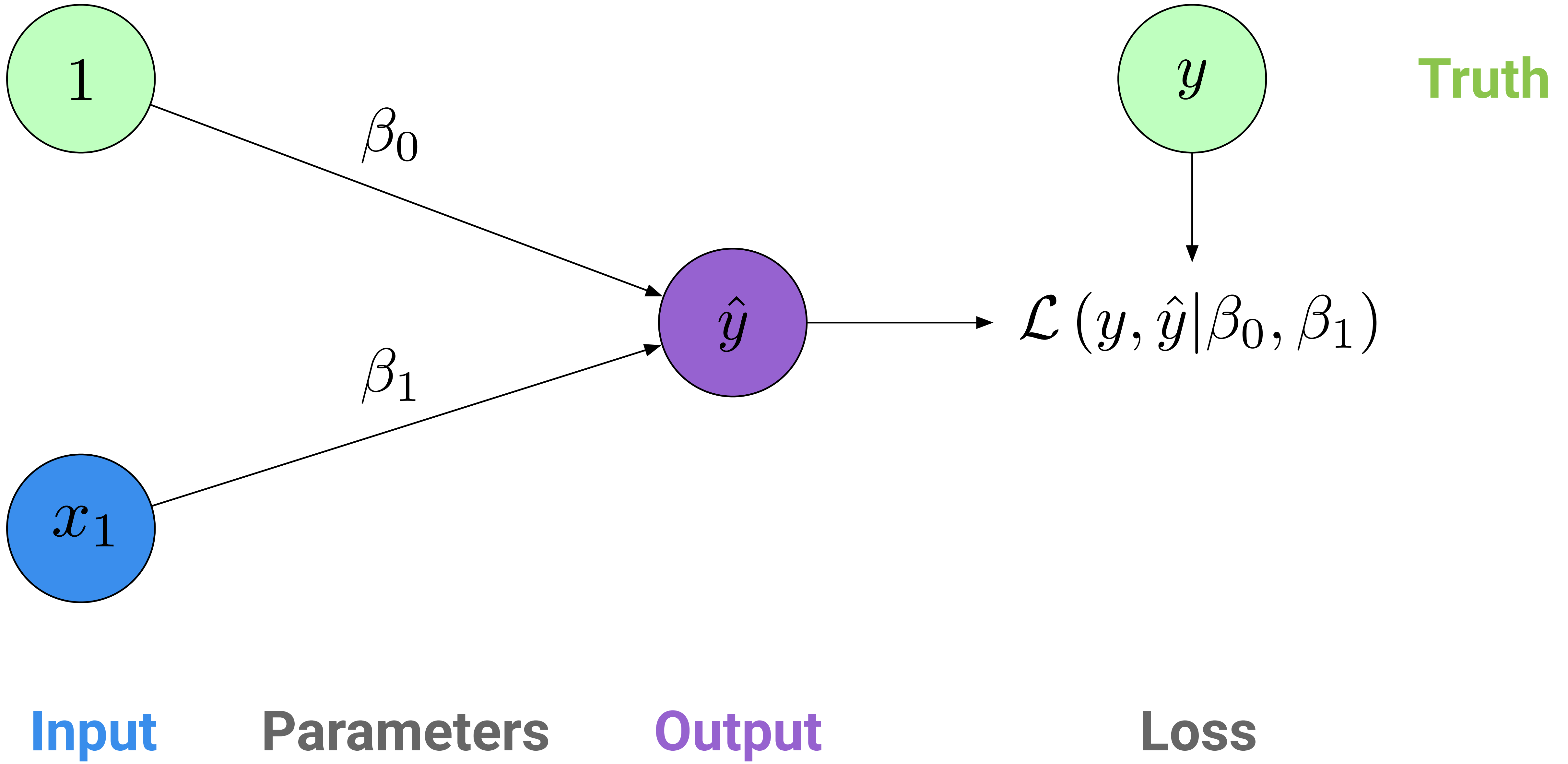
Unknown Function

$$f(\boldsymbol{x}) = y$$

Computational Graph

SLR shown in the context of a graph

$$y_i = \beta_0 + \beta_1 x_i + \varepsilon_i$$



Aside

Organizing Data

Row-major

$$\mathbf{y}_{1\times n} = \boldsymbol{\beta}_{1\times 2}\mathbf{X}_{2\times n}$$

$$\boldsymbol{\beta}_{1\times 2} = \begin{bmatrix}\beta_0 & \beta_1\end{bmatrix}_{1\times 2}$$

$$\mathbf{X}_{2\times n} = \begin{bmatrix} 1 & \cdots & 1 \\ \mathbf{x}^{(1)} & \cdots & \mathbf{x}^{(n)} \end{bmatrix}_{2\times n}$$

Rows are **features**
Columns are **observations**

Column-major

$$\mathbf{y}_{n\times 1} = \mathbf{X}_{n\times 2}\boldsymbol{\beta}_{2\times 1}$$

$$\boldsymbol{\beta}_{2\times 1} = \begin{bmatrix}\beta_0 \\ \beta_1\end{bmatrix}_{2\times 1}$$

$$\mathbf{X}_{n\times 2} = \begin{bmatrix} 1 & \mathbf{x}^{(1)} \\ \vdots & \vdots \\ 1 & \mathbf{x}^{(n)} \end{bmatrix}_{n\times 2}$$

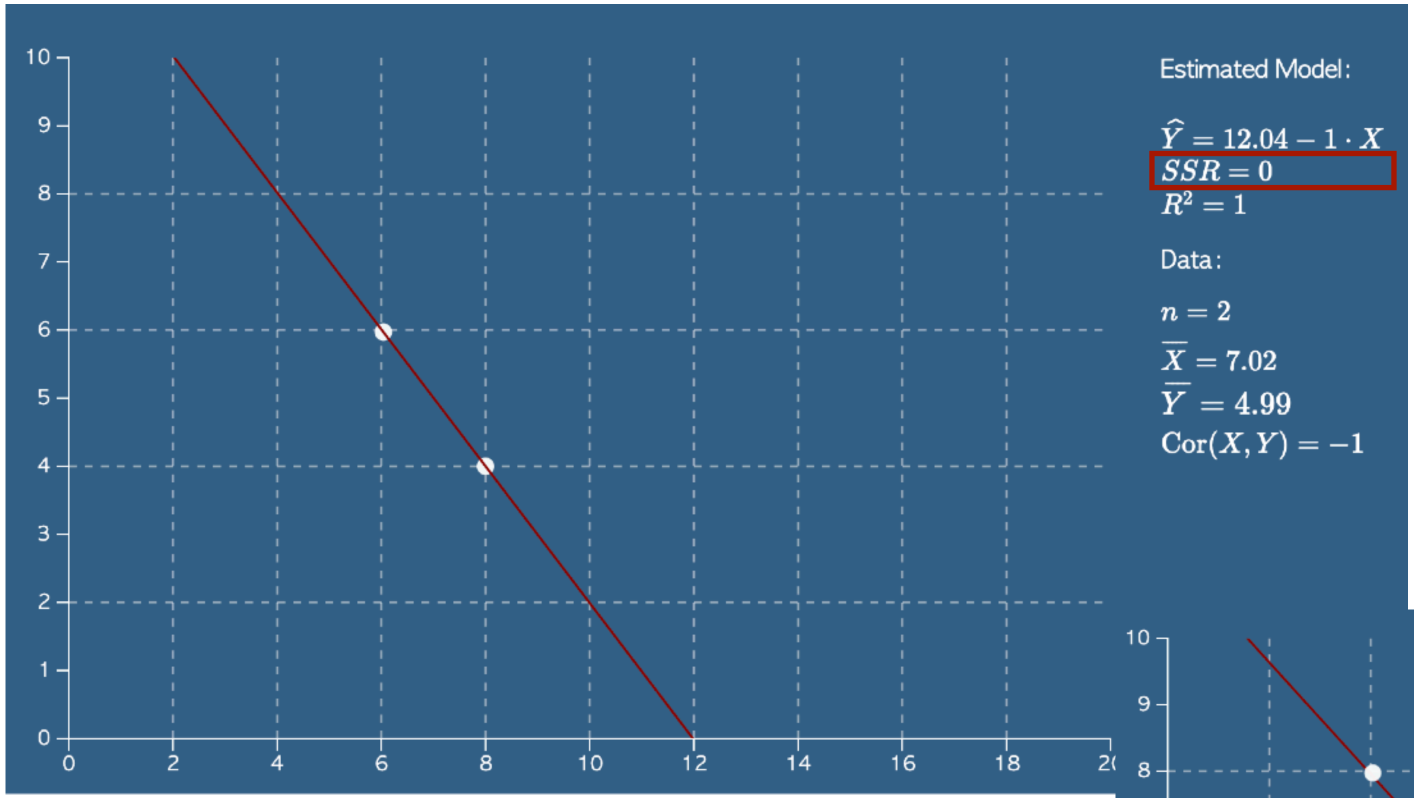
Rows are **observations**
Columns are **features**

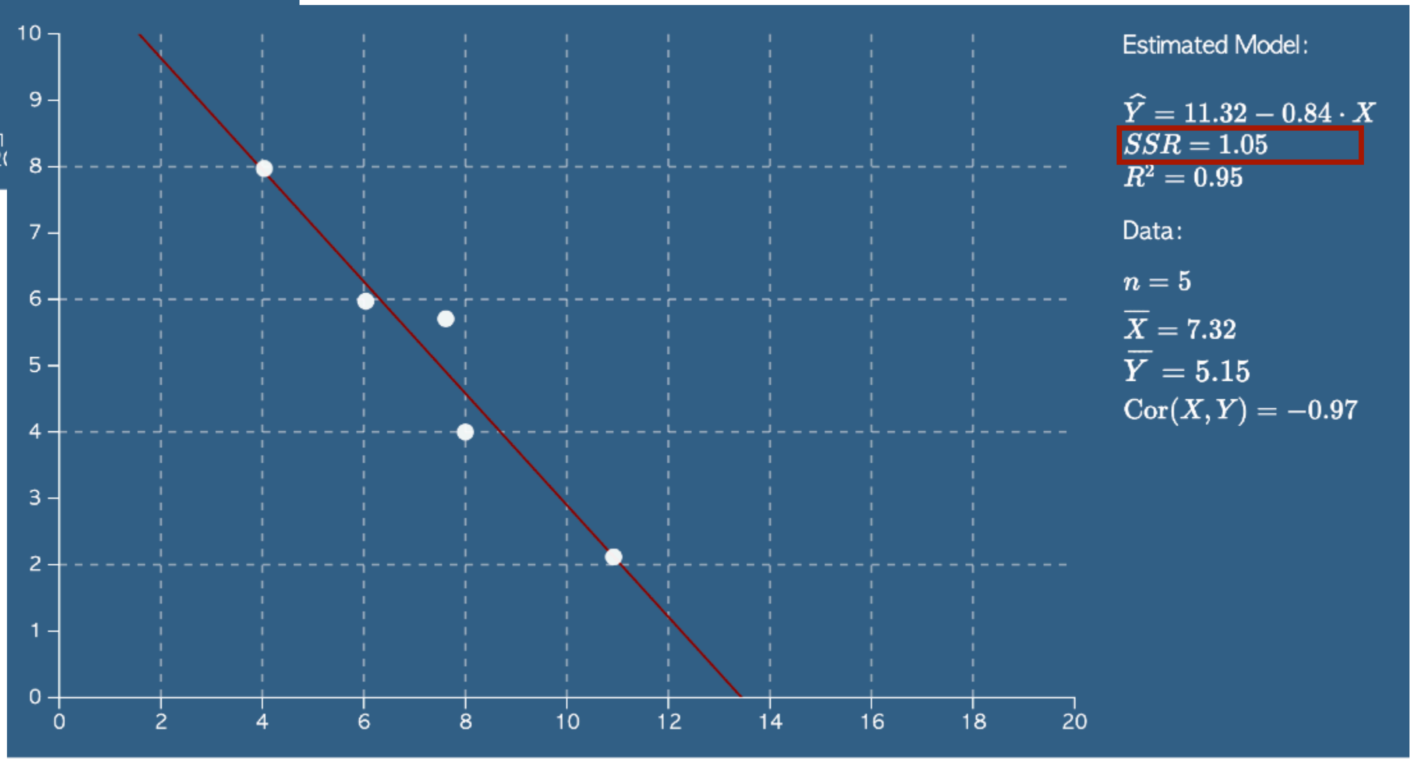
NumPy, TensorFlow, PyTorch uses row-major form

Aside

Dynamic SLR

Observing how data changes affect estimates





Interactive Web Widget

Aside

Jargon/Terms across Fields

Machine + Deep Learning	Statistical Learning	Description
Network, Graphs	Model	Assumption on how things work
Label, Target, Y	Dependent, Response, Output	Value being predicted
Feature, X	Independent, Explanatory, Input	Used to make predictions
Feature Engineering	Data Wrangling/Transformation	Changing the data to get more predictions
Learning	Fitting	Creating the model
Generalization	Test Set Performance	How the model works with new data
Weights	Parameters	Learned values
1D, 2D, 3D, …, nD	Dimensionality	Number of variables
Supervised Learning	Regression/Classification	Learning with a target
Unsupervised Learning	Density Estimation/Clustering	Learning without a target
Large grant: $1 million	Large grant: $50k	Statisticians are bad marketers
Conferences: French Alps	Conferences: Las Vegas in August	ML community goes on vacations

An extension of Rob Tibshirani's Glossary

Summary

- Differentiated between supervised and unsupervised learning.
- Discussed the closed-form solution to regression.