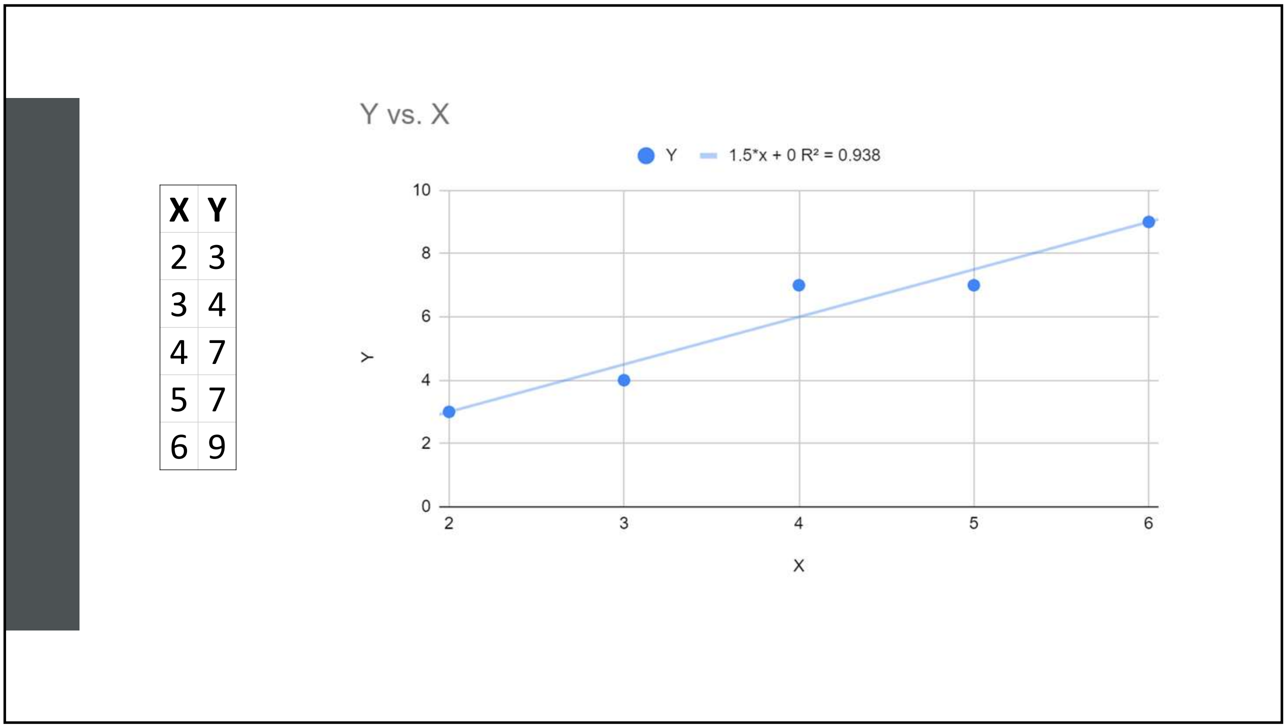
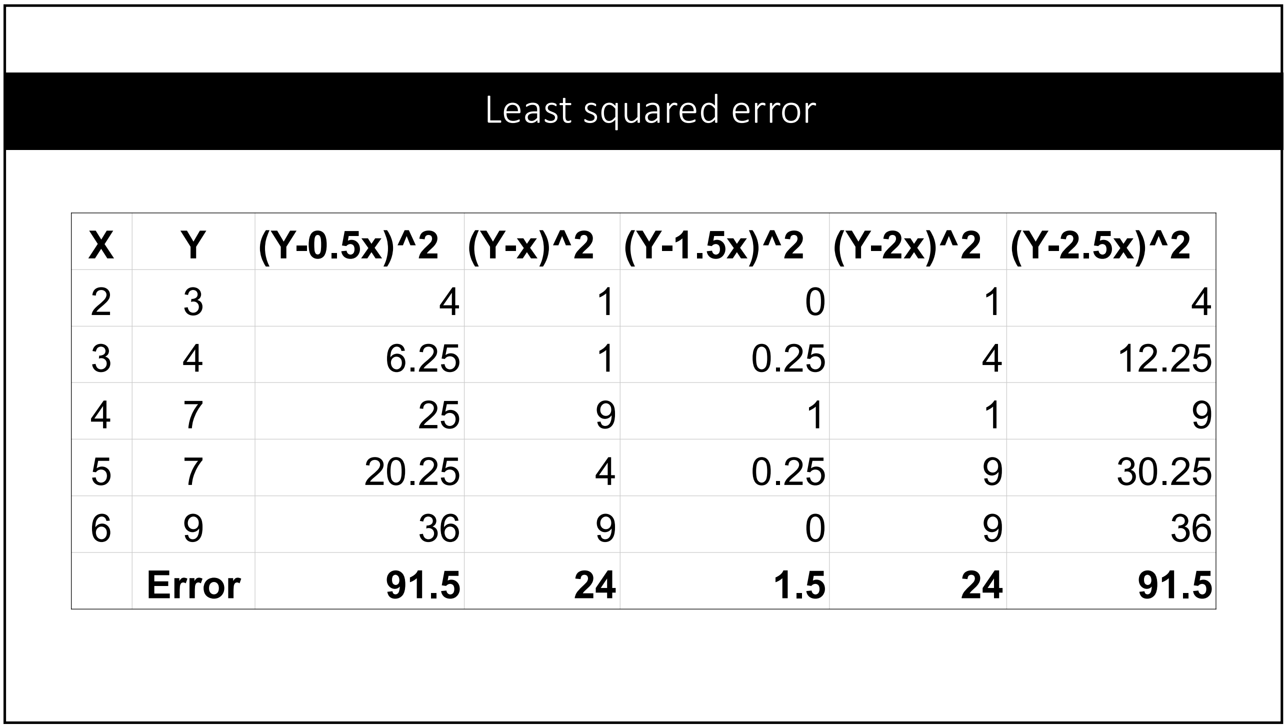
Supervised Learning
Regression
Linear Regression

1



X	Y
2	3
3	4
4	7
5	7
6	9

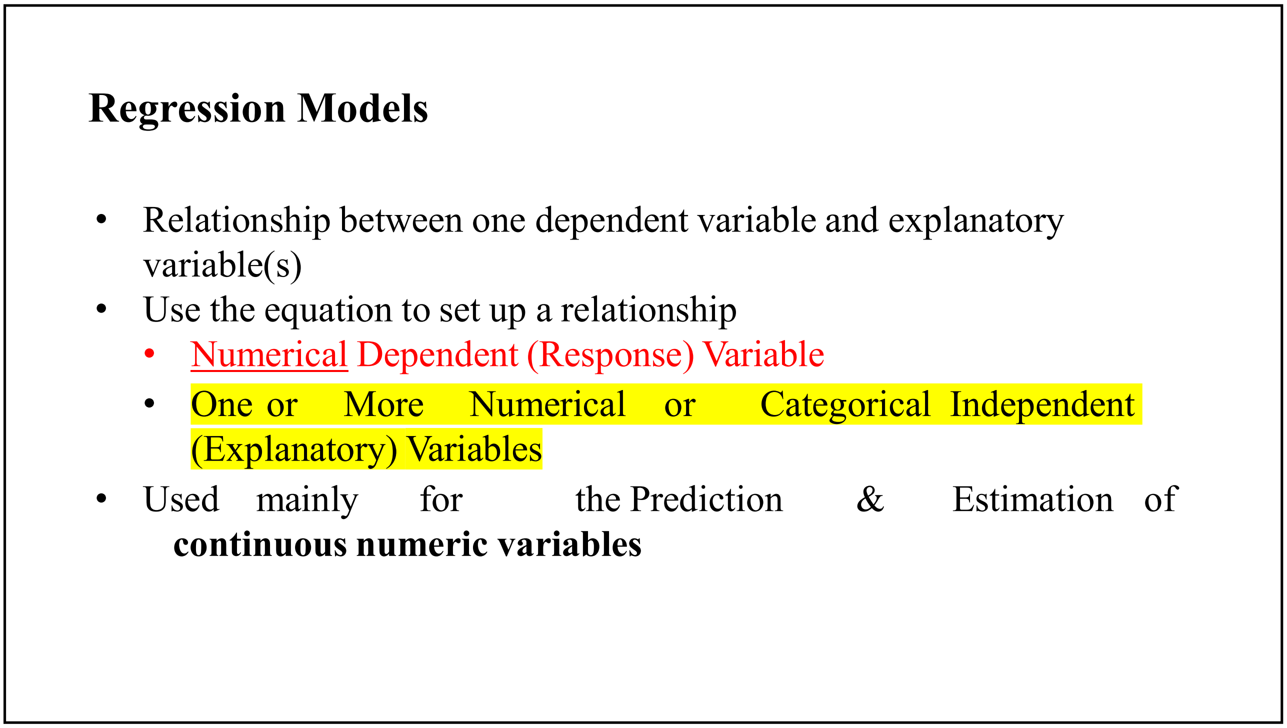
2



Least squared error

X	Y	(Y-0.5x)^2	(Y-x)^2	(Y-1.5x)^2	(Y-2x)^2	(Y-2.5x)^2
2	3	4	1	0	1	4
3	4	6.25	1	0.25	4	12.25
4	7	25	9	1	1	9
5	7	20.25	4	0.25	9	30.25
6	9	36	9	0	9	36
	Error	**91.5**	**24**	**1.5**	**24**	**91.5**

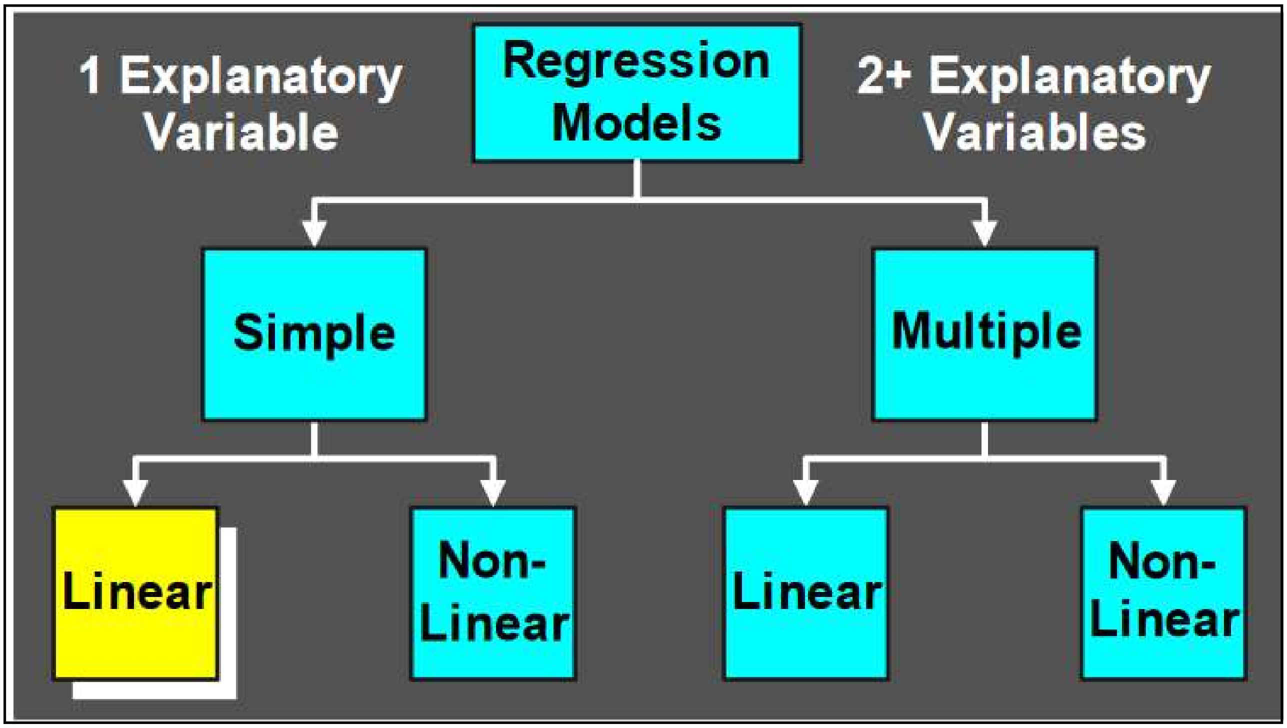
3



Regression Models

- Relationship between one dependent variable and explanatory variable(s)
- Use the equation to set up a relationship
  - Numerical Dependent (Response) Variable
  - One or More Numerical or Categorical Independent (Explanatory) Variables
- Used mainly for the Prediction & Estimation of **continuous numeric variables**

4



1 Explanatory Variable

Regression Models

2+ Explanatory Variables

Simple

Multiple

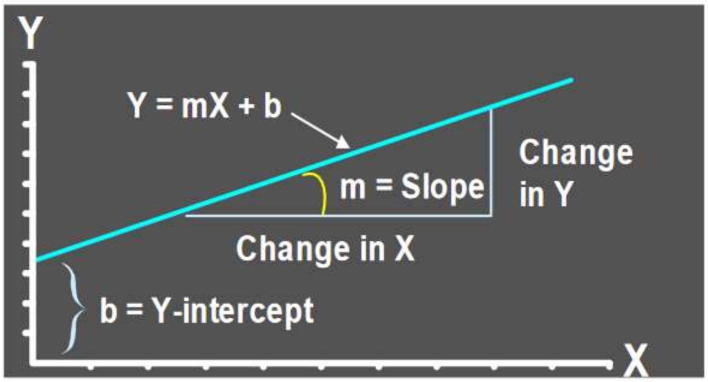
Linear

Non-Linear

Linear

Non-Linear

Linear Equations: A linear relationship between the predictors/independent variable (X) and the response/dependent variable (Y) is assumed.



Linear Equation : $\mathbf{y = \beta_0 + \beta_1 {*} x}$

Regression Modeling Steps

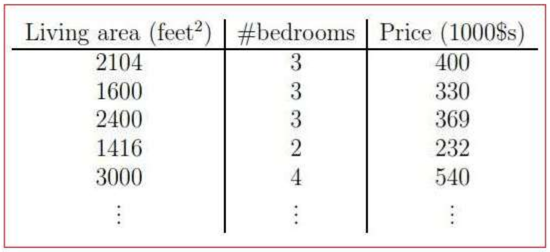
- 1.Hypothesize Deterministic Component
  - Estimate Unknown Parameters
- 2.Specify Probability Distribution of Random Error Term
  - Estimate Standard Deviation of Error
- 3.Evaluate the fitted Model
- 4.Use Model for Prediction & Estimation

7

Multiple Linear Regression

More than one predictor…

$$y = \beta_0 + \beta_1 * x_1 + \beta_2 * x_2$$

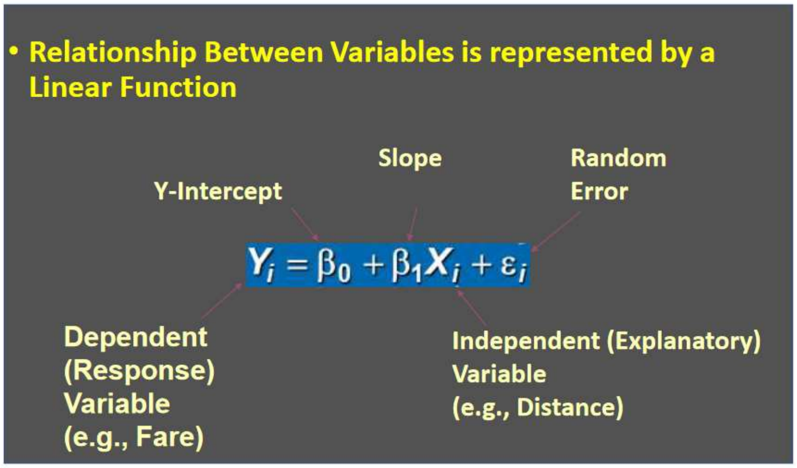


Living area (feet^2)	#bedrooms	Price (1000$s)
2104	3	400
1600	3	330
2400	3	369
1416	2	232
3000	4	540
⋮	⋮	⋮

Each regression coefficient is the amount of change in the outcome variable that would be expected per one-unit change of the predictor, if all other variables in the model were held constant.

8

Linear Regression Model

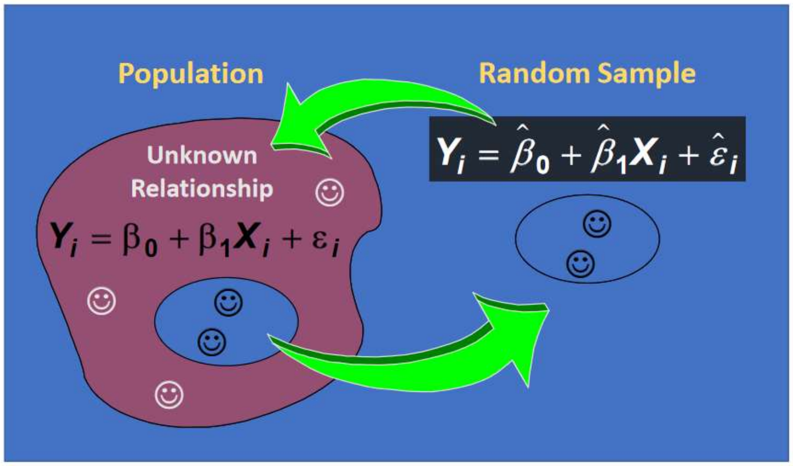


- Relationship Between Variables is represented by a Linear Function

$$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$$

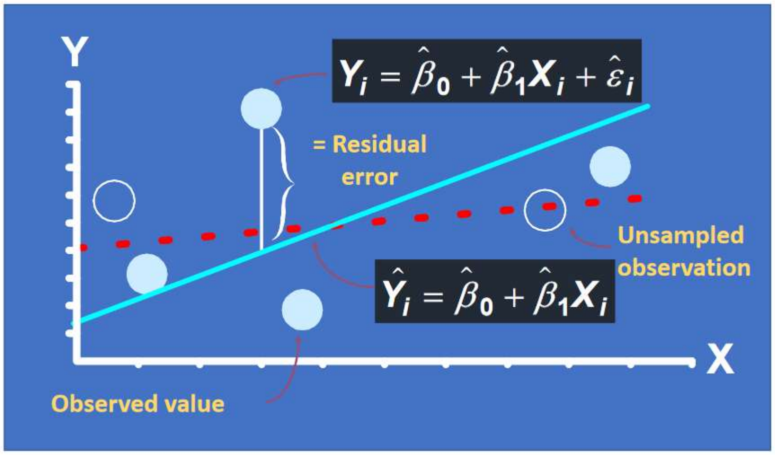
Y-Intercept
Slope
Random Error
Dependent (Response) Variable (e.g., Fare)
Independent (Explanatory) Variable (e.g., Distance)

Population & Sample Regression Models



Population
Random Sample
Unknown Relationship
$Y_i = \beta_0 + \beta_1 X_i + \varepsilon_i$
$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + \hat{\varepsilon}_i$

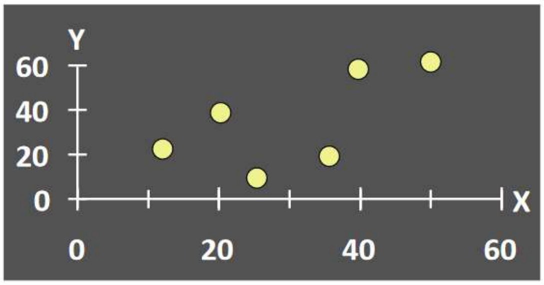
Linear Regression Model


Y
$Y_i = \hat{\beta}_0 + \hat{\beta}_1 X_i + \hat{\varepsilon}_i$
= Residual error
Unsampled observation
$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$
X
Observed value

11

Scatter Plot

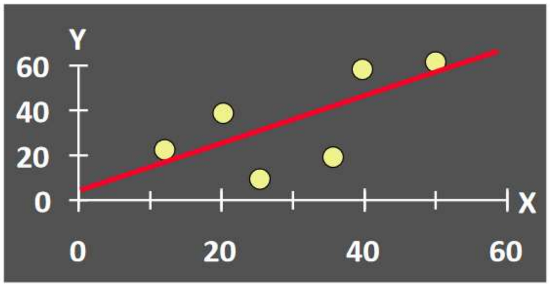
- 1. Plot of All (X_i, Y_i) Pairs
- 2. Suggests How Well Model Will Fit



12

Estimate Parameters

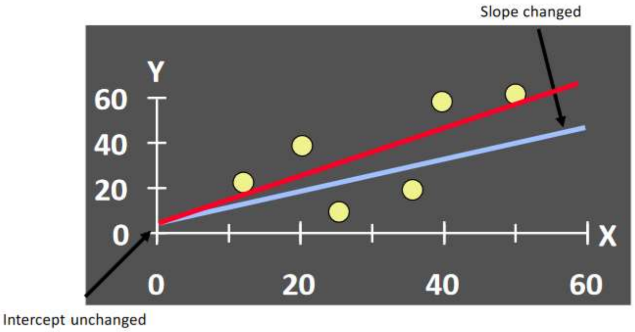
**How would you draw a line through the points?
How do you determine which line 'fits best'?**



13

Estimate Parameters

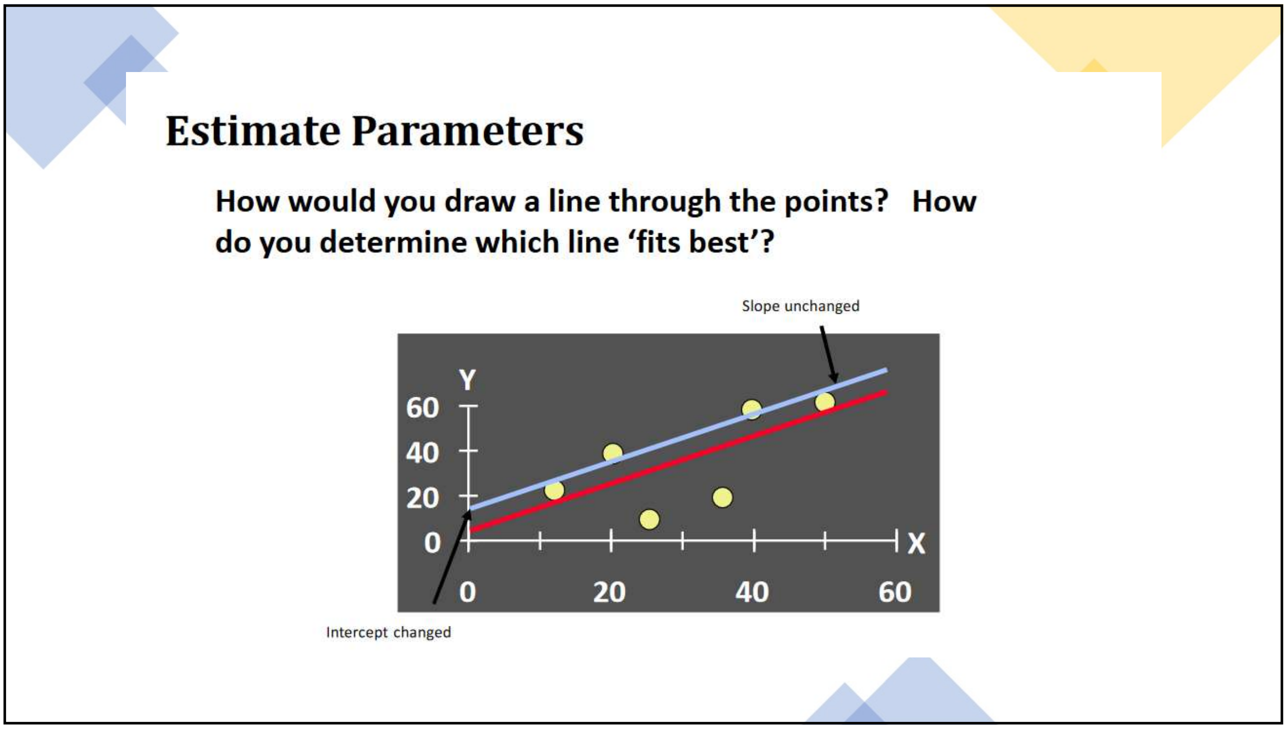
How would you draw a line through the points? How do you determine which line 'fits best'?



Slope changed

Intercept unchanged

14



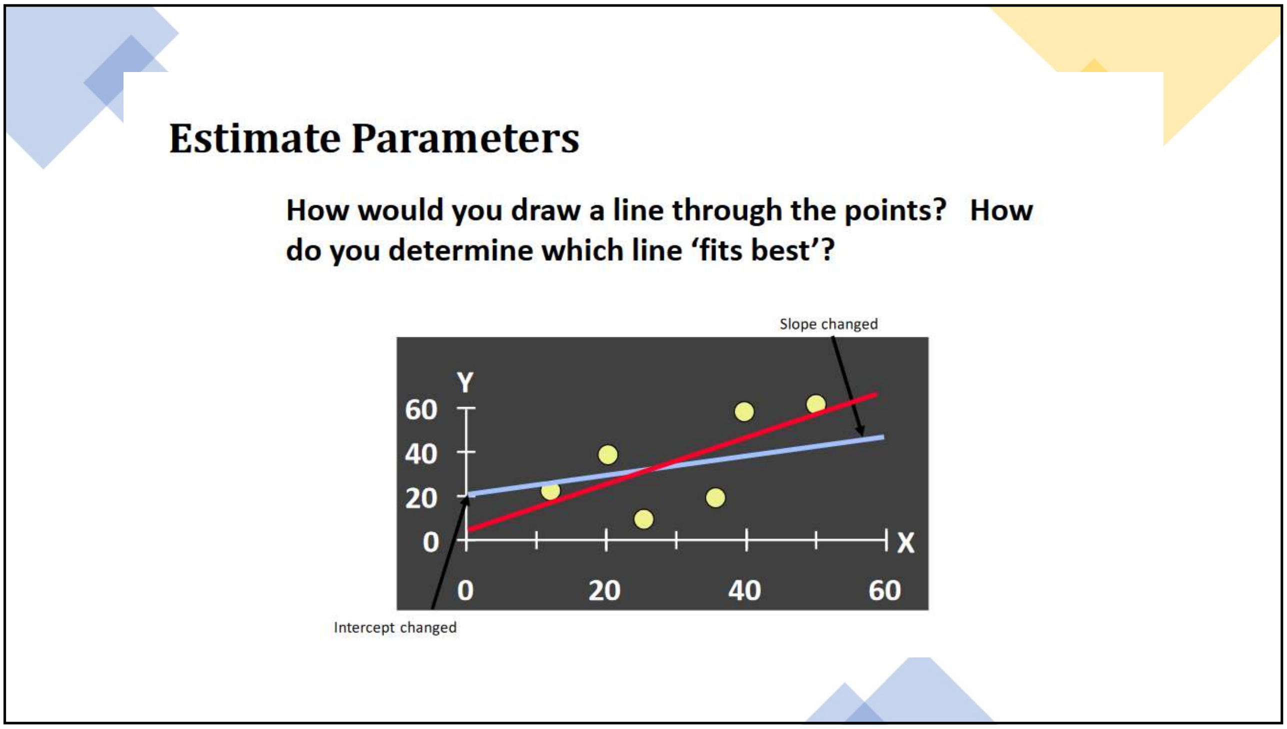
Estimate Parameters

How would you draw a line through the points? How do you determine which line 'fits best'?

Slope unchanged

Intercept changed

15



Estimate Parameters

How would you draw a line through the points? How do you determine which line 'fits best'?

Slope changed

Intercept changed

16

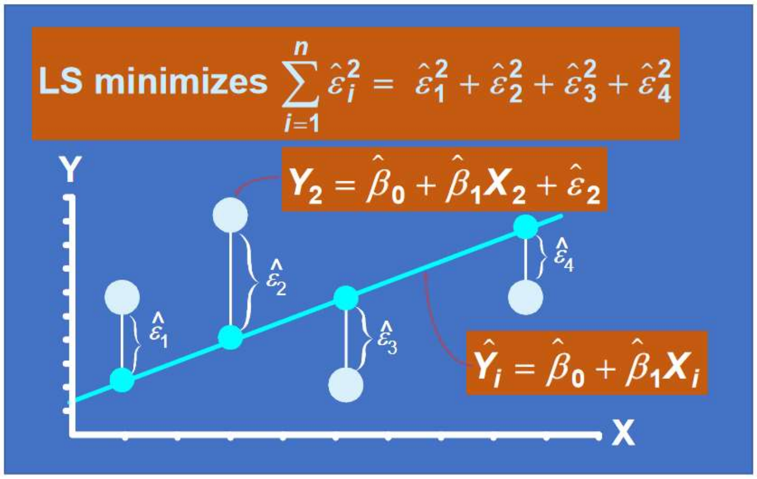
Least Squares

- 1. ‘Best Fit’ Means difference between actual ***y*** values & predicted $\hat{y}$ values are a minimum.
- *But* Positive Differences Off-Set Negative. So square errors!
- 2. LS Minimizes the Sum of the Squared Differences (errors) (SSE)

$$\sum_{i=1}^{n}\left(Y_i - \hat{Y}_i\right)^2 = \sum_{i=1}^{n}\hat{\varepsilon}_i^2$$

17

Least Squares Graphically



LS minimizes $\sum_{i=1}^{n}\hat{\varepsilon}_i^2 = \hat{\varepsilon}_1^2 + \hat{\varepsilon}_2^2 + \hat{\varepsilon}_3^2 + \hat{\varepsilon}_4^2$

$Y_2 = \hat{\beta}_0 + \hat{\beta}_1 X_2 + \hat{\varepsilon}_2$

$\hat{Y}_i = \hat{\beta}_0 + \hat{\beta}_1 X_i$

18

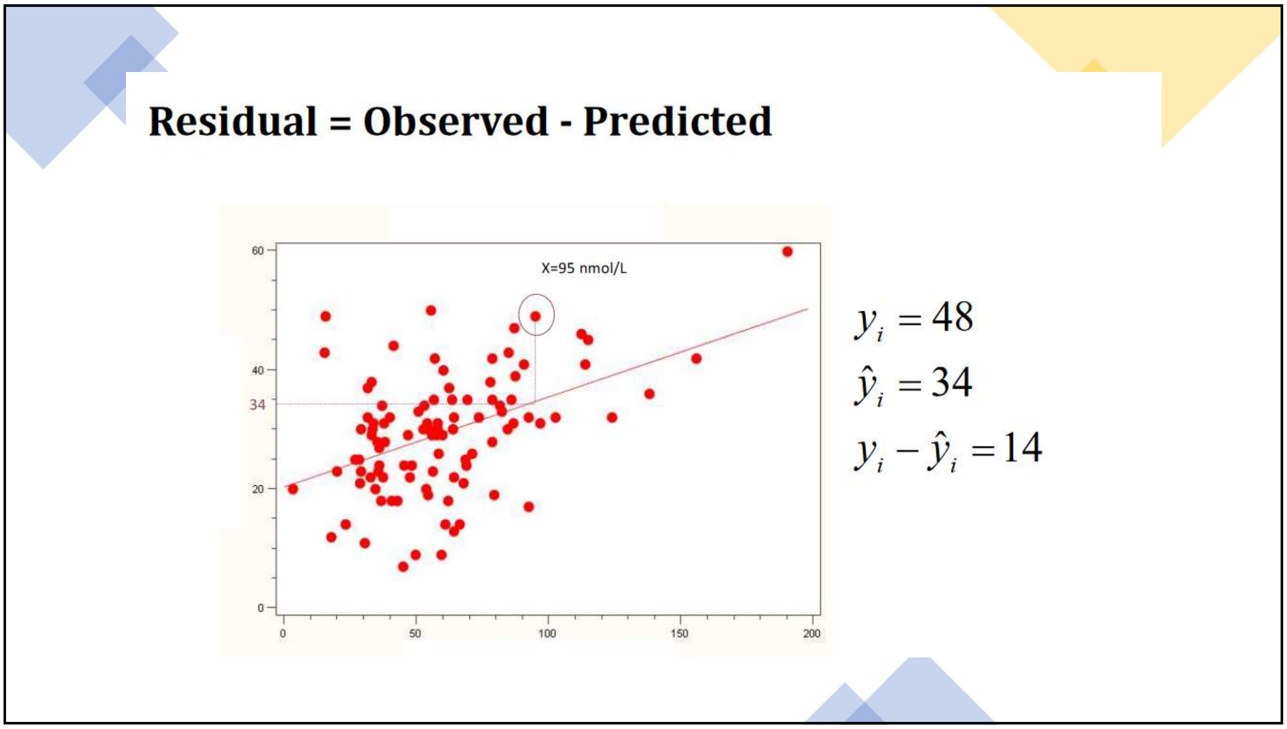
Assumptions

- Linear regression assumes that…
- **Linearity**: The relationship between X and Y must be linear. *(Check this assumption by examining a scatterplot of x and y.)*
- **Independence of errors**: There is no relationship between the residuals and the Y variable; in other words, Y is independent of errors. *(Check this assumption by examining a scatterplot of "residuals versus fits"; the correlation should be approximately 0. In other words, there should not look like there is a relationship.)*
- **Normality of errors:** The residuals must be approximately normally distributed. *(Check this assumption by examining a normal probability plot; the observations should be near the line. You can also examine a histogram of the residuals; it should be approximately normally distributed.)*
- **Equal Variances:** The variance of the residuals is the same for all values of X. *(Check this assumption by examining the scatterplot of "residuals versus fits"; the variance of the residuals should be the same across all values of the x-axis. If the plot shows a pattern (e.g., bowtie or megaphone shape), then variances are not consistent, and this assumption has not been met.)*

Residual Analysis: Check Assumptions

$$\boxed{e_i = Y_i - \hat{Y}_i}$$

- The residual for observation ***i***, $\mathbf{e_i}$, is the difference between its observed and predicted value.
- Check the assumptions of regression by examining the residuals
  - Examine for linearity assumption
  - Examine for constant variance for all levels of X (homoscedasticity)
  - Evaluate independence assumption
- Graphical Analysis of Residuals
  - Plot residuals vs. X



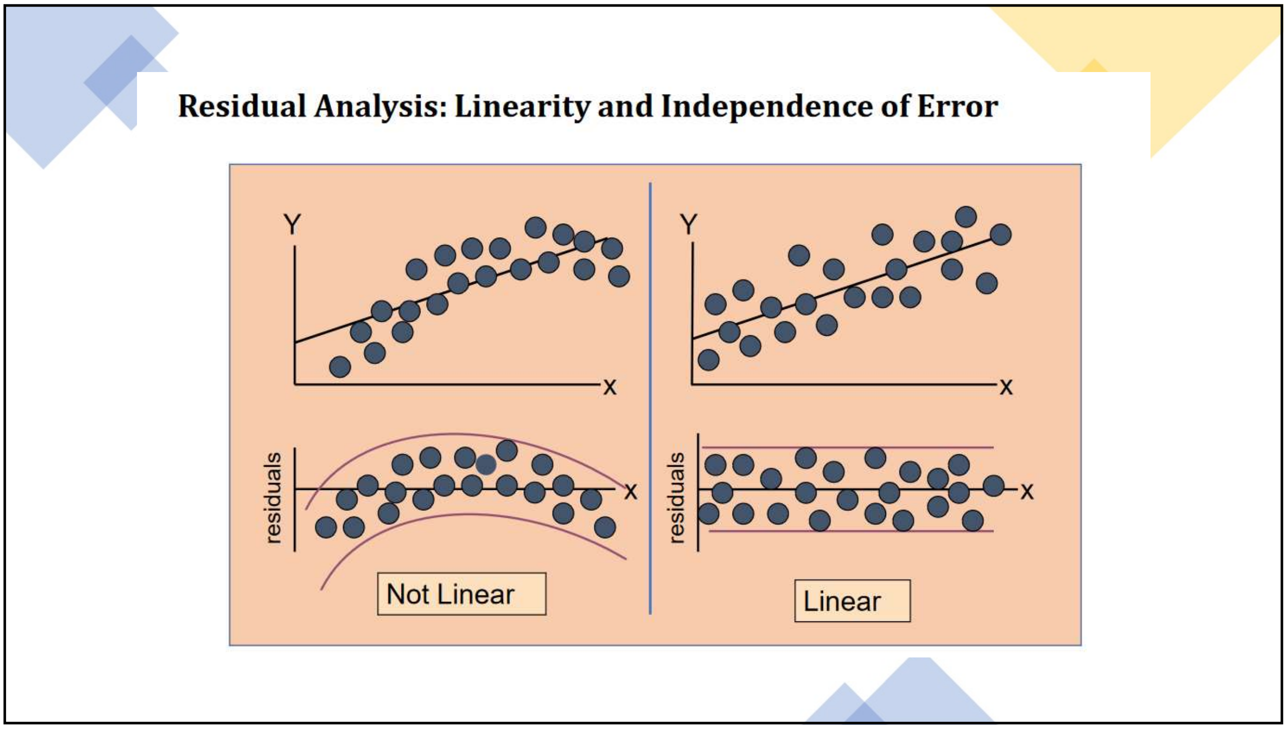
Residual = Observed - Predicted

$$y_i = 48$$

$$\hat{y}_i = 34$$

$$y_i - \hat{y}_i = 14$$

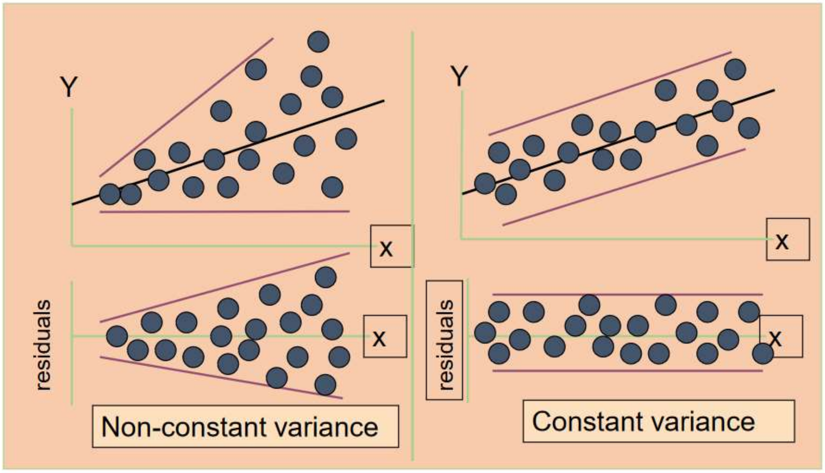
21



Residual Analysis: Linearity and Independence of Error

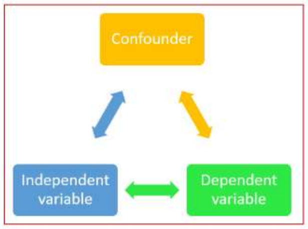
22

Residual Analysis for Homoscedasticity (Equal variance)



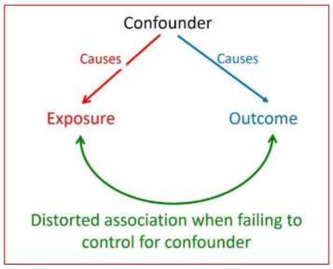
23

Confounding Variable


Confounder

Independent variable

Dependent variable


Confounder

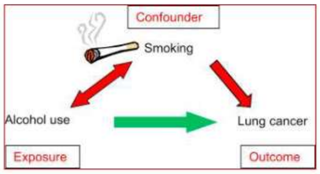
Causes

Causes

Exposure

Outcome

Distorted association when failing to control for confounder


Confounder

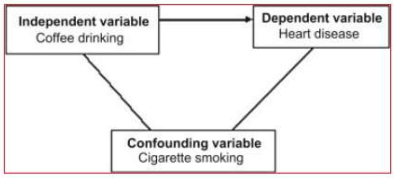
Smoking

Alcohol use

Lung cancer

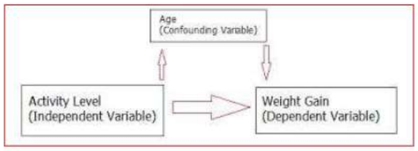
Exposure

Outcome


Independent variable
Coffee drinking

Dependent variable
Heart disease

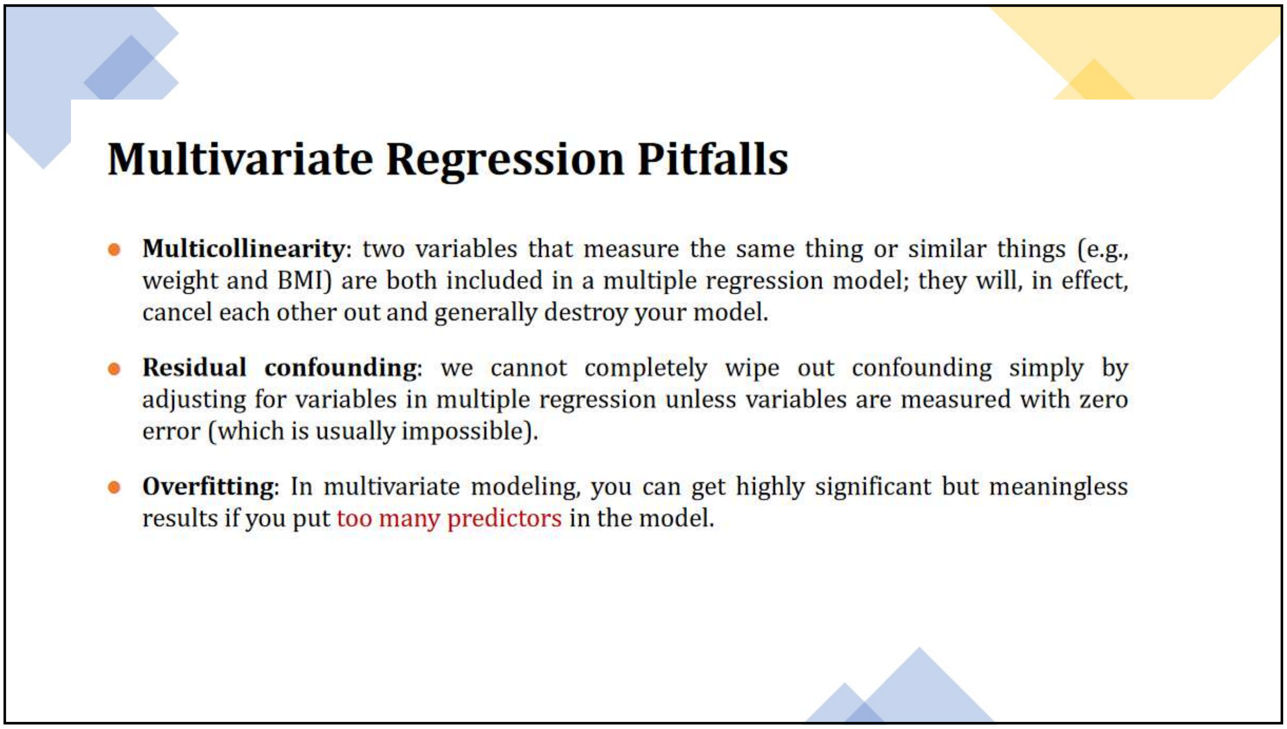
Confounding variable
Cigarette smoking


Age
(Confounding Variable)

Activity Level
(Independent Variable)

Weight Gain
(Dependent Variable)

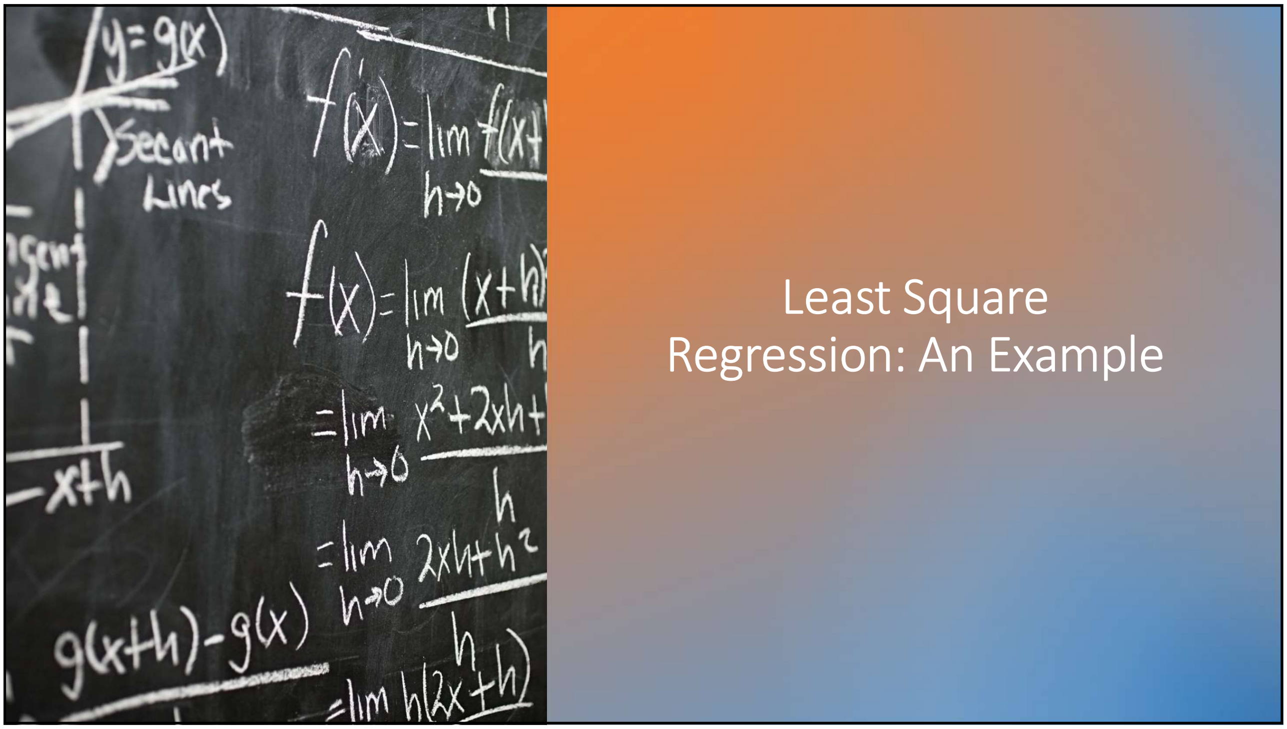
24



Multivariate Regression Pitfalls

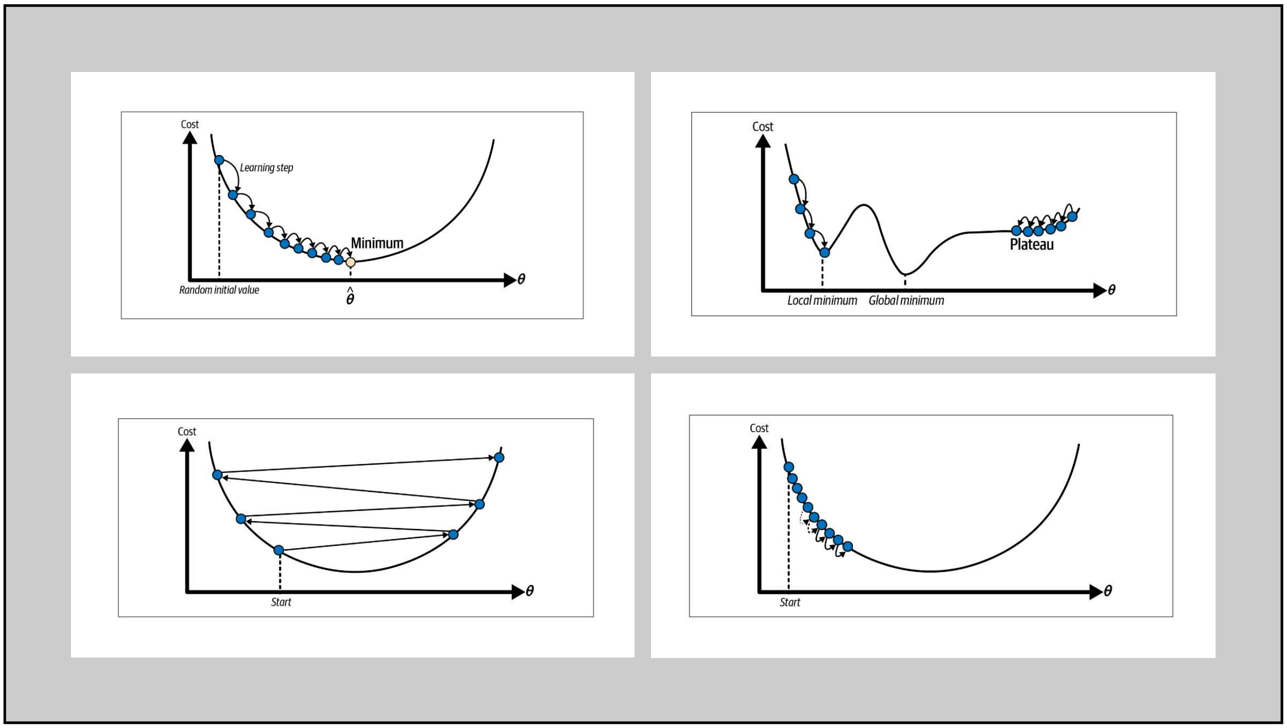
- **Multicollinearity**: two variables that measure the same thing or similar things (e.g., weight and BMI) are both included in a multiple regression model; they will, in effect, cancel each other out and generally destroy your model.
- **Residual confounding**: we cannot completely wipe out confounding simply by adjusting for variables in multiple regression unless variables are measured with zero error (which is usually impossible).
- **Overfitting**: In multivariate modeling, you can get highly significant but meaningless results if you put too many predictors in the model.

25

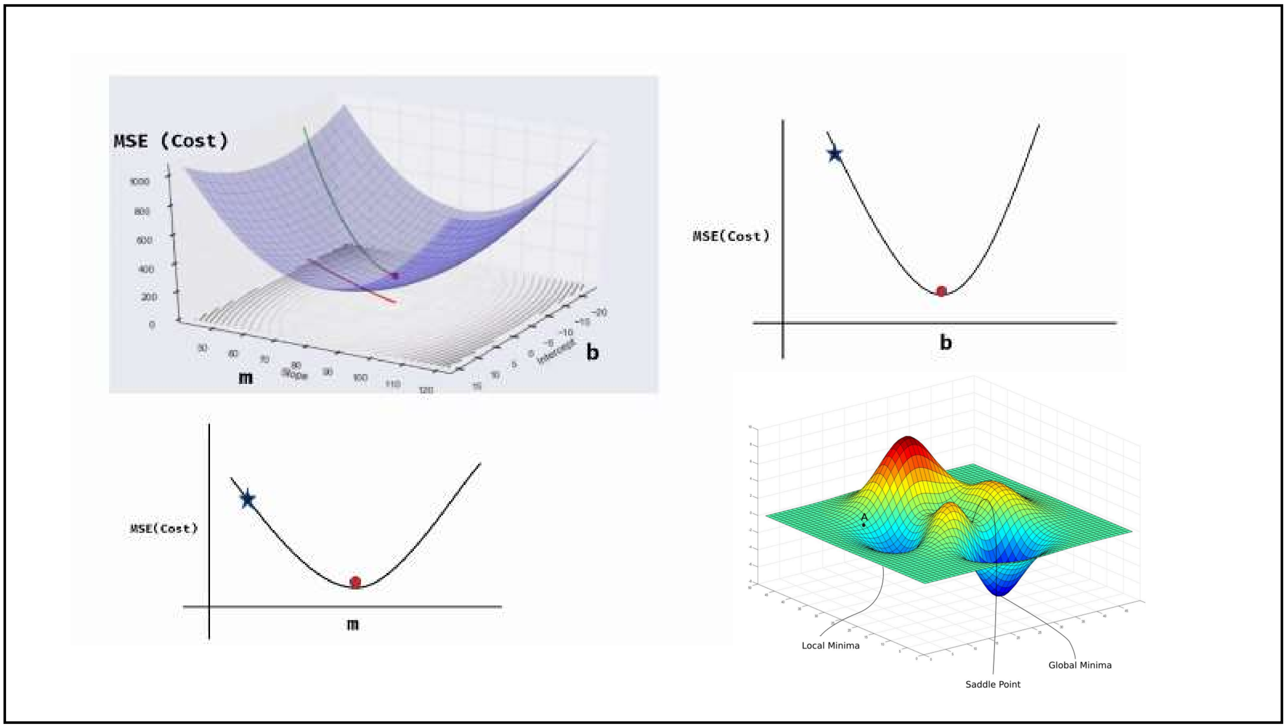


Least Square Regression: An Example

26


Cost
Learning step
Minimum
Random initial value
$\hat{\theta}$
θ
Cost
Plateau
Local minimum
Global minimum
θ
Cost
Start
θ
Cost
Start
θ

27


MSE (Cost)
1000
800
600
400
200
0
30
60
70
80
90
100
110
120
Slope
m
Intercept
-20
-10
-5
0
5
10
15
b
MSE(Cost)
b
MSE(Cost)
m
A
Local Minima
Saddle Point
Global Minima

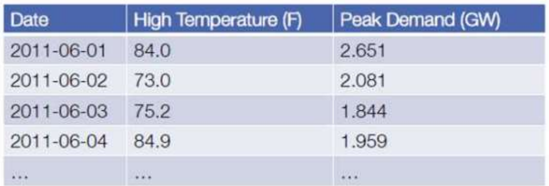
28

Predicting Electricity Use

What will peak power consumption be in Pittsburgh tomorrow?

Difficult to build an “a priori” model

But, relatively easy to record past days of consumption, plus additional features that affect consumption (i.e., weather)

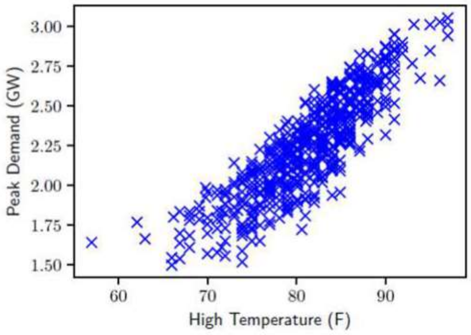


Date	High Temperature (F)	Peak Demand (GW)
2011-06-01	84.0	2.651
2011-06-02	73.0	2.081
2011-06-03	75.2	1.844
2011-06-04	84.9	1.959
…	…	…

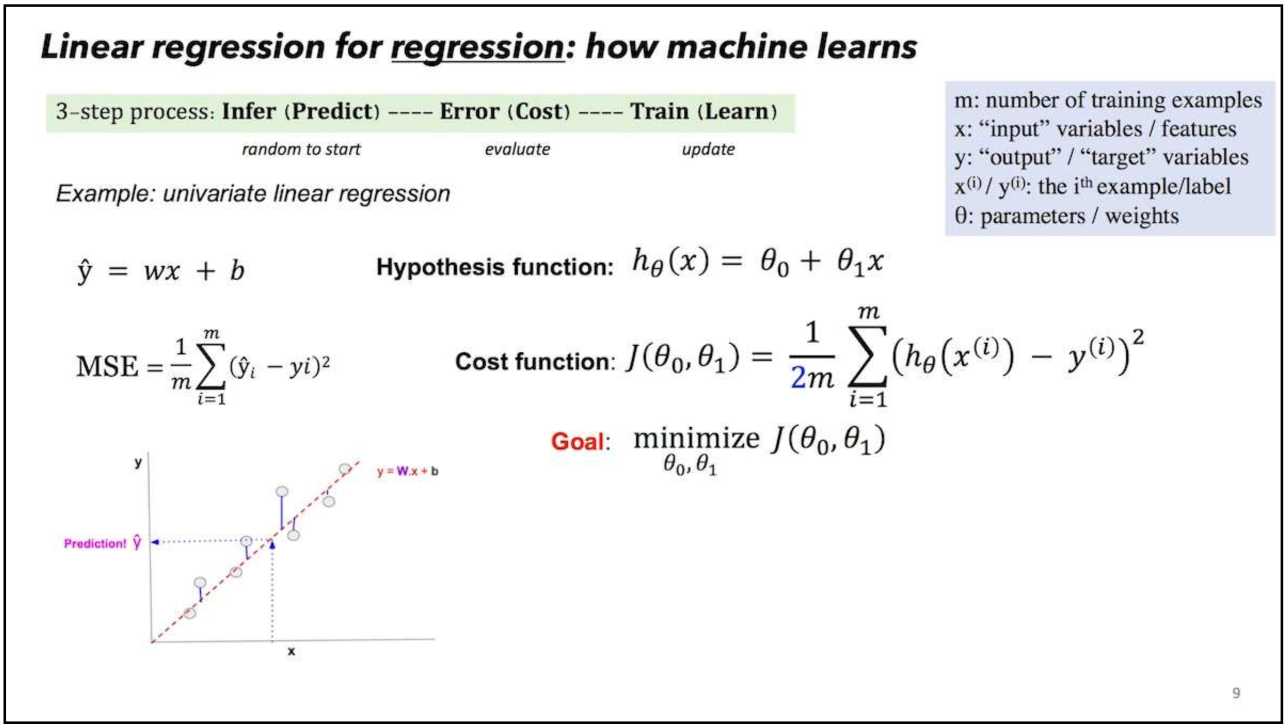
29

Plot Consumption vs. Temperature

Plot of high temperature vs. peak demand for summer months (June – August) for past six years



30



Linear regression for <u>regression</u>: how machine learns

3-step process: **Infer (Predict)** ---- **Error (Cost)** ---- **Train (Learn)**

random to start *evaluate* *update*

m: number of training examples
x: "input" variables / features
y: "output" / "target" variables
$x^{(i)}$ / $y^{(i)}$: the i^{th} example/label
θ: parameters / weights

Example: univariate linear regression

$$\hat{y} = wx + b$$

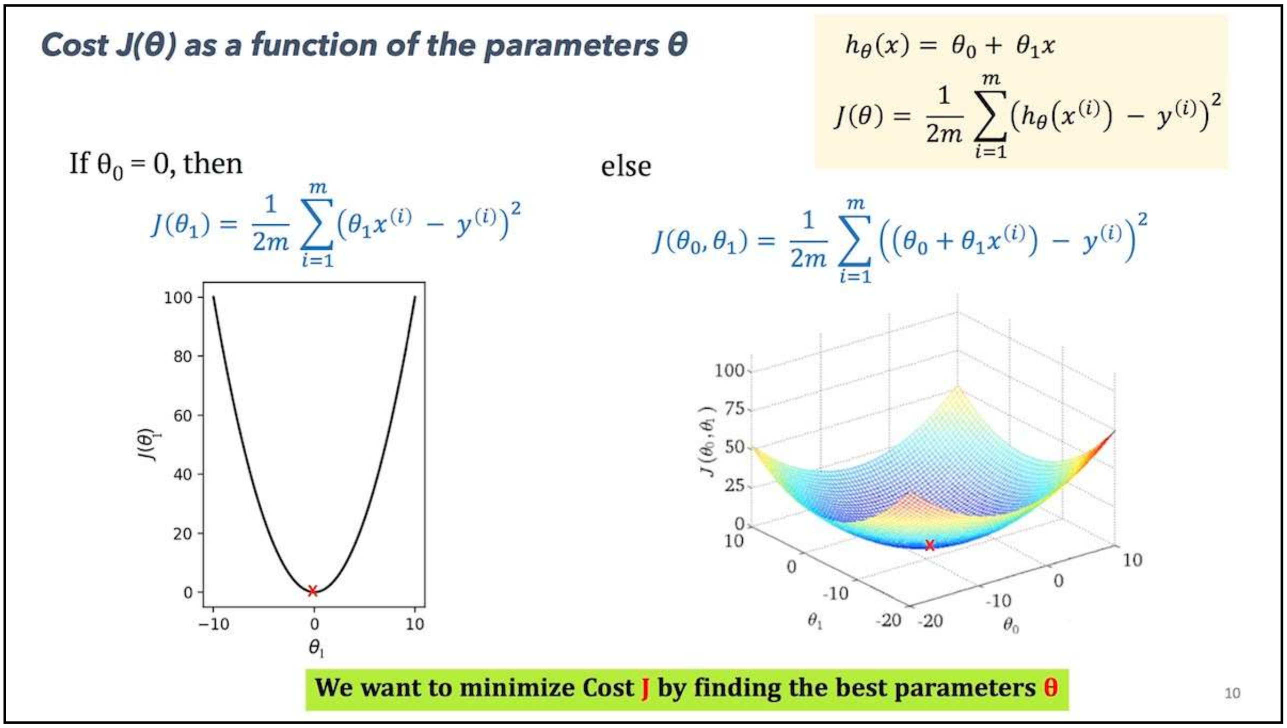
Hypothesis function: $h_\theta(x) = \theta_0 + \theta_1 x$

$$\text{MSE} = \frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - yi)^2$$

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

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

9



Cost J(θ) as a function of the parameters θ

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

$J(\theta) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_\theta\left(x^{(i)}\right) - y^{(i)}\right)^2$

If $\theta_0 = 0$, then

$$J(\theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(\theta_1 x^{(i)} - y^{(i)}\right)^2$$

else

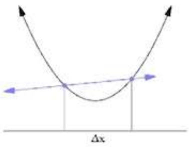
$$J(\theta_0, \theta_1) = \frac{1}{2m}\sum_{i=1}^{m}\left(\left(\theta_0 + \theta_1 x^{(i)}\right) - y^{(i)}\right)^2$$

We want to minimize Cost J by finding the best parameters θ

10

Using gradient descent algorithm to update θ

$$\theta := \theta - \alpha \frac{d}{d\theta} J(\theta)$$



θ: parameters

α: learning rate

$\frac{d}{d\theta} J(\theta)$: derivative of the cost function $J(\theta)$

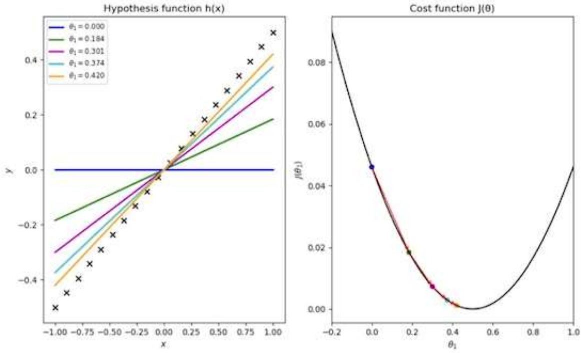
(direction to update)

```
for j = 1 to n
        $\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \theta_1, \ldots \theta_n)$
    simultaneously update $\theta_j$
        repeat until convergence
```

If $\theta_0 = 0$

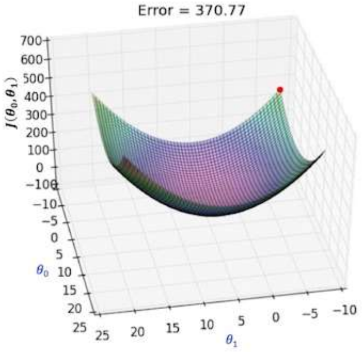
$$h_\theta(x) = \theta_1 x \qquad J(\theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\theta_1 x^{(i)} - y^{(i)}\right)^2$$

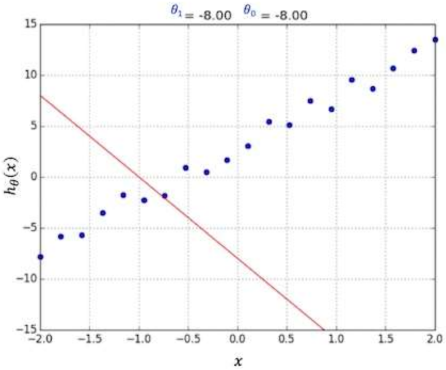
$$\theta_1 := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_1)$$



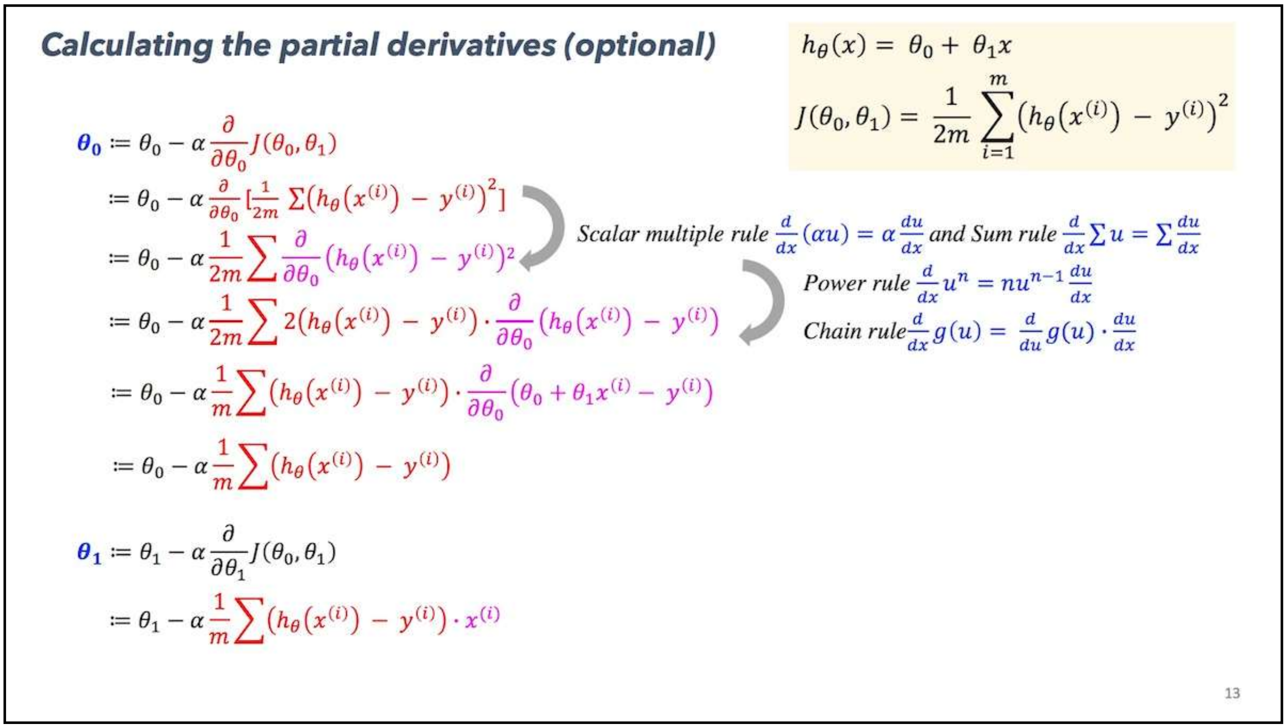
If $\theta_0 \neq 0$, $J(\theta_0, \theta_1) = \frac{1}{2m} \sum_{i=1}^{m} \left(\left(\theta_0 + \theta_1 x^{(i)}\right) - y^{(i)}\right)^2$

$$\boldsymbol{\theta_0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) \qquad \boldsymbol{\theta_1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)$$





Hacknoon.com



Calculating the partial derivatives (optional)

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

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

$$\boldsymbol{\theta_0} \coloneqq \theta_0 - \alpha\frac{\partial}{\partial\theta_0}J(\theta_0, \theta_1)$$

$$\coloneqq \theta_0 - \alpha\frac{\partial}{\partial\theta_0}\left[\frac{1}{2m}\sum\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2\right]$$

$$\coloneqq \theta_0 - \alpha\frac{1}{2m}\sum\frac{\partial}{\partial\theta_0}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$$

$$\coloneqq \theta_0 - \alpha\frac{1}{2m}\sum 2\left(h_\theta(x^{(i)}) - y^{(i)}\right)\cdot\frac{\partial}{\partial\theta_0}\left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

$$\coloneqq \theta_0 - \alpha\frac{1}{m}\sum\left(h_\theta(x^{(i)}) - y^{(i)}\right)\cdot\frac{\partial}{\partial\theta_0}\left(\theta_0 + \theta_1 x^{(i)} - y^{(i)}\right)$$

$$\coloneqq \theta_0 - \alpha\frac{1}{m}\sum\left(h_\theta(x^{(i)}) - y^{(i)}\right)$$

Scalar multiple rule $\frac{d}{dx}(\alpha u) = \alpha\frac{du}{dx}$ *and Sum rule* $\frac{d}{dx}\sum u = \sum\frac{du}{dx}$

Power rule $\frac{d}{dx}u^n = nu^{n-1}\frac{du}{dx}$

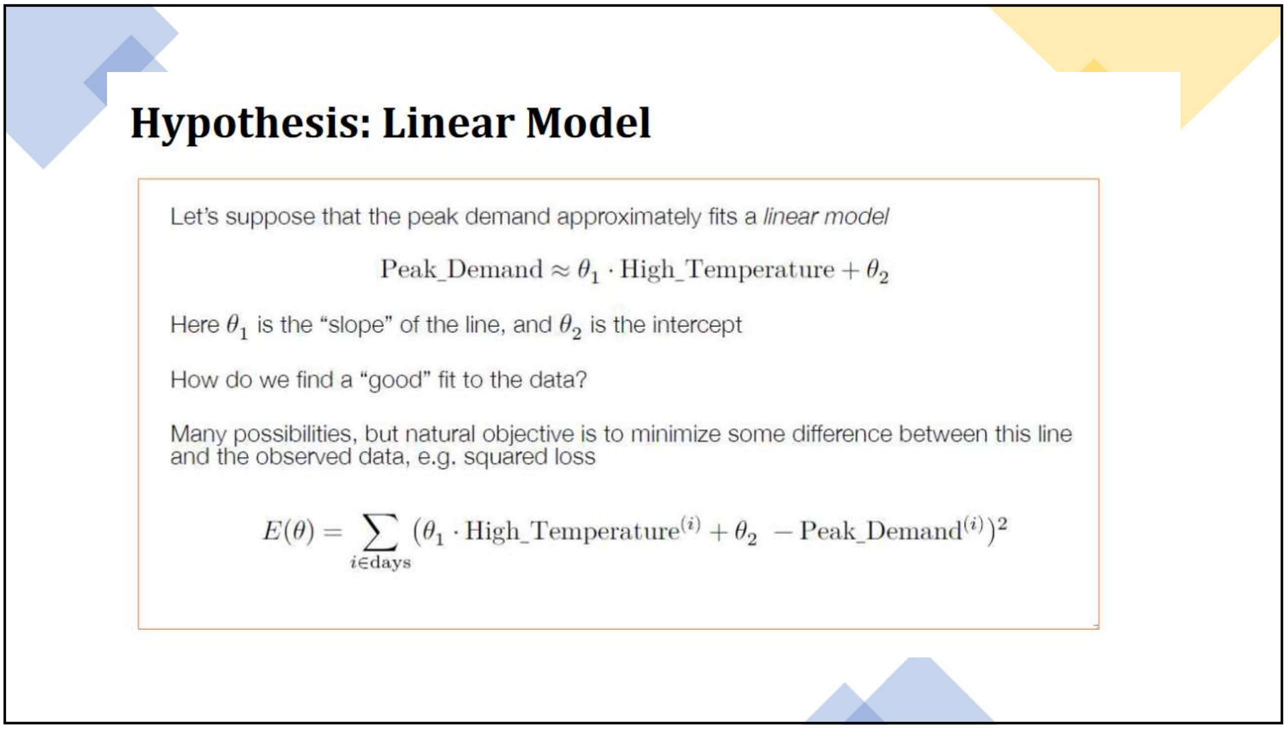
Chain rule $\frac{d}{dx}g(u) = \frac{d}{du}g(u)\cdot\frac{du}{dx}$

$$\boldsymbol{\theta_1} \coloneqq \theta_1 - \alpha\frac{\partial}{\partial\theta_1}J(\theta_0, \theta_1)$$

$$\coloneqq \theta_1 - \alpha\frac{1}{m}\sum\left(h_\theta(x^{(i)}) - y^{(i)}\right)\cdot x^{(i)}$$

13

35



Hypothesis: Linear Model

Let's suppose that the peak demand approximately fits a *linear model*

$$\text{Peak\_Demand} \approx \theta_1 \cdot \text{High\_Temperature} + \theta_2$$

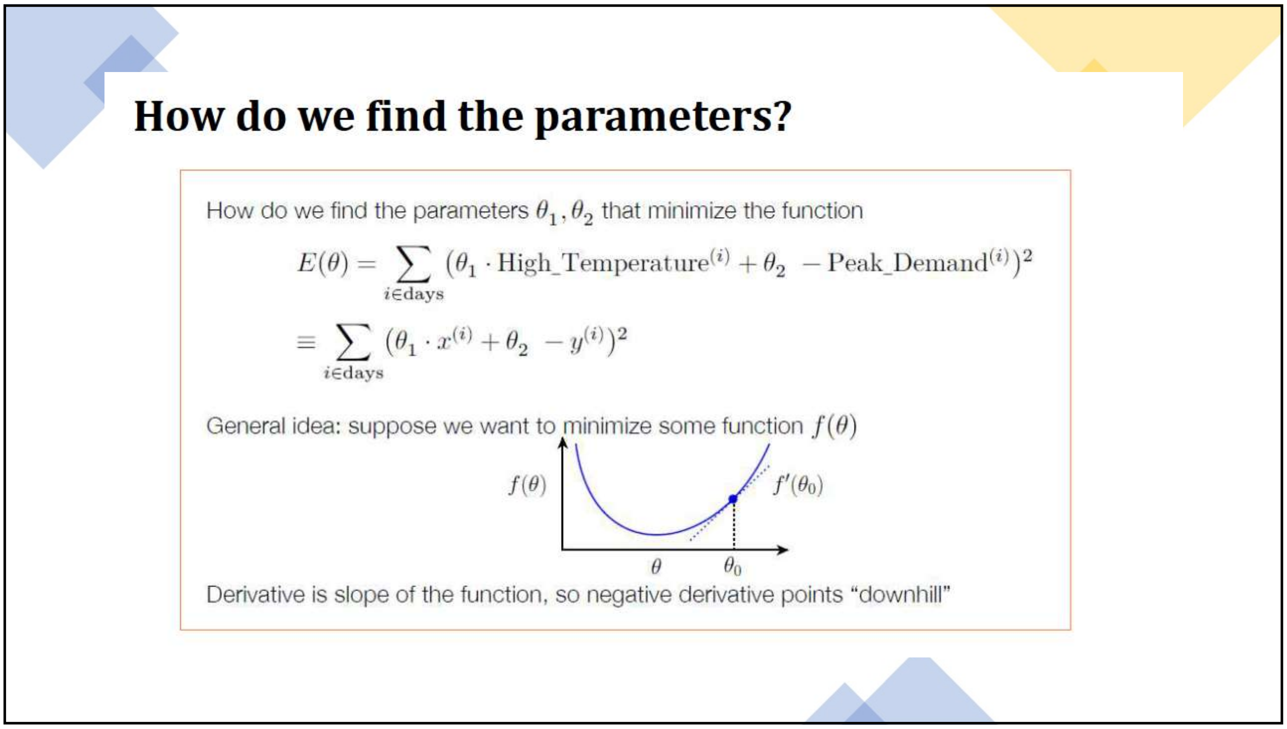
Here θ_1 is the "slope" of the line, and θ_2 is the intercept

How do we find a "good" fit to the data?

Many possibilities, but natural objective is to minimize some difference between this line and the observed data, e.g. squared loss

$$E(\theta) = \sum_{i \in \text{days}}(\theta_1 \cdot \text{High\_Temperature}^{(i)} + \theta_2 \; - \text{Peak\_Demand}^{(i)})^2$$

36



How do we find the parameters?

How do we find the parameters θ_1, θ_2 that minimize the function

$$E(\theta) = \sum_{i \in \text{days}} (\theta_1 \cdot \text{High\_Temperature}^{(i)} + \theta_2 \;\; - \text{Peak\_Demand}^{(i)})^2$$

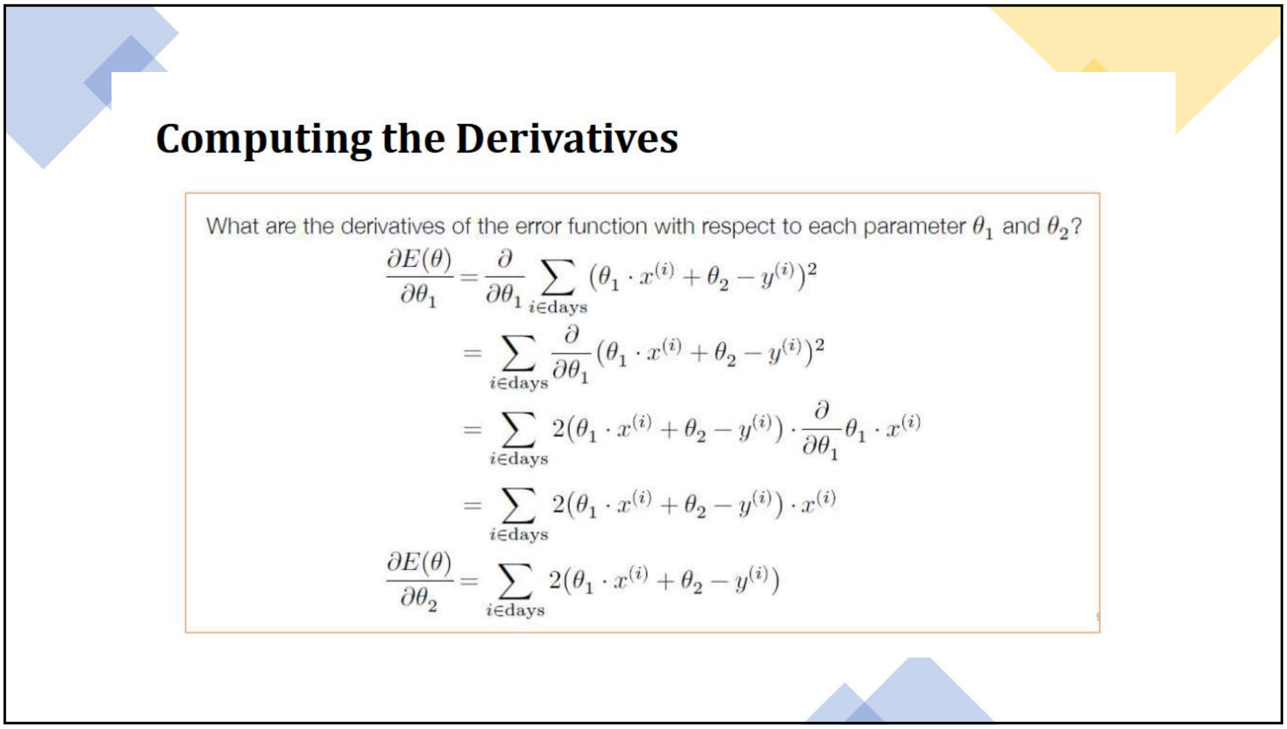
$$\equiv \sum_{i \in \text{days}} (\theta_1 \cdot x^{(i)} + \theta_2 \;\; - y^{(i)})^2$$

General idea: suppose we want to minimize some function $f(\theta)$

$f(\theta)$
$f'(\theta_0)$
θ
θ_0

Derivative is slope of the function, so negative derivative points “downhill”

37



Computing the Derivatives

What are the derivatives of the error function with respect to each parameter θ_1 and θ_2?

$$\frac{\partial E(\theta)}{\partial \theta_1} = \frac{\partial}{\partial \theta_1} \sum_{i \in \text{days}} (\theta_1 \cdot x^{(i)} + \theta_2 - y^{(i)})^2$$

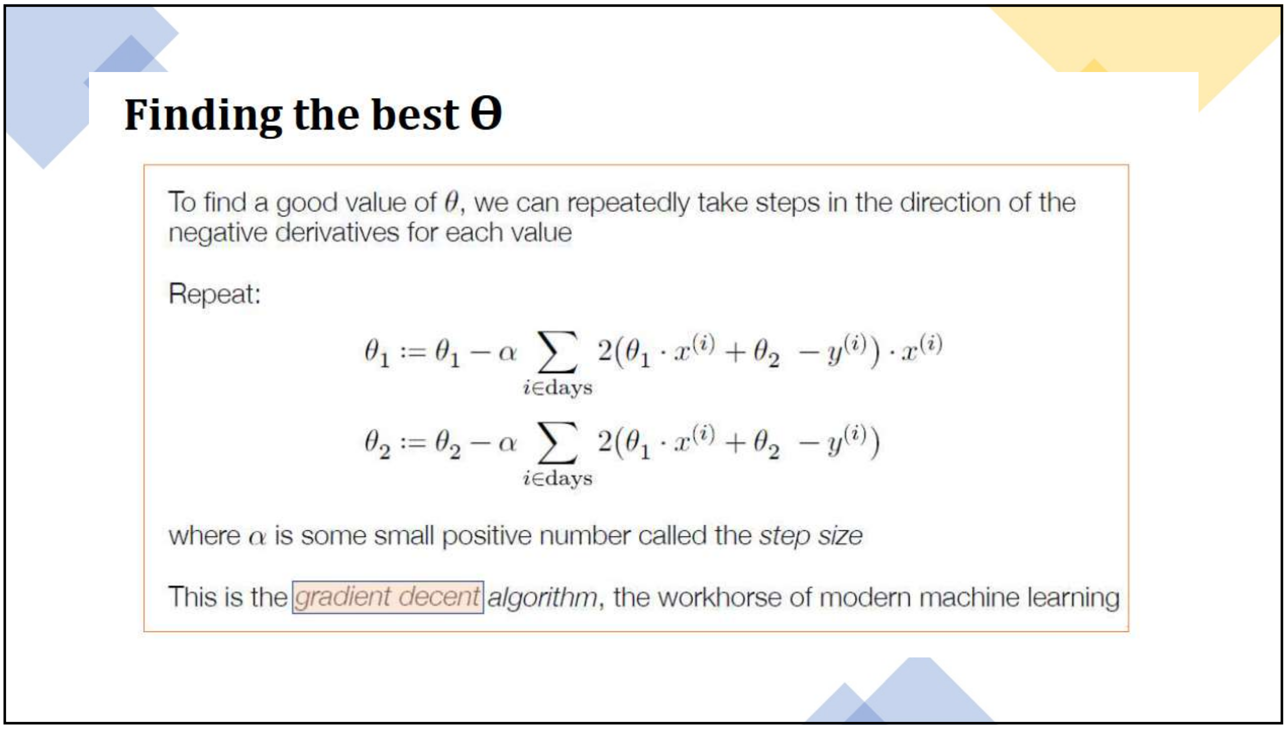
$$= \sum_{i \in \text{days}} \frac{\partial}{\partial \theta_1} (\theta_1 \cdot x^{(i)} + \theta_2 - y^{(i)})^2$$

$$= \sum_{i \in \text{days}} 2(\theta_1 \cdot x^{(i)} + \theta_2 - y^{(i)}) \cdot \frac{\partial}{\partial \theta_1} \theta_1 \cdot x^{(i)}$$

$$= \sum_{i \in \text{days}} 2(\theta_1 \cdot x^{(i)} + \theta_2 - y^{(i)}) \cdot x^{(i)}$$

$$\frac{\partial E(\theta)}{\partial \theta_2} = \sum_{i \in \text{days}} 2(\theta_1 \cdot x^{(i)} + \theta_2 - y^{(i)})$$

38



Finding the best Θ

To find a good value of θ, we can repeatedly take steps in the direction of the negative derivatives for each value

Repeat:

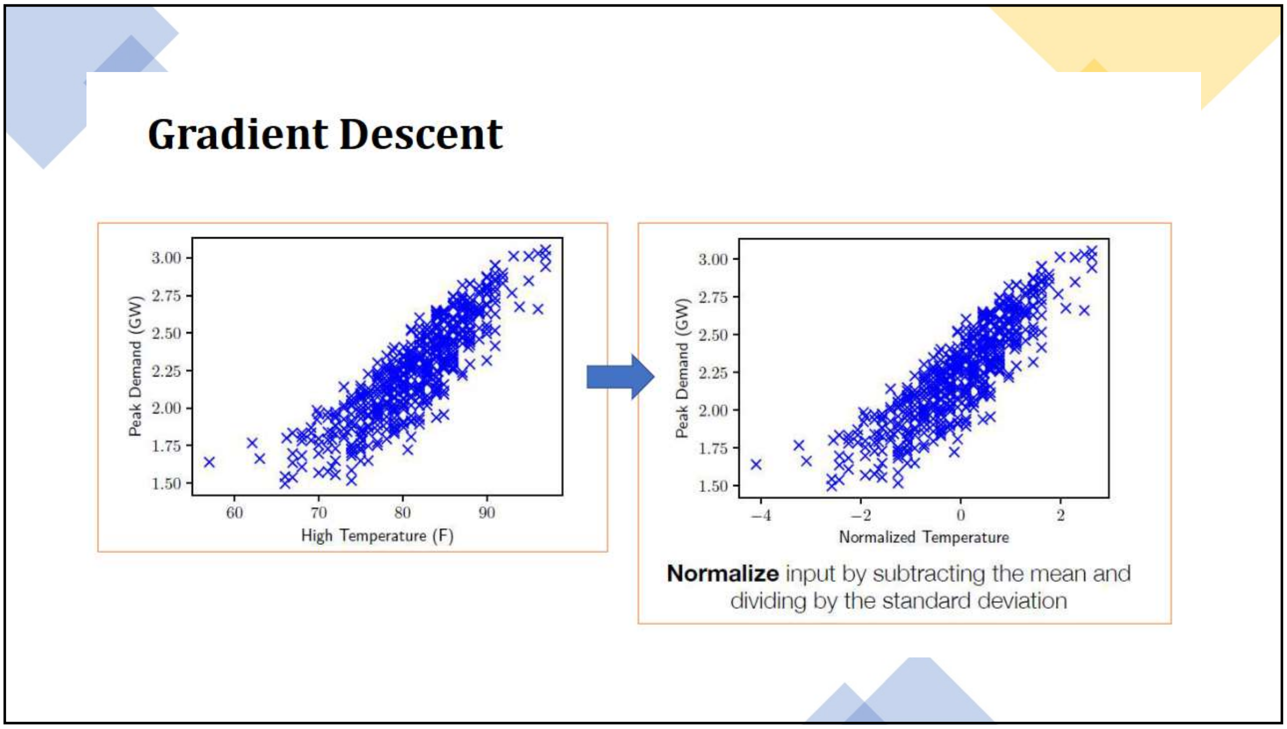
$$\theta_1 := \theta_1 - \alpha \sum_{i \in \text{days}} 2(\theta_1 \cdot x^{(i)} + \theta_2 \; - y^{(i)}) \cdot x^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \sum_{i \in \text{days}} 2(\theta_1 \cdot x^{(i)} + \theta_2 \; - y^{(i)})$$

where α is some small positive number called the *step size*

This is the *gradient decent algorithm*, the workhorse of modern machine learning

39

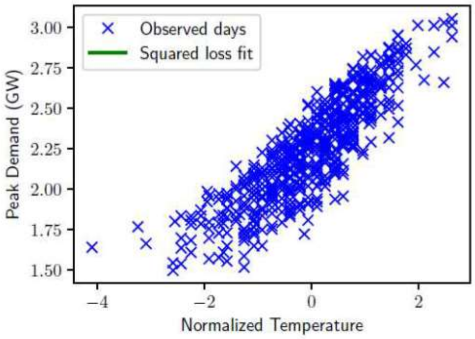


Gradient Descent

Normalize input by subtracting the mean and dividing by the standard deviation

40

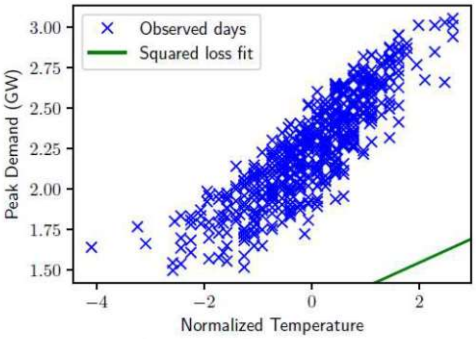
Gradient Descent - Iteration 1



$\theta = (0.00, 0.00)$
$E(\theta) = 1427.53$
$(\frac{\partial E(\theta)}{\partial \theta_1}, \frac{\partial E(\theta)}{\partial \theta_2}) = (-151.20, -1243.10)$

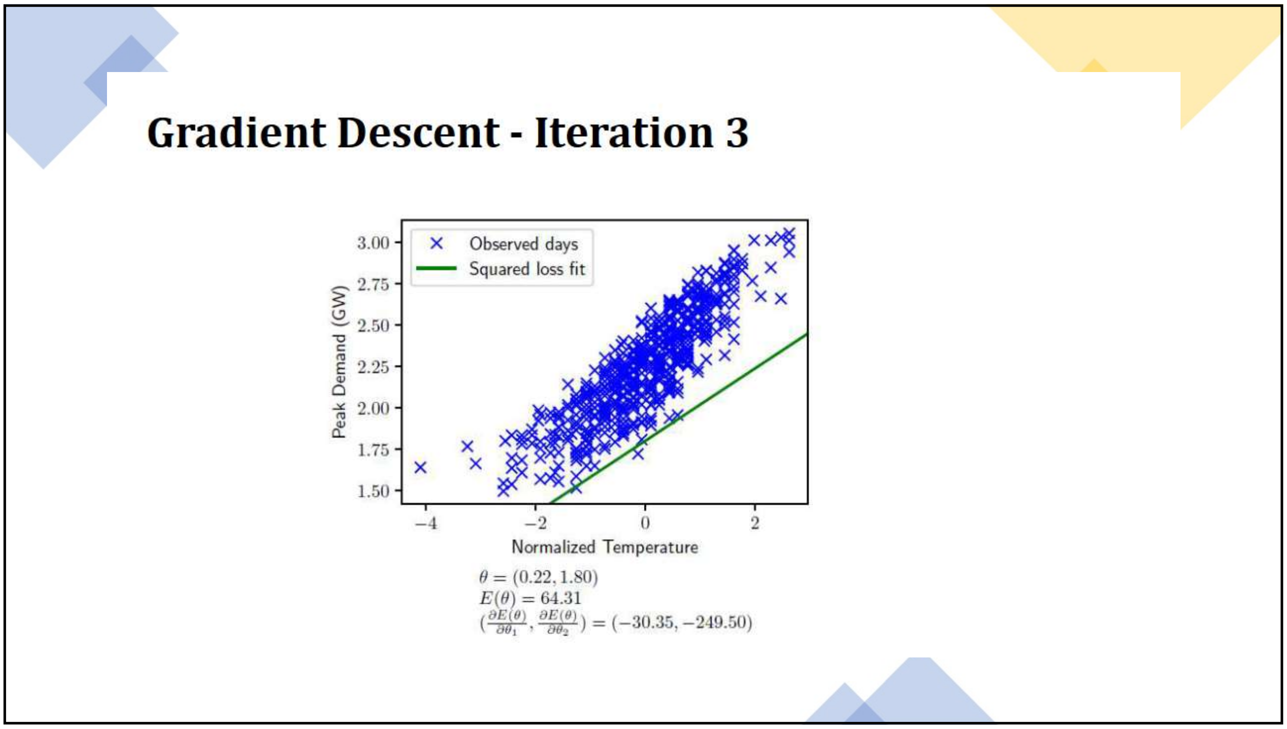
41

Gradient Descent - Iteration 2



$\theta = (0.15, 1.24)$
$E(\theta) = 292.18$
$(\frac{\partial E(\theta)}{\partial \theta_1}, \frac{\partial E(\theta)}{\partial \theta_2}) = (-67.74, -556.91)$

42



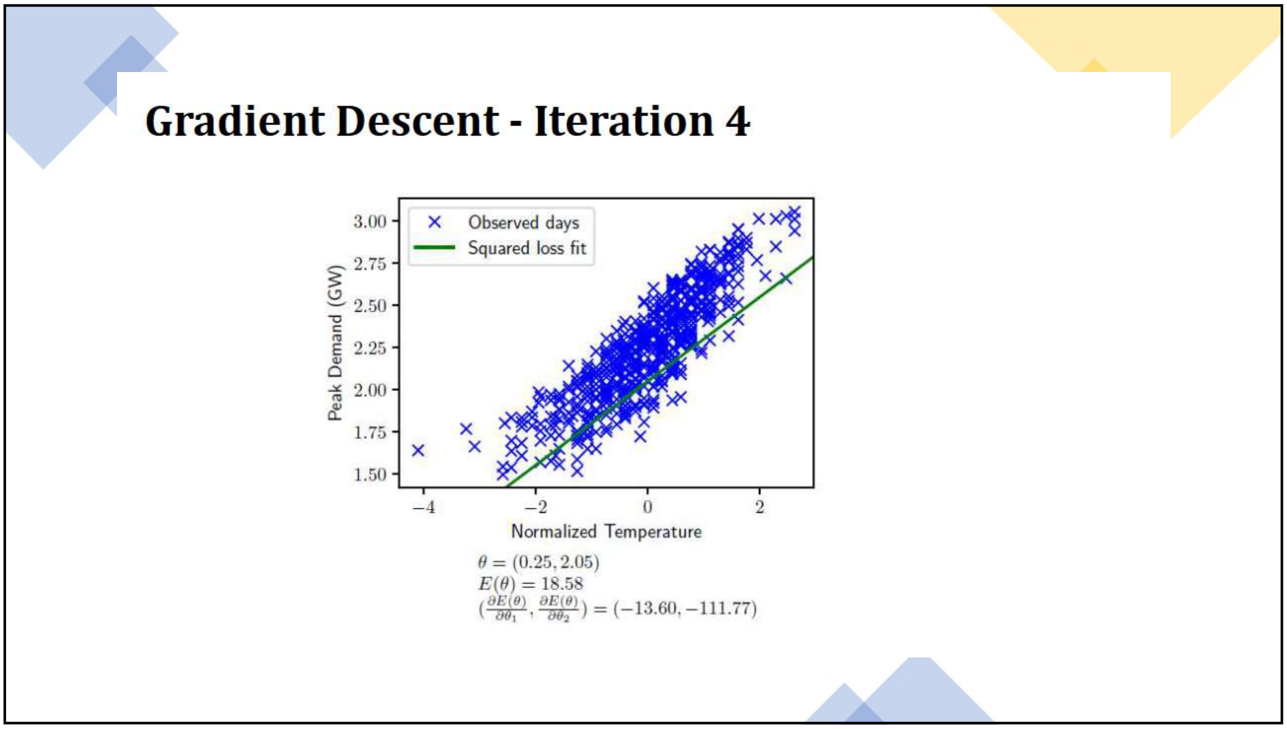
Gradient Descent - Iteration 3

$\theta = (0.22, 1.80)$

$E(\theta) = 64.31$

$(\frac{\partial E(\theta)}{\partial \theta_1}, \frac{\partial E(\theta)}{\partial \theta_2}) = (-30.35, -249.50)$

43



Gradient Descent - Iteration 4

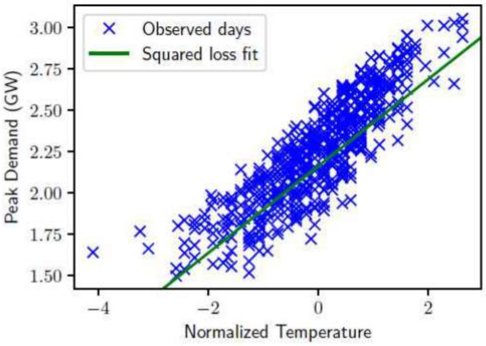
$\theta = (0.25, 2.05)$

$E(\theta) = 18.58$

$(\frac{\partial E(\theta)}{\partial \theta_1}, \frac{\partial E(\theta)}{\partial \theta_2}) = (-13.60, -111.77)$

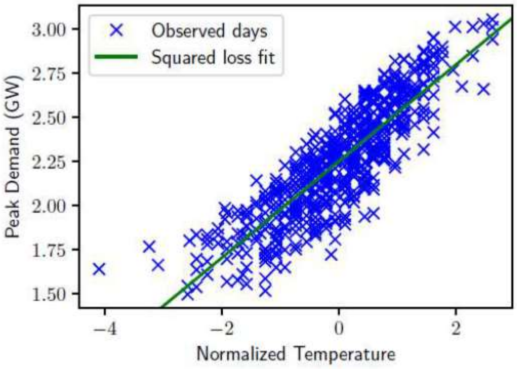
44

Gradient Descent - Iteration 5



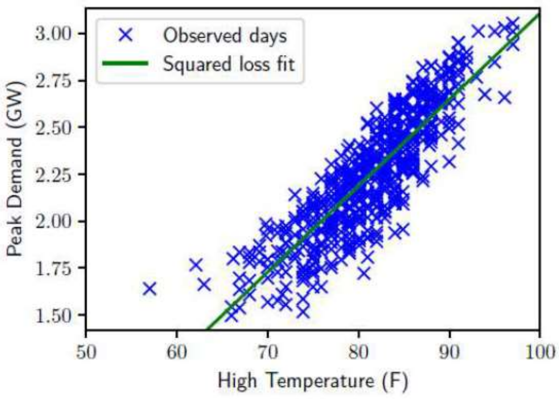
$\theta = (0.26, 2.16)$
$E(\theta) = 9.40$
$(\frac{\partial E(\theta)}{\partial \theta_1}, \frac{\partial E(\theta)}{\partial \theta_2}) = (-6.09, -50.07)$

Gradient Descent - Iteration 10



$\theta = (0.27, 2.25)$
$E(\theta) = 7.09$
$(\frac{\partial E(\theta)}{\partial \theta_1}, \frac{\partial E(\theta)}{\partial \theta_2}) = (-0.11, -0.90)$

Fitted Line in Original Coordinates



47

Making Prediction

Importantly, our model also lets us make *predictions* about new days

What will the peak demand be tomorrow?

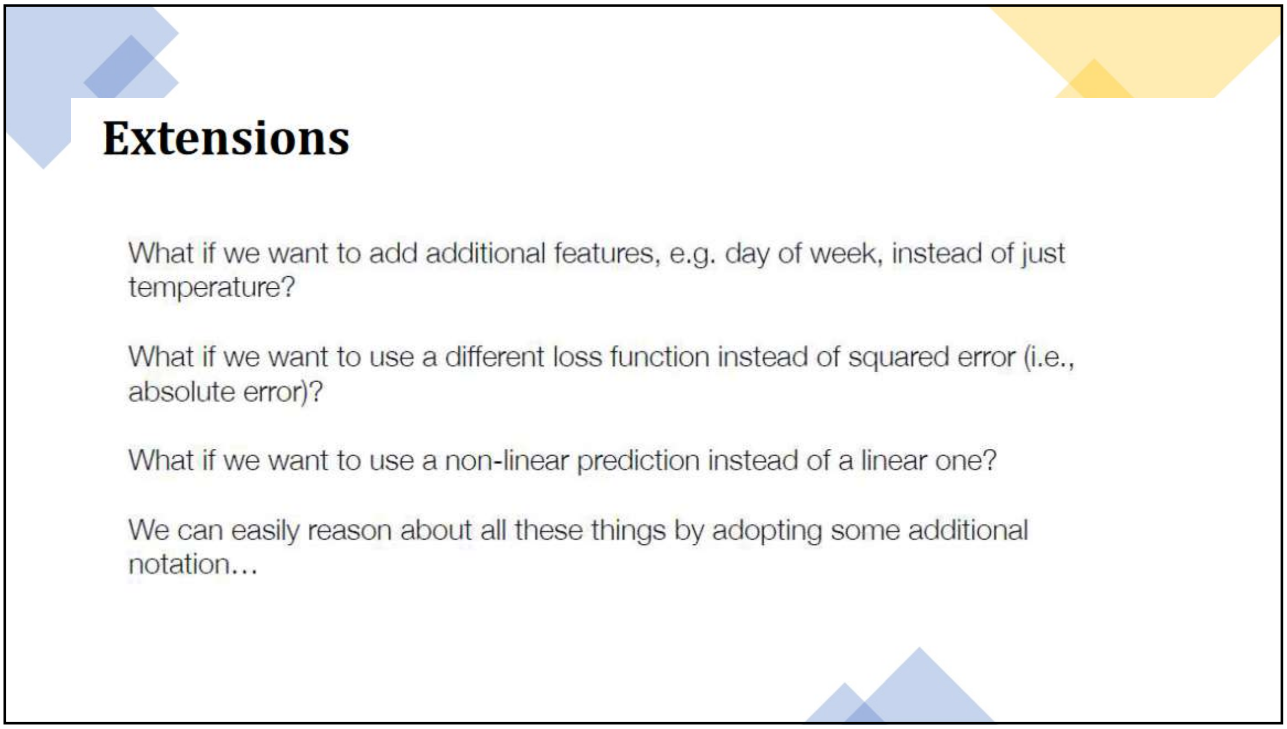
If we know the high temperature will be 72 degrees (ignoring for now that this is *also* a prediction), then we can predict peak demand to be:

$$\text{Predicted\_demand} = \theta_1 \cdot 72 + \theta_2 = 1.821 \text{ GW}$$

(requires that we rescale θ after solving to “normal” coordinates)

Equivalent to just “finding the point on the line”

48



Extensions

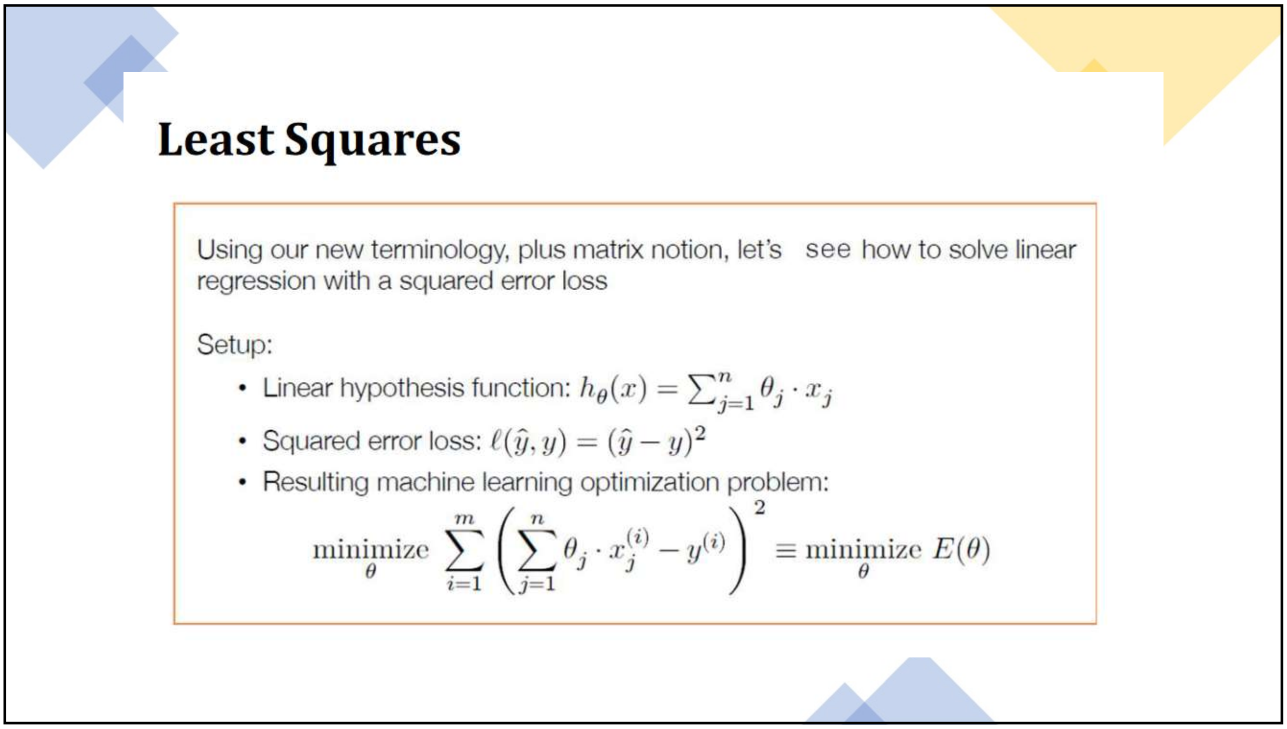
What if we want to add additional features, e.g. day of week, instead of just temperature?

What if we want to use a different loss function instead of squared error (i.e., absolute error)?

What if we want to use a non-linear prediction instead of a linear one?

We can easily reason about all these things by adopting some additional notation…

49



Least Squares

Using our new terminology, plus matrix notion, let's see how to solve linear regression with a squared error loss

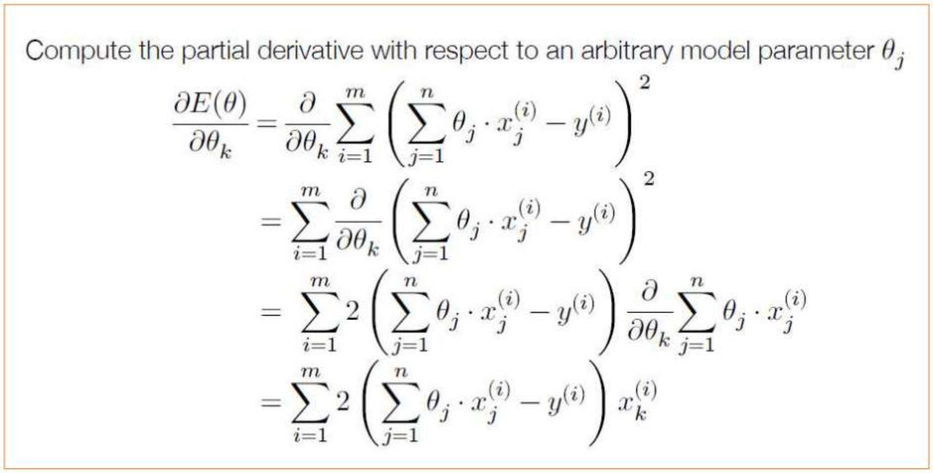
Setup:

- Linear hypothesis function: $h_\theta(x) = \sum_{j=1}^{n} \theta_j \cdot x_j$
- Squared error loss: $\ell(\hat{y}, y) = (\hat{y} - y)^2$
- Resulting machine learning optimization problem:

$$\underset{\theta}{\text{minimize}} \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \theta_j \cdot x_j^{(i)} - y^{(i)} \right)^2 \equiv \underset{\theta}{\text{minimize}}\ E(\theta)$$

50

Derivative of the Least Squares Objective



Compute the partial derivative with respect to an arbitrary model parameter θ_j

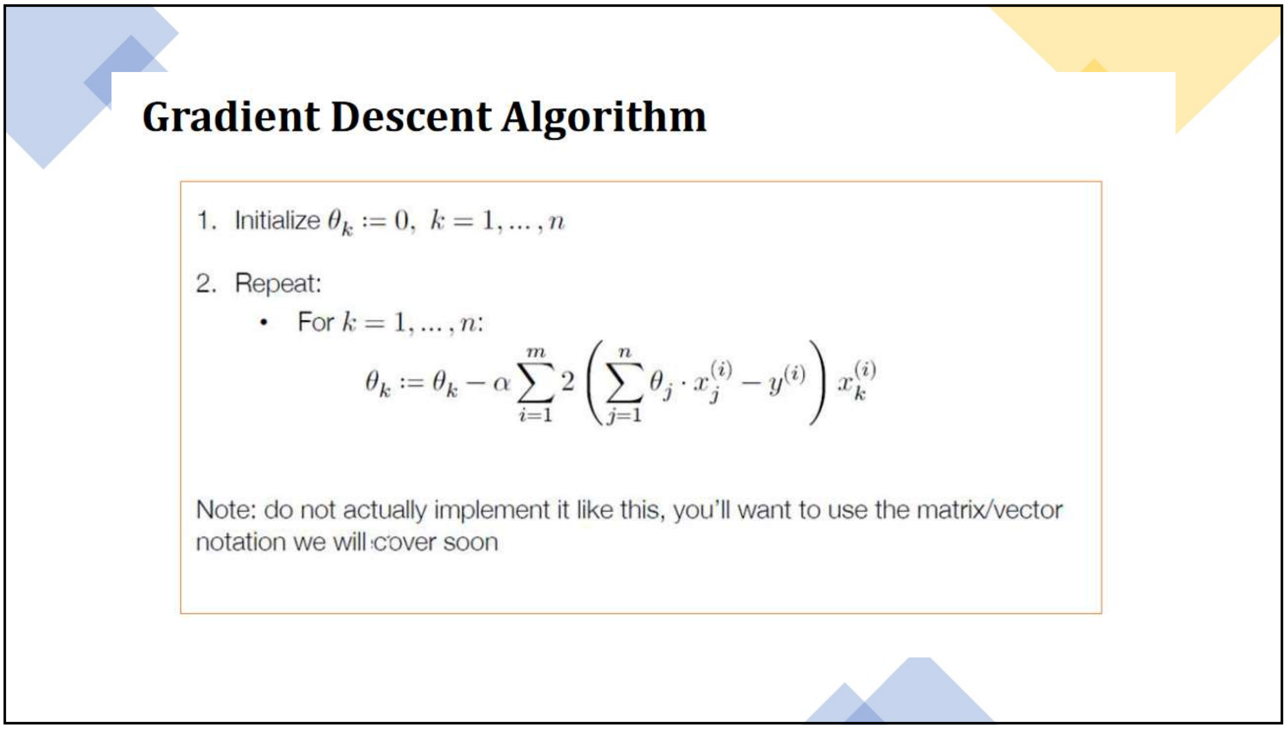
$$
\begin{aligned}
\frac{\partial E(\theta)}{\partial \theta_k} &= \frac{\partial}{\partial \theta_k} \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \theta_j \cdot x_j^{(i)} - y^{(i)} \right)^2 \\
&= \sum_{i=1}^{m} \frac{\partial}{\partial \theta_k} \left(\sum_{j=1}^{n} \theta_j \cdot x_j^{(i)} - y^{(i)} \right)^2 \\
&= \sum_{i=1}^{m} 2 \left(\sum_{j=1}^{n} \theta_j \cdot x_j^{(i)} - y^{(i)} \right) \frac{\partial}{\partial \theta_k} \sum_{j=1}^{n} \theta_j \cdot x_j^{(i)} \\
&= \sum_{i=1}^{m} 2 \left(\sum_{j=1}^{n} \theta_j \cdot x_j^{(i)} - y^{(i)} \right) x_k^{(i)}
\end{aligned}
$$

Note: E(θ) is same cost function J(θ)

51

Let’s check the coding demo…

52

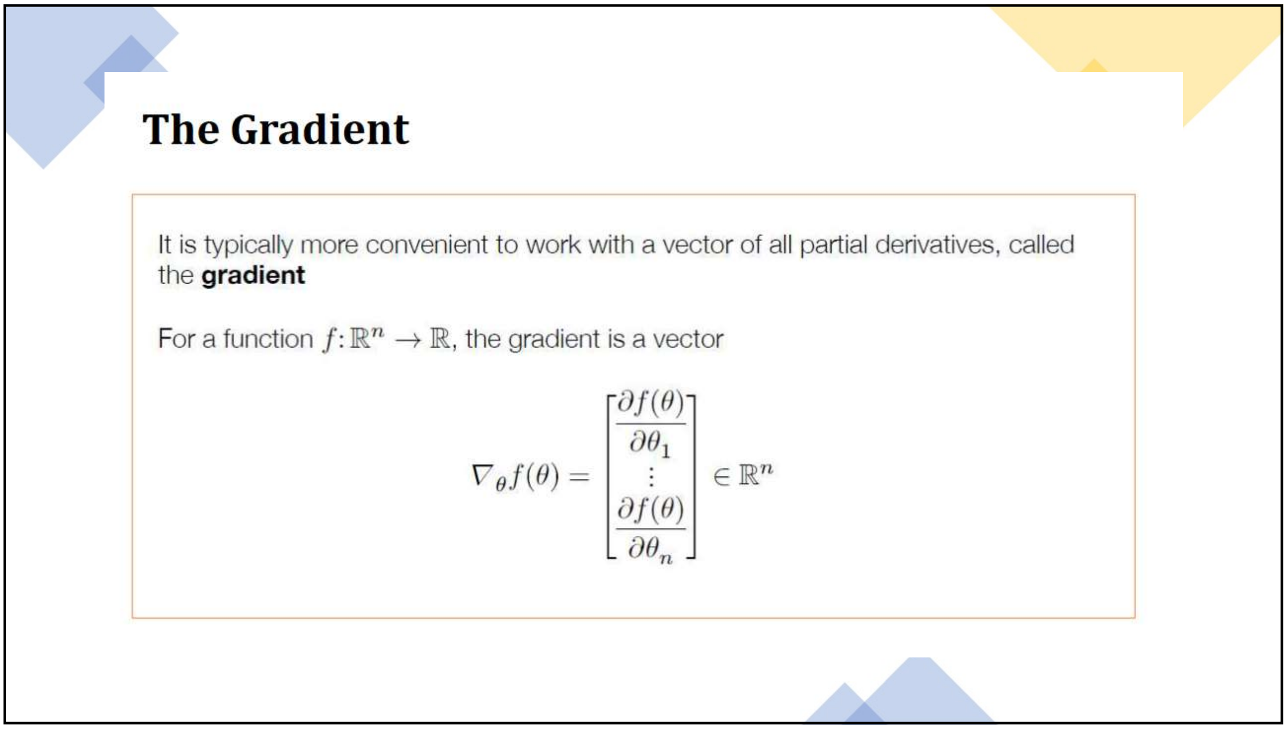


Gradient Descent Algorithm

```
1. Initialize $\theta_k := 0,\ k = 1, \ldots, n$
2. Repeat:
    • For $k = 1, \ldots, n$:
          $\theta_k := \theta_k - \alpha \sum_{i=1}^{m} 2 \left( \sum_{j=1}^{n} \theta_j \cdot x_j^{(i)} - y^{(i)} \right) x_k^{(i)}$
```

Note: do not actually implement it like this, you'll want to use the matrix/vector notation we will cover soon

53



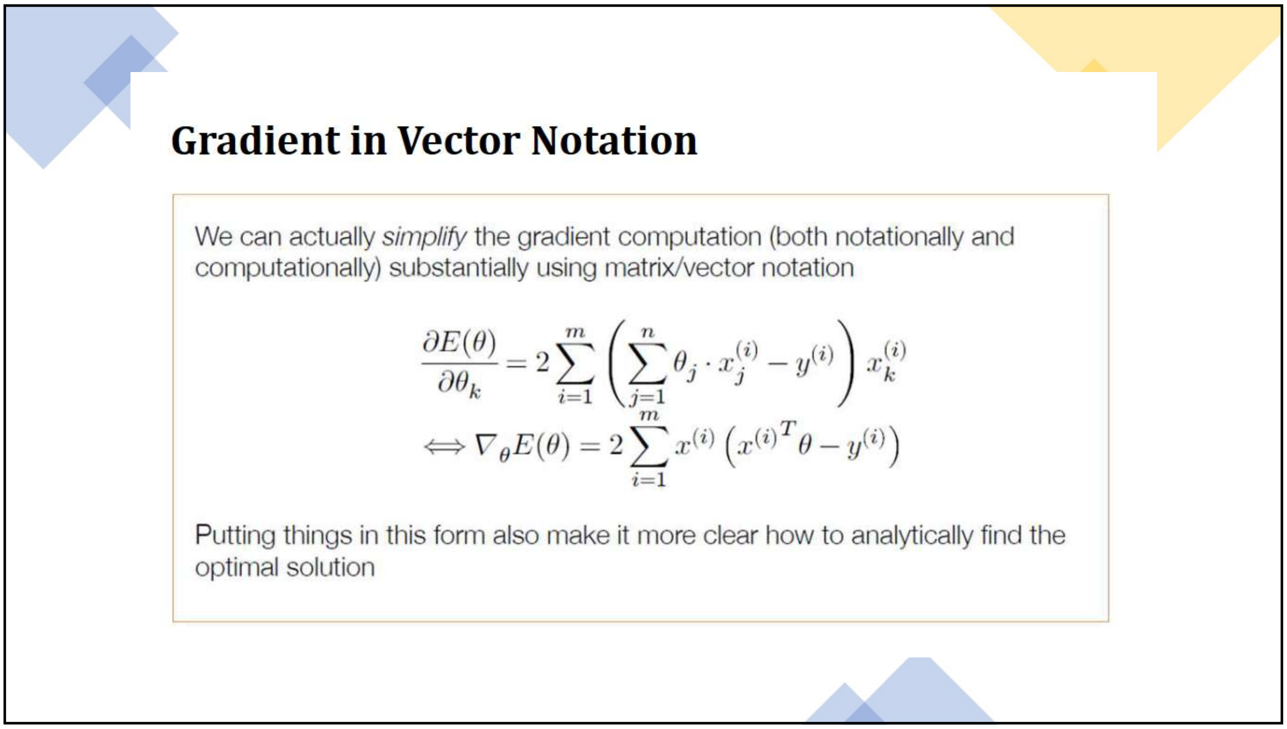
The Gradient

It is typically more convenient to work with a vector of all partial derivatives, called the **gradient**

For a function $f\colon \mathbb{R}^n \to \mathbb{R}$, the gradient is a vector

$$\nabla_\theta f(\theta) = \begin{bmatrix} \dfrac{\partial f(\theta)}{\partial \theta_1} \\ \vdots \\ \dfrac{\partial f(\theta)}{\partial \theta_n} \end{bmatrix} \in \mathbb{R}^n$$

54



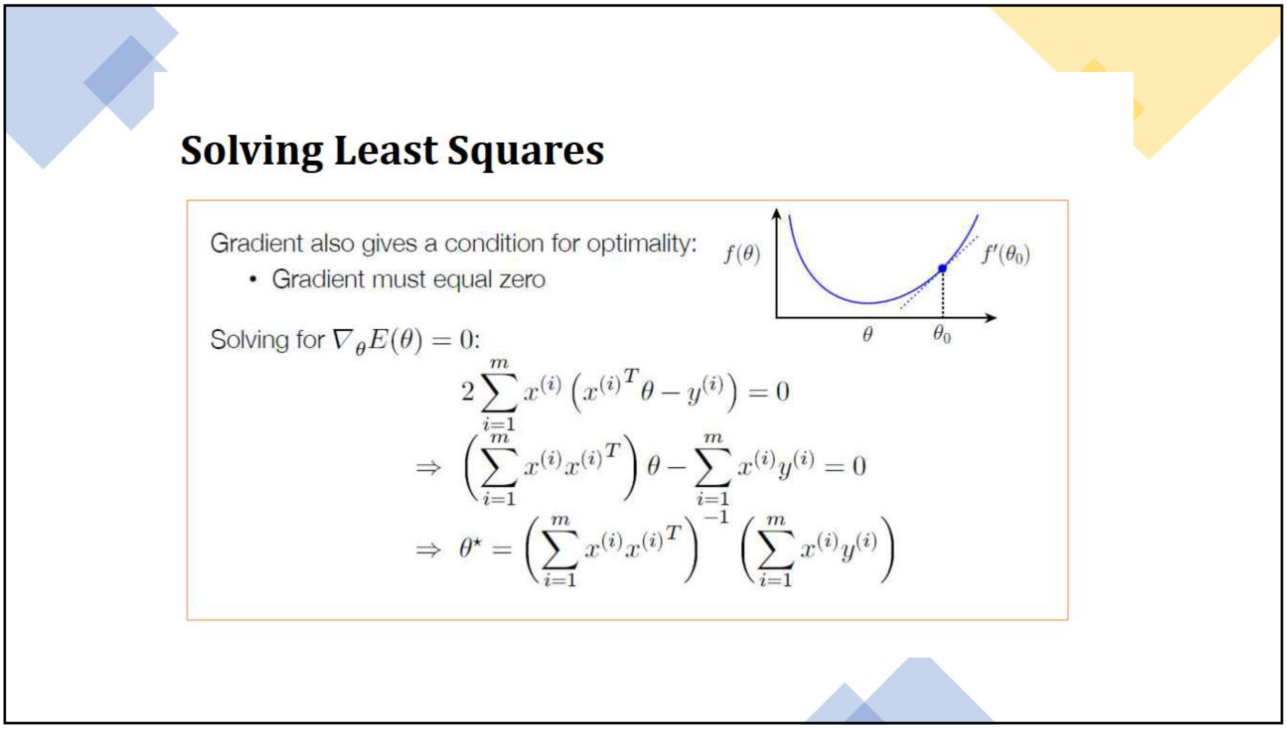
Gradient in Vector Notation

We can actually *simplify* the gradient computation (both notationally and computationally) substantially using matrix/vector notation

$$\frac{\partial E(\theta)}{\partial \theta_k} = 2\sum_{i=1}^{m}\left(\sum_{j=1}^{n}\theta_j \cdot x_j^{(i)} - y^{(i)}\right) x_k^{(i)}$$

$$\Longleftrightarrow \nabla_\theta E(\theta) = 2\sum_{i=1}^{m} x^{(i)}\left({x^{(i)}}^T\theta - y^{(i)}\right)$$

Putting things in this form also make it more clear how to analytically find the optimal solution



Solving Least Squares

Gradient also gives a condition for optimality:

- Gradient must equal zero

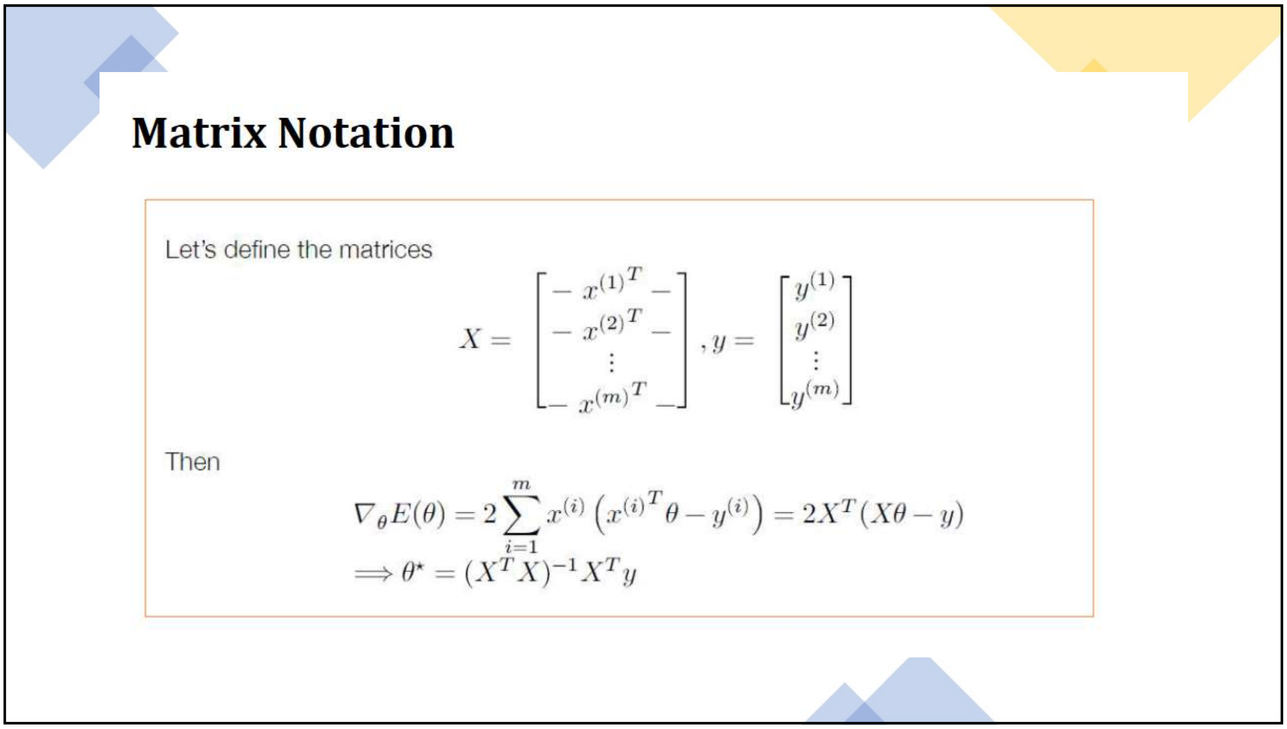
$f(\theta)$
$f'(\theta_0)$
θ
θ_0

Solving for $\nabla_\theta E(\theta) = 0$:

$$2\sum_{i=1}^{m} x^{(i)}\left({x^{(i)}}^T\theta - y^{(i)}\right) = 0$$

$$\Rightarrow \left(\sum_{i=1}^{m} x^{(i)}{x^{(i)}}^T\right)\theta - \sum_{i=1}^{m} x^{(i)}y^{(i)} = 0$$

$$\Rightarrow \theta^\star = \left(\sum_{i=1}^{m} x^{(i)}{x^{(i)}}^T\right)^{-1}\left(\sum_{i=1}^{m} x^{(i)}y^{(i)}\right)$$



Matrix Notation

Let's define the matrices

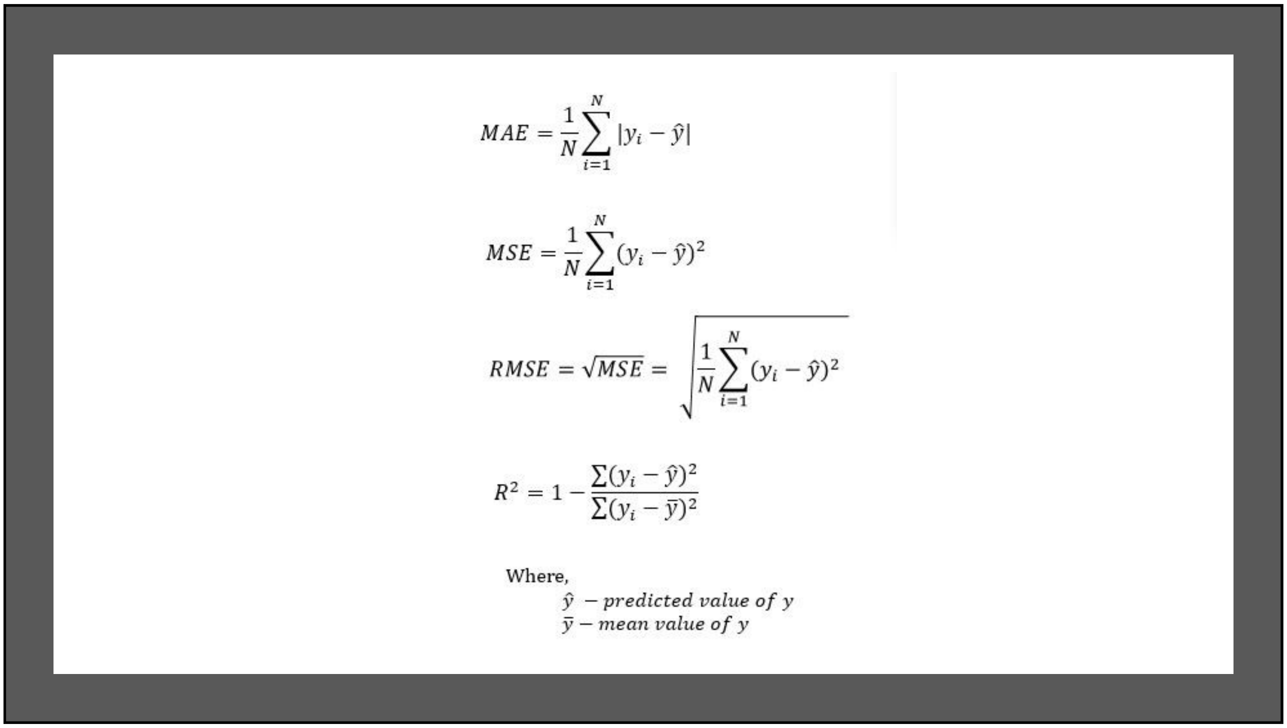
$$X = \begin{bmatrix} - & {x^{(1)}}^T & - \\ - & {x^{(2)}}^T & - \\ & \vdots & \\ - & {x^{(m)}}^T & - \end{bmatrix}, y = \begin{bmatrix} y^{(1)} \\ y^{(2)} \\ \vdots \\ y^{(m)} \end{bmatrix}$$

Then

$$\nabla_\theta E(\theta) = 2 \sum_{i=1}^{m} x^{(i)} \left({x^{(i)}}^T \theta - y^{(i)} \right) = 2X^T(X\theta - y)$$

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

57



$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y_i - \hat{y}|$$

$$MSE = \frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y})^2$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y})^2}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2}$$

Where,

$\hat{y}$ $-$ *predicted value of y*

$\bar{y}$ $-$ *mean value of y*

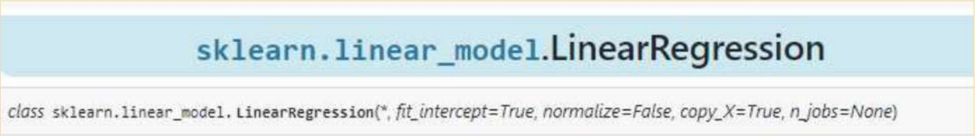
58

How to *build* a Regression Model in *Scikit-Learn*

59

Linear Regression in Scikit-Learn

scikit-learn



sklearn.linear_model.LinearRegression

```
class sklearn.linear_model.LinearRegression(*, fit_intercept=True, normalize=False, copy_X=True, n_jobs=None)
```

- Build an estimator model like LinearRegression()
- Use fit() function to ***Train*** the model with **Training Dataset**
- Use **predict()** function to ***Test/Evaluate*** the model with **Test Dataset**
- Use **predict()** function to make ***prediction/inference*** on **New Unseen Data**

60