Regression model

Supervised learning

Regression models

Q: Why do we need regression?

A: Predict trends (values) of the future outcomes (Y) according to feature inputs (X)

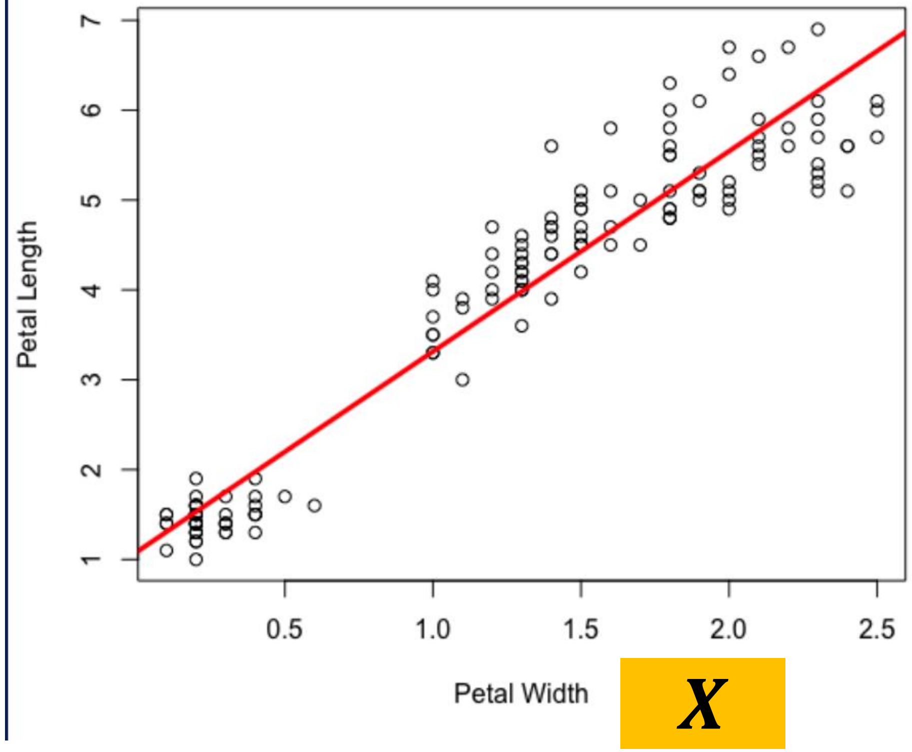
: Approximate a trend model (an equation) of input relations (Curve fitting)

$$\boldsymbol{Y = F(X)}$$

Regression models

$$Y = F(X)$$

- **Linear Regression**
  - **Perform Trend Prediction**
    - **Curve fitting**



X

Techniques for estimating Regression model

$$Y = F(X)$$

- **Example Techniques**
  - **Linear Regression**
    - **Linear approximation without regulation or constraint**
  - **Support Vector Regression**
    - **Linear and Non-linear approximation with constraint**

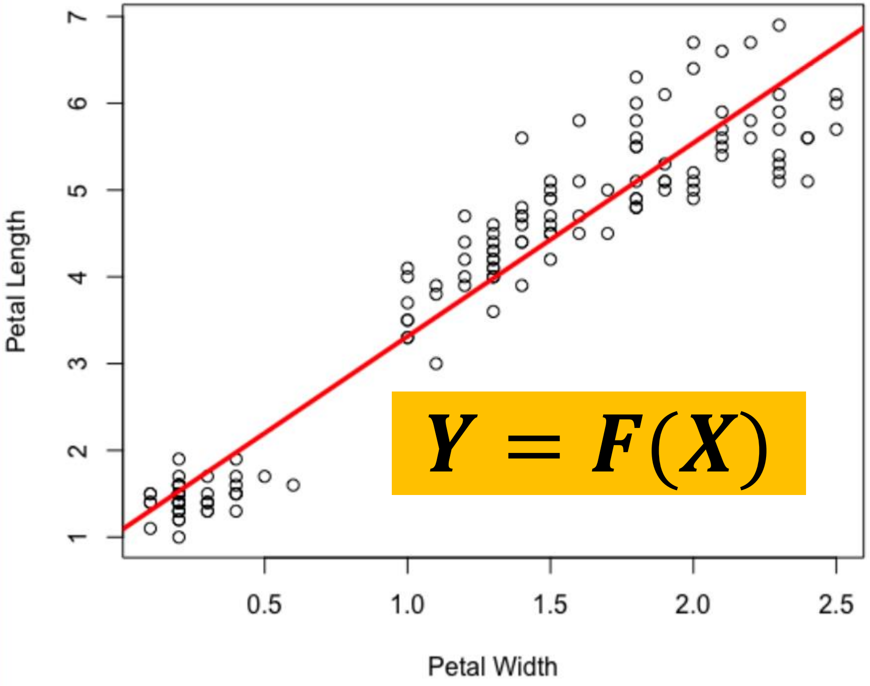
Linear regression

$$Y = F(X)$$

- **Single variable**
- **Multiple variables (Multivariate)**

Linear Regression models

SINGLE VARIABLE



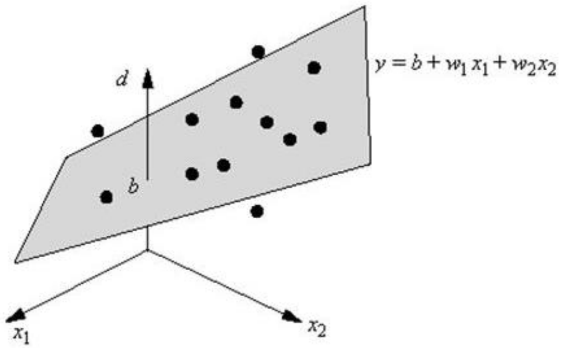
$$Y = F(X)$$

MULTIPLE VARIABLES (MULTIVARIATE)

https://slideplayer.com/slide/5004010/

Part I – MULTIVARIATE ANALYSIS

C2 Multiple Linear Regression I



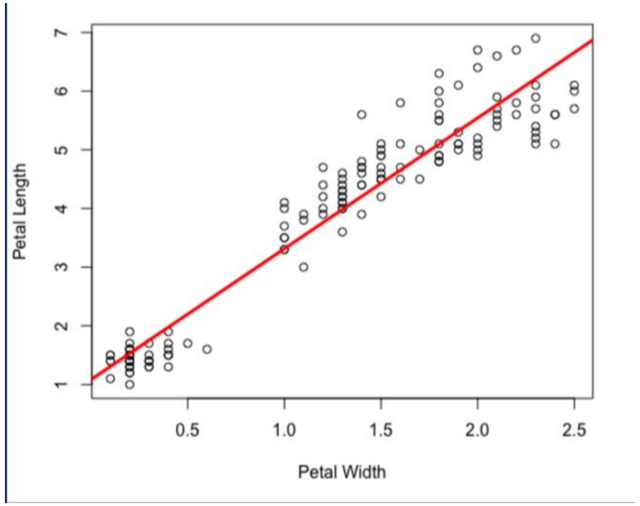
$$Y = F(x_1, x_2, \dots, x_n)$$

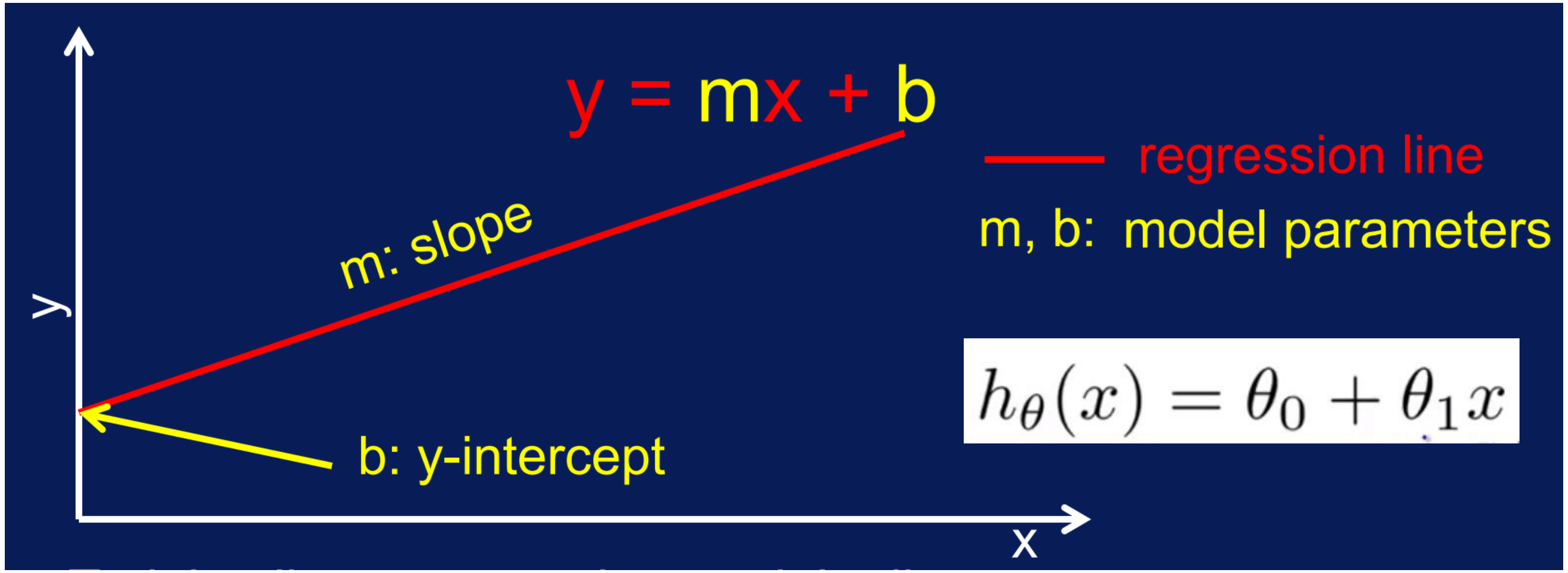
Single variable
Linear regression

$$Y = F(X)$$

- **What would be a model and parameters for single variable linear regression?**

Single variable
Linear Regression models



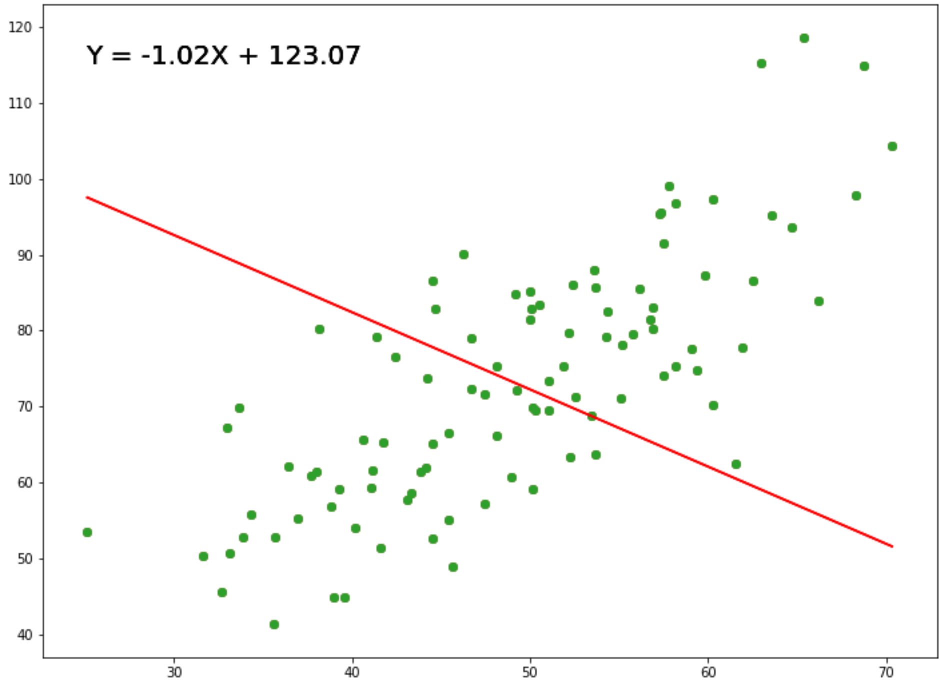


- **Linear Regression Line (a sum of weighted variables + a bias)**
  - **Linear relationship between input x and output** $h_\theta(x)$
    - **With m: slope and b: y-axis intercept parameters**

θ_1 θ_0

How can we estimate the regression parameters?

Techniques for Estimating Regression parameters



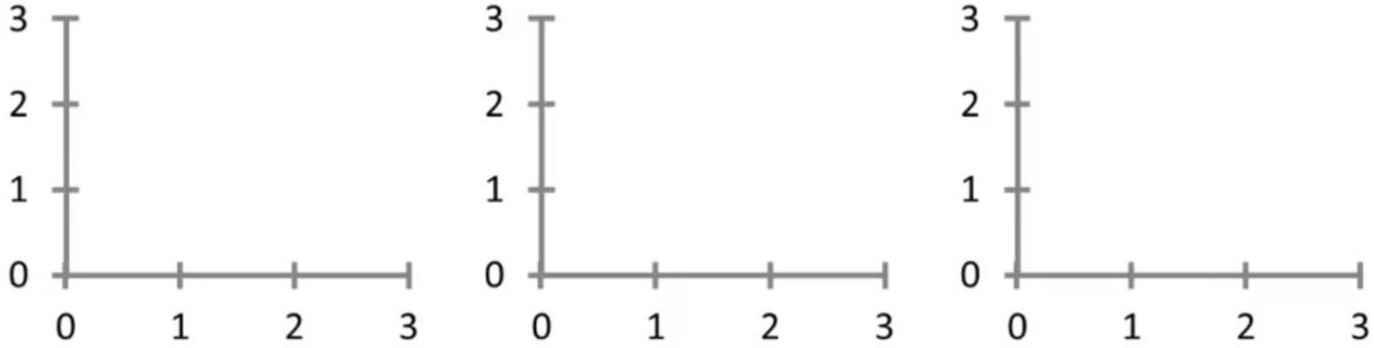
- **Trial and errors**
  - **With interested parameters**
- **Parameter optimization**
  - **Least Square Estimation**
    - Solve linear system
  - **Gradient Decent Search**
    - Search algorithm

Trial-errors

Trial-errors

y = mx + b

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



$\theta_0 = 1.5$, $\theta_1 = 0$ | $\theta_0 = 0$, $\theta_1 = 0.5$ | $\theta_0 = 1$, $\theta_1 = 0.5$

- **Ex.**
  - **Brute force search for whole parameter space**
    - **M1 <= m <= M2 / b1 <= b <= b2**

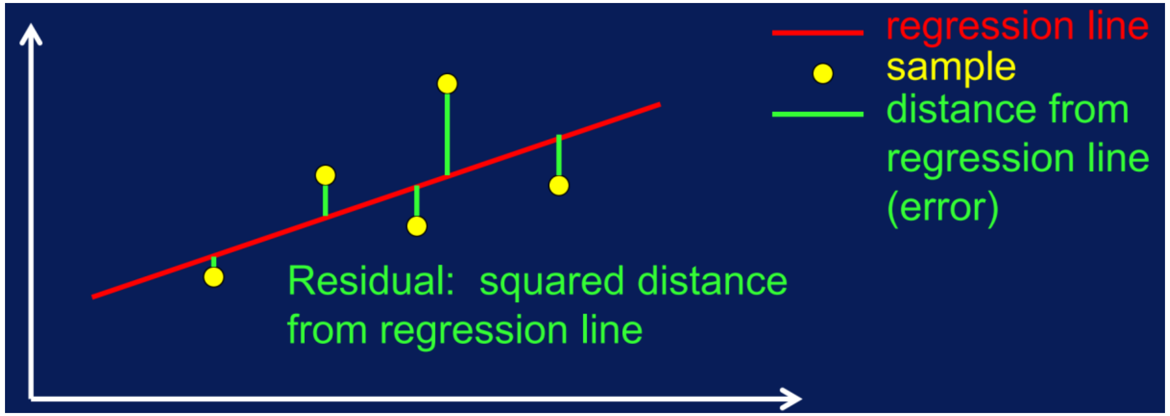
θ_1 θ_0

Least square estimation

for parameter optimization

Least square estimation

for parameter optimization



Goal: Find regression line that makes sum of residuals as small as possible

y = ground truth

$\hat{y}_i = h_\theta$

$= \theta_1 x_i + \theta_0$

= prediction

- **Objective is to find parameters with**
  - **Minimize or Maximize Cost function**
    - **Cost function -> objective function**
      - **Ex. Function of Residual (Difference)**
        - **MSE: Means Square Error**

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

$$= \frac{1}{N}\sum_{i=1}^{N}(y_i - h_\theta)^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}\big(y_i - (\theta_1 x_i + \theta_0)\big)^2$$

Least square estimation

for parameter optimization

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

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

$$= \frac{1}{N}\sum_{i=1}^{N}(y_i - h_\theta)^2$$

$$= \frac{1}{N}\sum_{i=1}^{N}\left(y_i - (\theta_1 x_i + \theta_0)\right)^2$$

$\mathrm{J}(\theta_1, \theta_0)$ = **objective function of MSE**

$$= \frac{1}{N}\sum_{i=1}^{N}\left(y_i - (\theta_1 x_i + \theta_0)\right)^2$$

$$\text{optimum parameters}(\theta_1, \theta_0) = \min_{(\theta_1,\theta_0)} J(\theta_1, \theta_0)$$

$$= \min_{(\theta_1,\theta_0)} MSE$$

$$= \min_{(\theta_1,\theta_0)} \frac{1}{N}\sum_{i=1}^{N}\left(y_i - (\theta_1 x_i + \theta_0)\right)^2$$

$$\frac{\partial \mathrm{J}(\theta_1, \theta_0)}{\partial \theta_1} = 0$$

$$\frac{\partial \mathrm{J}(\theta_1, \theta_0)}{\partial \theta_1} = 0$$

Least square estimation

for parameter optimization

$\mathrm{J}(\theta_1, \theta_0)$ = **objective function of MSE**

$$= \frac{1}{N}\sum_{i=1}^{N}\left(y_i - (\theta_1 x_i + \theta_0)\right)^2$$

$$\frac{\partial \mathrm{J}(\theta_1, \theta_0)}{\partial \theta_1} = -2\sum_{i=1}^{n} x_i\left(y_i - (\theta_1 x_i + \theta_0)\right) = 0 \quad \Rightarrow \quad \theta_1 \sum_{i=1}^{n} x_i^2 + \theta_0 \sum_{i=1}^{n} x_i = \sum_{i=1}^{n} x_i y_i$$

$$\frac{\partial \mathrm{J}(\theta_1, \theta_0)}{\partial \theta_0} = -2\sum_{i=1}^{n} \left(y_i - (\theta_1 x_i + \theta_0)\right) = 0 \quad \Rightarrow \quad \theta_1 \sum_{i=1}^{n} x_i + n\theta_0 = \sum_{i=1}^{n} y_i$$

Solve Linear Equations for θ_1, θ_0

$$\theta_1 = \frac{\left(\sum_{i=1}^{n} x_i y_i\right) - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

$$\theta_0 = \frac{\bar{y}\left(\sum_{i=1}^{n} x_i^2\right) - \bar{x}\left(\sum_{i=1}^{n} x_i y_i\right)}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2} = \frac{\bar{y}\left(\sum_{i=1}^{n} x_i^2\right) - \bar{x}\left(\sum_{i=1}^{n} x_i y_i\right)}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \bar{y} - \theta_1 \bar{x}$$

http://mathworld.wolfram.com/LeastSquaresFitting.html

- **Example**

Least square estimation

for parameter optimization

i	xi	yi	$(x_i - \bar{x})$	$(y_i - \bar{y})$	$(x_i - \bar{x})(y_i - \bar{y})$	$(x_i - \bar{x})^2$
1	63	127				
2	64	121				
3	66	142				
4	69	157				
5	69	162				
6	71	156				
7	71	169				
8	72	165				
9	73	181				
10	75	208				
$\bar{x}$						
$\bar{y}$						
$\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})$						
$\sum_{i=1}^{n}(x_i - \bar{x})^2$						

$$\theta_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2}$$

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

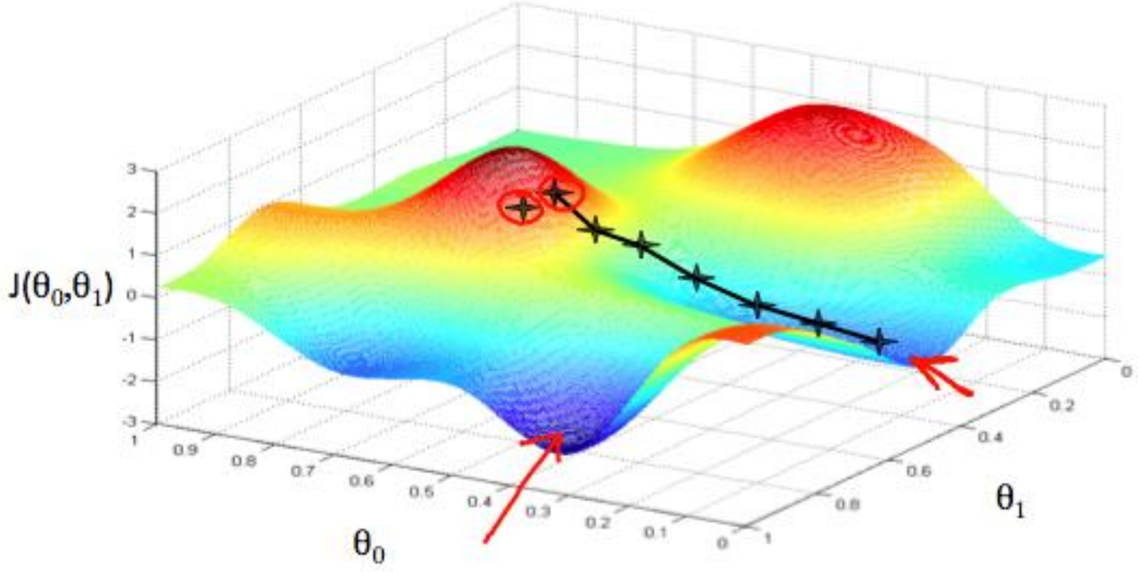
$$h_\theta = \theta_0 + \theta_1 x_i$$

Gradient descent estimation

for parameter optimization

Gradient descent estimation

for parameter optimization



The gradient descent algorithm is:

repeat until convergence:

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

Correct: Simultaneous update

$$\begin{aligned}
&\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) \\
&\text{temp1} := \theta_1 - \alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1) \\
&\theta_0 := \text{temp0} \\
&\theta_1 := \text{temp1}
\end{aligned}$$

Incorrect:

$$\begin{aligned}
&\text{temp0} := \theta_0 - \alpha \frac{\partial}{\partial \theta_0} J(\theta_0, \theta_1) \\
&\theta_0 := \text{temp0} \\
&\text{temp1} := \theta_1 - \boxed{\alpha \frac{\partial}{\partial \theta_1} J(\theta_0, \theta_1)} \\
&\theta_1 := \text{temp1}
\end{aligned}$$

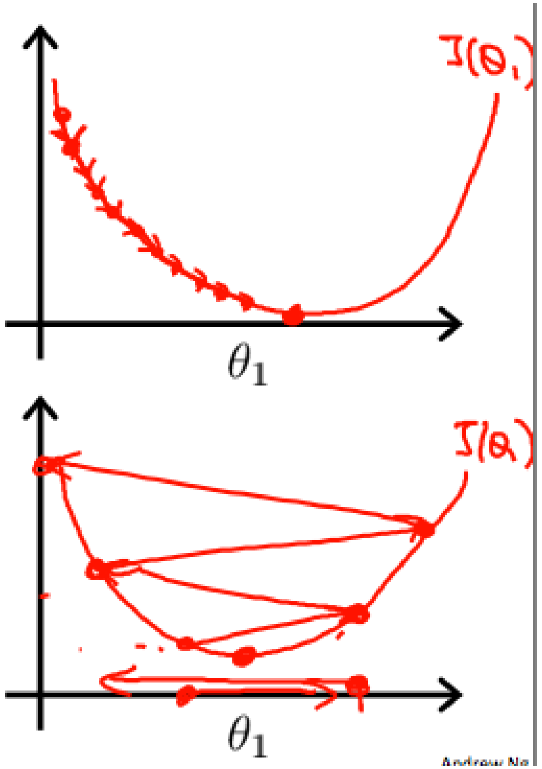
Gradient descent estimation

for parameter optimization

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

If α is too small, gradient descent can be slow.

If α is too large, gradient descent can overshoot the minimum. It may fail to converge, or even diverge.



How does gradient descent converge with a fixed step size α?

The intuition behind the convergence is that

$$\frac{\partial J}{\partial \theta} = 0$$

as we approach the bottom of our convex function.

At the minimum, the derivative will always be 0 and thus we get:

$$\theta_1 := \theta_1 - \alpha * 0$$

https://www.coursera.org/learn/machine-learning/supplement/QKEdR/gradient-descent-intuitio

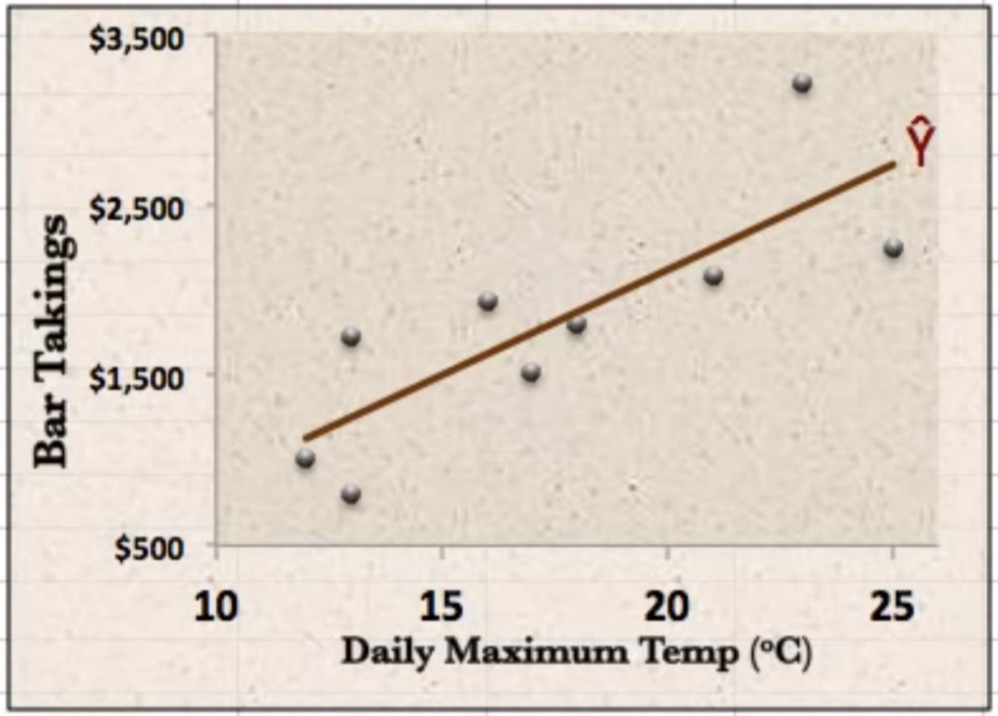
How can we measure the accuracy of the regression parameters?

Model evaluation

Day	Takings	Temp (°C)
3-Jun	$3,213	23
10-Jun	$2,089	21
17-Jun	$2,253	25
24-Jun	$1,801	18
1-Jul	$801	13
8-Jul	$1,934	16
15-Jul	$1,720	13
22-Jul	$1,514	17
29-Jul	$1,017	12

SAMPLE REGRESSION LINE

$\hat{Y} = -353.11 + 123.54X$



- **Evaluation Criteria:**
- **A**ccuracy- using the coefficient of determination
- **R**obustness- using hypothesis testing

https://www.youtube.com/watch?v=aq8VU5KLmkY&list=PLTNMv857s9WUI1Nz4SssXDKAELESXz-bi

$$MAE = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

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

$$RMSE = \sqrt{\frac{1}{n}\sum_{j=1}^{n}(y_j - \hat{y}_j)^2}$$

l evaluation

MSE (Minimum Mean Square Error) = $\frac{\sum(Y_i - \hat{Y})^2}{N} = \frac{SSE}{N}$

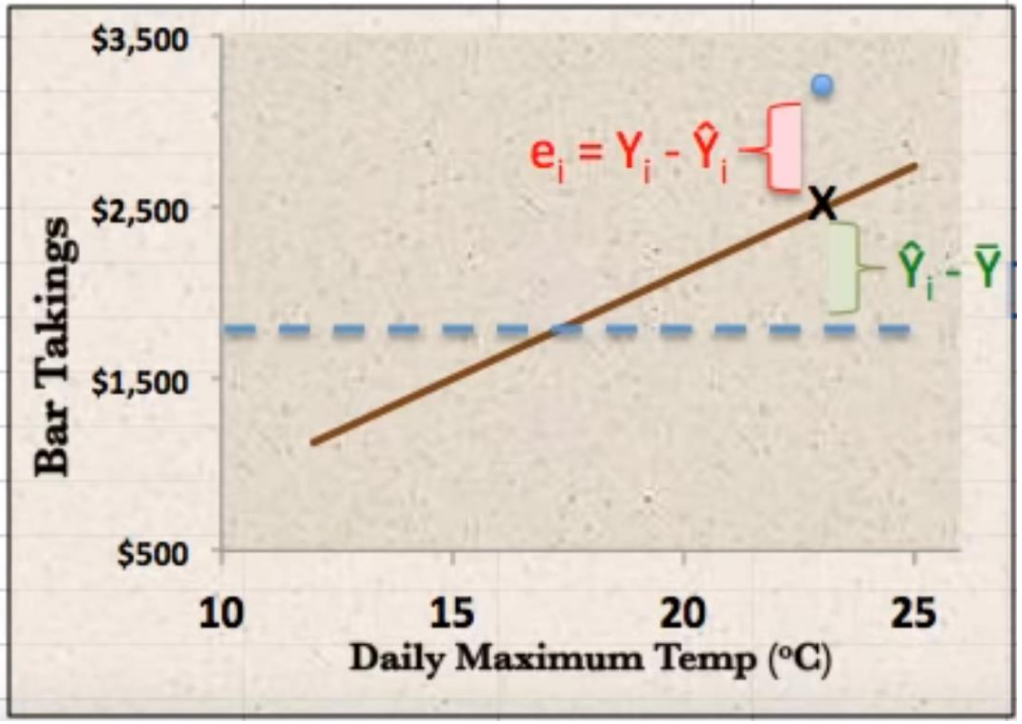
$SSR = \Sigma(\hat{Y}_i - \bar{Y})^2$

$SSE = \Sigma(Y_i - \hat{Y}_i)^2$

$SST = SSR + SSE$

$SST = \Sigma(Y_i - \bar{Y})^2$

$R^2 = SSR/SST$

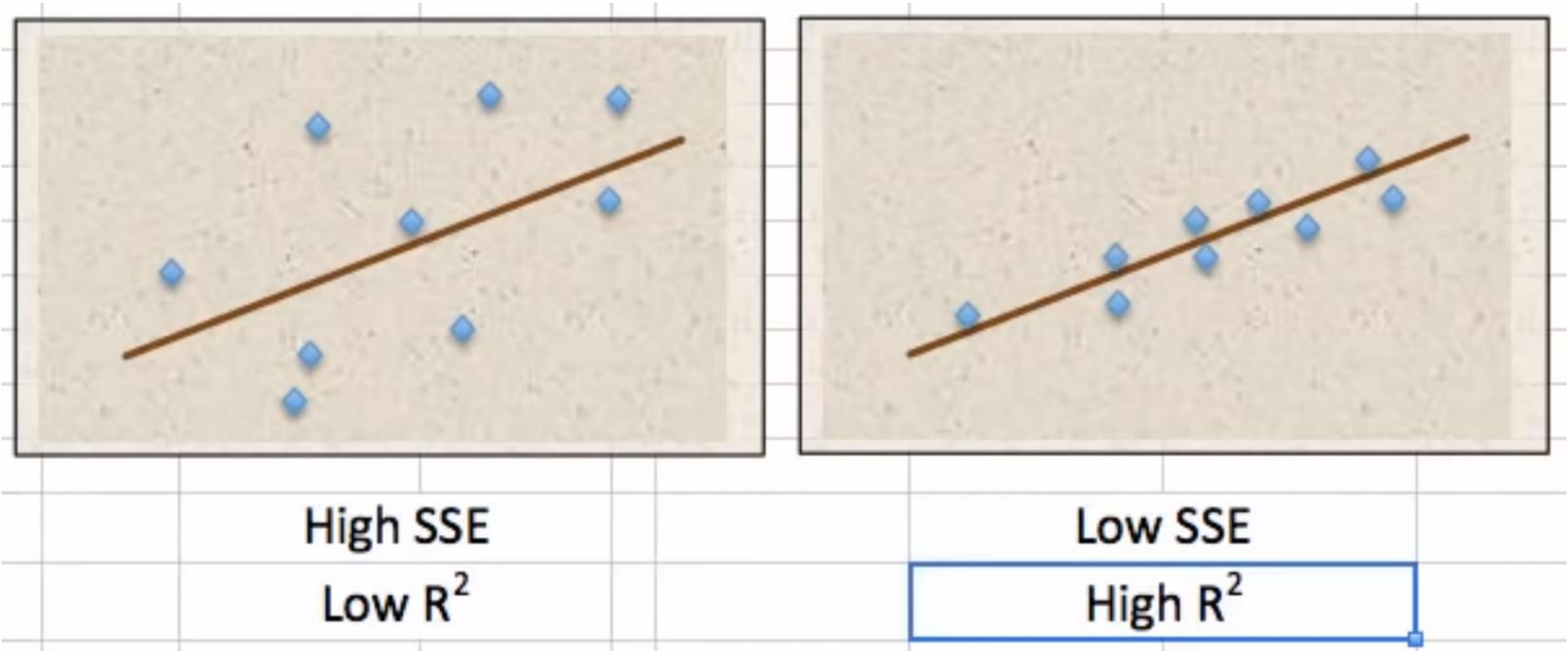


- **Evaluation Criteria:**

- Accuracy- using the coefficient of determination
  - R-squared
    - Between [0,1]

https://www.youtube.com/watch?v=aq8VU5KLmkY&list=PLTNMv857s9WUI1Nz4SssXDKAELESXz-bi

Model evaluation

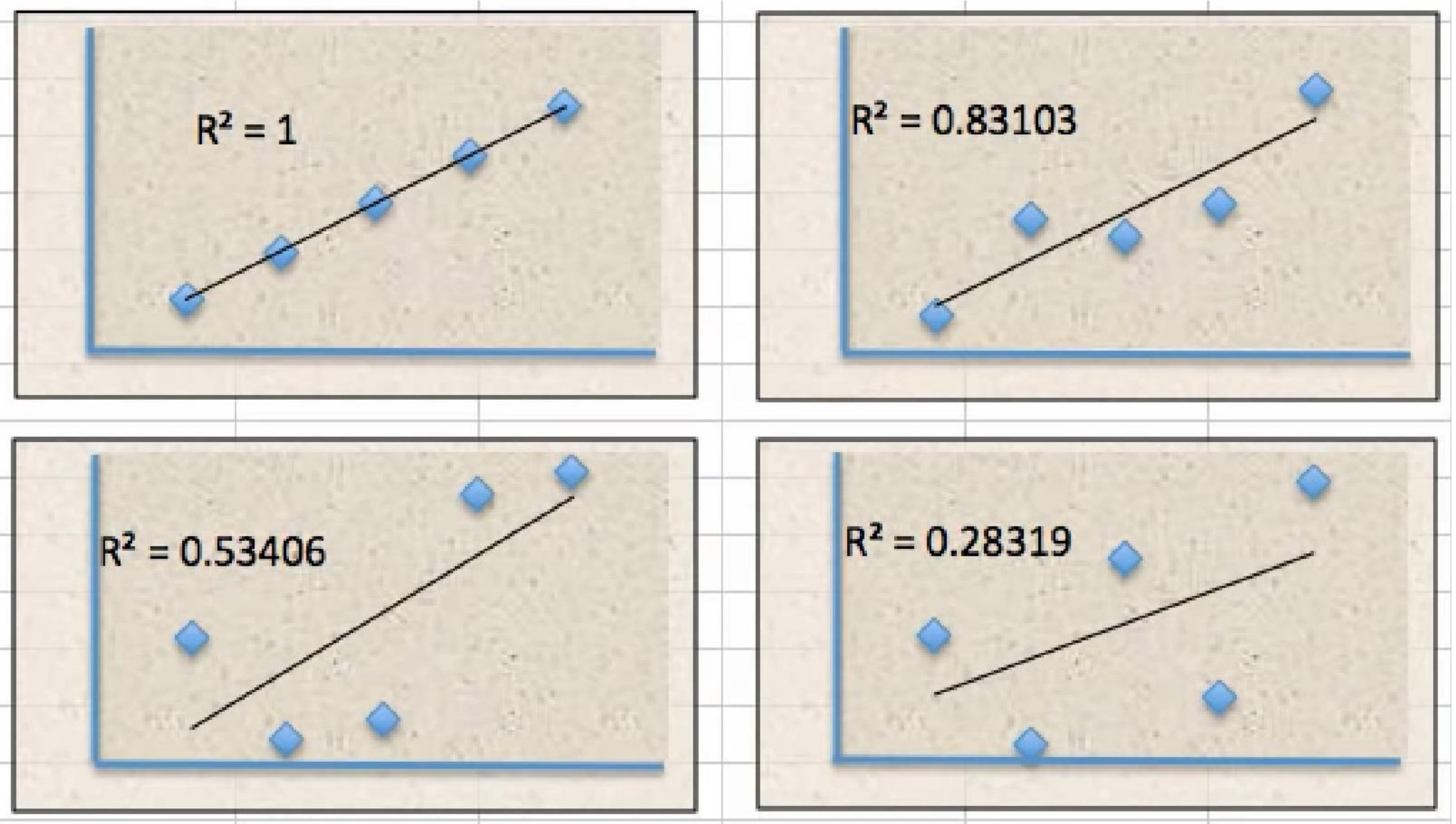


- **Evaluation Criteria:**

- Accuracy- using the coefficient of determination
  - R-squared
    - R-High: Low Error (SSE)
      - Better fit
    - R-Low: High Error (SSE)
      - Not fitting well enough

https://www.youtube.com/watch?v=aq8VU5KLmkY&list=PLTNMv857s9WUI1Nz4SssXDKAELESXz-bi

Model evaluation

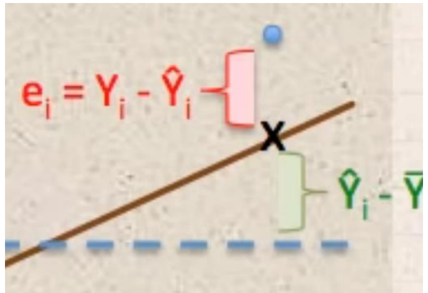


- **Evaluation Criteria:**

- **A**ccuracy- using the coefficient of determination
  - R-squared
    - R-High: Low Error (SSE)
      - Better fit
    - R-Low: High Error (SSE)
      - Not fitting well enough

https://www.youtube.com/watch?v=aq8VU5KLmkY&list=PLTNMv857s9WUI1Nz4SssXDKAELESXz-bi

Model evaluation



Day	Bar Takings (y)	Temp (x)	Y predict	Y- avg(y)	(Y- avg(y))^2	Y-y	(Y-y)^2
03-Jun	3213	23	2488.31	550.31	302841.096	-724.69	525175.6
10-Jun	3089	21	2241.23	303.23	91948.4329	-847.77	718714
17-Jun	2253	25	2735.39	797.39	635830.812	482.39	232700.1
24-Jun	1801	18	1870.61	-67.39	4541.4121	69.61	4845.552
01-Jul	901	13	1252.91	-685.09	469348.308	351.91	123840.6
08-Jul	1934	16	1623.53	-314.47	98891.3809	-310.47	96391.62
15-Jul	1720	13	1252.91	-685.09	469348.308	-467.09	218173.1
22-Jul	1514	17	1747.07	-190.93	36454.2649	233.07	54321.62
29-Jul	1017	12	1129.37	-808.63	653882.477	112.37	12627.02
	1938	17.55555556			2763086.49		1986789
					SSR		SSE
				SST	4749875.7		
				R-square	0.58171764		

$$SST = SSR + SSE$$
$$SST = \Sigma(Y_i - \bar{Y})^2$$

$$R^2 = SSR/SST$$

$$SSR = \Sigma(\hat{Y}_i - \bar{Y})^2$$
$$SSE = \Sigma(Y_i - \hat{Y}_i)^2$$

Accuracy measurement

- **Based on R-square**

$$Y_i = h_\theta = \theta_0 + \theta_1 x_i$$

i	xi	yi	predicted Y	$(Y_i - \hat{Y}_i)$	$(Y_i - \hat{Y}_i)^2$	$(\hat{Y}_i - \bar{Y})$	$(\hat{Y}_i - \bar{Y})^2$
1	63	127					
2	64	121					
3	66	142					
4	69	157					
5	69	162					
6	71	156					
7	71	169					
8	72	165					
9	73	181					
10	75	208					

$SSR = \Sigma(\hat{Y}_i - \bar{Y})^2$

$SSE = \Sigma(Y_i - \hat{Y}_i)^2$

$SST = SSR + SSE$

$SST = \Sigma(Y_i - \bar{Y})^2$

$R^2 = SSR/SST$

What would be a different between
single vs multiple variables regression parameters?

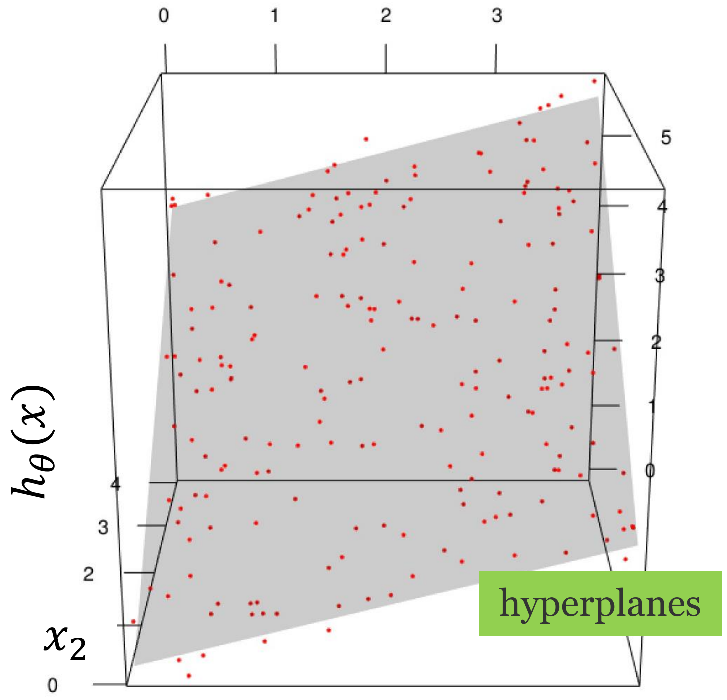
$$h_\theta = \theta_0 + \theta_1 x$$

$$h_\theta = \theta_0 + \theta_1 x_1 + \theta_1 x_2 + \cdots + \theta_1 x_n$$

Multivariate regression model

$$h_\theta(x) = [\theta_0 \quad \theta_1 \quad \dots \quad \theta_n] \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T x$$

- Model can be viewed as a dot product between
    - model parameters and input featur $\theta^T x$



$x_0 = 1, \theta_0 = b$ $\quad x_1$

Multivariate regression parameter estimation

$$h_\theta(x) = \begin{bmatrix} \theta_0 & \theta_1 & \dots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T x$$

- **L**east **S**quare Approximation
- **G**radient **D**escent

Least Square estimation

- Why should people think that Least Squares regression is the "right" kind of linear regression?

- (a) It was invented by Carl Friedrich Gauss (one of the world's most famous mathematicians) in about 1795, and then rediscovered by Adrien-Marie Legendre (another famous mathematician) in 1805, making it one of the earliest general prediction methods known to humankind.
- (b) It is easy to implement on a computer using commonly available algorithms from linear algebra.
- (c) Its implementation on modern computers is efficient, so it can be very quickly applied even to problems with hundreds of features and tens of thousands of data points.
- (d) It is easier to analyze mathematically than many other regression techniques.
- (e) It is not too difficult for non-mathematicians to understand at a basic level.

https://www.youtube.com/watch?v=2I_AYIECCOQ

Least Square estimation

- **Problems and Pitfalls of Applying Least Squares Regression**
  - Outliers
    - perform very badly
      - It will dramatically shift the least squares solution
  - Large number of variables (features)
    - particularly when
      - # features > # training data points
      - the least squares solution will not be unique, and hence the least squares algorithm will fail
    - Estimation is slow

https://www.youtube.com/watch?v=2I_AYIECCOQ

Multivariate regression model

milesTraveled, (x_1)	numDeliveries, (x_2)	gasPrice, (x3)	travelTime(hrs), (y)
89	4	3.84	7
66	1	3.19	5.4
78	3	3.78	6.6
111	6	3.89	7.4
44	1	3.57	4.8
77	3	3.57	6.4
80	3	3.03	7
66	2	3.51	5.6
109	5	3.54	7.3
76	3	3.25	6.4

Will all variables be necessary?

travelTime = f(milesTraveled, numDeliveries, gasPrice)

$$= \theta_0 + \theta_1.\,milesTraveled + \theta_2.numDeliveries + \theta_3.\,gasPrice$$

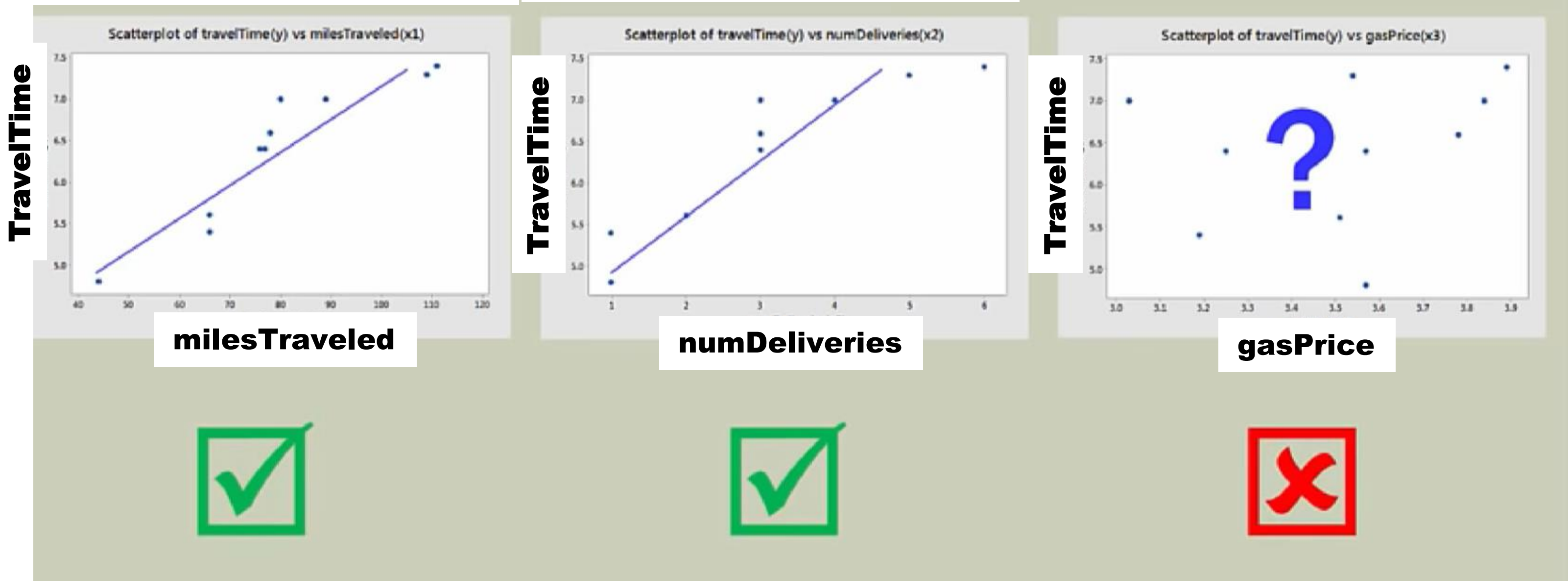
https://www.youtube.com/watch?v=2I_AYIECCOQ

Multivariate regression model

Remove unnecessary input variables

it can be beneficial to only include those features that are likely to be good predictors of our output variable

Scatterplot of TravelTime vs milesTraveled | **Scatterplot of TravelTime vs numDeliveries** | **Scatterplot of TravelTime vs gasPrice**



TravelTime = f(milesTraveled, numDeliveries, ~~gasPrice~~)

$$= \theta_0 + \theta_1 . \textit{milesTraveled} + \theta_2 . \textit{numDeliveries} + \cancel{\theta_3 . \textit{gasPrice}}$$

Remove gasPrice from input variable since it does not have useful relationship with our output

https://www.youtube.com/watch?v=2I_AYIECCOQ

Multivariate regression model

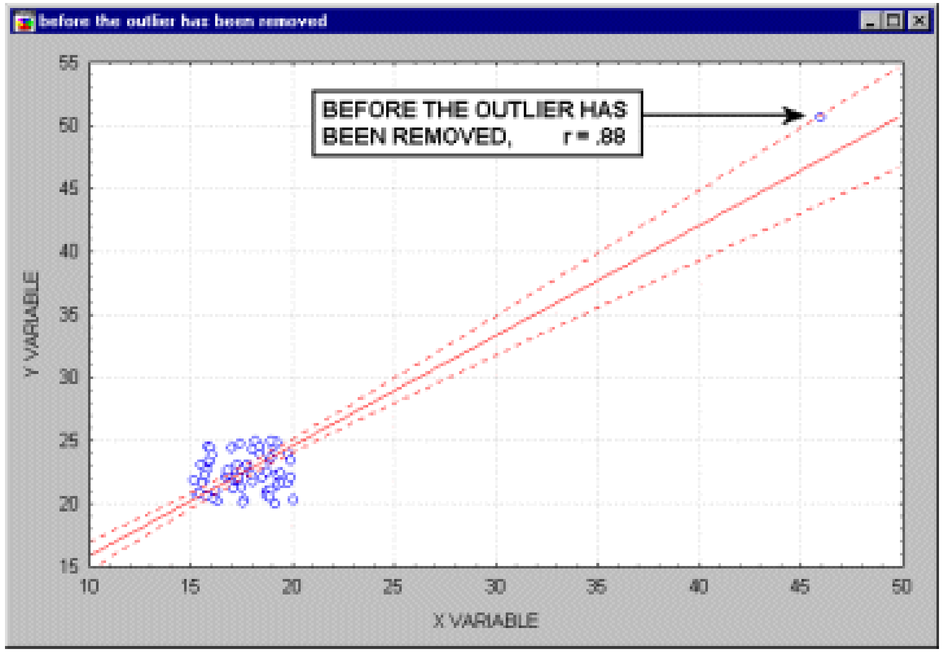
Can we reduce input variables further?

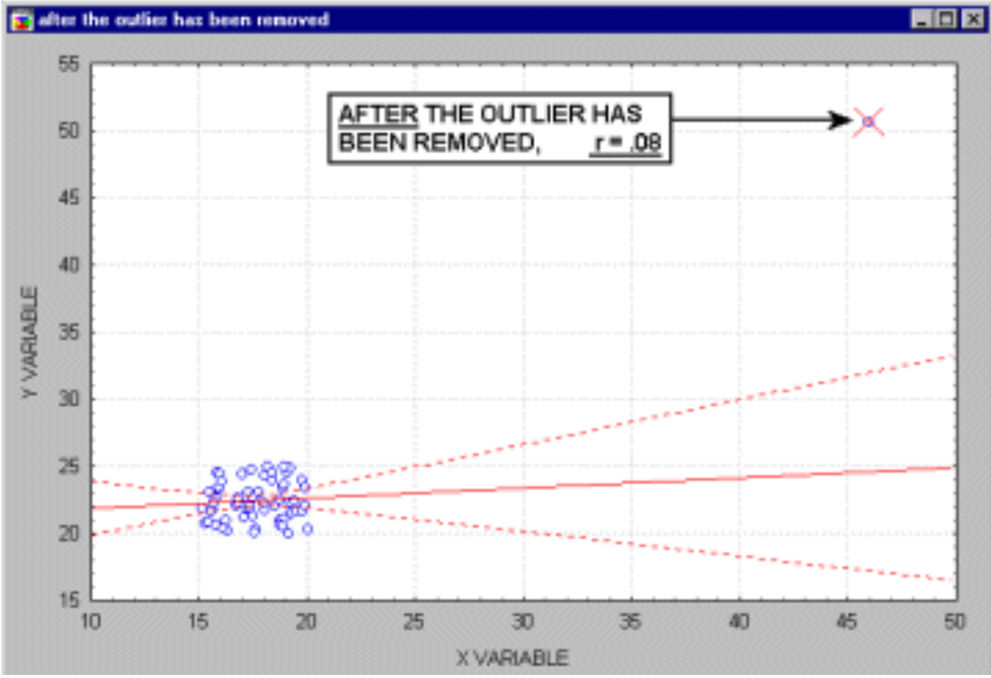
- **Any Dimensional Reduction Technique can be applied?**
  - **With carefully evaluation**
    - **# necessary components**

- EX. PCA / LSA / AutoEncoder

Multivariate regression model

What would regression be before and after outlier removal?





http://www.statsoft.com/textbook/multiple-regression

Multivariate regression parameter estimation

$$h_\theta(x) = \begin{bmatrix} \theta_0 & \theta_1 & \dots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T x$$

- **G**radient **D**escent Estimation
  - More preferable
  - Could be trapped in
    - Local optimum

repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \qquad \text{for j := 0...n}$$

}

If α is too small: slow convergence.
If α is too large: may not decrease on every iteration and thus may not converge.

Multivariate regression parameter estimation

$$h_\theta(x) = \begin{bmatrix} \theta_0 & \theta_1 & \dots & \theta_n \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix} = \theta^T x$$

repeat until convergence: {

$$\theta_0 := \theta_0 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_0^{(i)}$$

$$\theta_1 := \theta_1 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_1^{(i)}$$

$$\theta_2 := \theta_2 - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_2^{(i)}$$

...

}

Gradient Descent Estimation

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

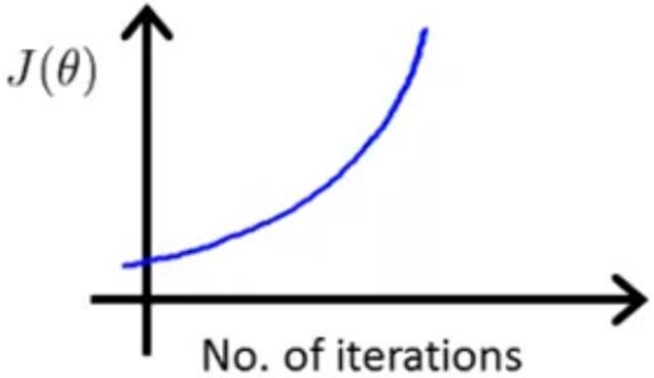
repeat until convergence: {

$$\theta_j := \theta_j - \alpha \frac{1}{m} \sum_{i=1}^{m} (h_\theta(x^{(i)}) - y^{(i)}) \cdot x_j^{(i)} \qquad \text{for j := 0...n}$$

}

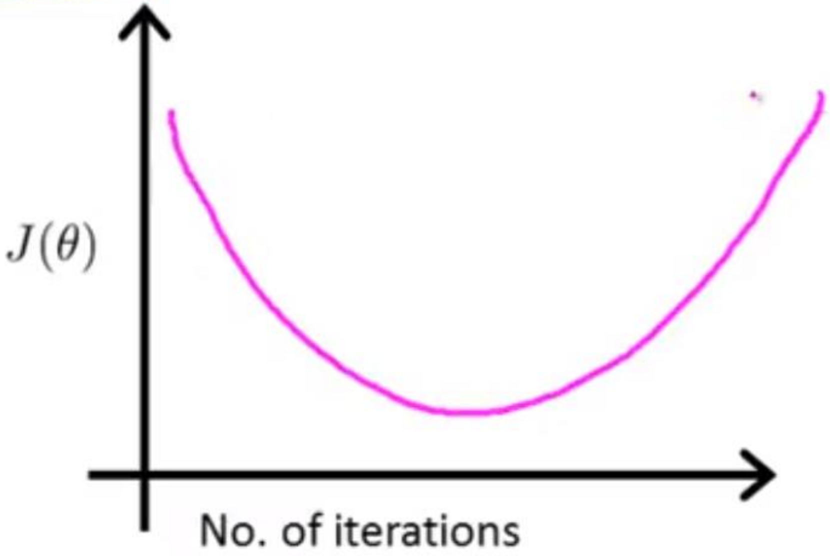
Gradient descent estimation

Making sure gradient descent is working correctly.



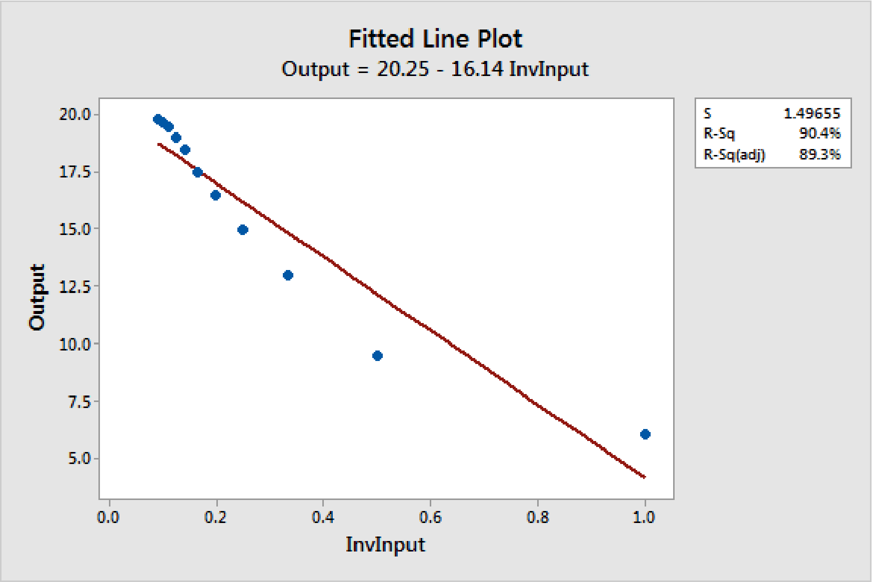
Gradient descent not working.

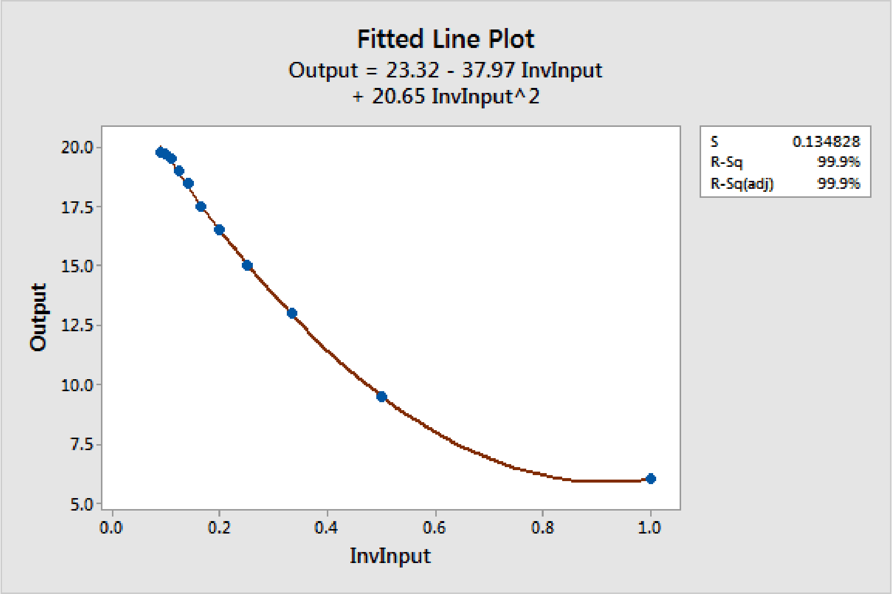
Use smaller α.



Will linear regression fit for all data?

Nonlinear regression model
using curve fitting





- https://blog.minitab.com/blog/adventures-in-statistics-2/curve-fitting-with-linear-and-nonlinear-regression