Math 189Z

Lecture 1: Overview and Linear Regression

 https://math189covid19.github.io/

Overview

- Course description
- Syllabus
- Schedule
- Term project
- Homework
- Course Resources

- https://math189covid19.github.io/

COVID-19: Data Analytics and Machine Learning

PROF. WEIQING GU

SPRING 2020

COURSE DESCRIPTION

This is a special topics course responding to the coronavirus pandemic. We will employ big data analytics and machine learning (ML) techniques to process, identify key data features, infer, predict, integrate, classify, and extract unique insights from the COVID-19 Open Research Dataset. This open dataset brings together nearly 30,000 scientific articles about the virus known as SARS-CoV-2 as well as related viruses in the broader coronavirus group, and it contains the most extensive collection of machine readable coronavirus literature to date. Math189Z is a project-based online course using the materials selected from this dataset. Some of the project goals include helping the science community to understand data genetics, incubation, and symptoms or helping fill some gaps when scientists are pursuing knowledge around prevention, treatment and a vaccine. Additionally, another goal of this course is to become comfortable using GitHub as this tool is extremely prevalent in industry and academia when developing and deploying models. To that end, all code, reading summaries, and your final project will be hosted on GitHub. Background in calculus and/or linear algebra required. HMC students may add without a PERM. Off-campus students should submit a PERM, including a description of their math coursework completed or underway.

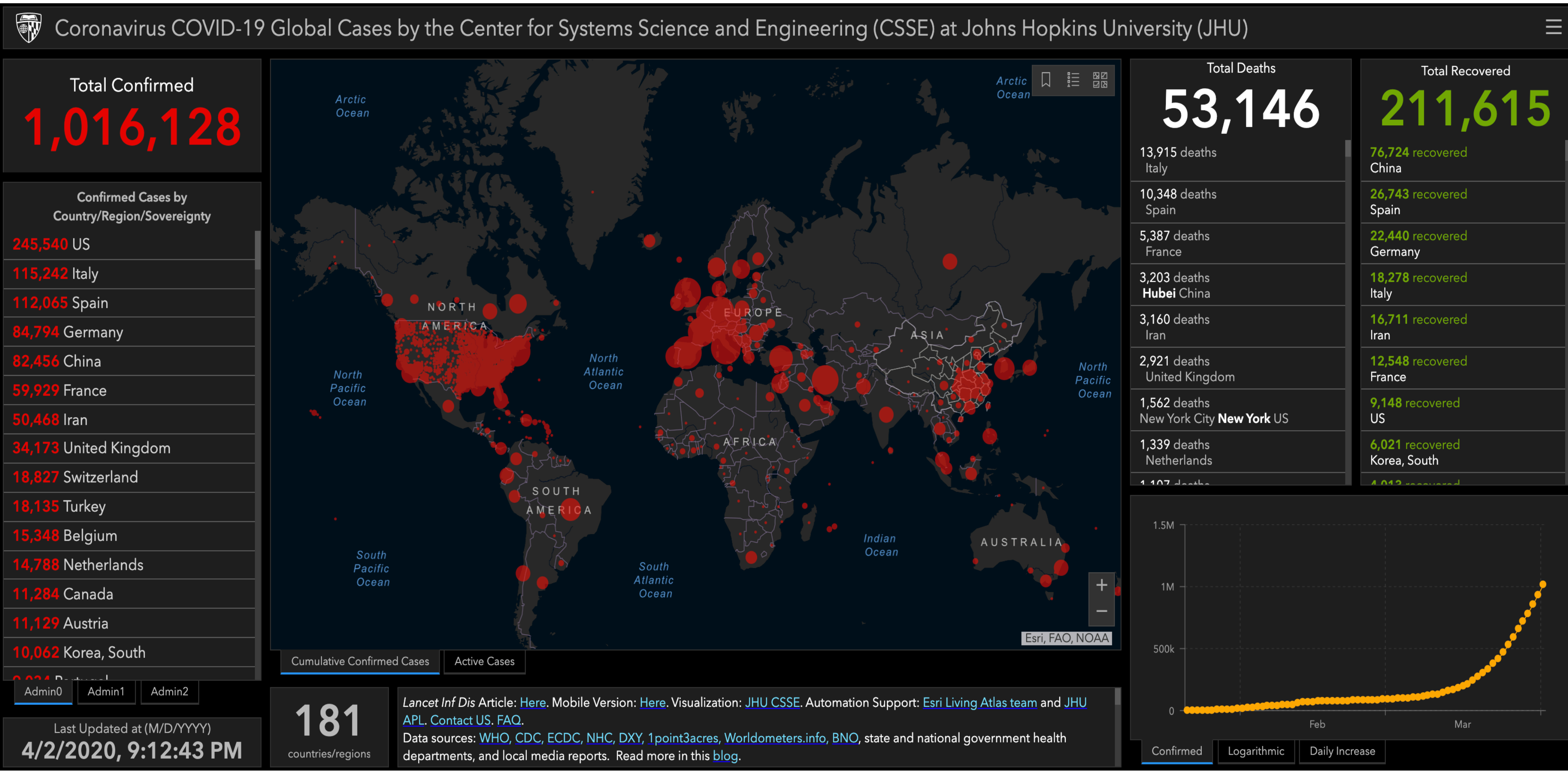
You may find your homework assignments on the link below

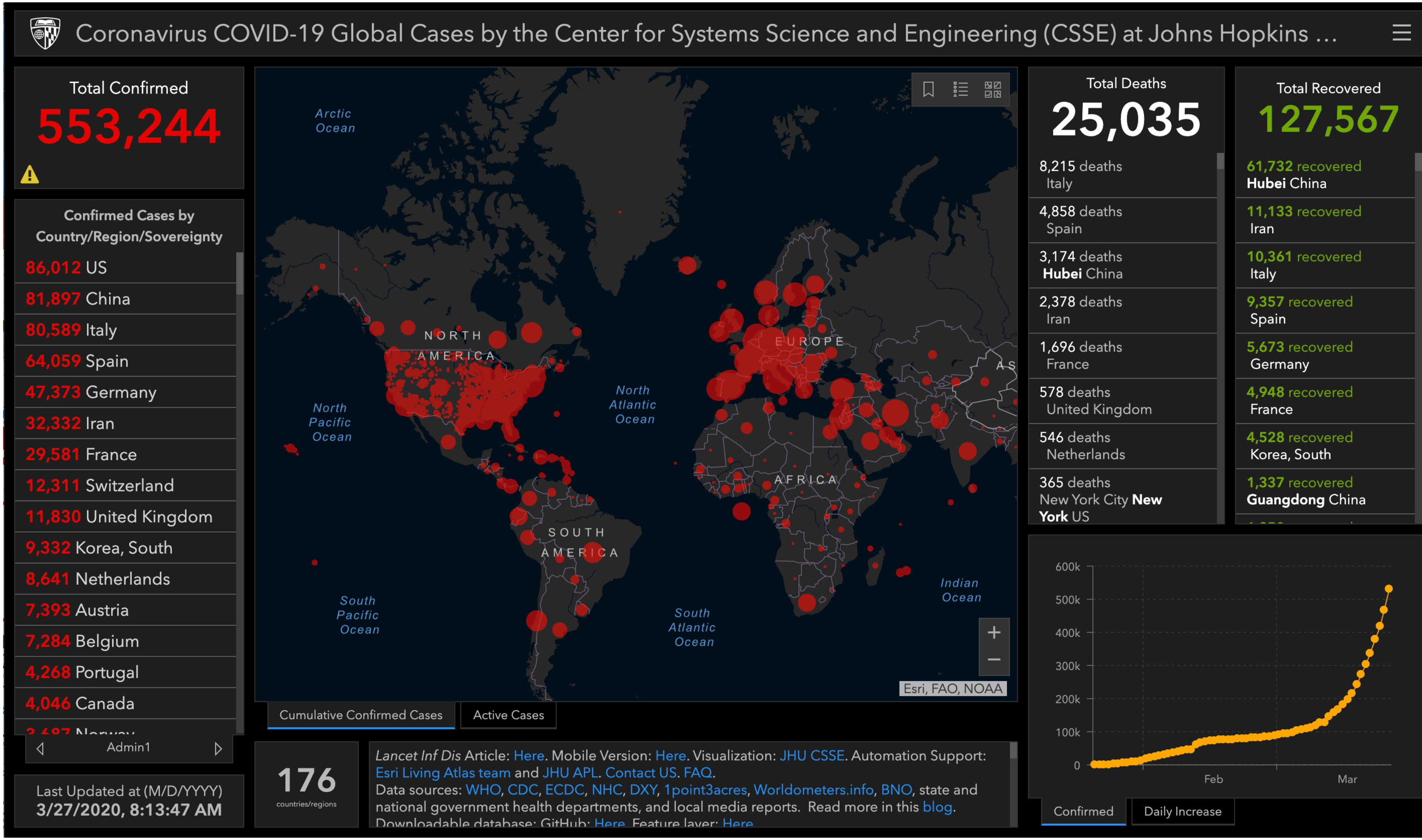
- https://math189covid19.github.io/resources.html

COVID-19 Spread Status

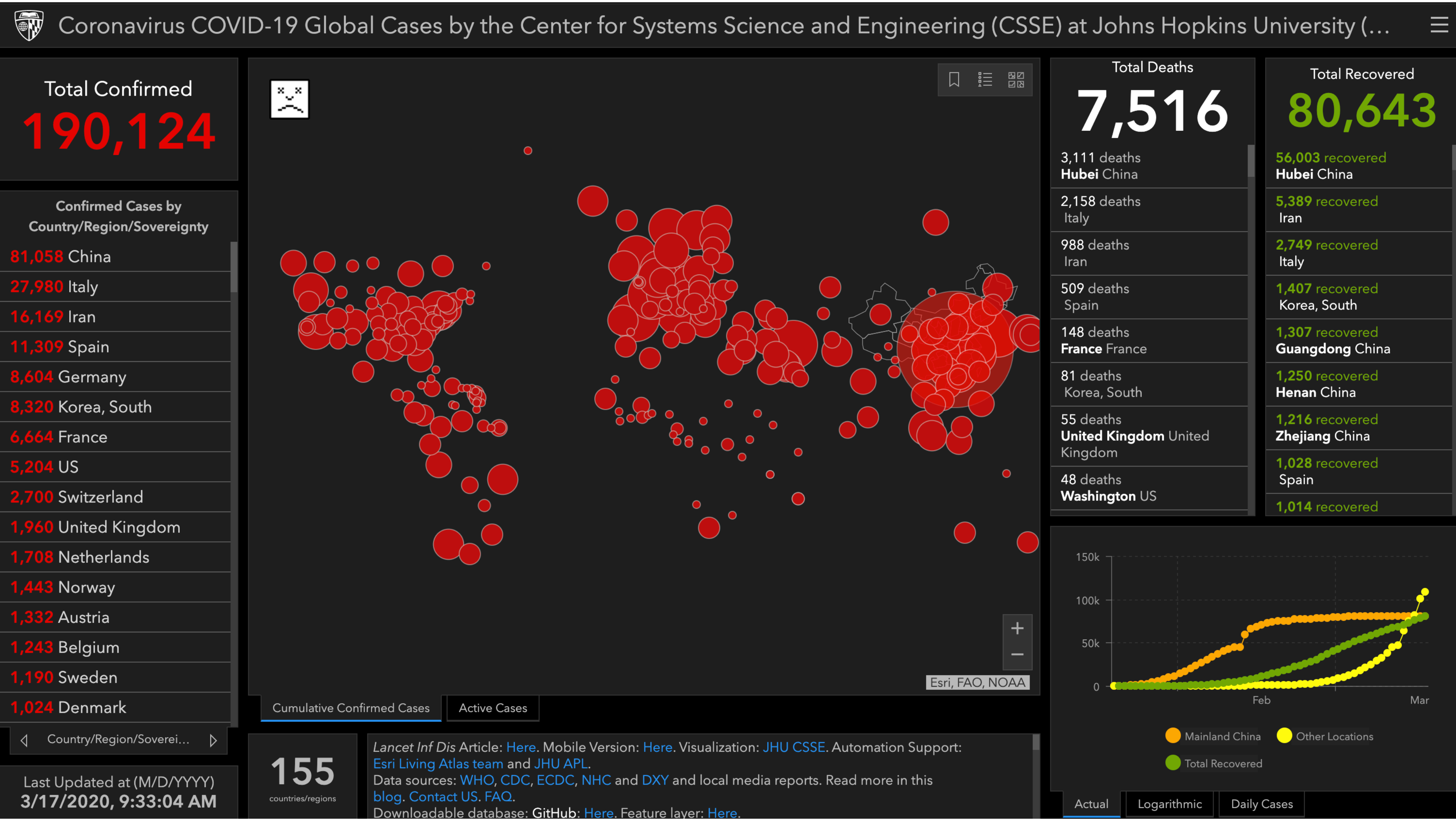
- COVID-19 confirmed cases have been increased since our last meeting

- It is an exponential spread now
- How to mathematically quantify the spread?


Coronavirus COVID-19 Global Cases by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (JHU)
Total Confirmed
1,016,128
Confirmed Cases by Country/Region/Sovereignty
245,540 US
115,242 Italy
112,065 Spain
84,794 Germany
82,456 China
59,929 France
50,468 Iran
34,173 United Kingdom
18,827 Switzerland
18,135 Turkey
15,348 Belgium
14,788 Netherlands
11,284 Canada
11,129 Austria
10,062 Korea, South
Admin0
Admin1
Admin2
Last Updated at (M/D/YYYY)
4/2/2020, 9:12:43 PM
Arctic Ocean
NORTH AMERICA
EUROPE
ASIA
AFRICA
SOUTH AMERICA
AUSTRALIA
North Pacific Ocean
North Atlantic Ocean
South Pacific Ocean
South Atlantic Ocean
Indian Ocean
Esri, FAO, NOAA
Cumulative Confirmed Cases
Active Cases
181
countries/regions
Lancet Inf Dis Article: Here. Mobile Version: Here. Visualization: JHU CSSE. Automation Support: Esri Living Atlas team and JHU APL. Contact US. FAQ.
Data sources: WHO, CDC, ECDC, NHC, DXY, 1point3acres, Worldometers.info, BNO, state and national government health departments, and local media reports. Read more in this blog.
Total Deaths
53,146
13,915 deaths Italy
10,348 deaths Spain
5,387 deaths France
3,203 deaths Hubei China
3,160 deaths Iran
2,921 deaths United Kingdom
1,562 deaths New York City New York US
1,339 deaths Netherlands
Total Recovered
211,615
76,724 recovered China
26,743 recovered Spain
22,440 recovered Germany
18,278 recovered Italy
16,711 recovered Iran
12,548 recovered France
9,148 recovered US
6,021 recovered Korea, South
1.5M
1M
500k
0
Feb
Mar
Confirmed
Logarithmic
Daily Increase



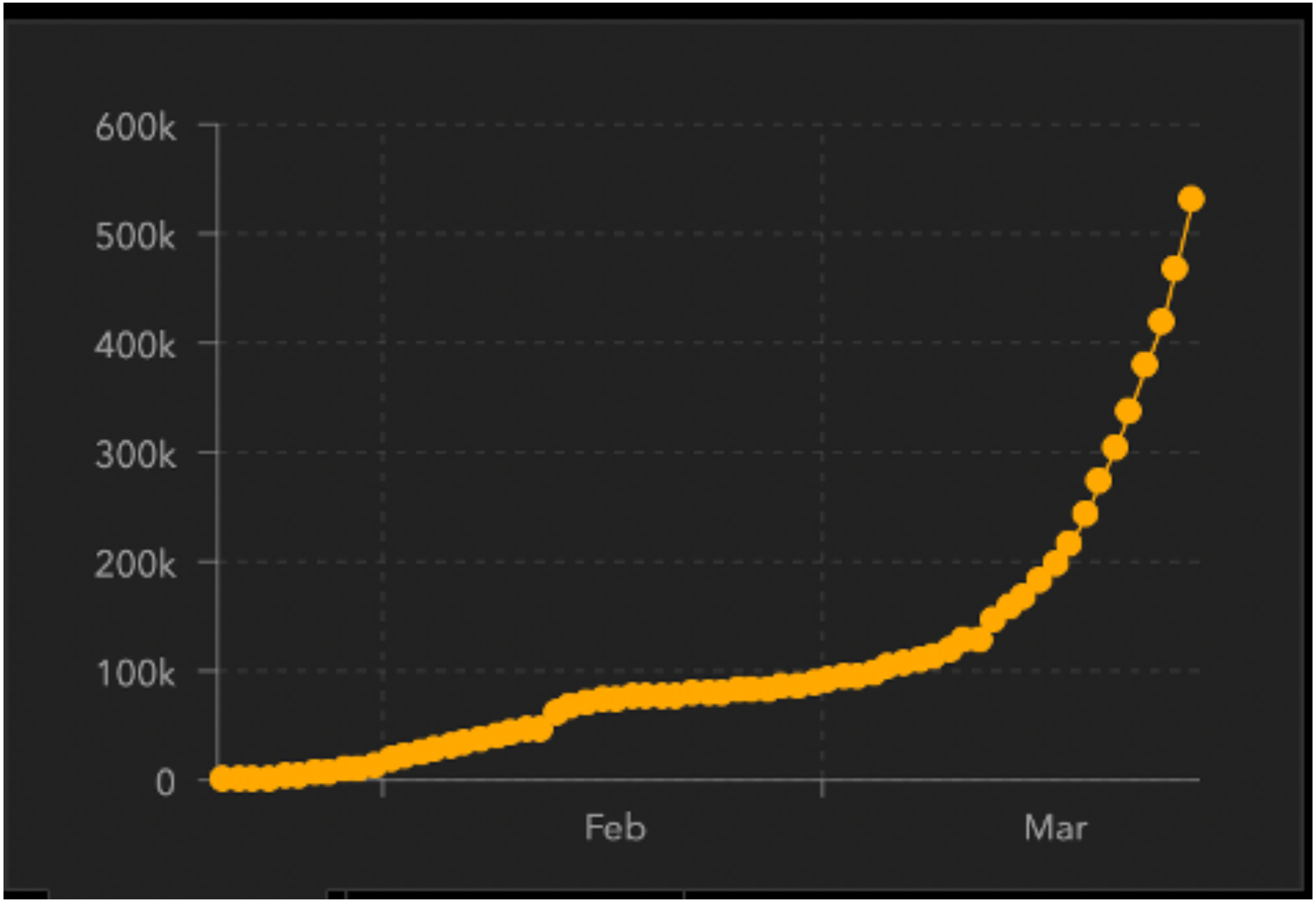
Coronavirus COVID-19 Global Cases by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins ...
Total Confirmed
553,244
Confirmed Cases by Country/Region/Sovereignty
86,012 US
81,897 China
80,589 Italy
64,059 Spain
47,373 Germany
32,332 Iran
29,581 France
12,311 Switzerland
11,830 United Kingdom
9,332 Korea, South
8,641 Netherlands
7,393 Austria
7,284 Belgium
4,268 Portugal
4,046 Canada
Admin1
Last Updated at (M/D/YYYY)
3/27/2020, 8:13:47 AM
Arctic Ocean
NORTH AMERICA
North Pacific Ocean
North Atlantic Ocean
EUROPE
AFRICA
SOUTH AMERICA
South Pacific Ocean
South Atlantic Ocean
Indian Ocean
Esri, FAO, NOAA
Cumulative Confirmed Cases
Active Cases
176
countries/regions
Lancet Inf Dis Article: Here. Mobile Version: Here. Visualization: JHU CSSE. Automation Support: Esri Living Atlas team and JHU APL. Contact US. FAQ.
Data sources: WHO, CDC, ECDC, NHC, DXY, 1point3acres, Worldometers.info, BNO, state and national government health departments, and local media reports. Read more in this blog.
Total Deaths
25,035
8,215 deaths Italy
4,858 deaths Spain
3,174 deaths Hubei China
2,378 deaths Iran
1,696 deaths France
578 deaths United Kingdom
546 deaths Netherlands
365 deaths New York City New York US
Total Recovered
127,567
61,732 recovered Hubei China
11,133 recovered Iran
10,361 recovered Italy
9,357 recovered Spain
5,673 recovered Germany
4,948 recovered France
4,528 recovered Korea, South
1,337 recovered Guangdong China
600k
500k
400k
300k
200k
100k
0
Feb
Mar
Confirmed
Daily Increase


Coronavirus COVID-19 Global Cases by the Center for Systems Science and Engineering (CSSE) at Johns Hopkins University (…
Total Confirmed
190,124
Confirmed Cases by Country/Region/Sovereignty
81,058 China
27,980 Italy
16,169 Iran
11,309 Spain
8,604 Germany
8,320 Korea, South
6,664 France
5,204 US
2,700 Switzerland
1,960 United Kingdom
1,708 Netherlands
1,443 Norway
1,332 Austria
1,243 Belgium
1,190 Sweden
1,024 Denmark
Country/Region/Soverei…
Last Updated at (M/D/YYYY)
3/17/2020, 9:33:04 AM
Esri, FAO, NOAA
Cumulative Confirmed Cases
Active Cases
155
countries/regions
Lancet Inf Dis Article: Here. Mobile Version: Here. Visualization: JHU CSSE. Automation Support: Esri Living Atlas team and JHU APL.
Data sources: WHO, CDC, ECDC, NHC and DXY and local media reports. Read more in this blog. Contact US. FAQ.
Downloadable database: GitHub: Here. Feature layer: Here.
Total Deaths
7,516
3,111 deaths Hubei China
2,158 deaths Italy
988 deaths Iran
509 deaths Spain
148 deaths France France
81 deaths Korea, South
55 deaths United Kingdom United Kingdom
48 deaths Washington US
Total Recovered
80,643
56,003 recovered Hubei China
5,389 recovered Iran
2,749 recovered Italy
1,407 recovered Korea, South
1,307 recovered Guangdong China
1,250 recovered Henan China
1,216 recovered Zhejiang China
1,028 recovered Spain
1,014 recovered
150k
100k
50k
0
Feb
Mar
Mainland China
Other Locations
Total Recovered
Actual
Logarithmic
Daily Cases

What do we mean by exponential spread?

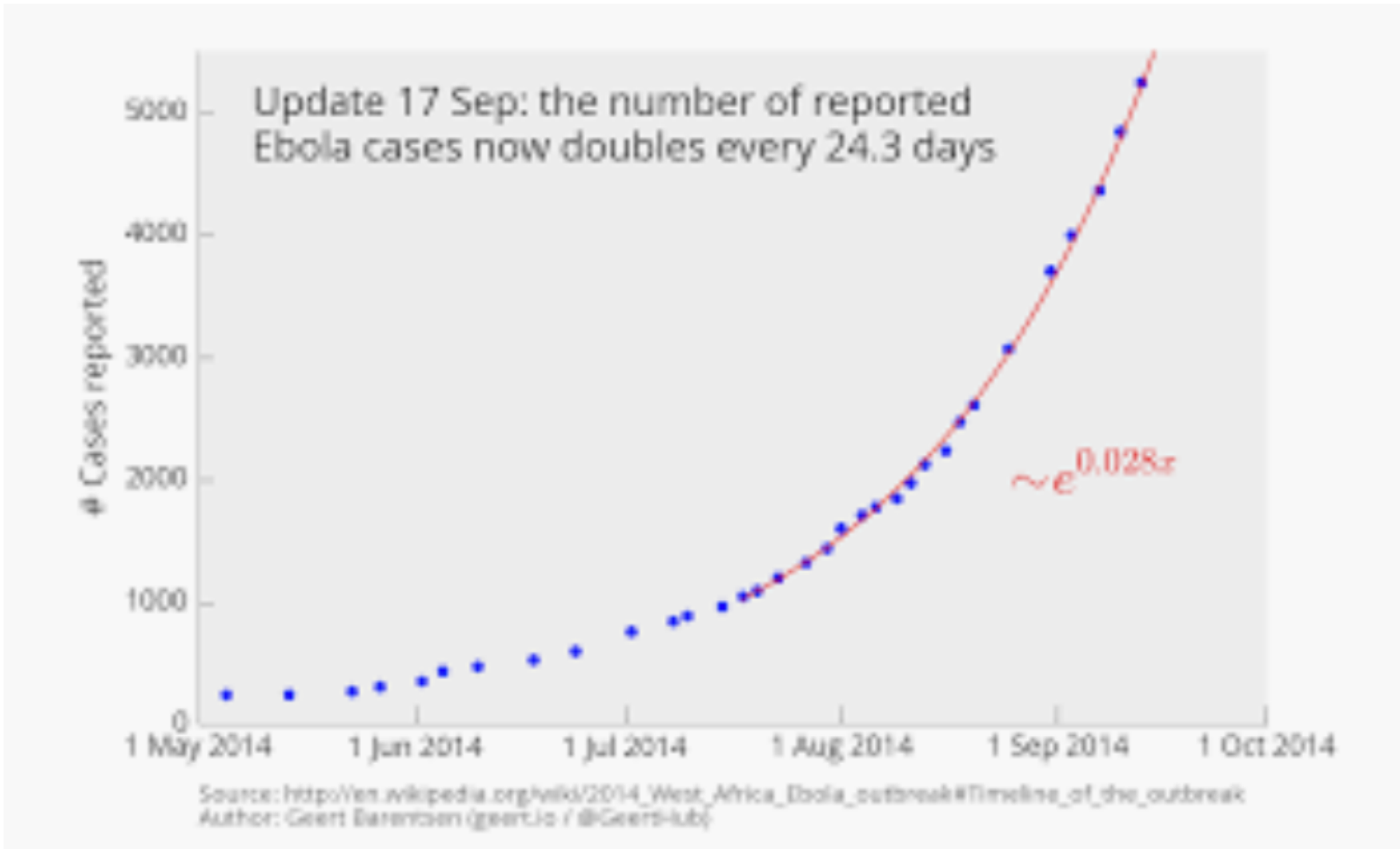
- Today's number new infected
- Yesterday's number new infected
- Let r = Today's number new infected/Yesterday's number new infected
- Now if r>1, then it is exponential

Notice: Each piece is different in increasing



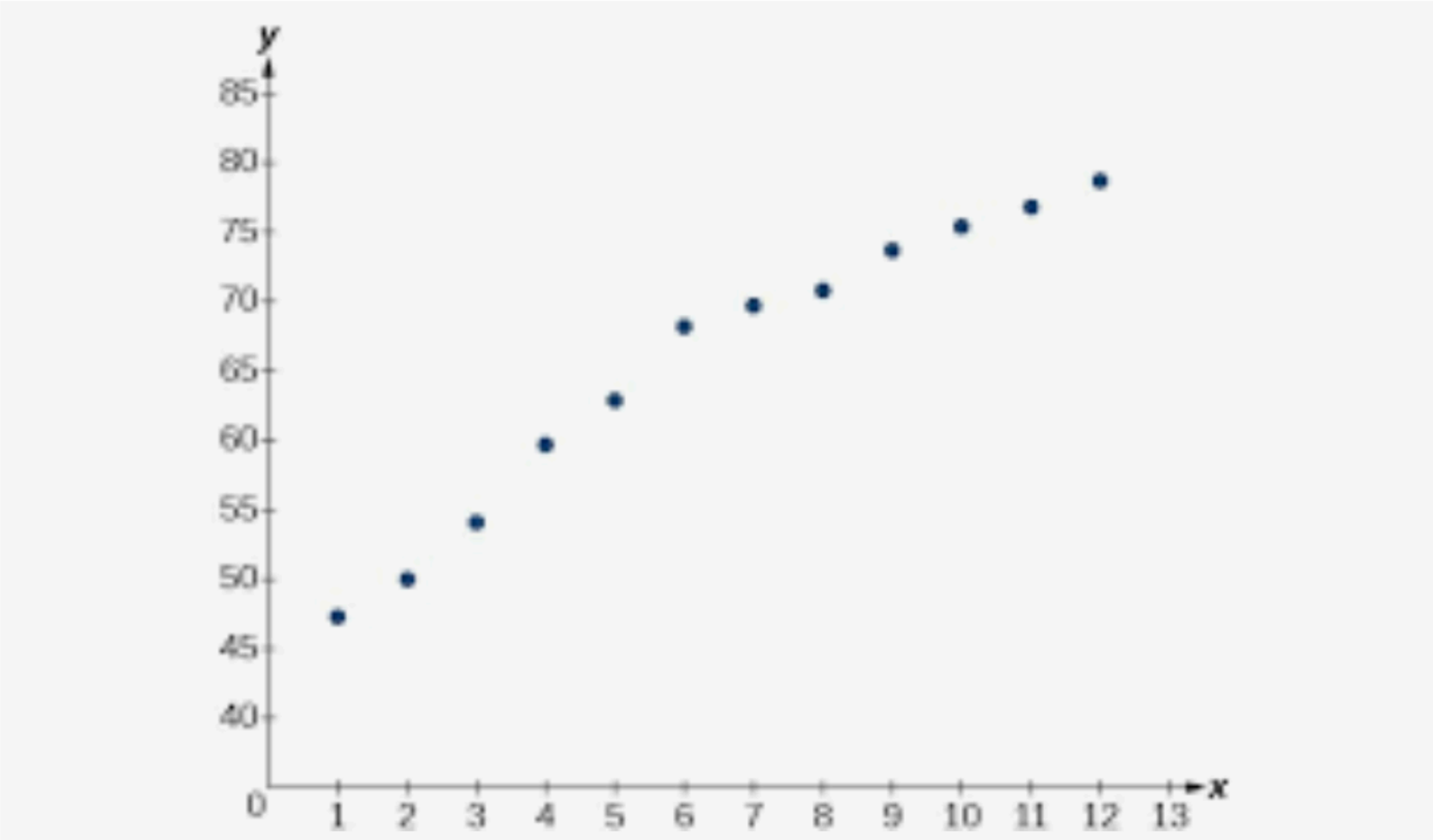
Our goal: Mathematically quantify the difference and let ML auto learn it

- Example



The exponential spread of the Ebola virus

Trick:



Build a logarithmic model from data ...

Machine Learning & Big Data Analytics will be covered in this course

This course will cover several major approaches in ML/Data Analytics

- *Regression*
- *MM and HMM*
- *Neural Network (e.g. RNN)*
- *Other approaches: most needed in your term projects*

- *For more systematic Machine Learning and Big Data Analytics methods, I will cover them this summer after the core summer course in*
- *Math 189L: Mathematics of Big Data, I.*

Today's Lecture

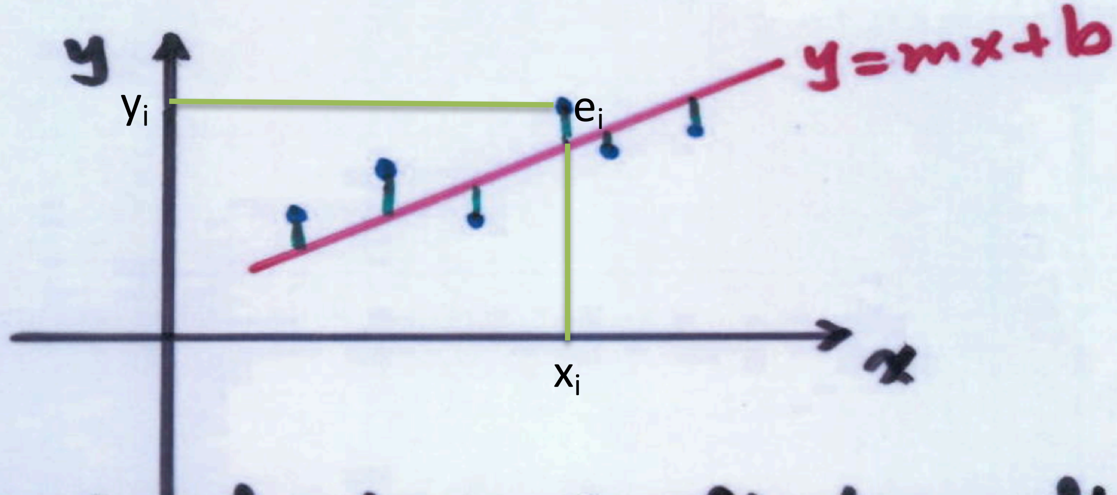
- **Frist: Overview COVID-19 Spread Status**
- **Second: Use linear regression as an example to analyze big data including COVID-19 data.**

Note: Linear regress techniques could be generalized to

- *Polynomial Regression*
- *Piecewise Linear Regression*
- *Other type of regression including transform data first, then use linear regression and then transform them back.*

1. Statistical Calculus Approach
(Classical Least Square Approximation)

Spse we have data pts (x_i, y_i) and want to find the line $y = mx + b$ which best describes the data.



The problem boils down to find m & b.

The error between one point and the line is

$$e_i = y_i - (mx_i + b)$$

Our objective is minimizing the total error.

- However, the errors e_i, some could be positive and some could be negative. A simple sum of the errors would not work well.
- Can you think about an example why not working well?
- How to fix this problem?
- Instead we consider the following **objective or cost function**:

L_2 *norm*

- $\boxed{J(m,b) = \sum (e_i)^2 = \sum (y_i - mx_i - b)^2}$
- Can we use $\sum |e_i|$ instead?

L_1 *norm*

Goal: Find m and b to minimize the cost function J

- How?
- Set all partials equal to zero!
- Work out the details with the students on the board.

Obtained solution using Cramer's rule

- Give a linear system: $$\begin{cases} a_1 x + b_1 y & = c_1 \\ a_2 x + b_2 y & = c_2 \end{cases}$$
- Write it into matrix form: $$\begin{bmatrix} a_1 & b_1 \\ a_2 & b_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} c_1 \\ c_2 \end{bmatrix}$$

Assume the coefficient matrix is invertible, i.e. the det = $a_1b_2 - b_1a_2$ is nonzero. Then

$$x = \frac{\begin{vmatrix} c_1 & b_1 \\ c_2 & b_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}} = \frac{c_1 b_2 - b_1 c_2}{a_1 b_2 - b_1 a_2}, \quad y = \frac{\begin{vmatrix} a_1 & c_1 \\ a_2 & c_2 \end{vmatrix}}{\begin{vmatrix} a_1 & b_1 \\ a_2 & b_2 \end{vmatrix}} = \frac{a_1 c_2 - c_1 a_2}{a_1 b_2 - b_1 a_2}.$$

Close formula for Least Square Approximation

Using Cramer's rule, we get solution for m, b:

$$m = \frac{n \sum_{i=1}^{n} x_i y_i - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} y_i\right)}{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

$$b = \frac{\left(\sum_{i=1}^{n} x_i^2\right)\left(\sum_{i=1}^{n} y_i\right) - \left(\sum_{i=1}^{n} x_i\right)\left(\sum_{i=1}^{n} x_i y_i\right)}{n \sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}$$

But the formula is massy. Next we'll find a compact form of this formula.

Homework problem

- Given 4 points as below:

(0, 1), (2, 3), (3, 6), (4, 8)

a) Find y = mx + b based on Cramer's rule.

- Hint:

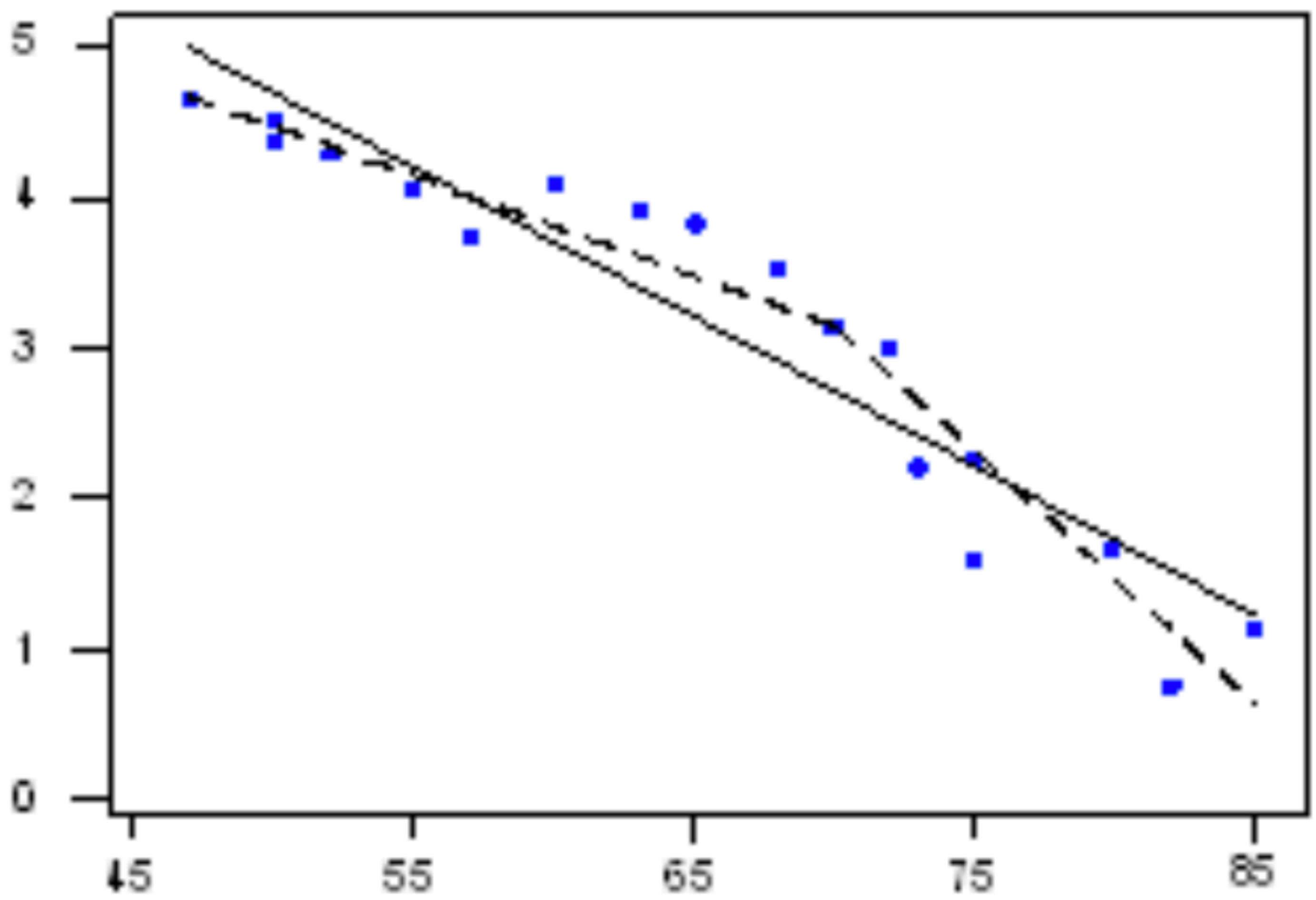
x_i	y_i	x_i^2	$x_i y_i$
0	1	0	0
2	3	4	6
3	6	9	18
4	8	16	32
$\Sigma x_i = 9$	$\Sigma y_i = 18$	$\Sigma x_i^2 = 29$	$\Sigma x_i y_i = 56$

- b) Use the normal formula to find the solution and compare it with that of a).
- c) Plot the data points, and draw y = mx +b.
- d) (All by coding) Find another 100 points near the line y = mx + b. Then find the least square approxim'n again & plot both the data points & the new line.

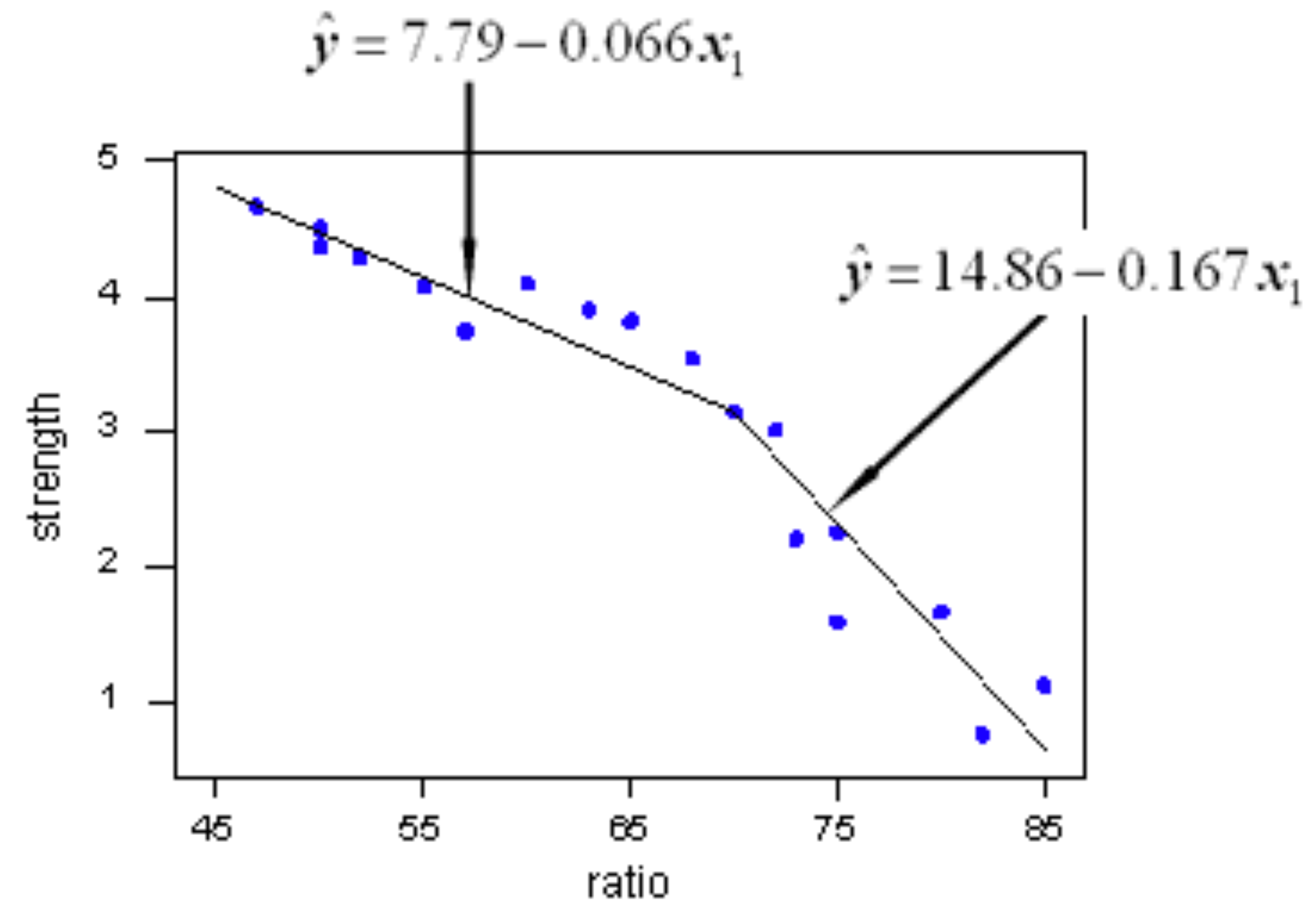
Piecewise linear regression

- **Piecewise linear regression** is a form of **regression** that allows multiple **linear** models to be fitted to the data for different ranges of X.
- The **regression** function at the breakpoint may be discontinuous, but it is possible to specify the model such that the model is continuous at all points.

Intuition: Piecewise linear regression

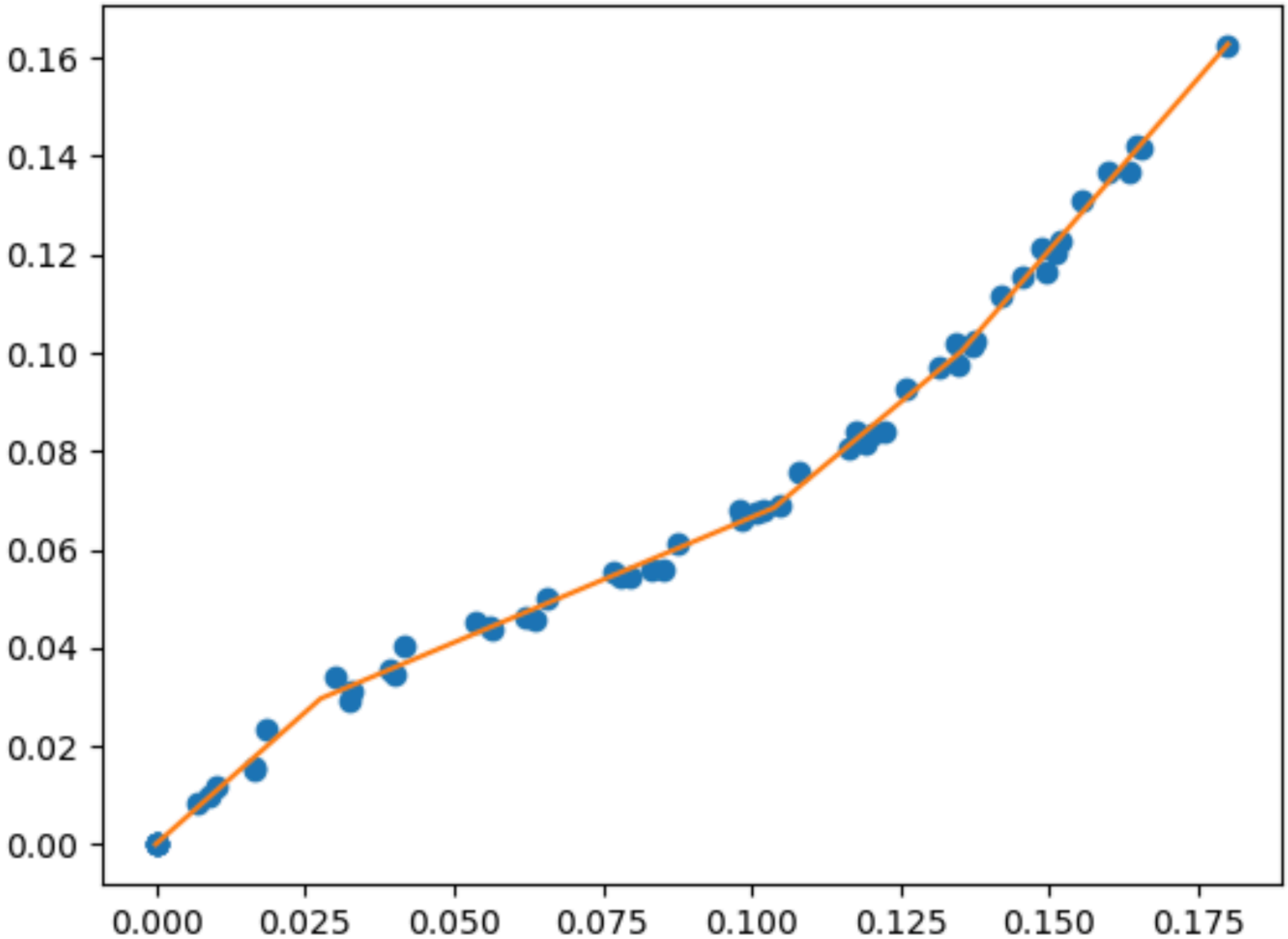


Example1



$\hat{y} = 7.79 - 0.066x_1$
$\hat{y} = 14.86 - 0.167x_1$
5
4
3
2
1
strength
45
55
65
75
85
ratio

Example 2



0.16
0.14
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.000
0.025
0.050
0.075
0.100
0.125
0.150
0.175

Machine Learning: Polynomial Regression

- First do a data visualization

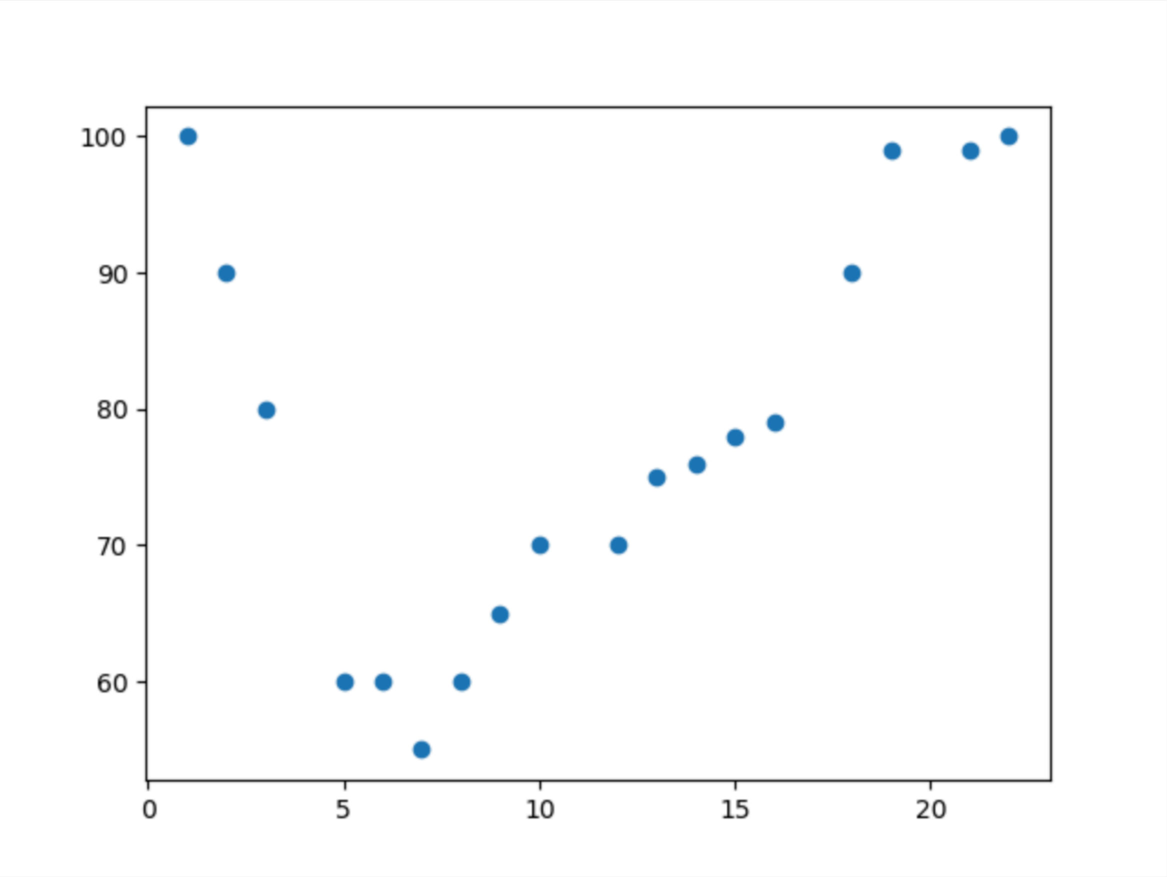
Example

Start by drawing a scatter plot:

```
import matplotlib.pyplot as plt

x = [1,2,3,5,6,7,8,9,10,12,13,14,15,16,18,19,21,22]
y = [100,90,80,60,60,55,60,65,70,70,75,76,78,79,90,99,99,100]

plt.scatter(x, y)
plt.show()
```

The data is nonlinear.
We can use polynomial regression.

Result:



Note: You always can use piecewise linear regression.

Decide a degree k of the polynomial

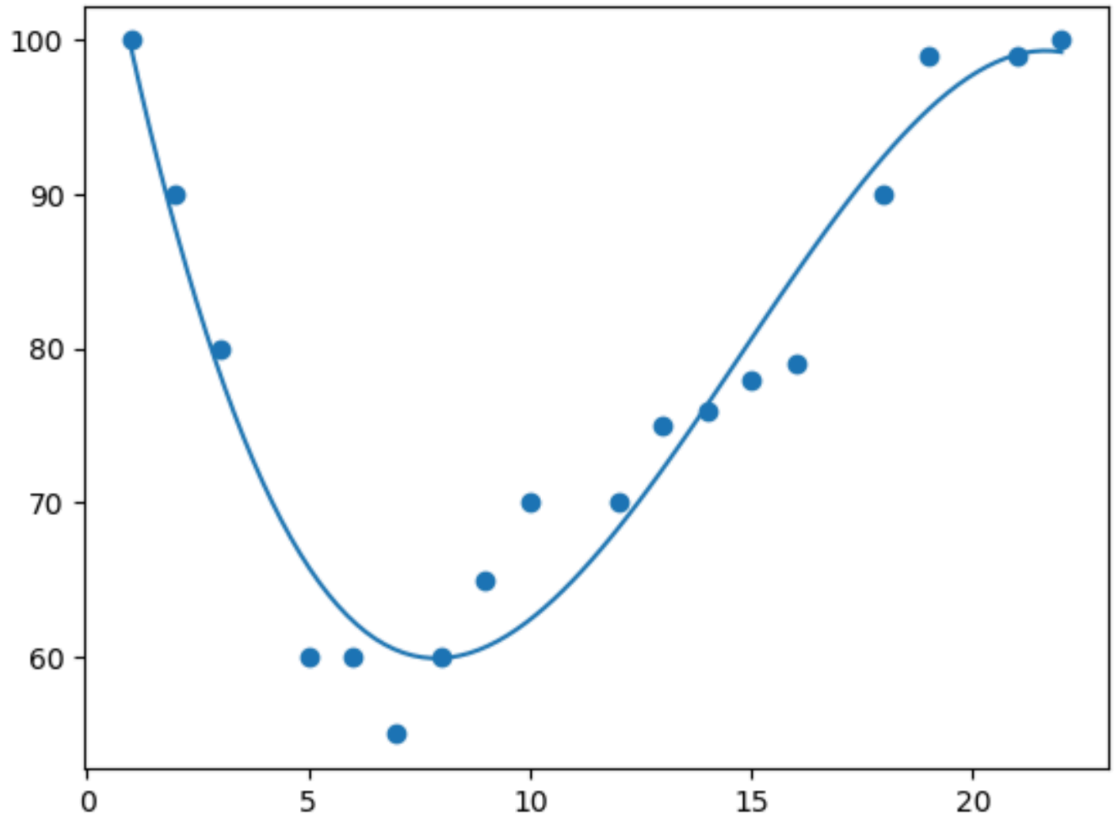
- Here k =3

Import numpy and matplotlib then draw the line of Polynomial Regression:

```
import numpy
import matplotlib.pyplot as plt

x = [1,2,3,5,6,7,8,9,10,12,13,14,15,16,18,19,21,22]
y = [100,90,80,60,60,55,60,65,70,70,75,76,78,79,90,99,99,100]

mymodel = numpy.poly1d(numpy.polyfit(x, y, 3))

myline = numpy.linspace(1, 22, 100)

plt.scatter(x, y)
plt.plot(myline, mymodel(myline))
plt.show()
```

Machine Learning: Polynomial Regression

- First do a data visualization

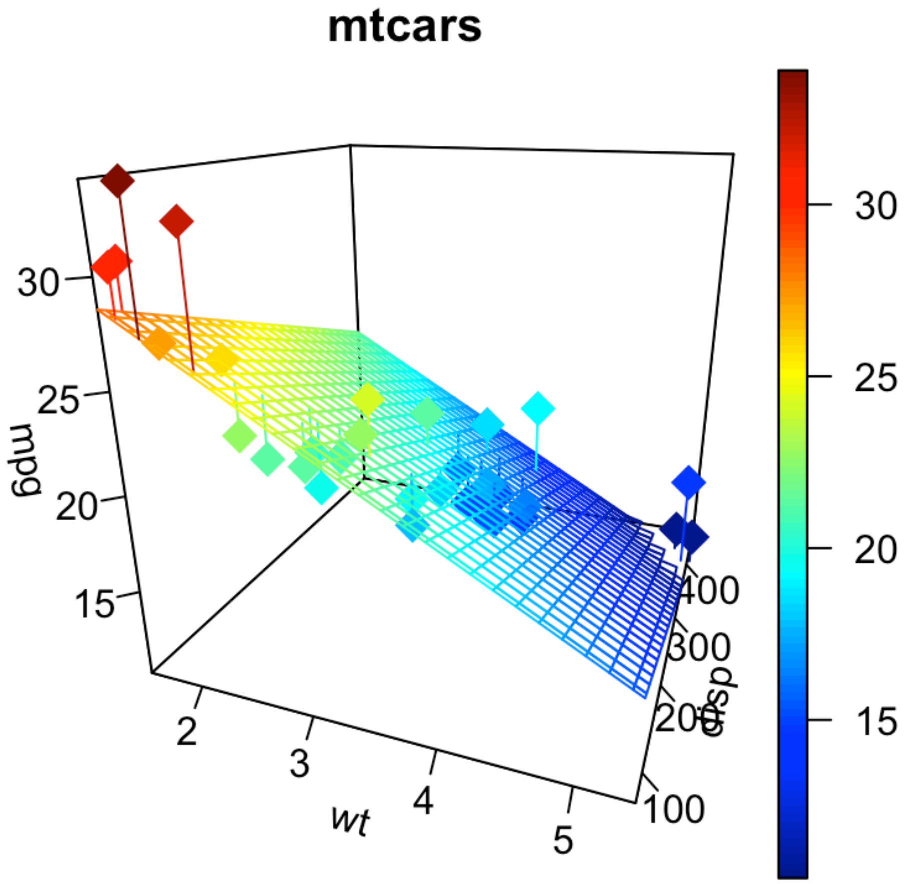


Note: Such a polynomial piece of degree 3 is called a cubic spline.

What had happened behind this code mathematically?

- Work out details with students on iPad.

How about fit data by a plane or even higher dimensions?



Get the same close solution by normal equation!

- Can you imagine what other cases you would get the same kind of solution?

Normal Equation for Least Square Approximation

- i.e. Representing the Least Square Solution in Matrix Form
- Work out the details with the students on the board.
- Recall the product rule:
- f, g: R→R: $(f \cdot g)' = f' \cdot g + f \cdot g'$
- f, g: R^n→R: $\nabla(f \cdot g) = \nabla f \cdot g + f \cdot \nabla g$
- **f**, **g**: R^n→R^n: $(\mathbf{f} \cdot \mathbf{g})' = \mathbf{f}\,' \cdot \mathbf{g} + \mathbf{f} \cdot \mathbf{g}\,'$

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

Taking Partial Derivatives for different Types of functions

Type 1: $\mathbb{R} \to \mathbb{R}$ (one-to-one)
$x \mapsto f(x)$

$$\frac{\partial f}{\partial x} = \frac{df}{dx}$$

☆☆ **Type 2**: $\mathbb{R}^n \to \mathbb{R}$ (Many-to-one)
$(x_1, x_2, \cdots, x_n) \mapsto f(x_1, \cdots, x_n)$

$$\left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, \cdots, \frac{\partial f}{\partial x_n}\right) \underset{\text{denoted}}{\triangleq} \nabla f$$

$$\nabla f(\vec{a}) = \left(\left.\frac{\partial f}{\partial x_1}\right|_{\vec{a}}, \left.\frac{\partial f}{\partial x_2}\right|_{\vec{a}}, \cdots, \left.\frac{\partial f}{\partial x_n}\right|_{\vec{a}}\right)$$

is called the gradient of f at $\vec{a}$.

Type 3: $\mathbb{R} \to \mathbb{R}^m$ (one-to-many)
$t \mapsto (f_1(t), \cdots, f_m(t)) \triangleq f(t)$

$$\begin{bmatrix} \frac{\partial f_1}{\partial t} \\ \vdots \\ \frac{\partial f_m}{\partial t} \end{bmatrix} = \begin{bmatrix} \frac{df_1}{dt} \\ \vdots \\ \frac{df_m}{dt} \end{bmatrix} \triangleq f'(t)$$

Key Technique:
Treat each component function as many-to-one function!

☆☆ **Type 4**: $\mathbb{R}^n \to \mathbb{R}^m$ (many-to-many)
$\underbrace{(x_1, \cdots, x_n)}_{\vec{x}} \mapsto (f_1(\vec{x}), \cdots, f_m(\vec{x}))$

$$Df(\underbrace{x_1, x_2, \ldots, x_n}_{\vec{x}}) = \underbrace{\begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} & \cdots & \frac{\partial f_2}{\partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial f_m}{\partial x_1} & \frac{\partial f_m}{\partial x_2} & \cdots & \frac{\partial f_m}{\partial x_n} \end{bmatrix}}_{\text{Derivative matrix}} \begin{array}{l} \leftarrow \nabla f_1(\vec{x}) \\ \leftarrow \nabla f_2(\vec{x}) \\ \\ \leftarrow \nabla f_m(\vec{x}) \end{array}$$

You must keep your mind clear what type of function your are dealing with!

Again we get the same solution!

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

Q: But what's wrong if we use Cramer's rule to solve it?

Or directly use the formula by finding the inverse $X^T X$?

Big Picture: Analytic Approaches Summarized

- **Use "linear regression" as an example to give an overview of big data analytics**

Modeling Approaches:

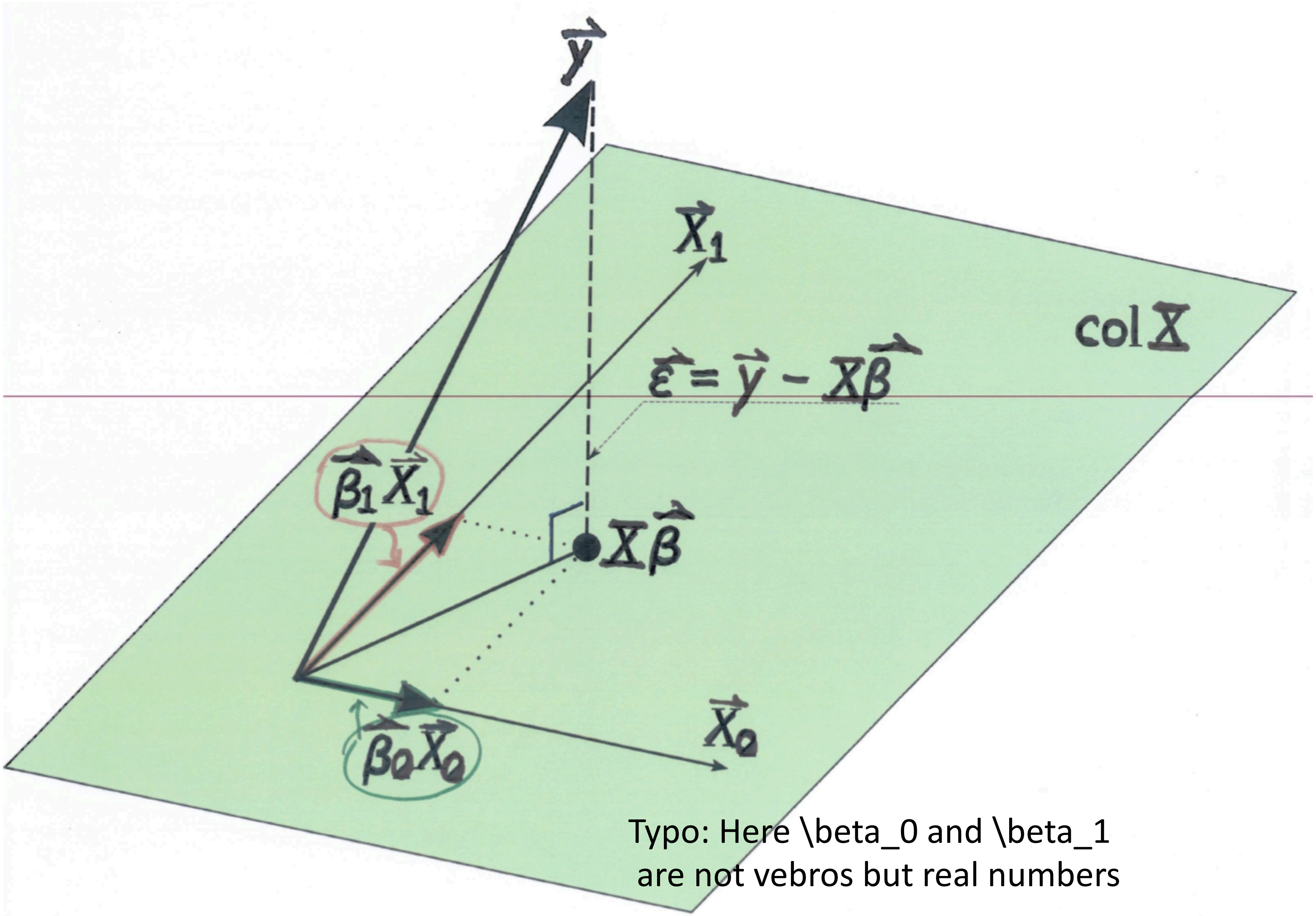
- *Statistical calculus*
- *Geometric analytic*
- *Probabilistic*

Each has its own merit

2. Geometric Analytic Approach (Geometric Least Square)

- Work out the details with the students on the board.

Key in *Geometric* Least Square Approximation



Typo: Here \beta_0 and \beta_1
are not vebros but real numbers

3. Probabilistic Approach (Maximal Likelihood)

- Work out the details with the students on iPad if time permits.