CPSC 340:
Machine Learning and Data Mining

Linear Regression

Admin

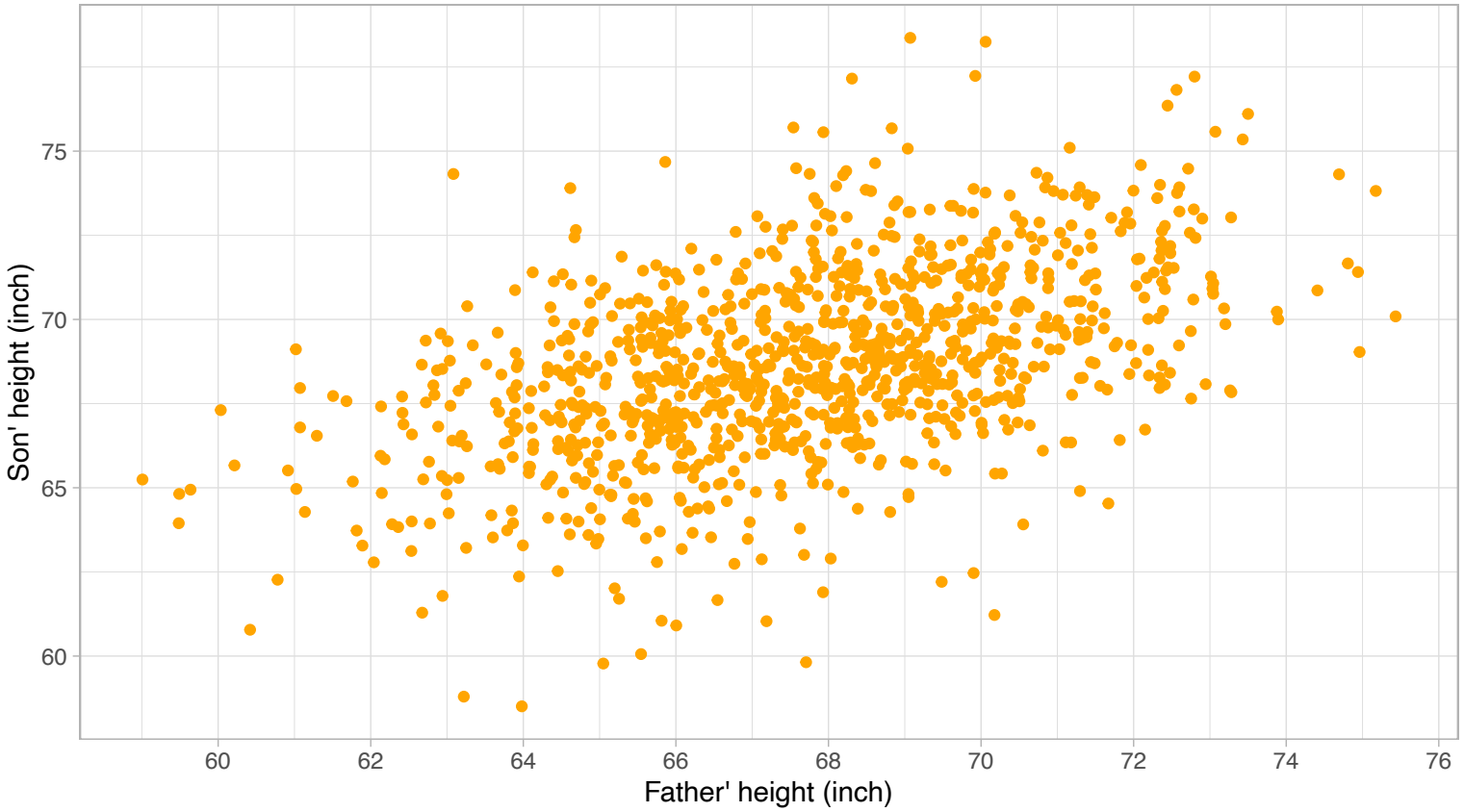
- Assignment 2 dues this Friday.
- On Monday, we just finished part2, outlier detection and clustering.
- We're going to start using calculus and linear algebra a lot.
  - You should start reviewing these ASAP if you are rusty.
  - A review of relevant calculus concepts is here.
  - A review of relevant linear algebra concepts is here.

Supervised Learning Round 2: Regression

- We're going to revisit supervised learning:

- Previously, we considered classification:
  - We assumed y_i was categorical: y_i = 'spam' or y_i = 'not spam'.
- Now we are going to consider regression:
  - We allow y_i to be numerical: y_i = 10.34cm.

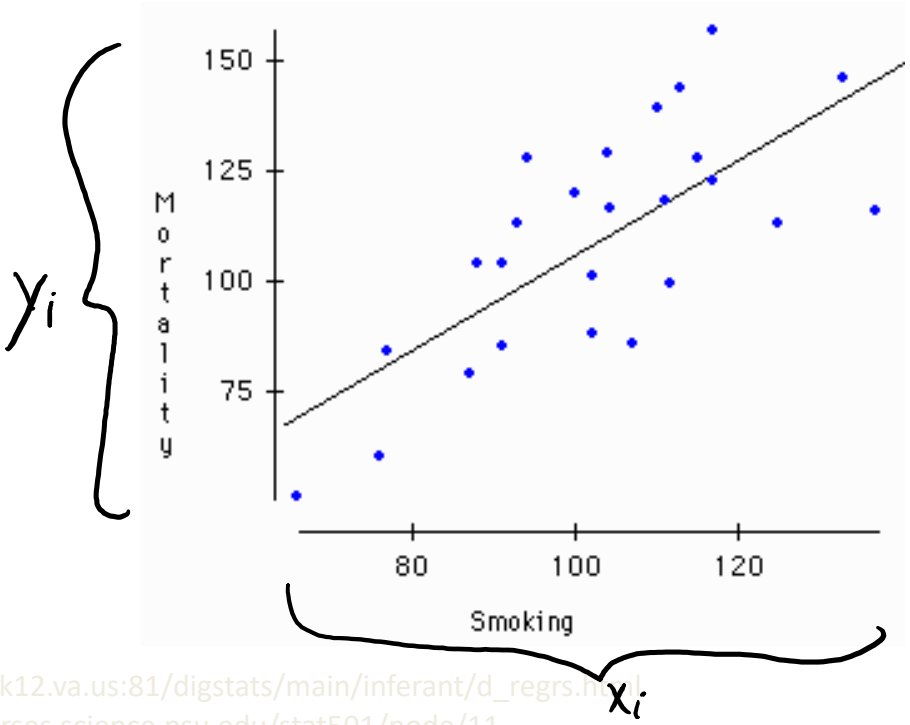
Regression Towards the Mean

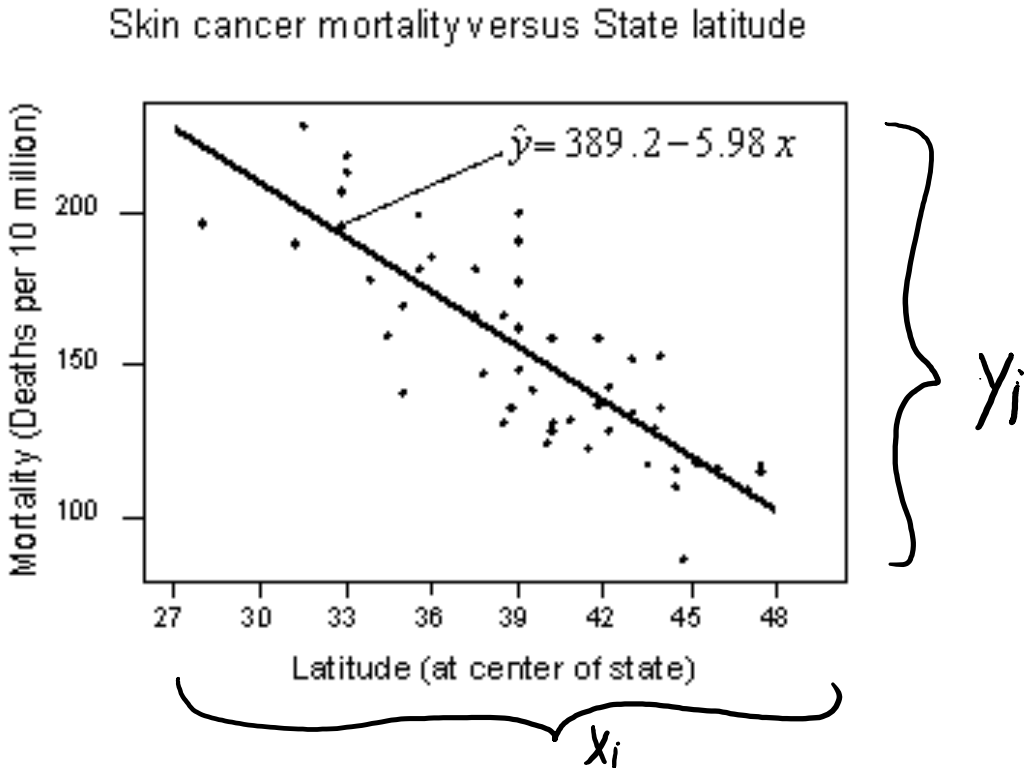


Sr. Francis Galton

Example: Dependent vs. Explanatory Variables

- We want to predict a numerical value given features:
  - Does number of lung cancer deaths change with number of cigarettes?
  - Does number of skin cancer deaths change with latitude?

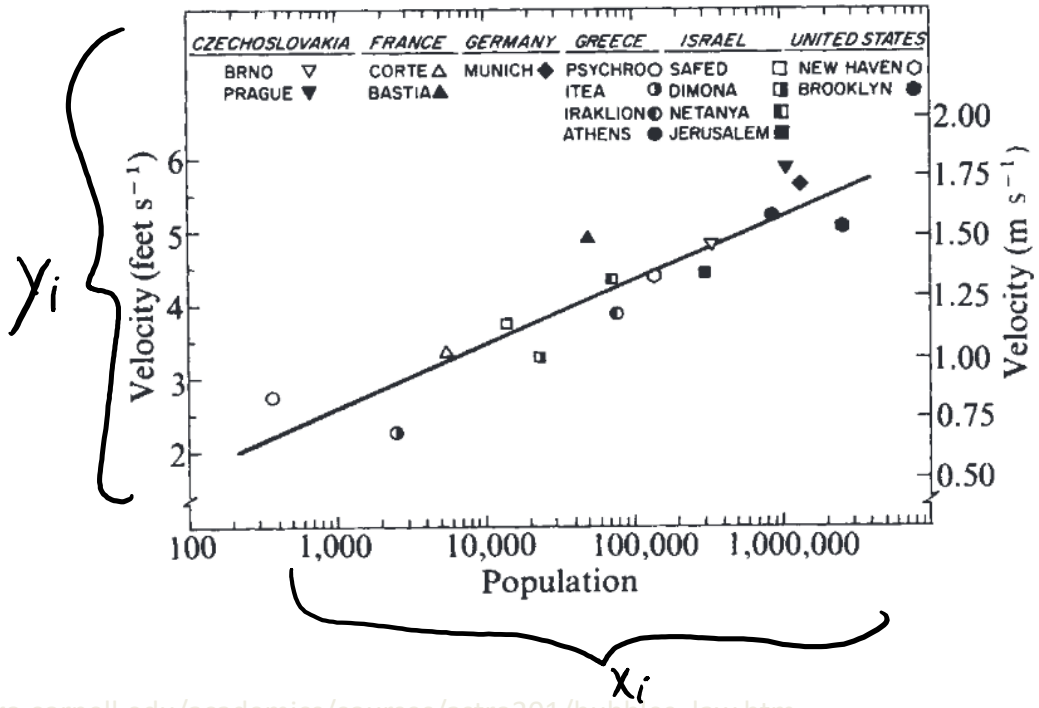


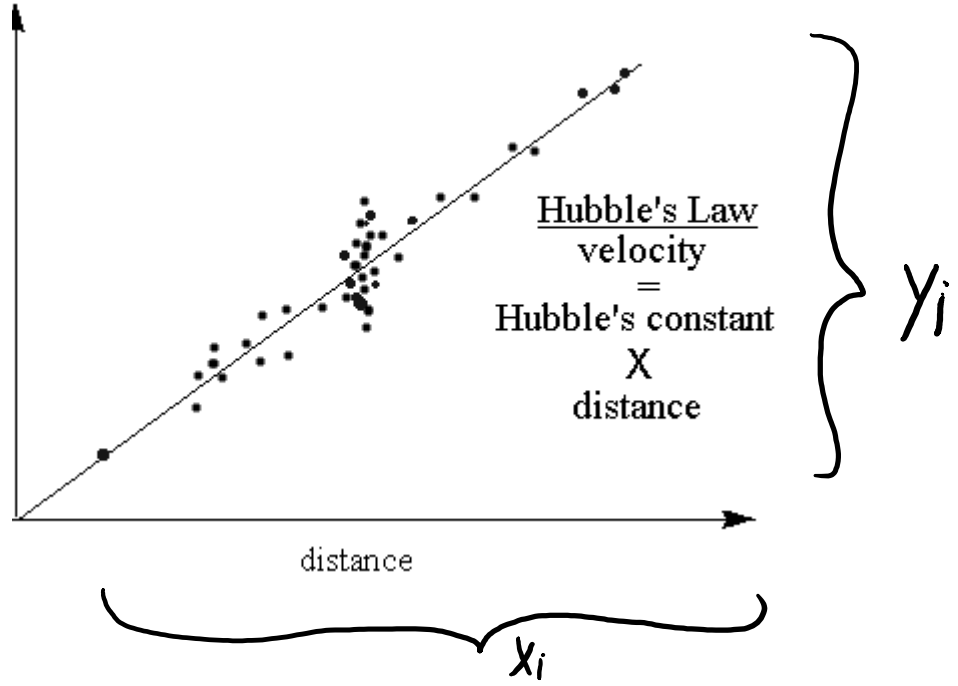


http://www.cvgs.k12.va.us:81/digstats/main/inferant/d_regrs.html
https://onlinecourses.science.psu.edu/stat501/node/11

Example: Dependent vs. Explanatory Variables

- We want to predict a numerical value given features:
  - Do people in big cities walk faster?
  - Is the universe expanding or shrinking or staying the same size?

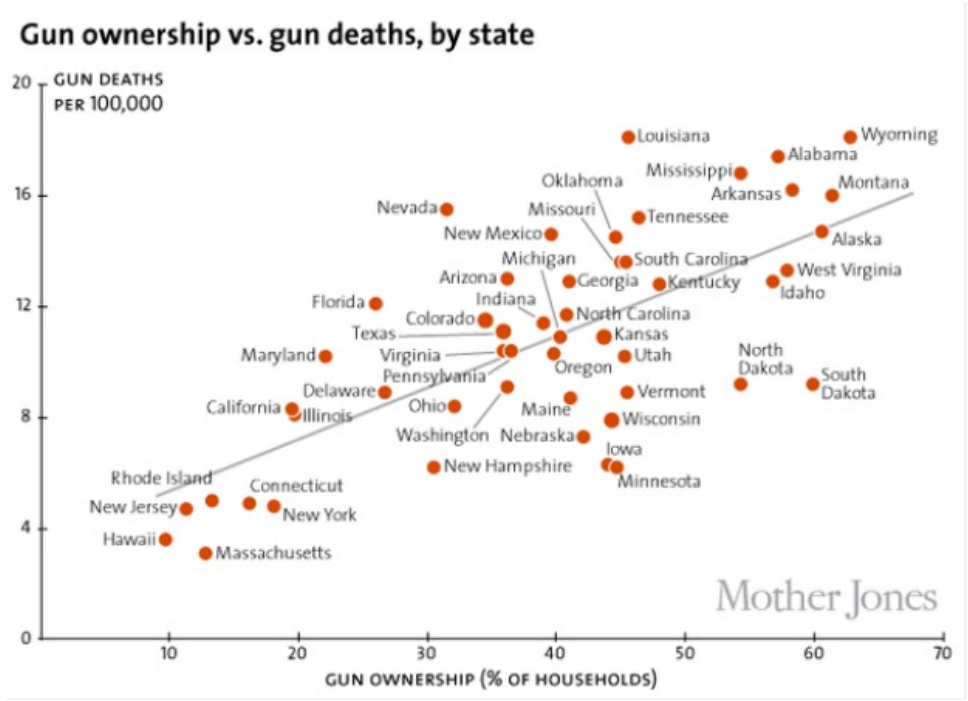


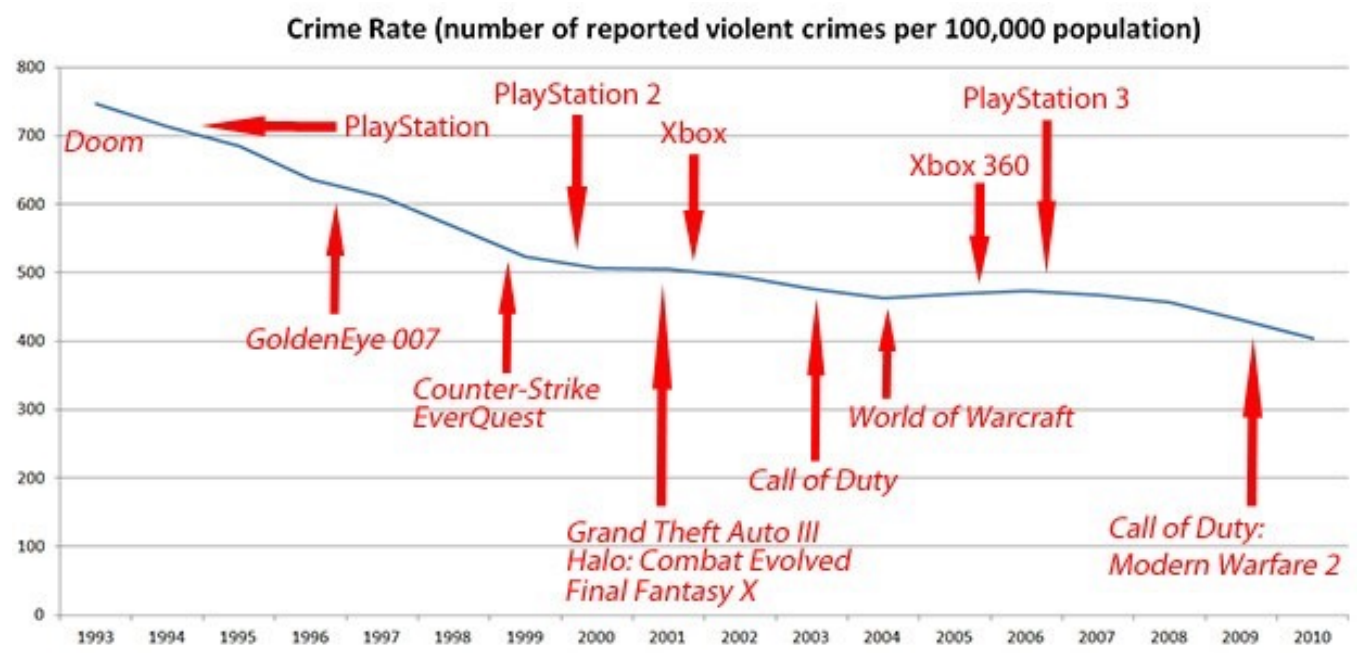


http://hosting.astro.cornell.edu/academics/courses/astro201/hubbles_law.htm
https://www.nature.com/articles/259557a0.pdf

Example: Dependent vs. Explanatory Variables

- We want to predict a numerical value given features:
  - Does number of gun deaths change with gun ownership?
  - Does number violent crimes change with violent video games?





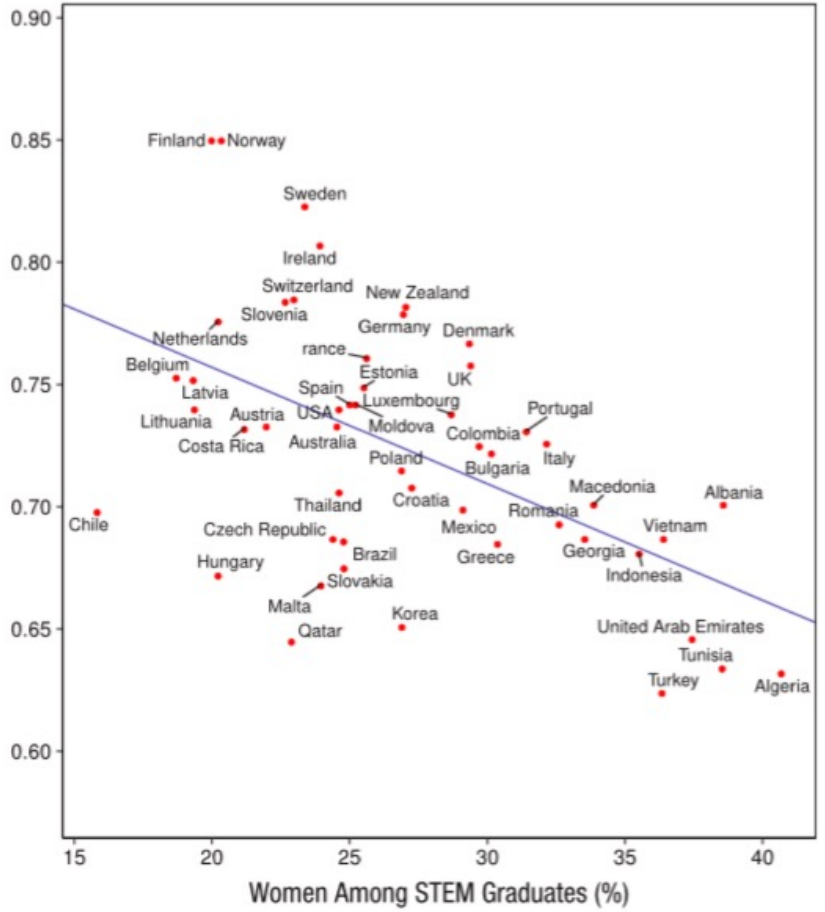
http://www.vox.com/2015/10/3/9444417/gun-violence-united-states-america
https://www.soundandvision.com/content/violence-and-video-games

Example: Dependent vs. Explanatory Variables

- We want to predict a numerical value given features:
  - Does higher gender equality index lead to more women STEM grads?

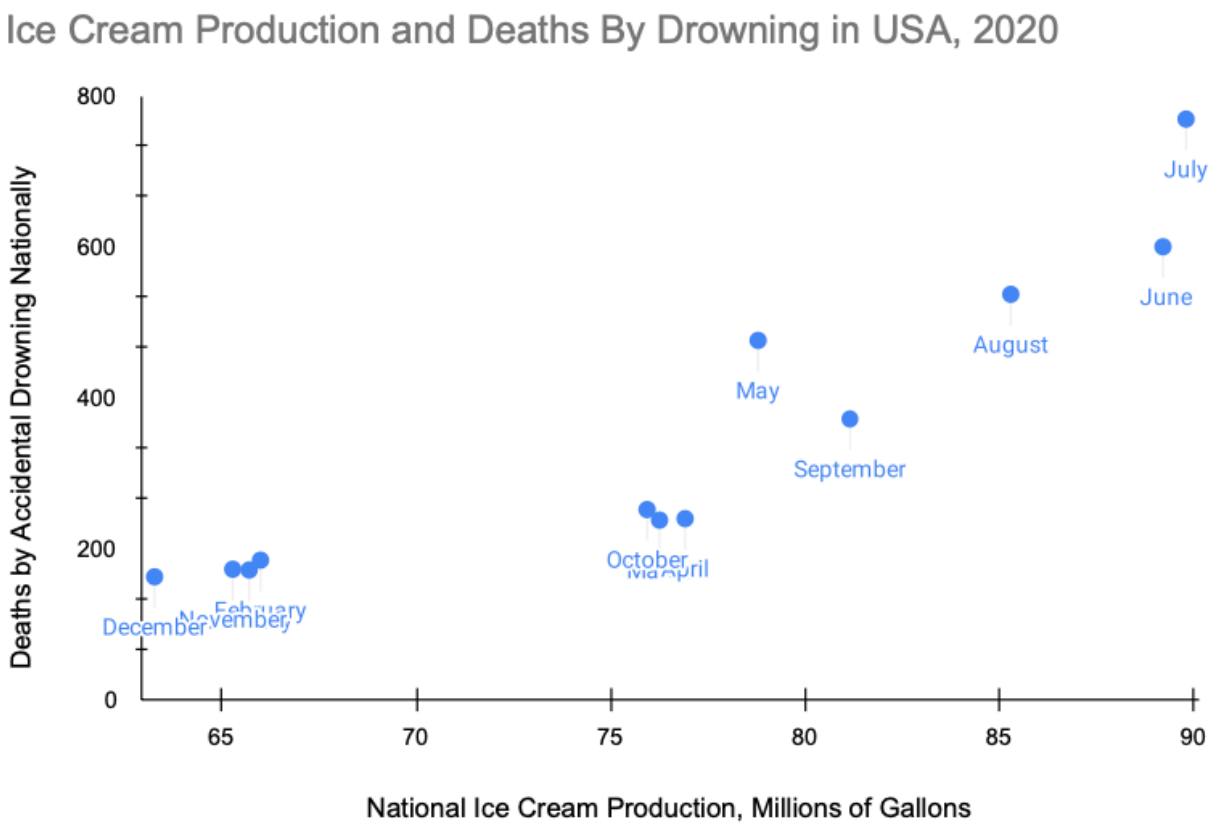
- Note that we are doing supervised learning:
  - Trying to predict value of 1 variable (the 'y_i' values). (instead of measuring correlation between 2).

- Supervised learning does not give causality:
  - OK: "Higher index is correlated with lower grad %".
  - OK: "Higher index helps predict lower grad %".
  - BAD: "Higher index leads to lower grads %".
    - People/media get these confused all the time, be careful!
    - There are lots of potential reasons for this correlation.



https://www.weforum.org/agenda/2018/02/does-gender-equality-result-in-fewer-female-stem-grads/

Correlation and Causation



"cigarette-smoking and lung cancer, though not mutually causative, are both influenced by a common cause, in this case the individual genotype." --Sir Ronald Fisher

"Humans are pretty bad at causal inference. If they were so good at it, they wouldn't assign the cause of unexplained or random phenomena to imaginary deities, and religion would not exist." -- Yann LeCun

Handling Numerical Labels

- One way to handle numerical y_i: discretize.
  - E.g., for 'age' could we use {'age ≤ 20', '20 < age ≤ 30', 'age > 30'}.
  - Now we can apply methods for classification to do regression.
  - But coarse discretization loses resolution.
  - And fine discretization requires lots of data ("coupon collecting").
- There exist regression versions of classification methods:
  - Regression trees, neighbour-based methods, and so on.
- Today: one of oldest, but still most popular/important methods:
  - Linear regression based on squared error.
  - Interpretable and the building block for more-complex methods.

Least Squares with one Feature

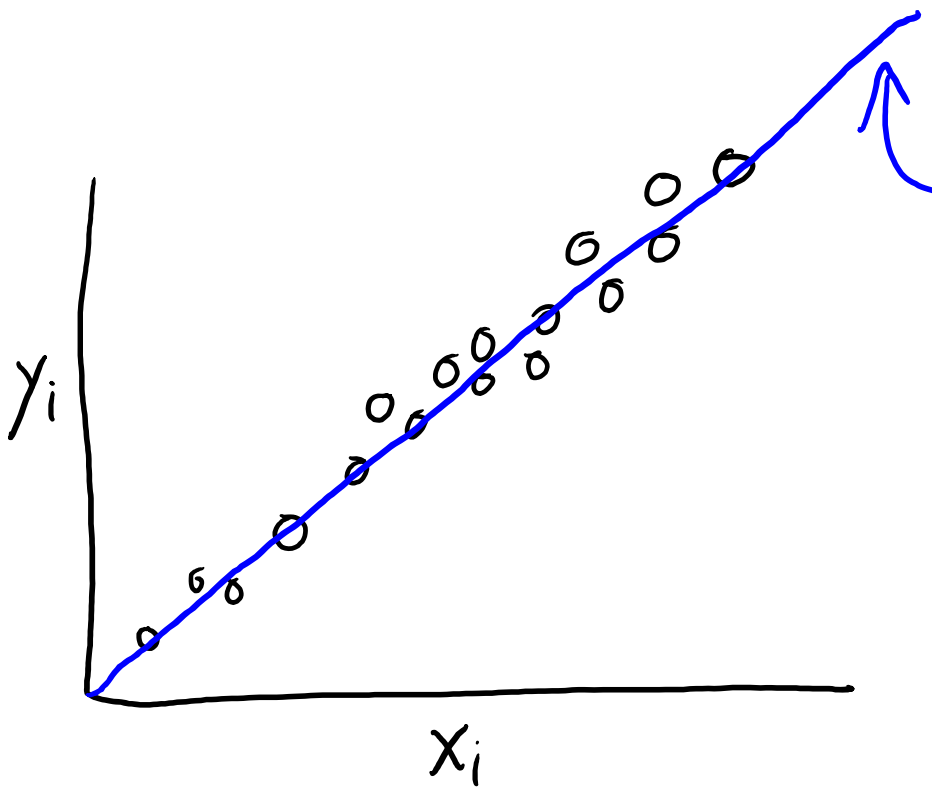
Linear Regression in 1 Dimension

- Assume we only have 1 feature (d = 1):
  - E.g., x_i is number of cigarettes and y_i is number of lung cancer deaths.
- Linear regression makes predictions $\hat{y}_i$ using a linear function of x_i:

$$\hat{y}_i = w x_i$$

- The parameter 'w' is the weight or regression coefficient of x_i.
  - We are temporarily ignoring the y-intercept.
- As x_i changes, slope 'w' affects the rate that $\hat{y}_i$ increases/decreases:
  - Positive 'w': $\hat{y}_i$ increase as x_i increases.
  - Negative 'w': $\hat{y}_i$ decreases as x_i increases.

Linear Regression in 1 Dimension



line $\hat{y}_i = wx_i$ for a particular slope w

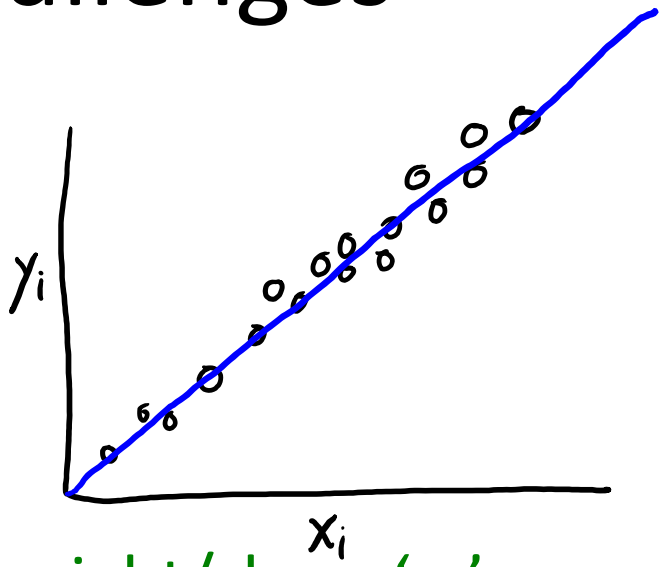
Aside: terminology woes

- Different fields use different terminology and symbols.
  - Data points ‘i’ = **objects** = **examples** = rows = observations.
  - **Inputs** x_i = predictors = **features** = explanatory variables= regressors = independent variables = covariates = columns.
  - **Outputs** y_i = outcomes = targets = response variables = dependent variables (also called a “label” if it’s categorical).
  - Regression coefficients ‘w’ = **weights** = parameters = betas.
- With linear regression, the symbols are inconsistent too:
  - In ML, the data is X and y, and the weights are w.
  - In statistics, the data is X and y, and the weights are β.
  - In optimization, the data is A and b, and the weights are x.

Linear Regression Training Challenges

- Linear regression makes predictions by using:

$$\hat{y}_i = w\tilde{x}_i$$



- To train a linear regression model, we need to find weight/slope ‘w’.
- Challenges in finding ‘w’ compared to fitting a decision stump:
  - Cannot enumerate all possible values of ‘w’ (could be any real number).
    - Instead, we will use calculus to find the best ‘w’.
  - It is unlikely that a line will go exactly through many data points.
    - Due to noise, relationship not being quite linear or just floating-point issues.
    - So it does not make sense to find the ‘w’ minimizing how many times $\hat{y}_i \neq y_i$.

Residuals and Sum of Squared Residuals

- The residual is the difference between our prediction and true value:

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

  - This can be positive or negative.
  - If this is close to zero, then our prediction is close to the true value.

- We typically look for a 'w' that makes residuals close to zero.
  - For example, many models minimize the sum of the squared residuals:

$$(\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + \cdots + (\hat{y}_n - y_n)^2$$

    - The smaller we make this, the smaller the distance between our predictions and targets.
  - Plugging in $\hat{y}_i = wx_i$ for the case of linear regression, we get:

$$(wx_1 - y_1)^2 + (wx_2 - y_2)^2 + \cdots + (wx_n - y_n)^2$$

    - The linear least squares model minimizes this function to choose the slope 'w'.

Linear Least Squares Objective Function

- Linear least squares sets 'w' is to minimize sum of squared residuals:

$$f(w) = \sum_{i=1}^{n} (wx_i - y_i)^2$$

Sum up the squared differences over all training examples.

wx_i: Our prediction $\hat{y}_i$

y_i: True value of y_i

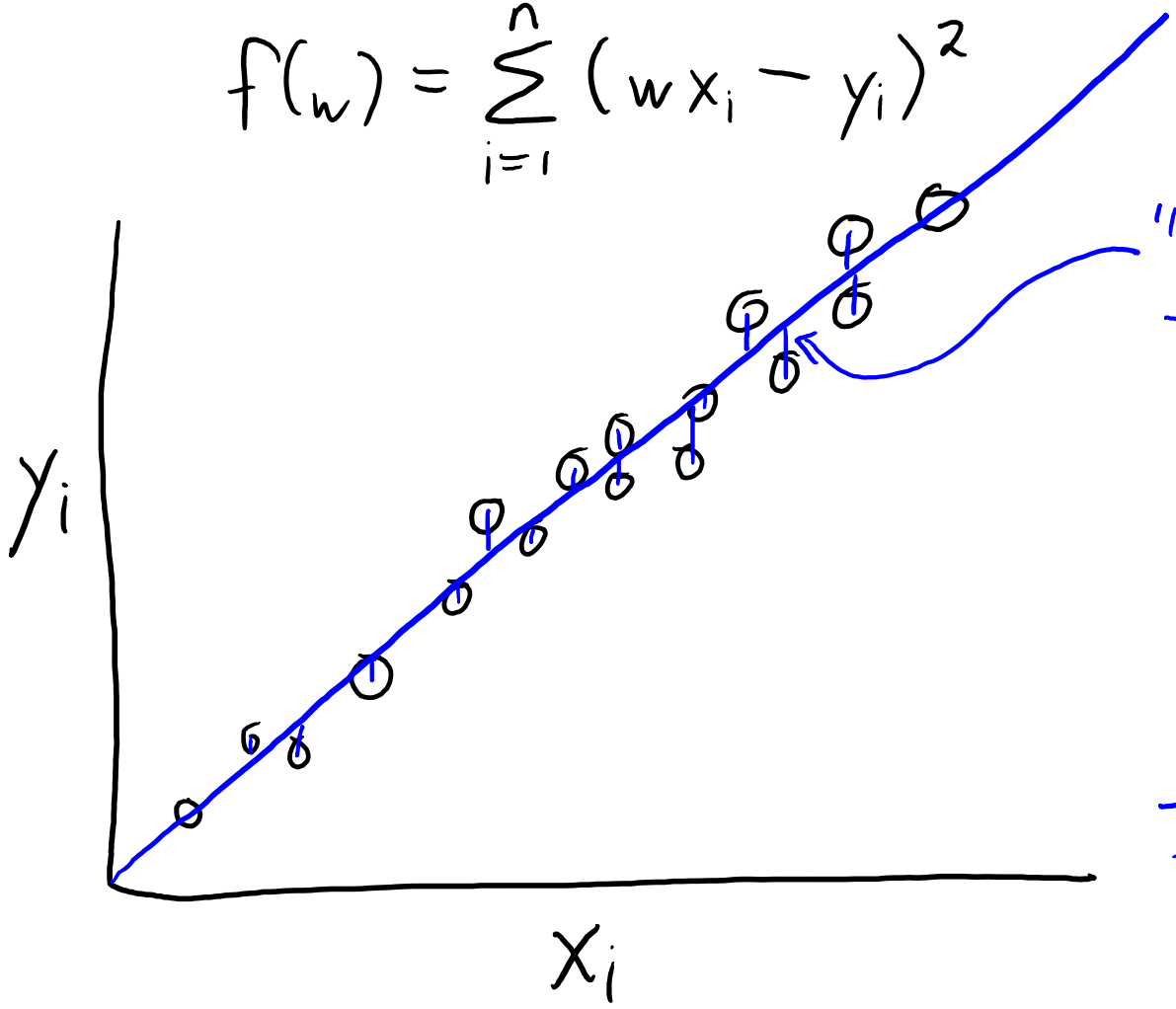
$wx_i - y_i$: Difference between prediction and true value for example 'i'. (residual)

- If this is zero, we exactly fit data. If this small, line is "close" to data.
- There are some justifications for choosing this function 'f'.
  - A probabilistic interpretation is coming later in the course.
- But usually, we choose this 'f' because it is easy to minimize.

Linear Least Squares Objective Function

- Linear least squares sets 'w' is to minimize sum of squared residuals:

$$f(w) = \sum_{i=1}^{n} (wx_i - y_i)^2$$



"Error" is the sum of the squared values of these vertical distances between the line (wx_i) and the targets (y_i)

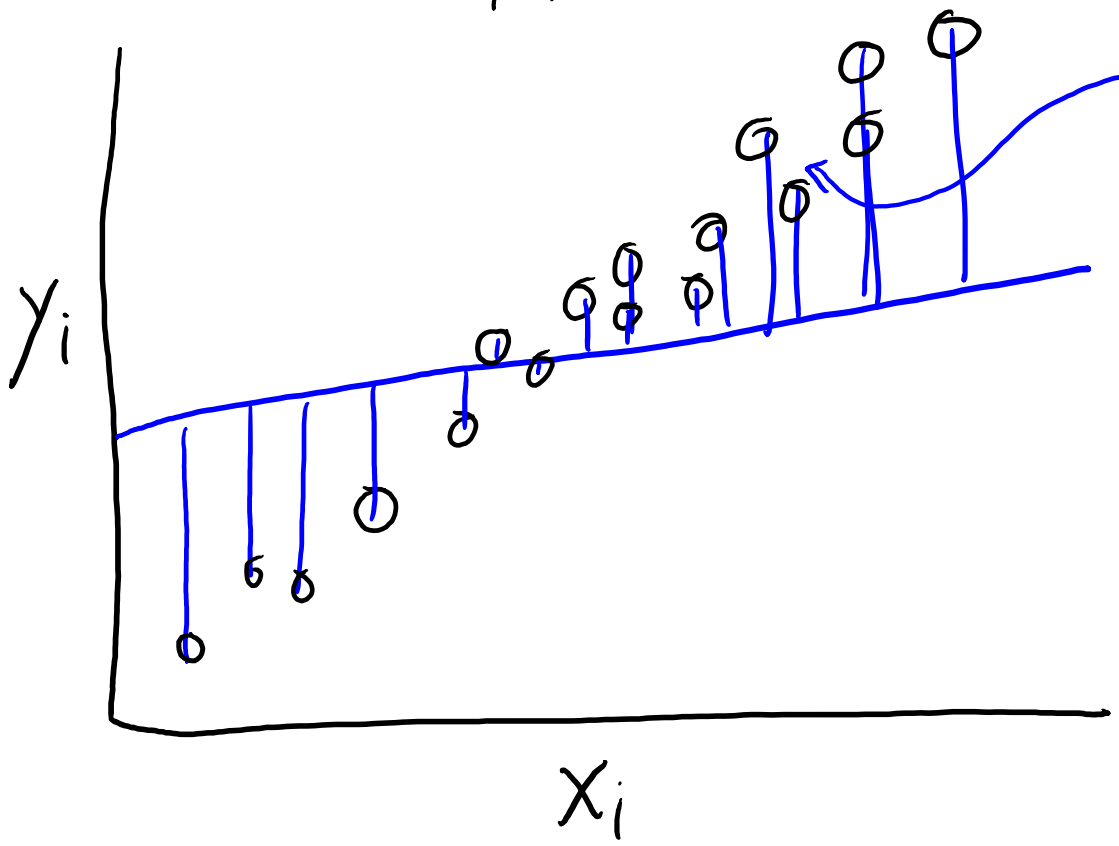
↓

If this error is small, then our predictions are close to the targets.

Linear Least Squares Objective Function

- Linear least squares sets 'w' is to minimize sum of squared residuals:

$$f(w) = \sum_{i=1}^{n} (wx_i - y_i)^2$$



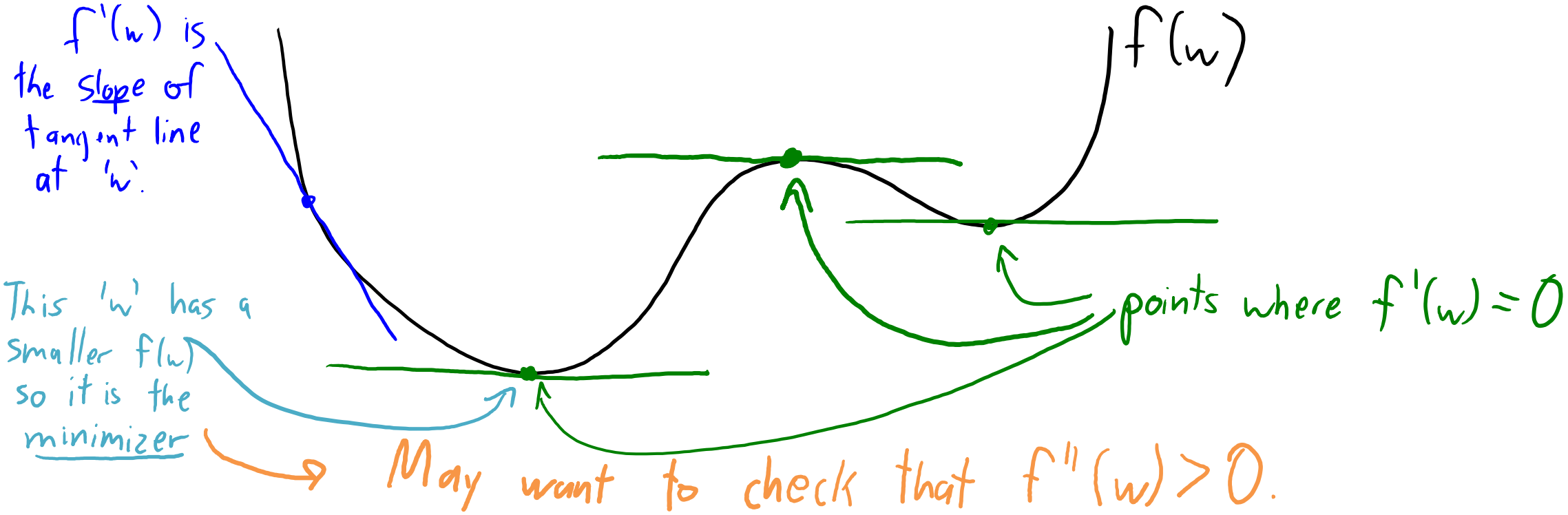
"Error" is the sum of the squared values of these vertical distances between the line (wx_i) and the targets (y_i)

↓

If this error is large, then our predictions are far from the targets.

Minimizing a Differential Function

- Math 101 approach to minimizing a differentiable function ‘f’:
  1. Take the derivative of ‘f’.
  2. Find points ‘w’ where the derivative f’(w) is equal to 0.
  3. Choose the smallest one (and check that f’’(w) is positive).



Digression: Multiplying by a Positive Constant

- Note that this problem:

$$f(w) = \sum_{i=1}^{n} (w x_i - y_i)^2$$

- Has the same set of minimizers as this problem:

$$f(w) = \frac{1}{2} \sum_{i=1}^{n} (w x_i - y_i)^2$$

- And these also have the same minimizers:

$$f(w) = \frac{1}{n} \sum_{i=1}^{n} (w x_i - y_i)^2 \qquad f(w) = \frac{1}{2n} \sum_{i=1}^{n} (w x_i - y_i)^2 + 1000$$

- I can multiply 'f' by any positive constant and not change solution.
  - Derivative will still be zero at the same locations.
  - We will use this trick a lot!

(Quora trolling on ethics of this)

Deriving Least Squares Solution

$$f(w) = \frac{1}{2}\sum_{i=1}^{n}(w x_i - y_i)^2 = \frac{1}{2}\sum_{i=1}^{n}\left[w^2 x_i^2 - 2 w x_i y_i + y_i^2\right]$$ (expand square)

$$= \frac{1}{2}\sum_{i=1}^{n} w^2 x_i^2 - \frac{1}{2}\sum_{i=1}^{n} 2 w x_i y_i + \frac{1}{2}\sum_{i=1}^{n} y_i^2$$ (split sum up into different terms)

$$= \frac{w^2}{2}\underbrace{\sum_{i=1}^{n} x_i^2}_{\text{constant 'a'}} - w\underbrace{\sum_{i=1}^{n} x_i y_i}_{\text{constant 'b'}} + \frac{1}{2}\underbrace{\sum_{i=1}^{n} y_i^2}_{\text{constant 'c'}}$$ (remove factors from sums that do not depend on example 'i')

$$= \frac{w^2}{2}a - wb + c$$

Take derivative: $f'(w) = wa - b + 0$

Setting $f'(w) = 0$ and solving gives $w = \frac{b}{a} = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$ (exists if we have a non-zero feature)

Finding Least Squares Solution

- Finding 'w' that minimizes sum of squared errors:

Setting $f'(w)=0$ and solving gives $w = \frac{\sum_{i=1}^{n} x_i y_i}{\sum_{i=1}^{n} x_i^2}$ (exists if we have one non-zero x_{ij})

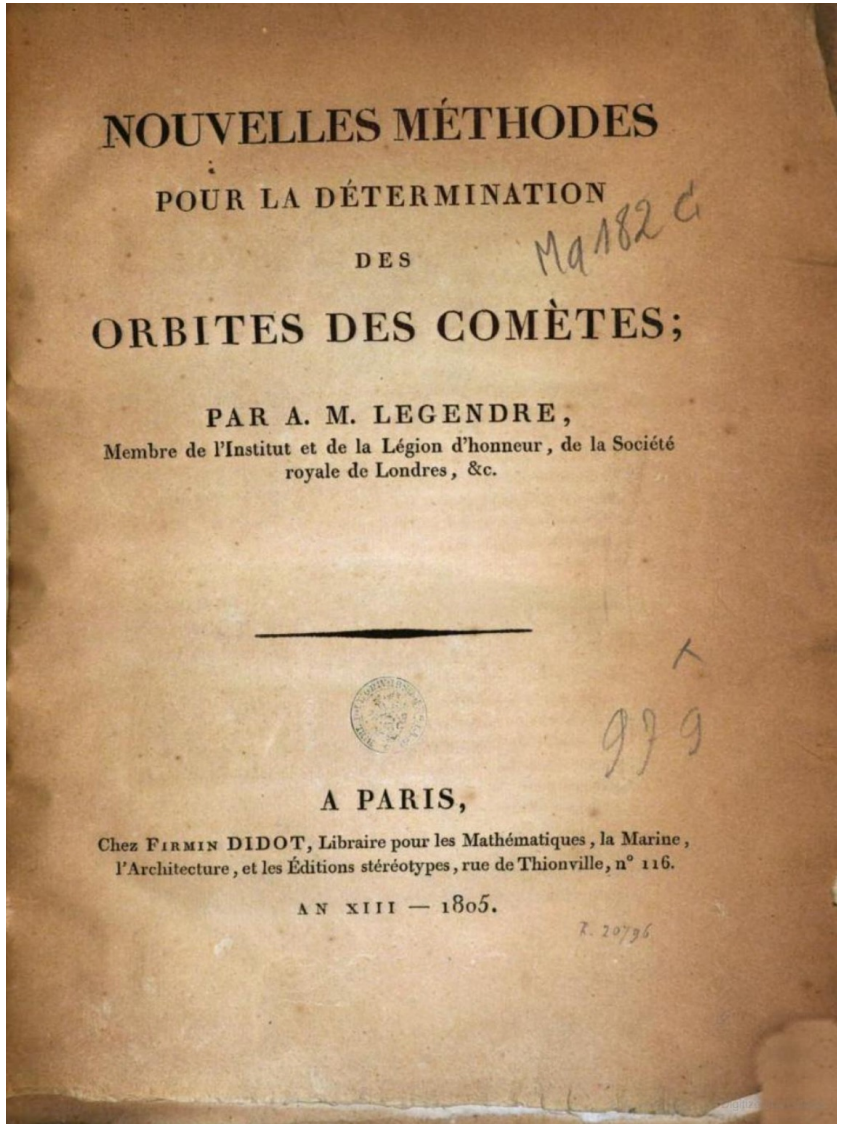
- Let's check that this is a minimizer by checking second derivative:

$$f'(w) = w\sum_{i=1}^{n} x_i^2 - \sum_{i=1}^{n} x_i y_i$$

$$f''(w) = \sum_{i=1}^{n} x_i^2$$

  - Since (anything)^2 is non-negative, we have $f''(w) \geq 0$.
  - If at least one feature is not zero, then $f''(w) > 0$ and 'w' is a minimizer.

The First Publication of Least Square



NOUVELLES MÉTHODES

POUR LA DÉTERMINATION

DES

ORBITES DES COMÈTES;

PAR A. M. LEGENDRE,

Membre de l'Institut et de la Légion d'honneur, de la Société royale de Londres, &c.

A PARIS,

Chez FIRMIN DIDOT, Libraire pour les Mathématiques, la Marine, l'Architecture, et les Éditions stéréotypes, rue de Thionville, n° 116.

AN XIII — 1805.

In 1809 Carl Friedrich Gauss published his method of calculating the orbits of celestial bodies. In that work he claimed to have been in possession of the method of least squares since 1795.[8] This naturally led to a priority dispute with Legendre. However, to Gauss's credit, he went beyond Legendre and succeeded in connecting the method of least squares with the principles of probability and to the normal distribution.

https://en.wikipedia.org/wiki/Least_squares

Next Topic: Least Squares in d-Dimensions

Motivation: Combining Explanatory Variables

- Smoking is not the only contributor to lung cancer.
  - For example, there environmental factors like exposure to asbestos.
- How can we model the combined effect of smoking and asbestos?
- A simple way is with a 2-dimensional linear function:

$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2}$$

"weight" of feature 1

Value of feature 1 in example 'i'

"weight" on feature 2.

Value of feature 2 in example 'i'

- We have a weight w_1 for feature '1' and w_2 for feature '2':

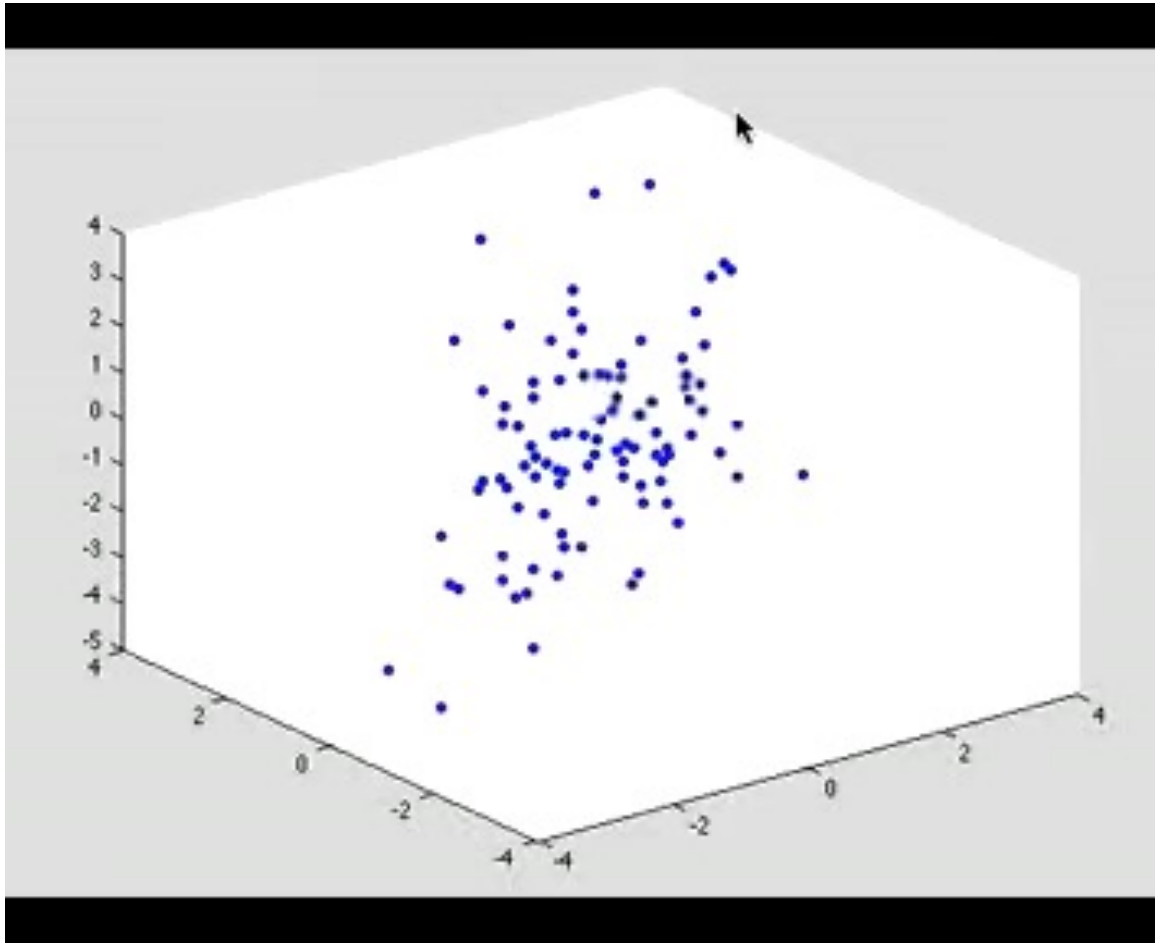
$$\hat{y}_i = 10(\#\text{cigarettes}) + 25(\#\text{asbetos})$$

Linear Regression in 2-Dimensions

- Linear model:

$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2}$$

- This defines a two-dimensional plane.



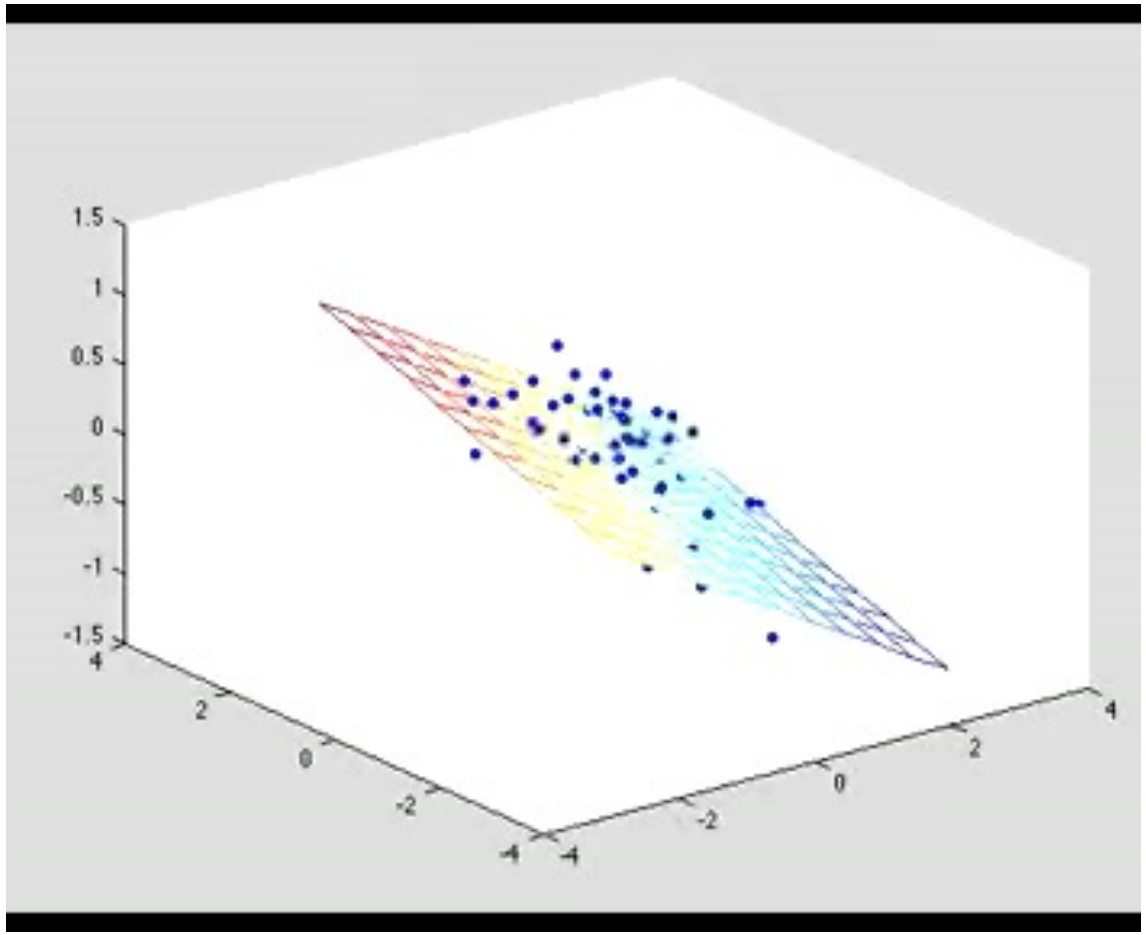
Linear Regression in 2-Dimensions

- Linear model:

$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2}$$

- This defines a two-dimensional plane.

- Not just a line!



Linear Regression in d-Dimensions

- If we have 'd' features, the d-dimensional linear model is:

$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2} + w_3 x_{i3} + \cdots + w_d x_{id}$$

  - In words, prediction is a weighted sum of the features.
- We can re-write this using summation notation as:

$$\hat{y}_i = \sum_{j=1}^{d} w_j x_{ij}$$

- We can again choose 'w' to minimize the sum of squared residuals:

$$f(w_1, w_2, \ldots, w_d) = \frac{1}{2} \sum_{i=1}^{n} \Big(\underbrace{\sum_{j=1}^{d} w_j x_{ij}}_{\hat{y}_i} - y_i \Big)^2$$

  - We can use multi-variable calculus to minimize 'f' with respect to the parameters w_1, w_2,…, w_d.

Minimizing Multi-Variable Differentiable Function

- With one variable, we "find 'w' where the derivative is equal to 0".
- The generalization of this idea to when we have 'd' variables:
  - "Find 'w' where the gradient vector is equal to the zero vector".

- Gradient is a vector with partial derivative 'j' in position 'j'.

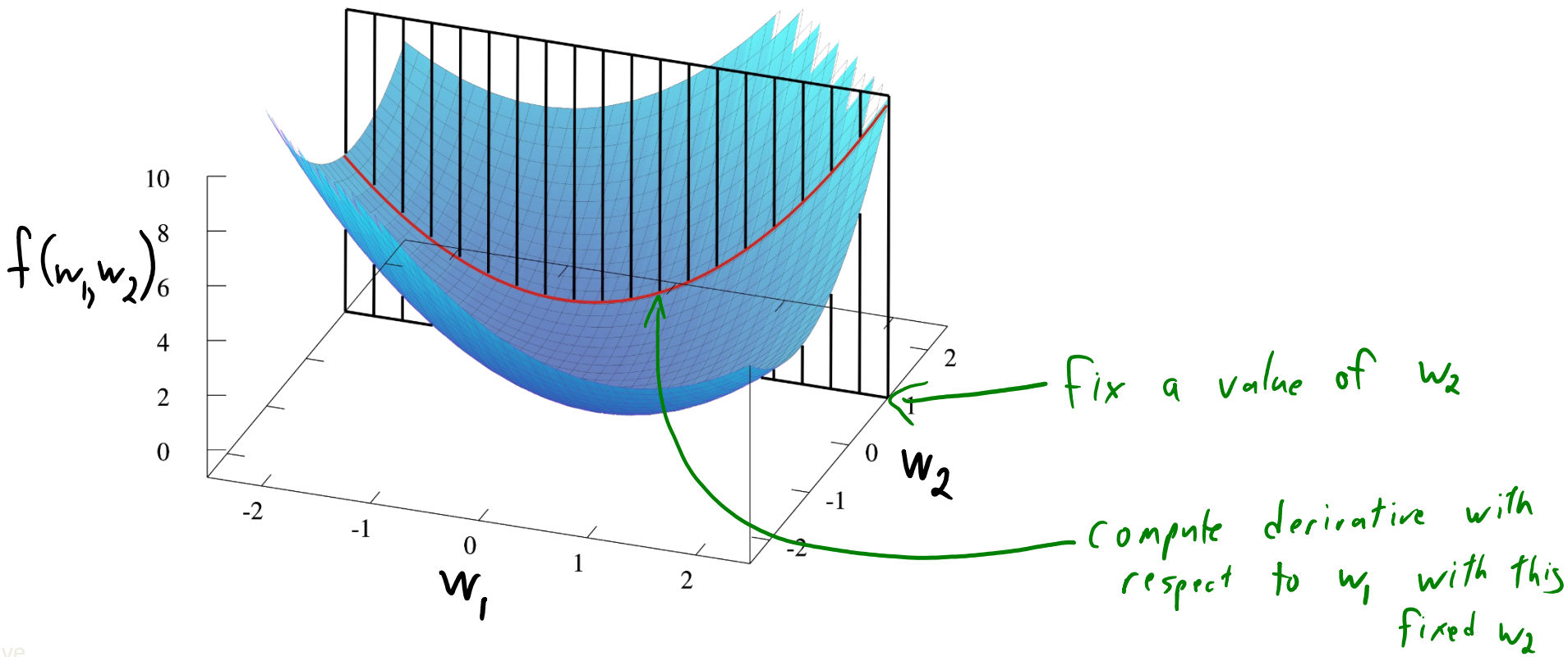
$$\nabla f(w) = \begin{bmatrix} \frac{\partial f}{\partial w_1} \\ \frac{\partial f}{\partial w_2} \\ \vdots \\ \frac{\partial f}{\partial w_d} \end{bmatrix}$$

gradient vector

partial derivative of 'f' with respect to variable w_2

Review: Partial Derivative

- Partial derivative with respect to w_j (written $\frac{\partial f}{\partial w_j}$).
  - Derivative with respect to w_j, keeping all others variables fixed.



https://en.wikipedia.org/wiki/Partial_derivative

Partial Derivative for Least Squares

- Partial derivative with respect to w_1 for least squares with n=1:

$$f(w_1, w_2, \dots, w_d) = \frac{1}{2}\left(\sum_{j=1}^{d} w_j x_{ij} - y_i\right)^2$$

$$\frac{\partial f}{\partial w_1} = \left(\sum_{j=1}^{d} w_j x_{ij} - y_i\right) x_{i1}$$

$$\sum_{j=1}^{d} w_j x_{ij} = w_1 x_{i1} + w_2 x_{i2} + \dots + w_d x_{id}$$

only one term involving w_1

$$\frac{\partial}{\partial w_1}\left[\frac{1}{2}\left(\sum_{j=1}^{d} w_j x_{ij} - y_i\right)^2\right] = \left(\sum_{j=1}^{d} w_j x_{ij} - y_i\right)\frac{\partial}{\partial w_1}\left[\sum_{j=1}^{d} w_j x_{ij} - y_i\right]$$

$$\frac{\partial}{\partial w_1}\frac{1}{2} g(w_1, w_2, \dots, w_d)^2 = g(w_1, w_2, \dots, w_d)\frac{\partial}{\partial w_1}\left[g(w_1, w_2, \dots, w_d)\right]$$

$$\frac{\partial}{\partial w_1}\left[w_1 x_{i1} + \text{terms with no } w_1\right] = x_{i1} + 0$$

0 0 0 0

Partial Derivative for Least Squares

- Partial derivative with respect to w_j for least squares with n=1:

$$f(w_1, w_2, \dots, w_d) = \frac{1}{2}\left(\sum_{j=1}^{d} w_j x_{ij} - y_i\right)^2$$

$$\frac{\partial f}{\partial w_j} = \left(\sum_{j'=1}^{d} w_{j'} x_{ij'} - y_i\right) x_{ij}$$

↳ We use j' to distinguish the summation index from the variable w_j we are differentiating

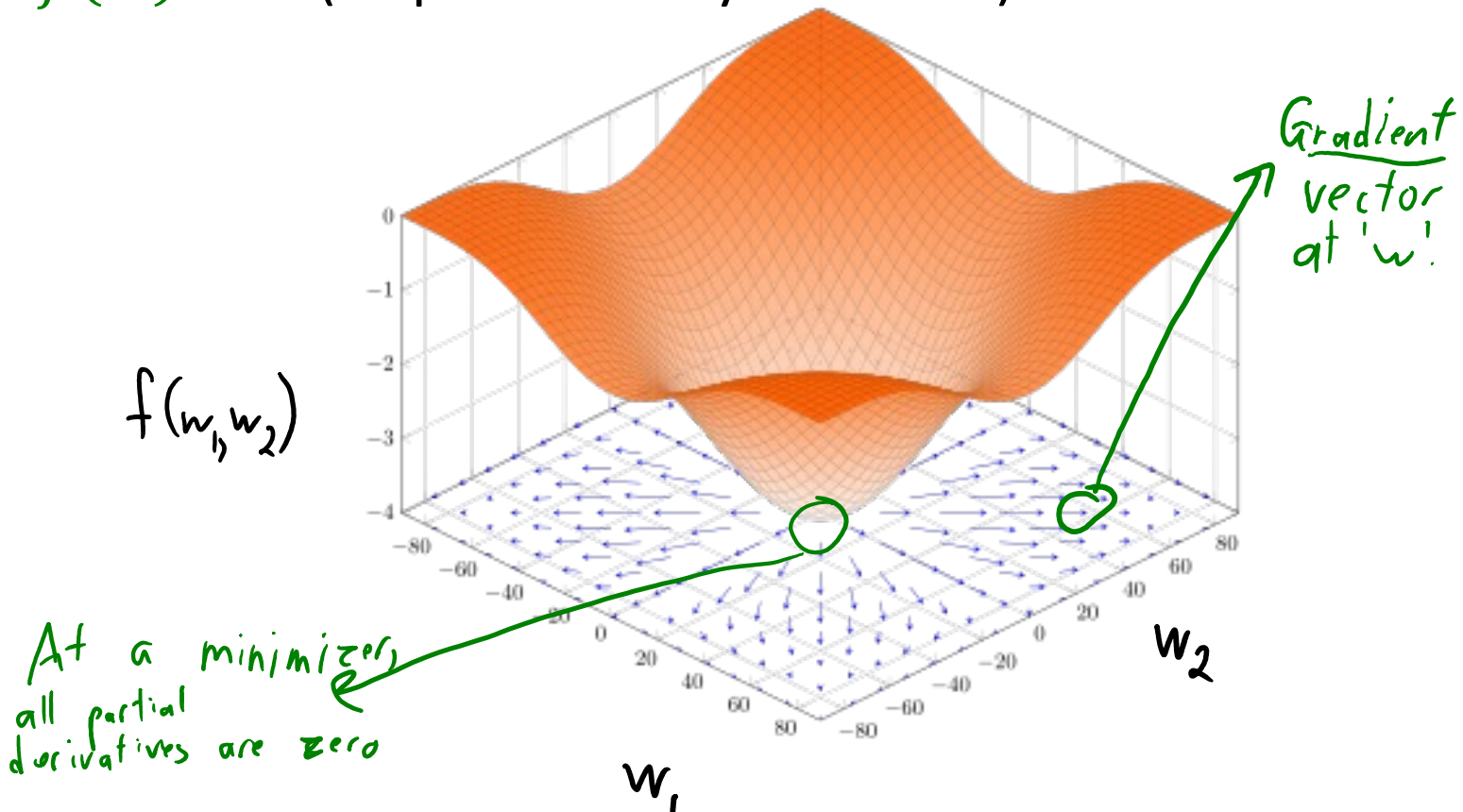
- Partial derivative with respect to w_j for least squares for general 'n':

$$f(w_1, w_2, \dots, w_d) = \frac{1}{2}\sum_{i=1}^{n}\left(\sum_{j=1}^{d} w_j x_{ij} - y_i\right)^2$$

$$\frac{\partial f}{\partial w_j} = \sum_{i=1}^{n}\left(\sum_{j'=1}^{d} w_{j'} x_{ij'} - y_i\right) x_{ij}$$

Gradient Vector for Least Squares

- The gradient vector is the concatenation of all partial derivatives:
  - At 'w', $\nabla f(w)$ is in the direction with steepest ascent.
  - At minimizers we have $\nabla f(w) = 0$ (slope is 0 every direction).



https://en.wikipedia.org/wiki/Gradient

Gradient Vector for Least Squares

- The gradient vector is the concatenation of all partial derivatives:
  - At 'w', $\nabla f(w)$ is in the direction with steepest ascent.
  - At minimizers we have $\nabla f(w) = 0$ (slope is 0 every direction).
- For linear least squares we have:

$$\nabla f(w) = \begin{bmatrix} \frac{\partial f}{\partial w_1} \\ \frac{\partial f}{\partial w_2} \\ \vdots \\ \frac{\partial f}{\partial w_d} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \left(\sum_{j=1}^{d} w_j x_{ij} - y_i \right) x_{i1} \\ \sum_{i=1}^{n} \left(\sum_{j=1}^{d} w_j x_{ij} - y_i \right) x_{i2} \\ \vdots \\ \sum_{i=1}^{n} \left(\sum_{j=1}^{d} w_j x_{ij} - y_i \right) x_{id} \end{bmatrix}$$

  - So to train a least squares model, we need this to equal the zero vector.

Fitting a Linear Least Squares Model

- Setting gradient to equal 0 vector for linear least squares gives:

$$\nabla f(w) = 0 \iff \begin{aligned} \sum_{i=1}^{n}\left(\sum_{j=1}^{d} w_j x_{ij} - y_i\right)x_{i1} &= 0 \\ \sum_{i=1}^{n}\left(\sum_{j=1}^{d} w_j x_{ij} - y_i\right)x_{i2} &= 0 \\ &\vdots \\ \sum_{i=1}^{n}\left(\sum_{j=1}^{d} w_j x_{ij} - y_i\right)x_{id} &= 0 \end{aligned}$$

  - This is a set of 'd' linear equations, with 'd' unknowns (w_1, w_2,..., w_d).
    - You can solve these equations using Gaussian elimination (linear algebra).

  - Claim: all 'w' with $\nabla f(w) = 0$ are minimizers (we will discuss why later).
    - May be more than one 'w' satisfying this, but all have the same minimum error.

Example Applications in Computational Biology

Article | Open access | Published: 08 February 2023

Dissecting cell identity via network inference and in silico gene perturbation

Kenji Kamimoto, Blerta Stringa, Christy M. Hoffmann, Kunal Jindal, Lilianna Solnica-Krezel & Samantha A. Morris

Nature **614**, 742–751 (2023) | Cite this article

machine-learning model. CellOracle builds a model that predicts the expression of a target gene on the basis of the expression of regulatory candidate genes:

$$x_j = \sum_{i=0}^{n} b_{i,j} x_i + c_j,$$

where x_j is single target gene expression and x_i is the gene expression value of the regulatory candidate gene that regulates gene x_j. $b_{i,j}$ is the coefficient value of the linear model (but $b_{i,j} = 0$ if $i = j$), and c is the intercept for this model. Here, we use the list of potential regulatory genes for each target gene generated in the previous base GRN construction step (ii).

Next Topic: Matrix Notation

Matrix Notation: Motivation

- We have expressed linear least squares with summation notation:

$$f(w_1, w_2, \dots, w_d) = \sum_{i=1}^{n} \left(\sum_{j=1}^{d} w_j x_{ij} - y_i \right)^2$$

- But you often see it equivalently expressed using matrix notation:

$$f(w) = \|Xw - y\|^2$$

- Why do people use matrix notation?
  - Can be easier to understand and lead to "nicer" code (once you are used to it).
  - Makes it easier to see some properties (like the connection to norms above).
    - Or derive properties, like showing that all 'w' with $\nabla f(w) = 0$ are minimizers.
  - Can lead to code with fewer bugs.
    - Since you can use existing implementations of standard operations.
  - Can lead to faster code.
    - If we are using packages that implement fast matrix operations.

Matrix/Norm Notation (MEMORIZE/STUDY THIS)

- In this course, vectors are assumed to be column vectors

$$w = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_d \end{bmatrix} \qquad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \qquad x_i = \begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix}$$

- Row i of '**X**' are actually the transpose of x_i

$$X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1d} \\ x_{21} & x_{22} & \cdots & x_{2d} \\ \vdots & & & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nd} \end{bmatrix} = \begin{bmatrix} \text{---} & x_1^T & \text{---} \\ \text{---} & x_2^T & \text{---} \\ & \vdots & \\ \text{---} & x_n^T & \text{---} \end{bmatrix}$$

Matrix Notation (MEMORIZE/STUDY THIS)

- Linear regression prediction for one example in matrix notation:

$$\hat{y}_i = w^T x_i$$

- Why?

$$\hat{y}_i = w_1 x_{i1} + w_2 x_{i2} + \cdots + w_d x_{id} = \underbrace{\begin{bmatrix} w_1 & w_2 & \cdots & w_d \end{bmatrix}}_{w^T} \underbrace{\begin{bmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{id} \end{bmatrix}}_{x_i} = w^T x_i$$

- Using $\hat{y}_i = w^T x_i$, we can re-write sum of squared residuals as:

$$\sum_{i=1}^{n} (\hat{y}_i - y_i)^2 = \sum_{i=1}^{n} (w^T x_i - y_i)^2$$

Matrix Notation (MEMORIZE/STUDY THIS)

- Linear regression prediction for all 'n' example in matrix notation:

$$\hat{y} = Xw$$

$n \times 1$ vector of predictions

$n \times d$ matrix of features

$d \times 1$ vector of parameters.

- Why?

$$Xw = \begin{bmatrix} — & x_1^T & — \\ — & x_2^T & — \\ & \vdots & \\ — & x_n^T & — \end{bmatrix} \begin{bmatrix} | \\ w \\ | \end{bmatrix} = \begin{bmatrix} x_1^T w \\ x_2^T w \\ \vdots \\ x_n^T w \end{bmatrix} = \begin{bmatrix} w^T x_1 \\ w^T x_2 \\ \vdots \\ w^T x_n \end{bmatrix} = \begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix} = \hat{y}$$

X

w

Prediction for example 'i' in row 'i'

We have $w^T x_i = x_i^T w$ because it is a scalar. (For example, $[5]^T = [5]$)

Matrix Notation (MEMORIZE/STUDY THIS)

- Linear regression residual vector in matrix notation:

$$r = X_w - y$$

r: $n \times 1$ vector of residuals $r_i = \hat{y}_i - y_i$

- Why?

$$r = \begin{bmatrix} r_1 \\ r_2 \\ \vdots \\ r_n \end{bmatrix} = \begin{bmatrix} \hat{y}_1 - y_1 \\ \hat{y}_2 - y_2 \\ \vdots \\ \hat{y}_n - y_n \end{bmatrix} = \underbrace{\begin{bmatrix} \hat{y}_1 \\ \hat{y}_2 \\ \vdots \\ \hat{y}_n \end{bmatrix}}_{\hat{y}} - \underbrace{\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}}_{y} = \hat{y} - y = X_w - y$$

$\hat{y} = X_w$ (last slide)

Matrix Notation (MEMORIZE/STUDY THIS)

- Different ways to write sum of residuals squared in linear regression model:

$$f(w) = \sum_{i=1}^{n}\left(\sum_{j=1}^{d} w_j x_{ij} - y_i\right)^2$$

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

$$= \sum_{i=1}^{n} r_i^2$$

$$= \|r\|^2$$

$$= \|Xw - y\|^2$$

Can also write $\|r\|^2 = r^T r$

Can also write $\|Xw-y\|^2 = (Xw-y)^T(Xw-y)$

or $\|\hat{y} - y\|^2$

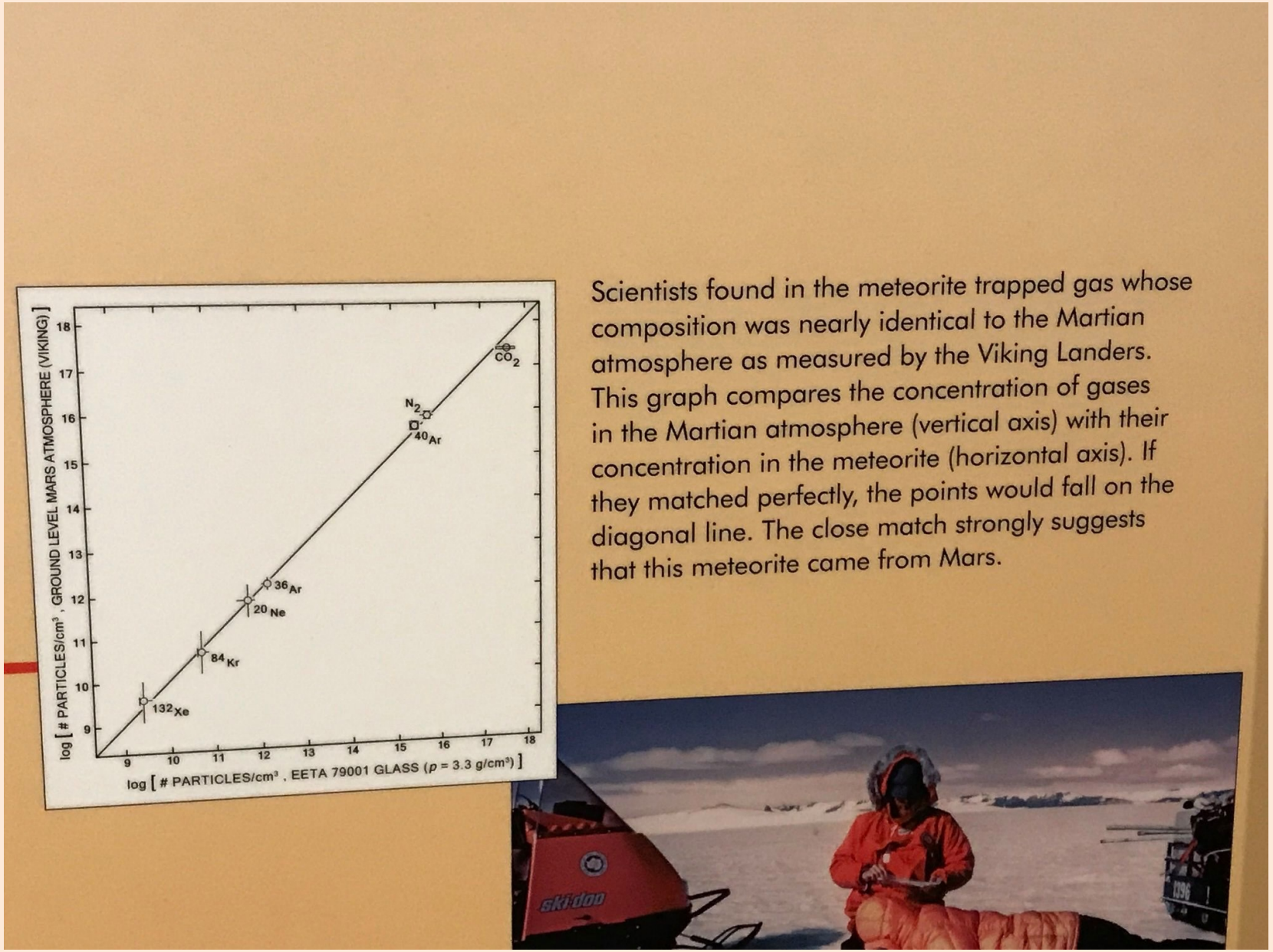
- So least squares minimizes L2-norm between target and predictions.

Summary

- Regression considers the case of a numerical y_i.
- Least squares is a classic method for fitting linear models.
  - Minimizes sum of squared residuals (prediction and true value difference).
  - With 1 feature, it has a simple closed-form solution.
  - Can be generalized to 'd' features, taking linear weighting of features.
- Gradient is vector containing partial derivatives of all variables.
- Matrix notation for expressing least squares problem: $||Xw - y||^2$.
- Next time:

minimizing $\frac{1}{2}\sum_{i=1}^{n}(w^T x_i - y_i)^2$ in terms of 'w' is: $w = (X'X)\backslash(X'y)$ (in Julia)

- In Smithsonian National Air and Space Museum (Washington, DC):



Causality, Interventions, and RCTs

- What if you want to assess causality?
- You can sometimes do this by collecting data in specific ways.
  - You need to set the values of the features "by intervention".
    - You do not passively observe, you *set* them and then watch the effect.
  - Most common way this is done is with a randomized control trial.
    - Say you want to evaluate the effectiveness of a pill for a certain disease.
    - You get a bunch of people with the disease for training data.
    - You randomly decide which of the people will take the pill, and which won't.
    - If the people who got the pill did better/worse on average, it was caused by the pill.
      - The randomness takes away the possibility that certain groups are more/less likely to take the pill.
      - Group not taking the pill often given placebo, removing effect of "feel like you are being treated".
      - Often the researchers do not even get to know who took the pills until after the study is over.
        - » "Double blind", to avoid the researchers giving hints about who got the pill.

Converting Partial Derivative to Matrix Notation

- Re-writing linear least squares partial derivative in matrix notation:

$$\frac{\partial f}{\partial w_j} = \sum_{i=1}^{n}\left(\sum_{j'=1}^{d} w_{j'} x_{ij'} - y_i\right)x_{ij} \qquad \text{(from earlier)}$$

$$= \sum_{i=1}^{n}\left(w^T x_i - y_i\right)x_{ij} \qquad \text{(no need for } j'\text{)}$$

$$= \sum_{i=1}^{n} r_i x_{ij} \qquad \text{(definition of } r_i\text{)}$$

$$= r^T x^j$$

residual vector

x^j: column 'j' of 'X'

Converting Gradient to Matrix Notation

- Re-writing linear least squares gradient in matrix notation:

$$\nabla f(w) = \begin{bmatrix} \frac{\partial f}{\partial w_1} \\ \frac{\partial f}{\partial w_2} \\ \vdots \\ \frac{\partial f}{\partial w_d} \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} \left(\sum_{j=1}^{d} w_j x_{ij} - y_i \right) x_{i1} \\ \sum_{i=1}^{n} \left(\sum_{j=1}^{d} w_j x_{ij} - y_i \right) x_{i2} \\ \vdots \\ \sum_{i=1}^{n} \left(\sum_{j=1}^{d} w_j x_{ij} - y_i \right) x_{id} \end{bmatrix} = \begin{bmatrix} r^T x^1 \\ r^T x^2 \\ \vdots \\ r^T x^d \end{bmatrix}$$

(from last slide)

$$= \begin{bmatrix} \text{---} & x^1 & \text{---} \\ \text{---} & x^2 & \text{---} \\ & \vdots & \\ \text{---} & x^d & \text{---} \end{bmatrix} r = X^T r$$

X^T

$X^T r$: Transpose of 'X' multiplied by residual vector