Machine Learning and Data Mining

Lecture 2.1: Linear Regression

CS 434

RECAP

From Last Lecture

Why do we care about probability?

The world is a ***very*** uncertain place

- 30 years of Artificial Intelligence research danced around this fact

- Then some AI researchers decided to use some ideas from the eighteenth century to model uncertainty…

  - **For us:** Really useful view of machine learning (both in practice for algorithm design and for deriving theoretic results about learning)

Slide adapted from Dhruv Batra. Comic a modification from 2016 webcomic Shen Comix.

After taking this course, you drop out of OSU and join an illegal betting company focused on **underground mascot fights** because you want to start making some **real money with ML**.

Your new boss asks you:

- If Benny Beaver and the Oregon Duck face off, what is the probability Benny wins?

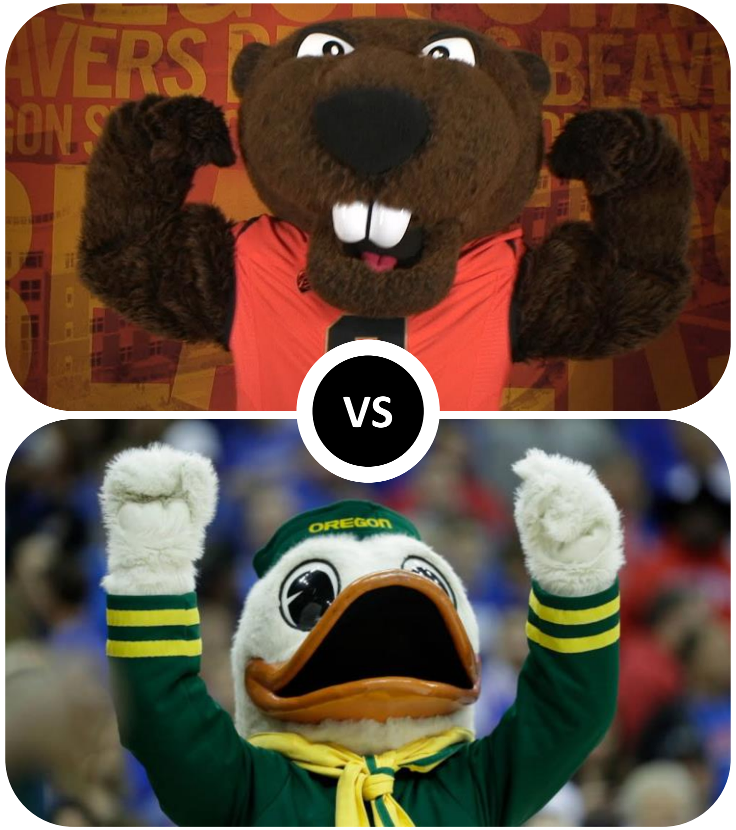
Knowing that data is everything, you ask:

- What happened in the past?

W, W, L, L, W

You say P(Benny Wins) = ?

Why?



This slide in no way condones or encourages underground mascot fights or illegal gambling. Even if we would win.

Your First Probabilistic Learning Algorithm: MLE

The intuitive answer is that 3/5 "fits the data" the best. We'll formalize that notion this lecture.

Maximum Likelihood Estimation – Find parameters that make the observed data most likely.

1. Assume a probabilistic model of how the data was generated $x \sim P(x; \theta)$ parameterized by some set of parameters θ
2. Find $\hat{\theta}_{MLE}$ that maximizes the probability (or likelihood) of generating the training data under the probabilistic model.

Why MLE?

- Often leads to "natural" or intuitive parameter estimates
- MLE is optimal if model class is correct (e.g. Normal model for normally distributed data)

Maximum A Posteriori – Find parameters that make the observed data most likely but consider a prior over the parameters.

1. Assume a prior distribution over θ, $P(\theta)$

2. Assume a probabilistic model of how the data is generated:
   -- parameter $\theta \sim P(\theta)$ and then data $x \sim P(x|\theta)$

3. Find $\hat{\theta}_{MAP}$ that maximize the posterior $P(\theta|D) \propto P(D|\theta)P(\theta)$

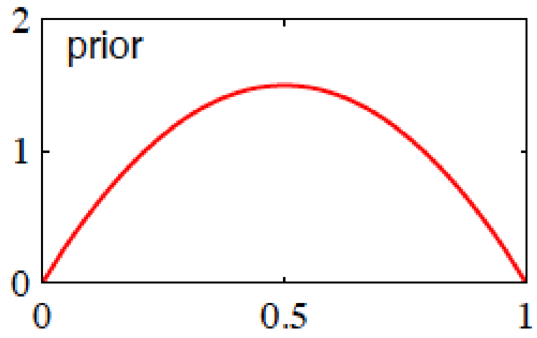
$$P(\theta|D) = \frac{P(D|\theta)P(\theta)}{P(D)}$$

Why MAP?

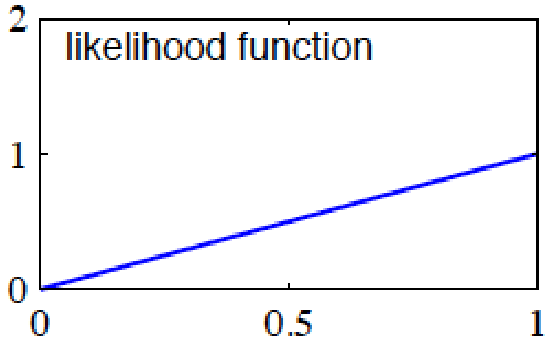
- Rigorous framework to combine observations (likelihood) with beliefs (prior)

Effect of Prior on Coin Flip

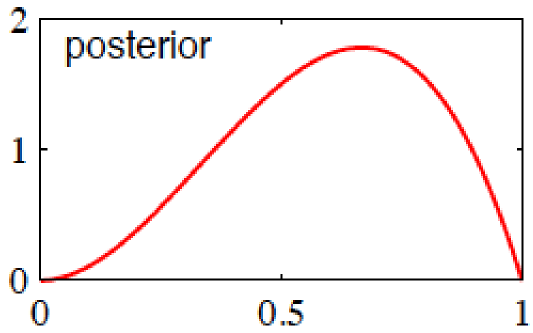
- **Prior = Beta(2,2)**
  - $\theta_{prior} = 0.5$



- **Dataset = {H}**
  - $L(\theta) = \theta$
  - $\theta_{MLE} = 1$



- **Posterior = Beta(3,2)**
  - $\theta_{MAP} = (3-1)/(3+2-2) = 2/3$



CS434 – ML + DM

Today's Learning Objectives

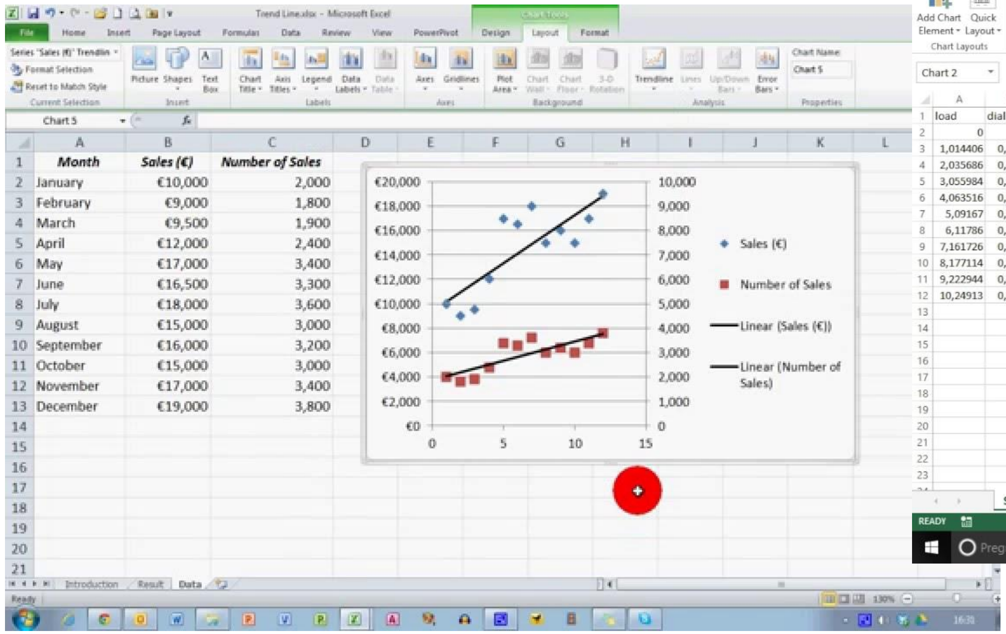
Be able to answer:

- What is linear regression?
- How do we find a solution for it?
  - With one dimensional input?
  - With multidimensional input?
    - We'll need to review some linear algebra

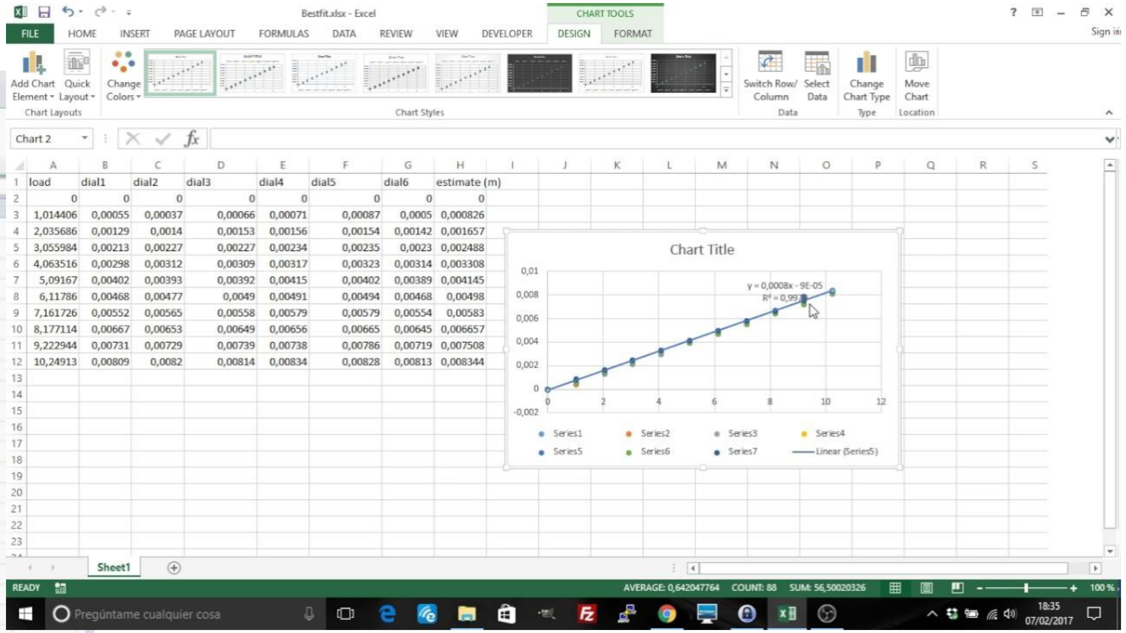
- How does this relate to MLE? Hint: Gaussians!
- What is regularization and how do they relate to priors from Bayesian statistics?

Regression → We are predicting some continuous output y

Linear → We assume there is a linear relationship between input x and output y



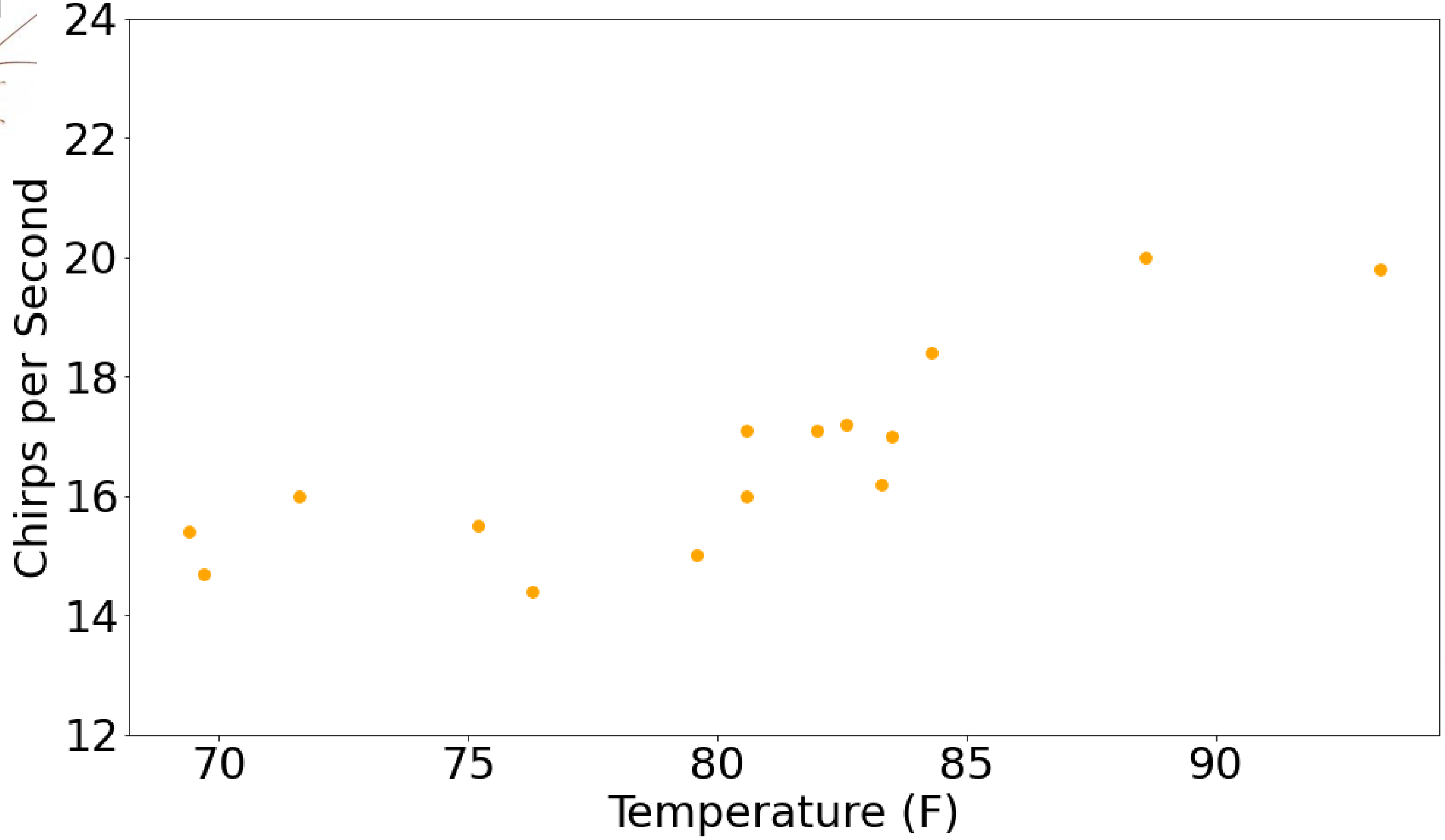
Month	Sales (€)	Number of Sales
January	€10,000	2,000
February	€9,000	1,800
March	€9,500	1,900
April	€12,000	2,400
May	€17,000	3,400
June	€16,500	3,300
July	€18,000	3,600
August	€15,000	3,000
September	€16,000	3,200
October	€15,000	3,000
November	€17,000	3,400
December	€19,000	3,800



load	dial1	dial2	dial3	dial4	dial5	dial6	estimate (m)
0	0	0	0	0	0	0	0
1,014406	0,00055	0,00037	0,00066	0,00071	0,00087	0,0005	0,000826
2,035686	0,00129	0,0014	0,00153	0,00156	0,00154	0,00142	0,001657
3,055984	0,00213	0,00227	0,00227	0,00234	0,00235	0,0023	0,002488
4,063516	0,00298	0,00312	0,00309	0,00317	0,00323	0,00314	0,003308
5,09167	0,00402	0,00393	0,00392	0,00415	0,00402	0,00389	0,004145
6,11786	0,00468	0,00477	0,0049	0,00491	0,00494	0,00468	0,00498
7,161726	0,00552	0,00565	0,00558	0,00579	0,00579	0,00554	0,00583
8,177114	0,00667	0,00653	0,00649	0,00656	0,00665	0,00645	0,006657
9,222944	0,00731	0,00729	0,00739	0,00738	0,00786	0,00719	0,007508
10,24913	0,00809	0,0082	0,00814	0,00834	0,00828	0,00813	0,008344

Can we predict how annoying crickets will be based on the temperature?

X	Y
F	CPS
88.6	20
71.6	16
93.3	19.8
84.3	18.4
80.6	17.1
75.2	15.5
69.7	14.7
82	17.1
69.4	15.4
83.3	16.2
79.6	15
82.6	17.2
80.6	16.
83.5	17
76.3	14.4

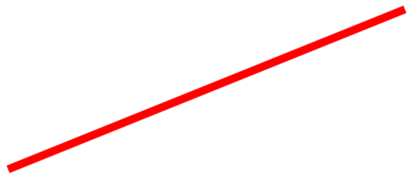


Data from: *The Song of Insects* by Dr.G.W. Pierce, Harvard College Press

Formalizing Linear Regression A Bit

We will only consider linear functions (thus the name **linear regression**):

Line



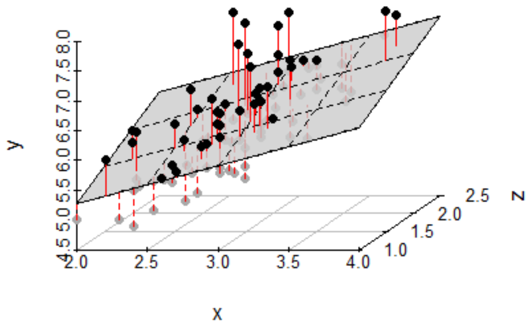
With one input dimension (d=1):

$$y = wx + b$$

Plane

With two input dimensions (d=2):

$$y = w_1 x_1 + w_2 x_2 + b$$



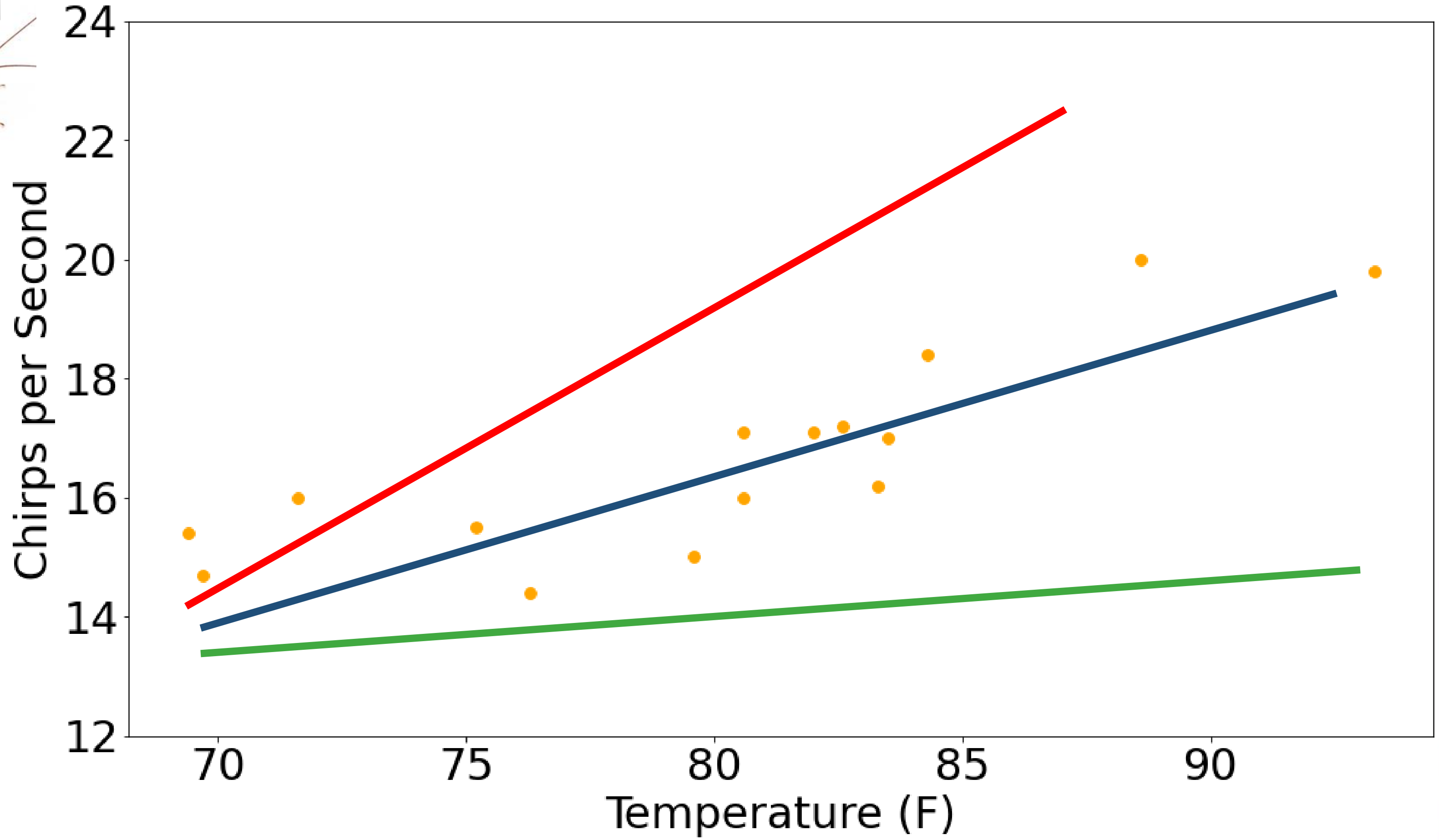
With d input dimensions

Hyperplane

$$y = b + \sum_{i=1}^{d} w_i x_i$$

?

Goal: fit a line through the points
Problem: the data isn't perfectly a line… how to measure "goodness" of a line?



Data from: The Song of Insects by Dr.G.W. Pierce, Harvard College Press

X	Y
F	CPS
88.6	20
71.6	16
93.3	19.8
84.3	18.4
80.6	17.1
75.2	15.5
69.7	14.7
82	17.1
69.4	15.4
83.3	16.2
79.6	15
82.6	17.2
80.6	16.
83.5	17
76.3	14.4

Dataset: Given a set of training examples $D = \{(x_i, y_i\}_{i=1}^{n}$

Goal: Learn w and b such that $wx_i + b$ predict y_i as closely as possible

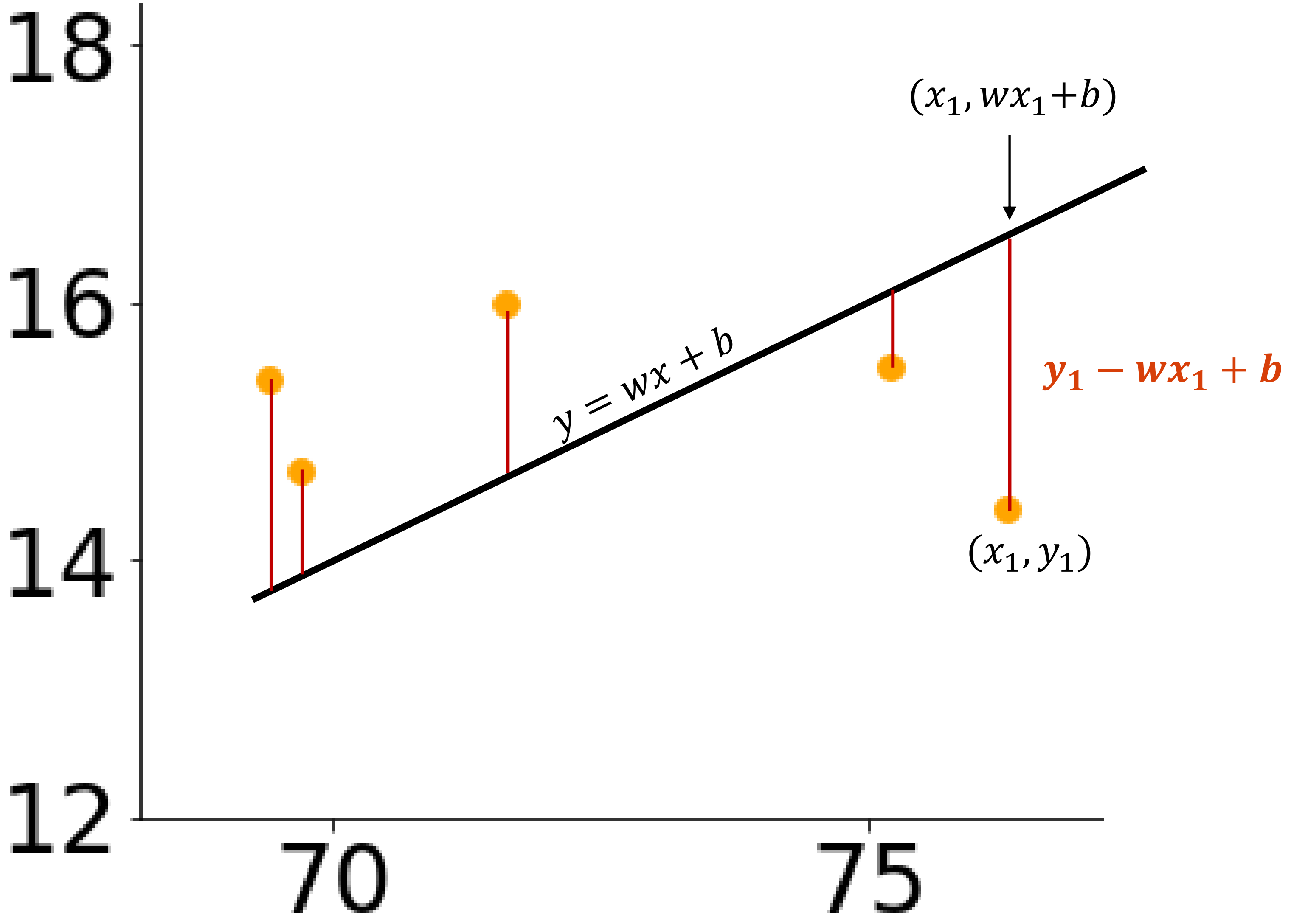
In mathematical terms, we want to find w and b that **minimize the error between $\boldsymbol{wx_i + b}$ and $\boldsymbol{y_i}$.** How should we measure this error?

Let me suggest squared error (or L2) error -- $\left(y_i - (wx_i + b)\right)^2$

Why? Gaussians my dear Watson. Gaussians. Also is a convex function which will make solving it easier.

Objective: Find parameters that minimize the **sum of squared error (SSE)**

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



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

Data from: The Song of Insects by Dr.G.W. Pierce, Harvard College Press

Objective: Find parameters that minimize the **sum of squared error (SSE)**

$$w^*, b^* = \underset{w,b}{\operatorname{argmin}} \sum_{i=1}^{n} (y_i - wx_i - b)^2$$

1) Take partial derivative w.r.t w and b respectively:

$$\frac{\delta SSE}{\delta w} = \qquad\qquad \frac{\delta SSE}{\delta b} =$$

Questions Break!

Find the partial derivative of the SSE objective w.r.t. w and b:

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

$$\frac{\delta SSE}{\delta w} =$$

$$\frac{\delta SSE}{\delta b} =$$

1) Take partial derivative w.r.t w and b respectively:

$$\frac{\delta SSE}{\delta w} = \sum_{i=1}^{n} 2(y_i - wx_i - b)x_i \qquad \frac{\delta SSE}{\delta b} = -\sum_{i=1}^{n} 2(y_i - wx_i - b)$$

2) Set equal to zero and solve:

$$\frac{\delta SSE}{\delta w} = -\sum_{i=1}^{n} 2(y_i - wx_i - b)x_i = 0 \xrightarrow{\text{yields}} w^* = \frac{\overline{xy} - \bar{y}\bar{x}}{\overline{x^2} - \bar{x}^2}$$

$$\frac{\delta SSE}{\delta b} = -\sum_{i=1}^{n} 2(y_i - wx_i - b) = 0 \xrightarrow{\text{yields}} b^* = \bar{y} - w^*\bar{x}$$

Let's see if this works:

X	Y
F	CPS
88.6	20
71.6	16
93.3	19.8
84.3	18.4
80.6	17.1
75.2	15.5
69.7	14.7
82	17.1
69.4	15.4
83.3	16.2
79.6	15
82.6	17.2
80.6	16.
83.5	17
76.3	14.4

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i = 80.03\bar{9}$$

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i = 16.65\bar{3}$$

$$\overline{xy} = \frac{1}{n}\sum_{i=1}^{n} x_i y_i = 1341.821\bar{3}$$

$$\bar{x}^2 = 6406.4015\bar{9}$$

$$\overline{x^2} = \frac{1}{n}\sum_{i=1}^{n} x_i^2 = 6448.390\bar{6}$$

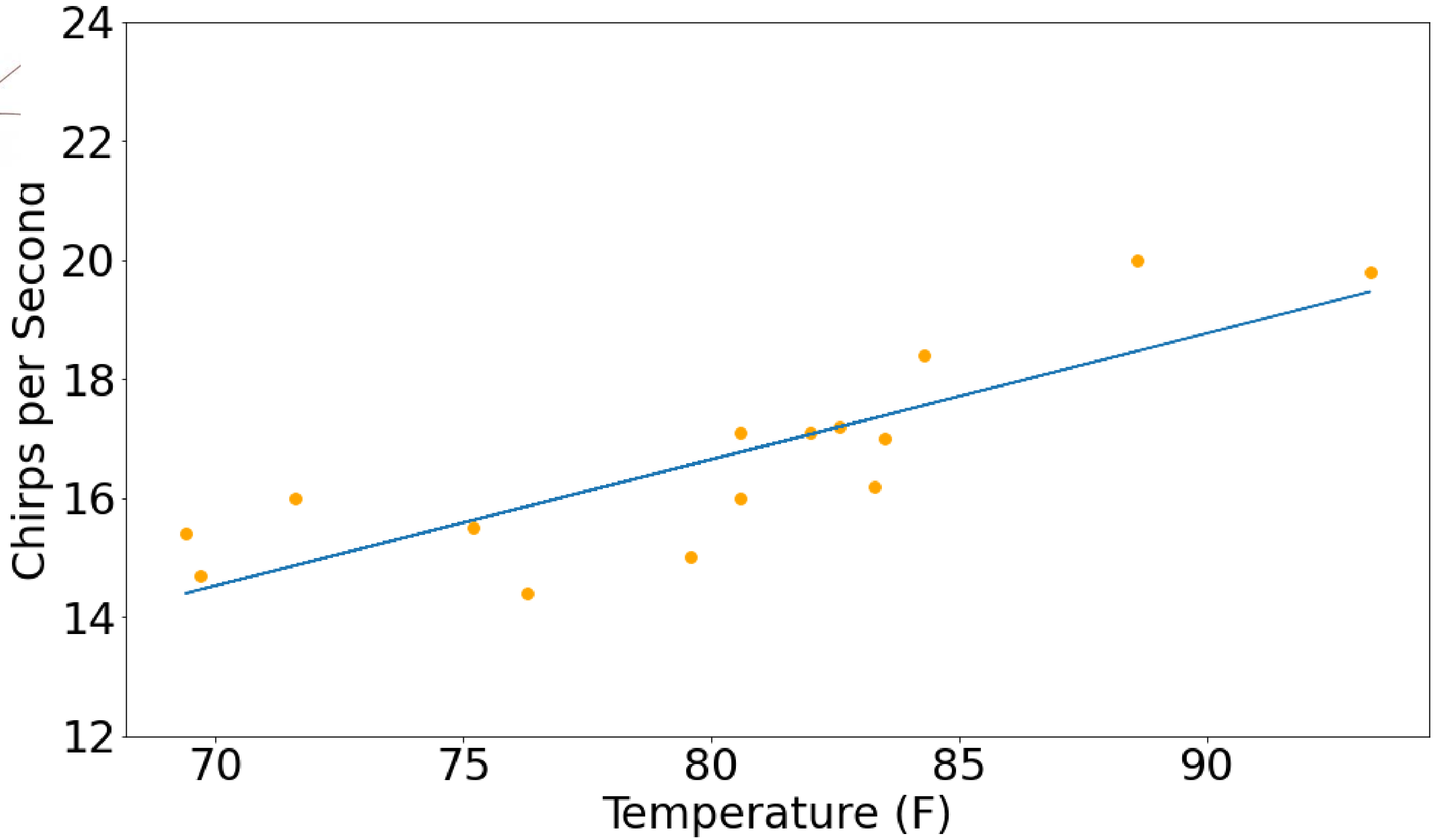
$$w^* = \frac{\overline{xy} - \bar{y}\bar{x}}{\overline{x^2} - \bar{x}^2} = \frac{1341.821\bar{3} - 16.65\bar{3} * 80.3\bar{9}}{6448.390\bar{6} - 80.03\bar{9}^2} = 0.2119$$

$$b^* = \bar{y} - w^*\bar{x} = 16.65\bar{3} - 0.2119 * 80.03\bar{9} = -3.0914$$

Let's see if this works:

X	Y
F	CPS
88.6	20
71.6	16
93.3	19.8
84.3	18.4
80.6	17.1
75.2	15.5
69.7	14.7
82	17.1
69.4	15.4
83.3	16.2
79.6	15
82.6	17.2
80.6	16.
83.5	17
76.3	14.4

$$w^* = 0.2119 \qquad b^* = -3.0914$$



Formalizing Linear Regression A Bit

We will only consider linear functions (thus the name **linear regression**):

Line

With one input dimension (d=1):

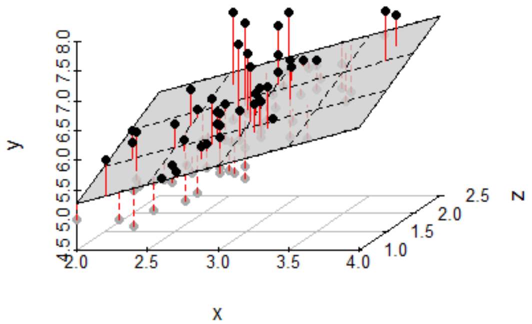
Nailed it

$$y = wx + b$$

Plane

With two input dimensions (d=2):

$$y = w_1 x_1 + w_2 x_2 + b$$



Hyperplane

With d input dimensions

$$y = b + \sum_{i=1}^{d} w_i x_i$$

?

Question Break!

Despite only having one input dimension, we had to learn two parameters w and b such that $y = wx + b$. What would happen if we chose not to include b? That is, if we just left b=0.

A The line would have to be horizontal

B Nothing. w would adjust to recover the same solution as if b was there.

C The line would have to go through the origin

D The line would not be defined

Same approach works in higher dimensions (i.e., when x_i is a vector)

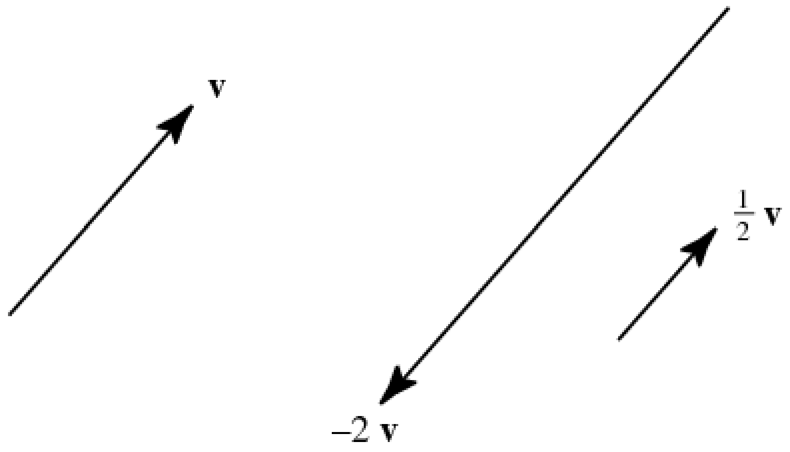
- Take partial derivative for each weight
- Set all partial derivatives to zero
- Solve the system of equations simultaneously

If only we had some tools for expressing systems of equations…

Time for a brief linear algebra and vector calculus "review"!

A **scalar** is a single number. Usually denoted by lowercase, unbolded letters. Example: $a = 3$

Called a "scalar" because multiplying a vector by a single number "scales it".



A **vector** is a one-dimensional array. Typically denote vectors as boldface lower case letters. Example:

$$\boldsymbol{x} = \begin{bmatrix} 2 \\ 3 \\ -2 \end{bmatrix}$$

If we don't specify otherwise, assume **x** is a column vector (i.e. one column with multiple rows). Will denote elements of a vector with x_i

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_d \end{bmatrix}$$

A **matrix** is a two-dimensional array. Typically denote matrices as unbolded upper case letters. Example:

$$X = \begin{bmatrix} 2 & 1 & 5 \\ -1 & 3 & -2 \end{bmatrix}$$

Will denote elements of a row with x_{ij} where i is the row and j is the column:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix}$$

Will talk about the **shape**/size/dimensionality of a matrix as number of rows x number of columns. Example: X has shape 2×3

Transposition (or the **transpose** operator) swaps rows and columns:

$$\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_d \end{bmatrix} \quad \boldsymbol{d \times 1}$$

$$\boldsymbol{x^T} = \begin{bmatrix} x_1 & x_2 & \dots & x_d \end{bmatrix} \quad \boldsymbol{1 \times d}$$

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} \quad \boldsymbol{2 \times 3}$$

$$X^T = \begin{bmatrix} x_{11} & x_{21} \\ x_{12} & x_{22} \\ x_{13} & x_{23} \end{bmatrix} \quad \boldsymbol{3 \times 2}$$

As a result, shape of the matrix swaps to. If X is n-by-m, then X^T is m-by-n

Matrix addition only defined for matrices of identical shape. Then is just performed element-wise:

$$X = \begin{bmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \end{bmatrix} \qquad Y = \begin{bmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \end{bmatrix}$$

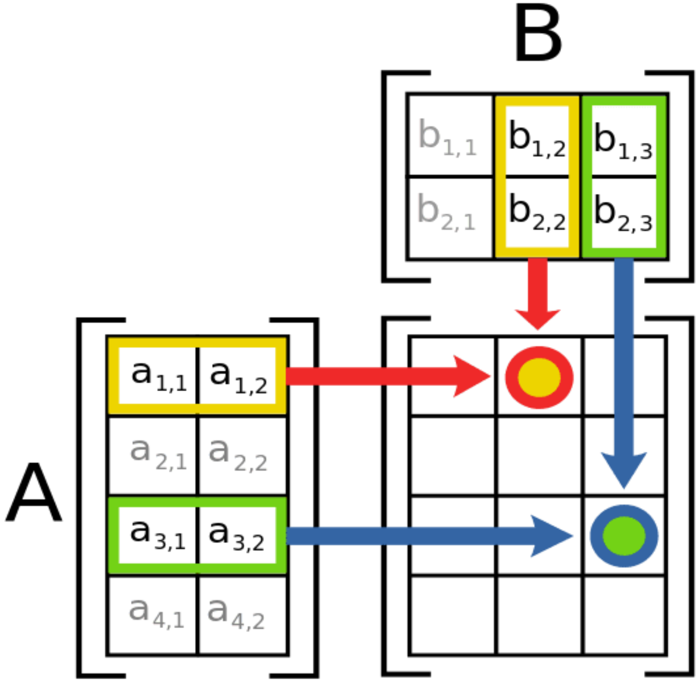
$$X + Y = \begin{bmatrix} x_{11} + y_{11} & x_{12} + y_{12} & x_{13} + y_{13} \\ x_{21} + y_{21} & x_{22} + y_{22} & x_{23} + y_{23} \end{bmatrix}$$

Output shape is the same as the input shapes.

Matrix multiplication only defined for matrices with identical *inner dimensions*. An m-by-n matrix can be multiplied by an r-by-c matrix if and only if n=r:

$$\boldsymbol{C} = \boldsymbol{AB}$$

$$c_{ij} = \sum_{k}^{n} a_{ik} * b_{kj}$$



Output shape is the *outer dimensions* of the matrices – m-by-c in this case.

Matrix multiplication only defined for matrices with identical *inner dimensions*. An m-by-n matrix can be multiplied by an r-by-c matrix if and only if n=r:

$$\begin{bmatrix} 2 & 5 & 2 \\ 1 & 0 & -2 \\ 3 & 1 & 1 \end{bmatrix} \begin{bmatrix} -2 & 1 & 0 \\ -2 & 2 & 1 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$$

Output shape is the *outer dimensions* of the matrices – m-by-c in this case.

Matrix multiplication only defined for matrices with identical *inner dimensions*. Unlike scalar multiplications, not commutative.

Property	Example
The commutative property of multiplication does not hold!	$AB \neq BA$
Associative property of multiplication	$(AB)C = A(BC)$
Distributive properties	$A(B+C) = AB + AC$
	$(B+C)A = BA + CA$
Multiplicative identity property	$IA = A$ and $AI = A$
Multiplicative property of zero	$OA = O$ and $AO = O$
Dimension property	The product of an $m \times n$ matrix and an $n \times k$ matrix is an $m \times k$ matrix.

Vector multiplication defined the same as matrix multiplication. Because vectors are just matrices with one dimension being 1.

$$\begin{bmatrix} y_1 & y_2 & \cdots & y_d \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_d \end{bmatrix} = y_1 x_1 + y_2 x_2 + \cdots + y_d x_d$$

This is often called the inner (or dot) product of two vectors. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = (\boldsymbol{x} \cdot \boldsymbol{y})$ **or** $\boldsymbol{x}^{\boldsymbol{T}} \boldsymbol{y}$ if x and y are both column vectors.

This is often called the outer product of two vectors. $\boldsymbol{x} \otimes \boldsymbol{y}$ or $\boldsymbol{x}\boldsymbol{y}^{\boldsymbol{T}}$ if x and y are both column vectors.

$$\begin{bmatrix} x_1 \\ x_2 \\ \cdots \\ x_d \end{bmatrix} \begin{bmatrix} y_1 & y_2 & \cdots & y_d \end{bmatrix} = \begin{bmatrix} x_1 y_1 & x_1 y_2 & x_1 y_3 \\ x_2 y_1 & x_2 y_2 & x_2 y_3 \\ x_3 y_1 & x_3 y_2 & x_3 y_3 \end{bmatrix}$$

Let $\mathrm{A} \in \mathbb{R}^{m \times n}, \mathrm{B} \in \mathbb{R}^{m \times n}, v \in \mathbb{R}^{mx1}$,
what is wrong with the following expression:

$$(A^T v)^T B^T + B^T (v \otimes v) v$$

A $(A^T v)^T B^T$ violates the rules of matrix multiplication

B Nothing is wrong with it

C It violates the rules of matrix addition

D $B^T (v \otimes v) v$ violates the rules of matrix multiplication

Vector norms define the "length" of a vector (or its distance from the zero vector). L2 should look familiar:

$$||\boldsymbol{x}||_2 = \sqrt{x_1^2 + x_2^2 + \cdots + x_d^2} = \sqrt{\boldsymbol{x} \cdot \boldsymbol{x}}$$

Our friend the Minkowski distance still hanging around as the Lp norm:

$$||\boldsymbol{x}||_p = \left(\sum_{i=1}^{d} |x_i|^p \right)^{\frac{1}{p}}$$

Question Break!

Let $\mathbf{v} = [7, 6]^{\mathrm{T}}$ and $\mathbf{w} = [3, 3]^{\mathrm{T}}$, what is $||\mathbf{v} - \boldsymbol{w}||_{\mathbf{2}}$?

A 7

B [4,3]

C 5

D 104

Matrix inverse of a square matrix A is a matrix A^{-1} such that $AA^{-1} = A^{-1}A = I$, where I is called an identity matrix.

$$A = \begin{bmatrix} 1 & 0 \\ 1 & \frac{1}{2} \end{bmatrix} \qquad A^{-1} = \begin{bmatrix} 1 & 0 \\ -2 & 2 \end{bmatrix} \qquad \begin{bmatrix} 1 & 0 \\ 1 & \frac{1}{2} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -2 & 2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

- A square matrix A is invertible iff its determinant is nonzero.
  - We won't play much with non-invertible matrices.
- Some properties:

$$(A^{-1})^{-1} = A \qquad (A^T)^{-1} = (A^{-1})^T$$

$$(kA)^{-1} = k^{-1}A^{-1} \qquad (AB)^{-1} = B^{-1}A^{-1}$$

Question Break!

If $x \in \mathbb{R}^{d \times 1}$ (x is a d-dimensional column vector), what would I call the operation xx^T and what shape would the result have?

A Dot product, 1x1

B Inner product, 1x1

C Outer product, dxd

D Cartesian product, dx1

Same approach works in higher dimensions (i.e., when x_i is a vector)

- Take partial derivative for each weight
- Set all partial derivatives to zero
- Solve the system of equations simultaneously

Now we consider the more general case that considers multiple input features.

X		Y
F	**H**	**CPS**
88.6	30	20
71.6	28	16
93.3	32	19.8
84.3	30	18.4
80.6	29	17.1
75.2	24	15.5
69.7	20	14.7
82	30	17.1
69.4	19	15.4
83.3	30	16.2
79.6	28	15
82.6	32	17.2
80.6	30	16.
83.5	30	17
76.3	24	14.4

In this example, each datapoint now has a 2-dimensional input $\boldsymbol{x} = [x_1, x_2]^{\boldsymbol{T}}$ where x_1 is temperature and x_2 is humidity.

Want to learn: $y = w_1 x_1 + w_2 x_2 + b$

It is inconvenient to have to represent b separately, so we use a small trick to fold it in to w.

- Add a feature to each **x** that is a constant 1 -- $\boldsymbol{x} = [1, x_1, x_2]^{\boldsymbol{T}}$ and let $\mathbf{w} = [b, w_1, w_2]^{\boldsymbol{T}}$

Now $y = \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{x} = 1b + x_1 w_1 + x_2 w_2$

How can we express the Sum of Squared Error objective?

One-dimensional Version:

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

Vectorized Multi-dimensional Version:

$$SSE(\boldsymbol{w}) = \sum_{i=1}^{n} \left(y_i - \boldsymbol{w}^T \boldsymbol{x}_i\right)^2$$

Can we express it entirely as matrix operations?!

X		Y
F	**H**	**CPS**
88.6	30	20
71.6	28	16
93.3	32	19.8
84.3	30	18.4
80.6	29	17.1
75.2	24	15.5
69.7	20	14.7
82	30	17.1
69.4	19	15.4
83.3	30	16.2
79.6	28	15
82.6	32	17.2
80.6	30	16.
83.5	30	17
76.3	24	14.4

$$\boldsymbol{X} = \begin{bmatrix} \boldsymbol{x}_1^T \\ \boldsymbol{x}_2^T \\ \vdots \\ \boldsymbol{x}_n^T \end{bmatrix} \quad n \times d$$

$$\boldsymbol{y} = \begin{bmatrix} \boldsymbol{y}_1 \\ \boldsymbol{y}_2 \\ \vdots \\ \boldsymbol{y}_n \end{bmatrix} \quad n \times 1$$

$$\mathbf{w} = [w_1, w_2, \dots, w_d]^{\boldsymbol{T}} \quad d \times 1$$

$$SSE(\boldsymbol{w}) = \sum_{i=1}^{n} \left(y_i - \boldsymbol{w}^{\boldsymbol{T}} \boldsymbol{x}_{\boldsymbol{i}}\right)^2 = (\boldsymbol{y} - X\boldsymbol{w})^T (\boldsymbol{y} - X\boldsymbol{w})$$

That seemed like magic. Show me how it works.

$$SSE(\boldsymbol{w}) = (\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w})$$

$$\boldsymbol{y} - X\boldsymbol{w} = \underset{n \times 1}{\begin{bmatrix} \boldsymbol{y_1} \\ \boldsymbol{y_2} \\ \vdots \\ \boldsymbol{y_n} \end{bmatrix}} - \underset{n \times d}{\begin{bmatrix} \boldsymbol{x_1^T} \\ \boldsymbol{x_2^T} \\ \vdots \\ \boldsymbol{x_n^T} \end{bmatrix}} \underset{d \times 1}{\begin{bmatrix} \boldsymbol{w_1} \\ \boldsymbol{w_2} \\ \vdots \\ \boldsymbol{w_d} \end{bmatrix}} = \underset{n \times 1}{\begin{bmatrix} y_1 - \boldsymbol{x_1^T w} \\ y_2 - \boldsymbol{x_2^T w} \\ \vdots \\ y_n - \boldsymbol{x_n^T w} \end{bmatrix}}$$

That seemed like magic. Show me how it works.

$$SSE(\boldsymbol{w}) = (\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w})$$

$$(\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w}) = \underbrace{\begin{bmatrix} y_1 - \boldsymbol{x}_1^T\boldsymbol{w} \\ y_2 - \boldsymbol{x}_2^T\boldsymbol{w} \\ \vdots \\ y_n - \boldsymbol{x}_n^T\boldsymbol{w} \end{bmatrix}^T}_{1 \times n} \underbrace{\begin{bmatrix} y_1 - \boldsymbol{x}_1^T\boldsymbol{w} \\ y_2 - \boldsymbol{x}_2^T\boldsymbol{w} \\ \vdots \\ y_n - \boldsymbol{x}_n^T\boldsymbol{w} \end{bmatrix}}_{n \times 1} = \sum_{i=1}^{n} \left(y_i - \boldsymbol{w}^T\boldsymbol{x}_i\right)^2$$

That seemed like magic. Show me how it works.

$$SSE(\boldsymbol{w}) = (\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w})$$

1) Take partial derivative w.r.t each parameter. But we have d of them.
Take the gradient (vector of partial derivatives)

$$\nabla SSE(w) = \begin{bmatrix} \frac{\delta SSE(\boldsymbol{w})}{\delta w_1} \\ \vdots \\ \frac{\delta SSE(\boldsymbol{w})}{\delta w_d} \end{bmatrix} = 2X^T(\boldsymbol{y} - X\boldsymbol{w})$$

Matrix cookbook is a good resource to help you with doing things like matrix calculus.

That seemed like magic. Show me how it works.

$$SSE(\boldsymbol{w}) = (\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w})$$

2) Set equal to zero vector and solve:

$$2X^T(\boldsymbol{y} - X\boldsymbol{w}) = \vec{\boldsymbol{0}}$$

$$\Longrightarrow X^T X\boldsymbol{w} = X^T\boldsymbol{y}$$

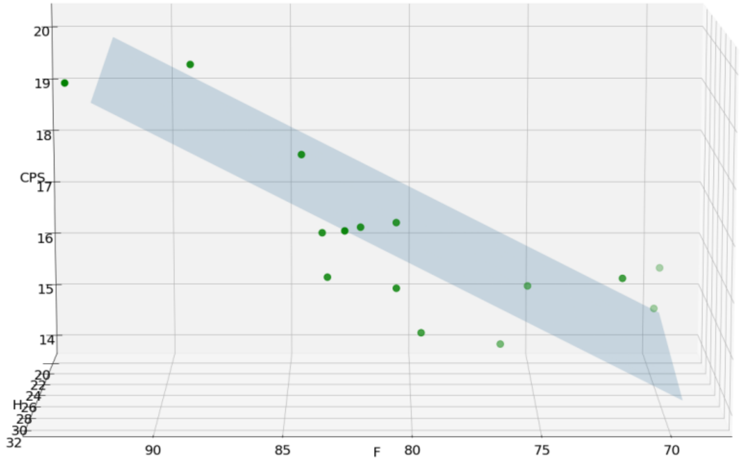
$$\Longrightarrow \boldsymbol{w}^* = (X^T X)^{-1}X^T\boldsymbol{y}$$

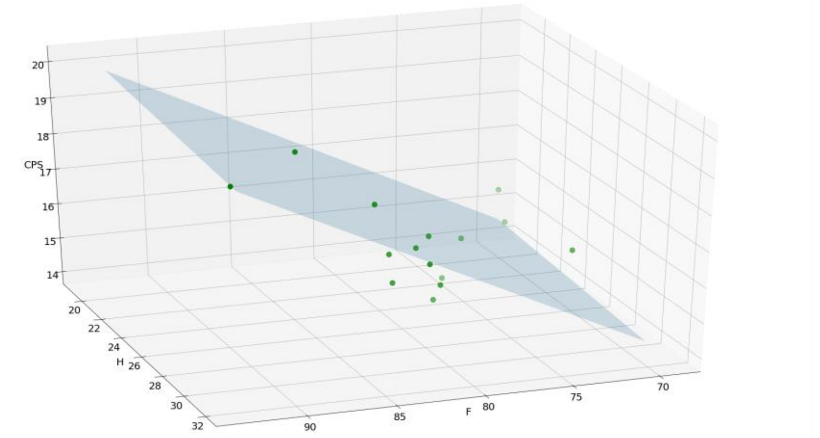
X			Y
Dummy	**F**	**H**	**CPS**
1	88.6	30	20
1	71.6	28	16
1	93.3	32	19.8
1	84.3	30	18.4
1	80.6	29	17.1
1	75.2	24	15.5
1	69.7	20	14.7
1	82	30	17.1
1	69.4	19	15.4
1	83.3	30	16.2
1	79.6	28	15
1	82.6	32	17.2
1	80.6	30	16.
1	83.5	30	17
1	76.3	24	14.4

$X^T X =$ [[3166.96573273 -55.56717431 46.7770757]
[-4477.28836672 106.98440986 -147.32249273]
[1309.51468408 -51.18581341 100.50713299]]

$\boldsymbol{w}^*$ =[-0.80794992, 0.23142213,-0.03828404]

$$y = -0.808 + \ 0.231 * F - 0.038 * H$$





CS434 – ML + DM

Today's Learning Objectives

Be able to answer:

- ~~What is linear regression?~~
- ~~How do we find a solution for it?~~
  - ~~With one dimensional input?~~
  - ~~With multidimensional input?~~
    - ~~We'll need to review some linear algebra~~

- How does this relate to the MLE stuff we were doing? Hint: Gaussians!

- What are some implications of this relationship?

Question Break!

Finding x that maximizes f(x) has the same result as finding x′ that minimizes –f(x)

$$\underset{x}{\operatorname{argmax}} f(x) = \underset{x}{\operatorname{argmin}} -f(x)$$

A True

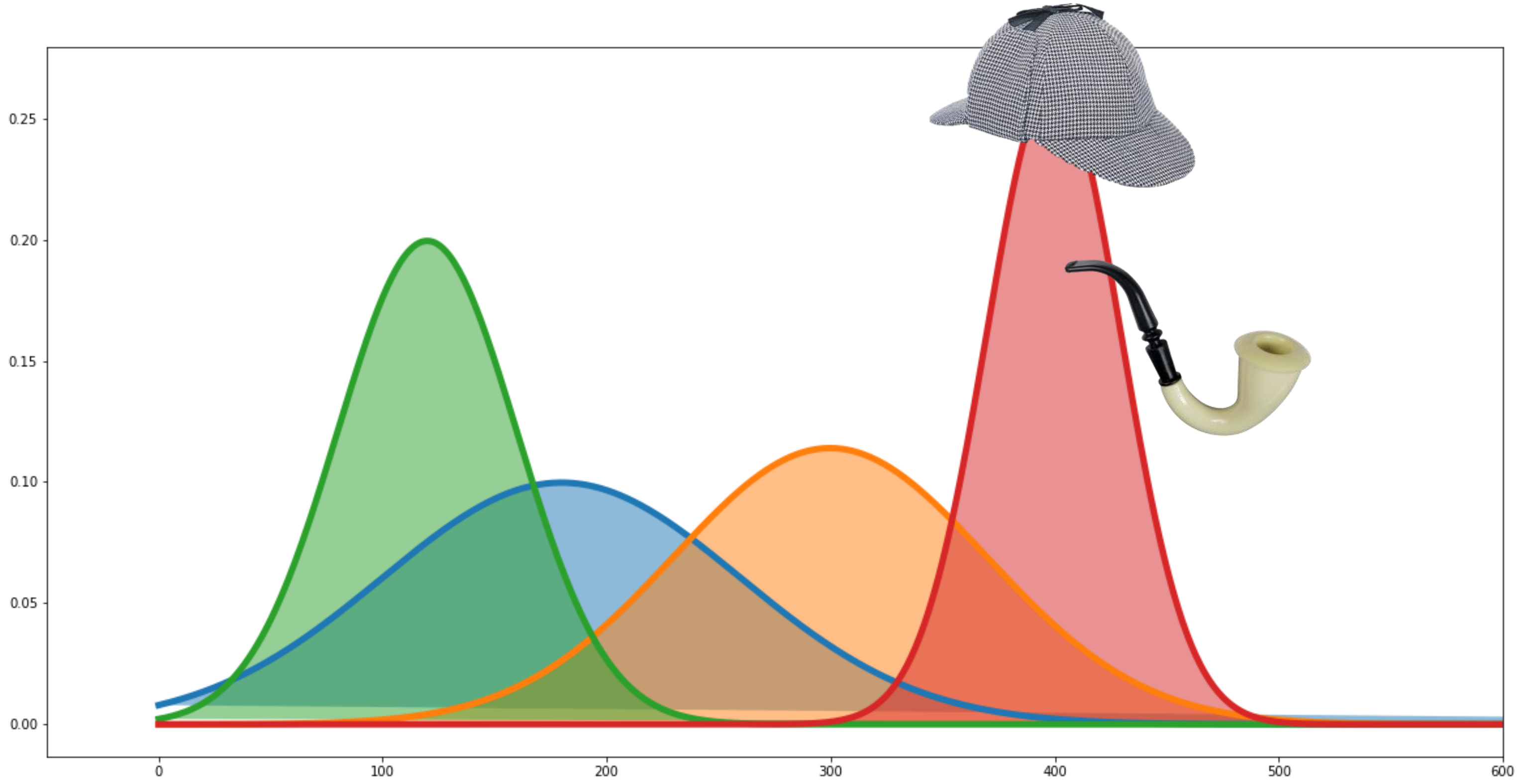
B False

C Duh.

D Bruh.

You: Linear regression has nothing to do with probability or MLE/MAP. You just wanted to torture us last class. *I learned for no good reason!*

Me:



Linear regression from a probabilistic viewpoint:

To do all our MLE/MAP stuff, we need to assume some model of how the data was generated. Before, we only had x's. Now we have x's and y's.

Linear regression from a probabilistic viewpoint:

To do all our MLE/MAP stuff, we need to assume some model of how the data was generated. Before, we only had x’s. Now we have x’s and y’s.

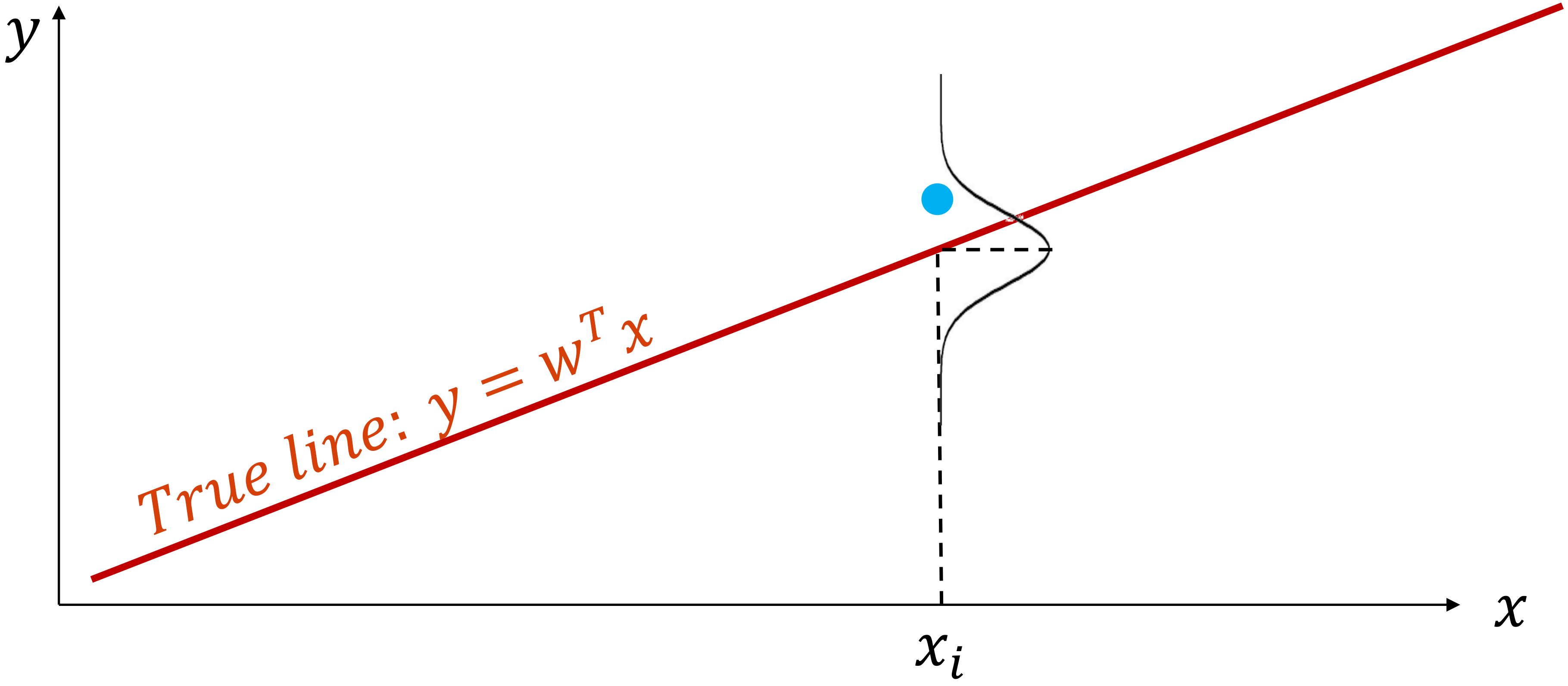
Conditional Model: We can still talk about conditional probabilities of y given x – i.e. P(y|x).

In this case, lets assume there is a linear relationships between x and y that is corrupted by a bit of Gaussian noise with mean 0 and some unknown-but-constant variance.

$$y_i = w^T x_i + \mathcal{N}(0, \sigma)$$

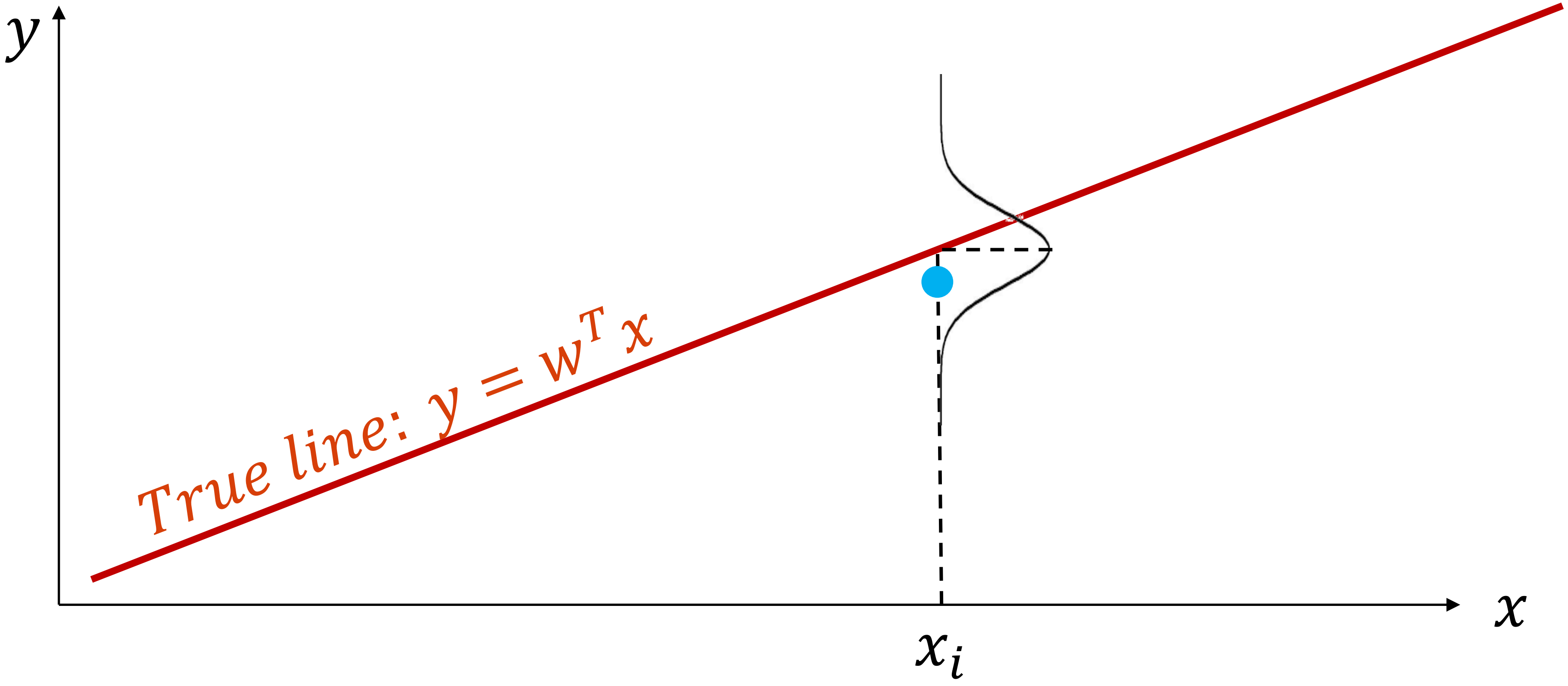
Our "generative story" for this data:

$$y_i = w^T x_i + \mathcal{N}(0, \sigma)$$



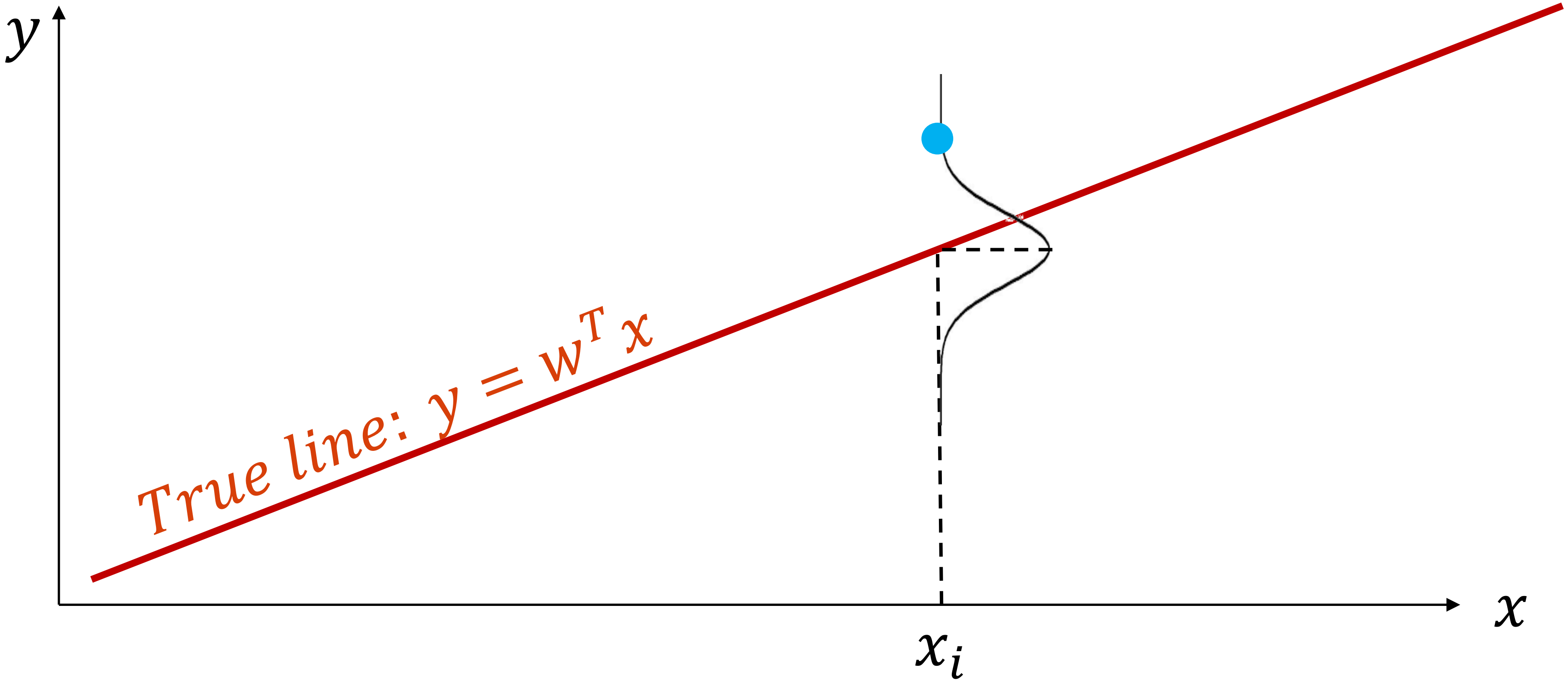
Our "generative story" for this data:

$$y_i = w^T x_i + \mathcal{N}(0, \sigma)$$



Our "generative story" for this data:

$$y_i = w^T x_i + \mathcal{N}(0, \sigma)$$



Okay. Still not seeing why we care? Let's start doing MLE.

Dataset: Given a dataset $D = \{(x_i, y_i)\}_{i=1}^{N}$ assume the above conditional probability.

Model assumption: $y_i = w^T x_i + \mathcal{N}(0, \sigma) \implies P(y_i | x_i, w) = \mathcal{N}(w^T x_i, \sigma)$

Write out likelihood of the training data as a function of parameters θ:

$$\mathcal{L}(\theta) = P(D \mid \theta) = \prod_{i=1}^{N} P(y_i \mid x_i, w) = \prod_{i=1}^{N} \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(y_i - \boldsymbol{w}^T \boldsymbol{x}_i)^2}{2\sigma^2}}$$

Yikes. Let's apply a log and write the log-likelihood to clean this up:

$$\mathcal{LL}(\theta) = P(D \mid \theta) = N \log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{1}{2\sigma^2} \sum_{i=1}^{N} (y_i - w^T x_i)^2$$

Let's view this from a different angle

Wait… this part is looking familiar…

$$\mathcal{LL}(\boldsymbol{\theta}) = N\log\left(\frac{1}{\sigma\sqrt{2\pi}}\right) - \frac{1}{2\sigma^2}\sum_{i=1}^{N}(y_i - \boldsymbol{w}^T\boldsymbol{x_i})^2$$

How do we maximize this log-likelihood with respect to w?

Find w^* such that $\sum_{i=1}^{N}(y_i - \boldsymbol{w}^T\boldsymbol{x_i})^2$ is as small as possible

AKA find weights that minimize the sum of squared error!

Linear regression is just MLE of a linear model with Gaussian noise!

I can choose a different noise model to arrive at different linear regression algorithms.

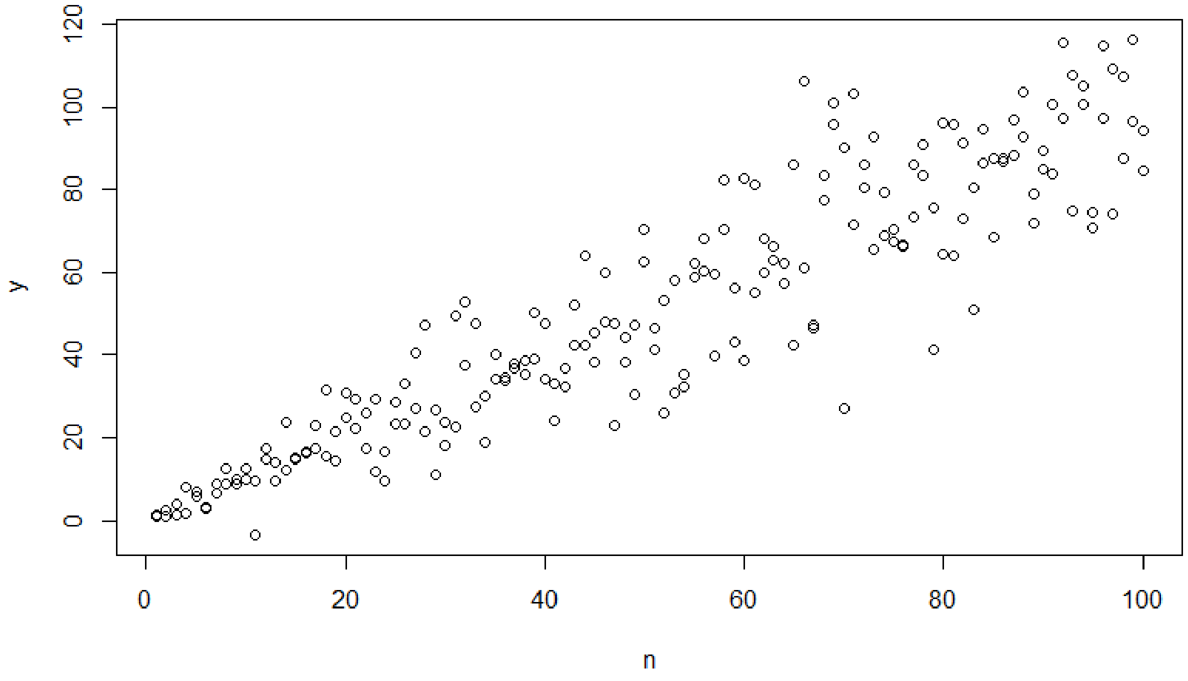
$$y_i = w^T x_i + Laplace(0, \sigma) \qquad Laplace(\mu, \sigma) = \frac{1}{2\sigma} e^{-\frac{|x-\mu|}{\sigma}}$$

Minimizing sum of absolute errors

I can understand and change the assumptions of my model. For instance, no longer assuming constant variance.

$$y_i \sim Normal(w^T x_i, \alpha^T x_i)$$

Correctly model variances that change with x



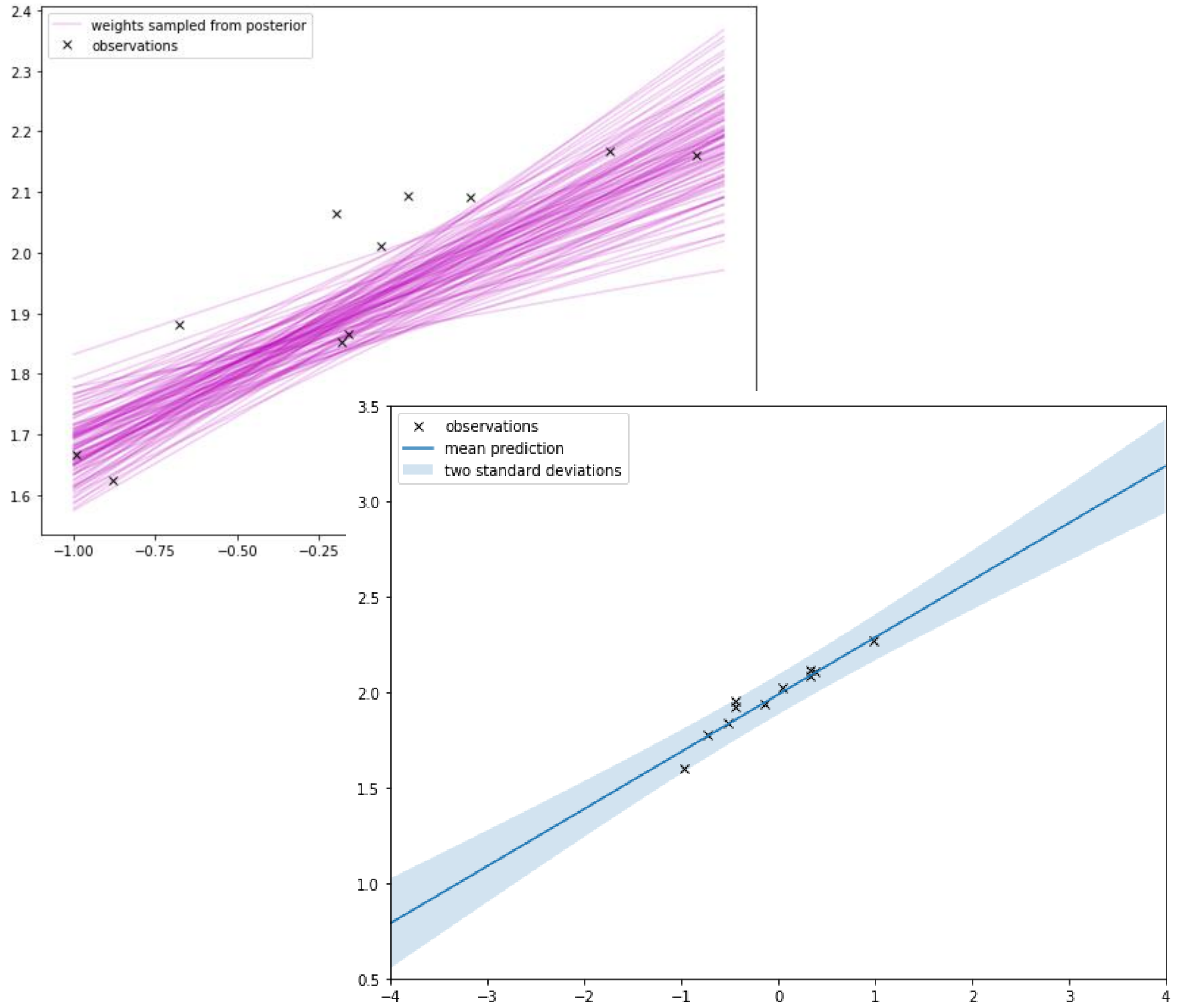
Called heteroscedastic when variance isn't constant

I can apply a Bayesian techniques to reason about uncertainty.

$$w \sim Normal(0, \Sigma)$$

$$y_i \sim Normal(w^T x_i, \sigma)$$

Sample from posterior distribution or compute estimate confidence



I can also use a prior to encode beliefs about what my weights should be, leads to "regularized least squares"

$$w \sim Normal(0, \Sigma) \qquad\qquad y_i \sim Normal(w^T x_i, \sigma)$$

$$\Downarrow$$

$$w^* = \underset{w}{\operatorname{argmin}}\ \lambda||w||_2 + \sum_{i=1}^{n}(y_i - \boldsymbol{w^T x_i})^2$$

Will talk more about regularization next time.

We introduced least-squares linear regression, which assumes the function that maps from **x** to y is linear.

We defined the **sum of squared error** objective function:

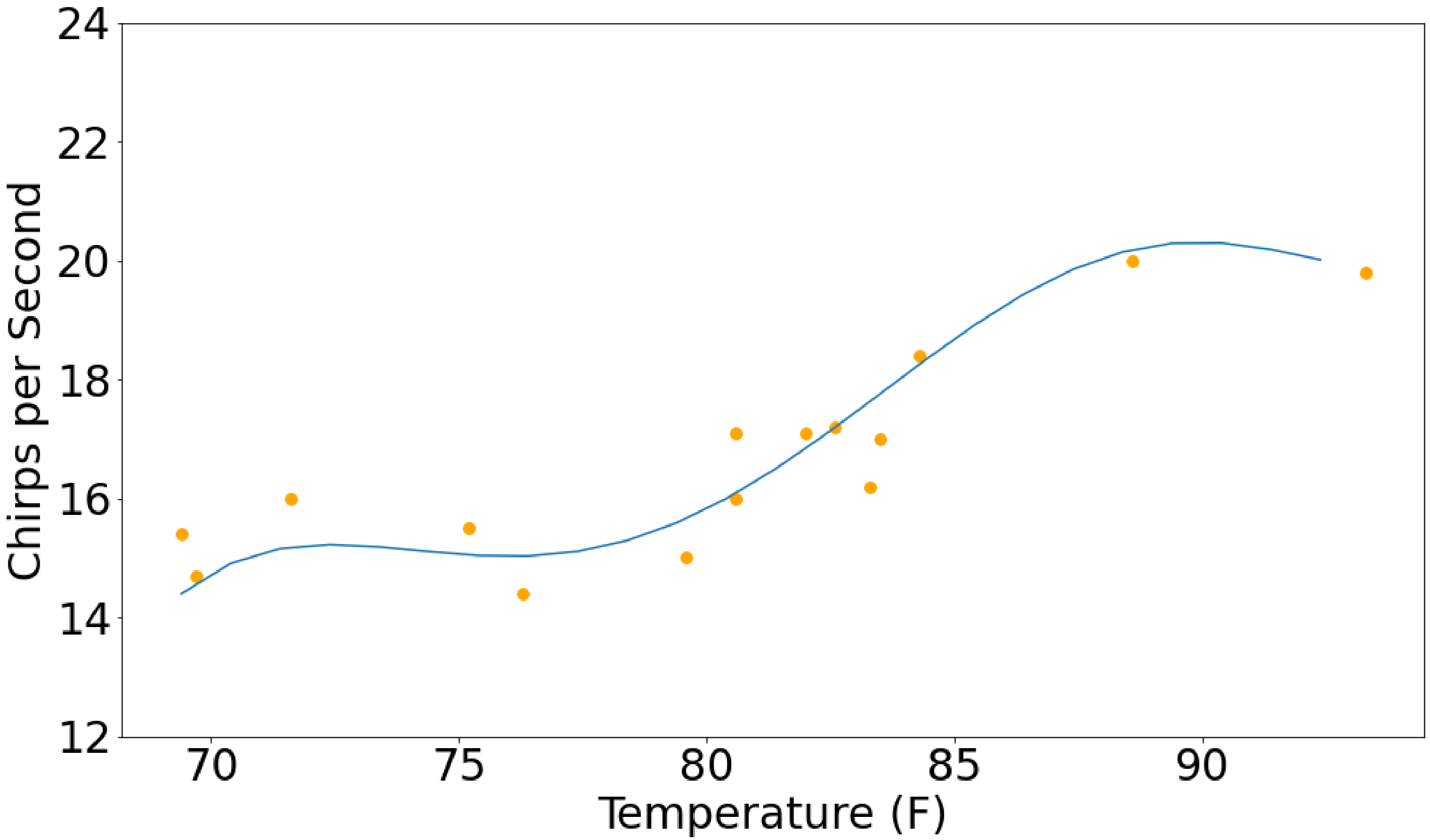
$$SSE(\boldsymbol{w}) = (\boldsymbol{y} - X\boldsymbol{w})^T(\boldsymbol{y} - X\boldsymbol{w})$$

… and converted it into pure matrix/vector operations shown above.

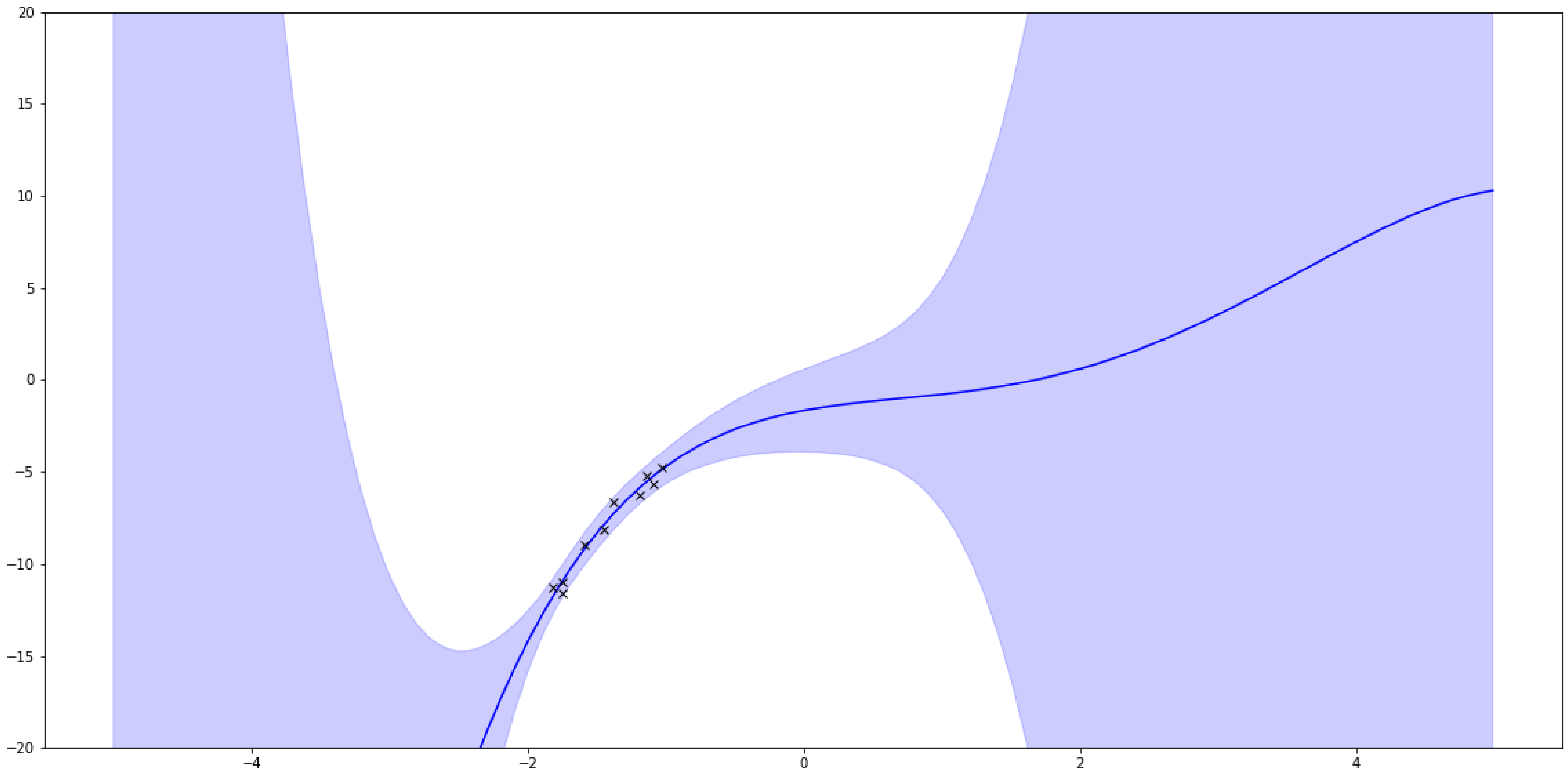
We showed that least-squares linear regression has a probabilistic interpretation as the **MLE of a linear model with Gaussian noise.**

You: It's just fitting straight lines. Boring.

Me: I fit this with least-squares linear regression too. Tune in next time.



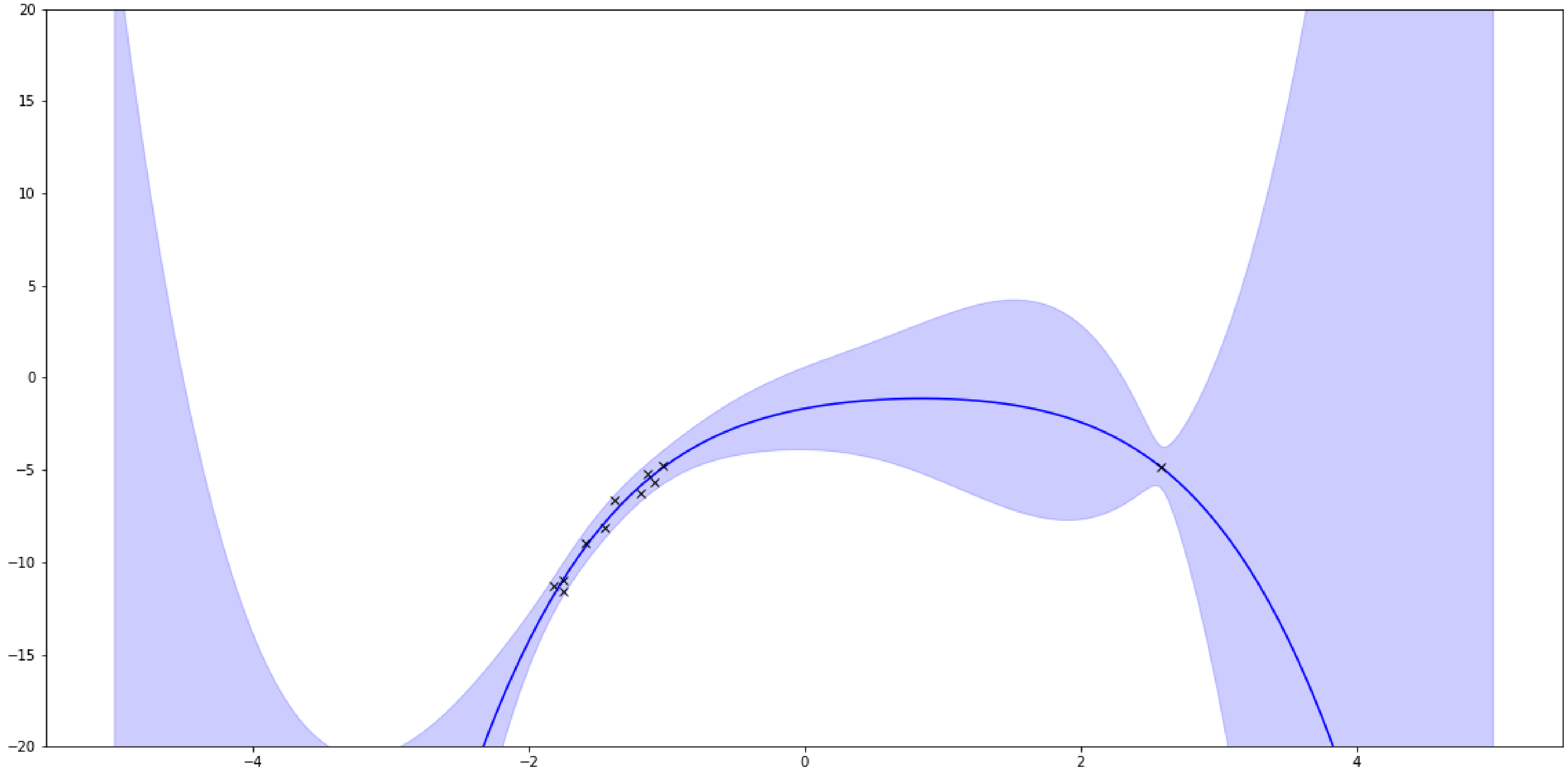
Me: You can combine this with the Bayesian stuff to make really cool fits.



Won't cover in class but the math is only somewhat worse than what we've already done (requires fiddling with multi-dimensional Gaussians)

https://jessicastringham.net/2018/01/10/bayesian-linreg-plots/

Me: You can combine this with the Bayesian stuff to make really cool fits.



Won't cover in class but the math is only somewhat worse than what we've already done (requires fiddling with multi-dimensional Gaussians)

https://jessicastringham.net/2018/01/10/bayesian-linreg-plots/

Today's Learning Objectives

Be able to answer:

- ~~What is linear regression?~~
- ~~How do we find a solution for it?~~
  - ~~With one dimensional input?~~
  - ~~With multidimensional input?~~
    - ~~We'll need to review some linear algebra~~
- ~~How does this relate to the MLE stuff we were doing?~~ ~~Hint: Gaussians!~~
- ~~What are some implications of this relationship?~~

Next time

Next Time: We'll dig a bit deeper into the concept of regularization in linear models. Then we'll move onto linear models for classification.