

Lecture 11

Introduction to Machine Learning

EECS 398: Practical Data Science, Spring 2025

practicaldsc.org • github.com/practicaldsc/sp25 •  See latest announcements [here on Ed](#)

Agenda



- Machine learning and models.
- The constant model.
- Another loss function.

The next few lectures (and Homework 6!) will be primarily **math-based**.

- For these lectures, we'll post blank slides as a PDF before class, and annotated slides after class.
- If there are any code demos, we'll post those before class, too.

Machine learning and models

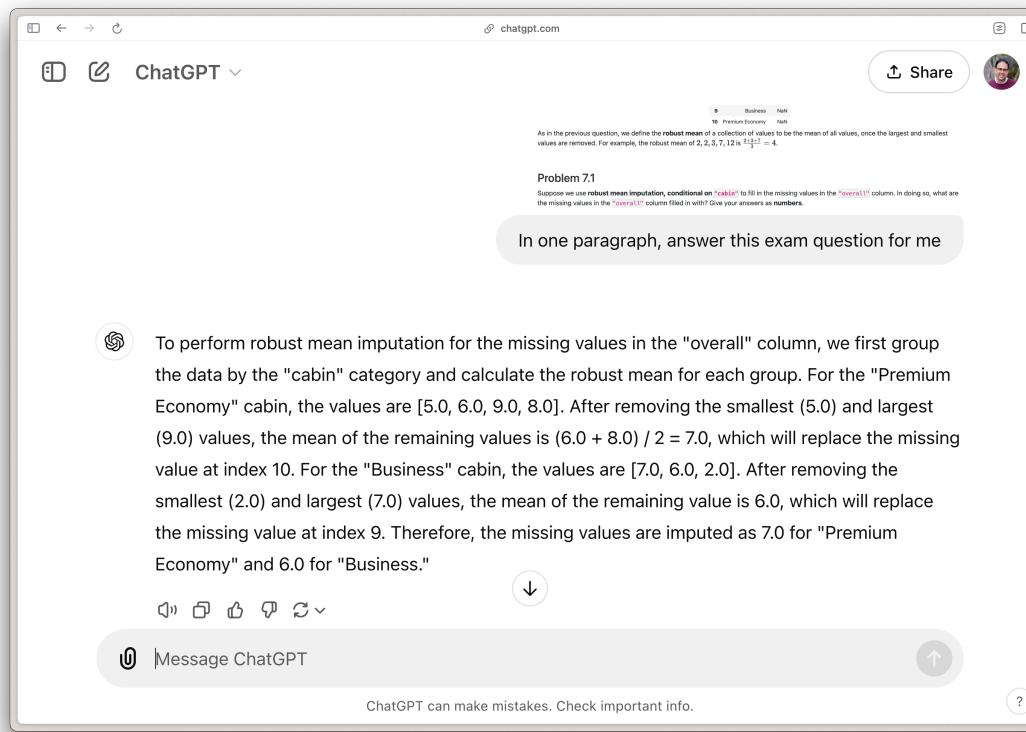
Machine learning is about **automatically** learning patterns from data.

Example: Handwritten digit classification

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
5 5 5 5 5 5 5 5 5 5 5 5 5 5 5
6 6 6 6 6 6 6 6 6 6 6 6 6 6 6
7 7 7 7 7 7 7 7 7 7 7 7 7 7 7
8 8 8 8 8 8 8 8 8 8 8 8 8 8 8
9 9 9 9 9 9 9 9 9 9 9 9 9 9 9

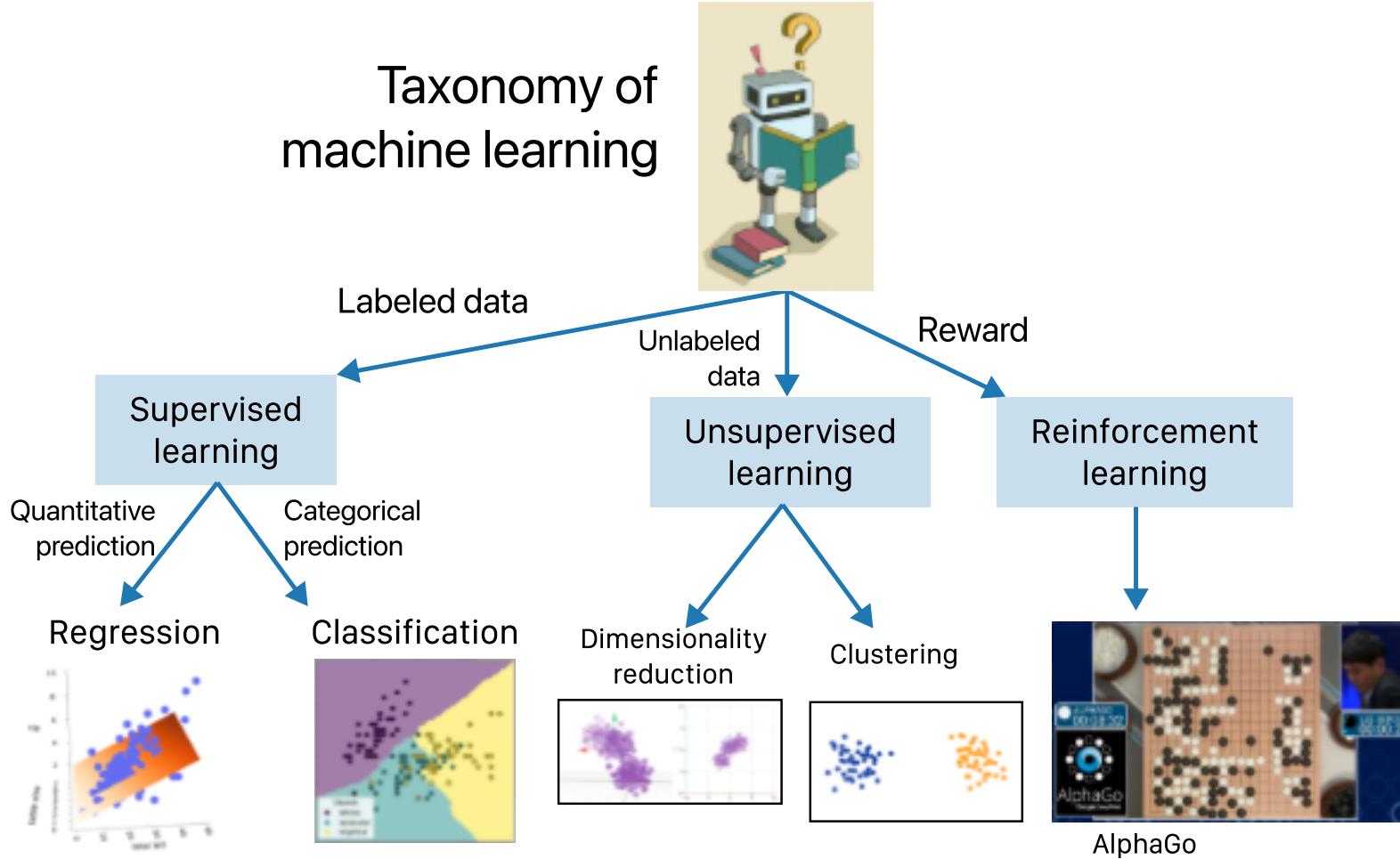
Humans are good at understanding handwriting,
but how do we get computers to understand handwriting?

Example: ChatGPT



How did ChatGPT know how to answer Question 7 from the Fall 2024 Midterm?

Taxonomy of machine learning



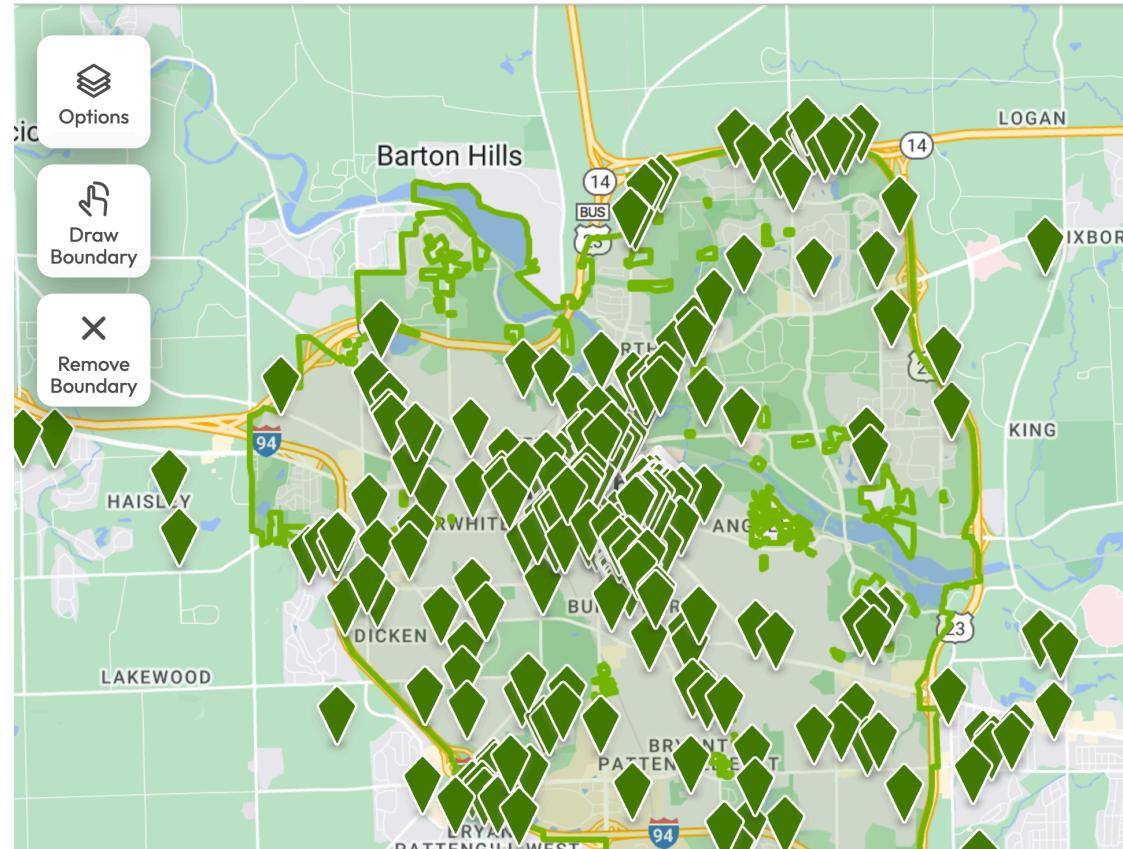
Ann Arbor, MI



Price



Beds/Baths



You might be starting to look for off-campus apartments for next year,
none of which are in your price range.

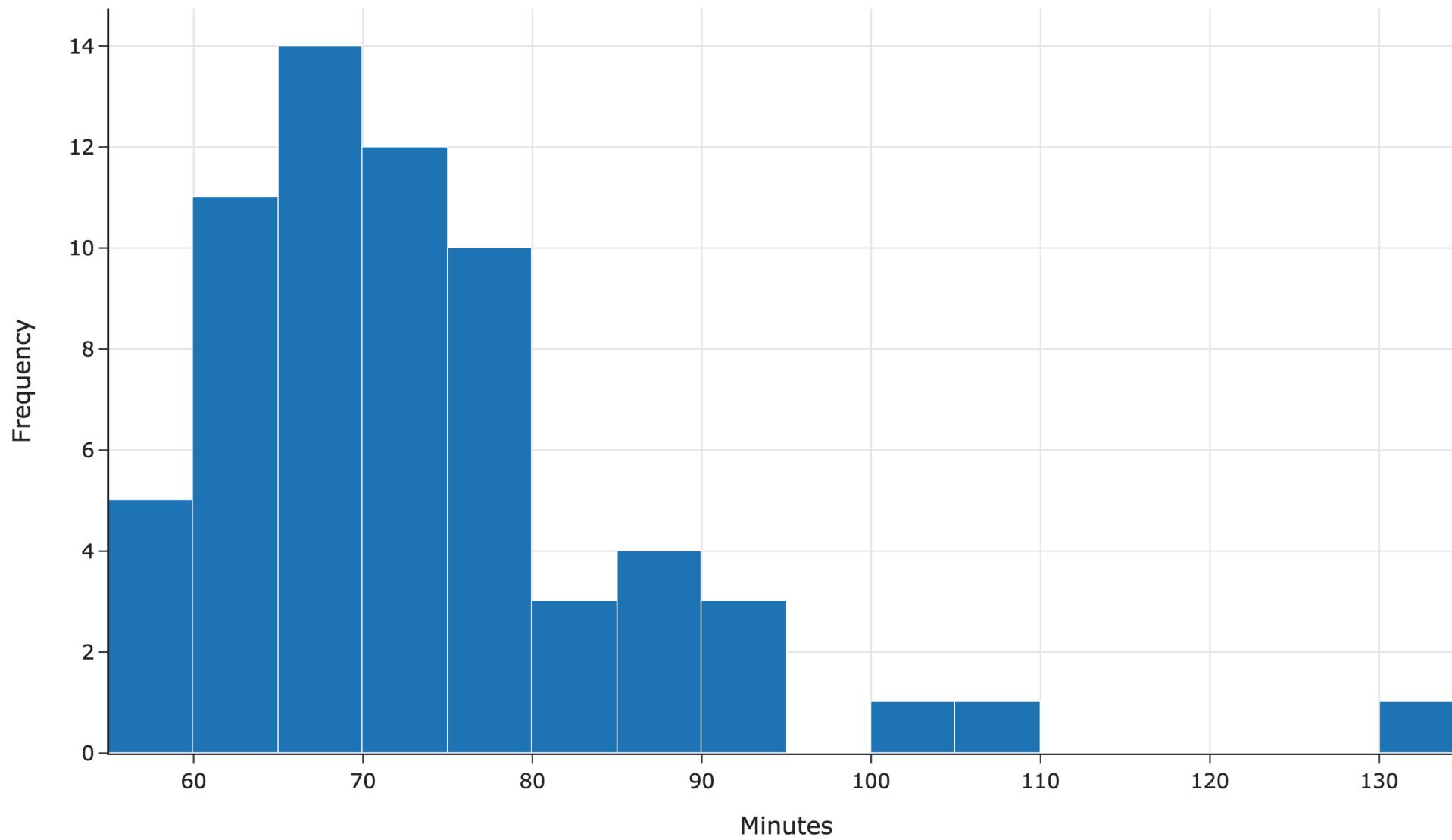
| | date | day | departure_hour | minutes |
|----------|-------------|------------|-----------------------|----------------|
| 0 | 5/22/2023 | Mon | 8.450000 | 63.0 |
| 1 | 9/18/2023 | Mon | 7.950000 | 75.0 |
| 2 | 10/17/2023 | Tue | 10.466667 | 59.0 |
| 3 | 11/28/2023 | Tue | 8.900000 | 89.0 |
| 4 | 2/15/2024 | Thu | 8.083333 | 69.0 |

...

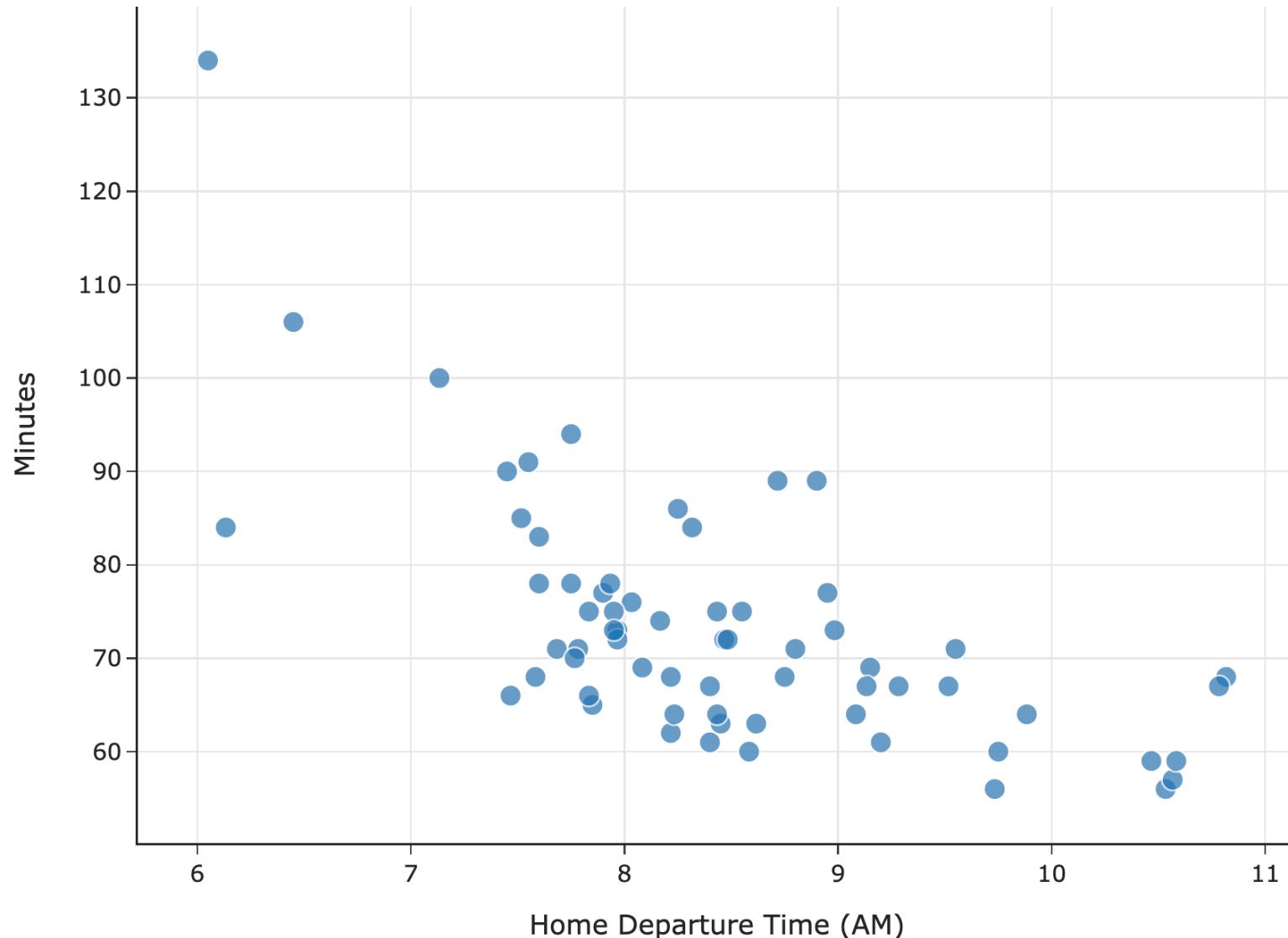
You decide to live with your parents in Detroit and commute.
You keep track of how long it takes you to get to school each day.

This is a real dataset, collected by [Joseph Hearn](#)! However, he lived in the Seattle area, not Metro Detroit.

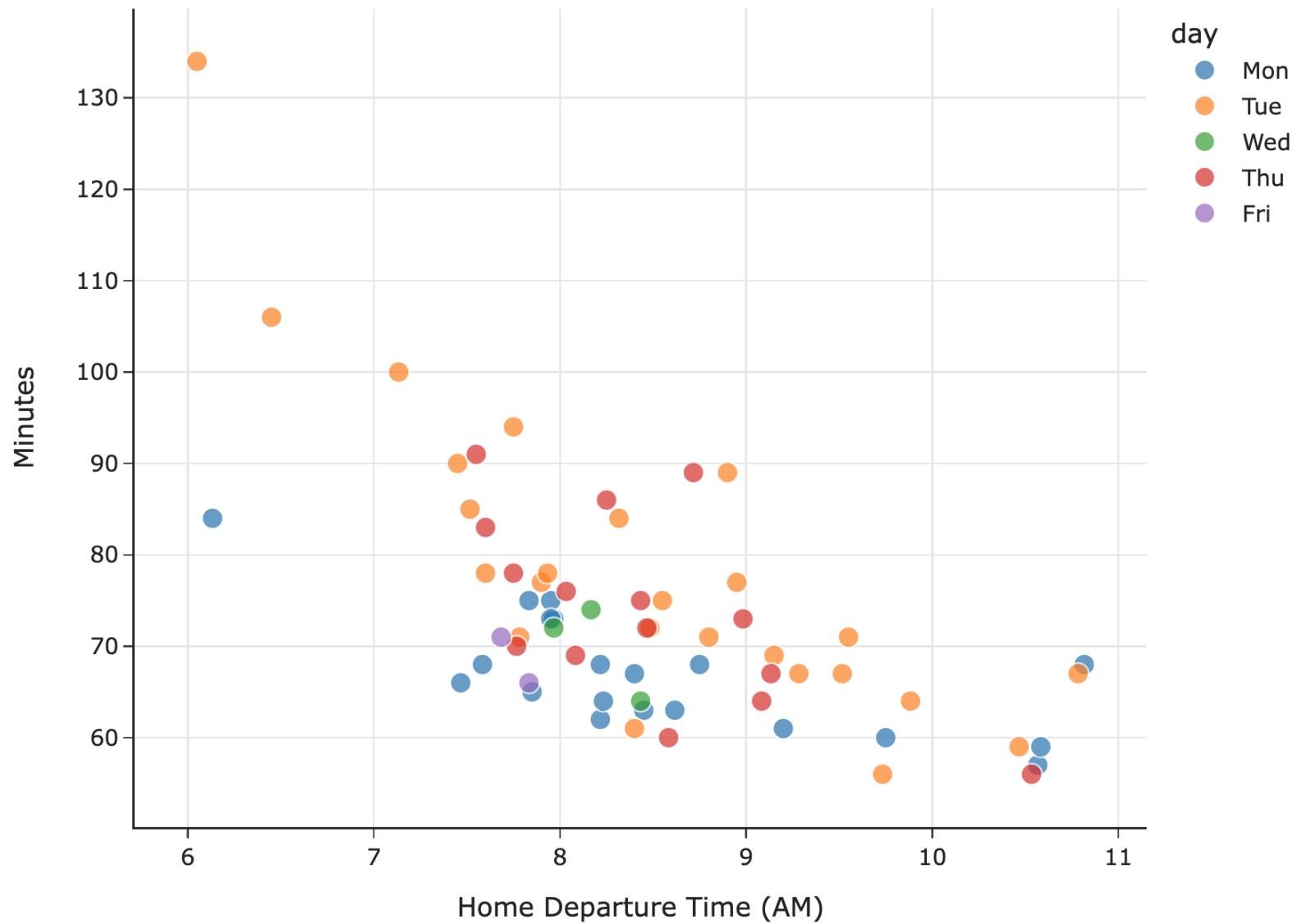
Distribution of Commuting Time



Commuting Time vs. Home Departure Time



Commuting Time vs. Home Departure Time



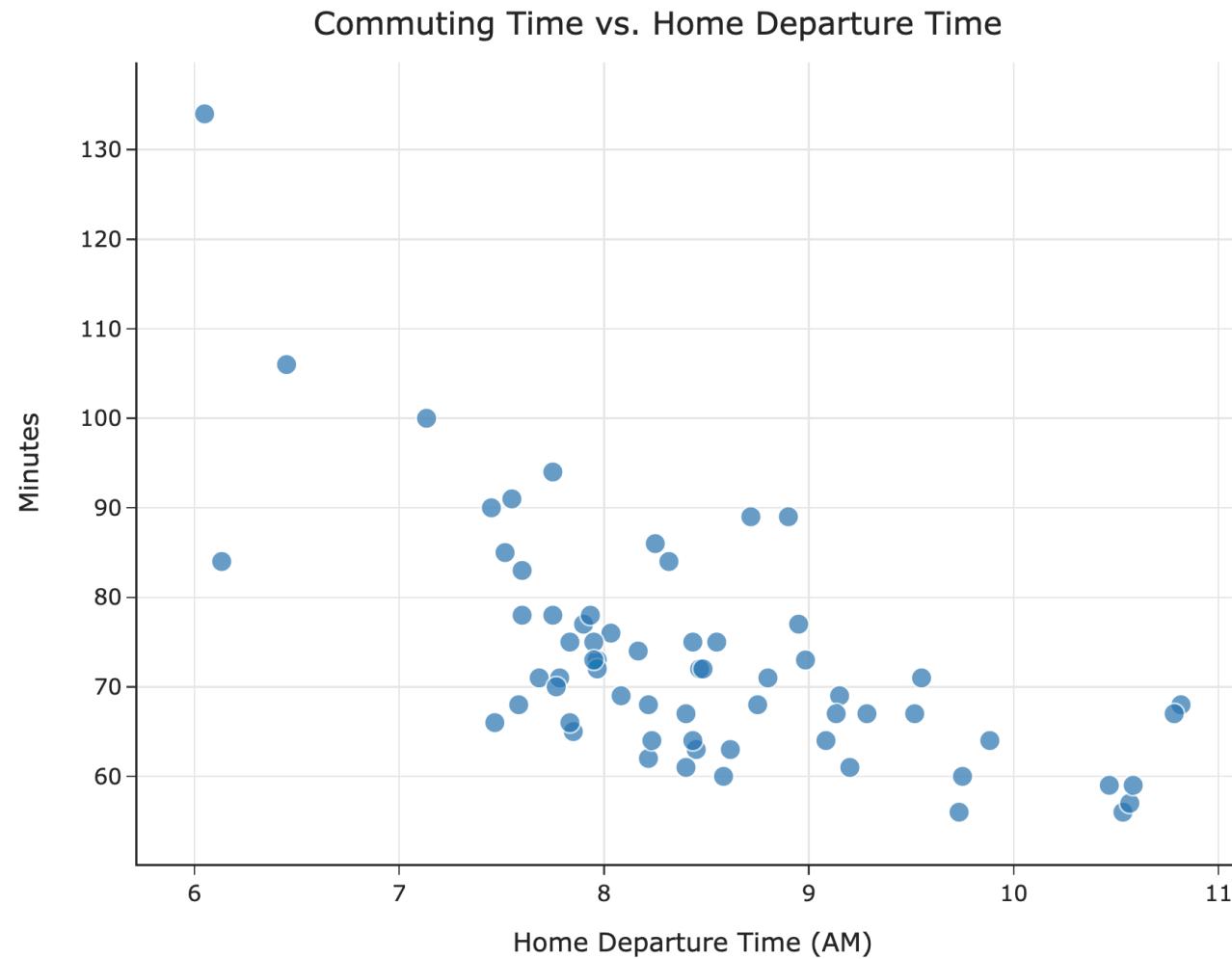
Goal: Predict your **commute time**, i.e. how long it will take to get to school.

This is a **regression** problem.

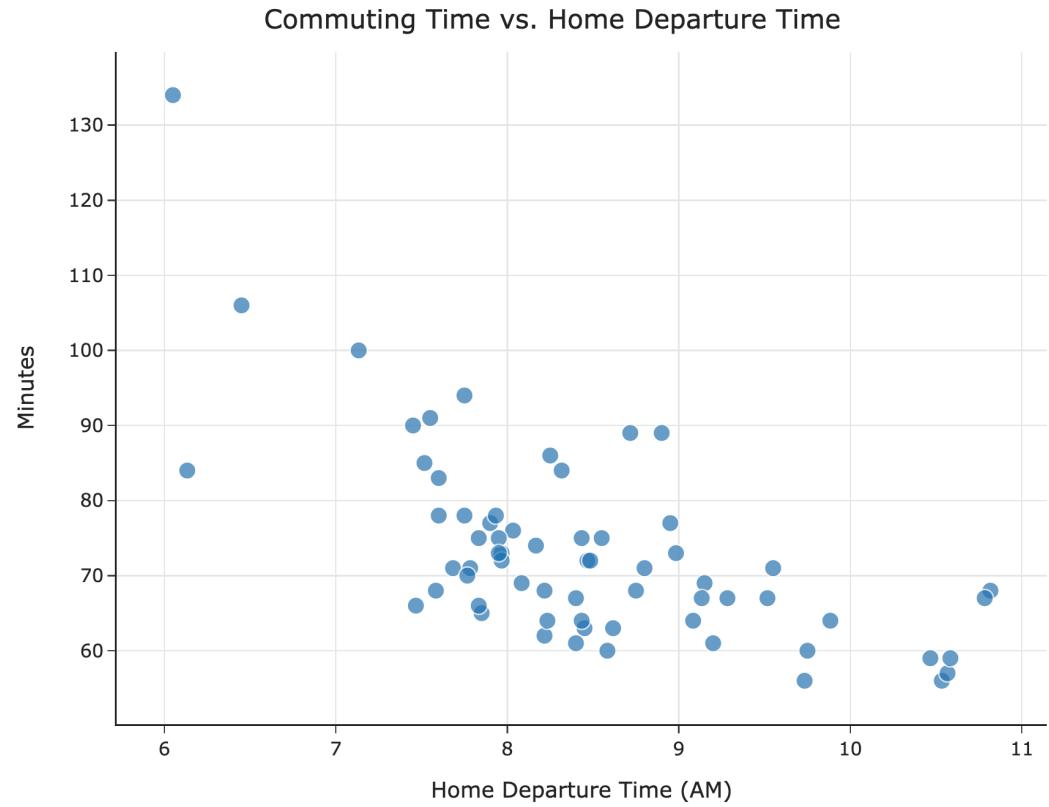
How can we do this? What will we need to assume?

A model is a set of assumptions about how data were generated.

Possible models



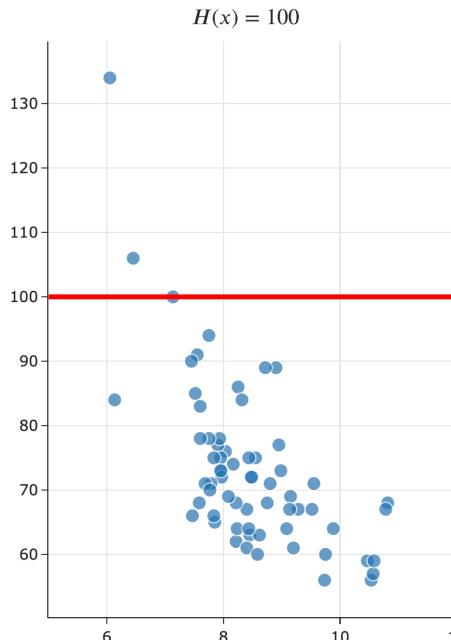
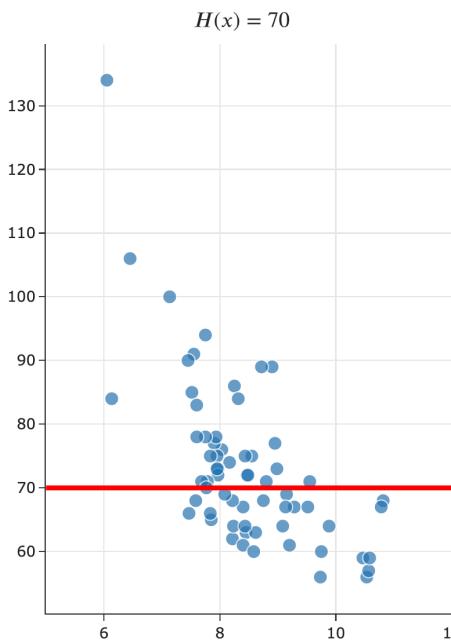
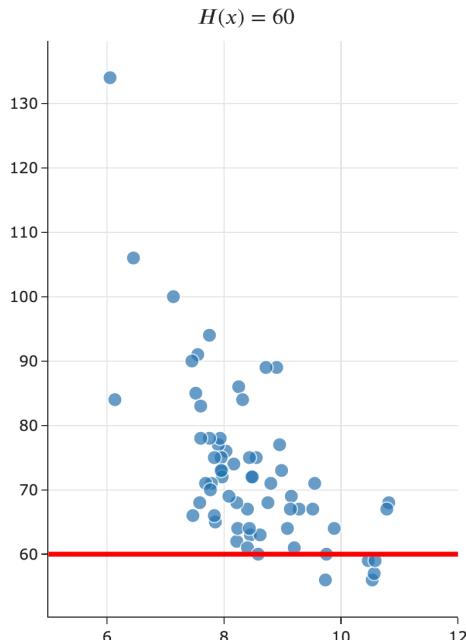
Notation



- x : "input", "independent variable", or "feature".
- y : "response", "dependent variable", or "target".
- The i th observation is denoted (x_i, y_i) .
- **We use x_i s to predict y_i s.**

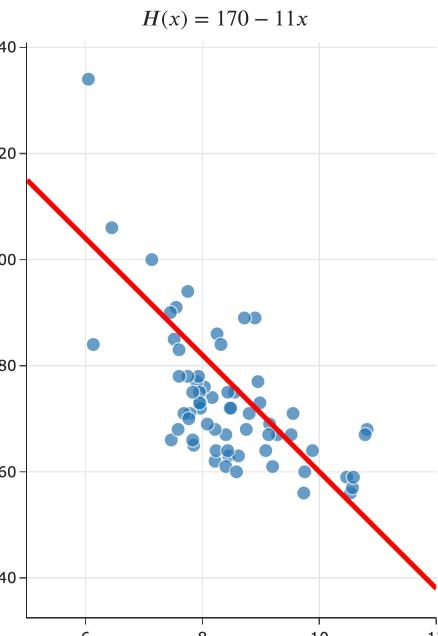
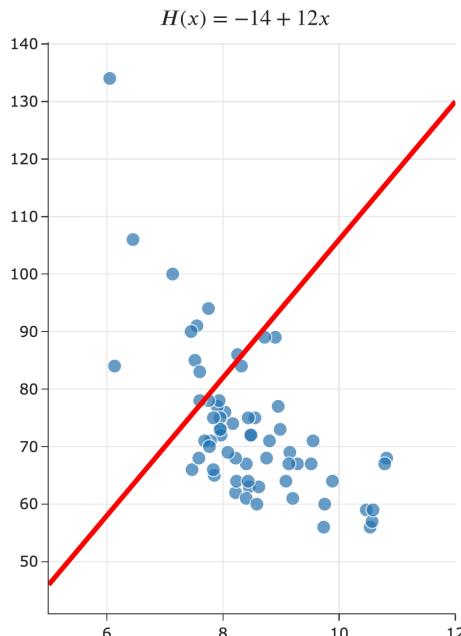
Hypothesis functions and parameters

- A hypothesis function, H , takes in an x_i as input and returns a predicted y_i .
- **Parameters** define the relationship between the input and output of a hypothesis function.
- **Example:** The constant model, $H(x_i) = h$, has one parameter: h .



Hypothesis functions and parameters

- A hypothesis function, H , takes in an x_i as input and returns a predicted y_i .
- **Parameters** define the relationship between the input and output of a hypothesis function.
- **Example:** The simple linear regression model, $H(x_i) = w_0 + w_1 x_i$, has two parameters: w_0 and w_1 .



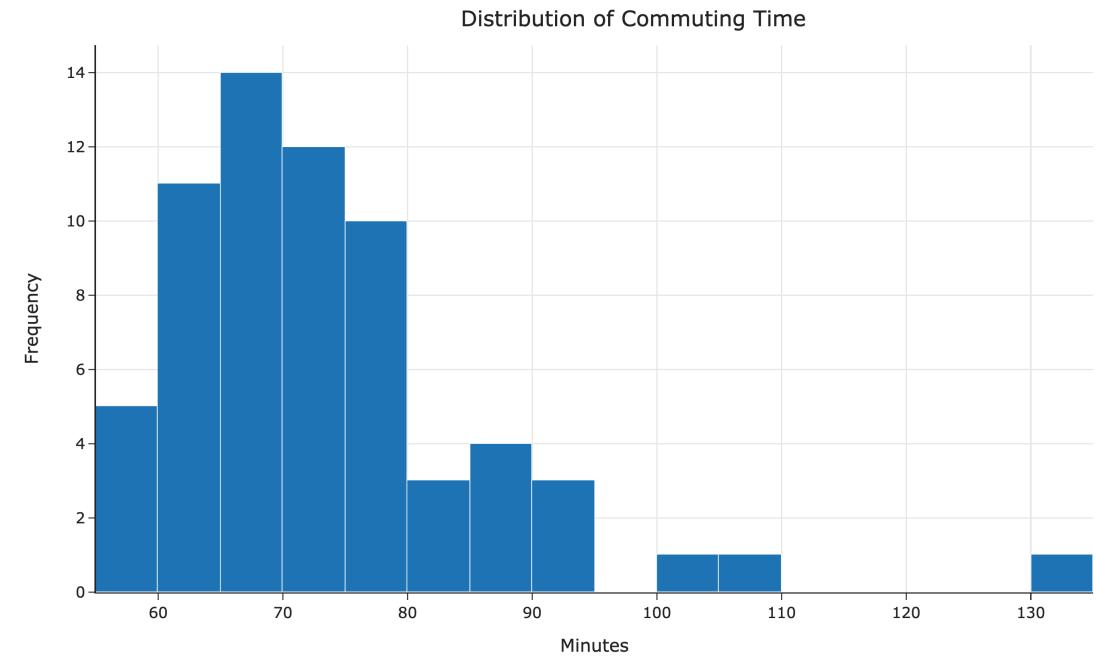
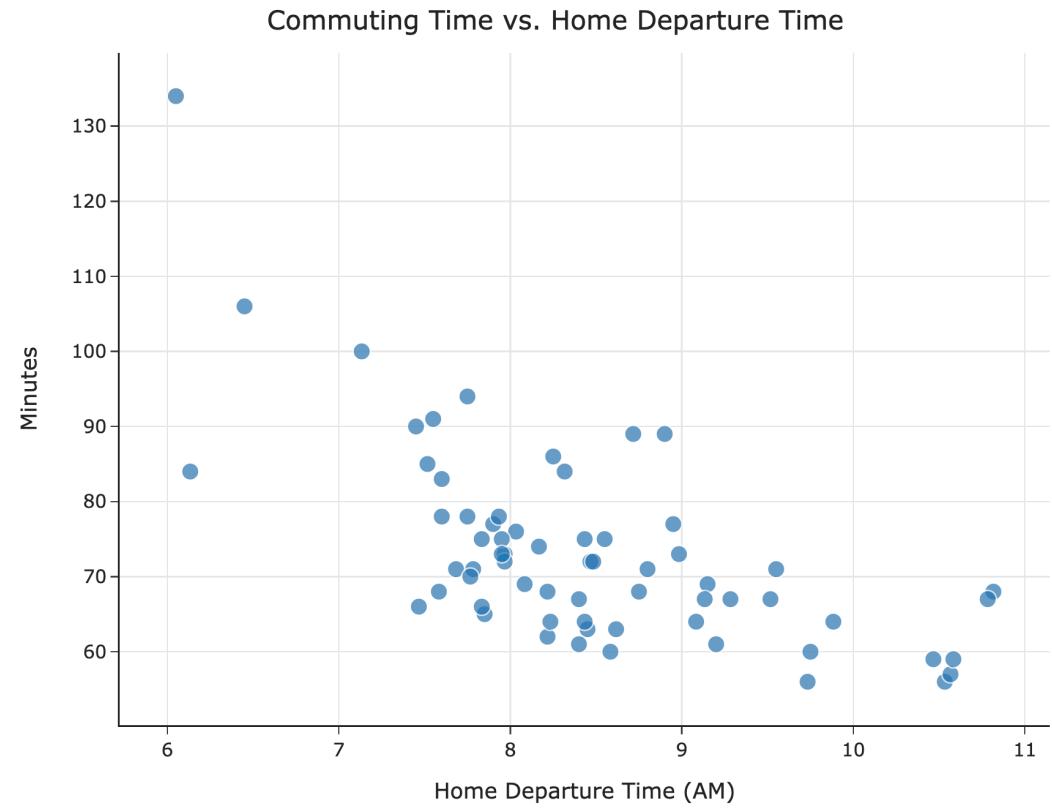
Question 🤔

Answer at practicaldsc.org/q

What questions do you have?

The constant model

The constant model



A concrete example

- Let's suppose we have just a smaller dataset of just five historical commute times in minutes.

$$y_1 = 72$$

$$y_2 = 90$$

$$y_3 = 61$$

$$y_4 = 85$$

$$y_5 = 92$$

- Given this data, can you come up with a prediction for your future commute time?
How?

Some common approaches

- The mean:

$$\frac{1}{5}(72 + 90 + 61 + 85 + 92) = \boxed{80}$$

- The median:

61 72 **85** 90 92

- Both of these are familiar **summary statistics**.

Summary statistics summarize a collection of numbers with a single number, i.e. they result from an **aggregation**.

- But which one is better? Is there a "best" prediction we can make?

The cost of making predictions

- A **loss function** quantifies how bad a prediction is for a single data point.
 - If our prediction is **close** to the actual value, we should have **low** loss.
 - If our prediction is **far** from the actual value, we should have **high** loss.
- A good starting point is error, which is the difference between **actual** and **predicted** values.

$$e_i = \textcolor{blue}{y_i} - \textcolor{orange}{H(x_i)}$$

- Suppose my commute **actually** takes 80 minutes.
 - If I predict 75 minutes:
 - If I predict 72 minutes:
 - If I predict 100 minutes:

Squared loss

- One loss function is squared loss, L_{sq} , which computes $(\text{actual} - \text{predicted})^2$.

$$L_{\text{sq}}(\textcolor{blue}{y}_i, \textcolor{orange}{H}(x_i)) = (\textcolor{blue}{y}_i - \textcolor{orange}{H}(x_i))^2$$

- Note that for the constant model, $H(x_i) = h$, so we can simplify this to:

$$L_{\text{sq}}(\textcolor{blue}{y}_i, \textcolor{orange}{h}) = (\textcolor{blue}{y}_i - \textcolor{orange}{h})^2$$

- Squared loss is not the only loss function that exists!

Soon, we'll learn about absolute loss. Different loss functions have different pros and cons.

A concrete example, revisited

- Consider again our smaller dataset of just five historical commute times in minutes.

$$y_1 = 72$$

$$y_2 = 90$$

$$y_3 = 61$$

$$y_4 = 85$$

$$y_5 = 92$$

- Suppose we predict the median, $h = 85$. What is the squared loss of 85 for each data point?

Averaging squared losses

- We'd like a single number that describes the quality of our predictions across our entire dataset. One way to compute this is as the **average of the squared losses**.
- For the median, $h = 85$:

$$\frac{1}{5} ((72 - 85)^2 + (90 - 85)^2 + (61 - 85)^2 + (85 - 85)^2 + (92 - 85)^2) = \boxed{163.8}$$

- For the mean, $h = 80$:

$$\frac{1}{5} ((72 - 80)^2 + (90 - 80)^2 + (61 - 80)^2 + (85 - 80)^2 + (92 - 80)^2) = \boxed{138.8}$$

- Which prediction is better? Could there be an even better prediction?

Mean squared error

- Another term for average squared loss is mean squared error (MSE).
- The mean squared error on our smaller dataset for any prediction h is of the form:

$$R_{\text{sq}}(h) = \frac{1}{5} ((72 - h)^2 + (90 - h)^2 + (61 - h)^2 + (85 - h)^2 + (92 - h)^2)$$

R stands for "risk", as in "empirical risk." We'll see this term again soon.

- For example, if we predict $h = 100$, then:

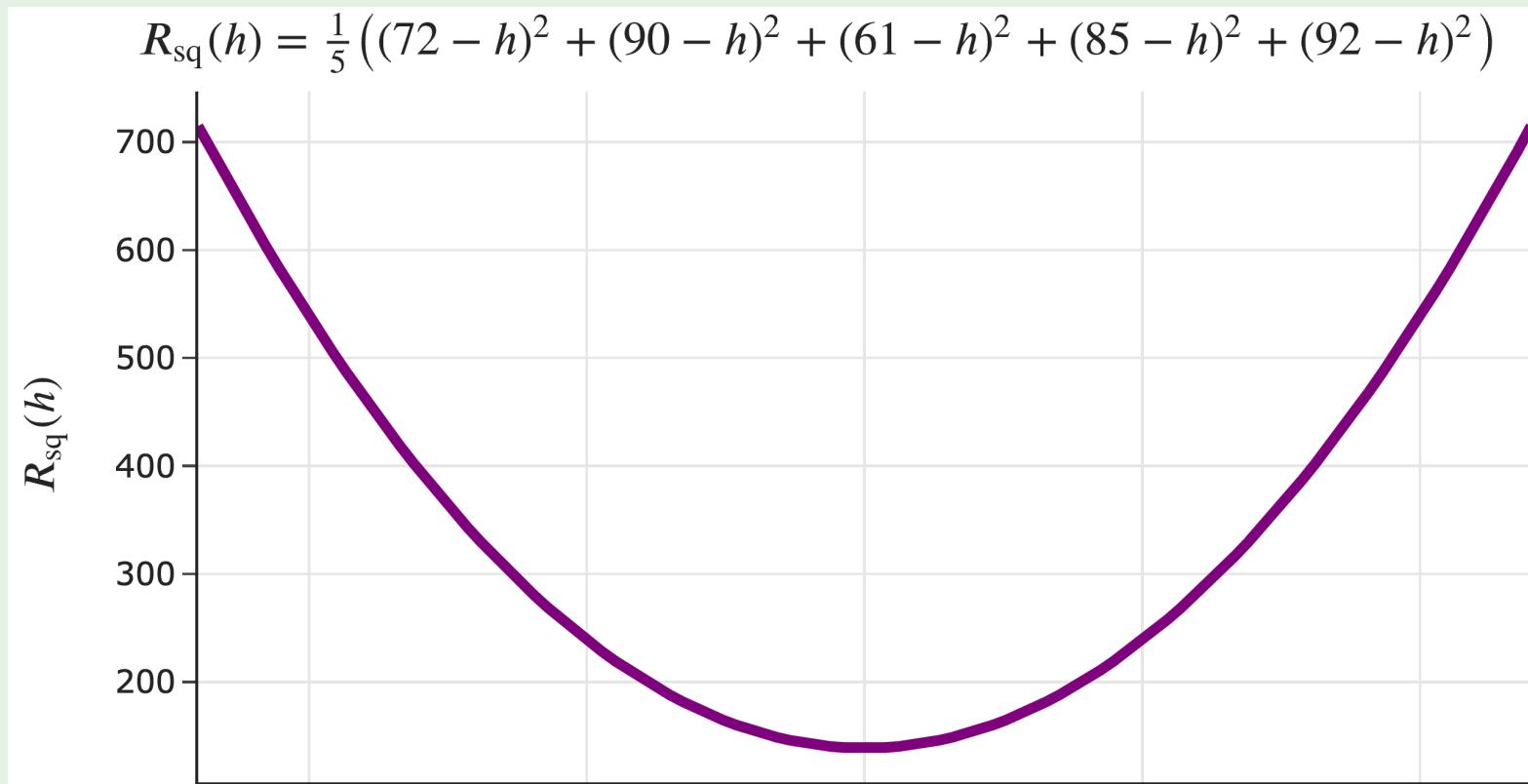
$$\begin{aligned} R_{\text{sq}}(100) &= \frac{1}{5} ((72 - 100)^2 + (90 - 100)^2 + (61 - 100)^2 + (85 - 100)^2 + (92 - 100)^2) \\ &= \boxed{538.8} \end{aligned}$$

- We can pick any h as a prediction, but the smaller $R_{\text{sq}}(h)$ is, the better h is!

Activity

Answer at practicaldsc.org/q (use the free response box!)

$$R_{\text{sq}}(h) = \frac{1}{5} ((72 - h)^2 + (90 - h)^2 + (61 - h)^2 + (85 - h)^2 + (92 - h)^2)$$



Which h corresponds to the vertex of $R_{\text{sq}}(h)$?

Mean squared error, in general

- Suppose we collect n commute times, y_1, y_2, \dots, y_n .
- The mean squared error of the prediction h is:
- Or, using **summation notation**:

The best prediction

$$R_{\text{sq}}(h) = \frac{1}{n} \sum_{i=1}^n (y_i - h)^2$$

- We want the **best** constant prediction, among all constant predictions h .
- The smaller $R_{\text{sq}}(h)$ is, the better h is.
- **Goal:** Find the h that minimizes $R_{\text{sq}}(h)$.
The resulting h will be called h^* .
- **How do we find h^* ?**

Minimizing mean squared error using calculus

Minimizing using calculus

- We'd like to minimize:

$$R_{\text{sq}}(h) = \frac{1}{n} \sum_{i=1}^n (y_i - h)^2$$

- In order to minimize $R_{\text{sq}}(h)$, we:
 1. take its derivative with respect to h ,
 2. set it equal to 0,
 3. solve for the resulting h^* , and
 4. perform a second derivative test to ensure we found a minimum.
- $R_{\text{sq}}(h)$ is an example of an **objective function**, a function that needs to be minimized.

Step 0: The derivative of $(y_i - h)^2$

- Remember from calculus that:
 - if $c(x) = a(x) + b(x)$, then
 - $\frac{d}{dx}c(x) = \frac{d}{dx}a(x) + \frac{d}{dx}b(x)$.
- This is relevant because $R_{\text{sq}}(h) = \frac{1}{n} \sum_{i=1}^n (y_i - h)^2$ involves the sum of n individual terms, each of which involve h .
- So, to take the derivative of $R_{\text{sq}}(h)$, we'll first need to find the derivative of $(y_i - h)^2$.

$$\frac{d}{dh}(y_i - h)^2 =$$

Question 🤔

Answer at practicaldsc.org/q

$$R_{\text{sq}}(h) = \frac{1}{n} \sum_{i=1}^n (y_i - h)^2$$

Which of the following is $\frac{d}{dh} R_{\text{sq}}(h)$?

- A. 0
- B. $\sum_{i=1}^n y_i$
- C. $\frac{1}{n} \sum_{i=1}^n (y_i - h)$
- D. $\frac{2}{n} \sum_{i=1}^n (y_i - h)$
- E. $-\frac{2}{n} \sum_{i=1}^n (y_i - h)$

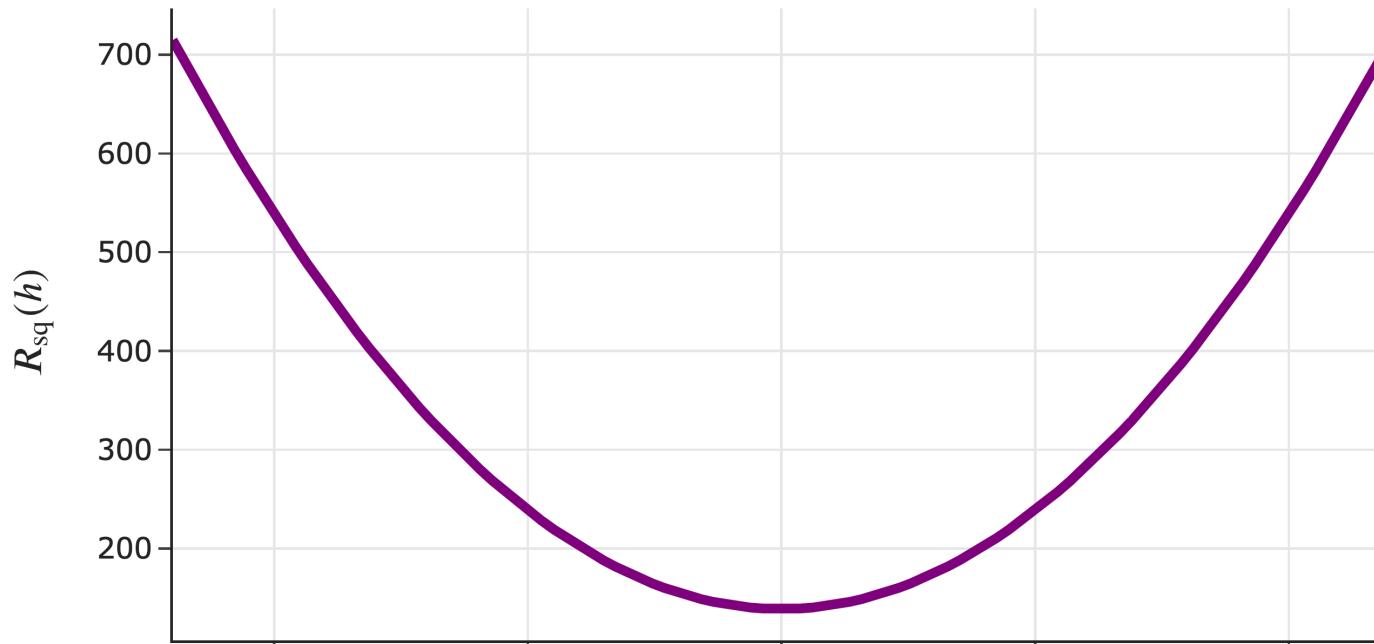
Step 1: The derivative of $R_{\text{sq}}(h)$

$$\frac{d}{dh} R_{\text{sq}}(h) = \frac{d}{dh} \left(\frac{1}{n} \sum_{i=1}^n (y_i - h)^2 \right)$$

Steps 2 and 3: Set to 0 and solve for the minimizer, h^*

Step 4: Second derivative test

$$R_{\text{sq}}(h) = \frac{1}{5}((72 - h)^2 + (90 - h)^2 + (61 - h)^2 + (85 - h)^2 + (92 - h)^2)$$



We already saw that $R_{\text{sq}}(h)$ is **convex**, i.e. that it opens upwards, so the h^* we found must be a minimum, not a maximum.

The mean minimizes mean squared error!

- The problem we set out to solve was, find the h^* that minimizes:

$$R_{\text{sq}}(h) = \frac{1}{n} \sum_{i=1}^n (y_i - h)^2$$

- The answer is:

$$h^* = \text{Mean}(y_1, y_2, \dots, y_n)$$

- The **best constant prediction**, in terms of mean squared error, is always the **mean**.
- We call h^* our **optimal model parameter**, for when we use:
 - the constant model, $H(x_i) = h$, and
 - the squared loss function, $L_{\text{sq}}(y_i, h) = (y_i - h)^2$.

Aside: Terminology

- Another way of writing:

h^* is the value of h that minimizes $\frac{1}{n} \sum_{i=1}^n (y_i - h)^2$

is:

$$h^* = \operatorname{argmin}_h \left(\frac{1}{n} \sum_{i=1}^n (y_i - h)^2 \right)$$

- h^* is the solution to an **optimization problem**, where the objective function is $R_{\text{sq}}(h) = \frac{1}{n} \sum_{i=1}^n (y_i - h)^2$.

The modeling recipe

- We've implicitly introduced a three-step process for finding optimal model parameters (like h^*) that we can use for making predictions:
 1. Choose a model.
 2. Choose a loss function.
 3. Minimize average loss to find optimal model parameters.
- Most modern machine learning methods today, including neural networks, follow this recipe, and we'll see it repeatedly this semester!

Question 🤔

Answer at practicaldsc.org/q

What questions do you have?

Another loss function

Another loss function

- We started by computing the **error** for each of our predictions, but ran into the issue that some errors were positive and some were negative.

$$e_i = \textcolor{blue}{y}_i - \textcolor{orange}{H}(x_i)$$

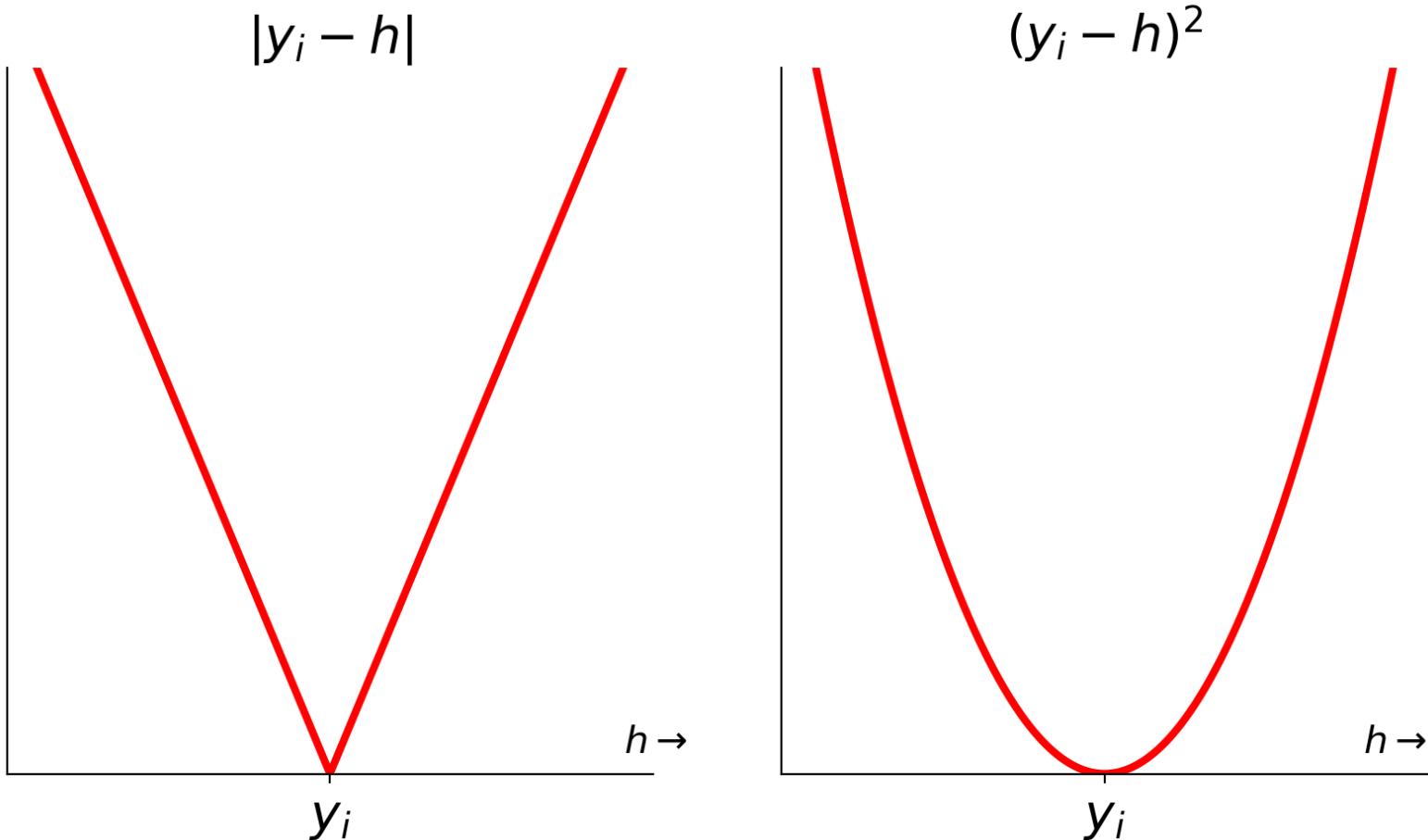
- The solution was to **square** the errors, so that all are non-negative. The resulting loss function is called **squared loss**.

$$L_{\text{sq}}(\textcolor{blue}{y}_i, \textcolor{orange}{H}(x_i)) = (\textcolor{blue}{y}_i - \textcolor{orange}{H}(x_i))^2$$

- Another loss function, which also measures how far $H(x_i)$ is from y_i , is **absolute loss**.

$$L_{\text{abs}}(\textcolor{blue}{y}_i, \textcolor{orange}{H}(x_i)) = |\textcolor{blue}{y}_i - \textcolor{orange}{H}(x_i)|$$

Absolute loss vs. squared loss



Mean absolute error

- Suppose we collect n commute times, y_1, y_2, \dots, y_n .
- The average absolute loss, or mean absolute error (MAE), of the prediction h is:

$$R_{\text{abs}}(h) = \frac{1}{n} \sum_{i=1}^n |y_i - h|$$

- We'd like to find the best constant prediction, h^* , by finding the h that minimizes **mean absolute error** (a new objective function).
- Any guesses?

The median minimizes mean absolute error!

- It turns out that the constant prediction h^* that minimizes mean absolute error,

$$R_{\text{abs}}(h) = \frac{1}{n} \sum_{i=1}^n |y_i - h|$$

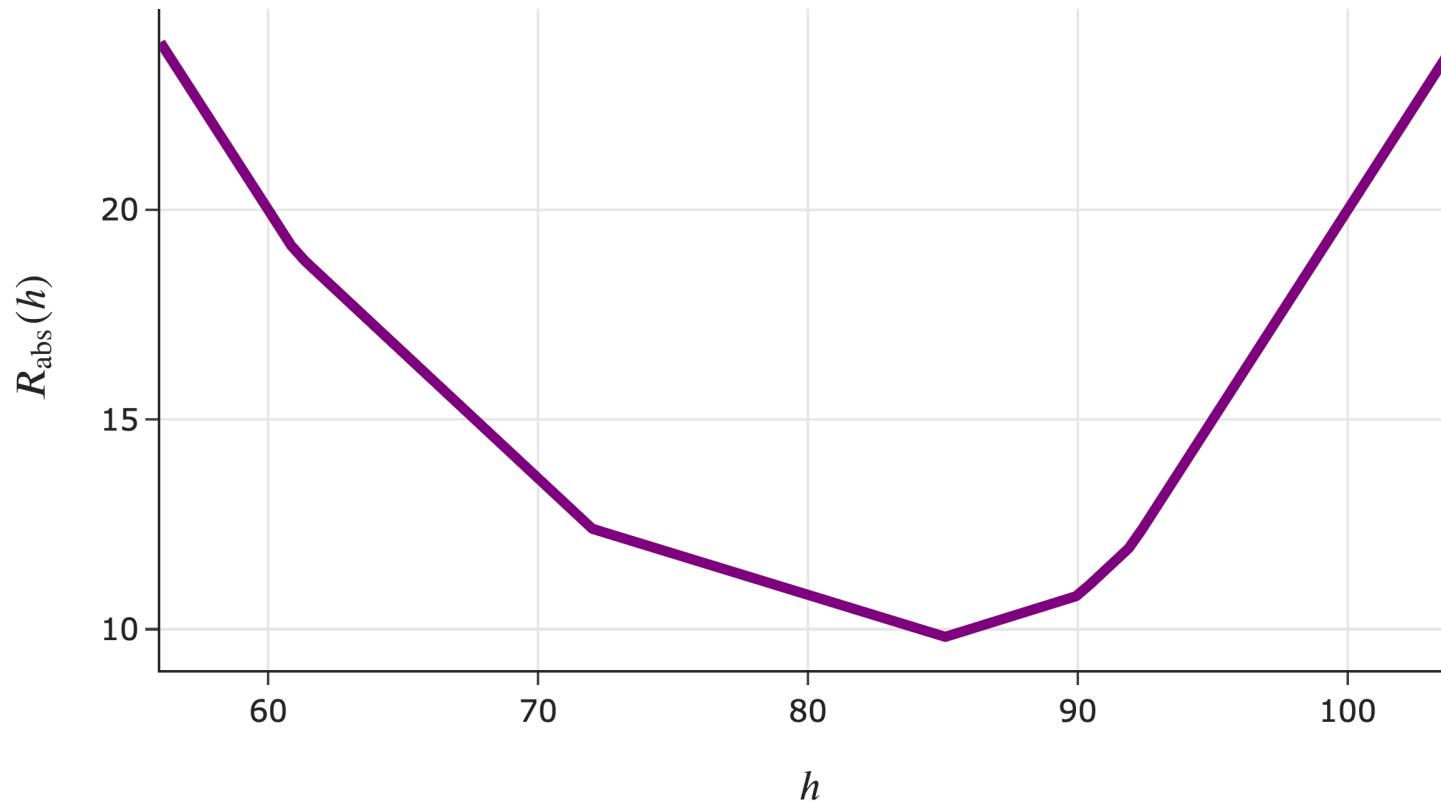
is:

$$h^* = \text{Median}(y_1, y_2, \dots, y_n)$$

- We won't prove this in lecture, but [this extra video](#) walks through it.
Watch it!
- To make a bit more sense of this result, let's graph $R_{\text{abs}}(h)$.

Visualizing mean absolute error

$$R_{\text{abs}}(h) = \frac{1}{5}(|72 - h| + |90 - h| + |61 - h| + |85 - h| + |92 - h|)$$

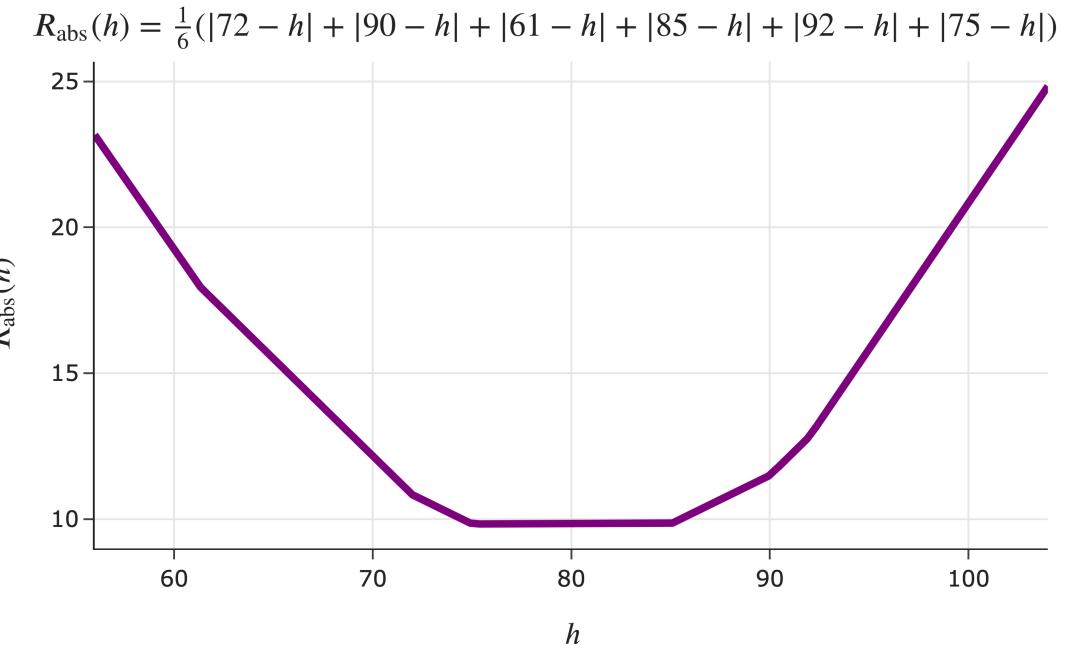


- Consider, again, our example dataset of five commute times.

72, 90, 61, 85, 92

- Where are the "bends" in the graph of $R_{\text{abs}}(h)$ – that is, where does its slope change?

Visualizing mean absolute error, with an even number of points



- What if we add a sixth data point?

72, 90, 61, 85, 92, 75

- Is there a unique h^* ?

The median minimizes mean absolute error!

- The new problem we set out to solve was, find the h^* that minimizes:

$$R_{\text{abs}}(h) = \frac{1}{n} \sum_{i=1}^n |y_i - h|$$

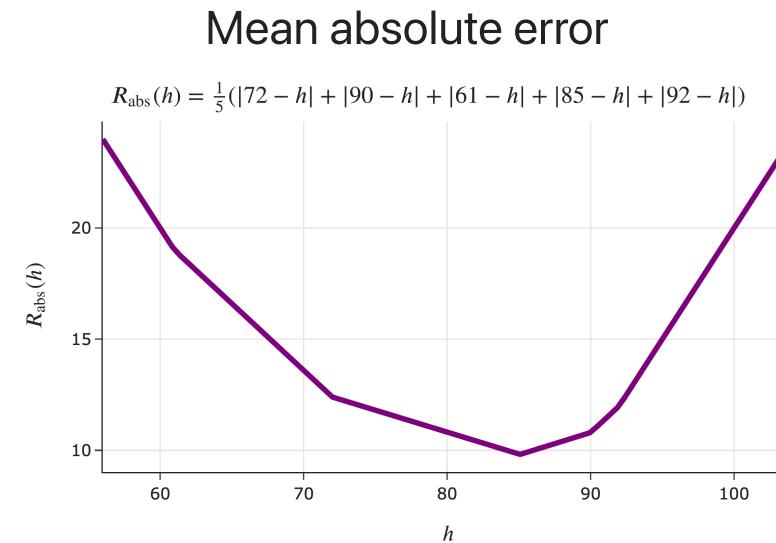
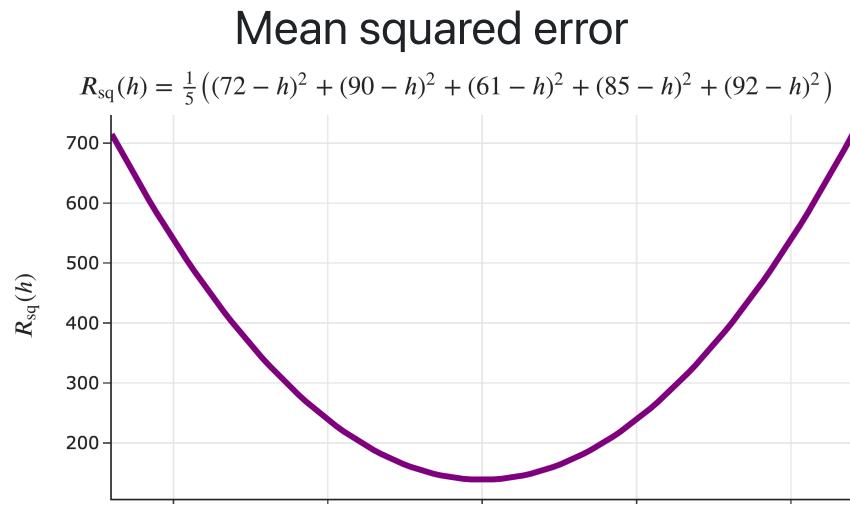
- The answer is:

$$h^* = \text{Median}(y_1, y_2, \dots, y_n)$$

- The **best constant prediction**, in terms of mean absolute error, is always the **median**.
 - When n is odd, this answer is unique.
 - When n is even, any number between the middle two data points (when sorted) also minimizes mean absolute error.
 - When n is even, define the median to be the mean of the middle two data points.

Choosing a loss function

- For the constant model $H(x_i) = h$, the **mean** minimizes mean **squared** error.
- For the constant model $H(x_i) = h$, the **median** minimizes mean **absolute** error.
- In practice, squared loss is the more common choice, as the resulting objective function is more easily **differentiable**.



- But how does our choice of loss function impact the resulting optimal prediction?

Comparing the mean and median

- Consider our example dataset of 5 commute times.

$$y_1 = 72 \quad y_2 = 90 \quad y_3 = 61 \quad y_4 = 85 \quad y_5 = 92$$

- As of now, the **median** is 85 and the **mean** is 80.
 - What if we add 200 to the largest commute time, 92?

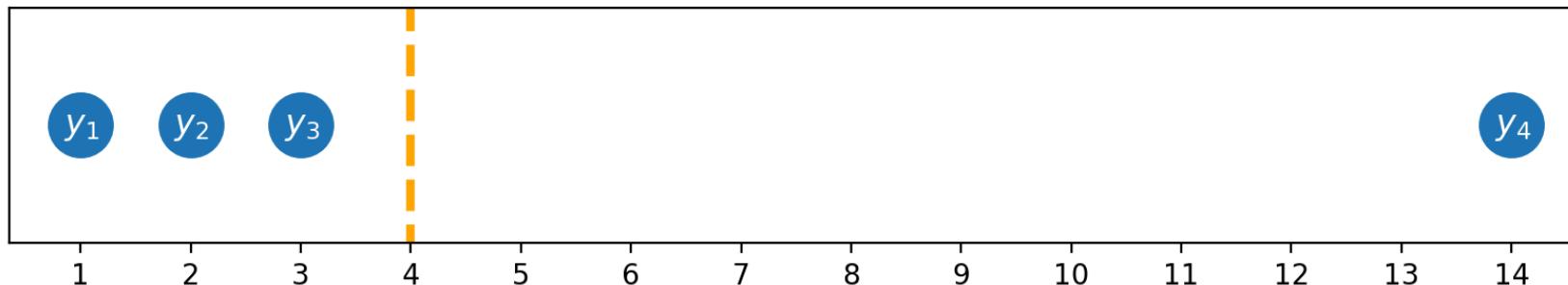
$$y_1 = 72 \quad y_2 = 90 \quad y_3 = 61 \quad y_4 = 85 \quad y_5 = 292$$

- Now, the median is but the mean is !
 - **Key idea:** The mean is quite **sensitive** to outliers.

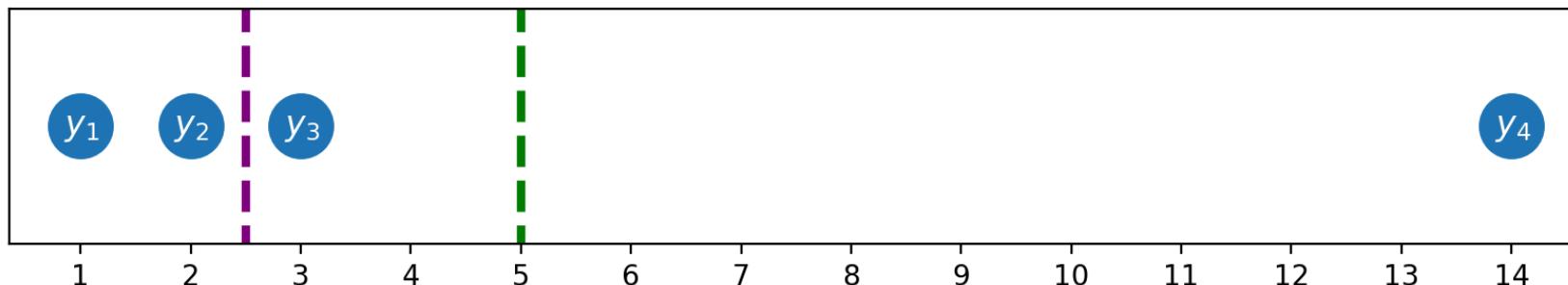
But why?

Outliers

- Below, $|y_4 - h|$ is 10 times as big as $|y_3 - h|$, but $(y_4 - h)^2$ is 100 times $(y_3 - h)^2$.

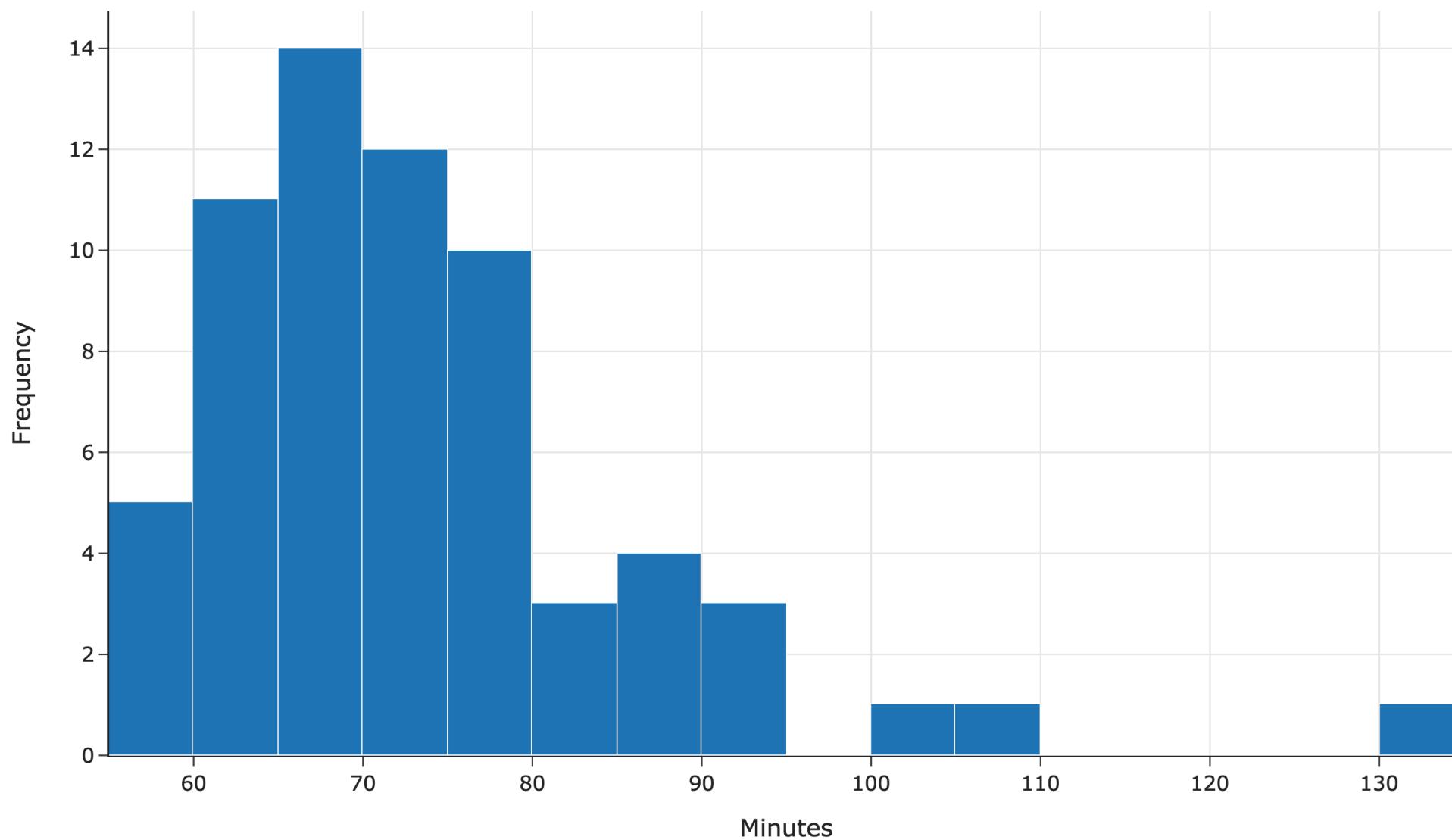


- The result is that the **mean** is "pulled" in the direction of outliers, relative to the **median**.



- As a result, we say the **median** – and absolute loss more generally – is **robust**.

Distribution of Commuting Time

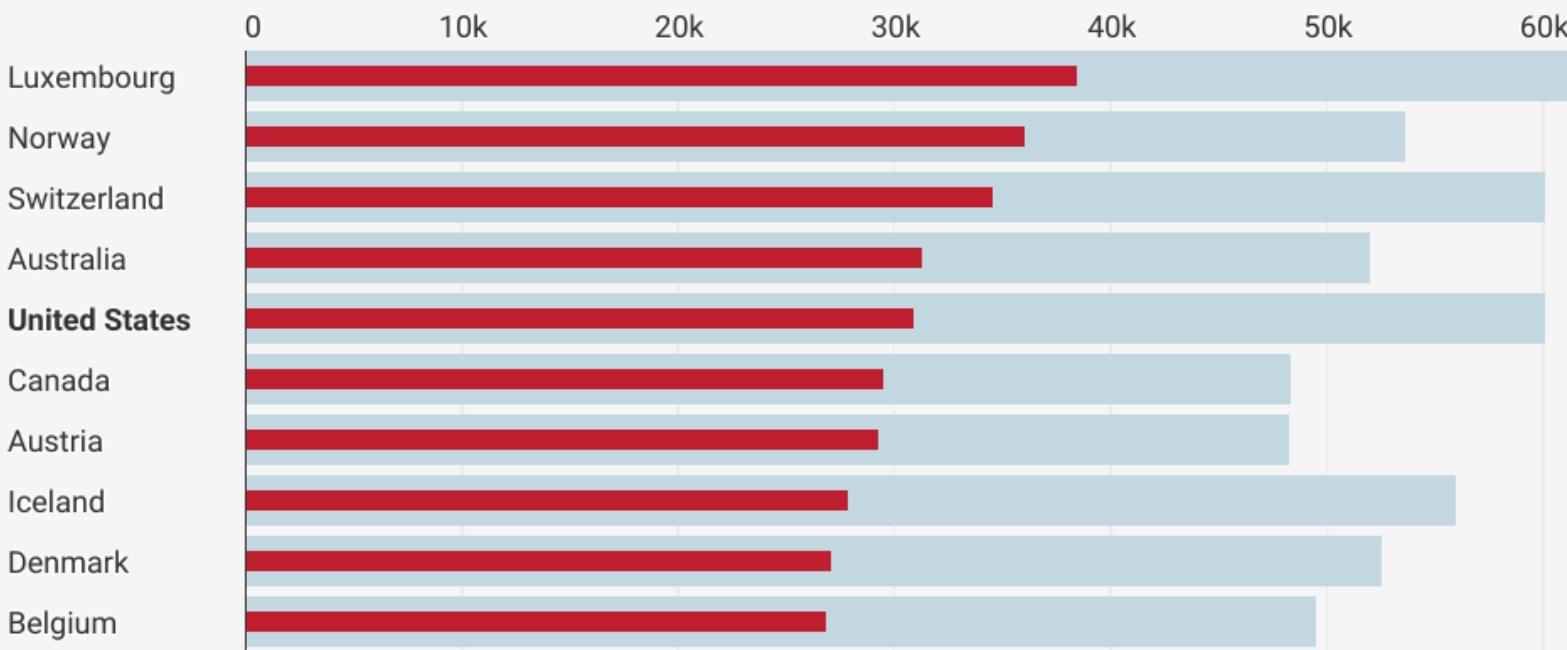


Example: Income inequality

Average vs median income

Median and mean income between 2012 and 2014 in selected OECD countries, in USD; weighted by the currencies' respective purchasing power (PPP).

■ Average income in USD ■ Median income



Summary: Choosing a loss function

- **Key idea:** Different loss functions lead to different best predictions, h^* !

| Loss | Minimizer | Always Unique? | Robust to Outliers? | Differentiable? |
|---|-----------|----------------|---------------------|-----------------|
| $L_{\text{sq}}(y_i, h) = (y_i - h)^2$ | mean | yes | no | yes |
| $L_{\text{abs}}(y_i, h) = y_i - h $ | median | no | yes | no |
| $L_{0,1}(y_i, h) = \begin{cases} 0 & y_i = h \\ 1 & y_i \neq h \end{cases}$ | mode | no | yes | no |
| $L_\infty(y_i, h)$ See HW 6. | ??? | yes | no | no |

- The optimal predictions, h^* , are all **summary statistics** that measure the **center** of the dataset in different ways.

Question 🤔

Answer at practicaldsc.org/q

What questions do you have?