Lecture 13: Midterm Review

EECS 398: Practical Data Science, Winter 2025

practicaldsc.org • github.com/practicaldsc/wn25 • 📣 See latest announcements [here on Ed]

Agenda

- We'll work through the first 8 questions of the Fall 2024 Final Exam: study.practicaldsc.org/fa24-final.
- I'll post these annotated slides after lecture.
- The solutions + recording for yesterday's review session are also posted.

	Type	Brand	Name	Price	Rating	Num Ingredients	Sensitive
0	Eye cream	PERRICONE MD	PRE:EMPT SERIES™ Brightening Eye Cream	55	4.2	33	1
1	Cleanser	CLINIQUE	Pep-Start 2-in-1 Exfoliating Cleanser	190	3.1	36	0
2	Eye cream	PETER THOMAS ROTH	FIRMx™ 360 Eye Renewal	75	5.0	42	0
3	Treatment	KIEHL'S SINCE 1851	Clearly Corrective™ Dark Spot Solution	500	4.5	24	1
4	Cleanser	PETER THOMAS ROTH	Irish Moor Mud Purifying Cleanser Gel	38	3.6	23	0

one product

np.count_nonzero(s) / s.shape[0]

Fall 2024 Final Exam

Problem 1

An expensive product is one that costs **at least $100**.

Problem 1.1

s

Write an expression that evaluates to the **proportion** of products in `skin` that are expensive.

```
(skin["Price"] >= 100).mean()
```

```
(False
 True
 False
 True
 False).mean()
```

(155, 7)

[0] [1]

equivalent:

```
skin[skin["Price"] >= 100].shape[0] / skin.shape[0]
```

	Type	Brand	Name	Price	Rating	Num Ingredients	Sensitive
0	Eye cream	PERRICONE MD	PRE:EMPT SERIES™ Brightening Eye Cream	55	4.2	33	1
1	Cleanser	CLINIQUE	Pep-Start 2-in-1 Exfoliating Cleanser	19	3.1	36	0
2	Eye cream	PETER THOMAS ROTH	FIRMx™ 360 Eye Renewal	75	5.0	42	0
3	Treatment	KIEHL'S SINCE 1851	Clearly Corrective™ Dark Spot Solution	50	4.5	24	1
4	Cleanser	PETER THOMAS ROTH	Irish Moor Mud Purifying Cleanser Gel	38	3.6	23	0

Problem 1.2

Fill in the blanks so that the expression below evaluates to the number of brands that sell **fewer than 5** expensive products.

group condition

```
skin.groupby(__(i)__).__(ii)__(__(iii)__)["Brand"].nunique()
```

function

lambda df: (df["Price"] >= 100).sum() < 5

returns a single True or a single False

`(i)`:

- `"Brand"`
- `"Name"`
- `"Price"`
- `["Brand", "Price"]`

`(ii)`:

- `agg`
- `count`
- `filter`
- `value_counts`

`(iii)`: (Free response)

Tangent: loc

df[ ]

df.loc[which rows do I want?, which columns do I want?]

e.g. to get the "Price" column out of skin,

skin.loc[: , "Price"]

skin["Price"]

e.g. to get the expensive products

skin.loc[skin["Price"] >= 100, ["Price", "Sensitive"]]

Problem 3

loc: labels , iloc: integer positions

Consider the Series `small_prices` and `vc`, both of which are defined below.

```
small_prices = pd.Series([36, 36, 18, 100, 18, 36, 1, 1, 1, 36])

vc = small_prices.value_counts().sort_values(ascending=False)
```

In each of the parts below, select the value that the provided expression evaluates to. If the expression errors, select "Error".

	vc	
	36	4
	1	3
	18	2
	100	1

- 0
- 1
- 2
- 3
- 4
- 18
- 36
- 100
- Error
- None of these

`vc.iloc[0]` 4

`vc.loc[0]` Error

`vc.index[0]` 36

`vc.iloc[1]` 3

`vc.loc[1]` 3

`vc.index[1]` 1

Problem 4

Consider the DataFrames `type_pivot`, `clinique`, `fresh`, and `boscia`, defined below.

```
type_pivot = skin.pivot_table(index="Type",
                              columns="Brand",
                              values="Sensitive",
                              aggfunc=lambda s: s.shape[0] + 1)
```

```
clinique = skin[skin["Brand"] == "clinique"]
fresh = skin[skin["Brand"] == "fresh"]
boscia = skin[skin["Brand"] == "BOSCIA"]
```

one row per unique type

Three columns of `type_pivot` are shown below **in their entirety**.

one row per unique type

Brand	CLINIQUE	FRESH	BOSCIA
Type			
Cleanser	5 ~~6.0~~	NaN	~~2.0~~ 1
Eye cream	3 ~~4.0~~	NaN	~~2.0~~ 1
Face Mask	2 ~~3.0~~	~~4.0~~ 3	~~4.0~~ 3
Moisturizer	2 ~~3.0~~	~~3.0~~ 2	NaN
Sun protect	1 ~~2.0~~	NaN	NaN

In each of the parts below, give your answer as an **integer**.

Problem 4.1

How many rows are in the following DataFrame?

```
clinique.merge(fresh, on="Type", how="inner")
```

$\boxed{10}$

Problem 4.2

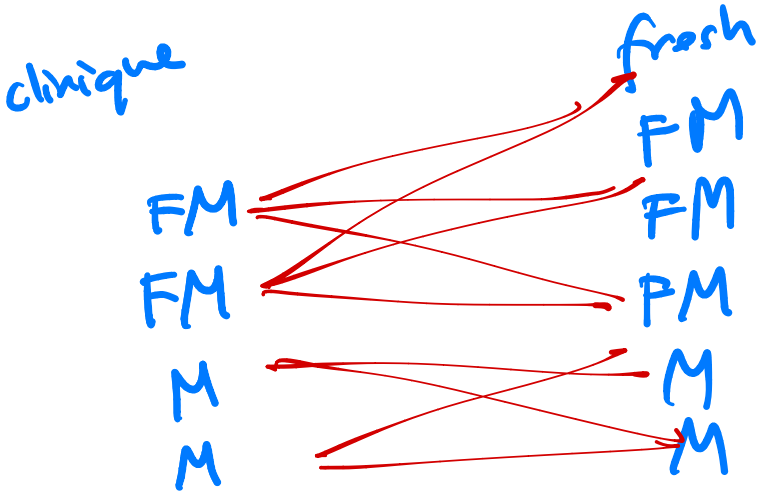
How many rows are in the following DataFrame?

```
(clinique.merge(fresh, on="Type", how="outer")
 .merge(boscia, on="Type", how="outer"))
```

intermediate		boscia
5	x	1
3	x	1
6	x	3
4		nan
1		nan

$= 5 + 3 + 18 + 4 + 1 = \boxed{31}$

8, 26, 30



$$\overbrace{2\times 3}^{FM} + \overbrace{2\times 2}^{M}$$

$$= 6+4$$

$$= \boxed{10}$$

Problem 5

see study site solutions for most detail

Consider a sample of 60 skincare products. The name of one product from the sample is given below:

"our **drops** cream is the best **drops drops** for eye **drops drops** proven formula..."

The total number of terms in the product name above is unknown, but we know that the term **drops** only appears in the name 5 times.

Suppose the TF-IDF of **drops** in the product name above is $\frac{2}{3}$. Which of the following statements are **NOT possible**, assuming we use a base-2 logarithm? Select all that apply.

- [ ] **All** 60 product names contain the term **drops**, **including** the one above.
- [ ] 14 **other** product names contain the term **drops**, in addition to the one above.
- [ ] None of the 59 **other** product names contain the term **drops**.
- [ ] There are 15 terms in the product name above **in total**.
- [ ] There are 25 terms in the product name above **in total**.

Let T be # of terms in product name above,

n be # of documents with "drops", including ~~above~~ document above

$$\text{TF-IDF}(\text{"drops"}, \text{doc above}) = \boxed{\frac{5}{T} \cdot \log_2\left(\frac{60}{n}\right) = \frac{2}{3}}$$

problem boils down to checking if the T, n in the question satisfy this eq'n.

Problem 6

Suppose `soup` is a BeautifulSoup object representing the homepage of a Sephora competitor.

Furthermore, suppose `prods`, defined below, is a list of strings containing the name of every product on the site.

```
prods = [row.get("prod") for row in soup.find_all("row", class_="thing")]
```

Given that `prods[1]` evaluates to `"Cleansifier"`, which of the following options describes the source code of the site?

needs to be within

<row prod="___" >

- Option 1:

```
<row class="thing">prod: Facial Treatment Essence</row>
<row class="thing">prod: Cleansifier</row>
<row class="thing">prod: Self Tan Dry Oil SPF 50</row>
...
```

- Option 2:

```
<row class="thing" prod="Facial Treatment Essence"></row>
<row class="thing" prod="Cleansifier"></row>
<row class="thing" prod="Self Tan Dry Oil SPF 50"></row>
...
```

- Option 3:

```
<row prod="thing" class="Facial Treatment Essence"></row>
<row prod="thing" class="Cleansifier"></row>
<row prod="thing" class="Self Tan Dry Oil SPF 50"></row>
...
```

- Option 4:

```
<row class="thing">prod="Facial Treatment Essence"</row>
<row class="thing">prod="Cleansifier"</row>
<row class="thing">prod="Self Tan Dry Oil SPF 50"</row>
...
```

Problem 7

Consider a dataset of n values, $y_1, y_2, \ldots, y_n$, all of which are **positive**. We want to fit a constant model, $H(x) = h$, to the data.

Let h_p^* be the optimal constant prediction that minimizes average degree-p loss, $R_p(h)$, defined below.

$$R_p(h) = \frac{1}{n} \sum_{i=1}^{n} |y_i - h|^p$$

For example, h_2^* is the optimal constant prediction that minimizes $R_2(h) = \frac{1}{n} \sum_{i=1}^{n} |y_i - h|^2$.

In each of the parts below, determine the value of the quantity provided. By "the data", we are referring to $y_1, y_2, \ldots, y_n$.

- The standard deviation of the data
- The variance of the data
- The mean of the data
- The median of the data
- The midrange of the data, $\frac{y_{\min} + y_{\max}}{2}$
- The mode of the data
- None of the above

h_0^* minimizes $\frac{1}{n} \sum_{i=1}^{n} \underbrace{|y_i - h|^0}$

$\min \frac{1}{n} \sum_{i=1}^{n} |y_i - h|^1$

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

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

h_0^* : mode

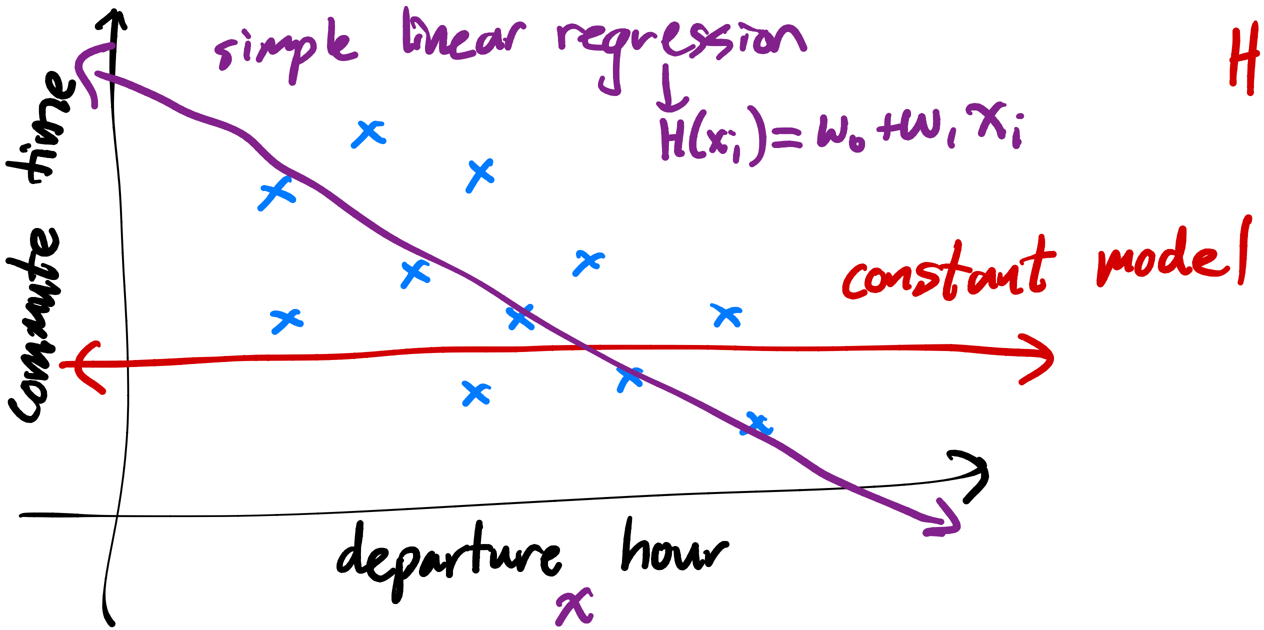
h_1^* Median minimizes absolute loss

$R_1(h_1^*)$ none

h_2^* mean

$R_2(h_2^*)$ Variance

mean



$H(x_i) = h$

loss functions are how we find **which**

constant line or SLR line to use,

i.e. how we find parameters h or w_0, w_1.

constant model: ignore departure hours, x

$$\underbrace{H(x_i) = h}$$

how our predictions are made

Loss function: how wrong an individual prediction is

- squared loss: $(\text{actual} - \text{predicted})^2$
- absolute loss: $|\text{actual} - \text{predicted}|$

Given: $y_1, y_2, \ldots, y_n$ → e.g. $y_1 = 10 \quad y_2 = 50 \quad y_3 = 55$

average loss

$$\underbrace{\frac{1}{3}\left((10-h)^2 + (50-h)^2 + (55-h)^2\right)}$$

find the h^* that minimizes average loss!!!

Chosen:

- constant model, $H(x_i) = h$
- squared loss, $(\text{actual} - \text{predicted})^2$

average squared loss $\quad R_2(h) = \frac{1}{n}\left[(y_1-h)^2 + (y_2-h)^2 + \cdots + (y_n-h)^2\right]$

squared loss

empirical Risk (aka average loss)

Goal: we want best predictions

$\Rightarrow$ to do that, we choose the parameter that minimizes $R_2(h)$

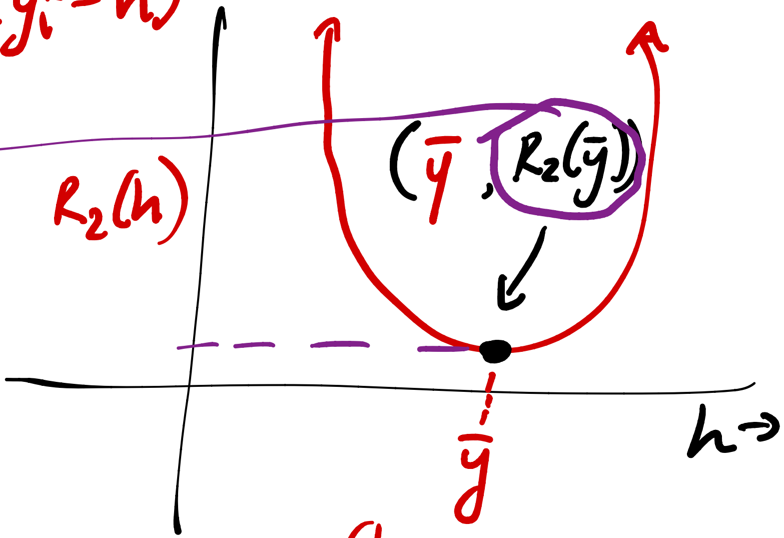
$$\Rightarrow h^* = \text{Mean}(y_1, y_2, \cdots, y_n)$$

- if we choose squared loss : $h^* = \text{Mean}$
- if we choose abs loss : $h^* = \text{Median}$

$\vdots$

variance, standard deviation

$$R_2(h) = \frac{1}{n} \sum_{i=1}^{n} (y_i - h)^2$$



(because mean minimizes $R_2(h)$)

$$\underbrace{R_2(\bar{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \bar{y})^2}_{\text{variance of } y_1, y_2, \ldots, y_n}$$

$R_2(\bar{y})$ = variance of $y_1, y_2, \ldots, y_n$

$\sqrt{R_2(\bar{y})}$ = standard deviation of $y_1, y_2, \ldots, y_n$

$$R_p(h) = \frac{1}{n}\sum_{i=1}^{n} |y_i - h|^p$$

$$R_0(h) = \frac{1}{n}\sum_{i=1}^{n} \underbrace{|y_i - h|^0}$$

$x^0 = 1$

0^0 undefined, but assume $\underbrace{0^0 = 0}$.

either 0 or 1

$[1, 1, 2, 2, 2, 3]$

$h=0 \rightarrow$ all wrong $\rightarrow \frac{1}{6}\sum_{i=1}^{6}(1) = 1$

$h^*=1 \rightarrow$ 2 right, 4 wrong $\rightarrow \frac{1}{6}\cdot(4) = \frac{4}{6}$

$h^*=2 \rightarrow$ 3 right, 3 wrong $\rightarrow \frac{3}{6}$

$h^*=3 \rightarrow \frac{5}{6}$

Now, suppose we want to find the optimal constant prediction, h^*_{U}, using the "Ulta" loss function, defined below.

yi are all positive!

$1000\ (1000 - h)^2$

$$L_U(y_i, h) = y_i(y_i - h)^2$$

$L_2(y_i, h) = (y_i - h)^2$

$(5)\ (5-h)^2$

Problem 7.6

To find h^*_{U}, suppose we minimize average Ulta loss (with no regularization). How does h^*_{U} compare to the mean of the data, M?

- $h^*_{\text{U}} > M$
- $h^*_{\text{U}} \geq M$
- $h^*_{\text{U}} = M$
- $h^*_{\text{U}} \leq M$
- $h^*_{\text{U}} < M$

Lu big when yi is big

Lu small when yi small (close to 0)

Big penalty when yi is a large number

if all yi are the same, $h^*_u = M$

e.g. $\quad 2, \quad 3, \quad 70$

$$R_2(h) = \frac{1}{3}\left[(2-h)^2 + (3-h)^2 + (70-h)^2\right]$$

$$R_u(h) = \frac{1}{3}\left[2(2-h)^2 + 3(3-h)^2 + \underbrace{70(70-h)}(^2\right]$$

to make this small, we make $(70-h)^2$ small, i.e. bring h closer to 70

Now, to find the optimal constant prediction, we will instead minimize **regularized** average Ulta loss, $R_\lambda(h)$, where λ is a non-negative regularization hyperparameter:

$$R_\lambda(h) = \left(\frac{1}{n}\sum_{i=1}^{n} y_i(y_i - h)^2\right) + \lambda h^2$$

objective function

It can be shown that $\frac{\partial R_\lambda(h)}{\partial h}$, the derivative of $R_\lambda(h)$ with respect to h, is:

$$\frac{\partial R_\lambda(h)}{\partial h} = -2\left(\frac{1}{n}\sum_{i=1}^{n} y_i(y_i - h) - \lambda h\right) = 0$$

not in the sum!

Problem 7.7

Find h^*, the constant prediction that minimizes $R_\lambda(h)$. Show your work, and put a $\boxed{\text{box}}$ around your final answer, which should be an **expression in terms of** y_i, n, **and/or** λ.

$$\frac{1}{n}\sum_{i=1}^{n} y_i(y_i - h) - \lambda h = 0$$

$$\frac{1}{n}\sum(y_i^2 - hy_i) - \lambda h = 0$$

$$\frac{1}{n}\left(\sum y_i^2 - \sum hy_i\right) - \lambda h = 0$$

not in the sum!

$$\frac{1}{n}\sum_{i=1}^{n} y_i(y_i-h) - \lambda h = 0$$

$$\frac{1}{n}\sum(y_i^2 - hy_i) - \lambda h = 0$$

$$\frac{1}{n}\left(\sum y_i^2 - \sum h y_i\right) - \lambda h = 0$$

$$\frac{1}{n}\left(\sum y_i^2 - h\sum y_i\right) - \lambda h = 0$$

$$\frac{1}{n}\sum y_i^2 - h\,\frac{\sum y_i}{n} - \lambda h = 0$$

$$\frac{1}{n}\sum y_i^2 = \left(\frac{\sum y_i}{n} + \lambda\right) h$$

$$h^* = \frac{\frac{1}{n}\sum y_i^2}{\frac{1}{n}\sum y_i + \lambda}$$

$$\boxed{h^* = \frac{\sum y_i^2}{\sum y_i + n\lambda}}$$

final answer

$$L(y_i, H(x_i)) = (\text{some loss function})$$

$$\boxed{R(H) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, \underbrace{H(x_i)}_{})}$$

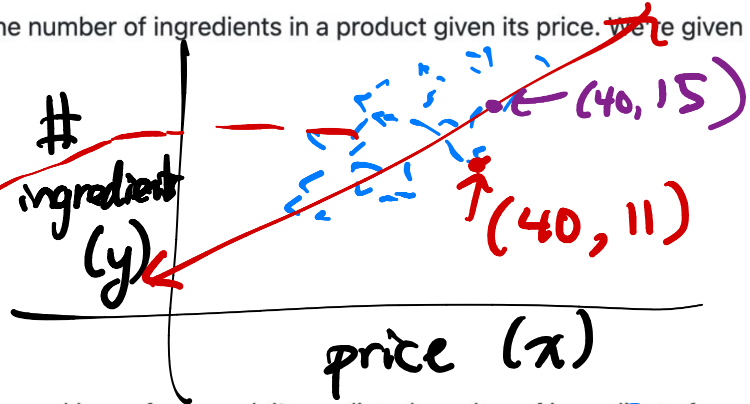
$$H(x_i) = h$$

3

Problem 8

Suppose we want to fit a simple linear regression model (using squared loss) that predicts the number of ingredients in a product given its price. We're given that:

- The average cost of a product in our dataset is \$40, i.e. $\bar{x} = 40$.
- The average number of ingredients in a product in our dataset is 15, i.e. $\bar{y} = 15$.



The intercept and slope of the regression line are $\underbrace{w_0^* = 11}_{\text{intercept}}$ and $\underbrace{w_1^* = \frac{1}{10}}_{\text{slope}}$, respectively.

Problem 8.1

Suppose Victors' Veil (a skincare product) costs \$40 and has 11 ingredients. What is the squared loss of our model's predicted number of ingredients for Victors' Veil? Give your answer as a **number**.

$$\text{squared loss} = (\underbrace{\text{actual}}_{11} - \text{predicted})^2 = (11 - 15)^2 = \boxed{16}$$

$$H(x_i) = 11 + \frac{1}{10} x_i$$

$$H(40) = 11 + \frac{1}{10} \cdot 40 = 11 + 4 = \underbrace{15}_{\text{predicted \# of ingredients}}$$

Problem 8

Suppose we want to fit a simple linear regression model (using squared loss) that predicts the number of ingredients in a product given its price. We're given that:

- The average cost of a product in our dataset is \$40, i.e. $\bar{x} = 40$.
- The average number of ingredients in a product in our dataset is 15, i.e. $\bar{y} = 15$.

The intercept and slope of the regression line are $w_0^* = 11$ and $w_1^* = \frac{1}{10}$, respectively.

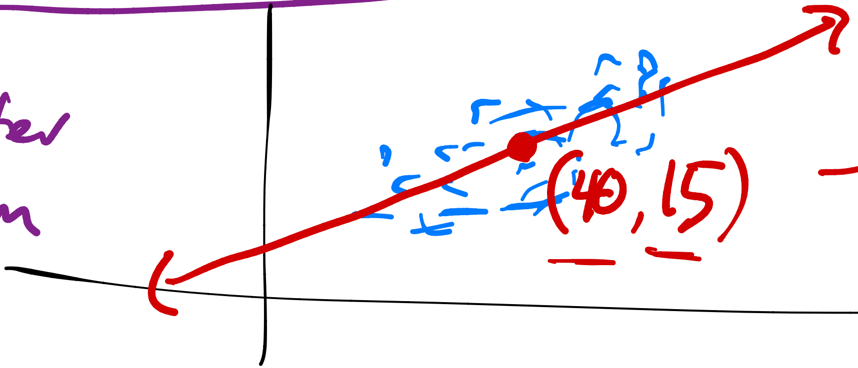
Fact: the regression line ALWAYS passes through the point $(\bar{x}, \bar{y})$

Problem 8.2

Is it possible to answer part (a) above **just** by knowing $\bar{x}$ and $\bar{y}$, i.e. **without** knowing the values of w_0^* and w_1^*?

- Yes; the values of w_0^* and w_1^* don't impact the answer to part (a).
- No; the values of w_0^* and w_1^* are necessary to answer part (a).

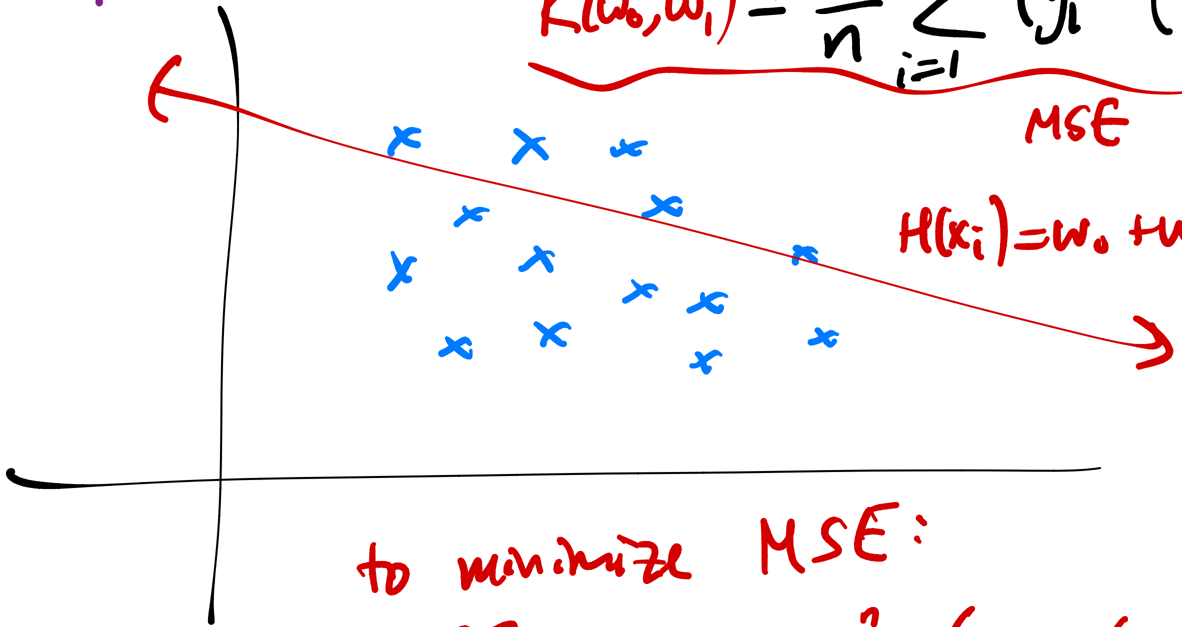
proof after midterm



→ always true if w_0, w_1 minimize mean sq. error

Simple linear regression

$$\underbrace{R(w_0,w_1) = \frac{1}{n}\sum_{i=1}^{n} \left(y_i - (w_0 + w_1 x_i)\right)^2}_{MSE}$$



$H(x_i) = w_0 + w_1 x_i$

set to 0

to minimize MSE:

$$\frac{\partial R}{\partial w_0} = -\frac{2}{n}\sum_{i=1}^{n} \left(y_i - (w_0 + w_1 x_i)\right) = 0$$

solve for w_0^*, w_1^*

$$\frac{\partial R}{\partial w_1} = -\frac{2}{n}\sum \left(y_i - (w_0 + w_1 x_i)\right) x_i = 0$$

$$\underbrace{w_0^*}_{\text{intercept}} = \bar{y} - \underbrace{w_1^*}_{\text{slope}} \bar{x}$$

$$\underbrace{w_1^*}_{\text{slope}} = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2} = \frac{\sum (x_i - \bar{x}) y_i}{\sum (x_i - \bar{x}) x_i}$$

$$= r \frac{\sigma_y}{\sigma_x}$$

SD y

SD x

correlation coefficient

$$w_0^* = \bar{y} - w_1^* - \bar{x} \qquad w_1^* = r\frac{\sigma_y}{\sigma_x}$$

$$H(x_i) = w_0^* + w_1^* x_i$$

$$H(\bar{x}) =$$

$$\vdots$$

$$= \bar{y}$$