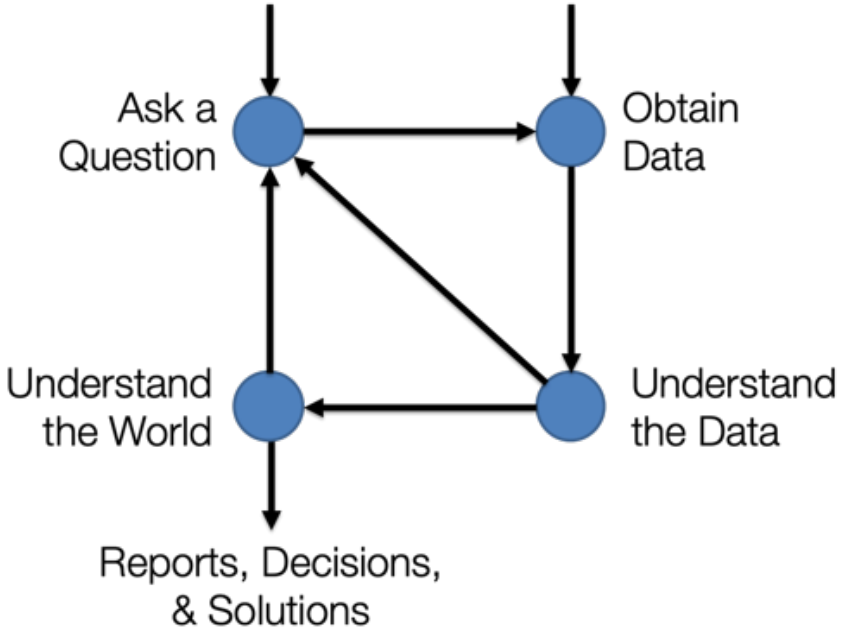
Data Science (COSE471) Spring 2021

Midterm Review

Dept. of Computer Science and Engineering

Korea University



We call this the **Data Science Lifecycle.**

Sampling from a finite population

A census is great, but expensive and difficult to execute.

A **sample** is a subset of the population.

- Samples are often used to make **inferences about the population**.
- How you draw the sample will affect your accuracy.
- Two common sources of error:
  - **chance error**: random samples can vary from what is expected, in any direction.
  - **bias**: a systematic error in one direction.

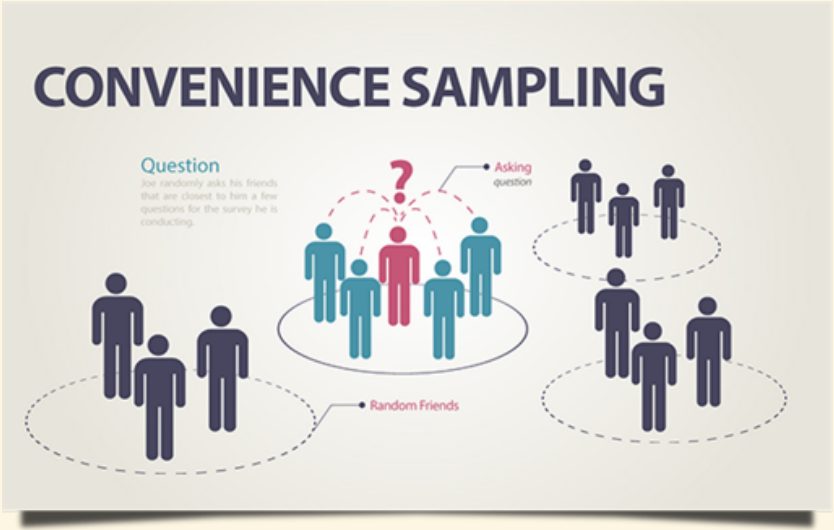
We will now look at some types of **non-random** samples, before formalizing what it means for a sample to be random.

Convenience samples

A **convenience sample** is whomever/whatever is convenient for investigator

- Sources of bias can introduce themselves in ways you may not think of!

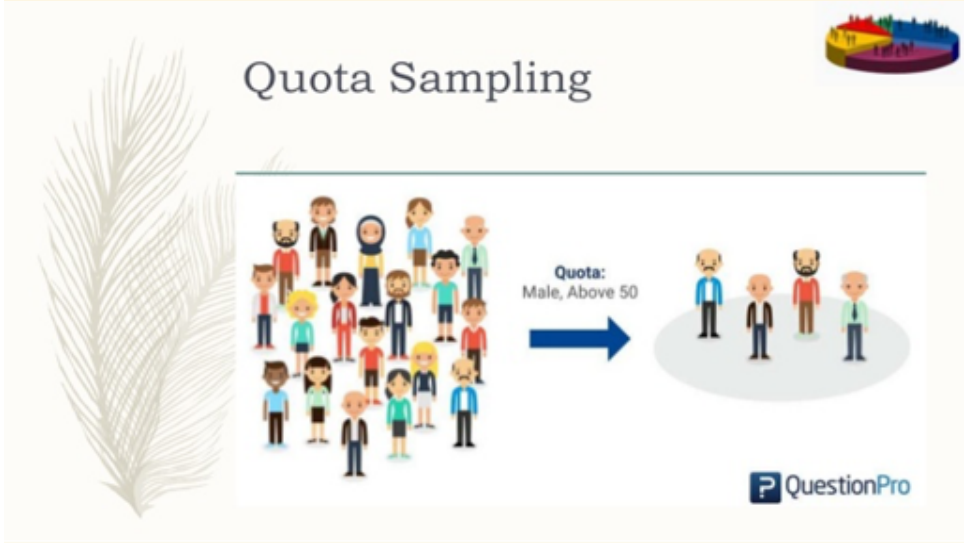
Convenience samples are not random.



Quota samples

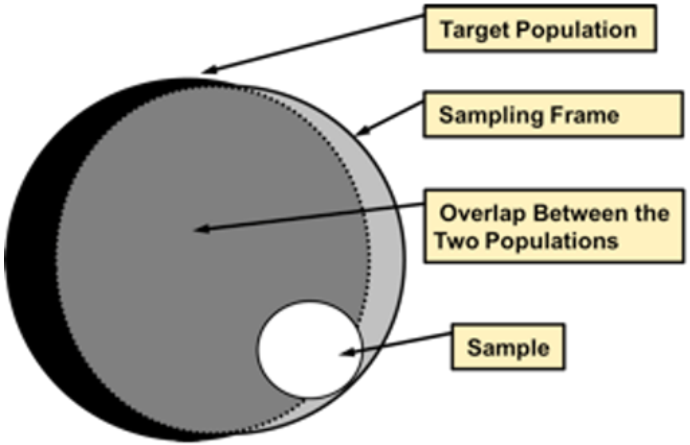
Quota sampling is a procedure

- that restricts the selection of the sample by controlling the number of respondents by one or more criterion
- For example, an interviewer may be told to sample 200 females and 300 males between the age of 45 and 60.



Population, samples, and sampling frame

- **Population:** The group that you want to learn something about.
- **Sampling Frame**: The list from which the sample is drawn.
    - If you're sampling people, the sampling frame is the set of all people that could possibly end up in your sample.
- **Sample**: Who you actually end up sampling.
    - A subset of your sampling frame.



Note: There may be individuals in your sampling frame (and hence, your sample) that are **not** in your population!

Common Biases

Selection Bias

- Systematically excluding (or favoring) particular groups.
- How to avoid: Examine the sampling frame and the method of sampling.

Response Bias

- People don't always respond truthfully.
- How to avoid: Examine the nature of questions and the method of surveying.

Non-response Bias

- People don't always respond.
- How to avoid: Keep your surveys short, and be persistent.
- People who don't respond aren't like the people who do!

Simple Random Sample

A useful representation for sampling is a box model

- where the population of interest is represented by a box of N tickets, each with values written on them (data!)

A **simple random sample (SRS)** is a sample drawn **uniformly** at random **without** replacement from the box.

- For a small sample compared to the population, SRS is very close to sampling at random with replacement.

Stratified vs. Cluster Sample

Cluster vs. Stratified Sample

- In cluster sampling, we use a single SRS to select groups
- In Stratified sampling, we use one SRS per group to select individuals

Stratified sampling results in increased precision and representation.

The City of Berkeley wants to hear from its homeowners on issues related to zoning laws.

(For the purposes of this question, homeowners are individuals who own their home, instead of leasing or renting from someone else.)

(a) (1 pt) One method of surveying would be to have city workers come to UC Berkeley's campus and ask passing by students and faculty members for their thoughts. Suppose for now that the question "Are you a homeowner?" is not asked.

What type of sample is this?

Probability sample, but not simple random sample

Simple random sample

Quota sample

Convenience sample

(b) (1 pt) Many students and faculty members aren't homeowners, but will be surveyed anyways.

What form of bias or error is this?

Response bias

Non-response bias

Selection bias

Chance error

The City of Berkeley wants to hear from its homeowners on issues related to zoning laws.

(For the purposes of this question, homeowners are individuals who own their home, instead of leasing or renting from someone else.)

(a) (1 pt) One method of surveying would be to have city workers come to UC Berkeley's campus and ask passing by students and faculty members for their thoughts. Suppose for now that the question "Are you a homeowner?" is not asked.

What type of sample is this?

- ○ Probability sample, but not simple random sample
- ○ Simple random sample
- ○ Quota sample
- ● Convenience sample

(b) (1 pt) Many students and faculty members aren't homeowners, but will be surveyed anyways.

What form of bias or error is this?

- ○ Response bias
- ○ Non-response bias
- ● Selection bias
- ○ Chance error

(c) (1 pt) The City of Berkeley has a list of all the homeowners' email addresses. Instead of the previous surveying technique, now suppose they take the list of all homeowners' email addresses, shuffle it, and send a survey to every other email address. That is, from the shuffled list, they email the first, third, fifth, seventh, and so on.

(You may assume that the shuffling is done uniformly at random, meaning that each email address has the same probability of landing in any particular position. You may also assume that the City of Berkeley has the email address for every single homeowner, and that every single homeowner has a unique email address.)

What type of sample is this?

Probability sample

Quota sample

Convenience sample

(d) (1 pt) Fill in the blank: In this new sampling technique, the sampling frame is _____ the population of interest.

smaller than

equal to

greater than

(c) (1 pt) The City of Berkeley has a list of all the homeowners' email addresses. Instead of the previous surveying technique, now suppose they take the list of all homeowners' email addresses, shuffle it, and send a survey to every other email address. That is, from the shuffled list, they email the first, third, fifth, seventh, and so on.

(You may assume that the shuffling is done uniformly at random, meaning that each email address has the same probability of landing in any particular position. You may also assume that the City of Berkeley has the email address for every single homeowner, and that every single homeowner has a unique email address.)

What type of sample is this?

- ● Probability sample
- ○ Quota sample
- ○ Convenience sample

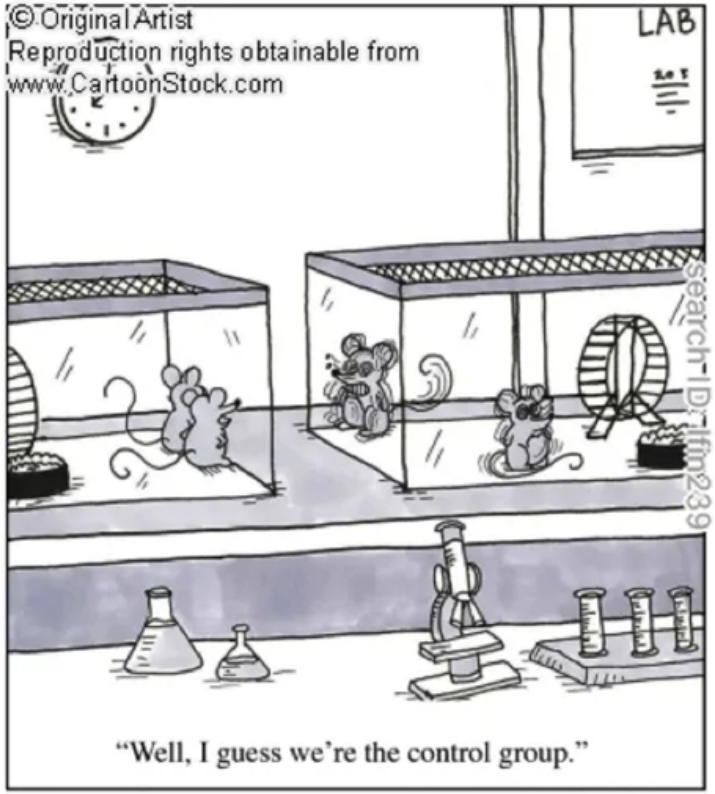
(d) (1 pt) Fill in the blank: In this new sampling technique, the sampling frame is _____ the population of interest.

- ○ smaller than
- ● equal to
- ○ greater than

Designed Experiments

Examine the association/effect of a treatment on an outcome when the variable of interest is under the control of the investigator.

- E.g. clinical trial to test effect of new drug on patients with Alzheimer's disease.
- E.g. A/B test for two versions of a website

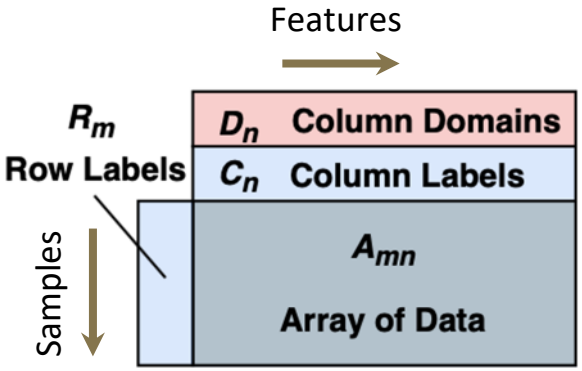


http://abcnews.go.com/Lifestyle/silly-baby-panda-falls-flat-face-public-debut/story?id=42481478

The world, a statistician's view

A (statistical) population from which we draw **samples**.
Each sample has certain **features**.

	Candidate	Party	%	Year	Result
0	Obama	Democratic	52.9	2008	win
1	McCain	Republican	45.7	2008	loss
2	Obama	Democratic	51.1	2012	win
3	Romney	Republican	47.2	2012	loss
4	Clinton	Democratic	48.2	2016	loss
5	Trump	Republican	46.1	2016	win

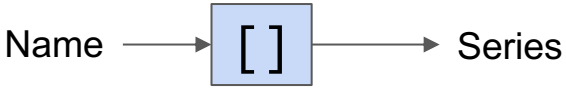


A generic DataFrame
(from https://arxiv.org/abs/2001.00888)

Pandas and Jupyter Notebooks

- Introduced DataFrame concepts
  - **Series**: A named column of data with an index
  - **Indexes**: The mapping from keys to rows
  - **DataFrame**: collection of series with common index
- Dataframe access methods
  - **Filtering** on predicts and **slicing**
  - **df.loc**: location by index
  - **df.iloc**: location by integer address
  - **groupby** data

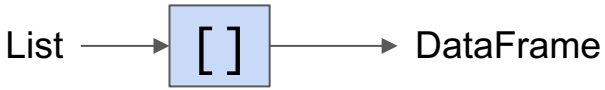
[] Summary



Single Column Selection

```
elections["Candidate"].head(6)
```

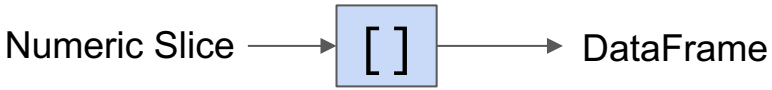
```
Year
1980      Reagan
1980      Carter
1980    Anderson
1984      Reagan
1984     Mondale
1988        Bush
Name: Candidate, dtype: object
```



Multiple Column Selection

```
elections[["Candidate"]].head(6)
```

Year	Candidate
1980	Reagan
1980	Carter
1980	Anderson
1984	Reagan
1984	Mondale
1988	Bush



(Multiple) Row Selection

```
elections[0:3]
```

Year	Candidate	Party	%	Result
1980	Reagan	Republican	50.7	win
1980	Carter	Democratic	41.0	loss
1980	Anderson	Independent	6.6	loss

Boolean Array Input

Boolean Series can be combined using the & operator, allowing filtering of results by multiple criteria.

```
elections[(elections['Result'] == 'win')
          & (elections['%'] < 50)]
```

	Candidate	Party	%	Year	Result
7	Clinton	Democratic	43.0	1992	win
10	Clinton	Democratic	49.2	1996	win
14	Bush	Republican	47.9	2000	win
22	Trump	Republican	46.1	2016	win

Loc with Lists

The most basic use of loc is to provide a list of row and column labels, which returns a DataFrame.

```
elections.loc[[0, 1, 2, 3, 4], ['Candidate','Party', 'Year']]
```

	Candidate	Party	Year
0	Reagan	Republican	1980
1	Carter	Democratic	1980
2	Anderson	Independent	1980
3	Reagan	Republican	1984
4	Mondale	Democratic	1984

iloc: Integer-Based Indexing for Selection by Position

In contrast to loc, iloc doesn't think about labels at all. Instead, it returns the items that appear in the numerical positions specified.

```
elections.iloc[0:3, 0:3]
```

	Candidate	Party	%
0	Reagan	Republican	50.7
1	Carter	Democratic	41.0
2	Anderson	Independent	6.6

```
mottos.iloc[0:3, 0:3]
```

State	Motto	Translation	Language
Alabama	Audemus jura nostra defendere	We dare defend our rights!	Latin
Alaska	North to the future	—	English
Arizona	Ditat Deus	God enriches	Latin

Advantages of loc:

- Harder to make mistakes.
- Easier to read code.
- Not vulnerable to changes to the ordering of rows/cols in raw data files.

Nonetheless, iloc can be more convenient. *Use iloc judiciously*.

Sample

If you want a DataFrame consisting of a random selection of rows, you can use the sample method.

- By default, *it is by default without replacement*. Use **`replace=true`** for replacement.
- Naturally, can be chained with our selection operators [], loc, iloc.

```
elections.sample(10)
```

	Candidate	Party	%	Year	Result
15	Kerry	Democratic	48.3	2004	loss
16	Bush	Republican	50.7	2004	win
22	Trump	Republican	46.1	2016	win
9	Perot	Independent	18.9	1992	loss
21	Clinton	Democratic	48.2	2016	loss
11	Dole	Republican	40.7	1996	loss
20	Romney	Republican	47.2	2012	loss
14	Bush	Republican	47.9	2000	win
8	Bush	Republican	37.4	1992	loss
1	Carter	Democratic	41.0	1980	loss

```
elections.query("Year < 1992").sample(4, replace=True)
```

	Candidate	Party	%	Year	Result
1	Carter	Democratic	41.0	1980	loss
4	Mondale	Democratic	37.6	1984	loss
6	Dukakis	Democratic	45.6	1988	loss
1	Carter	Democratic	41.0	1980	loss

`head`, `size`, `shape`, and `describe`

`head`: Displays only the top few rows.

`size`: Gives the total number of data points.

`shape`: Gives the size of the data in rows and columns.

`describe`: Provides a summary of the data.

`index` and `columns`

`index`: Returns the index (a.k.a. row labels).

`columns`: Returns the labels for the columns.

The `sort_values` Method

One incredibly useful method for DataFrames is `sort_values`, which creates a copy of a DataFrame sorted by a specific column.

```
elections.sort_values('%', ascending=False)
```

	Candidate	Party	%	Year	Result
3	Reagan	Republican	58.8	1984	win
5	Bush	Republican	53.4	1988	win
17	Obama	Democratic	52.9	2008	win
19	Obama	Democratic	51.1	2012	win
0	Reagan	Republican	50.7	1980	win

The `value_counts` Method

Series also has the function **`value_counts`**, which creates a new Series showing the counts of every value.

```
elections['Party'].value_counts()
```

```
Democratic     10
Republican     10
Independent     3
Name: Party, dtype: int64
```

The `unique` Method

Another handy method for Series is **`unique`**, which returns all unique values as an array.

```
mottos['Language'].unique()
```

```
array(['Latin', 'English', 'Greek', 'Hawaiian', 'Italian', 'French',
       'Spanish', 'Chinook Jargon'], dtype=object)
```

A More Advanced Approach

Approach 1: Use a list comprehensions.

```
j_names = babynames[ [x.startswith('J') for x in babynames["Name"]] ]
```

Approach 2: Use a `str` method from the `Series` class (more on this shortly).

```
j_names = babynames[babynames["Name"].str.startswith('J')]
```

Question: What's better about this second approach?

- **More readable!** Others can understand your code. ← the main great thing
- First one is likely to be less efficient.

Sorting by Arbitrary Functions

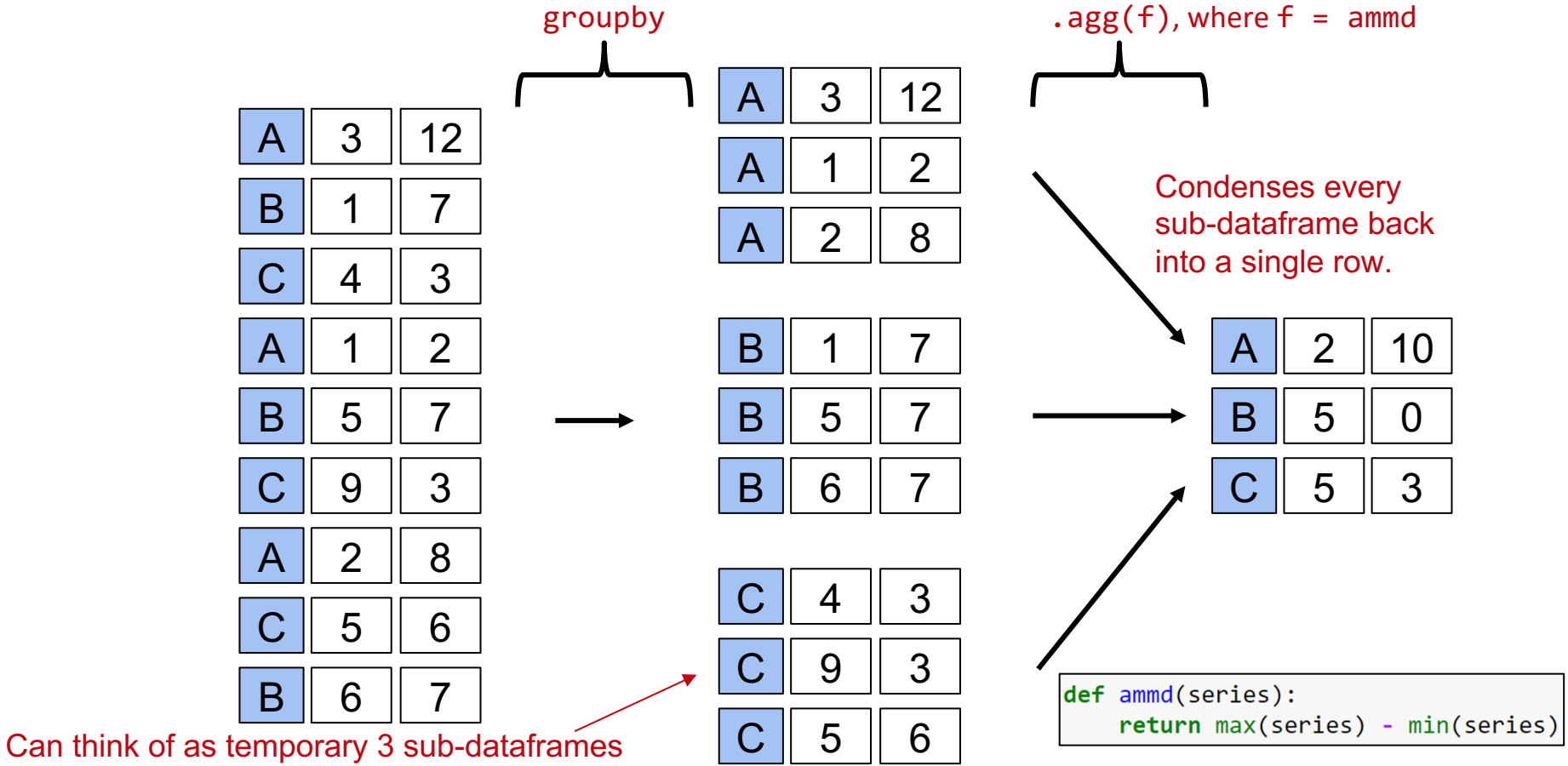
Suppose we want to sort by the number of occurrences of “dr” + number of occurrences of “ea”.

- Use the Series `.map` method.

```
def dr_ea_count(string):
    return string.count('dr') + string.count('ea')

babynames["dr_ea_count"] = babynames["Name"].map(dr_ea_count)
babynames = babynames.sort_values(by = "dr_ea_count", ascending=False)
```

	State	Sex	Year	Name	Count	dr_ea_count
108712	CA	F	1988	Deandrea	5	3
293396	CA	M	1985	Deandrea	6	3
101958	CA	F	1986	Deandrea	6	3
115935	CA	F	1990	Deandrea	5	3
131003	CA	F	1994	Leandrea	5	3

DataFrame groupby.agg Visually



4. (24.0 points)

Throughout this question, we are dealing with pandas DataFrame and Series objects. All code for this question, where applicable, must be written in Python. You may assume that pandas has been imported as `pd`.

The DataFrame `eb_hikes` provided below shows different hikes in the East Bay.
The information contained in the DataFrame includes:

- Trail: trail name which is unique (string)
- Elevation Gain: total elevation gain (ft) of a trail (int)
- Length: length of trail in miles (float)
- Location: City, State where the trails are located (string)

The first four rows of `eb_hikes` are shown below.

	Trail	Elevation Gain	Length	Location
0	Lake Temescal Loop	95	1.1	Oakland, CA
1	Inspiration Point	452	4.0	Berkeley, CA
2	Fire Trails	1496	4.2	Berkeley, CA
3	Piedmont Park Loop	137	0.8	Piedmont, CA

(a) (2.0 pt) Since all the trails are located in the East Bay, we know that they are all located in California. Write a line of Pandas code that changes `eb_hikes['Location']` to only display the city in which the trail is located. (For instance, the first four elements of `eb_hikes['Location']` would be `['Oakland', 'Berkeley', 'Berkeley', 'Piedmont']`.)

4. (24.0 points)

Throughout this question, we are dealing with pandas DataFrame and Series objects. All code for this question, where applicable, must be written in Python. You may assume that pandas has been imported as `pd`.

The DataFrame `eb_hikes` provided below shows different hikes in the East Bay.
The information contained in the DataFrame includes:

- Trail: trail name which is unique (string)
- Elevation Gain: total elevation gain (ft) of a trail (int)
- Length: length of trail in miles (float)
- Location: City, State where the trails are located (string)

The first four rows of `eb_hikes` are shown below.

	Trail	Elevation Gain	Length	Location
0	Lake Temescal Loop	95	1.1	Oakland, CA
1	Inspiration Point	452	4.0	Berkeley, CA
2	Fire Trails	1496	4.2	Berkeley, CA
3	Piedmont Park Loop	137	0.8	Piedmont, CA

(a) (2.0 pt) Since all the trails are located in the East Bay, we know that they are all located in California. Write a line of Pandas code that changes `eb_hikes['Location']` to only display the city in which the trail is located. (For instance, the first four elements of `eb_hikes['Location']` would be `['Oakland', 'Berkeley', 'Berkeley', 'Piedmont']`.)

```
eb_hikes['Location'] = eb_hikes['Location'].str.split(',', expand=True)[0]
```

(b) **(3.0 pt)** We like to go on hikes with large Elevation Gains, and we also like to go on hikes in cityname. Write a line of Pandas code to create a `Series` containing the names of the trails who satisfy at least one of the following conditions:

- They are located in cityname
- Their Elevation Gain is at least elevnum ft

Assign your result to the variable `varname`. You may assume that `eb_hikes['Location']` has already been modified according to the previous part.

(b) **(3.0 pt)** We like to go on hikes with large Elevation Gains, and we also like to go on hikes in cityname. Write a line of Pandas code to create a `Series` containing the names of the trails who satisfy at least one of the following conditions:

- They are located in cityname
- Their Elevation Gain is at least elevnum ft

Assign your result to the variable `varname`. You may assume that `eb_hikes['Location']` has already been modified according to the previous part.

```
varname = eb_hikes.loc[(eb_hikes['Elevation Gain'] >= elevnum) |
(eb_hikes['Location'] == 'cityname'), 'Trail']
```

Trail	Fire Trails	Inspiration Point	Piedmont Park Loop
User			
Bob Honey	35.714286	NaN	12.5
Josh Loop	NaN	NaN	25.0
Susie Thomas	60.714286	23.75	NaN

(f) **(2.0 pt)** Consider the DataFrame `df` you created in the previous part. Suppose someone told you that User "Michael James" was the fastest recorded individual, as measured by average minutes per mile, to finish the Piedmont Park Loop (with no ties). Which of the following is guaranteed to correctly determine Michael James' average minutes per mile for the Inspiration Point trail? Select all that apply.

```
df.loc[df['Piedmont Park Loop'] == df['Piedmont Park Loop'].min(), 'Inspiration Point']
```

```
df.loc[~(df['Piedmont Park Loop'] <= df['Piedmont Park Loop'].max()), 'Inspiration Point']
```

```
df.sort_values('Piedmont Park Loop').loc[:, 'Inspiration Point'].iloc[0]
```

```
df.sort_values('Piedmont Park Loop').loc[:, 'Inspiration Point'].loc[0]
```

None of the above

Trail / User	Fire Trails	Inspiration Point	Piedmont Park Loop
Bob Honey	35.714286	NaN	12.5
Josh Loop	NaN	NaN	25.0
Susie Thomas	60.714286	23.75	NaN

(f) (2.0 pt) Consider the DataFrame `df` you created in the previous part. Suppose someone told you that User "Michael James" was the fastest recorded individual, as measured by average minutes per mile, to finish the Piedmont Park Loop (with no ties). Which of the following is guaranteed to correctly determine Michael James' average minutes per mile for the Inspiration Point trail? Select all that apply.

- [x] `df.loc[df['Piedmont Park Loop'] == df['Piedmont Park Loop'].min(), 'Inspiration Point']`
- [ ] `df.loc[~(df['Piedmont Park Loop'] <= df['Piedmont Park Loop'].max()), 'Inspiration Point']`
- [x] `df.sort_values('Piedmont Park Loop').loc[:, 'Inspiration Point'].iloc[0]`
- [ ] `df.sort_values('Piedmont Park Loop').loc[:, 'Inspiration Point'].loc[0]`
- [ ] None of the above

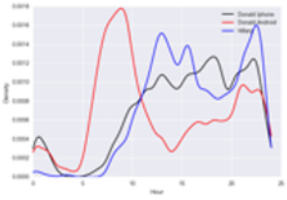
`df.loc[df['Piedmont Park Loop'] == df['Piedmont Park Loop'].min(), 'Inspiration Point']` correctly computes the result.

`df.loc[~(df['Piedmont Park Loop'] <= df['Piedmont Park Loop'].max()), 'Inspiration Point']` is wrong, and is a distractor. It will return an empty Series, since "not less than or equal to the max" equates to "greater than the max", which is nonsensical.

`df.sort_values('Piedmont Park Loop').loc[:, 'Inspiration Point'].iloc[0]` also correctly computes the result.

`df.sort_values('Piedmont Park Loop').loc[:, 'Inspiration Point'].loc[0]` is also wrong, because `.loc` requires us to pass in an index label, and 0 is not in our index labels'.

Data Acquisition



Exploratory Data Analysis

Topics For This Lecture

- Understanding the Data
  - Data Cleaning
  - Exploratory Data Analysis (EDA)
  - Basic data visualization
- Common Data Anomalies
  - ... and how to fix them

Data Cleaning

- The process of transforming **raw data** to facilitate subsequent analysis
- Data cleaning often addresses **issues**
  - structure / formatting
  - missing or corrupted values
  - unit conversion
  - encoding text as numbers
  - …
- Sadly, data cleaning is a big part of data science…

Exploratory Data Analysis (EDA)

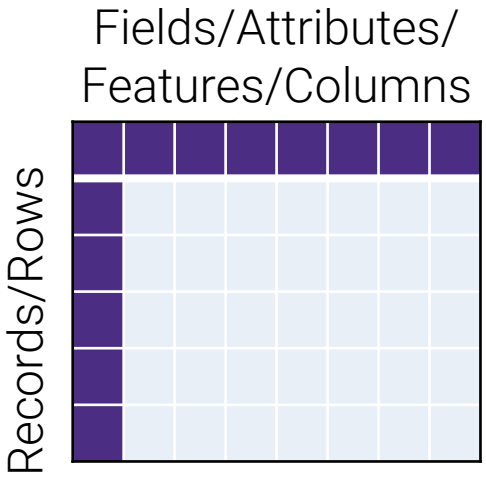
"Getting to know the data"

- The process of **transforming**, **visualizing**, and **summarizing** data to:
  - Build/confirm understanding of the data and its provenance
  - Identify and address potential issues in the data
  - Inform the subsequent analysis
  - discover *potential* hypothesis … (be careful)
- **EDA is an open-ended analysis**
  - Be willing to find something surprising

Key Data Properties to Consider in EDA

- **Structure --** *the "shape" of a data file*
- **Granularity --** *how fine/coarse is each datum*
- **Scope --** *how (in)complete is the data*
- **Temporality --** *how is the data situated in time*
- **Faithfulness --** *how well does the data capture "reality"*

Rectangular Data



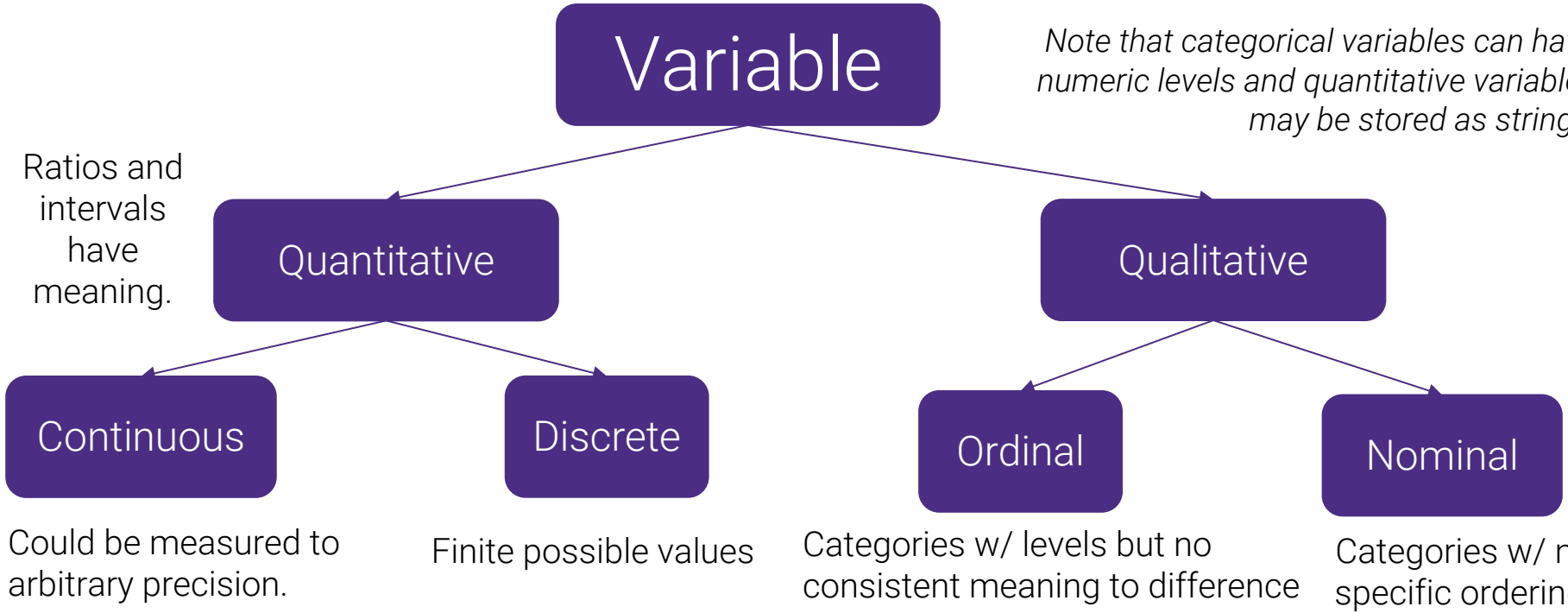
We prefer rectangular data for data analysis (why?)

- Regular structures are easy manipulate and analyze
- A big part of data cleaning is about transforming data to be more rectangular

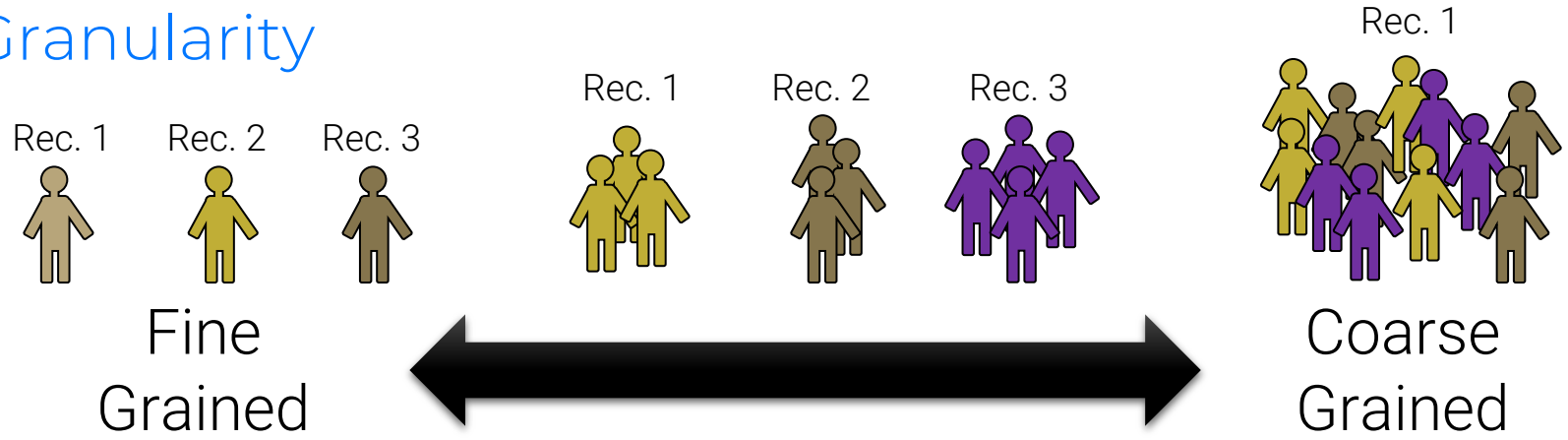
Two kinds of rectangular data: *Tables and Matrices*

(what are the differences?)

1. **Tables** (a.k.a. data-frames in R/Python and relations in SQL)
   - Named columns with different types
   - Manipulated using data transformation languages (map, filter, group by, join, …)
2. **Matrices**
   - Numeric data of the same type
   - Manipulated using linear algebra


Variable
Note that categorical variables can have numeric levels and quantitative variables may be stored as strings.
Ratios and intervals have meaning.
Quantitative
Qualitative
Continuous
Discrete
Ordinal
Nominal
Could be measured to arbitrary precision.
Finite possible values
Categories w/ levels but no consistent meaning to difference
Categories w/ no specific ordering.
Examples:
• Price
• Temperature
Examples:
• Number of siblings
• Yrs of education
Examples:
• Preferences
• Level of education
Examples:
• Political Affiliation
• ID number

Granularity



- What does each record represent?
  - Examples: a purchase, a person, a group of users
- Do all records capture granularity at the same level?
  - Some data will include summaries (aka rollups) as records
- If the data are coarse how was it aggregated?
  - Sampling, averaging, ...

Scope

- Does my data cover my area of interest?
  - **Example:** *I am interested in studying crime in Korea but I only have Seoul crime data.*
- Is my data too big?
  - **Example:** *I am interested in student grades for COSE471 but have student grades for all CS classes.*
  - **Solution:** *Filtering ⇒ Implications on sample?*
    - *If the data is a sample I may have poor coverage after filtering …*
- Does my data cover the right time frame?
  - More on this in temporality …

Temporality

- Data changes – when was the data collected?
- What is the meaning of the time and date fields?
  - When the “event” **happened**?
  - When the data was **collected** or was **entered** into the system?
  - Date the data was copied into a database (look for many matching timestamps)
- Time depends on where! (Time zones & daylight savings)
  - Learn to use **datetime** python library
  - Multiple string representation (depends on region): 07/08/09?
- Are there strange null values?
  - January 1st 1970, January 1st 1900
- Is there periodicity? Diurnal patterns

Faithfulness: *Do I trust this data?*

- Does my data contain **unrealistic** or **“incorrect”** values?
  - Dates in the future for events in the past
  - Locations that don’t exist
  - Negative counts
  - Misspellings of names
  - Large outliers
- Does my data violate **obvious dependencies**?
  - E.g., age and birthday don’t match
- Was the data **entered by hand**?
  - Spelling errors, fields shifted …
  - Did the form require fields or provide default values?
- Are there obvious signs of **data falsification**:
  - Repeated names, fake looking email addresses, repeated use of uncommon names or fields.

	User	Trail	Rating	Difficulty	Time Taken
0	Bob Honey	Piedmont Park Loop	2	Easy	10
1	Susie Thomas	Fire Trails	4.3	Very Hard	255
2	Josh Loop	Piedmont Park Loop	4	Easy	20
3	Susie Thomas	Inspiration Point	3	Medium	95
4	Bob Honey	Fire Trails	4	Hard	150

(d) (1.0 pt) What type of variable is Difficulty in the `user_hikes` DataFrame?

Quantitative discrete

Quantitative continuous

Qualitative nominal

Qualitative ordinal

	User	Trail	Rating	Difficulty	Time Taken
0	Bob Honey	Piedmont Park Loop	2	Easy	10
1	Susie Thomas	Fire Trails	4.3	Very Hard	255
2	Josh Loop	Piedmont Park Loop	4	Easy	20
3	Susie Thomas	Inspiration Point	3	Medium	95
4	Bob Honey	Fire Trails	4	Hard	150

(d) (1.0 pt) What type of variable is Difficulty in the `user_hikes` DataFrame?

- ○ Quantitative discrete
- ○ Quantitative continuous
- ○ Qualitative nominal
- ● Qualitative ordinal

"Very hard", for example, isn't a number, so this isn't quantitative. The different levels of Difficulty have some sense of ordering, and so this is ordinal.

Regular Expressions

A *formal language* is a set of strings, typically described implicitly.

- Example: “The set of all strings of length < 10 that contain ‘horse’”

A *regular language* is a formal language that can be described by a *regular expression* (which we will define soon).

Example: `[0-9]{3}-[0-9]{2}-[0-9]{4}`

The language of SSNs is described by this regular expression.

3 of any digit, then a dash, then 2 of any digit, then a dash, then 4 of any digit.

```
text = "My social security number is 123-45-6789.";
pattern = r"[0-9]{3}-[0-9]{2}-[0-9]{4}"
re.findall(pattern, text)
```

Regular Expression Syntax

The four basic operations for regular expressions.

- Can technically do anything with just these basic four (albeit tediously).

operation	order	example	matches	does not match
concatenation	3	AABAAB	AABAAB	every other string
or	4	AA|BAAB	AA BAAB	every other string
closure (zero or more)	2	AB*A	AA ABBBBBBA	AB ABABA
parenthesis	1	A(A|B)AAB	AAAAB ABAAB	every other string
		(AB)*A	A ABABABABA	AA ABBA

Order of Operations in Regexes

`m(uu(uu)*|oo(oo)*)n`

- Matches starting with m and ending with n, with either of the following in the middle:
  - `uu(uu)*`
  - `oo(oo)*`

Match examples:

`muun`
`muuuun`
`moon`
`moooon`

`m(uu(uu)*)|(oo(oo)*)n`

- Matches either of the following
  - `m` followed by `uu(uu)*`
  - `oo(oo)*` followed by `n`

Match examples:

`muu`
`muuuu`
`oon`
`oooon`

In regexes `|` comes last.

Expanded Regex Syntax

operation	example	matches	does not match
any character (except newline)	`.U.U.U.`	`CUMULUS` `JUGULUM`	`SUCCUBUS` `TUMULTUOUS`
character class	`[A-Za-z][a-z]*`	`word` `Capitalized`	`camelCase` `4illegal`
at least one	`jo+hn`	`john` `jooooooohn`	`jhn` `jjohn`
zero or one	`joh?n`	`jon` `john`	`any other string`
repeated exactly {a} times	`j[aeiou]{3}hn`	`jaoehn` `jooohn`	`jhn` `jaeiouhn`
repeated from a to b times: {a,b}	`j[ou]{1,2}hn`	`john` `juohn`	`jhn` `jooohn`

Even More Regular Expression Syntax

operation	example	matches	does not match
built-in character classes	`\w+` `\d+`	`fawef` `231231`	`this person` `423 people`
character class negation	`[^a-z]+`	`PEPPERS3982` `17211!↑å`	`porch` `CLAmS`
escape character	`cow\.com`	`cow.com`	`cowscom`

Suppose you want to match one of our special characters like . or [or]

- In these cases, you must “escape” the character using the backslash.
- You can think of the backslash as meaning “take this next character literally”.

Summary

Today we saw many different string manipulation tools.

- There are many many more!
- With just this basic set of tools, you can do most of what you'll need.

basic python	re	pandas
	re.findall	df.str.findall
str.replace	re.sub	df.str.replace
str.split	re.split	df.str.split
'ab' in str	re.search	df.str.contain
len(str)		df.str.len
str[1:4]		df.str[1:4]

(b) **(4.0 pt)** In the string `extract_browser`, write a regular expression that extracts the name of the web browser (without its version) from a request.

For example:

```
>>> request_1 = """POST /fa20/syllabus HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: ds100.org"""
>>> re.findall(extract_browser, request_1)[0]
'Mozilla'

>>> request_2 = """GET /su19/syllabus HTTP/1.1
User-Agent: Safari/13.1 (Macintosh; Intel Mac OS X 10_10)
Host: data8.org"""
>>> re.findall(extract_browser, request_2)[0]
'Safari'

>>> request_3 = """GARBAGE /useless HTTP/1.1
User-Agent: Garbage/0.0 (garbage)
Host: garbage.ca"""
>>> re.findall(extract_browser, request_3)[0]
'Garbage'
```

Again, please write the regex as you would in Python with the form `extract_browser = r"..."`.

(b) **(4.0 pt)** In the string `extract_browser`, write a regular expression that extracts the name of the web browser (without its version) from a request.

For example:

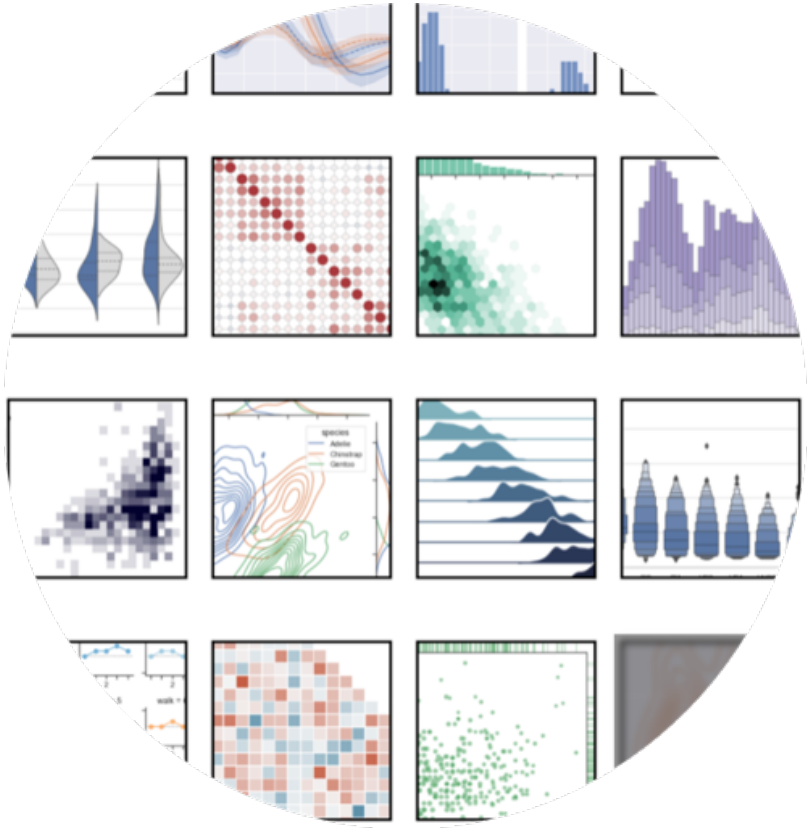
```
>>> request_1 = """POST /fa20/syllabus HTTP/1.1
User-Agent: Mozilla/4.0 (compatible; MSIE5.01; Windows NT)
Host: ds100.org"""
>>> re.findall(extract_browser, request_1)[0]
'Mozilla'

>>> request_2 = """GET /su19/syllabus HTTP/1.1
User-Agent: Safari/13.1 (Macintosh; Intel Mac OS X 10_10)
Host: data8.org"""
>>> re.findall(extract_browser, request_2)[0]
'Safari'

>>> request_3 = """GARBAGE /useless HTTP/1.1
User-Agent: Garbage/0.0 (garbage)
Host: garbage.ca"""
>>> re.findall(extract_browser, request_3)[0]
'Garbage'
```

Again, please write the regex as you would in Python with the form `extract_browser = r"..."`.

```
extract_browser = r"User-Agent: (\w+)\/.*"
```

Data Science (COSE471) Spring 2021

Visualizations

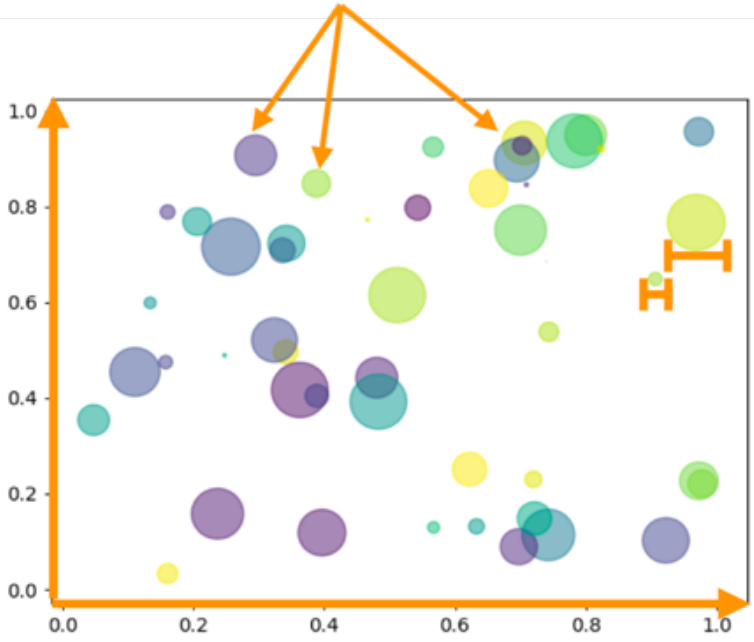


Dept. of Computer Science and Engineering

Korea University

Outline

- Introduction
  - Encoding
  - Distribution
- Types of Visualizations
  - Bar plots
  - Rug plots, histograms, density curves
  - Describing quantitative distributions
  - Box plots and violin plots
- Principles of Visualization
  - Scale
  - Conditioning
  - Perception
  - Context
  - Smoothing
  - Transformation



How many variables are we encoding here?

Answer: 4.

- x
- y
- area
- color

Bar plots

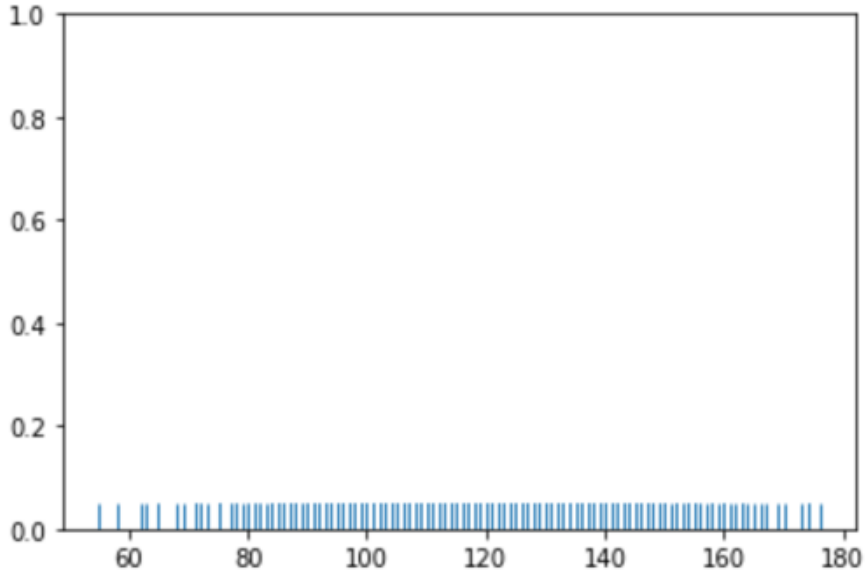
- Bar plots are the most common way of displaying the distribution of a qualitative (**categorical**) variable.
  - For example, the proportion of adults in the upper, middle, and lower classes.

- They are also used to display a numerical variable that has been measured on individuals in different categories.
  - For example, the average GPAs of students in several majors.
  - Not a distribution! But bar plots still make sense.

- Lengths encode values.
  - Widths encode **nothing!**
  - Color could indicate a sub-category (but not necessarily).

Rug plot

- Rug plots are used to show the distribution of a single quantitative (**numerical**) variable.
- They show us each and every value!
- Issues with rug plots:
  - Too much detail.
  - Hard to see the bigger picture.
  - **Overplotting**.
    - How many birth weights were at 120?
    - Can't tell – they're all on top of each other.




```
sns.rugplot(bweights)
```

Histograms

- Histograms can be thought of as a smoothed version of a rug plot.
  - Lose granularity, but gain interpretability.
- Horizontal axis: the number line, divided into **bins**.
- **Areas represent proportions**!
  - Total area = 1 (or 100%).
- Units of height: proportion per unit on the x-axis.
  - Can be seen by dividing the above equation by “width of bin”.

$$\text{proportion in bin} = \text{width of bin} \cdot \text{height of bar}$$

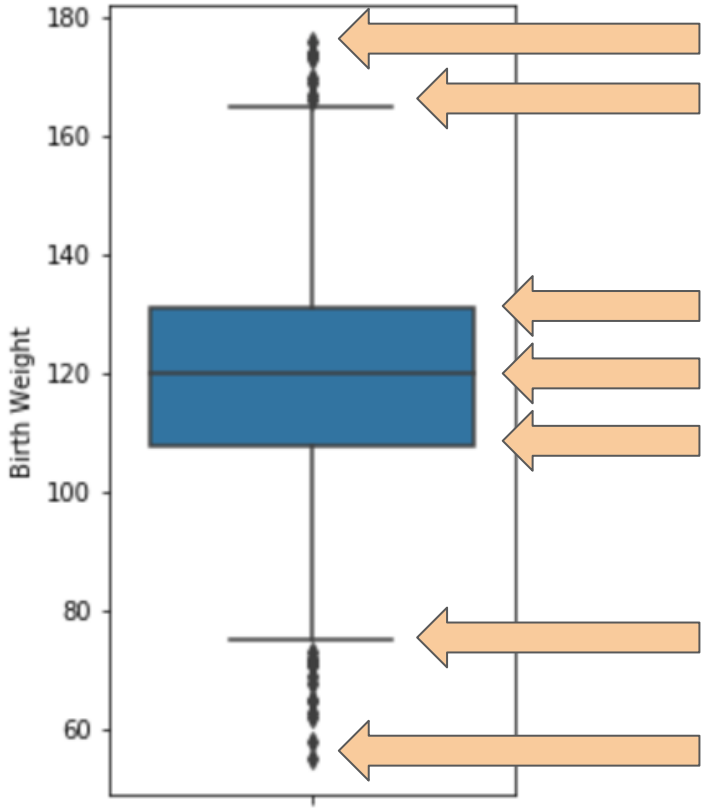
Describing distributions

One of the benefits of a histogram or density curve is that they show us the “bigger picture” of our distribution (something we don’t get with a rug plot).

Some of the terminology we use to describe distributions:

- **Modes.**
- **Skewness.**
  - Skewed left vs skewed right.
- **Tails.**
  - Left tail vs right tail.
- **Outliers.**
  - Define these arbitrarily.
  - Will see one definition in the next section.

Box plots



Outliers

Upper quartile plus 1.5*IQR

Upper quartile

Median

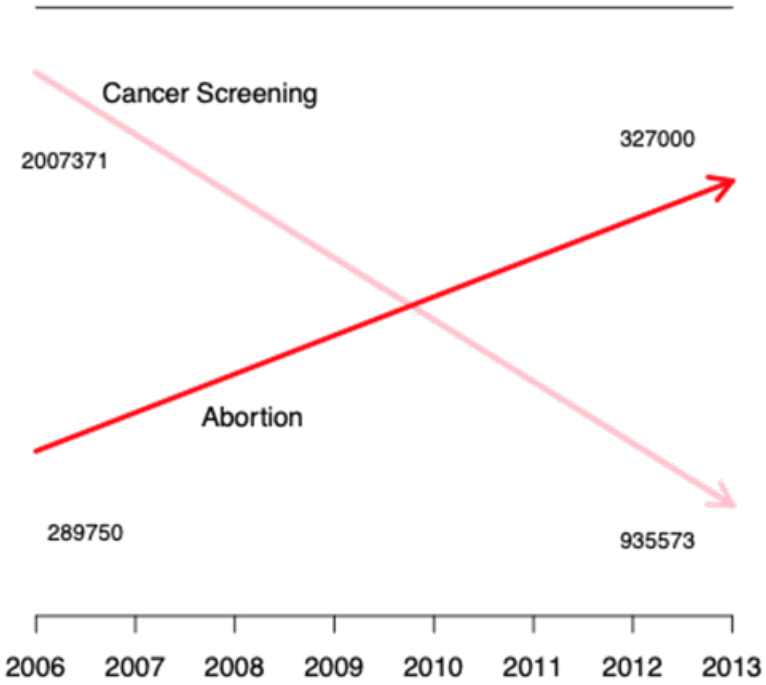
Lower quartile

Lower quartile minus 1.5*IQR

Outliers

Note: The box width is meaningless.

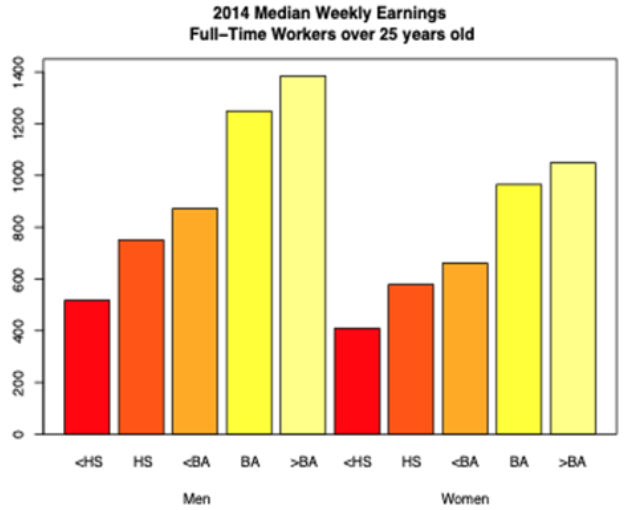
Keep axis scales consistent

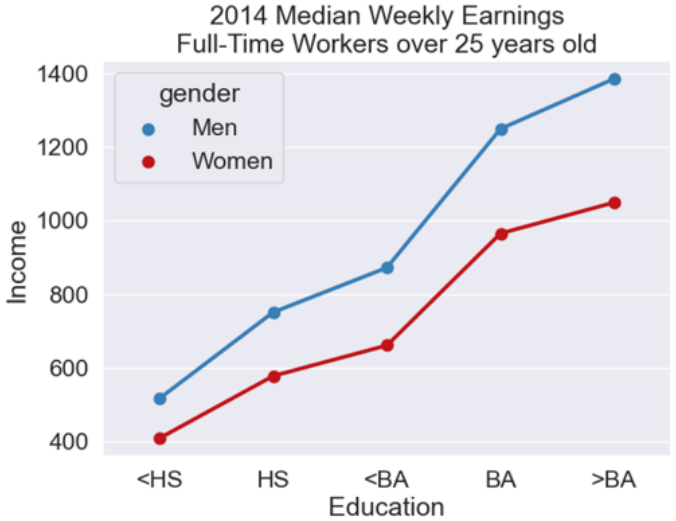


The scales for the two lines are completely different!

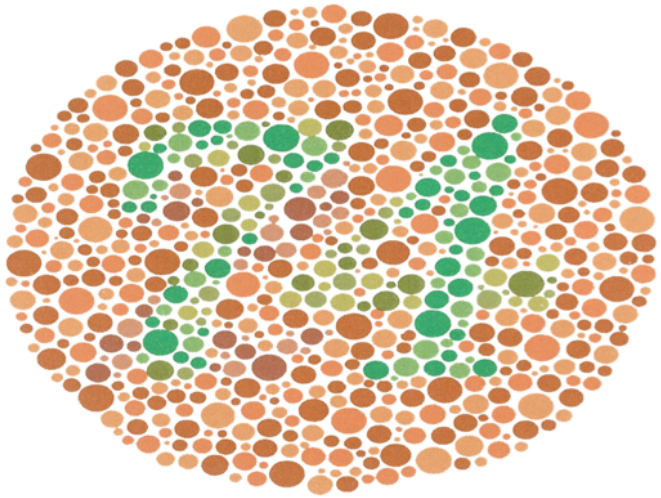
- 327000 is smaller than 935573, but appears to be way bigger.
- **Do not use two different scales for the same axis!**

Use conditioning to aid comparison



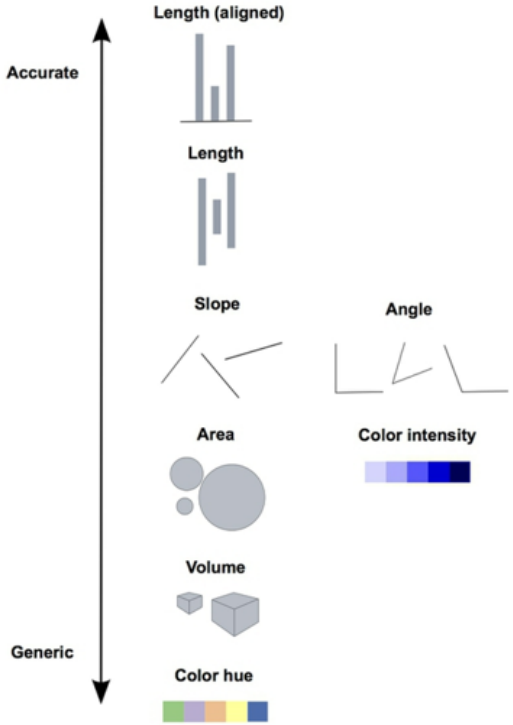


- Lines make it easy to see the large effect of having a BA on weekly earnings.
- Having two separate lines makes clear the wage difference between men and women.
    - It also highlights the fact that the wage difference increases, as education level does.



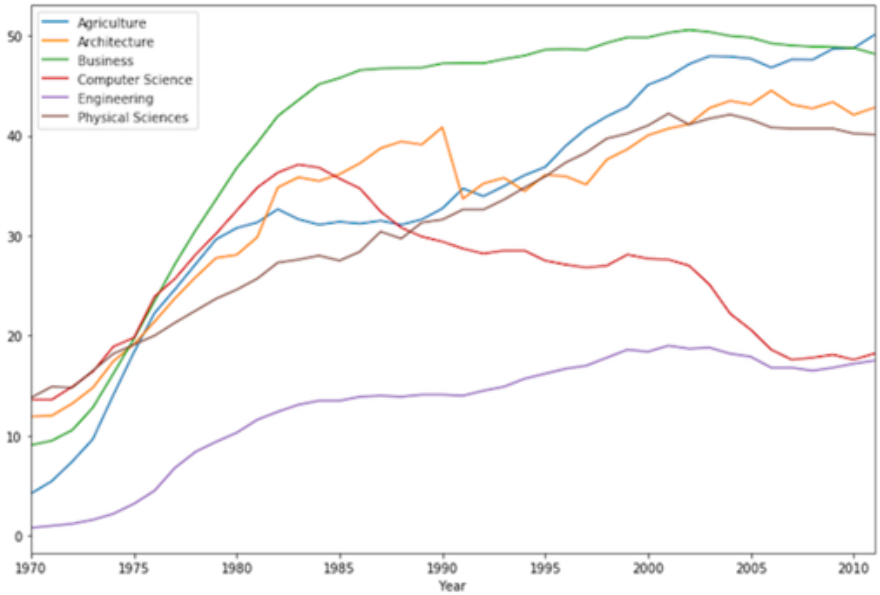
Choosing a set of colors which work together is a challenging task!

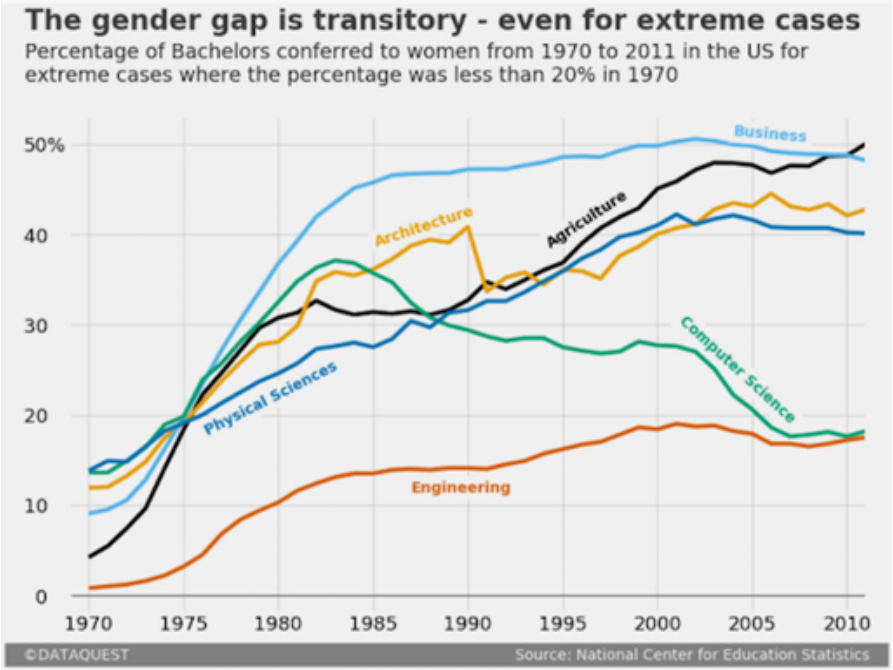
Perception of Color



The accuracy of our judgements depend on the type of marking.

Perception of Markings


Agriculture
Architecture
Business
Computer Science
Engineering
Physical Sciences
50
40
30
20
10
0
1970
1975
1980
1985
1990
1995
2000
2005
2010
Year


The gender gap is transitory - even for extreme cases
Percentage of Bachelors conferred to women from 1970 to 2011 in the US for extreme cases where the percentage was less than 20% in 1970
50%
40
30
20
10
0
1970
1975
1980
1985
1990
1995
2000
2005
2010
Business
Agriculture
Architecture
Computer Science
Physical Sciences
Engineering
©DATAQUEST
Source: National Center for Education Statistics

Kernel density estimation (KDE)

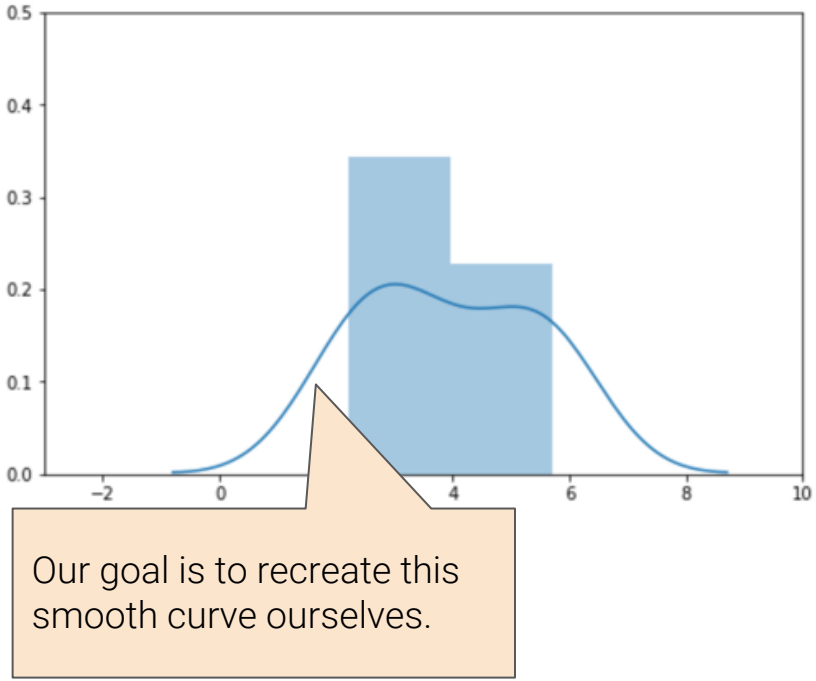
Kernel Density Estimation is used to estimate a **probability density function** (or density curve) from a set of data.

- Just like a histogram, a density function's total area must sum to 1.

To create a KDE:

- Place a **kernel** at each data point.
- Normalize kernels so that total area = 1.
- Sum all kernels together.

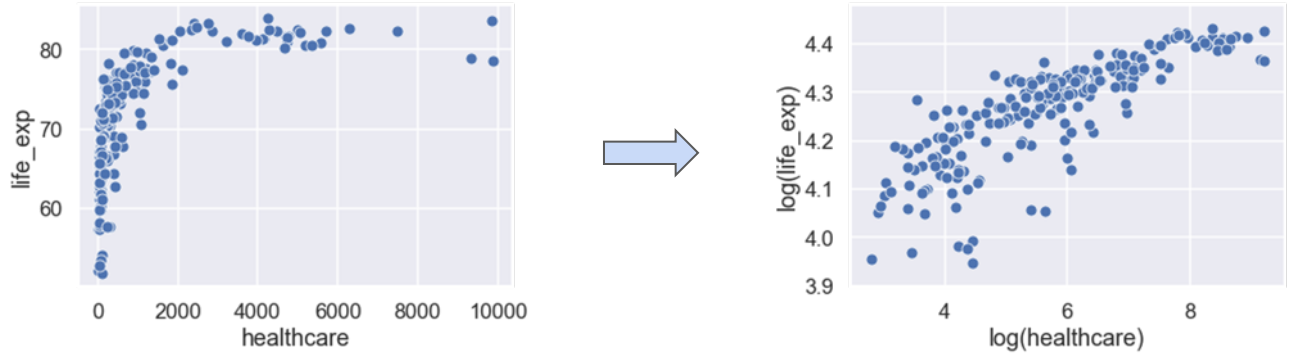
We also need to choose a kernel and **bandwidth.**



Why straighten relationships?

Now, we will look at how to **linearize** the scatter plot of two variables. Why?

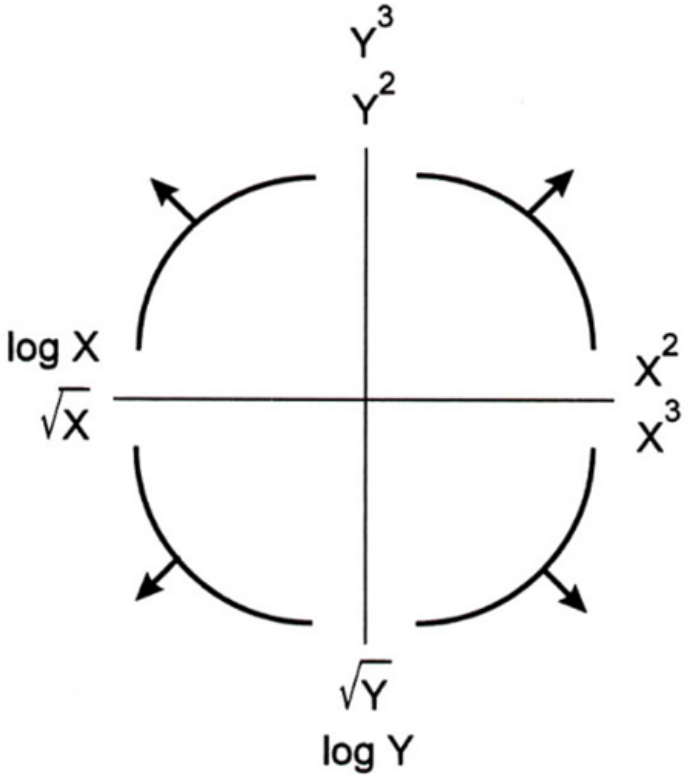
- If we know what transformation made our plot of y vs. x linear, we can “backtrack” to figure out the exact relationship between x and y.
- Linear relationships are particularly simple to interpret.
  - We know what slopes and intercepts mean.
  - We will be doing a lot of linear modeling – starting next lecture!

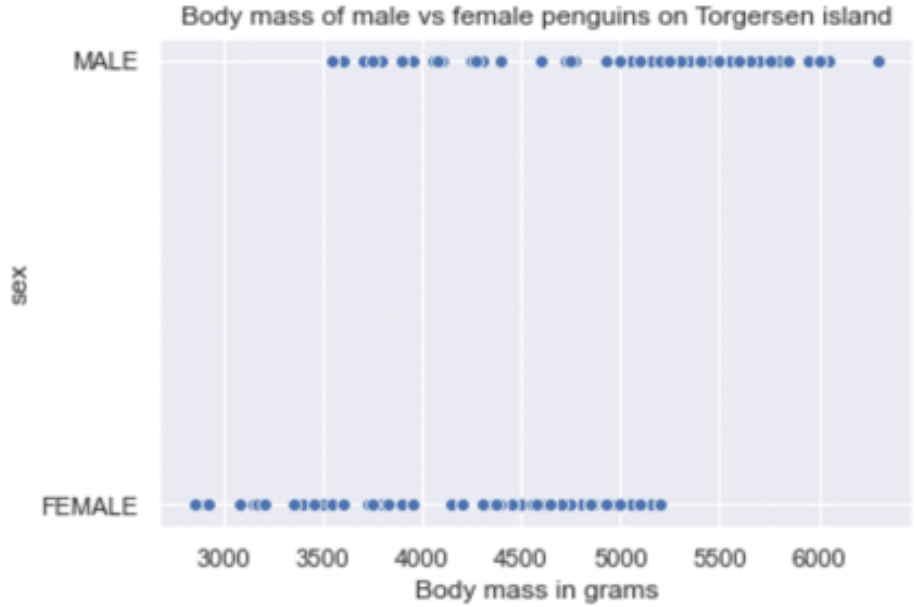


Tukey-Mosteller Bulge Diagram

This diagram can help us choose which transformation(s) to apply to our data in order to linearize it.

- There are multiple solutions. Some will fit better than others.
- sqrt and log make a value “smaller”. Raising to a value to a power makes it “bigger”.
- Each of these transformations equates to increasing or decreasing the scale of an axis.





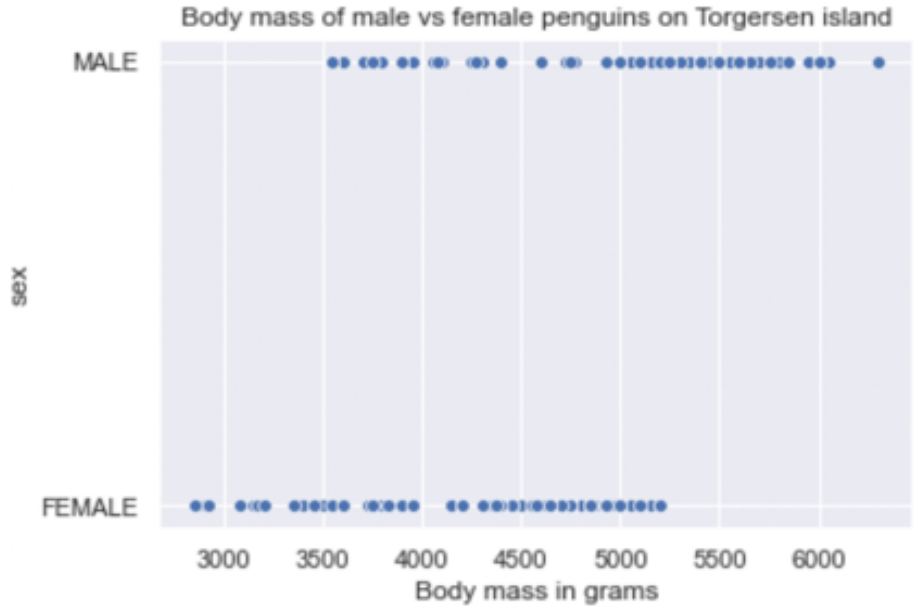
(a) (2.0 pt) The above visualization suffers from overplotting. Which of the following would be more appropriate types of visualization for this data? Select all that apply.

Side-by-side line plots

Overlaid density curves

Side-by-side boxplots

Bar chart

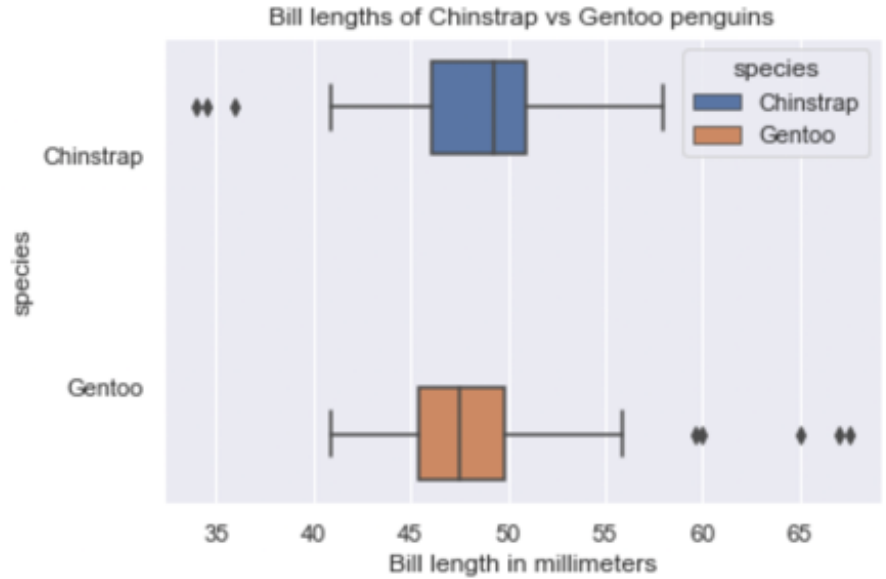


(a) (2.0 pt) The above visualization suffers from overplotting. Which of the following would be more appropriate types of visualization for this data? Select all that apply.

- ☐ Side-by-side line plots
- ■ Overlaid density curves
- ■ Side-by-side boxplots
- ☐ Bar chart

There is no sense of time in this data, so a line plot would not be appropriate. Instead, we want to see the distribution of the masses of both male and female penguins. To visualize a distribution we use density curves and boxplots, which is why overlaid density curves and side-by-side boxplots are appropriate. Bar charts would not be appropriate as they would only show one number for male penguins and one number for female penguins, which is not the entire distribution. (Bar charts are not to be confused with histograms.)

(c) (2.0 pt) Plot of bill lengths of Chinstrap vs Gentoo penguins:



We are now using a subset of the data, and visualizing the bill lengths of two penguin species using side-by-side box plots.

Which of these numbers is closest to the inter-quartile range of the Gentoo penguin bill lengths visualized above, in millimeters?

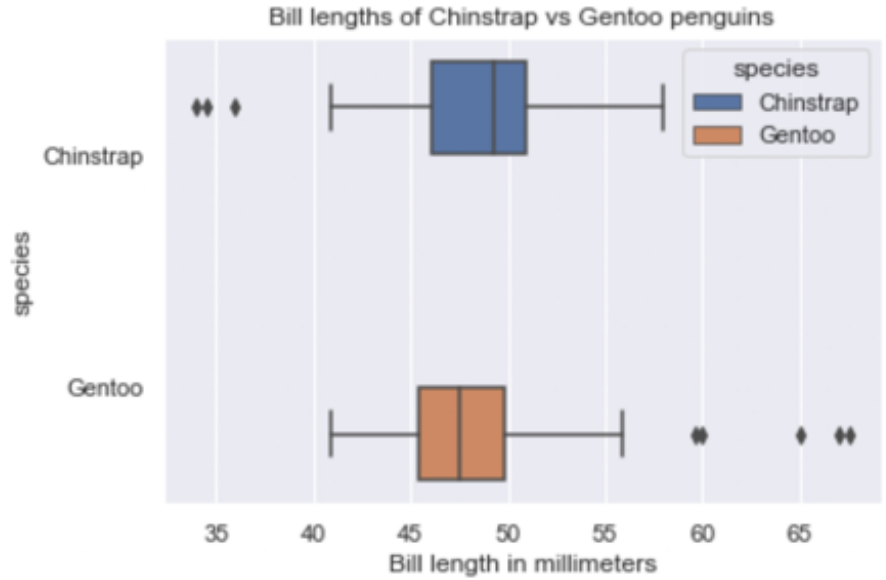
2.5

5

10

15

(c) (2.0 pt) Plot of bill lengths of Chinstrap vs Gentoo penguins:



We are now using a subset of the data, and visualizing the bill lengths of two penguin species using side-by-side box plots.

Which of these numbers is closest to the inter-quartile range of the Gentoo penguin bill lengths visualized above, in millimeters?

- ○ 2.5
- ● 5
- ○ 10
- ○ 15

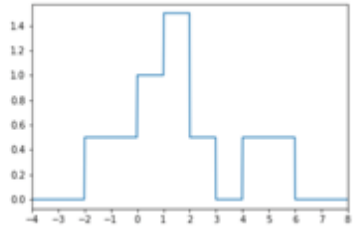
The inter-quartile range is the length of the box in a boxplot. The Gentoo box spans from roughly 45 mm to 50 mm, so the IQR is roughly 50 mm - 45 mm = 5 mm.

(a) (3.0 pt) For this question, suppose we have a dataset $X = [x_1, x_2, x_3, x_4, x_5] = [1, 1, -1, 5, 2]$.

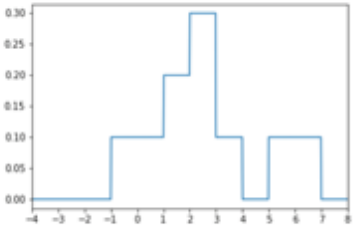
Which of the following is the estimated density of X with the Boxcar kernel and bandwidth parameter $\alpha = 2$? Recall that the boxcar kernel is

$$K_\alpha(x, x_i) = \begin{cases} \frac{1}{\alpha} & |x - x_i| \leq \frac{\alpha}{2} \\ 0 & \text{otherwise} \end{cases}$$

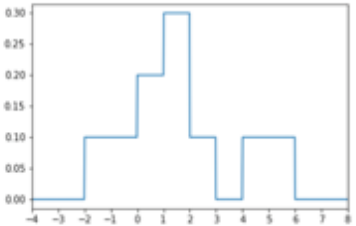
○



○



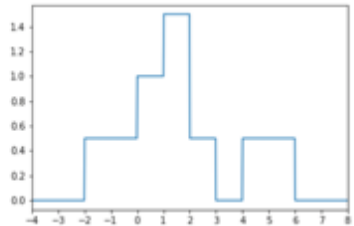
●

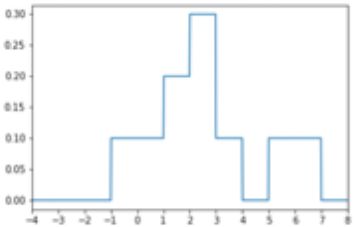


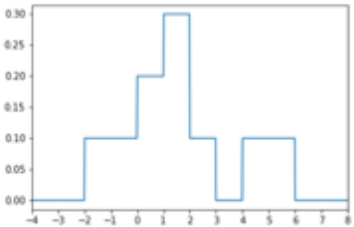
(a) (3.0 pt) For this question, suppose we have a dataset $X = [x_1, x_2, x_3, x_4, x_5] = [1, 1, -1, 5, 2]$.

Which of the following is the estimated density of X with the Boxcar kernel and bandwidth parameter $\alpha = 2$? Recall that the boxcar kernel is

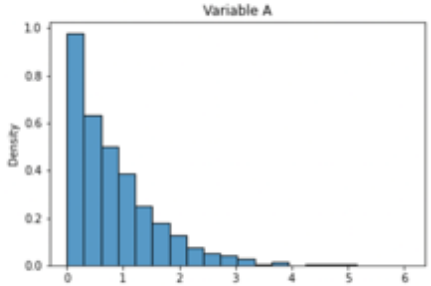
$$K_\alpha(x, x_i) = \begin{cases} \frac{1}{\alpha} & |x - x_i| \leq \frac{\alpha}{2} \\ 0 & \text{otherwise} \end{cases}$$

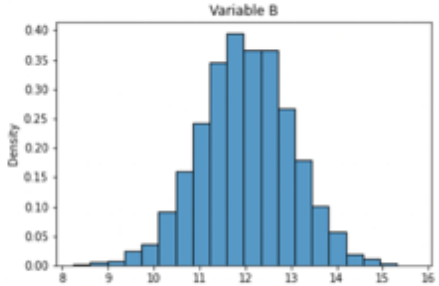


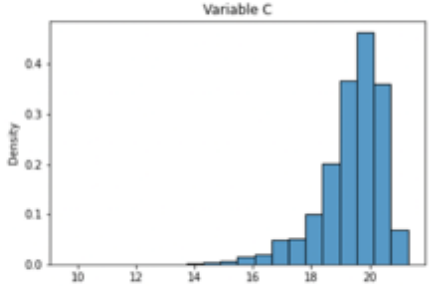




(b) (2.0 pt) Below, we show the distributions of three variables.







Which of the above distributions would be made more symmetric by applying a log transformation to the x-axis? Select all that apply.

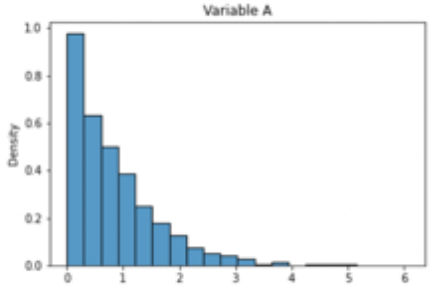
Variable A

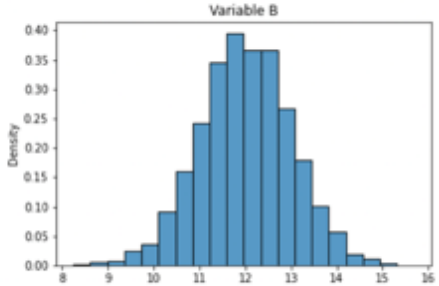
Variable B

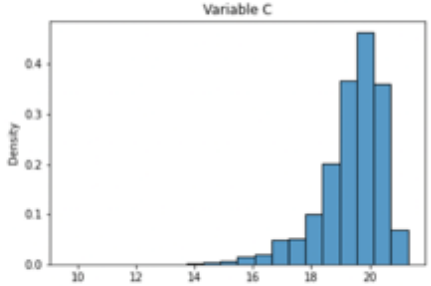
Variable C

None of the above

(b) (2.0 pt) Below, we show the distributions of three variables.







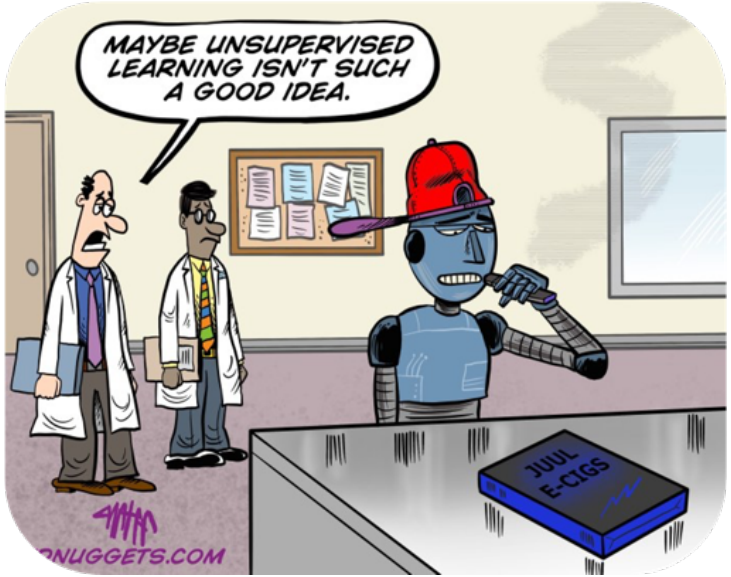
Which of the above distributions would be made more symmetric by applying a log transformation to the x-axis? Select all that apply.

- [x] Variable A
- [ ] Variable B
- [ ] Variable C
- [ ] None of the above

Data that is right-skewed can be made more symmetric with a log transformation on the x-axis. Only Variable A's distribution is right-skewed. Applying a log transformation to either of the other variables would make their distributions more left-skewed.

Data Science (COSE471) Spring 2021

Modeling

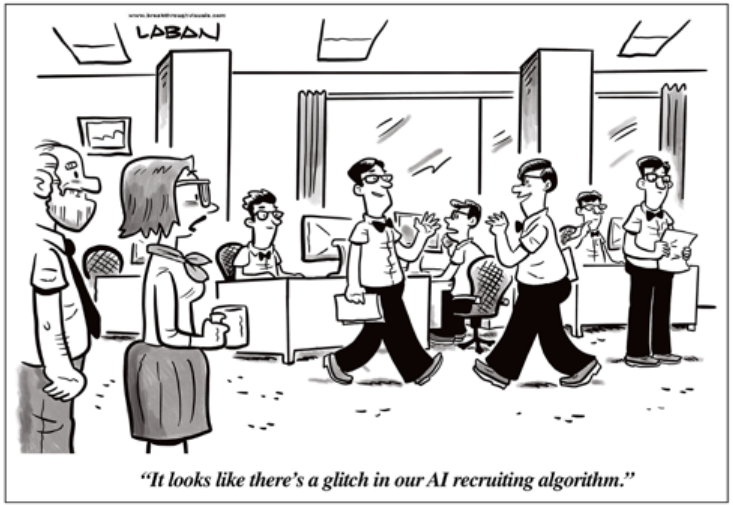


Dept. of Computer Science and Engineering

Korea University

The essence of learning from data

- We have data
- A pattern exists therein
- We cannot pin it down analytically



A simple model – the constant model

One choice of model would be to ignore any relationships between variables, and predict the same number for each individual – i.e., predicting a constant.

- We call this a summary statistic because it summarizes the data in our sample.
- For instance, tips given at restaurants likely depend on the total bill price, the time of day, how generous the customers are feeling, etc.
    - Ignoring these factors is a simplifying assumption!

MSE and MAE

If we choose squared loss as our loss function, then average squared loss is typically referred to as **mean squared error (MSE)**, and is of the following form:

$$MSE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} (y_i - \hat{y}_i)^2$$

If we choose absolute loss as our loss function, then average absolute loss is typically referred to as **mean absolute error (MAE)**, and is of the following form:

$$MAE(y, \hat{y}) = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i|$$

These definitions hold true, regardless of our model. We want to **minimize** these quantities.

MSE minimization using calculus

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

$$\implies \frac{d}{d\theta}R(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{d}{d\theta}(y_i - \theta)^2 = \frac{1}{n}\sum_{i=1}^{n}(-2)(y_i - \theta) = \frac{-2}{n}\sum_{i=1}^{n}(y_i - \theta)$$

Setting this term to 0, we have:

we can separate sums

$$0 = \frac{-2}{n}\sum_{i=1}^{n}(y_i - \theta)$$

$$0 = \sum_{i=1}^{n}(y_i - \theta) = \sum_{i=1}^{n} y_i - \sum_{i=1}^{n}\theta = \sum_{i=1}^{n} y_i - n\theta$$

c + c + ... + c = n * c

$$n\theta = \sum_{i=1}^{n} y_i$$

$$\implies \hat{\theta} = \frac{\sum_{i=1}^{n} y_i}{n} = \mathbf{mean}(y)$$

Thus, with squared loss and the constant model, the sample mean minimizes MSE.

MAE minimization using calculus

$$\frac{d}{d\theta}|y_i - \theta| = \begin{cases} -1 & \text{if } \theta < y_i \\ 1 & \text{if } \theta > y_i \end{cases}$$

From here, we again use the fact that the derivative of a sum is a sum of derivatives:

$$\frac{d}{d\theta}R(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{d}{d\theta}|y_i - \theta|$$

$$= \frac{1}{n}\left[\sum_{\theta<y_i}(-1) + \sum_{\theta>y_i}1\right]$$

Add -1 for each time an observation y_i is greater than our choice of theta.

Add 1 for each time an observation y_i is less than our choice of theta.

MAE minimization using calculus

Setting this derivative equal to 0:

$$0 = \frac{1}{n}\left[\sum_{\theta<y_i}(-1) + \sum_{\theta>y_i} 1\right]$$

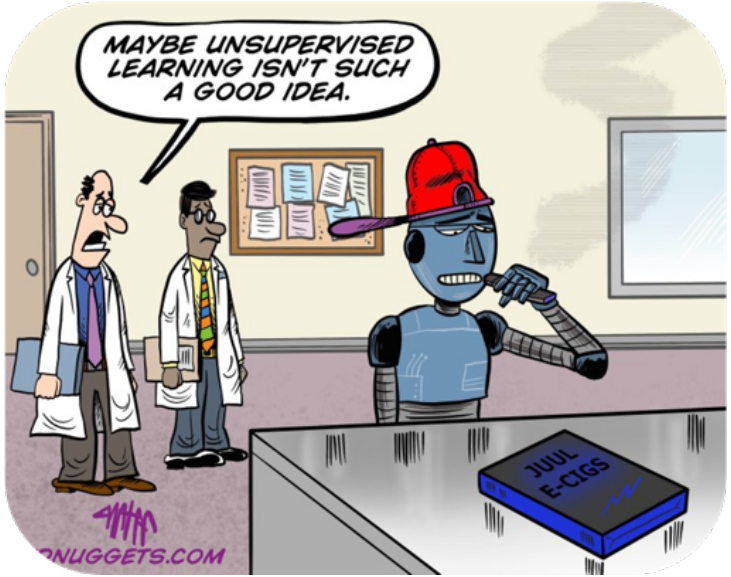
$$0 = -\sum_{\theta<y_i} 1 + \sum_{\theta>y_i} 1$$

$$\sum_{\theta<y_i} 1 = \sum_{\theta>y_i} 1$$

The last line is telling us that in order for our MAE to be minimized, we need to choose a theta such that **the number of observations less than theta** needs to be equal to **the number of observations greater than theta.**

Data Science (COSE471) Spring 2021

The Learning Problem



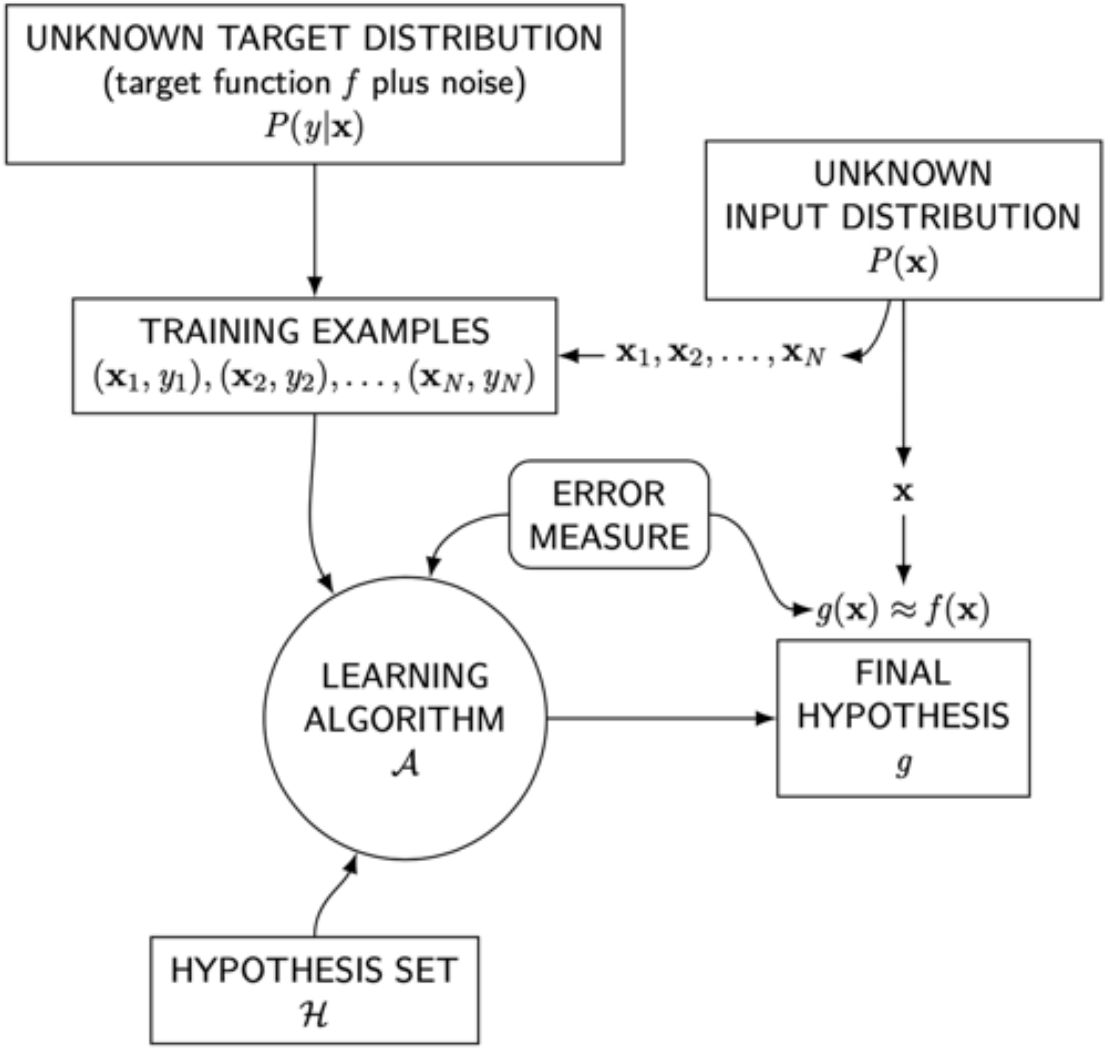
Dept. of Computer Science and Engineering

Korea University

Outline

- Introduction
  - Learning from Data
  - Problem Setup
  - A Simple Learning Model
  - Perceptron
- Types of Learning
- Summary

The Big Picture



The ‘perceptron’

- this linear formula $h \in \mathcal{H}$ can be written more compactly as

$$h(\mathbf{x}) = \text{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) - \text{threshold}\right) \quad (1)$$

$$= \text{sign}\left(\left(\sum_{i=1}^{d} w_i x_i\right) + b\right) \quad (2)$$

where b is called the bias and $\text{sign}(s)^1 = \begin{cases} +1 & \text{if } s > 0 \\ -1 & \text{if } s < 0 \end{cases}$

[1] value of sign(s) when s=0 is a simple technicality we can ignore for now

How PLA works

- the perceptron implements

$$h(\mathbf{x}) = \text{sign}(\mathbf{w}^\top \mathbf{x})$$

- given the training set

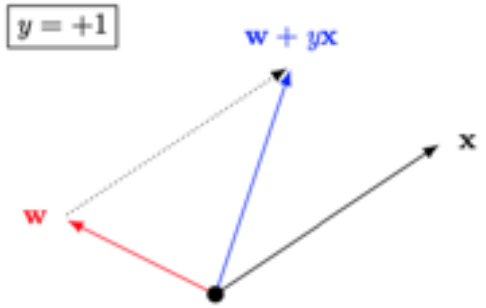
$$(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_N, y_N)$$

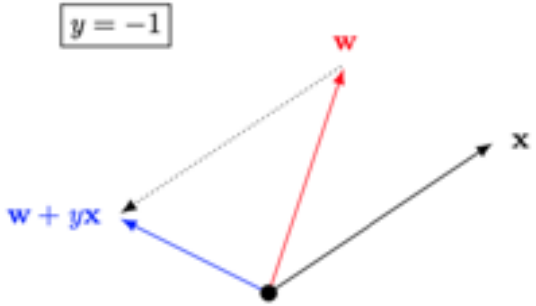
- PLA picks a misclassified point

$$\text{sign}(\mathbf{w}^\top \mathbf{x}_n) \neq y_n \qquad (4)$$

and updates the weight vector:

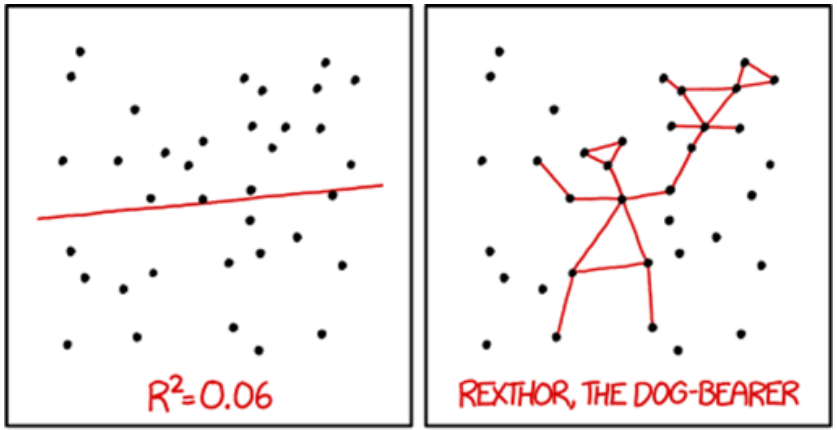
(5)





Data Science (COSE471) Spring 2021

Linear Regression



Dept. of Computer Science and Engineering

Korea University

- Thus, we resort to in-sample error instead:

$$\mathrm{E}_{\mathrm{in}}(h) = \frac{1}{N}\sum_{n=1}^{N}(h(\mathbf{x}_n) - y_n)^2$$

- In linear regression, $\boldsymbol{h}$ takes the form of
  - a linear combination of the components of $\mathbf{x}$:

$$h(\mathbf{x}) = \sum_{i=0}^{d} w_i x_i = \mathbf{w}^\top \mathbf{x}$$

  - $\mathbf{w}^\top \mathbf{x}$: also called signal

Matrix representation: in-sample error

- in-sample error is a function of $\mathbf{w}$ and data X, $\mathbf{y}$:

$$\begin{aligned}
\mathrm{E}_{\mathrm{in}}(\mathbf{w}) &= \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}^\top \mathbf{x}_n - y_n)^2 \\
&= \frac{1}{N}\left|\left|\begin{bmatrix} \mathbf{x}_1^\top \mathbf{w} - y_1 \\ \mathbf{x}_2^\top \mathbf{w} - y_2 \\ \vdots \\ \mathbf{x}_n^\top \mathbf{w} - y_n \end{bmatrix}\right|\right|^2 && (2) \\
&= \frac{1}{N}|| \qquad\quad ||^2 && (3) \\
&= \frac{1}{N}(\mathbf{w}^\top X^\top X\mathbf{w} - 2\mathbf{w}^\top X^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y}) && (4)
\end{aligned}$$

- $||\cdot||$: Euclidean norm of a vector
- Scalar $\mathbf{y}^\top X\mathbf{w} = \left(\mathbf{w}^\top X^\top \mathbf{y}\right)^\top = \mathbf{w}^\top X^\top \mathbf{y}$

- Example

$$X = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}$$

$$\mathrm{E}_{\mathrm{in}}(\mathbf{w}) = \frac{1}{4}\sum_{i=1}^{4}(\mathbf{w}^\top \mathbf{x}_n - y_n)^2$$

$$= \frac{1}{4}\left\| \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ 1 & x_3 \\ 1 & x_4 \end{bmatrix} \begin{bmatrix} w_0 \\ w_1 \end{bmatrix} - \begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} \right\|^2$$

Getting the solution

- $\mathbf{w}_{\text{lin}}$: the solution to linear regression
  - derived by minimizing $\boldsymbol{E}_{\text{in}}(\mathbf{w})$ over all possible $\mathbf{w} \in \mathbb{R}^{d+1}$

$$\mathbf{w}_{\text{lin}} = \underset{\mathbf{w}\in\mathbb{R}^{d+1}}{\operatorname{argmin}} \mathrm{E}_{\text{in}}(\mathbf{w}) \tag{5}$$

$$= \underset{\mathbf{w}\in\mathbb{R}^{d+1}}{\operatorname{argmin}} \frac{1}{N}||X\mathbf{w} - \mathbf{y}||^2 \tag{6}$$

The solution

- From eq. (4) $\frac{1}{N}(\mathbf{w}^\top X^\top X\mathbf{w} - 2\mathbf{w}^\top X^\top \mathbf{y} + \mathbf{y}^\top \mathbf{y})$ (4)

$$\nabla \mathrm{E}_{\mathrm{in}}(\mathbf{w}) = \frac{2}{N}(X^\top X\mathbf{w} - X^\top \mathbf{y})$$

  - both $\mathbf{w}$ and $\boldsymbol{\nabla E}_{\mathrm{in}}(\mathbf{w})$ are column vectors

- Finally, one should solve for $\mathbf{w}$ that satisfies the _____ equations:

$$X^\top X\mathbf{w} = X^\top \mathbf{y}$$

- $\boldsymbol{X}^\top \boldsymbol{X}$ is invertible:

$$X^\top X = \begin{bmatrix} 20.00 & 23.92 \\ 23.92 & 29.29 \end{bmatrix} \Rightarrow (X^\top X)^{-1} = \begin{bmatrix} 2.15 & -1.76 \\ -1.76 & 1.47 \end{bmatrix}$$

- $(\boldsymbol{X}^\top \boldsymbol{X})\boldsymbol{X}^\top \boldsymbol{y}$ yields

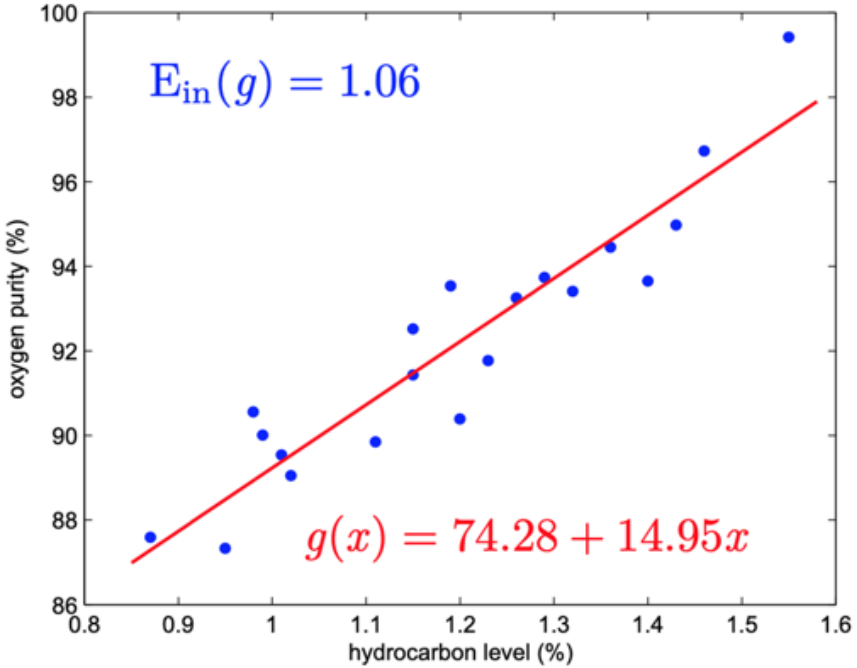
$$\mathbf{w}_{\text{lin}} = \begin{bmatrix} 74.28 \\ 14.95 \end{bmatrix}$$

- learned model:

$$\begin{aligned} g(x) &= \mathbf{w}_{\text{lin}}^\top \mathbf{x} \\ &= 74.28 + 14.95x \end{aligned}$$

- error:

$$\begin{aligned} \mathrm{E}_{\text{in}}(g) &= 1.06 \\ \mathrm{E}_{\text{out}}(g) &\approx 1.45 \end{aligned}$$



Hat matrix H (aka projection matrix)

- H relates to in-sample and out-of-sample errors
  - one of analysis tools developed in statistics
- H maps the vector of observed values to the vector of fitted values
  - if $\mathbf{y}$ and $\hat{\mathbf{y}}$ denote observed and fitted values, respectively

$$\hat{\mathbf{y}} = H\mathbf{y}$$

  - H: observed values $\longmapsto$ fitted values
  - matrix H 'puts a hat' on $\mathbf{y}$, hence the name

- therefore, estimate $\hat{\mathbf{y}}$ is a linear transformation of actual $\boldsymbol{y}$
  - through matrix multiplication with H where

$$H = X(X^\top X)^{-1}X^\top$$

- bottom line:

$$\begin{aligned}\hat{\mathbf{y}} &= X\mathbf{w}_{\text{lin}} \\ &= X\underbrace{(X^\top X)^{-1}X^\top \mathbf{y}}_{\mathbf{w}_{\text{lin}}} \\ &= \underbrace{X(X^\top X)^{-1}X^\top}_{H}\mathbf{y}\end{aligned}$$

$\therefore$ $\hat{\mathbf{y}}$: orthogonal projection of $\mathbf{y}$ onto the _____ of X

(a) (2.0 pt) Consider the simple linear regression model $\hat{y} = \theta_0 + \theta_1 x$.

Which of the following expressions evaluate to $\hat{\theta_1}$, the value of θ_1 that minimizes average squared loss for the simple linear regression model? (Note, r is the correlation coefficient. You can assume $\hat{\theta_0}$ is already defined, and that $\bar{x}, \bar{y}, \sigma_x, \sigma_y \neq 0$.)

$\hat{\theta_1} = \frac{r}{\sigma_x^2}$

$\hat{\theta_1} = \bar{y} - \hat{\theta_0}\bar{x}$

$\hat{\theta_1} = \frac{\bar{y} - \hat{\theta_0}}{\bar{x}}$

$\hat{\theta_1} = \frac{1}{n\sigma_x\sigma_y} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$

(a) (2.0 pt) Consider the simple linear regression model $\hat{y} = \theta_0 + \theta_1 x$.

Which of the following expressions evaluate to $\hat{\theta}_1$, the value of θ_1 that minimizes average squared loss for the simple linear regression model? (Note, r is the correlation coefficient. You can assume $\hat{\theta}_0$ is already defined, and that $\bar{x}, \bar{y}, \sigma_x, \sigma_y \neq 0$.)

- ○ $\hat{\theta}_1 = \frac{r}{\sigma_x^2}$
- ○ $\hat{\theta}_1 = \bar{y} - \hat{\theta}_0 \bar{x}$
- ● $\hat{\theta}_1 = \frac{\bar{y} - \hat{\theta}_0}{\bar{x}}$
- ○ $\hat{\theta}_1 = \frac{1}{n \sigma_x \sigma_y} \sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})$

Many of these options are intentionally tricky!

In class, we saw that $\hat{\theta}_0 = \bar{y} - \hat{\theta}_1 \bar{x}$. Rearranging for $\hat{\theta}_1$ yields $\hat{\theta}_1 = \frac{\bar{y} - \hat{\theta}_0}{\bar{x}}$ as required.

(b) (3.0 pt) Now consider two models:

- Model A: $\hat{y} = \theta_0 + \theta_1 x$
- Model B: $\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2$

We fit both models using the same dataset, and we fit both models using all data available to us. The x_1 feature in the second model is the same as the x feature in the first model. As usual, we determine optimal coefficients by minimizing average squared loss.

Let RMSE_A and RMSE_B represent the root mean squared error on our dataset for Model A and Model B, respectively. Furthermore, let R^2_A and R^2_B represent the Multiple R^2 coefficient values for Model A and Model B, respectively. Lastly, let $\hat{\theta_1}_{,A}$ and $\hat{\theta_1}_{,B}$ represent the optimal values of $\hat{\theta_1}$ for Model A and Model B, respectively.

Which of the following statements are guaranteed to be true? Select all that apply.

$\text{RMSE}_A \geq \text{RMSE}_B$

$\text{RMSE}_A \leq \text{RMSE}_B$

$R^2_A \geq R^2_B$

$R^2_A \leq R^2_B$

$\hat{\theta_1}_{,A} = \hat{\theta_1}_{,B}$

$\hat{\theta_1}_{,A} \neq \hat{\theta_1}_{,B}$

None of the above

(b) (3.0 pt) Now consider two models:

- Model A: $\hat{y} = \theta_0 + \theta_1 x$
- Model B: $\hat{y} = \theta_0 + \theta_1 x_1 + \theta_2 x_2$

We fit both models using the same dataset, and we fit both models using all data available to us. The x_1 feature in the second model is the same as the x feature in the first model. As usual, we determine optimal coefficients by minimizing average squared loss.

Let RMSE_A and RMSE_B represent the root mean squared error on our dataset for Model A and Model B, respectively. Furthermore, let R^2_A and R^2_B represent the Multiple R^2 coefficient values for Model A and Model B, respectively. Lastly, let $\hat{\theta_{1,A}}$ and $\hat{\theta_{1,B}}$ represent the optimal values of $\hat{\theta_1}$ for Model A and Model B, respectively.

Which of the following statements are guaranteed to be true? Select all that apply.

- [x] $\text{RMSE}_A \geq \text{RMSE}_B$
- [ ] $\text{RMSE}_A \leq \text{RMSE}_B$
- [ ] $R^2_A \geq R^2_B$
- [x] $R^2_A \leq R^2_B$
- [ ] $\hat{\theta_{1,A}} = \hat{\theta_{1,B}}$
- [ ] $\hat{\theta_{1,A}} \neq \hat{\theta_{1,B}}$
- [ ] None of the above

As we add features, RMSE on our training data either stays the same or goes down, it cannot go up. Similarly, as we add features, our R^2 either stays the same or goes up, it cannot go down. In general, there is no direct relationship between the coefficient on a particular feature in two different models. They may or may not be different.

(c) (3.0 pt) For the remainder of this question, we will use the multiple linear regression model, which is of the form

$$\hat{y} = x \cdot \theta = \sum_{j=0}^{p} \theta_j x_j$$

As in class, assume:

- $\mathbb{Y}$ is a vector containing our observed responses (i.e. true y's).
- $\mathbb{X}$ is a design matrix whose first column is 1 (i.e. $x_0 = 1$ for all observations), and $\mathbb{X}_i$ represents the ith row of $\mathbb{X}$.
- We determine $\hat{\theta}$ by minimizing average squared loss.

$\hat{\theta}$ is the minimizer of which of the following quantities? Select all that apply.

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

$\frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbb{X}_i \cdot \theta)^2$

$\sum_{i=1}^{n} (y_i - \mathbb{X}_i \cdot \theta)^2$

$\frac{1}{n} ||\mathbb{Y} - \mathbb{X}\theta||_2^2$

$||\mathbb{Y} - \mathbb{X}\theta||_2^2$

$\frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbb{X}_i \cdot \theta)$

$\sum_{i=1}^{n} (y_i - \mathbb{X}_i \cdot \theta)$

None of the above

(c) (3.0 pt) For the remainder of this question, we will use the multiple linear regression model, which is of the form

$$\hat{y} = x \cdot \theta = \sum_{j=0}^{p} \theta_j x_j$$

As in class, assume:

- $\mathbb{Y}$ is a vector containing our observed responses (i.e. true y's).
- $\mathbb{X}$ is a design matrix whose first column is 1 (i.e. $x_0 = 1$ for all observations), and $\mathbb{X}_i$ represents the ith row of $\mathbb{X}$.
- We determine $\hat{\theta}$ by minimizing average squared loss.

$\hat{\theta}$ is the minimizer of which of the following quantities? Select all that apply.

- ☐ $\sum_{i=1}^{n} (y_i - \theta)^2$
- ■ $\frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbb{X}_i \cdot \theta)^2$
- ■ $\sum_{i=1}^{n} (y_i - \mathbb{X}_i \cdot \theta)^2$
- ■ $\frac{1}{n} ||\mathbb{Y} - \mathbb{X}\theta||_2^2$
- ■ $||\mathbb{Y} - \mathbb{X}\theta||_2^2$
- ☐ $\frac{1}{n} \sum_{i=1}^{n} (y_i - \mathbb{X}_i \cdot \theta)$
- ☐ $\sum_{i=1}^{n} (y_i - \mathbb{X}_i \cdot \theta)$
- ☐ None of the above

The correct answers are all equivalent to either average squared loss or total squared loss (which is average squared loss without the $\frac{1}{n}$).

(d) (2.0 pt) Suppose we have $n = 4$ observations. What are possible valid values for the residual vector $e = \mathbb{Y} - \mathbb{X}\hat{\theta}$ given our model and $\hat{\theta}$? Select all that apply.

$e = [-1, 2, 3, -4]^T$

$e = [-1, 2, 3, 4]^T$

$e = [1, -1, 1, -1]^T$

$e = [0, 0.1, 0, 0.1]$

None of the above

(d) (2.0 pt) Suppose we have $n = 4$ observations. What are possible valid values for the residual vector $e = \mathbb{Y} - \mathbb{X}\hat{\theta}$ given our model and $\hat{\theta}$? Select all that apply.

- [x] $e = [-1, 2, 3, -4]^T$
- [ ] $e = [-1, 2, 3, 4]^T$
- [x] $e = [1, -1, 1, -1]^T$
- [ ] $e = [0, 0.1, 0, 0.1]$
- [ ] None of the above

Since our model has an intercept term as stated above, we know that the residuals must sum to 0.