EECS 398 W25 Midterm Review

February 23, 2025

Announcements

- The Midterm Exam is on **Tuesday, February 25th from 7-9PM**.
  - It covers Lectures 1-12, Homeworks 1-6, and Discussions 1-7.
- Midterm Review in lecture tomorrow (going over F24 Final 1-8.2).
- Study Tips
  - Go through lecture notebooks & homeworks to help make cheat sheet (one page, double-sided, **handwritten**).
  - Do discussion problems.
  - Take F24 Midterm and Problems 1-8.2 of F24 Final (besides SQL question).

Agenda

- We'll be working through https://study.practicaldsc.org/mt-review-sunday/index.html.
- We'll post these annotated slides and the recording after, along with enabling solutions on the study site for this worksheet.

Grouping, Querying, and Merging - Akanksha

The EECS 398 staff are looking into hotels — some in San Diego, for their family to stay at for graduation (and to eat Mexican food), and some elsewhere, for summer trips.

Each row of `hotels` contains information about a different hotel in San Diego. Specifically, for each hotel, we have:

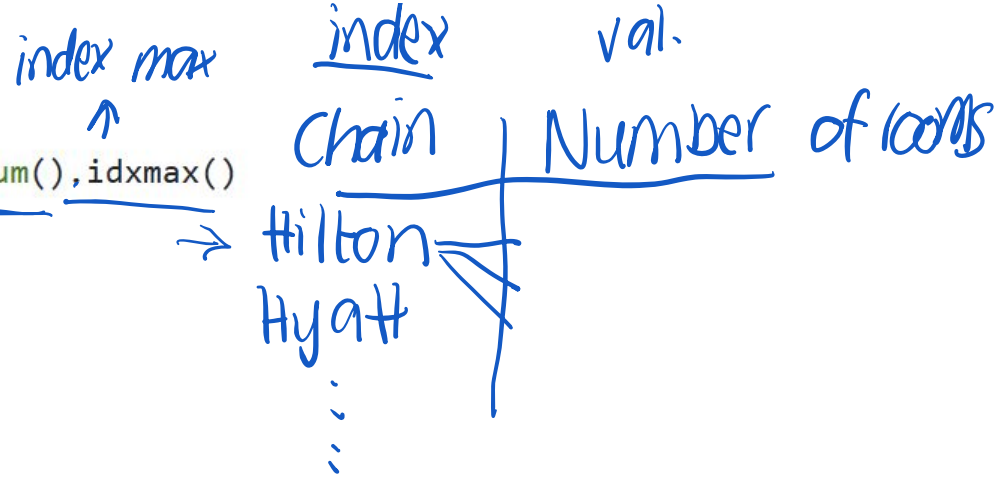
- `"Hotel Name" (str)`: The name of the hotel. **Assume hotel names are unique.**
- `"Location" (str):` The hotel's neighborhood in San Diego.
- `"Chain" (str):` The chain the hotel is a part of; either `"Hilton", "Marriott", "Hyatt", or "Other".` A hotel chain is a group of hotels owned or operated by a shared company.
- `"Number of Rooms" (int)`: The number of rooms the hotel has.

The first few rows of `hotels` are shown below, but hotels has many more rows than are shown.

	Hotel Name	Location	Chain	Number of Rooms
0	Hotel del Coronado	Coronado	Hilton	680
1	Manchester Grand Hyatt	Downtown	Hyatt	1628
2	Hilton San Diego Bayfront	Downtown	Hilton	1190
3	Pendry San Diego	Gaslamp Quarter	Other	317
4	The Westin San Diego Gaslamp Quarter	Gaslamp Quarter	Marriott	450
5	San Diego Marriott La Jolla	La Jolla	Marriott	462
6	La Valencia Hotel	La Jolla	Other	114
7	Coronado Island Marriott Resort & Spa	Coronado	Marriott	310

Now, consider the variable `summed`, defined below.

```
summed = hotels.groupby("Chain")["Number of Rooms"].sum(),idxmax()
```



Problem 1.1

What is `type(summed)`?

- `int`
- `str`
- `Series`
- `DataFrame`
- `DataFrameGroupBy`

	Hotel Name	Location	Chain	Number of Rooms
0	Hotel del Coronado	Coronado	Hilton	680
1	Manchester Grand Hyatt	Downtown	Hyatt	1628
2	Hilton San Diego Bayfront	Downtown	Hilton	1190
3	Pendry San Diego	Gaslamp Quarter	Other	317
4	The Westin San Diego Gaslamp Quarter	Gaslamp Quarter	Marriott	450
5	San Diego Marriott La Jolla	La Jolla	Marriott	462
6	La Valencia Hotel	La Jolla	Other	114
7	Coronado Island Marriott Resort & Spa	Coronado	Marriott	310

Problem 1.2

In one sentence, explain what the value of `summed` means. Phrase your explanation as if you had to give it to someone who is not a data science major; that is, don't say something like "it is the result of grouping `hotels` by `"Chain"`, selecting the `"Number of Rooms"` column, ...", but instead, give the value context.

```
summed = hotels.groupby("Chain")["Number of Rooms"].sum().idxmax()
```

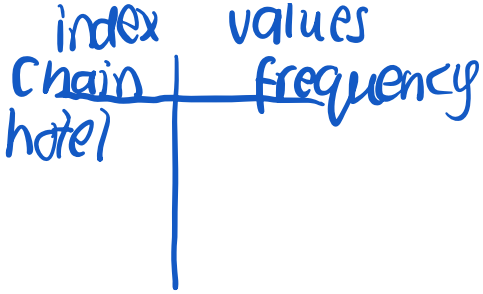
hotel chain with the highest total number of rooms

	Hotel Name	Location	Chain	Number of Rooms
0	Hotel del Coronado	Coronado	Hilton	680
1	Manchester Grand Hyatt	Downtown	Hyatt	1628
2	Hilton San Diego Bayfront	Downtown	Hilton	1190
3	Pendry San Diego	Gaslamp Quarter	Other	317
4	The Westin San Diego Gaslamp Quarter	Gaslamp Quarter	Marriott	450
5	San Diego Marriott La Jolla	La Jolla	Marriott	462
6	La Valencia Hotel	La Jolla	Other	114
7	Coronado Island Marriott Resort & Spa	Coronado	Marriott	310

Problem 1.3

Consider the variable `curious`, defined below.

```
curious = frame["Chain"].value_counts().idxmax()
```



Fill in the blank: `curious` is guaranteed to be equal to `summed` only if `frame` has one row for every ___ in San Diego.

- hotel
- hotel chain
- hotel room
- neighborhood

	Hotel Name	Location	Chain	Number of Rooms
0	Hotel del Coronado	Coronado	Hilton	680
1	Manchester Grand Hyatt	Downtown	Hyatt	1628
2	Hilton San Diego Bayfront	Downtown	Hilton	1190
3	Pendry San Diego	Gaslamp Quarter	Other	317
4	The Westin San Diego Gaslamp Quarter	Gaslamp Quarter	Marriott	450
5	San Diego Marriott La Jolla	La Jolla	Marriott	462
6	La Valencia Hotel	La Jolla	Other	114
7	Coronado Island Marriott Resort & Spa	Coronado	Marriott	310

Problem 1.4

Fill in the blanks so that `popular_areas` is an array of the names of the unique neighborhoods that have at least 5 hotels and at least 1000 hotel rooms.

```
f = lambda df: __(i)__
popular_areas = (hotels
                 .groupby(__(ii)__)
                 .__(iii)__
                 __(iv)__)
```

1. What goes in blank (i)?

2. What goes in blank (ii)?

   - `"Hotel Name"`
   - `"Location"`
   - `"Chain"`
   - `"Number of Rooms"`

3. What goes in blank (iii)?

   - `agg(f)`
   - `filter(f)`
   - `transform(f)`

4. What goes in blank (iv)?

	Hotel Name	Location	Chain	Number of Rooms
0	Hotel del Coronado	Coronado	Hilton	680
1	Manchester Grand Hyatt	Downtown	Hyatt	1628
2	Hilton San Diego Bayfront	Downtown	Hilton	1190
3	Pendry San Diego	Gaslamp Quarter	Other	317
4	The Westin San Diego Gaslamp Quarter	Gaslamp Quarter	Marriott	450
5	San Diego Marriott La Jolla	La Jolla	Marriott	462
6	La Valencia Hotel	La Jolla	Other	114
7	Coronado Island Marriott Resort & Spa	Coronado	Marriott	310

Problem 1.5

Consider the code below.

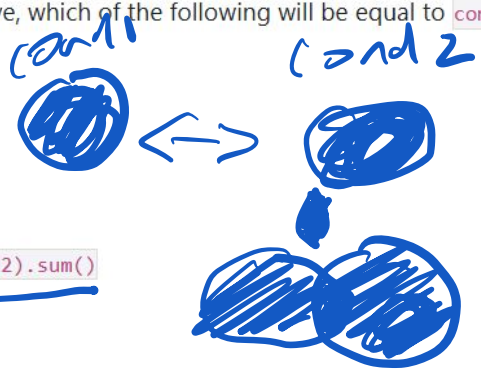
```
cond1 = hotels["Chain"] == "Marriott"
cond2 = hotels["Location"] == "Coronado"
combined = hotels[cond1].merge(hotels[cond2], on="Hotel Name", how=???)
```

1. If we replace `???` with `"inner"` in the code above, which of the following will be equal to `combined.shape[0]`?

   - `min(cond1.sum(), cond2.sum())`
   - `(cond1 & cond2).sum()`
   - `cond1.sum() + cond2.sum()`
   - `cond1.sum() + cond2.sum() - (cond1 & cond2).sum()`
   - `cond1.sum() + (cond1 & cond2).sum()`

2. If we replace `???` with `"outer"` in the code above, which of the following will be equal to `combined.shape[0]`?

   - `min(cond1.sum(), cond2.sum())`
   - `(cond1 & cond2).sum()`
   - `cond1.sum() + cond2.sum()`
   - `cond1.sum() + cond2.sum() - (cond1 & cond2).sum()`
   - `cond1.sum() + (cond1 & cond2).sum()`



	Hotel Name	Location	Chain	Number of Rooms
0	Hotel del Coronado	Coronado	Hilton	680
1	Manchester Grand Hyatt	Downtown	Hyatt	1628
2	Hilton San Diego Bayfront	Downtown	Hilton	1190
3	Pendry San Diego	Gaslamp Quarter	Other	317
4	The Westin San Diego Gaslamp Quarter	Gaslamp Quarter	Marriott	450
5	San Diego Marriott La Jolla	La Jolla	Marriott	462
6	La Valencia Hotel	La Jolla	Other	114
7	Coronado Island Marriott Resort & Spa	Coronado	Marriott	310

Random Simulations - Akanksha

Problem 2

Billina Records, a new record company focused on creating new TikTok audios, has its offices on the 23rd floor of a skyscraper with 75 floors (numbered 1 through 75). The owners of the building promised that 10 different random floors will be selected to be renovated.

Below, fill in the blanks to complete a simulation that will estimate the probability that Billina Records' floor will be renovated.

```
total = 0
repetitions = 10000
for i in np.arange(repetitions):
    choices = np.random.choice(__(a)__, 10, __(b)__)
    if __(c)__:
        total = total + 1
prob_renovate = total / repetitions
```

choices = [10 23 1,2,5,6,7,9 12,14]

[F, T, F, F]

0 1

What goes in blank (a)?

- np.arange(1, 75)
- np.arange(10, 75)
- np.arange(0, 76)
- np.arange(1, 76)

range is exclusive

What goes in blank (b)?

- replace=True
- replace=False

What goes in blank (c)?

- choices == 23
- choices is 23
- np.count_nonzero(choices == 23) > 0
- np.count_nonzero(choices) == 23
- choices.str.contains(23)

Joining 2 dataframes

↓

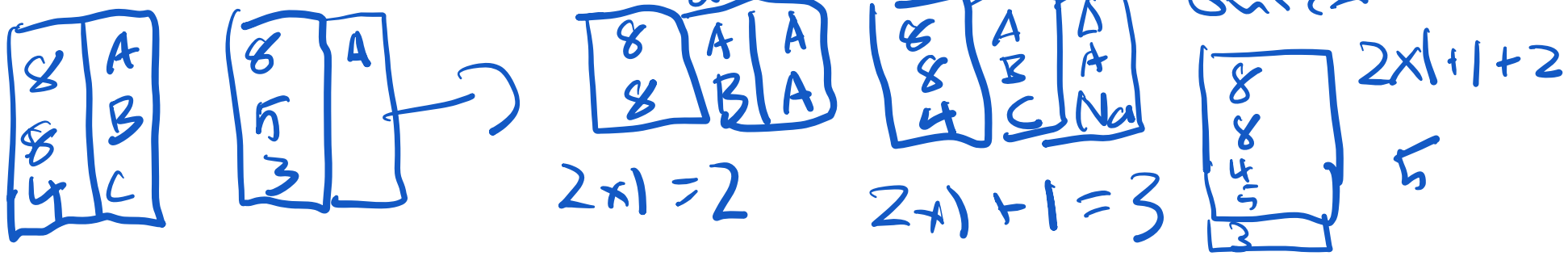
Merging - Caleb

rows from a merge

Inner Merge : $\sum$ # match in df1 x # match in df2

Left Merge: # rows from inner merge + # non-matches in left df

Outer Merge: # rows from inner merge + non-matches in df1 and left df



Suppose the DataFrame `today` consists of 15 rows — 3 rows for each of 5 different `"artist_names"`. For each artist, it contains the `"track_name"` for their three most-streamed songs today. For instance, there may be one row for `"olivia rodrigo"` and `"favorite crime"`, one row for `"olivia rodrigo"` and `"drivers license"`, and one row for `"olivia rodrigo"` and `"deja vu"`.

Another DataFrame, `genres`, is shown below in its entirety.

	artist_names	genre
0	harry styles	Pop
1	olivia rodrigo	Pop
2	glass animals	Alternative
3	drake	Hip-Hop/Rap
4	doja cat	Hip-Hop/Rap

Problem 3.1

Suppose we perform an **inner** merge between `today` and `genres` on `"artist_names"`. If the five `"artist_names"` in `today` are the same as the five `"artist_names"` in `genres`, what fraction of the rows in the merged DataFrame will contain `"Pop"` in the `"genre"` column? Give your answer as a simplified fraction.

6 rows with pop in genre / 15 rows total = 2/5

3 x 5

harry — Pop
harry — Pop
harry — Pop
olivia — Pop
olivia — Pop
olivia — Pop
glass — Alt

Suppose the DataFrame `today` consists of 15 rows — 3 rows for each of 5 different `"artist_names"`. For each artist, it contains the `"track_name"` for their three most-streamed songs today. For instance, there may be one row for `"olivia rodrigo"` and `"favorite crime"`, one row for `"olivia rodrigo"` and `"drivers license"`, and one row for `"olivia rodrigo"` and `"deja vu"`.

genres

	artist_names	genre
0	harry styles	Pop
1	olivia rodrigo	Pop
2	glass animals	Alternative
3	drake	Hip-Hop/Rap
4	doja cat	Hip-Hop/Rap

Problem 3.2

Suppose we perform an **inner** merge between `today` and `genres` on `"artist_names"`. Furthermore, suppose that the only overlapping `"artist_names"` between `today` and `genres` are `"drake"` and `"olivia rodrigo"`. What fraction of the rows in the merged DataFrame will contain `"Pop"` in the `"genre"` column? Give your answer as a simplified fraction.

3 rows with Pop in genre

6 rows total

3x2

$\frac{1}{2}$

olivia ~ Pop
olivia ~ Pop
olivia ~ Pop
drake ~ Hip
drake ~ Hip
drake ~ Hip

Suppose the DataFrame `today` consists of 15 rows — 3 rows for each of 5 different `"artist_names"`. For each artist, it contains the `"track_name"` for their three most-streamed songs today. For instance, there may be one row for `"olivia rodrigo"` and `"favorite crime"`, one row for `"olivia rodrigo"` and `"drivers license"`, and one row for `"olivia rodrigo"` and `"deja vu"`.

	artist_names	genre
0	harry styles	Pop
1	olivia rodrigo	Pop
2	glass animals	Alternative
3	drake	Hip-Hop/Rap
4	doja cat	Hip-Hop/Rap

genres

Problem 3.3

Suppose we perform an **outer** merge between `today` and `genres` on `"artist_names"`. Furthermore, suppose that the only overlapping `"artist_names"` between `today` and `genres` are `"drake"` and `"olivia rodrigo"`. What fraction of the rows in the merged DataFrame will contain `"Pop"` in the `"genre"` column? Give your answer as a simplified fraction.

6 Inner Merge

3 non-matches from genres

3 rows with 'Pop' from inner merge

1 row from Harry Styles non-match in genres, with Pop

4 non-matches from today

$\frac{4}{18} \rightarrow \boxed{\frac{2}{9}}$

18 total rows

Suppose the DataFrame `today` consists of 15 rows — 3 rows for each of 5 different `"artist_names"`. For each artist, it contains the `"track_name"` for their three most-streamed songs today. For instance, there may be one row for `"olivia rodrigo"` and `"favorite crime"`, one row for `"olivia rodrigo"` and `"drivers license"`, and one row for `"olivia rodrigo"` and `"deja vu"`.

	artist_names	genre
0	harry styles	Pop
1	olivia rodrigo	Pop
2	glass animals	Alternative
3	drake	Hip-Hop/Rap
4	doja cat	Hip-Hop/Rap

genres

Missing Value Imputation - Caleb

Listwise deletion: drop missing values from calculation

Mean imputation -> fill missing values with mean

Conditional mean imputation: fill missing values with mean of some group

Probabilistic Imputation: Replace missing values with a random sample of data.

The DataFrame `random_10` contains the `"track_name"` and `"genre"` of 10 randomly-chosen songs in Spotify's Top 200 today, along with their `"genre_rank"`, which is their rank in the Top 200 **among songs in their** `"genre"`. For instance, "the real slim shady" is the 20th-ranked Hip-Hop/Rap song in the Top 200 today. `random_10` is shown below in its entirety.

	track_name	genre rank	genre
0	good looking	7.0	Alternative
1	drowning (feat. kodak black)	NaN	Hip-Hop/Rap
2	the real slim shady	20.0	Hip-Hop/Rap
3	worldwide steppers	2.0	Hip-Hop/Rap
4	2055	5.0	Hip-Hop/Rap
5	drivers license	9.0	Pop
6	cinema	2.0	Pop
7	dos mil 16	4.0	Pop
8	happier than ever	NaN	Pop
9	bam bam (feat. ed sheeran)	NaN	Pop

The `"genre_rank"` column of `random_10` contains missing values. Below, we provide four different imputed `"genre_rank"` columns, each of which was created using a different imputation technique. On the next page, match each of the four options to the imputation technique that was used in the option.

Option A

genre rank	genre
7.0	Alternative
5.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
2.0	Pop
2.0	Pop

Option B

genre rank	genre
7.0	Alternative
7.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
7.0	Pop
7.0	Pop

Option C

genre rank	genre
7.0	Alternative
2.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
2.0	Pop
7.0	Pop

Option D

genre rank	genre
7.0	Alternative
9.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
5.0	Pop
5.0	Pop

Problem 4.1

In which option was unconditional mean imputation used?

mean of filled vals: 7

Option A

genre rank	genre
7.0	Alternative
5.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
2.0	Pop
2.0	Pop

Option B

genre rank	genre
7.0	Alternative
7.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
7.0	Pop
7.0	Pop

Option C

genre rank	genre
7.0	Alternative
2.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
2.0	Pop
7.0	Pop

Option D

genre rank	genre
7.0	Alternative
9.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
5.0	Pop
5.0	Pop

	track_name	genre rank	genre
0	good looking	7.0	Alternative
1	drowning (feat. kodak black)	NaN	Hip-Hop/Rap
2	the real slim shady	20.0	Hip-Hop/Rap
3	worldwide steppers	2.0	Hip-Hop/Rap
4	2055	5.0	Hip-Hop/Rap
5	drivers license	9.0	Pop
6	cinema	2.0	Pop
7	dos mil 16	4.0	Pop
8	happier than ever	NaN	Pop
9	bam bam (feat. ed sheeran)	NaN	Pop

Problem 4.2

In which option was mean imputation conditional on `"genre"` used?

Option A

genre rank	genre
7.0	Alternative
5.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
2.0	Pop
2.0	Pop

Option B

genre rank	genre
7.0	Alternative
7.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
7.0	Pop
7.0	Pop

Option C

genre rank	genre
7.0	Alternative
2.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
2.0	Pop
7.0	Pop

Option D

genre rank	genre
7.0	Alternative
9.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
5.0	Pop
5.0	Pop

	track_name	genre rank	genre
0	good looking	7.0	Alternative
1	drowning (feat. kodak black)	NaN	Hip-Hop/Rap
2	the real slim shady	20.0	Hip-Hop/Rap
3	worldwide steppers	2.0	Hip-Hop/Rap
4	2055	5.0	Hip-Hop/Rap
5	drivers license	9.0	Pop
6	cinema	2.0	Pop
7	dos mil 16	4.0	Pop
8	happier than ever	NaN	Pop
9	bam bam (feat. ed sheeran)	NaN	Pop

Problem 4.3

In which option was unconditional probabilistic imputation used?

[7, 9, 2]
[20, 5, 4]
unconditional

Option A

genre rank	genre
7.0	Alternative
5.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
2.0	Pop
2.0	Pop

Option B

genre rank	genre
7.0	Alternative
7.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
7.0	Pop
7.0	Pop

Option C

genre rank	genre
7.0	Alternative
2.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
2.0	Pop
7.0	Pop

Option D

genre rank	genre
7.0	Alternative
9.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
5.0	Pop
5.0	Pop

	track_name	genre rank	genre
0	good looking	7.0	Alternative
1	drowning (feat. kodak black)	NaN	Hip-Hop/Rap
2	the real slim shady	20.0	Hip-Hop/Rap
3	worldwide steppers	2.0	Hip-Hop/Rap
4	2055	5.0	Hip-Hop/Rap
5	drivers license	9.0	Pop
6	cinema	2.0	Pop
7	dos mil 16	4.0	Pop
8	happier than ever	NaN	Pop
9	bam bam (feat. ed sheeran)	NaN	Pop

Problem 4.4

In which option was probabilistic imputation conditional on `"genre"` used?

Option A

genre rank	genre
7.0	Alternative
5.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
2.0	Pop
2.0	Pop

Option B

genre rank	genre
7.0	Alternative
7.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
7.0	Pop
7.0	Pop

Option C

genre rank	genre
7.0	Alternative
2.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
2.0	Pop
7.0	Pop

Option D

genre rank	genre
7.0	Alternative
9.0	Hip-Hop/Rap
20.0	Hip-Hop/Rap
2.0	Hip-Hop/Rap
5.0	Hip-Hop/Rap
9.0	Pop
2.0	Pop
4.0	Pop
5.0	Pop
5.0	Pop

	track_name	genre rank	genre
0	good looking	7.0	Alternative
1	drowning (feat. kodak black)	NaN	Hip-Hop/Rap
2	the real slim shady	20.0	Hip-Hop/Rap
3	worldwide steppers	2.0	Hip-Hop/Rap
4	2055	5.0	Hip-Hop/Rap
5	drivers license	9.0	Pop
6	cinema	2.0	Pop
7	dos mil 16	4.0	Pop
8	happier than ever	NaN	Pop
9	bam bam (feat. ed sheeran)	NaN	Pop

Problem 6.2

Rahul wants to extract the `'instock availability'` status of the book titled 'A Light in the Attic'. Which of the following expressions will evaluate to `"In Stock"`? Assume that Rahul has already parsed the HTML into a BeautifulSoup object stored in the variable named `soup`.

Code Snippet A

```
soup.find('p',attrs = {'class': 'instock availability'})\
.get('icon-ok').strip()
```

Code Snippet B

```
soup.find('p',attrs = {'class': 'instock availability'}).text.strip()
```

Code Snippet C

```
soup.find('p',attrs = {'class': 'instock availability'}).find('i')\
.text.strip()
```

Code Snippet D

```
soup.find('div', attrs = {'class':'product_price'})\
.find('p',attrs = {'class': 'instock availability'})\
.find('i').text.strip()
```

```
<HTML>
<H1>The Book Club</H1>
<BODY BGCOLOR="FFFFFF">
Email us at <a href="mailto:support@thebookclub.com">
support@thebookclub.com</a>.

<div>
    <ol class="row">
    <li class="book_list">

        <article class="product_pod">
            <div class="image_container">
                    <img src="pic1.jpeg" alt="A Light in the Attic"
                    class="thumbnail">
            </div>

            <p class="star-rating Three"></p>

            <h3>
            <a href="cat/index.html" title="A Light in the Attic">
            A Light in the Attic
            </a>
            </h3>

            <div class="product_price">
                <p class="price_color">£51.77</p>

                <p class="instock availability">
                    <i class="icon-ok"></i>
                    In stock
                </p>

            </div>
        </article>
    </li>
    </ol>

</div>
</BODY>
</HTML>
```

Regular Expressions - Angela

Problem 5 regex101.com RexEgg.com → regex cheatsheet tables

You want to use regular expressions to extract out the number of ounces from the 5 product names below.

Index	Product Name	Expected Output
0	Adult Dog Food 18-Count, 3.5 oz Pouches	3.5
1	Gardetto's Snack Mix, 1.75 Ounce	1.75
2	Colgate Whitening Toothpaste, 3 oz Tube	3
3	Adult Dog Food, 13.2 oz. Cans 24 Pack	13.2
4	Keratin Hair Spray 2.6 oz	6

Regex finds a pattern, then captures it

not identified in regex

The names are stored in a pandas Series called `names`. For each snippet below, select the indexes for all the product names that **will not** be matched correctly.

pattern looks for: — . — — oz

For the snippet below, which indexes correspond to products that will **not** be matched correctly?

```
regex = r'(\d[\d.]+) oz'
names.str.findall(regex)
```

extract the number of oz from each product name

- [ ] 0
- [x] 1
- [ ] 2
- [ ] 3
- [ ] 4
- [ ] All names will be matched correctly.

( ) : groups sequence of digits and/or periods

capture group

\d : one digit from [0-9]

[ ] : character class, matches any one character inside [ ]

→ [abc] : matches either a, b, or c

→ "gr[ae]y" : matches "gray" or "grey"

→ NOT literal "[ ]"

. : inside [ ], so literal period "."

+ : one or more of the preceding element

You want to use regular expressions to extract out the number of ounces from the 5 product names below.

Index	Product Name	Expected Output
0	Adult Dog Food 18-Count, 3.5 oz Pouches ✓	3.5
1	Gardetto's Snack Mix, 1.75 Ounce	1.75
2	Colgate Whitening Toothpaste, 3 oz Tube	3
3	Adult Dog Food, 13.2 oz. Cans 24 Pack ✓	13.2
4	Keratin Hair Spray 2!6 oz	2!6

matches too little
matches without trying to capture any preceding number

no decimal

captures "2!6" instead of 6

The names are stored in a pandas Series called `names`. For each snippet below, select the indexes for all the product names that **will not** be matched correctly.

requires at least 1 digit, then some character, then one or more digits or "ounce"

For the snippet below, which indexes correspond to products that will **not** be matched correctly?

```
regex = r'(\d+?.\d+) oz|Ounce'
names.str.findall(regex)
```

- [ ] 0
- [x] 1
- [x] 2
- [ ] 3
- [x] 4
- [ ] All names will be matched correctly.

\d+ : matches one or more digits

? : after the + makes this quantifier lazy (non-greedy)
→ matches as few digits as possible
"if it exists or not" OK with no extra digits.

. : not escaped "\." or in [] ⇒ matches any single character except a newline → probably intended to match a decimal point

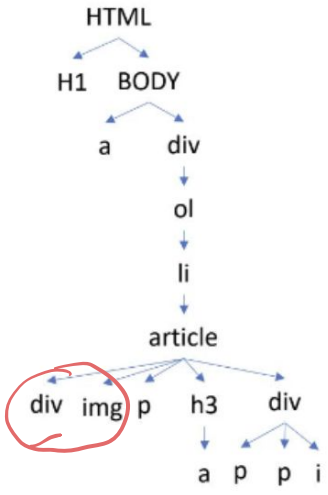
() : capture group extracts matched number separately

| : "or" operator splits regex into two alternatives
1.) "(\d+?.\d+) oz" 2.) Ounce

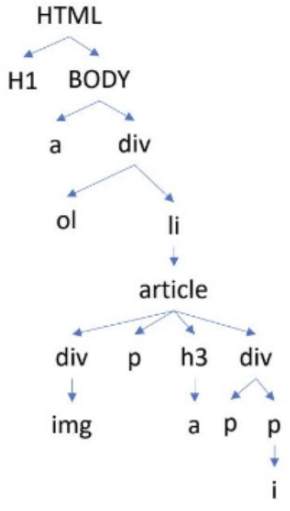
Web Scraping - Abhi

Which is the equivalent Document Object Model (DOM) tree of this HTML file?

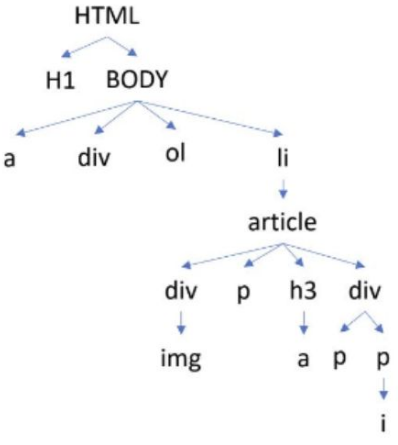
Tree A



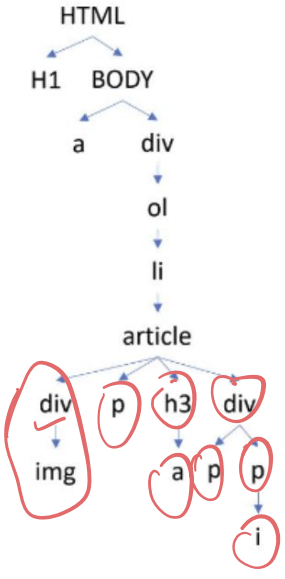
Tree B



Tree C



Tree D



div
ol
li

```
<HTML>
<H1>The Book Club</H1>
<BODY BGCOLOR="FFFFFF">
Email us at <a href="mailto:support@thebookclub.com">
support@thebookclub.com</a>.

<div>
    <ol class="row">
    <li class="book_list">

        <article class="product_pod">
            <div class="image_container">
                    <img src="pic1.jpeg" alt="A Light in the Attic"
                    class="thumbnail">
            </div>

            <p class="star-rating Three"></p>

            <h3>
            <a href="cat/index.html" title="A Light in the Attic">
            A Light in the Attic
            </a>
            </h3>

            <div class="product_price">
                <p class="price_color">£51.77</p>

                <p class="instock availability">
                    <i class="icon-ok"></i>
                    In stock
                </p>

            </div>
        </article>
    </li>
    </ol>

</div>
</BODY>
</HTML>
```

Problem 6.2

Rahul wants to extract the `'instock availability'` status of the book titled 'A Light in the Attic'. Which of the following expressions will evaluate to `"In Stock"`? Assume that Rahul has already parsed the HTML into a BeautifulSoup object stored in the variable named `soup`.

Code Snippet A

```
soup.find('p',attrs = {'class': 'instock availability'})\
.get('icon-ok').strip()
```

Code Snippet B

```
soup.find('p',attrs = {'class': 'instock availability'}).text.strip()
```

Code Snippet C

```
soup.find('p',attrs = {'class': 'instock availability'}).find('i')\
.text.strip()
```

Code Snippet D

```
soup.find('div', attrs = {'class':'product_price'})\
.find('p',attrs = {'class': 'instock availability'})\
.find('i').text.strip()
```

```
<HTML>
<H1>The Book Club</H1>
<BODY BGCOLOR="FFFFFF">
Email us at <a href="mailto:support@thebookclub.com">
support@thebookclub.com</a>.

<div>
    <ol class="row">
    <li class="book_list">

        <article class="product_pod">
            <div class="image_container">
                    <img src="pic1.jpeg" alt="A Light in the Attic"
                    class="thumbnail">
            </div>

            <p class="star-rating Three"></p>

            <h3>
            <a href="cat/index.html" title="A Light in the Attic">
            A Light in the Attic
            </a>
            </h3>

            <div class="product_price">
                <p class="price_color">£51.77</p>

                <p class="instock availability">
                    <i class="icon-ok"></i>
                    In stock
                </p>

            </div>
        </article>
    </li>
    </ol>

</div>
</BODY>
</HTML>
```

Problem 6.3

Rahul also wants to extract the number of stars that the book titled 'A Light in the Attic' received. If you look at the HTML file, you will notice that the book received a star rating of three. Which code snippet will evaluate to `"Three"`?

Code Snippet A

```
soup.find('article').get('class').strip()
```

Code Snippet B

```
soup.find('p').text.split(' ')
```

Code Snippet C

```
soup.find('p').get('class')[1]
```

None of the above

```
<HTML>
<H1>The Book Club</H1>
<BODY BGCOLOR="FFFFFF">
Email us at <a href="mailto:support@thebookclub.com">
support@thebookclub.com</a>.

<div>
    <ol class="row">
    <li class="book_list">

        <article class="product_pod">
            <div class="image_container">
                    <img src="pic1.jpeg" alt="A Light in the Attic"
                    class="thumbnail">
            </div>

            <p class="star-rating Three"></p>

            <h3>
            <a href="cat/index.html" title="A Light in the Attic">
            A Light in the Attic
            </a>
            </h3>

            <div class="product_price">
                <p class="price_color">£51.77</p>

                <p class="instock availability">
                    <i class="icon-ok"></i>
                    In stock
                </p>

            </div>
        </article>
    </li>
    </ol>

</div>
</BODY>
</HTML>
```

Text as Data - Abhi

$$\cos\theta = \frac{\vec{u} \cdot \vec{v}}{\|u\| \|v\|}$$

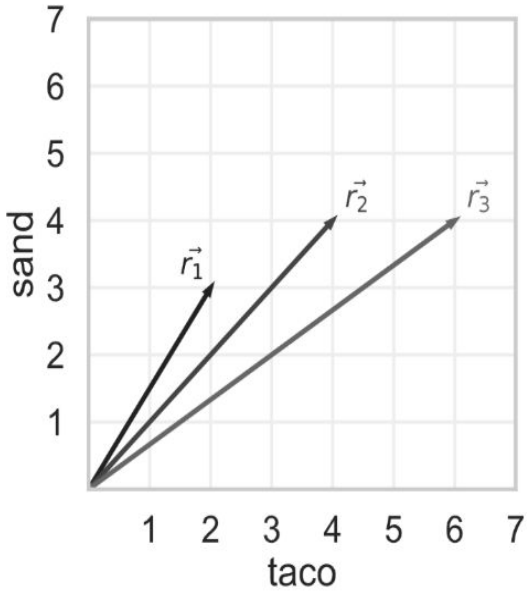
$\vec{u}$ $\vec{v}$

Problem 7

Tahseen decides to look at reviews for the same hotel, but he modifies them so that the only terms they contain are `"taco"` and `"sand"`. The bag-of-words representations of three reviews are shown as vectors below.

$r_1 = \begin{bmatrix} 2 \\ 3 \end{bmatrix}$ $r_2 = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$

$r_3 = \begin{bmatrix} 6 \\ 4 \end{bmatrix}$



- [x] $\vec{r}_1$ and $\vec{r}_2$
- [ ] ~~$\vec{r}_1$ and $\vec{r}_3$~~
- [x] $\vec{r}_2$ and $\vec{r}_3$

$\frac{2 \cdot 4 + 3 \cdot 4}{\sqrt{13} \cdot \sqrt{32}} = \frac{20}{4\sqrt{26}}$

$\sqrt{13} \cdot 4\sqrt{2}$

$= \frac{5}{\sqrt{26}}$

Using cosine similarity to measure similarity, which pair of reviews are the most similar? If there are multiple pairs of reviews that are most similar, select them all.

$\frac{4 \cdot 6 + 4 \cdot 4}{\sqrt{32} \cdot \sqrt{52}} = \frac{40}{8\sqrt{26}} = \frac{5}{\sqrt{26}}$

You create a table called `gums` that only contains the chewing gum purchases of `df`, then you create a bag-of-words matrix called `bow` from the `name` column of `gums`. The `bow` matrix is stored as a DataFrame shown below:

	pur	gum	...	paperboard	80
0	0	1	...	0	1
1	0	1	...	1	1
...	...	...	...	...	...
38	0	0	...	0	0
39	0	0	...	0	1

You also have the following outputs:

```
>>> bow_df.sum(axis=0)
pur             5
gum            41
sugar           2
               ..
90              4
paperboard     22
80             20
Length: 139
```

```
>>> bow_df.sum(axis=1)
0     21
1     22
2     22
      ..
37    22
38    10
39    17
Length: 40
```

```
>>> bow_df.loc[0, 'pur']
0

>>> (bow_df['paperboard'] > 0).sum()
20

>>> bow_df['gum'].sum()
41
```

For each question below, write your answer as an unsimplified math expression (no need to simplify fractions or logarithms) in the space provided, or write "Need more information" if there is not enough information provided to answer the question.

Problem 8.1

What is the TF-IDF for the word "pur" in document 0?

TF-IDF = TF · IDF

TF = 0

TF-IDF = 0

	pur	gum	...	paperboard	80
0	0	1	...	0	1
1	0	1	...	1	1
...	...	...	...	...	...
38	0	0	...	0	0
39	0	0	...	0	1

```
>>> bow_df.sum(axis=0)
pur            5
gum           41
sugar          2
              ..
90             4
paperboard    22
80            20
Length: 139
```

```
>>> bow_df.sum(axis=1)
0     21
1     22
2     22
      ..
37    22
38    10
39    17
Length: 40
```

```
>>> bow_df.loc[0, 'pur']
0

>>> (bow_df['paperboard'] > 0).sum()
20

>>> bow_df['gum'].sum()
41
```

Problem 8.2

What is the TF-IDF for the word "gum" in document 0?

IDF = log (# of doc. / # of docs w/ term)

TF = 1/21

IDF = log (40/??)

Need more information

	pur	gum	...	paperboard	80
0	0	1	...	0	1
1	0	1	...	1	1
...	...	...	...	...	...
38	0	0	...	0	0
39	0	0	...	0	1

```
>>> bow_df.sum(axis=0)
pur           5
gum          41
sugar         2
             ..
90            4
paperboard   22
80           20
Length: 139
```

```
>>> bow_df.sum(axis=1)
0     21
1     22
2     22
      ..
37    22
38    10
39    17
Length: 40
```

```
>>> bow_df.loc[0, 'pur']
0

>>> (bow_df['paperboard'] > 0).sum()
20

>>> bow_df['gum'].sum()
41
```

Problem 8.3

What is the TF-IDF for the word "paperboard" in document 1?

$\log_2$

$TF = \frac{1}{22}$

$IDF = \log_2\left(\frac{40}{20}\right)$

$TF\text{-}IDF = \frac{1}{22} \cdot 1 = \boxed{\frac{1}{22}}$ ← final answer

	pur	gum	...	paperboard	80
0	0	1	...	0	1
1	0	1	...	1	1
...	...	...	...	...	...
38	0	0	...	0	0
39	0	0	...	0	1

```
>>> bow_df.sum(axis=0)
pur           5
gum          41
sugar         2
             ..
90            4
paperboard   22
80           20
Length: 139
```

```
>>> bow_df.sum(axis=1)
0     21
1     22
2     22
      ..
37    22
38    10
39    17
Length: 40
```

```
>>> bow_df.loc[0, 'pur']
0

>>> (bow_df['paperboard'] > 0).sum()
20

>>> bow_df['gum'].sum()
41
```

>0, → 1

<=0

Constant Model - Angela

Problem 9.1

Which of the following is closest to the constant prediction h^* that minimizes:

30

from histogram, is the mode of this distribution

choose a constant prediction h

same number h for all data points

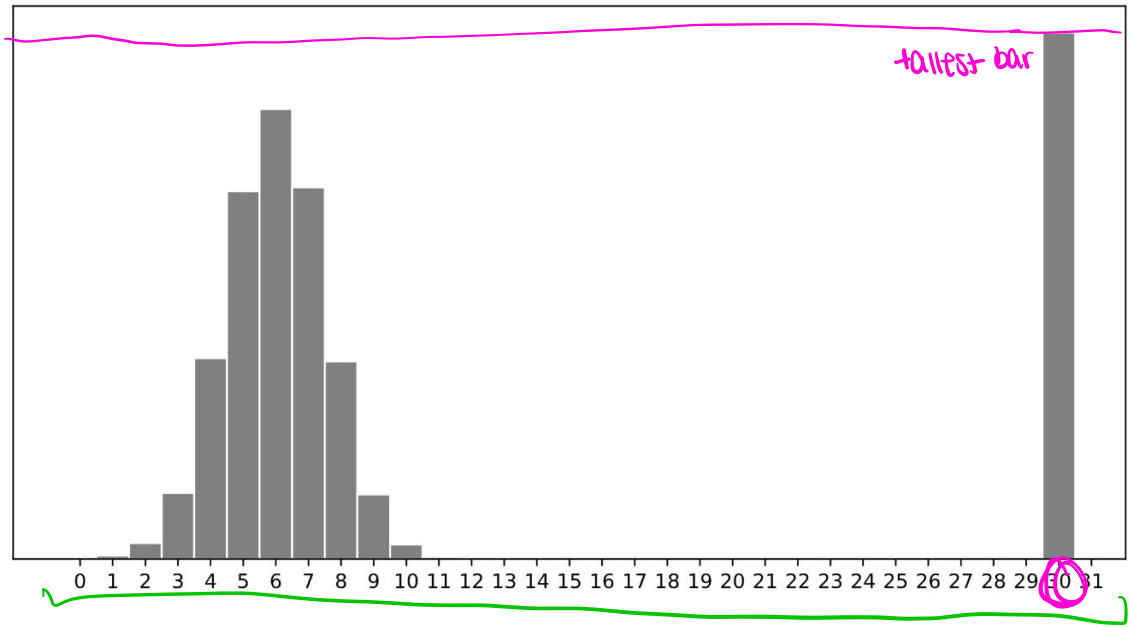
$$L_{0/1}(h, y) = \frac{1}{n} \sum_{i=1}^{n} \begin{cases} 0 & y_i = h \\ 1 & y_i \neq h \end{cases}$$

minimize zero-one loss ↑ single prediction

- 1
- 5
- 6
- 7
- 11
- 15
- **30**

find the integer h that makes the fewest mistakes compared to the observed values of y_i

Consider a dataset of n **integers**, $y_1, y_2, \ldots, y_n$, whose histogram is given below:



multiple data points $y_1, y_2 \ldots y_n$

→ the empirical risk (average loss)

$$R(h) = \frac{1}{n} \sum_{i=1}^{n} L_{0/1}(h, y_i)$$

This counts the fraction of points for which h is not equal to y_i

↑

To minimize this count, choose h that appears most frequently

i.e. the mode

intuition: if h is the most common value (mode), you will have the fewest points that differ from h, hence the fewest mistakes.

Problem 9.2

Which of the following is closest to the constant prediction h^* that minimizes:

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

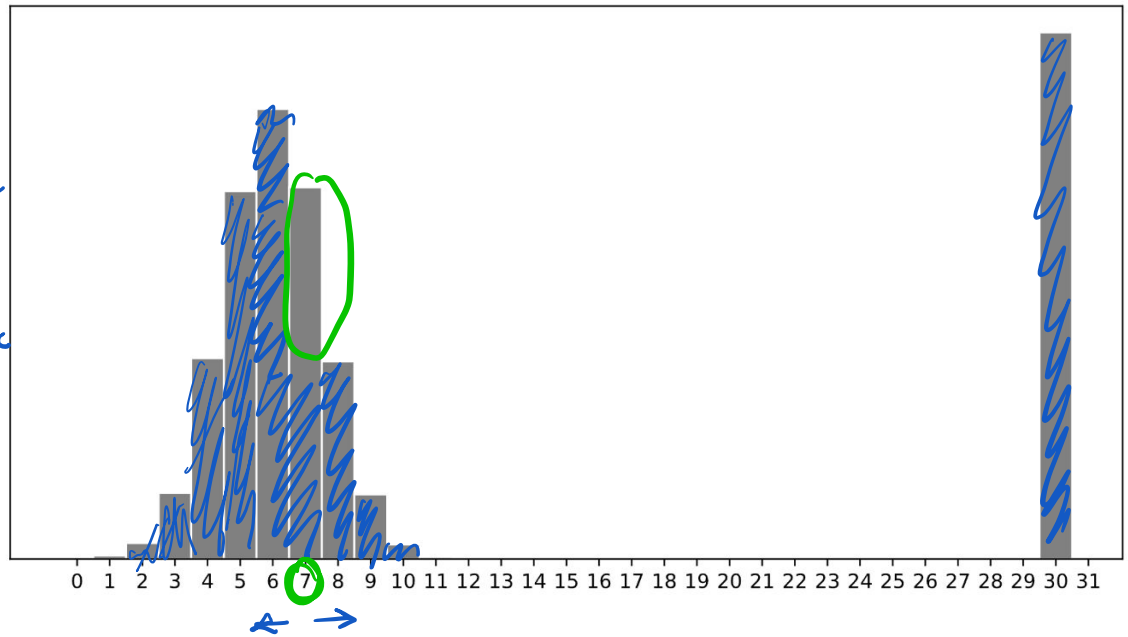
minimize absolute loss $|y-h|$

- 1
- 5
- 6
- 7
- 11
- 15
- 30

minimizer for empirical risk for the constant model when using absolute loss is the median.

7

Consider a dataset of n **integers**, $y_1, y_2, \ldots, y_n$, whose histogram is given below:



intuition: if you move h left of the median, you increase the absolute distance to all the data points on the right

→ if you move h right of the median, you decrease the absolute distance to all the data on the left.

"best balance" point is where half the data is on each side

→ median

Problem 9.3

Which of the following is closest to the constant prediction h^* that minimizes:

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

- 1
- 5
- 6
- 7
- 11
- 15
- 30

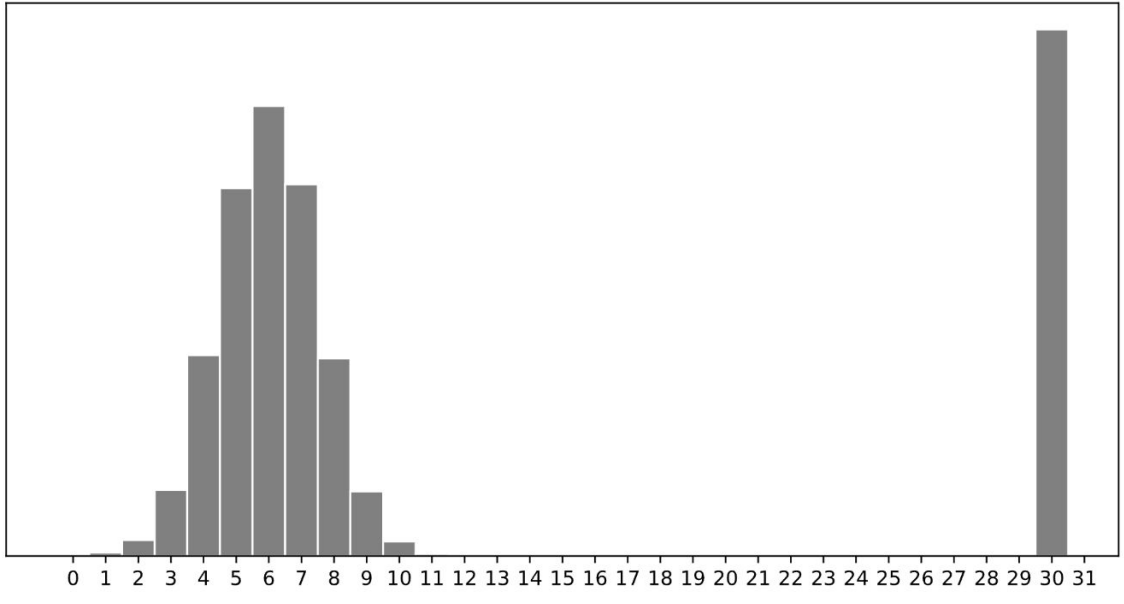
minimizer for empirical risk for the constant model when using squared loss is the mean

mean > 7

11 next largest option

minimizes squared loss

Consider a dataset of n **integers**, $y_1, y_2, \ldots, y_n$, whose histogram is given below:



intuition: this function is minimized when

$$h = \frac{1}{n}\sum_{i=1}^{n} y_i \quad \text{(mean)}$$

if you take the derivative of the sum of squared errors and set it to zero, you get this result.

median < mean since there is a large outlier skewing the distribution to the right.

Problem 9.4

Which of the following is closest to the constant prediction h^* that minimizes:

$$\lim_{p \to \infty} \frac{1}{n} \sum_{i=1}^{n} |y_i - h|^p$$

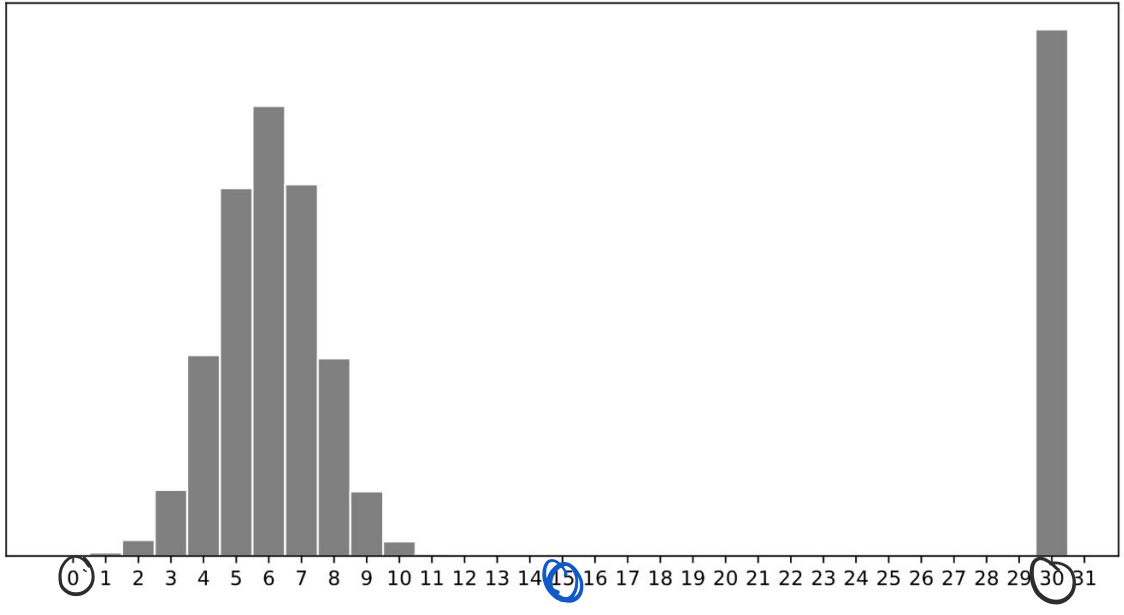
minimize infinity loss

- 1
- 5
- 6
- 7
- 11
- 15
- 30

minimizer of empirical risk for the constant model when using infinity loss is the midrange, i.e. halfway between the min and max

$\frac{30-0}{2} = 15$

Consider a dataset of n **integers**, $y_1, y_2, \ldots, y_n$, whose histogram is given below:



intuition: $\max_{i \in \text{data}} |y_i - h|$ essentially means max absolute error

→ choose constant h that minimizes this worst-case (maximum) error.

→ center our prediction so that the worst error on the low side is exactly balanced by the worst error on the right side.

→ choose h as the midrange

left error: $\left| \frac{y_{min} + y_{max}}{2} - y_{min} \right| = y_{max} - y_{min}$

right error: $\left| y_{max} - \frac{y_{min} + y_{max}}{2} \right| = y_{max} - y_{min}$

both errors are equal, max error is as small as it can be given the spread of the data.

Regression - Angela

Problem 10

derivative of risk function
set to zero, then solve for optimal h^*

Consider a dataset that consists of $y_1, \cdots, y_n$. In class, we used calculus to minimize mean squared error, $R_{sq}(h) = \frac{1}{n}\sum_{i=1}^{n}(h - y_i)^2$. In this problem, we want you to apply the same approach to a slightly different loss function defined below:

Loss: $$L_{\text{midterm}}(y, h) = (\alpha y - h)^2 + \lambda h$$

for a single data point y_i

Problem 10.1

Write down the empiricial risk $R_{\text{midterm}}(h)$ by using the above loss function.

average loss across all datapoints $y_1 \cdots y_n$

$$R_{\text{midterm}}(h) = \frac{1}{n}\sum_{i=1}^{n} L_{\text{midterm}}(y_i, h)$$

$$\Rightarrow R_{\text{midterm}}(h) = \frac{1}{n}\sum_{i=1}^{n}\left[(\alpha y_i - h)^2 + \lambda h\right]$$

depends on i

$$\Rightarrow \boxed{R_{\text{midterm}}(h) = \left[\frac{1}{n}\sum_{i=1}^{n}(\alpha y_i - h)^2\right] + \lambda h}$$

does not depend on i

Problem 10.2

The mean of dataset is $\bar{y}$, i.e. $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$. Find h^* that minimizes $R_{\text{midterm}}(h)$ using calculus. Your result should be in terms of $\bar{y}$, α and λ.

$$L_{\text{midterm}}(y, h) = (\alpha y - h)^2 + \lambda h$$

1.) take derivative of risk function in respect to h

chain rule: $\frac{d}{dh}[f(g(h))] = f'(g(h)) \cdot g'(h)$

$$\frac{d}{dh} R_{\text{midterm}}(h) = \frac{d}{dh}\left[\left[\frac{1}{n}\sum_{i=1}^{n}(\alpha y_i - h)^2\right] + \lambda h\right]$$

$$\frac{d}{dh} R_{\text{midterm}}(h) = \frac{1}{n}\sum_{i=1}^{n} 2[(\alpha y_i - h)(-1)] + \lambda$$

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

$$= -\frac{2}{n}\cdot\left[\sum_{i=1}^{n}\alpha y_i - \sum_{i=1}^{n} h\right] + \lambda$$

(h is a constant)

$$= -\frac{2}{n}\cdot\left[\alpha\sum_{i=1}^{n} y_i - nh\right] + \lambda$$

$$= -\frac{2}{n}\cdot[\alpha n\bar{y} - nh] + \lambda$$

$$= -2\alpha\bar{y} + 2h + \lambda$$

$$= 2h - 2\alpha\bar{y} + \lambda = 0 \quad \text{set to 0}$$

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

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

$$2h^* - 2\alpha\bar{y} + \lambda = 0$$

$$2h^* = 2\alpha\bar{y} - \lambda$$

$$\boxed{h^* = \alpha\bar{y} - \frac{\lambda}{2}}$$

intuition: term $\alpha\bar{y}$ would be the best constant if the loss were purely $\Sigma(\alpha y_i - h)^2$
→ extra $-\frac{\lambda}{2}$ shift comes from the linear penalty λh.