Data C100/C200, Midterm 1

Spring 2022

Name: _____

Email: _____@berkeley.edu

Student ID: _____

Exam Room: _____

Name and SID of left neighbor: _____

Name and SID of right neighbor: _____

Instructions:

This midterm exam consists of **70 points** spread out over **9 questions** and the Honor Code and must be completed in the **110 minute** time period ending at **10:00**, unless you have accommodations supported by a DSP letter.

Note that some questions have circular bubbles to select a choice. This means that you should only **select one choice**. Other questions have boxes. This means you should **select all that apply**. Please shade in the box/circle to mark your answer.

Honor Code [1 Pt]:

As a member of the UC Berkeley community, I act with honesty, integrity, and respect for others. I am the person whose name is on the exam and I completed this exam in accordance with the Honor Code.

Signature: _____

This page has been intentionally left blank.

1 Squid Game [11 Pts]

As a diehard fan of BTS (a music group), you want the band to remain successful in the United States among *American youth*. However, knowing the stiff competition from Blackpink (another music group), you're afraid that Blackpink will overtake BTS in popularity. As a result, you and your Data 100 classmates decide to create a survey to appropriately estimate BTS's popularity among American youth.

For this question, assume an *American youth* is an American citizen younger than 25 years old.

(a) [2 Pts] You come up with a low cost way to collect data: You and your friends will set up tables at the UC Berkeley and Berkeley High School campuses and hand out paper surveys to passersby, who can then drop their response into a dropbox next to these same tables. The survey is a simple form that is the same as question 1.f from this exam, i.e. participants fill only a single bubble and write nothing else on the form. Select all of the following which are true.

- ☐ **Left as is, your sample may suffer from significant non-response bias.**
- ☐ Assuming every individual who walks by responds to the survey, you can safely generalize the results of your sample to the population of interest.
- ☐ This survey is a form of quota sampling for high-school and college students.
- ☐ For this sample, there's no need to worry about response bias.

(b) [1 Pt] For the sampling procedure in part (a), what is the sampling frame?

- ◯ Everyone who lives in the city of Berkeley.
- ◯ Everyone who attends UC Berkeley or Berkeley High School.
- ◯ Students under the age of 25 who attend either UC Berkeley or Berkeley High School.
- ◯ **Everyone who passes by your table at UC Berkeley or Berkeley High School.**

(c) [0 Pts] Setting up a table on campus is a lot of work! As an alternative sampling strategy for learning about the popularity of BTS vs. Blackpink among American youth, you decide to include the question on the DS100 midterm instead. Which of the following pieces of terminology apply to this sampling strategy? Select all that apply. Note: This question is not for a grade, so don't spend too much time on it.

- ☐ **Simple random sampling**
- ☐ **Probability sampling**
- ☐ Quota sampling
- ☐ **Convenience sampling**

Solution: At least one of Simple Random Sampling, Probability Sampling, and Convenience Sampling gets full credit.

Convenience Sampling: This is a form of convenience sampling - asking our question on the Weekly Survey allows for the ease of surveying whoever we can get ahold of.

Simple Random Sampling: This is a SRS because our sample is being uniformly at random, where every individual, and group of individuals has the same probability of being chosen. This is because we're assuming that everyone enrolled in the course answers every question on the Weekly Check, so this probability, by default is 100%

Probability Sampling: This is a probability sample because we are able to specify the probability that someone/some group is chosen in our sample. In our case, that probability is 100%

(d) [2 Pts] For the sampling procedure in part (c), name a group of individuals in the sampling frame, but not in the population of interest. For example, "Students majoring in Data Science at UC Berkeley" is an example of a group of individuals.

Solution: International students taking the midterm.
Students older than 25 taking the midterm

(e) [1 Pt] What forms of error/bias are present in the sampling technique presented in part (c)?

- ☐ **Selection bias**
- ☐ **Response bias**
- ☐ **Non-response bias**
- ☐ **Chance error**
- ☐ None of the above

Solution: Note: An older version of this question included an assumption that everyone completed the question. Without this assumption, chance error and non-response bias are introduced into the survey. Chance error and non-response bias will be marked as correct.

(f) [0 Pts] Which of the following two groups are you a fan of? If both, select your favorite between the two.

- ◯ **BTS.**
- ◯ Blackpink.
- ◯ Neither.

(g) [3 Pts] Note for the next few subparts:

- We define *a majority* of a group as more than half of that group.
- Recall the definition of a *binomial probability*: If we draw at random with replacement n times, from a population in which a proportion p of the individuals are called "successes" (and the remaining $1 - p$ are "failures"), then the probability of k successes (and hence, $n - k$ failures) is

$$\binom{n}{k} p^k (1-p)^{n-k}$$

After all enrolled students take your survey, you learn that 60% of Data 100 students are BTS fans and 40% of Data 100 students are either Blackpink fans or are fans of neither. Suppose you randomly sample **with replacement** 100 students from the class. What is the probability

that a *majority* of the sample is comprised of BTS fans? Please leave your answer as an expression; there is no need to fully calculate it out.

Solution: $\sum_{k=51}^{100} \binom{100}{k} (.6)^k (.4)^{100-k}$

(h) [2 Pts] Suppose we also take a sample with replacement of size 150, and another sample with replacement of size 50. Assume that all three samples are drawn from the same Data 100 class (i.e., with the proportions of BTS and Blackpink/neither fans from part (g)). What is the probability that **at least one** of the three different samples contains a majority of BTS fans? You must use at least one of the following variables in your answer:

- The probability that the size 100 sample contains a majority of BTS fans: $p_{m,100}$. Note that this is also the correct answer to part g.
- The probability that the size 150 sample contains a majority of BTS fans: $p_{m,150}$
- The probability that the size 50 contains a majority of BTS fans: $p_{m,50}$

Solution:

$$1-[(\sum_{k=51}^{100}\binom{100}{k}(.6)^k(.4)^{100-k})*(\sum_{k=76}^{150}\binom{150}{k}(.6)^k(.4)^{150-k})*(\sum_{k=26}^{50}\binom{50}{k}(.6)^k(.4)^{50-k})]$$

which can also be written as

$$1-((1-p_{m,100})\cdot(1-p_{m,150})\cdot(1-p_{m,50}))$$

2 Pandas Cinematic Universe [8 Pts]

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 following DataFrame `netflix` contains records of all *Netflix releases*. For this question, define a Netflix release as a Movie or TV Show from 1925 to 2021 released on any Netflix platform worldwide. Five lines of the table are shown below. You may assume that the `show_id` column is the primary key of the table.

	show_id	type	title	director	cast	country	date_added	release_year	rating	duration	listed_in	description	IMDb rating
0	s1	Movie	Dick Johnson Is Dead	Kirsten Johnson	NaN	United States	September 25, 2021	2020	PG-13	90 min	Documentaries	As her father nears the end of his life, filmm...	7.4
4	s5	TV Show	Kota Factory	NaN	Mayur More, Jitendra Kumar, Ranjan Raj, Alam K...	India	September 24, 2021	2021	TV-MA	2 Seasons	International TV Shows, Romantic TV Shows, TV ...	In a city of coaching centers known to train I...	9.2
9	s10	Movie	The Starling	Theodore Melfi	Melissa McCarthy, Chris O'Dowd, Kevin Kline, T...	United States	September 24, 2021	2021	PG-13	104 min	Comedies, Dramas	A woman adjusting to life after a loss contend...	6.3
15	s16	TV Show	Dear White People	NaN	Logan Browning, Brandon P. Bell, DeRon Horton,...	United States	September 22, 2021	2021	TV-MA	4 Seasons	TV Comedies, TV Dramas	Students of color navigate the daily slights a...	6.1
24	s25	Movie	Jeans	S. Shankar	Prashanth, Aishwarya Rai Bachchan, Sri Lakshmi...	India	September 21, 2021	1998	TV-14	166 min	Comedies, International Movies, Romantic Movies	When the father of the man she loves insists t...	6.5

(a) [3 Pts] Identify the feature type that best describes each of the following variables:

	Quantitative Continuous	Quantitative Discrete	Qualitative Ordinal	Qualitative Nominal
(i) type	○	○	○	●
(ii) IMDb rating	●	○	○	○
(iii) release year	○	○	○	○

(b) [2 Pts] Choose the lines of code that correctly display the top 5 countries that have the greatest total number of TV Show releases, e.g. if a country had 5 TV Show releases in 2020, 6 TV Show releases in 2021, and no releases in any other year, this country has a total of 11 TV Show releases. The result should show a Series with the `country` name as the index and the number of releases as the value. Select all that apply.

- ☐ **`netflix[netflix['type']=='TV Show']['country'].value_counts().head(5)`**
- ☐ `netflix['country'].value_counts().sort_values(ascending=False).head(5)`
- ☐ `netflix.iloc[netflix['type']=='TV Show','country'].value_counts().head()`
- ☐ **`netflix.loc[netflix['type']=='TV Show','country'].value_counts().head(5)`**

(c) [3 Pts] Fill in the blanks below in order to answer the question: What is the average `IMDb rating` of TV shows per each release year? The result should show a series with the `release_year` as the index and the average `IMDb rating` as the value.

```
netflix[_____].groupby(_____)[_____]._____
```

We will not be giving *any* points to solutions that don't follow the above skeleton code. Write your answer in the blanks in the skeleton code below.

```
netflix[_____________________________].groupby(_____________) \


[_____________________________]._____________________________
```

Solution:

```
netflix[netflix['type'] == 'TV Show'] \
.groupby('release_year')['IMDb rating'].mean()
```

3 Painting Pandas [6 Pts]

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`.

You are doing some research into the iconic American painter Bob Ross (1942-1995) for an Art History class. Looking online, you find the `bob_ross` dataset, which contains all of the episodes of Bob Ross's television show and all the elements painted in these episodes. The first few lines of this DataFrame are displayed below. Each entry from `APPLE_FRAME` to `WOOD_FRAMED` is either 0 or 1, indicating the absence or presence of a particular element, respectively.

	EPISODE	TITLE	APPLE_FRAME	...	WINDOW_FRAME	WINTER	WOOD_FRAMED
0	S01E01	"A WALK IN THE WOODS"	0	...	0	0	0
1	S01E02	"MT. MCKINLEY"	0	...	0	1	0
2	S01E03	"EBONY SUNSET"	0	...	0	1	0
3	S01E04	"WINTER MIST"	0	...	0	0	0
4	S01E05	"QUIET STREAM"	0	...	0	0	0

5 rows × 69 columns

(a) [2 Pts] For your research, you want to find the episodes where Bob Ross painted a lake, river, or tree. Write a line of code using the `.loc` operator that will return a new DataFrame object with the columns titled "EPISODE", "LAKE", "RIVER", "TREE". Just like for "EPISODE", you can assume the other three column names are columns in the `bob_ross` dataset. **You must use the `.loc` method to receive credit for this question.**

```
bob_ross.loc_____
```

Solution: One possible solution:

```
bob_ross.loc[:, ["EPISODE", "LAKE", "RIVER", "TREE"]]
```

Another possible solution (taking only rows with at least one lake, river, or tree):

```
bob_ross.loc[(bob_ross['LAKE']>=1)|(bob_ross['RIVER']>=1)| \
                (bob_ross['TREE']>=1),
             ['EPISODE', 'LAKE', 'RIVER', 'TREE']]
```

(b) [2 Pts] Now, you do the same thing, but this time using the `.iloc` method. Write a line of code using the `.iloc` method that will return a new DataFrame object with the columns titled "EPISODE", "LAKE", "RIVER", "TREE". **You must use the `.iloc` method to receive credit for this question.**
Here are some of the column labels and the index of these labels in the list of columns. Note that the table below is not a DataFrame, it's just a list of which column numbers correspond to the given column names:

Column Label	EPISODE	LAKE	RIVER	TREE
Column Index	0	34	50	60

```
bob_ross.iloc_____________________________________________________________
```

Solution: `bob_ross.iloc[:, [0, 34, 50, 60]]`

(c) [2 Pts] Now, you want to plot a histogram of the number of elements that occur in each epsiode. For example, the paintings in episode S01E09 include "BEACH", "CLOUDS", "FENCE", and "OCEAN", and thus this episode has 4 elements.

This question may require using named arguments in the Pandas `sum` method that you haven't used before. Similar to homeworks, we have provided you with the `pandas.DataFrame.sum` documentation to assist you with this question below:

We've provided some skeleton code – please note that points will only be given to solutions that fit the skeleton.

```
import matplotlib.pyplot as plt
import seaborn as sns
br = bob_ross.copy()
___ = ___.sum(___)     # (i)
_____________________  # (ii)
```

(i) Copy line `(i)` and fill in the blanks to add a column named `"SUM"` to the DataFrame `br` that contains the sum of the types of Bob Ross elements (the column labels). You may include 0 or more parameters for the call to sum.

_____

Solution: `br["SUM"] = br.sum(axis=1, numeric_only=True)`

Note: At the time of this writing, the usage of `numeric_only` is optional, but in a future version of pandas it will be required. If you exclude `numeric_only` you will receive: `FutureWarning: Dropping of nuisance columns is deprecated`.

Note: `axis = "columns"` also works.

(ii) Copy line `(ii)` and fill in the blanks to create the desired plot. You may use either matplotlib or seaborn, as imported for you.

Solution: `plt.hist(br["SUM"])`

An example result is given below. Note that your number of bins and style may vary depending on which of the two libraries you use.

4 Go Regex Go! [7 Pts]

For this question, you're given the following code:

```
re.findall(pattern, "godoggogo100")
```

For each possible pattern, list the number of times that the string `"go"` appears as an item in the list returned by the above code. The first two have been completed for you: Pattern 1 returns `["go", "go", "go"]`, so we wrote 3; pattern 2 returns `["godo"]` and does not contain the string `"go"` as an item, so we wrote 0.

Each response is worth 1 point.

1. `pattern = r'go'` ___3___
2. `pattern = r'godo'` ___0___
3. `pattern = r'go.*'` _____
4. `pattern = r'.*go.*'` _____
5. `pattern = r'go{2}'` _____
6. `pattern = r'(go){1}'` _____
7. `pattern = r'(go)[dg1]'` _____
8. `pattern = r'[go](go)'` _____
9. `pattern = r'[go]*(go)'` _____

Solution:

1. 3 (given)
2. 0 (given)
3. 0
4. 0
5. 0

6. 3

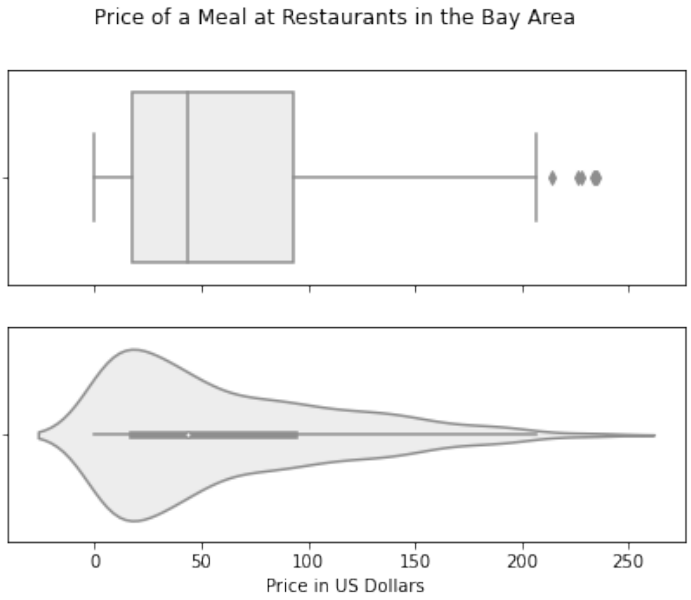
7. 2

8. 1

9. 2

5 Boxed Meals, or Boxed Violins? [7 Pts]

For all parts of this question, refer to the plot below, which graphs the price of a meal at restaurants in the Bay Area.



(a) [2 Pts] Which of the following are true statements about the distribution of meal prices? Select all that apply. **Note: We gave points if unimodal nor bimodal were not selected (whereas the solutions require unimodal to be selected)**

- ☐ **The distribution of meal prices is unimodal.**
- ☐ The distribution of meal prices is bimodal.
- ☐ The distribution of meal prices is symmetric.
- ☐ The distribution of meal prices has a long left tail.
- ☐ **The distribution of meal prices has a long right tail.**

(b) [1 Pt] What is the approximate median meal price?

- ◯ 0
- ◯ 15
- ◯ **50**
- ◯ 80
- ◯ 95
- ◯ 215
- ◯ Cannot tell from the plot

(c) [2 Pts] What is the approximate mean meal price?

- ◯ 0
- ◯ 15
- ◯ 50
- ◯ 80
- ◯ 95
- ◯ 215
- ◯ **Cannot tell from the plot**

(d) [2 Pts] What are the approximate mode(s) of the distribution of meal prices?

- ☐ 0
- ☐ **15**
- ☐ 50
- ☐ 80
- ☐ 95
- ☐ 215
- ☐ Cannot tell from the plot

6 Kernel Density Estimation [4 Pts]

For all parts of this question, refer to the four Kernel Density Estimator plots below:
(A) (B)
(C) (D)

(a) [2 Pts] Which of the above KDE plots is most likely to have the highest bandwidth parameter value of the four plots?

- ◯ A
- ◯ **B**
- ◯ C
- ◯ D

(b) [2 Pts] Which of the above KDE plots is most likely to have the lowest bandwidth parameter value of the four plots?

- ◯ A
- ◯ B
- ◯ **C**
- ◯ D

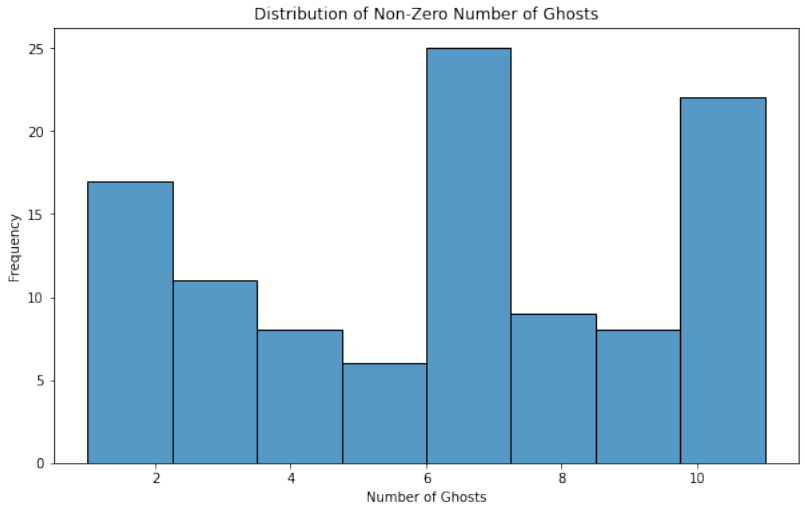
7 No Ghosting [12 Pts]

You are working for Ghostbusters Inc, a ghost removal service based on data science! You are tasked with analyzing and developing a model to predict ghost sightings using a dataset `ghosts` containing information about the number of ghost sightings in a location based on factors such as air viscosity. The first 5 rows (i.e., locations) of the dataset are shown below.

	latitude	longitude	num_ghosts	air_visc
0	34.0	17.0	0.0	2.388096
1	86.0	49.0	0.0	2.828219
2	66.0	79.0	1.0	0.889161
3	54.0	66.0	0.0	2.218303
4	14.0	73.0	0.0	2.029698

Since the Ghostbusters have been doing a great job, **the majority of locations have no ghost sightings**! **Importantly, 90% of all data points in `ghosts[num_ghosts]` is 0!** Further, you can assume there are no null values in the dataset, though not every location (given as a latitude, longitude pair) is covered in the dataset. You will be working through some of the remaining challenges for the Ghostbusters.

(a) [3 Pts] You wish to plot a histogram of all the non-zero `num_ghosts` values. Which of the following can be set to the variable `to_plot` (i.e. to fill in the blank on the next page) to generate the plot shown? Select all that apply.




```
to_plot = _____

plt.figure(figsize = (10, 6))
sns.histplot(to_plot, kde = False)
plt.title('Distribution of Non-Zero Number of Ghosts')
plt.xlabel('Number of Ghosts'); plt.ylabel('Frequency')
```

☐ `ghosts.loc[ghosts['num_ghosts'] != 0, 'num_ghosts']`

☐
```
ghosts.groupby('num_ghosts').filter(
    lambda sdf: sdf['num_ghosts'].sum() > 0
)['num_ghosts']
```

☐ `ghosts[ghosts['num_ghosts'] != 0, 'num_ghosts']`

☐ `ghosts.iloc[ghosts['num_ghosts'] != 0, 0]`

Solution: The first option is correct since it is a correct application of `loc`.

The second option is correct since it is a correct application of `groupby` and `filter`. Even though the groups are insignificant since we filter on the same variable that we grouped by (which is equivalent to indexing), we can still do it this way.

The third option is incorrect because this syntax is not supported.

The fourth option is incorrect since `num_ghosts` is not the 1st column.

(b) [4 Pts] Fill in the blanks on the next page to calculate a Pandas "map" of the number of total ghost sightings. In other words, for each combination of longitude and latitude, calculate the total number of ghost sightings to output the DataFrame below. Assume that longitude and latitude are discretized as shown in the question header. Note: The minimum and maximum longitudes in the Ghostbusters Inc. dataset are 0.0 and 90.0, respectively. The same is true of the latitudes.

longitude	0.0	1.0	2.0	3.0	4.0	5.0	6.0	7.0	8.0	9.0	...	81.0	82.0	83.0	84.0	85.0	86.0	87.0	88.0	89.0	90.0
latitude																					
0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	3.0	0.0	...	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3.0	0.0	0.0	0.0	0.0	0.0	2.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
86.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
87.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
88.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
89.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
90.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

91 rows × 91 columns

```
ghost_map = ghosts._____(
```

_____

_____

_____

```
).fillna(0)
```

Solution:

```
ghosts.pivot_table(columns = 'longitude',
                   index = 'latitude',
                   values = 'num_ghosts',
                   aggfunc = 'sum').fillna(0)
```

(c) [3 Pts] Note: Tricky problem! Suppose that you try to train a simple linear regression model to predict the number of ghost sightings y using the air viscosity v. You take a random sample of 8 datapoints in order to check the performance of your model on a small dataset.

Here's the tricky twist: Your model will try to predict y from the viscosity after v is converted into standard units. Recall from data 8 that data in standard units has a mean of zero and a standard deviation of 1.

More precisely, let $x = \frac{v-\bar{v}}{\sigma_v}$ be the *standard units* of the feature v, where $\bar{v}$ and σ_v are the mean and standard deviation of v, respectively. For example, if $v = 1$, then $x = \frac{1-\bar{v}}{\sigma_v} = -8/9$. After computing the x values, You train a simple linear regression using the standard units x:

$$\hat{y} = a + bx$$

What are the optimal least squares $\hat{a}$ (intercept) and $\hat{b}$ (slope) values based on the data sample shown below? Here are some potentially useful statistics: $\bar{v} = 3$, $\bar{y} = 1$, $\sigma_v = 2.25$, $\sigma_y = \sqrt{3}$. Hint: Using tons of arithmetic is the wrong approach.

v	x	y
$7/2$	$2/9$	4
$3/2$	$-2/3$	0
1	$-8/9$	0
2	$-4/9$	0
$7/2$	$2/9$	0
$3/2$	$-2/3$	0
$5/2$	$-2/9$	4
$17/2$	$22/9$	0

◯ $\hat{a} = 0, \hat{b} = 1$

◯ $\hat{a} = 1, \hat{b} = 0$

◯ $\hat{a} = 0, \hat{b} = \sqrt{3}$

◯ $\hat{a} = 1, \hat{b} = \sqrt{3}$

Solution: The easiest way to spot the correct solution is through process of elimination. If $\hat{b} = \frac{r\sigma_y}{\sigma_x} = \sqrt{3}$, then $r = 1$. This is clearly not true. Between the remaining two options, we can say that $\hat{a} = \bar{y} - \hat{b}\bar{x} = 1 - \hat{b} \cdot 0 = 1$ without calculating $\hat{b}$ since $\bar{x} = 0$.

Alternatively, the math-heavy way is shown below as well for the sake of completeness. Recall that the correlation is defined as:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{n\sigma_y\sigma_x}$$

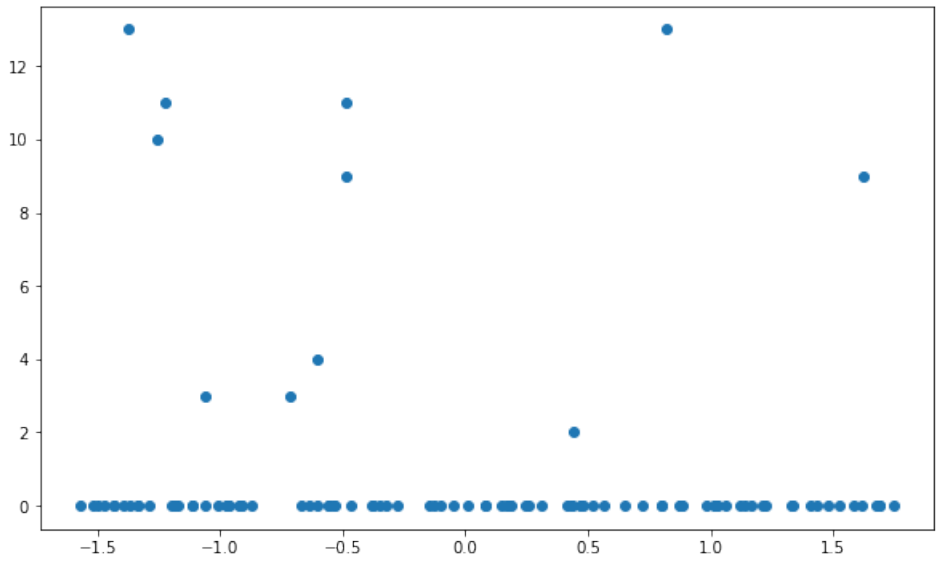
Since $s_x = 0$ and $\bar{x} = 0$:

$$r = \frac{\sum_i x_i y_i - \bar{y}\sum_i x_i}{n\sigma_y}$$

We know that $\sum_i x_i y_i = \frac{8}{9} - \frac{8}{9} = 0$, and $\bar{y}\sum_i x_i = n\bar{y}\bar{x} = 0$.

There is no correlation - so $\hat{b} = 0$. Then, $\hat{a} = \bar{y} = 1$.

(d) [2 Pts] Suppose you use the same definitions of v, x, and y as part (e) to train another least squares SLR model on a *larger* dataset; i.e., your data includes more observations, not just the 8 from part (e). Below we show a plot, but we do not label the axes. Which of the following could we be plotting? Select all that apply. At least one is correct.



☐ x-axis: x, y-axis: $y - \hat{y}$

☐ x-axis: x, y-axis: $\hat{y}$

☐ **x-axis: x, y-axis: y**

☐ x-axis: x, y-axis: $y - \bar{y}$

Justify your answer.

Solution: The first option is incorrect since the sum (and mean) of residuals $\sum_i y_i - \hat{y}$ must be 0. The y-axis starts at 0, so this is impossible.

The second option is incorrect since $\hat{y}$ is always linear in x; the model cannot make such non-linear predictions.

The third option is possible since x seems to be standardized and y is mostly 0 as expected.

The final option is also incorrect, because of similar reasons as the first option.

8 Linear Regression Fundamentals [5 Pts]

You want to estimate a quantity y as a function of x. Suppose you decide to model your estimate $\hat{y}$ as follows:

$$\hat{y} = \theta\sqrt{x}$$

Note that our model has one parameter, θ. Here's our data:

y	x	$\sqrt{x}$
2	9	3
1	4	2
0	1	1

(a) [2 Pts] What is the mean squared error (MSE) of our model over the data if we select $\theta = 2$?

- ◯ 2
- ◯ $11/3$
- ◯ $14/3$
- ◯ $29/3$
- ◯ $56/3$
- ◯ 56
- ◯ $\sqrt{2}$
- ◯ $\sqrt{11/3}$
- ◯ $\sqrt{14/3}$
- ◯ $\sqrt{29/3}$
- ◯ $\sqrt{56/3}$
- ◯ $\sqrt{56}$

(b) [3 Pts] Find the $\hat{\theta}$ that minimizes the mean squared error (MSE).

- ◯ 1/7
- ◯ 3/7
- ◯ **4/7**
- ◯ 5/7
- ◯ 1
- ◯ 10/7
- ◯ 12/7
- ◯ 2
- ◯ 14/7

Solution:

$$R(\theta) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \frac{1}{3}\sum_{i=1}^{3}(y_i - \theta\sqrt{t_i})^2$$

$$= \frac{1}{3}((2-3\theta)^2 + (1-2\theta)^2 + (0-\theta)^2)$$

Taking the derivative:

$$R'(\theta) = \frac{1}{3}(6(2-3\theta) + 4(1-2\theta) + 2(0-\theta))$$

$$= -\frac{1}{3}(12 - 18\theta + 4 - 8\theta - 2\theta)$$

$$= -\frac{1}{3}(16 - 28\theta)$$

Then, we set $L'(\theta) = 0$ to obtain $\hat{\theta} = \frac{4}{7}$.

9 Absolutely Simple Linear Regression [9 Pts]

Suppose you have a dummy dataset, sampled from an absolute value function, $y = \frac{1}{2}|x|$:

x	y
-1	$1/2$
$-1/2$	$1/4$
$1/2$	$1/4$
1	$1/2$

(a) [2 Pts] Calculate the constant model estimator for y using the data shown above assuming we are minimizing the average squared loss (i.e. MSE). Recall that the constant model is:

$$\hat{y} = \theta$$

Solution: The optimal solution, as covered in lecture is simply the mean.

The mean of the above set of y is $\frac{3}{8}$.

(b) [2 Pts] Which of the following is the least squares estimator $\hat{b}$ if we apply simple linear regression (SLR) to x and y with the data shown above? Recall that the SLR model is given by the equation below. Hint: Try plotting the data.

$$\hat{y} = a + bx$$

◯ $-\frac{\sigma_y}{\sigma_x}$

◯ $-\frac{\sigma_y}{2\sigma_x}$

◯ 0

◯ $\frac{\sigma_y}{2\sigma_x}$

◯ $\frac{\sigma_y}{\sigma_x}$

Solution: The correlation between x and y is 0, hence the correct answer is 0. Intuitively, this is because an absolute value function cannot be represented by a single straight line any better than a horizontal line going through the middle. Drawing out the absolute value function and attempting to fit a line to it can verify this.

An alternative mathematical solution follows, where we calculate r manually using a few nice properties from the above points - this is not the preferred way of approaching the problem as it involves some rote algebra. The derivation **looks quite complicated**, but it should become clear that all the terms will cancel at step 4 (this should also be almost the same derivation as with the MAE constant model). Another option is to manually plug in all x_i at step 2, which would yield 0 as well. Namely, we will use that $\bar{x} = 0$, $y_i = |x_i|$, and $\bar{y} = k$.

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{n\sigma_x\sigma_y} \tag{1}$$

$$= \frac{\sum_i x_i(|x_i| - k)}{n\sigma_x\sigma_y} \tag{2}$$

$$= \frac{\sum_{x_i<0} x_i(|x_i| - k) + \sum_{x_i\geq 0} x_i(|x_i| - k)}{n\sigma_x\sigma_y} \tag{3}$$

$$= \frac{\sum_{x_i<0} -x_i^2 - kx_i + \sum_{x_i\geq 0} x_i^2 - kx_i}{n\sigma_x\sigma_y} \tag{4}$$

$$= \frac{\sum_{x_i<0} -kx_i + \sum_{x_i\geq 0} -kx_i}{n\sigma_x\sigma_y} \tag{5}$$

$$= \frac{k\sum_i -x_i}{n\sigma_x\sigma_y} \tag{6}$$

$$= 0 \tag{7}$$

(c) [1 Pt] Compare the **loss** incurred on the training set by the SLR estimator in part (b) compared to the constant model estimator in part (a).

◯ Greater

◯ **Equal**

◯ Lesser

◯ Impossible to tell

Solution: Since $\hat{b} = 0$, $\hat{a} = \bar{y} - \hat{b}\bar{x}$. Since both $\hat{b}$ and $\bar{x}$ are 0, $\hat{a} = \bar{y}$. Note that these parameters are the exact same as the constant model, so the loss will be the same as well!

(d) [2 Pts] Suppose we apply a squared transformation to x such that $\tilde{x} = x^2$, and we fit another SLR model $y = \alpha_1 + \beta_1 \tilde{x}$ to the data, using least squares. Which of the following is true about the new least squares estimator $\hat{\beta}_1$?

- ◯ **It is greater than 0**
- ◯ It is zero
- ◯ It is less than 0
- ◯ Impossible to tell

Solution: If we apply a squared transformation to x, then all the $\tilde{x}$ will become positive. Hence, there will be a positive association between $\tilde{x}$ and y since all the negative x points are reflected across the y-axis. Since $\hat{b} = 0$, the new slope $\hat{\beta}_1$ must be greater.

A more concrete solution is to establish that $y = \frac{1}{2}|x| = \frac{1}{2}\sqrt{|x|^2} = \frac{1}{2}\sqrt{\tilde{x}}$. For the points given above, this forms a perfectly straight line!

(e) [2 Pts] Suppose we apply a squared transformation to *both* x and y such that $\tilde{x} = x^2$ and $\tilde{y} = y^2$, and we fit another SLR model $\widehat{\tilde{y}} = \alpha_3 + \beta_3 \tilde{x}$ to the data. Which of the following are the optimal $\hat{\alpha}_3$ and $\hat{\beta}_3$, assuming we minimize MSE?

- ◯ $\hat{\alpha}_3 = 0, \hat{\beta}_3 = \frac{1}{2}$
- ◯ $\hat{\alpha}_3 = 0, \hat{\beta}_3 = \frac{1}{4}$
- ◯ $\hat{\alpha}_3 = \frac{1}{2}, \hat{\beta}_3 = 0$
- ◯ $\hat{\alpha}_3 = \frac{1}{4}, \hat{\beta}_3 = 0$
- ◯ None of the above

Solution: Applying the transformation:

$y^2 = (\frac{1}{2}|x|)^2 = \frac{1}{4}x^2$

Since $\tilde{x} = x^2$ and $y = y^2$, $\tilde{y} = \frac{1}{4}\tilde{x}$. Hence, $\hat{\beta}_3 = \frac{1}{4}$ and $\hat{\alpha}_3 = 0$.

This page has been intentionally left blank.

Spring 2022 Data 100/200 Midterm 1 Reference Sheet

Pandas

Suppose `df` is a DataFrame; `s` is a Series. `pd` is the Pandas package.

Function	Description
`df[col]`	Returns the column labeled `col` from `df` as a Series.
`df[[col1, col2]]`	Returns a DataFrame containing the columns labeled `col1` and `col2`.
`s.loc[rows] / df.loc[rows, cols]`	Returns a Series/DataFrame with rows (and columns) selected by their index values.
`s.iloc[rows] / df.iloc[rows, cols]`	Returns a Series/DataFrame with rows (and columns) selected by their positions.
`s.isnull() / df.isnull()`	Returns boolean Series/DataFrame identifying missing values
`s.fillna(value) / df.fillna(value)`	Returns a Series/DataFrame where missing values are replaced by `value`
`df.drop(labels, axis)`	Returns a DataFrame without the rows or columns named `labels` along `axis` (either 0 or 1)
`df.rename(index=None, columns=None)`	Returns a DataFrame with renamed columns from a dictionary `index` and/or `columns`
`df.sort_values(by, ascending=True)`	Returns a DataFrame where rows are sorted by the values in columns `by`
`s.sort_values(ascending=True)`	Returns a sorted Series.
`s.unique()`	Returns a NumPy array of the unique values
`s.value_counts()`	Returns the number of times each unique value appears in a Series
`pd.merge(left, right, how='inner', on='a')`	Returns a DataFrame joining DataFrames `left` and `right` on the column labeled `a`; the join is of type `inner`
`left.merge(right, left_on=col1, right_on=col2)`	Returns a DataFrame joining DataFrames `left` and `right` on columns labeled `col1` and `col2`.
`df.pivot_table(index, columns, values=None, aggfunc='mean')`	Returns a DataFrame pivot table where columns are unique values from `columns` (column name or list), and rows are unique values from `index` (column name or list); cells are collected `values` using `aggfunc`. If `values` is not provided, cells are collected for each remaining column with multi-level column indexing.
`df.set_index(col)`	Returns a DataFrame that uses the values in the column labeled `col` as the row index.
`df.reset_index()`	Returns a DataFrame that has row index 0, 1, etc., and adds the current index as a column.

Let `grouped = df.groupby(by)` where `by` can be a column label or a list of labels.

Function	Description
`grouped.count()`	Return a Series containing the size of each group, excluding missing values
`grouped.size()`	Return a Series containing size of each group, including missing values
`grouped.mean()`/`grouped.min()`/`grouped.max()`	Return a Series/DataFrame containing mean/min/max of each group for each column, excluding missing values
`grouped.filter(f)` `grouped.agg(f)`	Filters or aggregates using the given function f

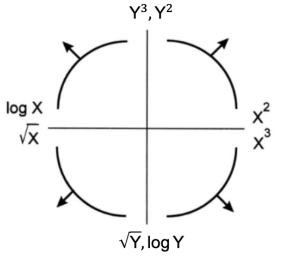
Function	Description
`s.str.len()`	Returns a Series containing length of each string
`s.str.lower()`/`s.str.upper()`	Returns a Series containing lowercase/uppercase version of each string
`s.str.replace(pat, repl)`	Returns a Series after replacing occurences of substrings matching regular expression `pat` with string `repl`
`s.str.contains(pat)`	Returns a boolean Series indicating whether a substring matching the regular expression `pat` is contained in each string
`s.str.extract(pat)`	Returns a Series of the first subsequence of each string that matches the regular expression `pat`. If `pat` contains one group, then only the substring matching the group is extracted

Visualization

Matplotlib: `x` and `y` are sequences of values.

Function	Description
`plt.plot(x, y)`	Creates a line plot of `x` against `y`
`plt.scatter(x, y)`	Creates a scatter plot of `x` against `y`
`plt.hist(x, bins=None)`	Creates a histogram of `x`; `bins` can be an integer or a sequence
`plt.bar(x, height)`	Creates a bar plot of categories `x` and corresponding heights `height`

Tukey-Mosteller Bulge Diagram.



Seaborn: `x` and `y` are column names in a DataFrame `data`.

Function	Description
`sns.countplot(data, x)`	Create a barplot of value counts of variable `x` from `data`
`sns.histplot(data, x, kde=False)` `sns.displot(x, data, rug = True, kde = True)`	Creates a histogram of `x` from `data`; optionally overlay a kernel density estimator. `displot` is similar but can optionally overlay a rug plot.
`sns.boxplot(data, x=None, y)` `sns.violinplot(data, x=None, y)`	Create a boxplot of y, optionally factoring by categorical `x`, from `data`. `violinplot` is similar but also draws a kernel density estimator of `y`.
`sns.scatterplot(data, x, y)`	Create a scatterplot of `x` versus `y` from `data`
`sns.lmplot(x, y, data, fit_reg=True)`	Create a scatterplot of `x` versus `y` from `data`, and by default overlay a least-squares regression line
`sns.jointplot(x, y, data, kind)`	Combine a bivariate scatterplot of `x` versus `y` from `data`, with univariate density plots of each variable overlaid on the axes; `kind` determines the visualization type for the distribution plot, can be `scatter`, `kde` or `hist`

Regular Expressions

List of all metacharacters: `. ^ $ * + ? ] [ \ | ( ) { }`

Operator	Description	Operator	Description
`.`	Matches any character except **\n**	`*`	Matches preceding character/group zero or more times
`\\`	Escapes metacharacters	`?`	Matches preceding character/group zero or one times
`|`	Matches expression on either side of expression; has lowest priority of any operator	`+`	Matches preceding character/group one or more times
`\d`, `\w`, `\s`	Predefined character group of digits (0–9), alphanumerics (a–z, A–Z, 0–9, and underscore), or whitespace, respectively	`^`, `$`	Matches the beginning and end of the line, respectively
`\D`, `\W`, `\S`	Inverse sets of `\d`, `\w`, `\s`, respectively	`( )`	Capturing group used to create a sub-expression
`{m}`	Matches preceding character/group exactly `m` times	`[ ]`	Character class used to match any of the specified characters or range (e.g. `[abcde]` is equivalent to `[a-e]`)
`{m, n}`	Matches preceding character/group at least `m` times and at most `n` times if either `m` or `n` are omitted, set lower/upper bounds to 0 and ∞, respectively	`[^ ]`	Invert character class; e.g. `[^a-c]` matches all characters except `a`, `b`, `c`

Function	Description
`re.match(pattern, string)`	Returns a match if zero or more characters at beginning of `string` matches `pattern`, else None
`re.search(pattern, string)`	Returns a match if zero or more characters anywhere in `string` matches `pattern`, else None
`re.findall(pattern, string)`	Returns a list of all non-overlapping matches of `pattern` in `string` (if none, returns empty list)
`re.sub(pattern, repl, string)`	Returns `string` after replacing all occurrences of `pattern` with `repl`

Modified lecture example for a single capturing group:

```
lines = '169.237.46.168 - - [26/Jan/2014:10:47:58 -0800] "GET ... HTTP/1.1"'
re.findall(r'\[\d+\/(\w+)\/\d+:\d+:\d+:\d+ .+\]', line) # returns ['Jan']
```

Modeling

Concept	Formula	Concept	Formula
L_1 loss	$L_1(y, \hat{y}) = \mid y - \hat{y} \mid$	Correlation r	$r = \frac{1}{n}\sum_{i=1}^{n} \frac{x_i - \bar{x}}{\sigma_x}\frac{y_i - \bar{y}}{\sigma_y}$
L_2 loss	$L_2(y, \hat{y}) = (y - \hat{y})^2$	Linear regression prediction of y	$\hat{y} = a + bx$
Empirical risk with loss L	$R(\theta) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i)$	Least squares linear regression, slope $\hat{b}$	$\hat{b} = r\frac{\sigma_y}{\sigma_x}$
		Least squares linear regression, intercept $\hat{a}$	$\hat{a} = \bar{y} - \hat{b}\bar{x}$