Data C100/C200, Final

Spring 2022

Name: ________________________________

Email: ____________________________@berkeley.edu

Student ID: ________________________________

Exam Room: ________________________________

Name and SID of left neighbor: ________________________________

Name and SID of right neighbor: ________________________________

Instructions:

This final exam consists of **160 points** spread out over **14 questions** and the Honor Code and must be completed in the **170 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.

Q0 [1 Pt]: Honor Code

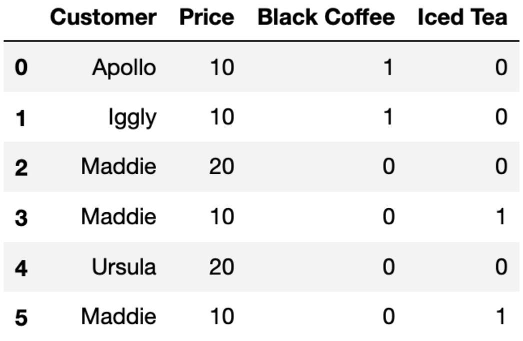
As a member of the UC Berkeley community, I act with honesty, integrity, and respect for others. I am the person whose name is on the exam and I completed this exam in accordance with the Honor Code.

Signature: ________________________________

This page has been intentionally left blank.

1 Coo…You Want Pigeon Milk in That? [15 pts]

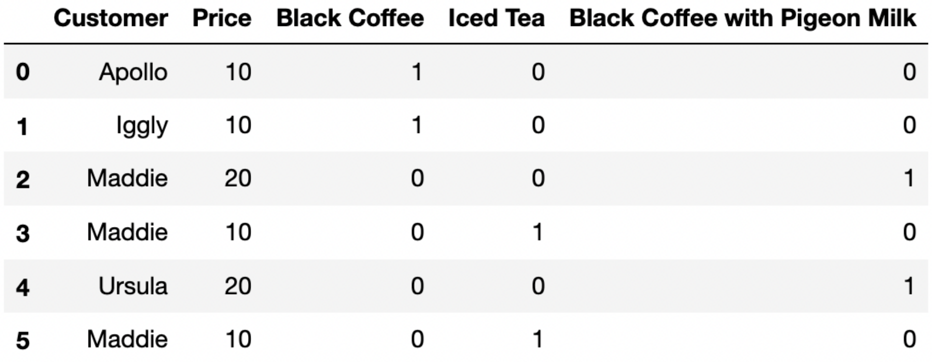
Brewster, a coffee barista, has opened a new café called “The Roost.” You are tasked with helping him manage his daily sales. You are provided with a pandas DataFrame `roost` of today’s customer transactions, the first 6 rows of which are shown below. **Note that customers can only buy one drink at a time.**



	Customer	Price	Black Coffee	Iced Tea
0	Apollo	10	1	0
1	Iggly	10	1	0
2	Maddie	20	0	0
3	Maddie	10	0	1
4	Ursula	20	0	0
5	Maddie	10	0	1

(a) [3 Pts] “The Roost” sells three drinks, but only two are listed in the `roost` DataFrame. It seems Brewster one-hot encoded the column representing the drink the customer bought. However, he accidentally dropped one of the columns. Update `roost` with the missing column representing the third drink, “Black Coffee with Pigeon Milk.”

After running your code, the first 6 rows of the updated `roost` DataFrame should be:



	Customer	Price	Black Coffee	Iced Tea	Black Coffee with Pigeon Milk
0	Apollo	10	1	0	0
1	Iggly	10	1	0	0
2	Maddie	20	0	0	1
3	Maddie	10	0	1	0
4	Ursula	20	0	0	1
5	Maddie	10	0	1	0

```
roost['Black Coffee with Pigeon Milk'] =
```

Solution:

```
roost['Black Coffee with Pigeon Milk'] =
    1 - roost['Black Coffee'] - roost['Iced Tea']
```

(b) [3 Pts] Suppose we want to build a linear model that predicts the column you just added ("Black Coffee with Pigeon Milk"):

$$\hat{y} = \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3$$

Let features x_1 be `Price`, x_2 be `Black Coffee`, x_3 be `Iced Tea`, and $\hat{y}$ be your prediction of `Black Coffee with Pigeon Milk`. What would be the optimal parameter estimates $\theta = \begin{bmatrix}\theta_1, & \theta_2, & \theta_3\end{bmatrix}^T$ such that the model predicts $\hat{y}$ exactly equal to the "Black Coffee with Pigeon Milk" column?

○ A. $\begin{bmatrix}1/2, & 1/20, & 1/20\end{bmatrix}^T$

○ **B.** $\begin{bmatrix}1/20, & -1/2, & -1/2\end{bmatrix}^T$

○ C. $\begin{bmatrix}1/2, & -1/2, & -1/2\end{bmatrix}^T$

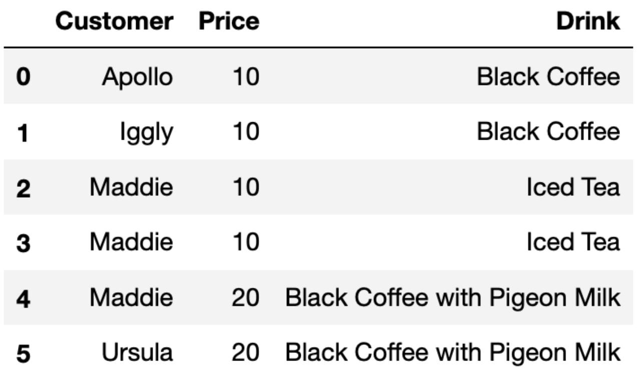
○ D. $\begin{bmatrix}1/20, & 1/20, & 1/20\end{bmatrix}^T$

○ E. $\begin{bmatrix}1/2, & 1/20, & -1/2\end{bmatrix}^T$

○ F. $\begin{bmatrix}1/10, & -1/2, & 1/10\end{bmatrix}^T$

○ G. $\begin{bmatrix}1/2, & 1/10, & 1/10\end{bmatrix}^T$

(c) [4 Pts] Suppose Brewster wants to transform the one-hot encoded data into a new DataFrame that has just three columns: the customer name `Customer`, drink price `Price`, and drink name `Drink`. The resulting DataFrame `roost_transactions` should as follows:



	Customer	Price	Drink
0	Apollo	10	Black Coffee
1	Iggly	10	Black Coffee
2	Maddie	10	Iced Tea
3	Maddie	10	Iced Tea
4	Maddie	20	Black Coffee with Pigeon Milk
5	Ursula	20	Black Coffee with Pigeon Milk

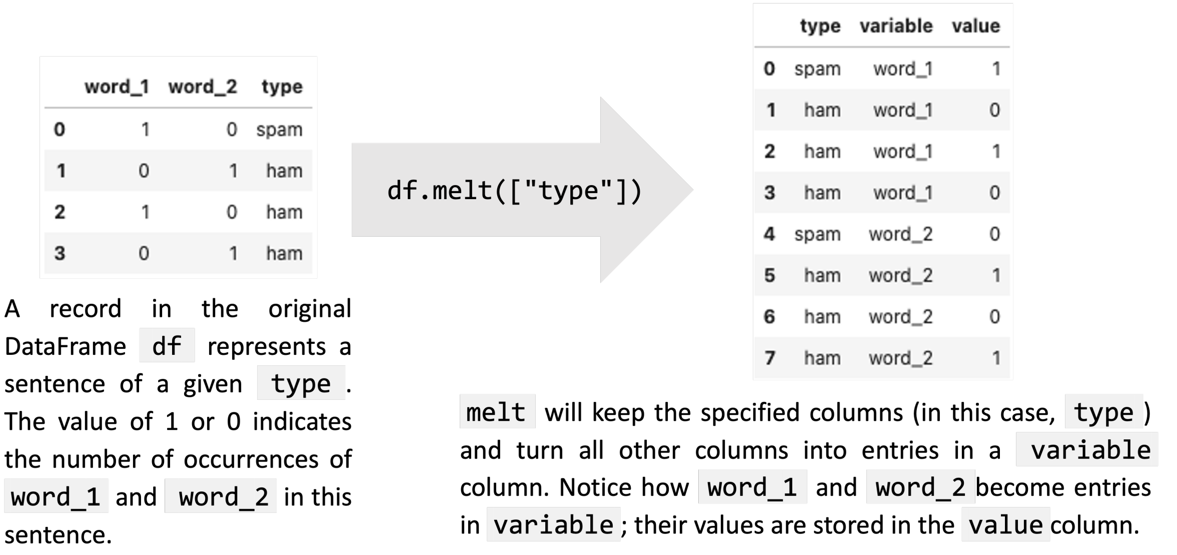
Complete the following skeleton code. (Hint: Documentation for `melt` is on the following page.)

```
df = roost.melt([__(i)__])
df = __(ii)__.drop(columns=['value'])
df = df.rename(columns={'variable':'Drink'})
roost_transactions = df.reset_index(drop=True)
```

(i): ____

(ii): ____

Here is a refresher on the Pandas `melt` method, paraphrased from Project 2A:



	word_1	word_2	type
0	1	0	spam
1	0	1	ham
2	1	0	ham
3	0	1	ham

```
df.melt(["type"])
```

	type	variable	value
0	spam	word_1	1
1	ham	word_1	0
2	ham	word_1	1
3	ham	word_1	0
4	spam	word_2	0
5	ham	word_2	1
6	ham	word_2	0
7	ham	word_2	1

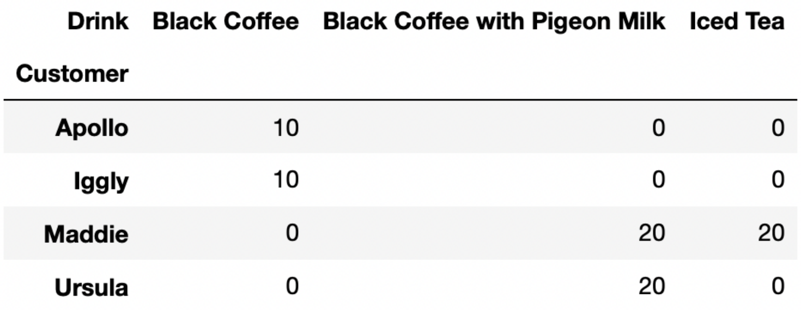
A record in the original DataFrame `df` represents a sentence of a given `type`. The value of 1 or 0 indicates the number of occurrences of `word_1` and `word_2` in this sentence.

`melt` will keep the specified columns (in this case, `type`) and turn all other columns into entries in a `variable` column. Notice how `word_1` and `word_2` become entries in `variable`; their values are stored in the `value` column.

Solution:

```
df = roost.melt(['Customer', 'Price'])
df = df[df['value'] == 1].drop(columns=['value'])
roost_transactions = df.rename(columns={'variable':'Drink'})
```

(d) [5 Pts] Finally, Brewster needs your help to construct a pivot table from `roost_transactions` that looks similar to the below table. The resulting pivot table should contain a column for each type of drink, where each record now represents a **unique customer** and how much they spent on **each type of item**.



Drink	Black Coffee	Black Coffee with Pigeon Milk	Iced Tea
Customer			
Apollo	10	0	0
Iggly	10	0	0
Maddie	0	20	20
Ursula	0	20	0

```
roost_transactions.pivot_table(__D__).fillna(0)
```

Fill in the blank for D:

____

____

Solution:

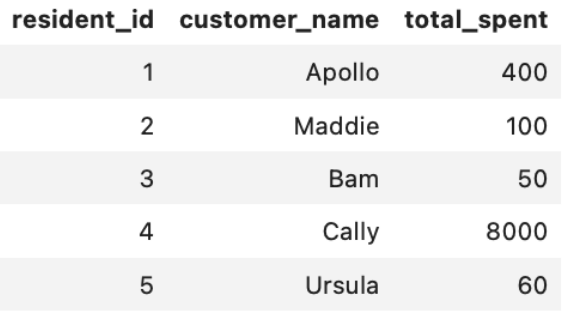
```
# alternate
pd.pivot_table(roost_transactions, values='Price',
    index='Customer', columns='Drink', aggfunc = np.sum,
    ).fillna(0)

# matches reference sheet
roost_transactions.pivot_table(
    index='Customer', columns='Drink',
    values='Price', aggfunc = np.sum,
   ).fillna(0)
```

2 Excellent Purchase! (...Purchase!) [15 pts]

(a) [6 Pts] Isabelle and Tom Nook have opened up competing stores on an island. They've each collected data, and they have asked you to help determine who is the better salesperson.

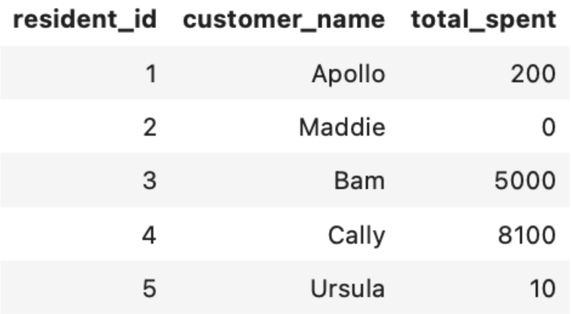
Isabelle and Tom collect data in two separate tables; each table has one record for every resident on the island, as well as how much that resident spent at that store in a particular day. You are provided two tables: one for Isabelle's store (`Isabelle`) and one for Tom's store (`Tom`). The first five rows of each table are shown below:

Isabelle



resident_id	customer_name	total_spent
1	Apollo	400
2	Maddie	100
3	Bam	50
4	Cally	8000
5	Ursula	60

Tom

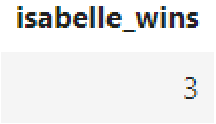


resident_id	customer_name	total_spent
1	Apollo	200
2	Maddie	0
3	Bam	5000
4	Cally	8100
5	Ursula	10

Construct a SQL query below that returns just one row that contains the total number of wins Isabelle got. Someone gets a 'win' if they get a resident to spend more than the other person. For example, Apollo spent more with Isabelle than Tom, so that would represent one win for Isabelle. For the first five rows, Isabelle has a total of 3 wins, since Apollo, Maddie, and Ursula spent more money at Isabelle's store.

Construct this SQL query using the skeleton code below. The result for the first five rows is shown to the right.

```
SELECT _(i)_ AS isabelle_wins
FROM __(ii)__ INNER JOIN Isabelle
_(iii)_
```



isabelle_wins
3

(i): ____

(ii): ____

(iii): ____

Solution:

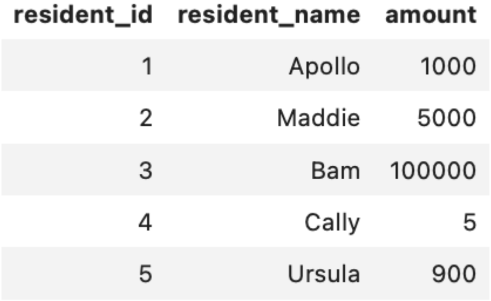
```
SELECT COUNT(*) AS isabelle_wins
FROM tom INNER JOIN isabelle
```

```
ON tom.resident_id = isabelle.resident_id
    AND isabelle.total_spent > tom.total_spent

or

SELECT SUM(isabelle.total_spent > tom.total_spent) AS isabelle_wins
FROM tom INNER JOIN isabelle
ON isabelle.customer_name = tom.customer_name
```

(b) [9 Pts] Isabelle and Tom also jointly own an online shopping catalogue. Residents can purchase items from the catalogue only if they have a bank account.

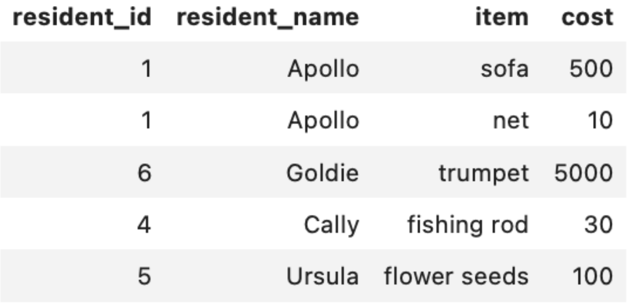
Pelly, the island's accountant, maintains a SQL table `Account` that contains all owners of bank accounts and the monetary amount in each account. Isabelle and Tom also have a shared SQL table `Transaction` of attempted transactions from customers through their online catalogue. Both tables are shown below:

Account



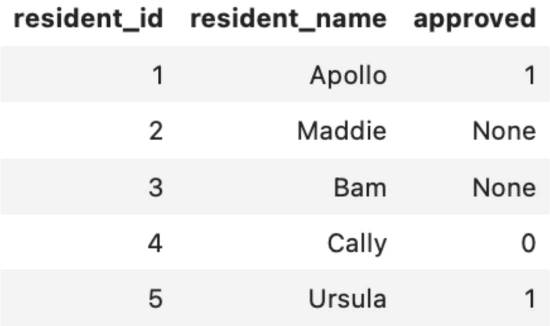
resident_id	resident_name	amount
1	Apollo	1000
2	Maddie	5000
3	Bam	100000
4	Cally	5
5	Ursula	900

Transaction



resident_id	resident_name	item	cost
1	Apollo	sofa	500
1	Apollo	net	10
6	Goldie	trumpet	5000
4	Cally	fishing rod	30
5	Ursula	flower seeds	100

Construct a SQL query below that returns the output table to the right, which contains a column `approved` that indicates if a customer has enough money stored in their bank account to complete **all the transactions** they are attempting: 1 (True), 0 (False), or NULL/None (has a bank account, but did not make any transactions).



resident_id	resident_name	approved
1	Apollo	1
2	Maddie	None
3	Bam	None
4	Cally	0
5	Ursula	1

- All residents with bank accounts should be in the output table.
- If a customer attempts to make a transaction but they don't have a bank account, they should not be included in the output table (e.g., Goldie is not included).
- The input tables have no NULL/None values in them.

Construct this SQL query using the skeleton code below.

```
SELECT a.resident_id, a.resident_name, _(i)_ AS approved
FROM Account AS a  _(ii)_ JOIN Transaction AS t
__(iii)__
```

(i): ____

(ii): ○ INNER ○ LEFT ○ RIGHT ○ OUTER ○ CROSS

(iii): ____

____

Solution:

```
SELECT a.resident_id, a.resident_name, a.amount > SUM(t.cost)

AS approved FROM Account AS a LEFT JOIN Transaction AS t

ON a.resident_id = t.resident_id GROUP BY a.resident_id
```

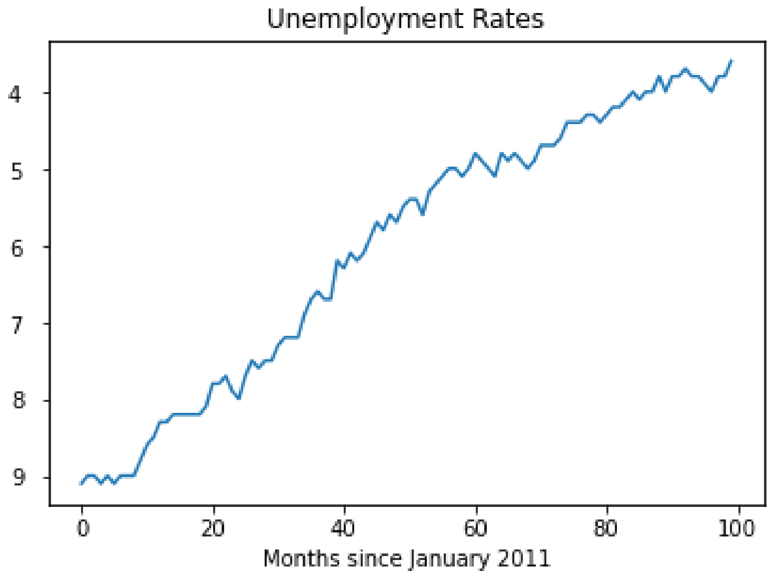
3 Jolly Redd's Visualization Trawler [8 pts]

You have just bought some visualizations from the shady visualization seller, Jolly Redd. However, there are some flaws in the visualizations he's sold you.

While we have some clear answers in mind, we will be lenient when grading this question. As long as you correctly identify some flawed aspect of the image and explain your answer thoroughly, you will receive full credit. Answers such as ”there is no flaw” or answers about the underlying data (and not the image itself) will receive no credit. **Please keep your answers concise—nothing more than a couple of sentences.**

Note that the titles for these plots are given directly above the image—answers such as “bad title” or “missing title” will also receive no credit.

(a) [4 Pts]



List two aspects of this plot that are incorrect or misleading.

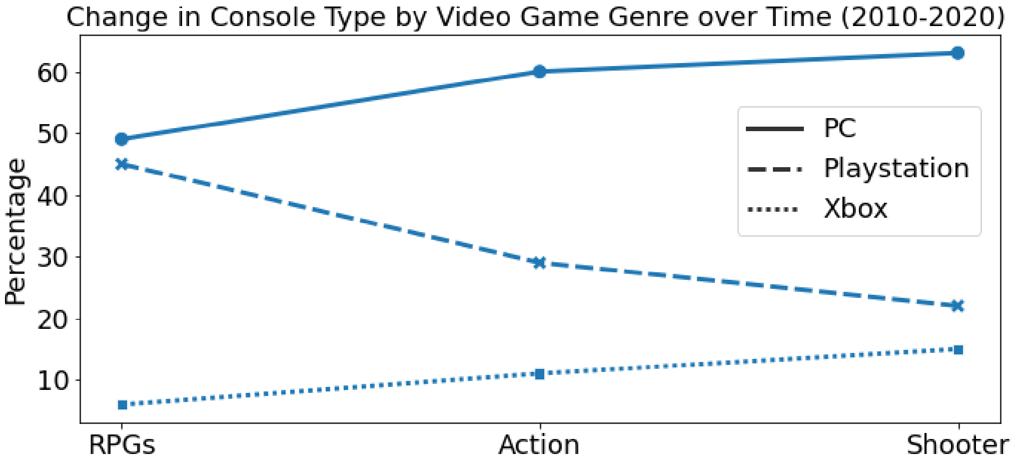
Solution: Possible answers:

- flipped axes
- missing axis labels

1.

2.

(b) [4 Pts] Below, “Console Type” refers to PC, Playstation and Xbox; and “Video Game Genre” refers to RPGs, Action, and Shooter.



Describe two flaws with this plot.

Solution: Possible answers for issue with plot:

- Plots categorical data as a line
- Mentions time in title, but time is not shown in plot

1.

2.

4 Dr. How Do You Spell That? [8 pts]

You are working with a British researcher, Dr. Who, to file a lab report in the Data 100 Research Journal. Unfortunately, the journal only accepts American versions of words, but Dr. Who's lab report uses British words. For example "colour" is a British spelling of the word "color".

(a) [3 Pts] First, determine the output of the `re.findall` statement below.

```
>>> sentences = [
        "Please analyse the catalysers.",
        "You can't psychoanalyse a lysed protein.",
    ]
>>> pattern = r"\w\wyse"
>>> [re.findall(pattern, s) for s in sentences]
...[['alyse', 'alyse'], _______X_______]
```

What result will be in the blank marked with an X?

○ A. `[]`

○ [✓ **B.** `['alyse']`]

○ C. `['lyse']`

○ D. `['alyse', 'lyse']`

○ E. `['lyse', 'lyse']`

(b) [5 Pts] Now you try to identify some British spelling idiosyncrasies in Dr. Who's writing. Write a regular expression that will find all words that contain "our", *excluding* words with 0 or 1 letters before "our". In other words, your regular expression should find "colour", and "favourite", but should not find "our", or "dour", as they have 0 and 1 letter preceding the "our", respectively. Unlike part (a), you want the results of your `findall` to be the **entire word**, not just some of the letters, as shown in the example below.

```
>>> sentences = [
    "Our favourite colour is blue.",
    "I am four hours away from the harbour.",
    "I am enamoured with our tour of the arbour."
    ]
>>> pattern = r"__B__"
>>> [re.findall(pattern, s) for s in sentences]
...[['favourite', 'colour'], ['harbour'], ['enamoured', 'arbour']]
```

Fill in the blank for B such that the above code works correctly.

```
pattern = r"________________________________________"
```

Solution:

```
pattern = r'\w{2,}our\w*'
```

5 Back to the Future [3 pts]

You successfully sneaked in a survey on KPop groups and a survey on cats vs dogs on this semester's Data 100 exams! Let's do a math problem on the result of the survey.

(a) [3 Pts] Recall the definition of a *multinomial probability* from lecture:

If we are drawing at random with replacement n times, from a population broken into three separate categories (where $p_1 + p_2 + p_3 = 1$):

- Category 1, with proportion p_1 of the individuals.
- Category 2, with proportion p_2 of the individuals.
- Category 3, with proportion p_3 of the individuals.

Then, the probability of drawing k_1 individuals from Category 1, k_2 individuals from Category 2, and k_3 individuals from Category 3 (where $k_1 + k_2 + k_3 = n$) is:

$$\frac{n!}{k_1!k_2!k_3!}p_1^{k_1}p_2^{k_2}p_3^{k_3}$$

From the **original** results of your survey, you learn that 14% of Data 100 students are BTS fans and 24% of Data 100 students are Blackpink fans and the rest are fans of neither. Suppose you randomly sample **with replacement** 99 students from the class. What is the probability that the students are evenly distributed between the three different groups? Please leave your answer as an expression; there is no need to fully calculate it out.

Solution:

$$\frac{99!}{33!33!33!}0.14^{33}0.24^{33}(1 - 0.14 - 0.24)^{33}$$

(b) [0 Pts] Which of the surveys did you prefer? Select your favorite between the two.

○ **A. BTS/Blackpink survey from Midterm 1.**

○ B. Cats vs Dogs Drawing survey from Midterm 2.

○ C. Neither.

6 Election Perfection [22 pts]

Ariel has recently joined a data science company in Phosphorus Valley to perform data analysis for their elections. To start off, Ariel's first task is to create a linear model that performs one task: The model must perfectly predict the winners of the local election that has 2 candidates given their vote share.

Ariel was provided the results of the **past 150 elections**, a sample of which is shown below. Assume for this question that the largest vote share wins the election (i.e., strictly more than 50%), and that there are never any ties (i.e., Candidate 0 and 1 will never both get 50% of the vote share).

Candidate 0	Candidate 1	Winner
0.35	0.65	Candidate 1
0.12	0.88	Candidate 1
0.51	0.49	Candidate 0
0.65	0.35	Candidate 0
0.40	0.60	Candidate 1
. . .	. . .	

$$\Rightarrow \quad \mathbb{X} = \begin{bmatrix} 0.35 & 0.65 \\ 0.12 & 0.88 \\ 0.51 & 0.49 \\ 0.65 & 0.35 \\ 0.40 & 0.60 \\ \ldots & \ldots \end{bmatrix} \quad \mathbb{Y} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ \ldots \end{bmatrix}$$

Above, Ariel has constructed a design matrix $\mathbb{X}$, where the first feature represents Candidate 0's vote share and the second feature represents Candidate 1's vote share in each election, and an output vector $\mathbb{Y}$, which represents the election winners (1: Candidate 1; 0: Candidate 0). Help Ariel predict the election winners given the design matrix $\mathbb{X}$ and your linear modeling toolkit.

(a) [5 Pts] Suppose you limit the training data matrix to **two** arbitrary training points and construct the 2x2 data matrix $\mathbb{X}_2$ and the corresponding election winners $\mathbb{Y}_2$. Fill in the blank:

If we train a linear regression model fit to the training data $\mathbb{X}_2$ and $\mathbb{Y}_2$, we can ____________ achieve zero training loss.

○ **A. always** ○ B. sometimes ○ C. never

Justify your answer:

Solution: The first option is correct. In ”normal” cases, the matrix is full rank and square, so it can be inverted. In this case, we always achieve perfect loss.

If the two elections had the same results in terms of the **vote share** in both elections (which is the only way to have linear dependence), where the vote shares were both (a, b). In this case, if Candidate 0 is the winner, we set $\theta_0 = \theta_1 = 0$ and if Candidate 1 is the winner, we set $\theta_0 = 0$ and $\theta_1 = \frac{1}{b}$. This isn't a unique solution, but that's okay in this question!

(b) [4 Pts] Ariel decides to improve the above model by training on the entire dataset of 150 past elections. However, Ariel decides to use ordinary least squares (OLS) with a a design matrix $\tilde{\mathbb{X}}$, which is the $\mathbb{X}$ matrix with an added intercept feature, to compute $\hat{\theta}$ using the normal equation: $\hat{\theta} = (\tilde{\mathbb{X}}^T\tilde{\mathbb{X}})^{-1}\tilde{\mathbb{X}}^T\mathbb{Y}$. For your convenience, a sample of $\tilde{\mathbb{X}}$ and $\mathbb{Y}$ are shown below.

$$\tilde{\mathbb{X}} = \begin{bmatrix} 1 & 0.35 & 0.65 \\ 1 & 0.12 & 0.88 \\ 1 & 0.51 & 0.49 \\ 1 & 0.65 & 0.35 \\ 1 & 0.40 & 0.60 \\ \dots & \dots & \dots \end{bmatrix} \qquad \mathbb{Y} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ \dots \end{bmatrix}$$

Which of the following are true about this modeling approach?

□ A. $\tilde{\mathbb{X}}^T\mathbb{Y}$ is of size $(150, 3)$.

□ **B. $\tilde{\mathbb{X}}^T\mathbb{Y}$ is a vector.**

□ **C. Ariel cannot achieve perfect accuracy nor zero loss on the training set using this approach.**

□ **D. Using L_2 regularization with $\lambda > 0$ is always a better choice than OLS for the dataset above.**

Solution: The first option is correct since $\mathbb{Y}$ essentially is a representation of when the second candidate wins; it is 0 when the first candidate wins and 1 when the second candidate wins. Hence, when computing $\mathbb{X}^T\mathbb{Y}$, the dot product of the bias vector with $\mathbb{Y}$ is: $\sum_i y_i$. Note that this is the number of times candidate 1 wins!

The third and fourth options are correct. Note that there is clear linear dependence if we have a bias term since the sum of all of the vectors in the original data matrix $\mathbb{X}$ is $\vec{1}$ (i.e. the total vote share). Hence, Sid cannot achieve perfect accuracy on any data set. By using L_2 regularization, we can avoid this issue!

(c) [4 Pts] Ariel decides to switch to a logistic regression model and also reverts to the original 2-feature design matrix $\mathbb{X}$, with **no bias term**. A sample of $\mathbb{X}$ and $\mathbb{Y}$ is shown below:

$$\mathbb{X} = \begin{bmatrix} 0.35 & 0.65 \\ 0.12 & 0.88 \\ 0.51 & 0.49 \\ 0.65 & 0.35 \\ 0.40 & 0.60 \\ \dots & \dots \end{bmatrix} \qquad \mathbb{Y} = \begin{bmatrix} 1 \\ 1 \\ 0 \\ 0 \\ 1 \\ \dots \end{bmatrix}$$

Which of the following is true about Ariel's modeling approach using **logistic regression** with the constructed design matrix?

□ A. The design matrix is not full column rank.

□ **B. Logistic regression outputs the probability that Candidate 1 wins the election.**

□ **C. There is no uniquely optimal parameter value $\hat{\theta}$ for logistic regression with this design matrix using binary cross-entropy loss.**

□ **D. Logistic regression would usually be used for this binary classification task.**

(d) [5 Pts] Regardless of your previous answer, suppose that Ariel fits a logistic regression model to the data in part (c). Provide an optimal parameter $\hat{\theta}$ such that we can achieve the maximum training **accuracy** possible for this data. What is the maximum possible training accuracy? Justify your answer.

Optimal $\hat{\theta}$: ____________________ Maximum possible training accuracy: ________________

Justification:

Solution: It's linearly separable, so there are infinitely many optimal parameters. One possible optimal parameter is $[-1, 1]^T$ since it divides losses and wins for Candidate 1 perfectly. Hence, we achieve perfect classification accuracy!

In general, any answer of the form $[-k, k]^T$ will work too.

(e) [4 Pts] One of Ariel's friends from Data 100 mentions that decision trees almost always achieve perfect accuracy, at least on the training dataset! As their final modeling approach, Ariel fits a decision tree to the 2-feature design matrix $\mathbb{X}$ and output $\mathbb{Y}$ as described in part (c).

What is the height of the optimal decision tree if we use weighted node cross-entropy loss? Justify your answer. Define the "height" of a decision tree as the maximum number of yes/no questions (a.k.a. splits) that can possibly be asked before arriving at a prediction.

Height of optimal decision tree: ____________________

Justification:

Solution: The height is 1, since we only need one split! If $x_1 > 0.5$, then Candidate 1 wins; otherwise, Candidate 0 wins. This happens with perfect accuracy.

7 Error Fn: Broken [21 pts]

Alex has compiled a list of **broken** loss functions and requests your help to diagnose drawbacks.

For each of the loss functions L below, assume a **linear model** $\hat{y} = f_\theta(x) = \theta^T x$. Then, the optimal parameter $\hat{\theta}$ is a **real-valued vector** that minimizes the empirical risk $\frac{1}{n}\sum_{i=1}^{n} L(y_i, \hat{y}_i)$ with respect to θ on a n-point dataset, where datapoint i has input x_i and output y_i.

(a) [3 Pts] Which of the following are true of the following cubic loss function L_1 when we minimize the associated empirical risk with respect to θ?

$$L_1(y, \hat{y}) = (y - \hat{y})^3$$

□ A. There are infinitely many optimal parameters for this loss function.

□ **B. Gradient descent on this loss function diverges, i.e. approaches one or more parameters of infinite magnitude.**

□ **C. This loss function is non-convex.**

□ D. This loss function is non-differentiable on at least one point in its domain.

(b) [3 Pts] Which of the following are true of the following negative squared loss function L_2 when we minimize the associated empirical risk with respect to θ?

$$L_2(y, \hat{y}) = -(y - \hat{y})^2$$

□ A. There are infinitely many optimal parameters for this loss function.

□ **B. Gradient descent on this loss function diverges, i.e. approaches one or more parameters of infinite magnitude.**

□ **C. This loss function is non-convex.**

□ D. This loss function is non-differentiable on at least one point in its domain.

(c) [3 Pts] Which of the following are true of the following loss function L_3 that calculates accuracy when we minimize the associated empirical risk with respect to θ?

$$L_3(y, \hat{y}) = \begin{cases} 0 & y \neq \hat{y} \\ 1 & y = \hat{y} \end{cases}$$

□ **A. There are infinitely many optimal parameters for this loss function.**

□ B. Gradient descent on this loss function diverges, i.e. approaches one or more parameters of infinite magnitude.

□ **C. This loss function is non-convex.**

□ **D. This loss function is non-differentiable on at least one point in its domain.**

Alex now provides you with the sample dataset shown to the right, with one feature x and response y.

x	y
36	4
25	9
16	16

(d) [6 Pts] Suppose we use a **simple linear regression model** with an intercept term, i.e., $\hat{y} = \theta_0 + \theta_1 x$ with the loss function L_4 (shown right). What is the optimal $\hat{\theta} = (\hat{\theta}_0, \hat{\theta}_1) \in \mathbb{R}^2$ that minimizes the empirical risk $\frac{1}{n}\sum_{i=1}^{n} L_4(y_i, \hat{y}_i)$ on the sample dataset? Justify your answer.

$$L_4(y, \hat{y}) = |\hat{y}| - |y|$$

$\hat{\theta}_0$: ______________ $\hat{\theta}_1$: ______________

Solution: Note that the $|y_i|$ term is simply a constant in the empirical risk, and consequently, we are essentially minimizing $|\hat{y}_i|$. The global minimum for this function is at $\hat{y}_i = 0$, which implies that for all i, we wish to predict 0.

This is only possible if $\hat{\theta} = [0, 0]^T$.

(e) [6 Pts] Suppose we use a **constant model**, i.e., $\hat{y} = \theta$ with the loss function L_5 (shown right). What is the optimal $\hat{\theta} \in \mathbb{R}$ that minimizes the empirical risk $\frac{1}{n}\sum_{i=1}^{n} L_5(y_i, \hat{y}_i)$ on the sample dataset? Justify your answer.

$$L_5(y, \hat{y}) = (\hat{y} - \sqrt{y})^2$$

$\hat{\theta}$: ____________

Solution: The optimal $\hat{\theta} = 3$ since this is a constant model with squared loss fit to $\sqrt{y}$. Hence, the optimal value is simply the mean of all of the $\sqrt{y}$, which is $\frac{2+3+4}{3} = 3$.

8 The English Alphabet Goes α, $|\mathcal{B}|$, C(ross-validation) [12 pts]

Jordan is using stochastic gradient descent to optimize a model with 1,000 parameters that can detect vehicles and pedestrians in images. However, Jordan is having trouble choosing hyperparameters such as the learning rate α and the batch size $|\mathcal{B}|$.

(a) [4 Pts] Complete the code snippet below to complete a function that performs `num_iter` stochastic gradient descent updates given a dataset `X` and `y`. Assume the functions `get_sample` and `grad_func` have already been defined.

```
def get_sample(X, y, B):
    """
    Returns a random sample with replacement of X and y,
    each of size B.
    """
    # Implementation not shown

def grad_func(X, y):
    """
    Returns the gradient of the empirical risk with respect to
    theta given the dataset X and y.
    The returned output is a np.ndarray of length 1000.
    """
    # Implementation not shown

def sgd_fit(X, y, alpha, B, num_iter=50000):
    theta = np.zeros(1000)
    for i in range(num_iter):
        X_batch, y_batch = get_sample(X, y, B)

        grad = grad_func(__________________, __________________)

        theta = ____________________________________________________
    return theta
```

(b) [2 Pts] As α increases beyond its optimal value, SGD will likely oscillate _____. Fill in the blank with the appropriate choice of less/more.

○ A. less ○ **B. more**

(c) [2 Pts] As $|\mathcal{B}|$ increases, SGD will likely oscillate _____. Fill in the blank with the appropriate choice of less/more.

○ **A. less** ○ B. more

(d) [2 Pts] Suppose Jordan employs k-fold cross-validation to compute optimal values of the hyperparameters α and $|\mathcal{B}|$. Which of the following is true?

□ A. k-fold cross-validation will not yield useful information for selecting α or $|\mathcal{B}|$, and is instead intended only for selecting the regularization parameter λ for regularized models.

□ **B. k-fold cross-validation may yield suboptimal hyperparameters for α and $|\mathcal{B}|$.**

□ C. k-fold cross-validation cannot output an optimal set of hyperparameters if the empirical risk is non-convex.

□ D. To use cross-validation to find the optimal α or $|\mathcal{B}|$, the loss must be differentiable with respect to α and $|\mathcal{B}|$.

(e) [2 Pts] Regardless of your answer to the previous question, suppose Jordan uses k-fold cross-validation with α chosen from $0.1, 0.2, 0.4$ and $|\mathcal{B}|$ chosen from $32, 64, 128$. The average cross-validated loss is shown in the below table for each combination of α and $|\mathcal{B}|$.

$\lvert\mathcal{B}\rvert$	α		
	0.1	0.2	0.4
32	0.0022	0.7031	0.0370
64	0.0051	0.9075	0.0471
128	0.0018	0.6007	0.0157

Which of the following is the **most optimal pair** of α and $|\mathcal{B}|$? Fill in the blanks.

α = ________________ and $|\mathcal{B}|$ = ________________

Solution: The most optimal pair to minimize the average CV loss is $\alpha = 0.1$ and $|\mathcal{B}| = 128$.

9 Re-Learning Gradient Descent [12 pts]

For a model $y = f_\theta(x)$ with two parameters $\theta = [\theta_1, \theta_2]^T$, we would like to use gradient descent to find the optimal parameters that minimize an objective function $\mathcal{R}(\theta, \mathbb{X}, \mathbb{Y})$ on training data $\mathbb{X}$, $\mathbb{Y}$. For a learning rate α, recall the gradient update step at time t:

$$\theta^{(t+1)} \quad = \quad \theta^{(t)} - \alpha \nabla_\theta \mathcal{R}(\theta^{(t)}, \mathbb{X}, \mathbb{Y})$$

Suppose you are provided the following **grid of gradients** for the objective function $\mathcal{R}$.
Each entry corresponds to the gradient $\nabla_\theta \mathcal{R}$ evaluated at the provided parameter values θ_1, θ_2.

θ_2

4	[-3,3]	[-1,1]	[0,1]	[0,1]	[3,3]
3	[-3,0]	[1,1]	[2,2]	[2,2]	[1,1]
2	[-1,0]	[-2,0]	[0,0]	[2,0]	[1,0]
1	[-2,0]	[0,-2]	[0,-2]	[2,-2]	[1,-1]
0	[-2,0]	[-3,-3]	[0,-3]	[0,-3]	[2,-2]
	0	1	2	3	4

θ_1

Example:
At the coordinates $(\theta_1, \theta_2) = (0, 0)$, the gradient is

$$\left[\frac{\partial \mathcal{R}}{\partial \theta_1}, \frac{\partial \mathcal{R}}{\partial \theta_2}\right]^T \Bigg|_{\theta_1=0,\theta_2=0} = [-2, 0]^T .$$

We define $\tilde{\theta} = [2, 2]^T$ as a local minimum because its gradient $\nabla_\theta \mathcal{R} = [0, 0]^T$. Help your friends Akon, Belcalis, and Carmen reach this local minimum $\tilde{\theta}$ through gradient descent with different choices for the learning rate α.

(a) [4 Pts] Akon believes that in order to converge to $\tilde{\theta}$, his learning rate should always be large. Suppose Akon chooses a learning rate of $\alpha = 1$ and initially starts at $\theta^{(0)} = [0, 0]^T$.

(i) Write $\theta^{(i)}$ for steps 1-3 that Akon's gradient descent algorithm takes on the (θ_1, θ_2) plane below. The $\theta^{(0)}$ step has been filled in for you.

θ_2

4					
3					
2					
1					
0	$\theta^{(0)}$				
	0	1	2	3	4

θ_1

Solution:

Starting at $\theta^{(0)} = [0, 0]$:

$\theta^{(1)} = [0, 0] - 1 \cdot [-2, 0] = [2, 0]$

$\theta^{(2)} = [2, 0] - 1 \cdot [0, -3] = [2, 3]$

$\theta^{(3)} = [2, 3] - 1 \cdot [2, 2] = [0, 1]$

Done plotting

$\theta^{(4)} = [0,1] - 1 \cdot [-2,0] = [2,1]$

$\theta^{(5)} = [2,1] - 1 \cdot [0,-2] = [2,3] = \theta_{(2)}$

We see that a cycle of length 3 is created where

$\theta^{(2)} \to \theta^{(3)} \to \theta^{(4)} \to \theta^{(2)}$

θ_2

4					
3			$\theta^{(2)}$		
2					
1	$\theta^{(3)}$				
0	$\theta^{(0)}$		$\theta^{(1)}$		
	0	1	2	3	4

θ_1

(ii) Which of the following is true about Akon's algorithm?

○ A. It reaches the local minimum $\tilde{\theta}$ within 10 steps.

○ B. It does not reach the local minimum $\tilde{\theta}$ in 10 steps, but can eventually reach $\tilde{\theta}$ given more time.

○ C. It will never converge on the local minimum $\tilde{\theta}$ because it forever alternates between two values.

○ **D. It will never converge on the local minimum $\tilde{\theta}$ because it gets stuck in a cycle of more than two values.**

(b) [4 Pts] Belcalis tries adjusting her learning rate as she updates the gradient. Suppose Belcalis also starts at $\theta^{(0)} = [0,0]^T$ with a learning rate of $\alpha = 1$ but decreases the learning rate to $\alpha = 0.5$ *after* the second update (i.e., use $\alpha = 1$ for $\theta^{(0)} \to \theta^{(1)} \to \theta^{(2)}$, then use $\alpha = 0.5$ for $\theta^{(2)} \to \theta^{(3)} \to \ldots$).

(i) Write $\theta^{(i)}$ for steps 1-3 that Belcalis's gradient descent algorithm takes on the (θ_1, θ_2) plane below. The $\theta^{(0)}$ step has been filled in for you.

θ_2

4					
3					
2					
1					
0	$\theta^{(0)}$				
	0	1	2	3	4

θ_1

Solution:

Starting at $\theta^{(0)} = [0, 0]$:

$\theta^{(1)} = [0, 0] - 1 \cdot [-2, 0] = [2, 0]$

$\theta^{(2)} = [2, 0] - 1 \cdot [0, -3] = [2, 3]$

$\theta^{(3)} = [2, 3] - 0.5 \cdot [2, 2] = [1, 2]$

Done plotting

$\theta^{(4)} = [1, 2] - 0.5 \cdot [-2, 0] = [2, 2]$

The minimum is reached at t = 4

	4					
	3			$\theta^{(2)}$		
	2		$\theta^{(3)}$			
θ_2	1					
	0	$\theta^{(0)}$		$\theta^{(1)}$		
		0	1	2	3	4

θ_1

(ii) Which of the following is true about Belcalis's algorithm?

○ **A. It reaches the local minimum $\tilde{\theta}$ within 10 steps.**

○ B. It does not reach the local minimum $\tilde{\theta}$ in 10 steps, but can eventually reach $\tilde{\theta}$ given more time.

○ C. It will never converge on the local minimum $\tilde{\theta}$ because it forever alternates between two values.

○ D. It will never converge on the local minimum $\tilde{\theta}$ because it gets stuck in a cycle of more than two values.

(c) [4 Pts] Carmen tries something unprecedented for their algorithm and creates a learning rate *schedule*, which adjusts the learning rate based on the time step t. Carmen's gradient update rule thus has a learning rate $\alpha^{(t)}$ as follows:

$$\theta^{(t+1)} = \theta^{(t)} - \alpha^{(t)} \nabla_\theta \mathcal{R}(\theta^{(t)}, \mathbb{X}, \mathbb{Y})$$

Suppose that Carmen starts start at $\theta^{(0)} = [0, 0]^T$ and sets the learning rate schedule as a zero-indexed Python list, where the i^{th} element is the learning rate $\alpha^{(i)}$ for the i^{th} gradient update. Which of the following schedules *most quickly* converges to the local minimum $\tilde{\theta}$?

○ A. $[1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$

○ **B.** $[\frac{1}{2}, \frac{2}{3}, \frac{1}{2}, 1, \frac{1}{2}, 1, \frac{1}{2}, \frac{2}{3}, \frac{1}{2}, 1]$

○ C. $[1, 1, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}, \frac{1}{2}]$

○ D. $[\frac{1}{2}, \frac{1}{3}, 1, 1, 1, 1, 1, 1, \frac{1}{3}, \frac{1}{2}]$

Solution:

We can reuse our results from part a and b to tell us that A does not converge and C converges at t = 4.

Working through B Starting at $\theta_{(0)} = [0, 0]$:

$\theta_{(1)} = [0, 0] - \frac{1}{2} \cdot [-2, 0] = [1, 0]$

$\theta_{(2)} = [1, 0] - \frac{2}{3} \cdot [-3, -3] = [3, 2]$

$\theta_{(3)} = [3, 2] - \frac{1}{2} \cdot [2, 0] = [2, 2]$

converges at t=3, 1 step faster than B

Working through D

$\theta_{(1)} = [0, 0] - \frac{1}{2} \cdot [-2, 0] = [1, 0]$

$\theta_{(2)} = [1, 0] - \frac{1}{3} \cdot [-3, -3] = [2, 1]$

$\theta_{(3)} = [2, 1] - 1 \cdot [0, -2] = [2, 3]$

$\theta_{(4)} = [2, 3] - 1 \cdot [0, -2] = [2, 3]$

$\theta_{(5)} = [2, 3] - 1 \cdot [2, 2] = [0, 1]$

$\theta_{(6)} = [0, 1] - 1 \cdot [-2, 0] = [2, 1]$

$\theta_{(7)} = [2, 1] - 1 \cdot [0, -2] = [2, 3]$

it should clear by t = 7 that D is not the correct choice. It is slower than B and C.

10 Risky Biases [16 pts]

We will analyze the model bias and variance of a simple linear regression model with no intercept term, $f_\theta(x) = \theta x$. Note that x and θ are both 1-dimensional.

(a) [4 Pts] Suppose we estimate the values on the right using our model $f_{\hat{\theta}}(x)$ on the test data point $(x, Y) = (2, 3)$.

Calculate the model risk, assuming that there is no observational variance (i.e. $\epsilon = 0$). Show your work. (*Hint:* Calculate the individual quantities that make up the model risk.)

Quantity	Value
$\mathbb{E}[f_{\hat{\theta}}(x)]$	1
$\mathbb{E}[(f_{\hat{\theta}}(x))^2]$	6
σ^2	0

Model Risk: ____________________

Solution: Since the variance of a random variable X can be expressed as the following:

$$\text{Var}[X] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$$

We can calculate the model variance $\text{Var}[f_{\hat{\theta}}(x)] = \mathbb{E}[f_{\hat{\theta}}(x)^2] - \mathbb{E}[f_{\hat{\theta}}(x)]^2 = 6 - 1^2 = 5$.

The squared model bias is $(g(x) - \mathbb{E}[f_{\hat{\theta}}(x)])^2 = (3-1)^2 = 4$ where $g(x) = Y$.

The model risk is the sum of the squared model bias and model variance (in the absence of observational variance: $5 + 4 = 9$.

(b) [4 Pts] Using a simple linear regression model without an intercept (i.e. $f_\theta(x) = \theta x$), we estimate the model variance on a *new* datapoint $(x, Y) = (3, 12)$ as $\text{Var}(f_{\tilde{\theta}}(x)) = 3$ using some arbitrary model parameter $\tilde{\theta}$.

Suppose we create a new model where $\theta_{\text{new}} = 2\tilde{\theta}$. Calculate the new model variance, $\text{Var}(f_{\theta_{\text{new}}}(x))$ using θ_{new}. Show your work.

Model Variance, $\text{Var}(f_{\theta_{\text{new}}}(x))$: ___________________

Solution: Since $\text{Var}[kX] = k^2\text{Var}[X]$:

$$\begin{aligned}\text{Var}[f_{\hat{\theta}}(x)] &= \text{Var}[\hat{\theta}x] \\ &= \text{Var}[2\tilde{\theta}x] \\ &= 4\text{Var}[\tilde{\theta}x] \\ &= 4\text{Var}[f_{\tilde{\theta}}(x)] \\ &= 4(3) = 12\end{aligned}$$

(c) [2 Pts] Based on the models described in part (b), which of the following **may** be true? Note that the bias of the new model is $g(x) - \mathbb{E}[f_{\theta_{\text{new}}}(x)]$, and the bias of the original model is $g(x) - \mathbb{E}[f_{\tilde{\theta}}(x)]$.

- □ **A. The new model bias is smaller than the original model bias.**
- □ **B. The new model bias is larger than the original model bias.**
- □ C. The new model bias is the same as the original model bias.
- □ D. The new model bias becomes undefined, but the original model bias is a real number.

We will next explore how L_2 regularization impacts model bias.

(d) [4 Pts] Derive the optimal parameter $\hat{\theta}$ for simple linear regression without an intercept term $f_\theta(x) = \theta x$ with L_2 regularization and a dataset with (x_i, y_i) for $i = 1, 2, ..., n$. The corresponding objective function is shown below.

$$\arg\min_{\theta} \frac{1}{n}\sum_{i=1}^{n}(y_i - f_\theta(x_i))^2 + \lambda\theta^2$$

Optimal $\hat{\theta}$: ____________________

Solution: We can derive this with calculus or using the optimal Ridge regression analytical solution where $\mathbb{X} = \vec{x}$.

$$(X^T X + \lambda n I)^{-1} X^T y = (\vec{x}^T \vec{x} + \lambda n)^{-1} \vec{x}^T \vec{y}$$
$$= \frac{\vec{x}^T \vec{y}}{\vec{x}^T \vec{x} + \lambda n} = \frac{\sum_i x_i y_i}{\sum_i x_i^2 + \lambda n}$$

(e) [2 Pts] Which of the following is true of the model bias for the model in part (d) with L_2 regularization? Note that this subpart is completely independent of part (d)!

□ A. The model bias approaches ∞ as $\lambda \to \infty$.

□ **B. The model bias approaches $g(x)$ as $\lambda \to \infty$.**

□ C. The model bias is typically zero if $\lambda = 0$.

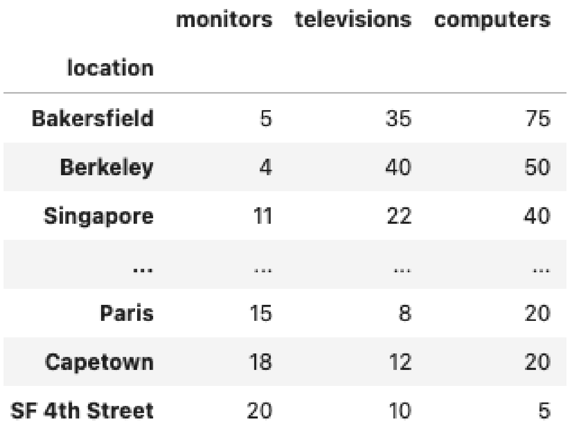
□ D. The model bias is typically undefined if $\lambda = 0$.

□ **E. The model bias is typically minimized if $\lambda = 0$.**

11 Principal Component Appliances [8 pts]

The big box electronics store, Good Buy, needs your help in applying Principal Components Analysis to their appliance sales data. You are provided records of monthly appliances sales (in thousands of units) for 100 different store locations worldwide. A few rows of the data are shown to the right.

Suppose you perform PCA as follows. First, you standardize the 3 numeric features above (i.e., transform to zero mean and unit variance). Then, you store these standardized features into X and use singular value decomposition to compute $X = U\Sigma V^T$.



	monitors	televisions	computers
location			
Bakersfield	5	35	75
Berkeley	4	40	50
Singapore	11	22	40
...	...	...	...
Paris	15	8	20
Capetown	18	12	20
SF 4th Street	20	10	5

(a) [1 Pt] What is the dimension of U?

○ A. 3×100 [✓ ○ **B.** 100×3] ○ C. 3×3 ○ D. 6×3

(b) [2 Pts] Given the definitions of U, Σ, and V above, which of the below expressions computes the principal components of X? Select all that apply.

[✓ □ **A.** $U\Sigma$] □ C. $U\Sigma V^T$ [✓ □ **E.** $X(V^T)^{-1}$]

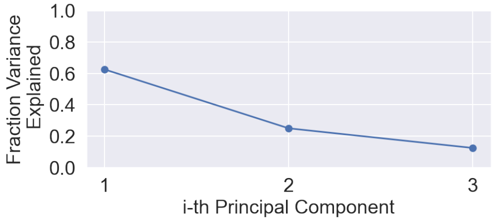
□ B. UV^T [✓ □ **D.** XV] □ F. XV^T

Solution: The principal components of X are defined as $U\Sigma$. Then right-multiplying the equation $U\Sigma V^T = X$ by $(V^T)^{-1}$ produces $U\Sigma = X(V^T)^{-1} = XV$, where the second equality is by orthonormality of the columns of V. Since the columns of V are an orthonormal set, by definition $V^TV = I$; in other words, $(V^T)^{-1} = V$.

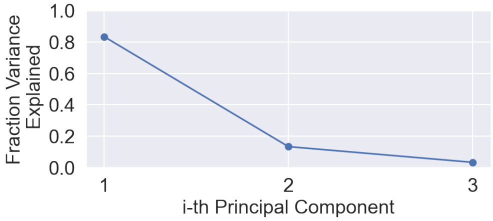
(c) [2 Pts] Suppose that your matrix Σ is given to the right. Which of the following is a possible scree plot for the data? Fill in the bubble corresponding to the most plausible plot.

$$\Sigma = \begin{bmatrix} 5 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

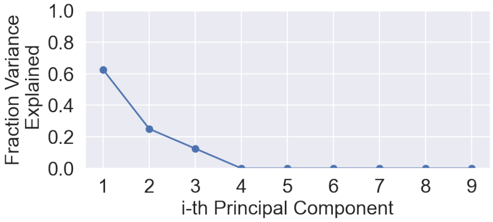
○ A.



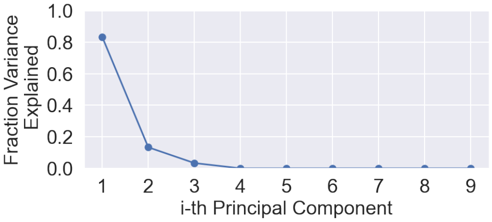
○ B.



○ C.



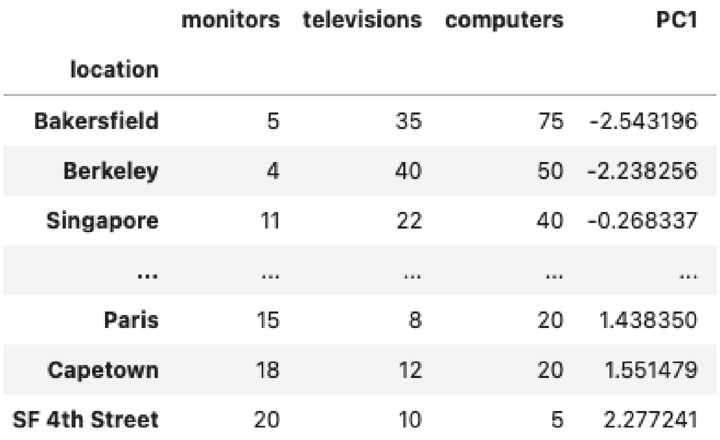
○ D.



Solution: The answer is B. The total variance of the data is the sum of the singular values, squared: $5^2 + 2^2 + 1^2 = 30$. Then, the contribution of the $i - th$ principal component with associated singular value s_i is $s_i^2 = 30$, so $25/30$, $4/30$, and $1/30$, respectively.

Incorrect choices: A plots the unsquared singular values, which do not correspond to variance. C and D assume 9 principal components, but X is max rank 3.

(d) [3 Pts] Suppose you focus on interpreting the first principal component, PC1. Below is the original data, now with PC1 values, as well as the first row of V^T (called V_1^T):



	monitors	televisions	computers	PC1
location				
Bakersfield	5	35	75	-2.543196
Berkeley	4	40	50	-2.238256
Singapore	11	22	40	-0.268337
...	...	...	...	...
Paris	15	8	20	1.438350
Capetown	18	12	20	1.551479
SF 4th Street	20	10	5	2.277241

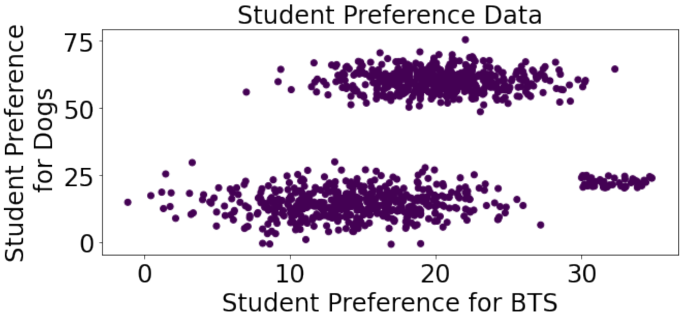
$$V_1^T = [0.59001398, -0.58165848, -0.55996153]$$

Interpret PC1's relationship with each feature x_i below. Positively related means that as $PC1$ increases, x_i increases; negatively related means that as $PC1$ increases, x_i decreases.

	Positively related	Negatively related	Not related
(i) Monitor sales	[✓ ○]	○	○
(ii) Television sales	○	[✓ ○]	○
(iii) Computer sales	○	[✓ ○]	○

12 Supervised or Unsupervised? [12 pts]

Tracy is studying an unlabeled dataset with two features x_1, x_2, which represent students' preferences for BTS and dogs, respectively, each on a scale from 0 to 100. The dataset is plotted in the visualization to the right:



(a) [2 Pts] Tracy would like to experiment with supervised and unsupervised learning methods. Which of the following is a supervised learning method? Select all that apply.

□ **A. Logistic regression**

□ **B. Linear regression**

□ **C. Decision tree**

□ D. Agglomerative clustering

□ E. K-Means clustering

(b) [2 Pts] Suppose Tracy decides to use K-Means clustering on the features x_1 and x_2. Which of the following choices for the number of clusters k is *most* appropriate for this task?

○ A. 1

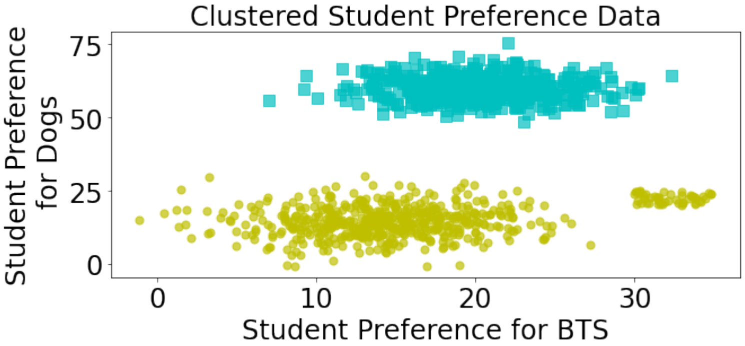
○ B. 2

○ **C. 3**

○ D. 4

○ E. None of the above

(c) [4 Pts] Regardless of your previous answer, suppose Tracy uses K-Means clustering with $k = 2$ and obtains the clustering below:



(This question part continues on the next page.)

(c, continued) Using these clusters, Tracy then generates labels y_i (0 or 1) for the i-th training datapoint $X_i = (x_{1i}, x_{2i})$. Tracy uses a logistic regression model to fit the two features (x_1, x_2) to the K-Means clustering labels y. Can we achieve 100% training accuracy with this modeling approach? Justify your answer.

Solution: Yes - the points are linearly separable.

(d) [4 Pts] Instead of using $k = 2$ as we did in part (c), suppose we used $k = 4$ instead. Using the output labels from this new K-Means clustering and with the same features as part (c), Tracy decides to train a decision tree without any pruning, maximum depth or any training restrictions.

Can we achieve 100% training accuracy with this modeling approach? Justify your answer.

Solution: Yes - there will never be any impure nodes since given a particular (x_{1i}, x_{2i}), there can only be one possible K-Means label y_i. We will be able to get 100% classification accuracy using a decision tree!

13 Guest Lectures [6 pts]

This course had two guest lectures in the last week of the semester.

The first lecture, "Privacy and Ethics Regulations," was presented by Professor Amol Deshpande from the University of Maryland. It discussed the current privacy laws that both define individual rights to data and regulate how businesses manage and store user data.

Based on your knowledge of the above lecture, please mark each statement as True or False.

(a) [1 Pt] The European General Data Protection Regulation (GDPR) set the standard for privacy regulations and is reputably the most comprehensive law to date.

○ **A. True** ○ B. False

(b) [1 Pt] Current privacy regulations like the GDPR also specify the technological implementations and/or algorithmic solutions to individual rights to data, e.g., data portability or data deletion.

○ A. True ○ **B. False**

(c) [1 Pt] The U.S. currently has a set of federal (i.e., nationwide) privacy regulations that supersede individual state laws such as the California Consumer Protection Act.

○ A. True ○ **B. False**

The second lecture, "Big Data Analytics with Apache Spark," was presented by Professor Matei Zaharia from Stanford University. It discussed MapReduce and Spark—two distributed, fault-tolerant programming models to handle big data stored across clusters of computing servers. The speaker also demonstrated how to manipulate data using the Apache Spark API (Application Programming Interface).

Based on your knowledge of the above lecture, please mark each statement as True or False.

(d) [1 Pt] Programming models like MapReduce and Apache Spark let data scientists specify high-level operations on data distributed across servers. The programming models then automatically optimize and parallelize computation and storage across the individual servers.

○ **A. True** ○ B. False

(e) [1 Pt] MapReduce is a distributed programming framework developed after Apache Spark that generalizes Spark's paradigms to support more data applications.

○ A. True ○ **B. False**

(f) [1 Pt] In lecture, Professor Zaharia showed lecture slides with Apache Spark code snippets that looked like Pandas code (e.g., DataFrame manipulation).

○ **A. True** ○ B. False

14 Following Instructions [1 Pt]

If you are taking your exam in-person, you can earn this point by writing your Student ID in the top right corner of each page.

If you are taking your exam online, you can earn this point by doing both of the following:

1. Write the answer to each question on a different page.
2. Assign all pages to the appropriate question subparts when submitting on Gradescope.

Reminder: Complete the Honor Code question (Q0) on the front of the exam for another point.

[Optional] If you could flip a coin, where a heads means you automatically get a 100% on this final, but a tails means you automatically get a 0% on this final, would you do it?

○ Yes

○ No

[Optional] Who's your favorite person on Data 100 Staff? Draw a picture of them!

Congratulations on finishing the final!

Spring 2022 Data C100/C200 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

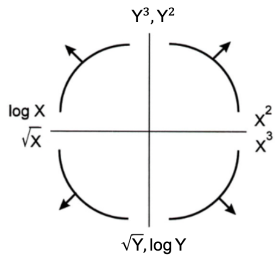
Function	Description
`s.str.len()`	Returns a Series containing length of each string
`s.str.lower()`/`s.str.upper()`	Returns a Series containing lowercase/uppercase version of each string
`s.str.replace(pat, repl)`	Returns a Series after replacing occurences of substrings matching regular expression `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 $\hat{a}$	$\hat{a} = \bar{y} - \hat{b}\bar{x}$

Spring 2022 Data C100/C200 Midterm 2 Reference Sheet

Ordinary Least Squares

Multiple Linear Regression Model: $\hat{\mathbb{Y}} = \mathbb{X}\theta$ with design matrix $\mathbb{X}$, response vector $\mathbb{Y}$, and predicted vector $\hat{\mathbb{Y}}$. If there are p features plus a bias/intercept, then the vector of parameters $\theta = [\theta_0, \theta_1, \ldots, \theta_p]^T \in \mathbb{R}^{p+1}$. The vector of estimates $\hat{\theta}$ is obtained from fitting the model to the sample $(\mathbb{X}, \mathbb{Y})$.

Concept	Formula	Concept	Formula
Mean squared error	$R(\theta) = \frac{1}{n}\lvert\lvert\mathbb{Y} - \mathbb{X}\theta\rvert\rvert_2^2$	Normal equation	$\mathbb{X}^T\mathbb{X}\hat{\theta} = \mathbb{X}^T\mathbb{Y}$
Least squares estimate, if $\mathbb{X}$ is full rank	$\hat{\theta} = (\mathbb{X}^T\mathbb{X})^{-1}\mathbb{X}^T\mathbb{Y}$	Residual vector, e	$e = \mathbb{Y} - \hat{\mathbb{Y}}$
		Multiple R^2 (coefficient of determination)	$R^2 = \frac{\text{variance of fitted values}}{\text{variance of } y}$
Ridge Regression L2 Regularization	$\frac{1}{n}\lvert\lvert\mathbb{Y} - \mathbb{X}\theta\rvert\rvert_2^2 + \alpha\lvert\lvert\theta\rvert\rvert_2^2$	Squared L2 Norm of $\theta \in \mathbb{R}^d$	$\lvert\lvert\theta\rvert\rvert_2^2 = \sum_{j=1}^d \theta_j^2$
Ridge regression estimate (closed form)	$\hat{\theta}_{\text{ridge}} = (\mathbb{X}^T\mathbb{X} + n\alpha I)^{-1}\mathbb{X}^T\mathbb{Y}$		
LASSO Regression L1 Regularization	$\frac{1}{n}\lvert\lvert\mathbb{Y} - \mathbb{X}\theta\rvert\rvert_2^2 + \alpha\lvert\lvert\theta\rvert\rvert_1$	L1 Norm of $\theta \in \mathbb{R}^d$	$\lvert\lvert\theta\rvert\rvert_1 = \sum_{j=1}^d \lvert\theta_j\rvert$

Scikit-Learn

Suppose `sklearn.model_selection` and `sklearn.linear_model` are both imported packages.

Package	Function(s)	Description
`sklearn.linear_model`	`LinearRegression(fit_intercept=True)`	Returns an ordinary least squares Linear Regression model.
	`LassoCV(fit_intercept=True)`, `RidgeCV(fit_intercept=True)`	Returns a Lasso (L1 Regularization) or Ridge (L2 regularization) linear model, respectively, and picks the best model by cross validation.
	`model.fit(X, y)`	Fits the scikit-learn `model` to the provided `X` and `y`.
	`model.predict(X)`	Returns predictions for the X passed in according to the fitted `model`.
	`model.coef_`	Estimated coefficients for the linear model, not including the intercept term.
	`model.intercept_`	Bias/intercept term of the linear model. Set to 0.0 if `fit_intercept=False`.
`sklearn.model_selection`	`train_test_split(*arrays, test_size=0.2)`	Returns two random subsets of each array passed in, with 0.8 of the array in the first subset and 0.2 in the second subset.

Probability

Let X have a discrete probability distribution $P(X = x)$. X has expectation $\mathbb{E}[X] = \sum_x xP(X = x)$ over all possible values x, variance $\mathrm{Var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2]$, and standard deviation $\mathrm{SD}(X) = \sqrt{\mathrm{Var}(X)}$.

The covariance of two random variables X and Y is $\mathbb{E}[(X - \mathbb{E}[X])(Y - \mathbb{E}[Y])]$. If X and Y are independent, then $\mathrm{Cov}(X, Y) = 0$.

Notes	Property of Expectation	Property of Variance
X is a random variable.		$\mathrm{Var}(X) = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$
X is a random variable. $a, b \in \mathbb{R}$ are scalars.	$\mathbb{E}[aX + b] = a\mathbb{E}[X] + b$	$\mathrm{Var}(aX + b) = a^2\mathrm{Var}$
X, Y are random variables.	$\mathbb{E}[X + Y] = \mathbb{E}[X] + \mathbb{E}[Y]$	$\mathrm{Var}(X + Y) = \mathrm{Var}(X) + \mathrm{Var}(Y) + 2\mathrm{Cov}(X, Y)$
X is a Bernoulli random variable that takes on value 1 with probability p and 0 otherwise.	$\mathbb{E}[X] = p$	$\mathrm{Var}(X) = p(1 - p)$
Y is a Binomial random variable representing the number of ones in n independent Bernoulli trials with probability p of 1.	$E[Y] = np$	$\mathrm{Var}(Y) = np(1 - p)$

Central Limit Theorem

Let $(X_1, \ldots, X_n)$ be a sample of independent and identically distributed random variables drawn from a population with mean μ and standard deviation σ. The sample mean $\overline{X}_n = \sum_{i=1}^{n} X_i$ is normally distributed, where $\mathbb{E}[\overline{X}_n] = \mu$ and $\mathrm{SD}(\overline{X}_n) = \sigma/\sqrt{n}$.

Parameter Estimation

Suppose for each individual with fixed input x, we observe a random response $Y = g(x) + \epsilon$, where g is the true relationship and ϵ is random noise with zero mean and variance σ^2.

For a new individual with fixed input x, define our random prediction $\hat{Y}(x)$ based on a model fit to our observed sample $(\mathbb{X}, \mathbb{Y})$. The model risk is the mean squared prediction error between Y and $\hat{Y}(x)$:

$$\mathbb{E}[(Y - \hat{Y}(x))^2] = \sigma^2 + \left(\mathbb{E}[\hat{Y}(x)] - g(x)\right)^2 + \mathrm{Var}(\hat{Y}(x)).$$

Suppose that input x has p features and the true relationship g is linear with parameter $\theta \in \mathbb{R}^{p+1}$. Then $Y = f_\theta(x) = \theta_0 + \sum_{j=1}^{p} \theta_j x_j + \epsilon$ and $\hat{Y} = f_{\hat{\theta}}(x)$ for an estimate $\hat{\theta}$ fit to the observed sample $(\mathbb{X}, \mathbb{Y})$.

Gradient Descent

Let $L(\theta, \mathbb{X}, \mathbb{Y})$ be an objective function to minimize over θ, with some optimal $\hat{\theta}$. Suppose $\theta^{(0)}$ is some starting estimate at $t = 0$, and $\theta^{(t)}$ is the estimate at step t. Then for a learning rate α, the gradient update step to compute $\theta^{(t+1)}$ is

$$\theta^{(t+1)} = \theta^{(t)} - \alpha \nabla_\theta L(\theta^{(t)}, \mathbb{X}, \mathbb{Y}),$$

where $\nabla_\theta L(\theta^{(t)}, \mathbb{X}, \mathbb{Y})$ is the partial derivative/gradient of L with respect to θ, evaluated at $\theta^{(t)}$.

SQL

SQLite syntax:

```
SELECT [DISTINCT]
    {* | expr [[AS] c_alias]
    {,expr [[AS] c_alias] ...}}
FROM tableref {, tableref}
[[INNER | LEFT ] JOIN table_name
    ON qualification_list]
[WHERE search_condition]
[GROUP BY colname {,colname...}]
[HAVING search_condition]
[ORDER BY column_list]
[LIMIT number]
[OFFSET number of rows];
```

Syntax	Description
`SELECT column_expression_list`	List is comma-separated. Column expressions may include aggregation functions (`MAX`, `FIRST`, `COUNT`, etc). `AS` renames columns. `DISTINCT` selects only unique rows.
`FROM s INNER JOIN t ON cond`	Inner join tables `s` and `t` using `cond` to filter rows; the `INNER` keyword is optional.
`FROM s LEFT JOIN t ON cond`	Left outer join of tables `s` and `t` using `cond` to filter rows.
`FROM s, t`	Cross join of tables `s` and `t`: all pairs of a row from `s` and a row from `t`
`WHERE a IN cons_list`	Select rows for which the value in column `a` is among the values in a `cons_list`.
`ORDER BY RANDOM LIMIT n`	Draw a simple random sample of `n` rows.
`ORDER BY a, b DESC`	Order by column `a` (ascending by default) , then `b` (descending).
`CASE WHEN pred THEN cons ELSE alt END`	Evaluates to `cons` if `pred` is true and `alt` otherwise. Multiple `WHEN`/`THEN` pairs can be included, and `ELSE` is optional.
`WHERE s.a LIKE 'p'`	Matches each entry in the column `a` of table `s` to the text pattern `p`. The wildcard `%` matches at least zero characters.
`LIMIT number`	Keep only the first `number` rows in the return result.
`OFFSET number`	Skip the first `number` rows in the return result.

Spring 2022 Data C100/C200 Final Reference Sheet

Principal Component Analysis (PCA)

The i-th Principal Component of the matrix X is defined as the i-th column of $U\Sigma$ defined by Singular Value Decomposition (SVD).

$X = U\Sigma V^T$ is the SVD of X if U and V^T are orthonormal matrices and Σ is a diagonal matrix. The diagonal entries of Σ, $[s_1, \ldots, s_r, 0, \ldots, 0]$, are known as singular values of X, where $s_i > s_j$ for $i < j$ and $r = \text{rank}(X)$.

Define the design matrix $X \in \mathbb{R}^{n \times p}$. Define the total variance of X as the sum of individual variances of the p features. The amount of variance captured by the i-th principal component is equivalent to s_i^2/n, where n is the number of datapoints.

Logistic Regression and Classification

Logistic Regression Model: For input feature vector x, $\hat{P}_\theta(Y = 1|x) = \sigma(x^T\theta)$. The estimate $\hat{\theta}$ is the parameter θ that minimizes the average cross-entropy loss on training data. For a single datapoint, define cross-entropy loss as $-\left[y \log(p) + (1-y)\log(1-p)\right]$, where p is the probability that the response is 1.

Logistic Regression Classifier: For a given input x and trained logistic regression model with parameter θ, compute $p = \hat{P}(Y = 1|x) = \sigma(x^T\theta)$. predict response $\hat{y}$ with classification threshold T as follows:

$$\hat{y} = \text{classify}(x) = \begin{cases} 1 & p \geq T \\ 0 & \text{otherwise} \end{cases}$$

Confusion Matrix

Columns are the predicted values $\hat{y}$ and rows are the actual classes y.

	0	1
0	True negative (TN)	False Positive (FP)
1	False negative (FN)	True Positive (TP)

Classification Performance

Suppose you predict n datapoints.

Metric	Formula	Other Names	Visualization	Plot
Accuracy	$\frac{TP+TN}{n}$		Precision-Recall Curve	Precision vs. Recall for different thresholds T
Precision	$\frac{TP}{TP+FP}$		ROC Curve	TPR vs. FPR for different thresholds T
Recall/TPR	$\frac{TP}{TP+FN}$	True Positive Rate, Sensitivity		
FPR	$\frac{FP}{FP+TN}$	False Positive Rate, Specificity		

Scikit-Learn

Suppose `linear_model` is an imported `sklearn` package.

Class/Attribute	Description	Function	Description
`linear_model.LogisticRegression( fit_intercept=True, penalty='l2', C=1.0)`	Returns an ordinary least squares Linear Regression model. Hyperparameter C is inverse of regularization parameter, C = 1/λ.	`model.fit(X, y)`	Fits the scikit-learn `model` to the provided `X` and `y`.
`model.coef_`	Estimated coefficients for the model, not including the intercept term.	`model.predict_proba(X)`	Returns predicted probabilities for the `X` passed in according to the fitted `model`. If binary classes, will return probabilities for both class 0 and 1.
`model.intercept_`	Bias/intercept term of the model. Set to 0.0 if `fit_intercept=False`.	`model.predict(X)`	Returns predictions for the `X` passed in according to the fitted `model`.
		`model.score(X, y)`	Returns the average `model` accuracy on the given test data `X` and labels `y`.

Suppose `tree` and `ensemble` are imported `sklearn` packages.

Class/Function	Description
`tree.DecisionTreeClassifier(criterion='entropy', max_depth=None)`	Returns a decision tree model which uses `criterion` to measure the quality of a split. `max_depth` is the maximum depth of the tree; if `None`, then nodes are expanded until all leaves are pure.
`ensemble.RandomForestClassifier(n_estimators=100, criterion='entropy', max_depth=None)`	Fit `n_estimators` decision tree classifiers on sub-samples of the dataset.
`model.fit(X, y)`	Decision tree: Fit a decision tree `model` to the provided `X` and `y`. Random forest classifier: Build a forest `model` of decision trees fit to the provided `X` and `y`.
`model.predict(X)`	Decision tree: Returns predicted response for the `X` passed in according to the fitted `model`. Random forest classifier: Returns the predicted class by highest mean probability estimate according to the trees in the forest `model`.

Clustering

K-Means Clustering: Pick an arbitrary k, and randomly place k "centers", each a different color. Then repeat until convergence:

1. Color points according to the closest center (defined as squared distance).
2. Move center for each color to center of points with that color.

K-Means minimizes inertia, defined as the sum of squared distances from each datapoint to its center.

Agglomerative Clustering: Assign each datapoint to its own cluster. Then, recursively merge pairs of clusters together until there are k clusters remaining.

A datapoint's **silhouette score** S is defined as $S = (B - A)/\max(A, B)$, where A is the mean distance to other points in its cluster, and B is the mean distance to points in its closest cluster.

Decision Trees and Random Forests

Suppose you have a **decision tree classifier** for k classes. For each node, define the probability for class $C \in \{1, \ldots, k\}$ as $p_C = d_C/d$, where d_C is the number of datapoints in class C (of the d total in the node). Then the entropy of the node (in bits) is defined as $S = -\sum_C p_C \log_2 p_C$, and the weighted entropy of the node is its entropy scaled by the fraction of datapoints in that node.

Decision tree generation algorithm: All of the data starts in the root node. Repeat until every node is either pure or unsplittable:

- Pick the best feature x and best split value β, where β is picked to maximize the change in weighted entropy between the parent node and the child nodes.
- Split data into two nodes, one where x < β, and one where x ≥ β.

A node that has only one samples from one class is called a "pure" node. A node that has overlapping data points from different classes and thus that cannot be split is called "unsplittable".

A **random forest** is a collection of many decision trees fit to variations of the same training data (e.g., bootstrapped samples, also called bagging; or random subsets of features). It is an ensemble method.