DS-100 Midterm Exam A

Fall 2018

Name: _____

Email: _____@berkeley.edu

Student ID: _____

Instructions:

- This midterm exam 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 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**.
- When selecting your choices, you must **fully shade** in the box/circle. Check marks will likely be mis-graded.
- You may use a one-sheet (two-sided) study guide.

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: _____

Syntax Reference

Regular Expressions

"|" matches expression on either side of symbol. Has lowest priority.

"\" match the following character literally.

"?" match preceding literal or sub-expression 0 or 1 times.

"+" match preceding literal or sub-expression *one* or more times.

"*" match preceding literal or sub-expression *zero* or more times.

"." match any character except new line.

"[ ]" match any one of the characters inside, accepts a range, e.g., `"[a-c]"`. All characters inside treated literally.

"( )" used to create a sub-expression.

"{n}" preceding expression repeated n times.

Some useful Python functions and syntax

`re.findall(pattern, st)` returns the list of all sub-strings in `st` that match `pattern`.

`np.random.choice(a, replace, size)` Generates a random sample from `a` consisting of `size` values (with replacement if `replace=True`). `a` can be 1-D array-like or int.

Useful Pandas Syntax

```
df.loc[row_selection, col_list]  # row selection can be boolean
df.iloc[row_selection, col_list] # row selection can be boolean
pd.get_dummies(data) # Convert categorical variable into indicator values
df.groupby(group_columns)[['colA', 'colB']].agg(agg_func)
df.groupby(group_columns)[['colA', 'colB']].filter(filter_func)
```

Variance and Expected Value

The expected value of X is $\mathbb{E}[X] = \sum_{j=1}^{m} x_j p_j$. The variance of X is $Var[X] = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - \mathbb{E}[X]^2$. The standard deviation of X is $SD[X] = \sqrt{Var[X]}$.

Sampling

1. Below is shown the first 5 rows of the table `restaurants`:

name	cuisine	size
Marufuku	Japanese	2241
Jack in the Box	Fast Food	1592
Thai Basil	Thai	820
Tako Sushi	Japanese	1739
McDonald's	Fast Food	1039

The table `restaurants` contains information regarding different restaurants. The name and cuisine columns contain strings, and the size column contains integers. **The name column is the primary key of the table and therefore contains unique values.** *This is a preview of the first 5 rows of the table. You may assume it has many more rows than what is shown, with the same structure and no missing data.*

Throughout problem 1 and 2, use the keywords and numbers from the answer bank below to fill in the blanks. Note that the same keyword/number can be used multiple times; some keywords may not be used at all. The documentation for some terms appears on the first page of this exam. This answer bank also appears on the back of your answer sheet.

1	2	3	4	5	6
7	8	9	10	11	12
<	>	<=	>=	==	
`restaurants`	`min_rating_df`	**pivot**	**agg**	**loc**	index
iloc	**sort_values**	**size**	**filter**	**groupby**	
barplot	**lineplot**	**jointplot**	**boxplot**		
'name'	'cuisine'	'size'	'rating'		
max	**min**	**median**	**mean**	**first**	**last**
np.array	**sample**	**pd.Series**	**list**	values	
True	False	n	x		

(a) Complete the following function `sample`, which takes in a series and a sample size n and returns a **simple random sample** of n values in that series. Recall that a SRS is drawn without replacement. The result should be a **list** of the n values that are in the sample. For example, `sample(restaurants['name'], 10)` should return a simple random sample of 10 restaurant names with no duplicates. The documentation for `np.random.choice` can be found on the first page of this exam.

```
def sample(series, n):
    return _<i>_______(np.random.choice(_<ii>______._<iii>______,
    size=_<iv>___________, replace=_<v>_____________))
```

Solution:

```
def sample(series, n):
    return list(np.random.choice(series.values,
    size=n, replace=False))
```

(b) Suppose that the probability that Jack in the Box appears in the simple random sample is $\frac{1}{10}$. What is the probability that McDonald's appears in the sample? **There is only one correct answer.**

◯ A. $< \frac{1}{10}$
◯ **B.** $\frac{1}{10}$
◯ C. $> \frac{1}{10}$
◯ D. Not enough information

Solution: Each restaurant has the same chance of being chosen in a simple random sample, so the probability that McDonalds appears is also $\frac{1}{10}$.

(c) What type of sample does `restaurants.groupby('cuisine')['name'].first()` collect? **Select all that apply.**

☐ A. Simple Random Sample
☐ B. Stratified Random Sample
☐ C. Cluster Sample
☐ D. Probability Sample
☐ **E. None of the above**

Solution: This would collect the same sample every time, so this is not a probability sample.

(d) Josh wants to collect a stratified random sample of restaurant names where the strata are the cuisine type, and he wants to collect 2 restaurant names per strata. Complete the following line of code to collect Josh's desired stratified random sample.

```
restaurants._<i>_____(_<ii>_______)[_<iii>_______]._<iv>_______(
lambda x: _<v>__________(_<vi>__________, _<vii>__________))
```

Solution:

```
restaurants.groupby('cuisine')['name'].agg(
lambda x: sample(x, 2))
```

(e) Suppose there are 10 unique cuisine types with 15 fast food restaurants, 20 Japanese Restaurants, and 5 Thai Restaurants. There are 100 restaurants overall in the table with at least 2 restaurants for each cuisine type. What is the probability that McDonald's is picked in Josh's stratified sample from the previous problem?

Solution: We can find the probability that McDonald's does not appear in the stratified sample and subtract this from one. $1 - \frac{14}{15} * \frac{13}{14} = \frac{2}{15}$

(f) Fernando wants to collect a cluster sample, where each cluster is a cuisine type. Suppose Fernando wants to have 2 clusters in his cluster sample. Which of the following lines of code would create Fernando's desired cluster sample? **Select all that apply.** At least one answer is correct. All of the code in all three possible answers is syntactically correct.

☐ A.
```
restaurants[restaurants['cuisine'].isin(
    np.random.choice(restaurants['cuisine'].unique(),
    size=2, replace=True))]['name']
```

☐ **B.**
```
restaurants[restaurants['cuisine'].isin(
    np.random.choice(restaurants['cuisine'].unique(),
    size=2, replace=False))]['name']
```

☐ C.
```
restaurants[restaurants['cuisine'].isin(
    np.random.choice(restaurants['cuisine'].values,
    size=2, replace=False))]['name']
```

Solution:

☐ A. False. Having `replace=True` in `np.random.choice` makes it possible for two of the same cluster to be chosen.

☐ **B. True. By selecting the unique values in the cuisine column, we make sure that each cuisine has a equal chance of being chosen.**

☐ C. False. Having `restaurants['cuisine'].values` makes it possible for multiple cuisines to appear, which means that it is possible for two of the same cluster to be chosen.

(g) With the same assumptions as in part (e), what is the probability that McDonald's appears in Fernando's cluster sample?

Solution: We can find the probability that McDonald's does not appear in the cluster sample and subtract this from one. $1 - \frac{9}{10} * \frac{8}{9} = \frac{2}{10} = \frac{1}{5}$

(h) Manana goes for a third sampling strategy. She decides to take a cluster sample (just like Fernando) of 2 cuisines, but rather than collecting information about every restaurant in those two clusters, she then collects a SRS of one restaurant from each of her two randomly selected clusters, for a total of two restaurants. What type of sample has Manana collected? **Select all that apply.**

☐ A. Simple Random Sample

☐ B. Stratified Random Sample

☐ **C. Cluster Sample**

☐ **D. Probability sample**

☐ E. None of the above

Solution: As discussed in discussion 1, this is a multi-staged cluster sample. We are taking a cluster sample and then taking a SRS of each cluster we have chosen. Cluster sampling is a form of probability sampling.

(i) Let H be the probability that McDonald's appears in Manana's sample from part (h). Suppose Manana repeats this the process in part (h) 5 times with replacement between samples so that each (two restaurant) sample is independent from the other 4 samples. Let X be a random variable that represents how many total times McDonald's appears in these 5 samples. What is $\mathbb{E}[X]$? Your answer should be in terms of H.

Solution: Let X_i be a Bernoulli random variable that is 1 if McDonalds appears in the i^{th} sample. Then, $\mathbb{E}[X] = \mathbb{E}[X_1+X_2+X_3+X_4+X_5] = 5\mathbb{E}[X_1] = 5\mathbb{E}[X_1] = 5H$.

(j) What is $Var(X)$? Again, your answer should be in terms of H.

Solution: $Var(X) = Var(X_1 + X_2 + X_3 + X_4 + X_5) = 5Var(X_1) = 5H(1 - H)$.

Pandas

2. For the following problems, suppose we add a new column to our restaurants table which contains the average rating of each restaurant by users of a restaurant review service. **Remember to only use terms from the table in the previous section (or on the back of your answer sheet)!**

name	cuisine	size	rating
Marufuku	Japanese	2241	4.5
Jack in the box	Fast Food	1592	3.4
Thai Basil	Thai	820	4.7
Tako Sushi	Japanese	1739	2.3
McDonald's	Fast Food	1039	3.5

(a) Let a "lowest rank restaurant" be a restaurant that has the lowest rating for a given cuisine. Create a table of all lowest rank restaurants (i.e. one row for each cuisine). Include the restaurant's name, size, and average rating.

```
min_rating_df = restaurants._<i>_____(_<ii>_____)
                            ._<iii>_____(_<iv>_____)
                            ._<v>_____()
```

Solution:

```
min_rating_df = restaurants.sort_values('rating')
                .groupby('cuisine')
                .first()
```

(b) Create a chart that is useful for visualizing the ratings for the lowest rank restaurants.

```
sns._<i>_____(min_rating_df.index, _<ii>_____[_<iii>_____])
```

Solution:

```
sns.barplot(min_rating_df.index, min_rating_df['rating'])
```

(c) Change the ratings of all Japanese restaurants to be 5.

```
restaurants.loc[_<i>_____[_<ii>_____]_<iii>_____ 'Japanese',
    _<iv>_____] = _<v>_____
```

Solution:

```
restaurants.loc[restaurants['cuisine']=='Japanese', 'rating'] = 5
```

(d) Return a list of all cuisines whose restaurants have an average size greater than or equal to 1000.

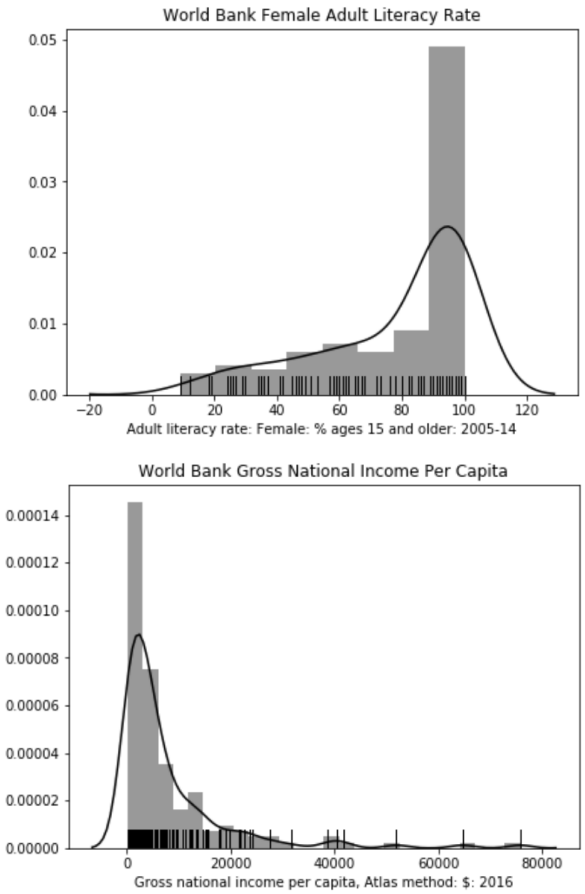
```
list(restaurants._<i>___________(_<ii>___________)
            ._<iii>_____(
            lambda x:x[_<iv>________]._<v>________() >= 1000)
            [_<vi>___________].unique())
```

Solution:

```
restaurants.groupby('cuisine').filter(lambda x: x['size']
.mean() >= 1000)['cuisine'].unique()
```

Visualization

3. Suppose we have the following histograms.



(a) Which of the histograms above is right skewed? **There is only one correct answer.**

◯ A. World Bank Female Adult Literacy Rate

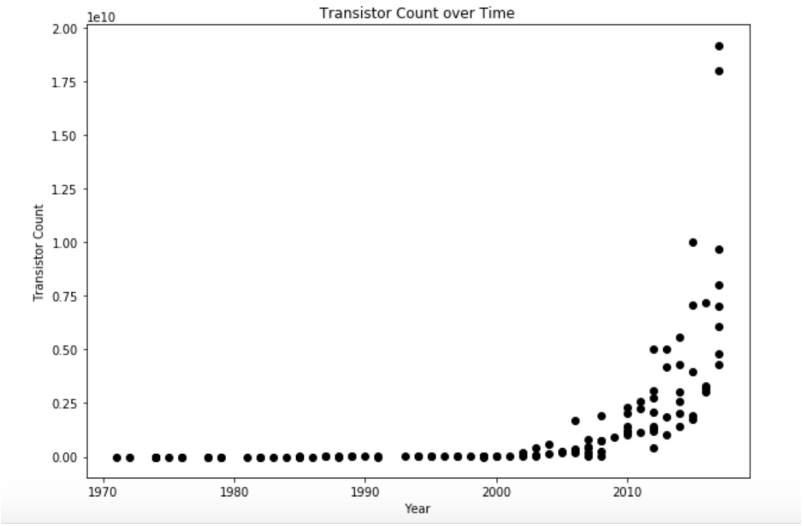
◯ **B. World Bank Gross National Income Per Capita**

(b) Which histogram would look more symmetric with a log transformation applied? **There is only one correct answer.**

◯ A. World Bank Female Adult Literacy Rate

◯ **B. World Bank Gross National Income Per Capita**

4. Consider the scatter plot shown below.



Which of the following transformations would make the relationship between x and y more linear, i.e. if we plotted $f_y(y)$ vs. $f_x(x)$, which would look most linear? **There is only one correct answer.**

- ◯ A. $f_x(x) = x^2$ $f_y(y) = y^2$
- ◯ B. $f_x(x) = log(x)$ $f_y(y) = y$
- ◯ **C.** $f_x(x) = x$ $f_y(y) = log(y)$
- ◯ D. $f_x(x) = log(x)$ $f_y(y) = y^2$

5. For each of the following relationships between x and y, select the appropriate transformation so that the transformed values are linearly related. In other words, select the transformations such that if we plotted $f_y(y)$ vs. $f_x(x)$, we'd expect to get a straight line. **There is only one correct answer in each part.**

(a) $y = ab^x$

- ◯ A. $f_y(y) = log(y)$ $f_x(x) = log(x)$
- ◯ B. $f_y(y) = y$ $f_x(x) = log(x)$
- ◯ **C.** $f_y(y) = log(y)$ $f_x(x) = x$
- ◯ D. $f_y(y) = \frac{1}{y}$ $f_x(x) = \frac{1}{x}$
- ◯ E. The relationship is already linear.

(b) $y = \frac{x}{a+bx}$

- ◯ A. $f_y(y) = \frac{1}{y}$ $f_x(x) = x$
- ◯ B. $f_y(y) = y$ $f_x(x) = \frac{1}{x}$
- ◯ **C.** $f_y(y) = \frac{1}{y}$ $f_x(x) = \frac{1}{x}$
- ◯ D. None of the transformations above create a linear relationship.
- ◯ E. The relationship is already linear.

Regular Expressions

6. Recall that the `re.findall(regex, string)` method returns a list of all matching strings, e.g. `re.findall('cow', 'A cow = cow.')` would return `['cow', 'cow']`.

   For the following regular expression, which of the following strings would result in exactly one match? By one match, we mean the list returned by `findall` is of length 1.

   `'[a-z][aueoi][a-z]+'`

   - ☐ A. `'ba'`
   - ☐ **B. `'bat'`**
   - ☐ **C. `'batch'`**
   - ☐ **D. `'batches'`**
   - ☐ E. `'batches␣of␣bees'`
   - ☐ F. None of the above

7. For the following regular expression, which of the following strings would result in exactly one match?

   `'(burrito|dog){2}'`

   - ☐ **A. `'dogdog'`**
   - ☐ **B. `'burritodog'`**
   - ☐ **C. `'dogburrito'`**
   - ☐ D. `'burrito␣dog'`
   - ☐ E. None of the above

8. How many matches would be returned by the code below (note the **square** brackets!): `re.findall("[Cow|Man]{2}", "CowManCowCowManMan999")`?

   ◯ A. 0 ◯ B. 1 ◯ C. 2 ◯ D. 3 ◯ E. 4 ◯ F. 6 ◯ G. 8 √ **H. 9**
   ◯ I. 12 ◯ J. 17 ◯ K. 18

9. What are the start and end positions for the match returned by `re.search` below? Use Python's 0-indexing and semi-open intervals [a, b) notation. If you believe that no match is returned, answer with the empty interval [0, 0).

   `re.search(r'\..*', 'pic.jpg.*bak')`

   (a) The inclusive start position is: ◯ A. 0 √ **B. 3** ◯ C. 4 ◯ D. 7 ◯ E. 8

   (b) The exclusive end position is: ◯ A. 0 ◯ B. 3 ◯ C. 7 ◯ D. 8 √ **E. 11**

Linear Models

10. Recall from lecture that a linear model is defined as a model where our prediction $\hat{y}$ is given by the equation below, where d is the number of parameters in our model:

$$\hat{y} = f_\theta(x) = \sum_{j=1}^{d} \theta_j \phi_j(x)$$

Which of the following models are linear? **Select all that apply.**

- ☐ **A.** $f_\theta(x) = \theta_1 x + \theta_2 \sin(x)$
- ☐ **B.** $f_\theta(x) = \theta_1 x + \theta_2 \sin(x^2)$
- ☐ **C.** $f_\theta(x) = \theta_1$
- ☐ **D.** $f_\theta(x) = (\theta_1 x + \theta_2)x$
- ☐ E. $f_\theta(x) = \ln(\theta_1 x + \theta_2) + \theta_3$

11. Suppose we have data about 5 people shown below.

name	level	trials	phase
Magda	1	10	1
Valerie	5	20	-1
Kumar	2	15	1
Octavia	6	30	1
Dorete	6	5	-1

(a) Suppose we want to model the **level** of each person, and use the following constant model: $f_\theta(x) = \theta_1$. What is $\hat{\theta}_1$, the value that minimizes the average L2 loss?

Solution: We know from lecture 8 that the answer is just the mean of the observations, i.e. $\frac{1+5+2+6+6}{5} = 4$.

(b) We can also compute $\hat{\theta}$ from the previous part by using the normal equation $\hat{\theta} = (\Phi^T\Phi)^{-1}\Phi^T Y$. If we use the normal equation to compute $\hat{\theta}$, how many rows and columns are in the feature matrix Φ? Write your answer in the form **# rows** $\times$ **# columns**, e.g. 1×1.

Solution: Φ will have 5 rows and 1 column. All values should be equal to 1.

(c) What is $(\Phi^T\Phi)^{-1}\Phi^T$? Write your answer in the form of a **Python list**, e.g. $[1, 2, 3]$.

Solution: `[0.2, 0.2, 0.2, 0.2, 0.2]`

Gradient Descent

12. Momentum is a common variation of gradient descent in which we include the gradient at a previous step of the iteration in our current update equation. More formally it is defined as follows, where γ is the weight of momentum.

$$\theta^{t+1} = \theta^t - \alpha \frac{\partial L}{\partial \theta}\bigg|_{\theta^t} - \gamma \frac{\partial L}{\partial \theta}\bigg|_{\theta^{t-1}}$$

Fill in the code with the following keywords and numbers to implement gradient descent with momentum. Assume when $t = 0$ and $t = -1$, $\theta^t =$ `t0`.

Note that the same keyword/number can be used multiple times; some keywords may not be used at all. Only use one keyword per blank. You may not need all blanks.

theta	phi	y	theta_prev	num_iter	t0
temp	alpha	gamma	range	len	t

```
def grad(phi, y, theta):
    """Returns dL/dtheta. Assume correct implementation."""

def grad_desc_momentum(phi, y, num_iter, alpha, gamma, t0):
    """ Returns theta computed after num_iter iterations.
    phi: matrix, design matrix
    y: vector, response vector
    num_iter: scalar, number of iterations to run
    alpha: scalar, learning rate
    gamma: scalar, weight of momentum
    t0: theta for t=0
    """
   theta, theta_prev = ___<a>_________, _<b>_______________
    for _<c>__________ in _<d>___________(_<e>__________):
        g = grad(phi, y, theta)
        m = grad(phi, y, _<f>_____________)
        _<g>_____, _<h>_____ = _<i>_____ - _<j>___ * g - _<k>_____ * m,
            _<l>_______________
    return theta
```

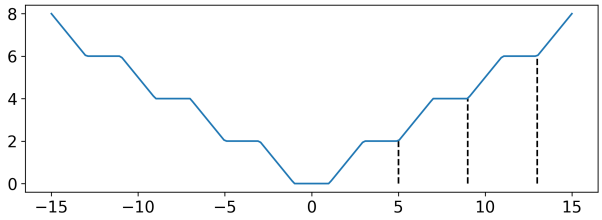
Recall that python allows multiple assignment, e.g. "`one, two = two, one`" would swap the values of `one` and `two`.

Solution:

```
def grad(phi, y, theta):
    """Returns dL/dtheta. Assume correct implementation."""

def grad_desc_momentum(phi, y, num_iter, alpha, gamma, t0):
    theta, theta_prev = t0, t0
    for t in range(num_iter):
        g = grad(phi, y, theta)
        m = grad(phi, y, theta_prev)
        theta, theta_prev = theta - alpha * g - gamma * m,
            theta
    return theta
```

13. Consider the following function of $f(\theta)$, which alternates between completely flat regions and regions of absolute slope equal to 1. **There is only one correct answer for each part.**



(a) Assuming that θ starts in a flat region that is not a minimum and $\alpha > 0$, will the basic gradient descent algorithm terminate at a minimum? Note that the basic gradient descent algorithm is just the same as version with momentum on the previous page, but where $\gamma = 0$.

◯ **A. Never** ◯ B. Maybe ◯ C. Yes with enough iterations

(b) Assuming that θ starts in a sloped region and $\alpha > 0$, will the basic gradient descent algorithm find the minimum?

◯ A. Never ◯ **B. Maybe** ◯ C. Yes with enough iterations

(c) Assuming that θ starts in a flat region that is not a minimum and $\alpha > 0$ and $\gamma > 0$, will the momentum gradient descent algorithm find the minimum?

◯ **A. Never** ◯ B. Maybe ◯ C. Yes with enough iterations

(d) Assuming that θ starts in a sloped region and $\alpha > 0$ and $\gamma > 0$, will the momentum gradient descent algorithm find the minimum?

◯ A. Never ◯ **B. Maybe** ◯ C. Yes with enough iterations

(e) Is $f(\theta)$ convex?

◯ A. Yes
◯ **B. No**
◯ C. No, but $-f(\theta)$ is convex
◯ D. No, but $f(-\theta)$ is convex

Feature Engineering

You Can't Forget About Those Tips

14. Recall from labs 5 and 6 the *tips* dataset from the seaborn library, which contains records about tips, total bills, and information about the person who paid the tip. There are a total of 244 records in **tips**. In addition, you can assume that there are no missing or NaN values in the dataset. The first 5 rows of the **tips** DataFrame are shown below, where *sex* takes on values $\in \{"Male", "Female"\}$, *smoker* takes on values $\in \{"Yes", "No"\}$, *day* takes on values from Monday to Sunday as strings, and *time* takes on values $\in \{"Breakfast", "Lunch", "Dinner"\}$.

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

(a) Suppose we use `pd.get_dummies` to create a one-hot encoding of only our `sex` column. This yields a feature matrix Φ_{q1} with **exactly 2 columns** `sex_Male`, `sex_Female`, where values can be either 0 or 1 in each column.

Which of the following are true? **Select all that apply.**

☐ **A. Φ_{q1} has 244 rows.**

☐ **B. Φ_{q1} has full column rank.**

☐ **C. $(\Phi_{q1}^T \Phi_{q1})$ is invertible.**

☐ D. None of the above

Solution:

☐ **A. True. The resulting matrix has one row for very row in the original tips table.**

☐ **B. True. The two columns are linearly independent.**

☐ **C. True, because Φ_{q1} has full column rank.**

☐ D. False. The above answer choices are true.

(b) Suppose we use `pd.get_dummies` to create a one-hot encoding of only our `sex` and `smoker` columns. This yields a feature matrix Φ_{q2} with 4 columns.

Which of the following are true? **Select all that apply.**

☐ **A. Φ_{q2} has 244 rows.**

☐ B. Φ_{q2} has full column rank.

☐ C. $(\Phi_{q2}^T \Phi_{q2})$ is invertible.

☐ D. None of the above

Solution:

☐ **A. True. The resulting matrix has one row for very row in the original tips table.**

☐ B. False. We can generate the sex_male column by adding our smoker columns and then subtracting the female column.

☐ C. False. Since Φ_{q2} doesn't have full column rank, this means that $(\Phi_{q2}^T \Phi_{q2})$ won't be invertible.

☐ D. False. We chose an answer choice above.

(c) Suppose we use `pd.get_dummies` to create a one-hot encoding of only our `sex` and `smoker` columns, and also include a bias column. This yields a feature matrix Φ_{q3} with 5 columns.

Which of the following are true? **Select all that apply.**

☐ **A. Φ_{q3} has 244 rows.**

☐ B. Φ_{q3} has full column rank.

☐ C. $(\Phi_{q3}^T \Phi_{q3})$ is invertible.

☐ D. None of the above

Solution:

☐ **A. True. The resulting matrix has one row for very row in the original tips table.**

☐ B. False. Same as the previous problem.

☐ C. False. Same as the previous problem.

☐ D. False. We chose an answer choice above.

(d) For the `day` column, we can either use a one-hot encoding or an integer encoding. By integer encoding, we mean mapping Monday to 1, Tuesday to 2, and so on. Which of the following statements are true? **Select all that apply.**

☐ A. One-hot encoding creates fewer columns than integer encoding.

☐ **B. One-hot encoding gives all days of the week the same weight, while integer encoding gives certain days of the week higher weight than others.**

☐ **C. The columns generated by the one-hot encoding of the days of the week are linearly independent of each other.**

☐ D. None of the above

Solution:

☐ A. False. One-hot encoding creates more columns than integer encoding, as a column is added for every non-numerical value a certain feature can take on.

☐ **B. True. This is exactly why we one-hot encode non-numerical values.**

☐ **C. True. The columns generated are linearly independent of each other (but may not be when combined with other columns, e.g. a bias column).**

☐ D. False. We chose 2 of the answer choices above.

More Feature Engineering

15. Which of the following are reasons to use N-Gram Encoding ($N > 1$) instead of Bag-of-words (1-Gram) encoding? For simplicity, you may assume that $N = 2$ for N-Gram Encoding. **Select all that apply.**

☐ A. Vectors for N-Gram encoding are less sparse than vectors from Bag-of-words encoding.

☐ B. Vectors for N-Gram encoding have lower dimension than vectors from Bag-of-words encoding.

☐ C. It is easier for N-Gram encoding to deal with unseen combinations at prediction time.

☐ **D. N-Gram helps preserve word order while Bag-of-words does not.**

☐ E. None of the above

Solution:

☐ A. False. Vectors for N-Gram are usually more sparse than Bag-of-words vectors.

☐ B. False. Vectors for N-Gram usually have higher dimension than Bag-of-words vectors.

☐ C. False. Both encoding schemes have to deal with the issue of unseen combinations at prediction time.

☐ **D. True. N-gram does help preserve word order while Bag-of-words does not.**

Regularization

Elastic Net Regularization

16. Elastic Net is a regression technique that combines L_1 and L_2 regularization. It is preferred in many situations as it possesses the benefits of both LASSO and Ridge Regression. Minimizing the L2 loss using Elastic Net is as follows, where $\lambda_1, \lambda_2 >= 0$, $\lambda_1 + \lambda_2 = \lambda$, $\lambda > 0$.

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{n} \sum_{i} (y_i - \theta x)^2 + \lambda_1 \sum_{j=1}^{p} |\theta_j| + \lambda_2 \sum_{j=1}^{p} \theta_j^2$$

Suppose our goal was to get sparse parameters, i.e. we want as many parameters as possible to be zero. Which of the following choices for λ_1, λ_2 are most consistent with this goal, assuming $\lambda = 1$? **There is only one correct answer.**

◯ A. $\lambda_1 = 0, \lambda_2 = 1$

◯ B. $\lambda_1 = 0.5, \lambda_2 = 0.5$

◯ **C.** $\lambda_1 = 1, \lambda_2 = 0$

Solution: We know that LASSO encourages sparsity in our optimal weights. Setting λ_1 to 1 means we are using LASSO.

17. What happens to bias and variance as we increase the value of λ? Assume $\lambda_2 = \lambda_1$. **There is only one correct answer in each part.** You will be asked to justify why in the next question.

(a) Bias:

◯ **A. Bias goes up**

◯ B. Bias stays the same

◯ C. Bias goes down

(b) Variance:

◯ A. Variance goes up

◯ B. Variance stays the same

◯ **C. Variance goes down**

18. Justify why by marking the true statements. **Select all that apply for each part.**

(a) Bias:

☐ A. Bias goes down because increasing λ reduces over fitting.

☐ B. Bias goes down because bias is minimized when $\lambda_2 = \lambda_1$.

☐ **C. Bias goes up because increasing λ penalizes complex models, limiting the set of possible solutions.**

☐ D. Bias goes up because the loss function becomes non-convex for sufficiently large λ.

☐ E. None of the above

(b) Variance:

☐ A. Variance goes down because increasing λ encourages the value of the loss to decrease.

☐ **B. Variance goes down because increasing λ penalizes large model weights.**

☐ C. Variance goes up because because increasing λ increases bias.

☐ D. Variance goes up because increasing λ increases the magnitude of terms in the loss function.

☐ E. None of the above

19. What happens to the model parameters $\hat{\theta}$ as $\lambda \to \infty$, i.e. what is $\lim_{\lambda \to \infty} \hat{\theta}$? **Select all that apply.**

☐ **A. Converge to 0.**

☐ B. Diverge to infinity.

☐ C. Converge to values that minimize the L2 loss.

☐ D. Converge to equal but non-zero values.

☐ E. Converge to a sparse vector.

Solution: The model parameters go to 0.

20. Of the choices below, why do we prefer to use ridge regression over linear regression (i.e. the normal equation) in certain cases? **Select all that apply.**

☐ **A. Ridge regression always guarantees an analytic solution, but the normal equation does not.**

☐ B. Ridge regression encourages sparsity in our model parameters, which is helpful for inferring useful features.

☐ C. Ridge regression isn't sensitive to outliers, which makes it preferable over linear regression.

☐ D. Ridge regression always performs just as well as linear regression, with the added benefit of reduced variance.

☐ E. None of the above

Solution:

☐ **A. The regularization term guarantees $(A^TA + \lambda I)$ is invertible, as discussed in discussion 7.**

☐ B. This is the description for LASSO.

☐ C. It is sensitive to outliers.

☐ D. Doesn't always perform better.

More Linear Models

21. Suppose in some universe, the true relationship between the measured luminosity of a single star Y can be written in terms of a single feature ϕ of that same star as

$$Y = \theta^* \phi + \epsilon$$

where $\phi \in \mathbb{R}$ is some non-random scalar feature, $\theta^* \in \mathbb{R}$ is a non-random scalar parameter, and ϵ is a random variable with $\mathbb{E}[\epsilon] = 0$ and $\text{Var}(\epsilon) = \sigma^2$. For each star, you have a set of features $\mathbf{\Phi} = \begin{bmatrix} \phi_1 & \phi_2 & \dots & \phi_n \end{bmatrix}^T$ and luminosity measurements $\mathbf{y} = \begin{bmatrix} y_1 & y_2 & \dots & y_n \end{bmatrix}^T$ generated by this relationship. Your Φ may or may not include the feature ϕ described above. The ϵ_i for the various y_i have the same probability distribution and are independent of each other.

(a) What is $\mathbb{E}[Y]$?

◯ A. 0 ◯ **B.** $\theta^* \phi$ ◯ C. $\phi(\mathbf{\Phi}^T \mathbf{\Phi})^{-1} \mathbf{\Phi}^T \mathbf{y}$
◯ D. θ^* ◯ E. None of the above

Solution: $\mathbb{E}[Y] = \mathbb{E}[\theta^* \phi + \epsilon] = \theta^* \phi + 0$

(b) What is $\text{Var}(Y)$?

◯ A. $\frac{\sigma^2}{n}$ ◯ B. $\frac{\sigma^2}{n^2}$ ◯ C. 0
◯ D. $\frac{1}{n-1} \sum_{i=1}^{n} \left(y_i - \frac{1}{n} \sum_{i=1}^{n} y_i \right)^2$ ◯ **E. None of the above**

Solution: $\text{Var}(Y) = \text{Var}(\theta^* x + \epsilon) = \text{Var}(\epsilon) = \sigma^2$

(c) Suppose you have information about the exact ϕ value for each star, but try to fit a linear model for Y that includes an intercept term θ_0.

$$Y = \theta_0 + \theta_1 \phi$$

Note the true relationship has no intercept term, so our model is not quite correct. Let $\hat{\theta}_0$ and $\hat{\theta}_1$ be the values that minimize the average L_2 loss. Let $\mathbf{y}$ be the actual observed data and $\hat{\mathbf{y}} = \hat{\theta}_0 + \hat{\theta}_1 \boldsymbol{\phi}$ be the fitted values.

i. Which of the following could possibly be the value of $\hat{\theta}_0$ after fitting our model? **Select all that apply; at least one is correct.**

☐ **A. -1** ☐ **B. 0** ☐ **C. 1** ☐ **D. 10**

Solution: There are no restrictions on $\hat{\theta}_0$ given our assumptions.

ii. Which of the following could possibly be the residual vector for our model? **Select all that apply; at least one is correct.**

☐ **A.** $\begin{bmatrix}-2 & -4 & 6\end{bmatrix}^T$ ☐ B. $\begin{bmatrix}0.0001 & 0.0003 & -0.0005\end{bmatrix}^T$
☐ C. $\begin{bmatrix}3 & 12 & -9\end{bmatrix}^T$ ☐ D. $\begin{bmatrix}1 & 1 & 1\end{bmatrix}^T$

Solution: Since we are including an intercept/bias term, $\mathbf{\Phi}$ has a column of 1s, which we denote with a boldface $\mathbf{1}$. Optimality requires orthogonality of the residual vector with the column space of $\mathbf{\Phi}$, which requires $\mathbf{1}^T\boldsymbol{e} = \sum_{i=1}^n 1 \times e_i = \sum_{i=1}^n e_i = 0$. (A) is the only choice satisfying this condition.

22. Suppose we create a new loss function called the OINK loss, defined as follows for a single observation:

$$L_{OINK}(\theta, x, y) = \begin{cases} a(f_\theta(x) - y) & f_\theta(x) \geq y \\ b(y - f_\theta(x)) & f_\theta(x) < y \end{cases}$$

You decide to use the constant model (given on the left) and average OINK loss (given on the right).

$$f_\theta(x) = \theta \qquad\qquad L(\theta, \mathbf{x}, \mathbf{y}) = \frac{1}{n}\sum_{i=1}^{n} L_{OINK}(\theta, x_i, y_i)$$

The data are given below. Find the optimal $\hat{\theta}$ that minimizes the loss.

x	3	1	5	4	2	0	6
y	40	0	50	30	20	60	10

(a) when $a = b = 1$

(b) when $a = 1$, $b = 5$

(c) when $a = 3$, $b = 6$

Solution: If a = b = 1, then the OINK loss is just the L1 loss and the optimal theta is simply the median, $\hat{\theta} = 30$.

With a = 1, and b = 5, the OINK loss is very similar to the L1 loss, it's just that estimates that are below the observed value are penalized 5 times as much. Thus, instead of balancing the number of observations above and below our estimate (which yields the median), we must balance the 5x the number below with the number above. This yields $\hat{\theta} = 50$.

With a = 3, and b = 6, the OINK loss is still very similar to the L1 loss, it's just that estimates that are below the true value are penalized 3 times as much, and estimates above are penalized 6 time as much. Thus, instead of balancing the number of observations above and below our estimate (which yields the median), we must balance the 3x the number below with 6x the number above. This is equivalent to making sure there are twice as many numbers below as above. This yields $\hat{\theta} = 40$.