DS 100/200: Principles and Techniques of Data Science **Date: August 12, 2019**

Exam Review Solutions

Name:

EDA & Visualization

1. For each of the following scenarios, determine which plot type is *most* appropriate to reveal the distribution of and/or the relationships between the following variable(s). For each scenario, select only one plot type. Some plot types may be used multiple times.

   A. histogram B. pie chart C. bar plot D. line plot

   E. side-by-side boxplots F. scatter plot G. stacked bar plot H. overlaid line plots

   (a) Sale price and number of bedrooms for houses sold in Berkeley in 2010.

   > **Solution: E. Side-by-side Boxplots.** We might imagine using a scatter plot since we are plotting the relationship between two numeric quantities. However because the number of bedrooms is an integer and most houses will only have a small number, we are likely to encounter *over-plotting* in the scatter plot. Therefore side-by-side boxplots are likely to be most informative.

   (b) Sale price and date of sale for houses sold in Berkeley between 1995 and 2015.

   > **Solution: F. Scatter Plot.** Here we are plotting two numeric quantities with sufficient spread on each axis.

   (c) Infant birth weight (grams) for babies born at Alta Bates hospital in 2016.

   > **Solution: A. Histogram.** Here we are plotting the distribution of a likely large number of observations and therefore a histogram would be most appropriate.

   (d) Mother's education-level (highest degree held) for students admitted to UC Berkeley in 2016.

   > **Solution: C. Bar Plot.** Here we want to visualize counts of a categorical variable.

   (e) SAT score and HS GPA of students admitted to UC Berkeley in 2016.

   > **Solution: F. Scatter Plot.** Here we are visualizing the relationship between two continuous quantities.

(f) The percentage of female student admitted to UC Berkeley each year from 1950 to 2000.

> **Solution: D. Line plot.** This allows us to see the trends over time.

(g) SAT score for males and females of students admitted to UCB from 1950 to 2000

> **Solution: E. side-by-side boxplots**. This allows us to see the distributions of SAT scores per gender and year.

Optimization

2. Fix the following buggy Python implementation of gradient descent:

```python
def grad_descent(X, Y, theta0, grad_function, max_iter = 1000):
    """X: A 2D array, the feature matrix.
    Y: A 1D array, the response vector.
    theta0: A 1D array, the initial parameter vector.
    grad_function: Maps a parameter vector, a feature matrix, and a
        response vector to the gradient of some loss function at the
        given parameter value. The return value is a 1D array."""
    theta = theta0
    for t in range(1, max_iter+1):
        grad = grad_function(theta, X, Y)
        theta = theta0 + t * grad
    return grad
```

Solution: The last two lines need to change:

```python
def grad_descent(X, Y, theta0, grad_function, max_iter = 1000):
    """X: A 2D array, the feature matrix.
    Y: A 1D array, the response vector.
    theta0: A 1D array, the initial parameter vector.
    grad_function: Maps a parameter vector, a feature matrix, and
        a response vector to the gradient of some loss function at
        the given parameter value. The return value is a 1D
        array."""
    theta = theta0
    for t in range(1, max_iter+1):
        grad = grad_function(theta, X, Y)
        theta = theta - (1/t) * grad
    return theta
```

3. Suppose you are given a dataset $\{(x_i, y_i)\}_{i=1}^n$ where $x_i \in \mathbb{R}$ is a one dimensional feature and $y_i \in \mathbb{R}$ is a real-valued response. You use f_θ to model the data where θ is the model parameter. You choose to use the following regularized loss:

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

   (a) This regularized loss is best described as:

(a) Average absolute loss with L^2 regularization.
(b) Average squared loss with L^1 regularization.
(c) Average squared loss with L^2 regularization.
(d) Average Huber loss with λ regularization.

(b) Suppose you choose the model $f_\theta(x_i) = \theta x_i^3$. Using the above objective derive the loss minimizing estimate for θ.

Solution:

Step 1: Take the derivative of the loss function.

$$\frac{\partial}{\partial\theta}L(\theta) = \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial\theta}\left(y_i - \theta x_i^3\right)^2 + \frac{\partial}{\partial\theta}\lambda\theta^2 \tag{1}$$

$$= -\frac{2}{n}\sum_{i=1}^{n}\left(y_i - \theta x_i^3\right)x_i^3 + 2\lambda\theta \tag{2}$$

Step 2: Set derivative equal to zero and solve for θ.

$$0 = -\frac{2}{n}\sum_{i=1}^{n}\left(y_i - \theta x_i^3\right)x_i^3 + 2\lambda\theta \tag{3}$$

$$\theta = \frac{1}{n\lambda}\sum_{i=1}^{n}\left(y_i - \theta x_i^3\right)x_i^3 \tag{4}$$

$$\theta = \frac{1}{n\lambda}\sum_{i=1}^{n}y_i x_i^3 - \theta\frac{1}{n\lambda}\sum_{i=1}^{n}x_i^6 \tag{5}$$

$$\theta\left(1 + \frac{1}{n\lambda}\sum_{i=1}^{n}x_i^6\right) = \frac{1}{n\lambda}\sum_{i=1}^{n}y_i x_i^3 \tag{6}$$

$$\tag{7}$$

Thus we obtain the final answer:

$$\boxed{\hat{\theta} = \frac{\frac{1}{n}\sum_{i=1}^{n}y_i x_i^3}{\left(\lambda + \frac{1}{n}\sum_{i=1}^{n}x_i^6\right)}} \tag{8}$$

Inference

4. **True or False.** Determine whether the following statements are true or false.
   (a) Suppose we have 100 samples drawn independently from a population. If we construct a 95% confidence interval for each sample, we expect 95 of them to include the **sample** mean.

   > **Solution: False.** All of them should include the sample mean.

   (b) We often prefer a pseudo-random number generator because our simulations results can be exactly reproduced by controlling the seed.

   > **Solution: True.** This is an essential aspect of reproducible data analyses and simulation studies.

5. Suppose we have a Pandas Series called **`thePop`** which contains a census of **25000 subjects**. We also have a simple random sample of **400 individuals** saved in the Series **`theSample`**. We are interested in studying the behavior of the bootstrap procedure on the simple random sample. Fill in the blanks in the code below to construct **10000 bootstrapped estimates** for the **median**.

```
boot_stats = [

        ______________________

        .sample(n = ____________, replace = ____________)

        .______________________()


        for j in range(________________)
    ]
```

> **Solution:**
>
> ```
> boot_stats = [
>
>         theSample
> ```

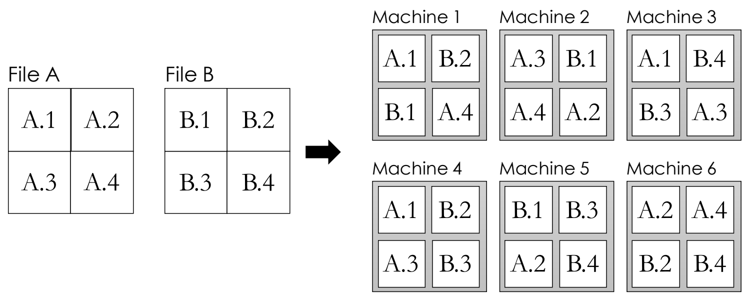
```
        .sample(n = 400, replace = True)

        .median()

        for j in range(10000)

        ]
```

6. Consider the following layout of the files `A` and `B` onto a distributed file-system of 6 machines.



Assume that all blocks have the same file size and computation takes the same amount of time.

(a) (1 point) If we wanted to load file `A` in parallel which of the following sets of machines would give the best load performance:

A. $M1, M2$ B. $M1, M2, M3$ C. $M2, M4, M5, M6$

> **Solution:** While all choices would be able to load the file, only $M2, M4, M5, M6$ could load the file in parallel.

(b) (1 point) If we were to lose machines $M1$, $M2$, and $M3$ which of the following file or files would we lose (select all that apply).

A. File `A` B. File `B` C. We would still be able to load both files.

(c) (1 point) If each of the six machines fail with probability p, what is the probability that we will lose block $B.1$ of file `B`.?

A. $3p$ B. p^3 C. $(1-p)^3$ D. $1-p^3$