Homework 9

PSTAT 5A: Spring 2023, with Ethan P. Marzban

Instructions

- Please submit your work to Gradescope by no later than **11:59pm on Wednesday, June 7**. As a reminder, late homework will not be accepted.
- Recall that you will be asked to upload a **single** PDF containing your work for *both* the programming and non-programming questions to Gradescope.
  - You can merge PDF files using either Adobe Acrobat, or using adobe's online PDF merger at this link.

Caution

Be aware that some parts may be easier (or, in fact, may *need* to be) computed using Python. If you do use Python for any part, please write down the code you used.

Problem 1: Weight Loss

A new weight loss regimen claims to significantly reduce the weights of its participants. To test these claims, a researcher takes a representative sample of 100 volunteers, records their weights before the regimen, and then records their weights after the regimen. (All measurements are in lbs.) The summary statistics are displayed below:

	Sample Mean	Sample Std. Dev.
Pre-Regimen	151.2	21.3
Post-Regimen	145.2	19.1

Let Population 1 be the set of all pre-regimen weights, and Population 2 be the set of all post-regimen weights. Additionally, assume (for now) that all independence assumptions are satisfied.

a. Define the parameters of interest, μ_1 and μ_2.

Solution: μ_1 = the average weight of all pre-regimen individuals, and μ_2 = the average weight of all post-regimen individuals.

b. State the null and alternative hypotheses. (Remember that the null can be thought of as the "status quo".) Think carefully about the alternative: as a hint, this is *not* a two-sided test.

Solution: In this case, the "status quo" is that the regimen does *not* work; i.e. that $\mu_1 = \mu_2$. The alternative is that the regimen *does* work, which would mean $\mu_1 > \mu_2$. Hence:

$$\begin{bmatrix} H_0 : & \mu_1 = \mu_2 \\ H_A : & \mu_1 > \mu_2 \end{bmatrix}$$

or, in terms of differences,

$$\begin{bmatrix} H_0 : & \mu_2 - \mu_1 = 0 \\ H_A : & \mu_2 - \mu_1 < 0 \end{bmatrix}$$

(As a rule-of-thumb: for the purposes of this class, the null hypothesis will always be a statement of equality.)

c. Compute the value of the test statistic.

Solution:

$$\text{TS} = \frac{\overline{Y} - \overline{X}}{\sqrt{\frac{s_x^2}{n_1} + \frac{s_y^2}{n_2}}} = \frac{145.2 - 151.2}{\sqrt{\frac{21.3^2}{100} + \frac{19.1^2}{100}}} = -2.09721$$

d. Assuming the null is correct, what is the approximate distribution of the sampling distribution? Be sure to include any/all relevant parameters.

Solution: Because we are assuming the necessary independence conditions, we know that the test statistic will, under the null, follow a t-distribution with degrees of freedom obtained using the Satterthwaite Approximation:

$$\text{df} = \text{round}\left\{ \frac{\left[\left(\frac{s_X^2}{n_1}\right) + \left(\frac{s_Y^2}{n_2}\right)\right]^2}{\frac{\left(\frac{s_X^2}{n_1}\right)^2}{n_1 - 1} + \frac{\left(\frac{s_Y^2}{n_2}\right)^2}{n_2 - 1}} \right\} = \text{round}\left\{ \frac{\left[\left(\frac{21.3^2}{100}\right) + \left(\frac{19.1^1}{100}\right)\right]^2}{\frac{\left(\frac{21.3^2}{100}\right)^2}{100 - 1} + \frac{\left(\frac{19.1^2}{100}\right)^2}{100 - 1}} \right\} = 196$$

That is to say,

$$\text{TS} \overset{H_0}{\sim} t_{196}$$

e. What is the p-value of the observed test statistic?

Solution: Using the Python command `stats.scipy.t.cdf(-2.09721, 196)`, we find a p-value of around 0.01863 .

f. Now, carry out the test at an $\alpha = 0.05$ level of significance. Be sure to phrase your conclu-

sions in terms of the context of the problem.

Solution: We only reject when the p-value is less than the level of significance. This is the case for the problem, so we have evidence to reject the null:

> At a 5% level of significance, there was sufficient evidence reject the null hypothesis that the weight loss program is ineffective against the alternative that it *is* effective in lowering weight.

or, more colloquially:

> At a 5% level of significance, there was sufficient evidence to support the weight loss program's claims that it significantly reduces weight.

g. **Critical Thinking:** Do you think our assumption that "all independence assumptions are satisfied" is justified? Why or why not?

Solution: It is probably *not* justified to assume independence, since the pre- and post-regimen datapoints were collected from the same people (i.e. people were tracked over time). [As a side not: now that we know how to classify experiments/studies, we can see this is a longitudinal study and, hence, observations will be serially correlated.]

Problem 2: ANOVA By Hand

In this problem, we will work through the computations of an ANOVA by hand. You must show all of your work clearly.

The data we will consider is:

$$\boldsymbol{x_1} = \{-1,\ 0,\ 1\}$$
$$\boldsymbol{x_2} = \{0,\ 1,\ 1,\ 2\}$$
$$\boldsymbol{x_3} = \{1,\ 2,\ 1\}$$

a. Compute the numerator and denominator degrees of freedom.

Solution: The numerator degrees of freedom is $k - 1$, where k is the number of groups: i.e. $\text{df}_1 = 2$. The denominator degrees of freedom is $n - k$ where n is the aggregate number of observations, which in this case is $3 + 4 + 3 = 10$; i.e. $\text{df}_2 = 10 - 3 = 7$.

b. Compute the **group means**, $\overline{x}_1$, $\overline{x}_2$, and $\overline{x}_3$.

Solution:

$$\overline{x}_1 = \frac{1}{3}(-1 + 0 + 1) = 0$$

$$\bar{x}_2 = \frac{1}{4}(0+1+1+2) = 1$$
$$\bar{x}_3 = \frac{1}{3}(1+2+1) = \frac{4}{3}$$

c. Compute the **grand mean**, $\bar{x}$ (i.e. the mean across all observations).

Solution:

$$\bar{x} = \frac{1}{10}(-1+0+1+0+1+1+2+1+2+1) = \frac{4}{5}$$

d. Compute the **sum of squares between groups**:

$$\text{SS}_\text{G} = \sum_{i=1}^{k} n_i(\bar{x}_i - \bar{x})^2$$

where n_i denotes the size of the i^{th} group.

Solution:

$$\text{SS}_\text{G} = 3\left(0-\frac{4}{5}\right)^2 + 4\left(1-\frac{4}{5}\right)^2 + 3\left(\frac{4}{3}-\frac{4}{5}\right)^2 = \frac{44}{15} = 2.9\overline{3}$$

e. Compute the **sum of squares total**:

$$\text{SS}_\text{T} = \sum_{i=1}^{k}(x_i - \bar{x})^2$$

where the sum is taken over *all* observations in the dataset.

Solution:

$$\begin{aligned}\text{SS}_\text{G} = &\left(-1-\frac{4}{5}\right)^2 + \left(0-\frac{4}{5}\right)^2 + \left(1-\frac{4}{5}\right)^2 \\ &+ \left(0-\frac{4}{5}\right)^2 + \left(1-\frac{4}{5}\right)^2 + \left(1-\frac{4}{5}\right)^2 + \left(2-\frac{4}{5}\right)^2 \\ &+ \left(1-\frac{4}{5}\right)^2 + \left(2-\frac{4}{5}\right)^2 + \left(1-\frac{4}{5}\right)^2 = 7.6\end{aligned}$$

f. Compute the **sum of squared errors**:

$$\mathrm{SS_E} = \mathrm{SS_T} - \mathrm{SS_{SS_G}}$$

Solution:

$$\mathrm{SS_E} = \mathrm{SS_T} - \mathrm{SS_{SS_G}} = 7.6 - \frac{44}{15} = \frac{14}{3} = 4.\overline{6}$$

g. Compute the **mean-square between groups** and **mean-square error**:

$$\mathrm{MS_G} = \frac{\mathrm{SS_G}}{\mathrm{df_G}}; \qquad \mathrm{MS_E} = \frac{\mathrm{SS_E}}{\mathrm{df_E}}$$

Solution:

$$\mathrm{MS_G} = \frac{\left(\frac{44}{15}\right)}{2} = \frac{22}{15} = 1.4\overline{6}$$

$$\mathrm{MS_E} = \frac{\left(\frac{14}{3}\right)}{7} = \frac{2}{3} = 0.\overline{6}$$

h. Compute the value of the F-statistic.

Solution:

$$F = \frac{\mathrm{MS_G}}{\mathrm{MS_E}} = \frac{\left(\frac{22}{15}\right)}{\left(\frac{2}{3}\right)} = \frac{11}{5} = 2.2$$

i. Compute the p-value of the statistic. (You will need to use Python.)

Solution: Using the Python command `1 - scipy.stats.F.cdf(2.2, 2, 7)` we find the p-value to be 0.1814.

j. Finally, combine your answers to produce an ANOVA table.

Solution:

	DF	**Sum Sq.**	**Mean Sq.**	F**–value**	$\mathbb{P}(>F)$
Btwn. Grps.	2	$\left(\frac{44}{15}\right)$	$\left(\frac{22}{15}\right)$	2.2	0.1814
Residuals	7	$\left(\frac{14}{3}\right)$	$\left(\frac{2}{3}\right)$		

Problem 3: Filling In an ANOVA Table

In the following parts, you will be presented with an ANOVA table that has some entries missing. Fill in the missing entries, and provide justification as to how you found those missing values.

Part (a):

	DF	Sum Sq	Mean Sq	F value	Pr(>F)
Btwn. Groups	4	10	<???>	<???>	<???>
Residuals	<???>	50	0.5		

Solution: Let's tackle MS_G first. We know that

$$\text{MS}_\text{G} = \frac{\text{SS}_\text{G}}{\text{df}_\text{G}} = \frac{10}{4} = 2.75$$

Similarly, we have

$$\text{MS}_\text{E} = \frac{\text{SS}_\text{E}}{\text{df}_\text{E}} \implies \text{df}_\text{E} = \frac{\text{SS}_\text{E}}{\text{MS}_\text{E}} = \frac{50}{0.5} = 100$$

Additionally,

$$F = \frac{\text{MS}_\text{G}}{\text{MS}_\text{E}} = \frac{2.75}{0.5} = 5.5$$

and, using Python, we compute the p-value to be around 0.00048. Hence, our complete ANOVA table looks like:

	DF	Sum Sq.	Mean Sq.	$F-$value	$\mathbb{P}(> F)$
Btwn. Grps.	4	10	2.75	5.5	0.00048
Residuals	100	50	0.5		

Part (b):

	DF	Sum Sq	Mean Sq	F value	Pr(>F)
Btwn. Groups	10	20	2	<???>	0.8636
Residuals	120	<???>	<???>		

Solution: Here, we have to start working from right-to-left. That is, we first need to find the $F-$statistic. We can do so using

`scipy.stats.f.ppf(1 - 0.8636, 10, 120)` $= 0.53339$

Then,

$$F = \frac{\text{MS}_\text{G}}{\text{MS}_\text{E}} \implies \text{MS}_\text{E} = \frac{\text{MS}_\text{G}}{F} = \frac{2}{0.53339} = 3.7496$$

Finally,

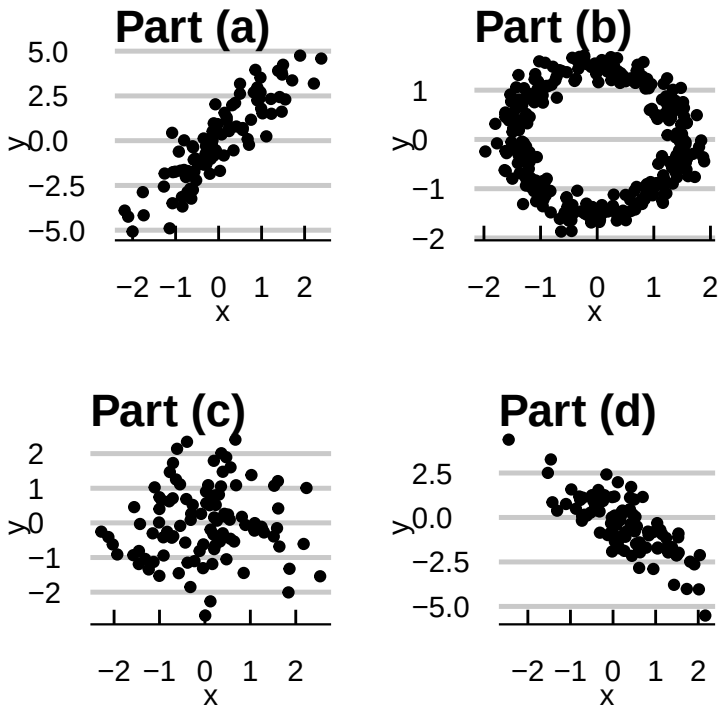
$$\text{MS}_\text{E} = \frac{\text{SS}_\text{E}}{120} \implies \text{MS}_\text{E} = 120 \cdot (3.7496) = 449.952$$

Hence, our complete ANOVA table looks like:

	DF	**Sum Sq.**	**Mean Sq.**	***F*−value**	$\mathbb{P}(> F)$
Btwn. Grps.	10	20	2	0.53339	0.00048
Residuals	120	449.952	3.7496		

Problem 4: Graphical Correlations

In each of the following parts you will be presented with a scatterplot of two variables x and y. Based on the scatterplot, determine whether you believe the correlation between x and y to be positive, negative, or zero. Justify your answers.



Solution:

a) We would expect the correlation to be positive, as the scatterplot displays a positive

linear trend.

b) There is no *linear* trend in the scatterplot, meaning r would likely be somewhere close to zero.

c) There is no trend in the scatterplot, meaning r would likely be close to zero.

d) We would expect the correlation to be positive, as the scatterplot displays a negative linear trend.

Problem 5: Numerical Computations

Consider the following two sets of numbers:

$$x = \{1, 2, 3, 1, 2, 5, 4\}$$
$$y = \{3, 4, 1, 4, 4, 2, 1\}$$

a. Compute the correlation between x and y. Do **not** use Python, except for arithmetic computations (i.e. you may use Python as a calculator, but do **NOT** use any more advanced functions like `numpy.std()`, or `np.mean()`.)

Solution:

$$\overline{x} = \frac{1}{7}(1+2+3+1+2+5+4) = \frac{18}{7}$$
$$\overline{y} = \frac{1}{7}(3+4+1+4+4+2+1) = \frac{19}{7}$$
$$s_X^2 = \frac{1}{6}\left[\left(1-\frac{18}{7}\right)^2 + \left(2-\frac{18}{7}\right)^2 + \left(3-\frac{18}{7}\right)^2 + \left(1-\frac{18}{7}\right)^2 + \left(2-\frac{18}{7}\right)^2 + \left(5-\frac{18}{7}\right)^2 + \left(4-\frac{18}{7}\right)^2\right]$$
$$= \frac{16}{7}$$
$$s_X = \sqrt{s_X^2} = \sqrt{\frac{16}{7}} = \frac{4}{\sqrt{7}}$$
$$s_Y^2 = \frac{1}{6}\left[\left(3-\frac{19}{7}\right)^2 + \left(4-\frac{19}{7}\right)^2 + \left(1-\frac{19}{7}\right)^2 + \left(4-\frac{19}{7}\right)^2 + \left(4-\frac{19}{7}\right)^2 + \left(2-\frac{19}{7}\right)^2 + \left(1-\frac{19}{7}\right)^2\right]$$
$$= \frac{40}{21}$$
$$s_Y = \sqrt{s_X^2} = \sqrt{\frac{40}{21}} = 2\sqrt{\frac{10}{21}}$$
$$\sum_{i=1}^{n}(x_i-\overline{x})(y_i-\overline{y}) = \left(1-\frac{18}{7}\right)\cdot\left(3-\frac{19}{7}\right) + \left(2-\frac{18}{7}\right)\cdot\left(4-\frac{19}{7}\right)$$

$$
\begin{aligned}
&\quad + \left(3 - \frac{18}{7}\right) \cdot \left(1 - \frac{19}{7}\right) + \left(1 - \frac{18}{7}\right) \cdot \left(4 - \frac{19}{7}\right) \\
&\quad + \left(2 - \frac{18}{7}\right) \cdot \left(4 - \frac{19}{7}\right) + \left(5 - \frac{18}{7}\right) \cdot \left(2 - \frac{19}{7}\right) \\
&\quad + \left(4 - \frac{18}{7}\right) \cdot \left(1 - \frac{19}{7}\right) \\
&= -\frac{62}{7}
\end{aligned}
$$

$$
r = \frac{1}{n-1} \cdot \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{s_X \cdot s_Y} = \frac{1}{6} \cdot \frac{\left(-\frac{62}{7}\right)}{\left(\frac{4}{\sqrt{7}}\right) \cdot \left(2 \cdot \sqrt{\frac{10}{21}}\right)}
$$

$$
= -\frac{31\sqrt{30}}{240} \approx -0.707475
$$

b. Compute the coefficients of the OLS regression line when regressing y onto x (i.e. treating y as the response variable and x as the explanatory variable). Again, only use Python as a calculator for arithmetic computations.

Solution:

$$
\widehat{\beta_1} = \frac{s_Y}{s_X} \cdot r
$$

$$
= \frac{\left(2 \cdot \sqrt{\frac{10}{21}}\right)}{\left(\frac{4}{\sqrt{7}}\right)} \cdot \left(-\frac{31\sqrt{30}}{240}\right) = -\frac{31}{48} = -0.64583\overline{3}
$$

$$
\widehat{\beta_0} = \overline{y} - \widehat{\beta_1} \cdot \overline{x} = \frac{19}{7} + \frac{31}{48} \cdot \frac{18}{7} = \frac{35}{8} = 4.375
$$

That is, the equation of the OLS regression line is

$$
\hat{y} = \frac{35}{8} - \frac{31}{48}x = \frac{210 - 31x}{48}
$$

Problem 6: Programming

Part (a)

Task 1

Write a function called `cor()` that takes in two lists x and y, and returns the Pearson's Correlation between x and y. Check that `cor([1, 2, 3], [1, 2, 3])` returns 1.

Solution

```
def cor(x, y):
    """
    computes the correlation between two lists x and y of the same length.
    """
    import numpy as np

    if(len(x) != len(y)):
        print("Error: inputs must be of the same length")
    else:
        xbar = np.mean(x)
        ybar = np.mean(y)

        sx = np.std(x, ddof = 1)
        sy = np.std(y, ddof = 1)

        n = len(x)

        return 1 / (n - 1) * sum( ((x - xbar) / sx) * ((y - ybar) / sy) )
```

Test the function:

```
cor([1, 2, 3], [1, 2, 3])
```

```
1.0
```

Part (b)

Task 2

Write a function called `reg()` that takes in two lists x and y and returns the OLS estimates of the intercept and slope of regressing y onto x. Your function should return a list with two elements: $\widehat{\beta}_0$ and $\widehat{\beta}_1$, in that order.

Solution

```
def reg(x, y):
    """
    Returns the equation of the OLS regression line,
    regression y onto x.
    """
    import numpy as np

    r = cor(x, y) # this is the same cor() function from task 1
    beta_1_hat = (np.std(y, ddof = 1) / np.std(x, ddof = 1)) * r
    beta_0_hat = np.mean(y) - beta_1_hat * np.mean(x)

    return [beta_0_hat, beta_1_hat]
```