CSCE554 HW2

July 11, 2019

1 CSCE 554 HW 2

2 Marvin Newlin

3 11 Jul 19

```
In [1]: import numpy as np
        import pandas as pd
        import scipy
        from scipy import stats
        import matplotlib.pyplot as plt
        import pandas
        import random

        import matplotlib.pyplot as plt

        #make plots inline using jupyter magic
        %matplotlib inline
        from IPython.display import Markdown as md  #enable markdown within code cell
        from IPython.display import display, Math, Latex
        import matplotlib as mpl
```

3.1 2.1

3.1.1 Solution

We have that the standard deviation of the sample data is 3.12

Thus, we have that standard deviation, $\sigma = 3.12$. Variance is calculated as σ^2

Thus, Variance = $3.12^2 = 9.7344$

SE Mean is the standard error of the mean. This value is calculated as $\frac{s}{\sqrt{n}}$, where s is the standard deviation and n is the number of samples.

Thus, the SE mean is $\frac{3.12}{\sqrt{9}} = \frac{3.12}{3} = 1.04$

Answer: Variance = 9.7344, SE Mean = 1.04

3.2 2.3

3.2.1 Solution

We are testing $H_0 : \mu = \mu_0$ versus $H_1 : \mu \neq \mu_0$. Since this is a case of equal or not equal, this is a two-tailed z score case so we have to multply the p-values from the table by 2.

a $Z_0 = 2.25$ Examining the Z-table we see that the value for this Z-score is 0.9878. To get the p-value we subtract this value from 1 so p-value = (1 - 0.9878)*2 = 0.0244 #### Answer: p-value = 0.0244

b $Z_0 = 1.55$ Examining the Z-table we see that the value for this Z-score is 0.9394. To get the p-value we subtract this value from 1 so p-value = (1 - 0.9394)*2 = 0.1212 #### Answer: p-value = 0.1212

c $Z_0 = 2.10$ Examining the Z-table we see that the value for this Z-score is 0.9821. To get the p-value we subtract this value from 1 so p-value = (1 - 0.9821)*2 = 0.0358 #### Answer: p-value = 0.0358

d $Z_0 = 1.95$ Examining the Z-table we see that the value for this Z-score is 0.9744. To get the p-value we subtract this value from 1 so p-value = (1 - 0.9744)*2 = 0.0512 #### Answer: p-value = 0.0512

e $Z_0 = -0.10$ Examining the Z-table we see that the value for this Z-score is 0.9394. To get the p-value we multiply this value by 2 since it is negative so p-value = 0.4602*2 = 0.9204 #### Answer: p-value = 0.9204

3.3 2.4

We are testing $H_0 : \mu = \mu_0$ versus $H_1 : \mu > \mu_0$. This is a right-tailed test and means that we are interested in the area under the normal curve to the right.

3.3.1 a

$Z_0 = 2.45$

Examining the Z-table we see that the value for this Z-score is 0.9929. To get the p-value we subtract this value from 1 so p-value = (1 - 0.9929) = 0.0142 #### Answer: p-value = 0.0142

3.3.2 b

$Z_0 = -1.53$ Examining the Z-table we see that the value for this Z-score is 0.0643. Since this is a right tailed test and the Z-value is negative, the p-value is the score, so #### Answer: p-value = 0.0643

3.3.3 c

$Z_0 = 2.15$ Examining the Z-table we see that the value for this Z-score is 0.9842. To get the p-value we subtract this value from 1 so p-value = (1 - 0.9842) = 0.0158 #### Answer: p-value = 0.0158

3.3.4 d

$Z_0 = 1.95$ Examining the Z-table we see that the value for this Z-score is 0.9744. To get the p-value we subtract this value from 1 so p-value = (1 - 0.9744) = 0.0256 #### Answer: p-value = 0.0256

3.3.5 e

$Z_0 = -0.25$ Examining the Z-table we see that the value for this Z-score is 0.4013. Since this is a right tailed test and the Z-value is negative, the p-value is the score, so #### Answer: p-value = 0.4013

3.4 2.22

3.4.1 a

Null Hypothesis, $H_0 : \mu = 120$
Alternative Hypothesis, $H_1 : \mu > 120$

3.4.2 b

Since the number of samples is less than 30, we use a one tailed t-test to test our hypotheses.

```
In [2]: alpha = 0.01
        days = np.array([108, 124, 124, 106, 115, 138, 163, 159, 134, 139])
        n = days.size
        days_mean = np.mean(days)
        days_std = np.std(days)
        display(md('Mean number of days: {:.2f}'.format(days_mean)))
        t_score_days = scipy.stats.ttest_1samp(days, 120)
        display(md('t-score: {:.2f}'.format(t_score_days[0])))
        display(md('Degrees of Freedom: {}'.format(n-1)))
```

Mean number of days: 131.00
t-score: 1.78
Degrees of Freedom: 9
Examining the table for a one-tailed t-test with $\alpha = 0.01$ and df of 9, we see that the value is 2.821.
This means that the critical t-value for this test is 2.821.
Our t-score of 1.78 < 2.821, so the result is not statistically significant.

3.5 c

The python package for calculating t-score also provides the p-value. The p-value provided is for the two-tailed test, so we divide by 2 to get the one tailed p value. p-value is shown below.

```
In [3]: p_value = t_score_days[1]/2
        display(md('p-value: {:.4f}'.format(p_value)))
```

p-value: 0.0544

3.6 d

To construct a 99% confidence interval, we know that 99% of t_0 falls between the rejection limits of [-2.821,2.821].

```
In [4]: t_critical = 2.821
        lower = 131 - (days_std/np.sqrt(n))*t_critical
        upper = 131 + (days_std/np.sqrt(n))*t_critical
        display(md("99% Confidence interval: [{:.3f},{:.3f}]".format(lower,upper)))
```

99% Confidence interval: [114.459,147.541]

3.7 2.27

$\sigma_1 = \sigma_2 = 1.0$
$n_1 = 10$
$n_2 = 12$
$\bar{y_1} = 162.5$
$\bar{y_1} = 155.0$
For this test, we need plastic 1 to have a breaking strength greater than plastic 2 + 10. This makes for our hypothese as the following
$H_0 : bs_1 <= bs_2 + 10$ $H_a : bs_1 > bs_2 + 10$
To test these hypotheses, we will use a 2 sample one-tailed t-test. Since we are using two different types of plastic, we can assume that the means are independent.
The two sample t-test is given by the equation:

$$t_0 = \frac{\bar{y_1} - \bar{y_2}}{S_p\sqrt{(\frac{1}{n_1} + \frac{1}{n_2})}}$$

```
In [5]: n1 = 10
        n2 = 12
        s1 = 1
        s2 = 1
        y1_bar = 162.5
        y2_bar = 155.0


        s_var_pooled = ((n1-1)*s1**2 + (n2-1)*s2**2)/(n1+n2-2)
        sp = np.sqrt(s_var_pooled)


        t0 = (y1_bar - y2_bar)/(sp*np.sqrt((1/n1) + (1/n2)))
        display(md(r'$t_0 = {:.3f}$'.format(t0)))
```

$t_0 = 17.516$
To find the 99% confidence interval, we need to find the interval such that

$$\bar{y_1} - \bar{y_2} - t_{crit} * S_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}} \leq \mu_1 - \mu_2 \leq \bar{y_1} - \bar{y_2} + t_{crit} * S_p\sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

Where $t_{crit} = t_{\alpha, n_1+n_2-2}$
For our values, $t_{crit} = t_{0.01,20} = 2.528$

```
In [6]: t_crit = 2.528

        sample_mean_diff = y1_bar - y2_bar

        lower = sample_mean_diff - t_crit*sp*np.sqrt((1/n1)+(1/n2))
        upper = sample_mean_diff + t_crit*sp*np.sqrt((1/n1)+(1/n2))

        display(md('99% confidence interval for sample mean difference: [{:.3f}, {:.3f}]'.format
```

99% confidence interval for sample mean difference: [6.418, 8.582]

Answer: Given that 10 is not in the 99% confidence interval, the company should not use plastic 1.

3.8 2.31

```
In [7]: temp95 = np.array([11.176, 7.089, 8.097, 11.739, 11.291, 10.759, 6.467, 8.315])
        temp100 = np.array([5.263, 6.748, 7.461, 7.015, 8.133, 7.418, 3.772, 8.963])
```

3.8.1 a

Let l_1 be the mean thickness of the 95 celcius list and l_2 be the mean of the 100 celcius list.
$H_0 : l_1 >= l_2$
$H_a: l_1 < l_2$
To test this hypothesis, we use a one-tailed, two sample t-test.

```
In [8]: alpha = 0.05
        n1 = temp100.size
        n2 = temp95.size
        df = n1+n2-2
        t_crit = 1.761
        t_score = stats.ttest_ind(temp100,temp95)
        display(md("t-score: {:.3f}".format(t_score[0])))
        display(md("Degrees of Freedom: {}".format(df)))
        display(md(r"Critical t-value: $t(0.05,14)$ {:.3f}".format(t_crit)))
```

t-score: -2.675
Degrees of Freedom: 14
Critical t-value: $t(0.05, 14)$ 1.761
Since $|-2.675| > 1.761$ the observed difference is significant.

3.8.2 b

The two-sided p-value is calculated as part of the t-score from above, so to get the one-sided p-value we divide by 2.

```
In [9]: p_value = t_score[1]/2
        display(md("p-value: {:.4f}".format(p_value)))
```

p-value: 0.0091

3.8.3 c

To calculate the 95% confidence interval for the difference in means we do the following

$$\bar{y_1} - \bar{y_2} - t_{crit} * S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}} \leq \mu_1 - \mu_2 \leq \bar{y_1} - \bar{y_2} + t_{crit} * S_p \sqrt{\frac{1}{n_1} + \frac{1}{n_2}}$$

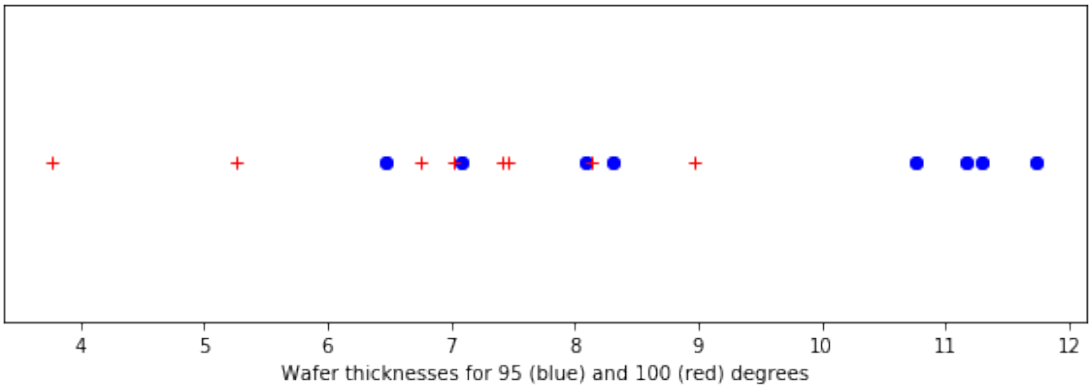
Where $t_{crit} = t_{\alpha, n_1+n_2-2}$

```
In [10]: y1_bar = temp100.mean()
         y2_bar = temp95.mean()
         s1 = temp100.std()
         s2 = temp95.std()
         s_var_pooled = ((n1-1)*s1**2 + (n2-1)*s2**2)/(n1+n2-2)
         sp = np.sqrt(s_var_pooled)

         lower = sample_mean_diff - t_crit*sp*np.sqrt((1/n1)+(1/n2))
         upper = sample_mean_diff + t_crit*sp*np.sqrt((1/n1)+(1/n2))

         display(md('#### Answer: 95% confidence interval for sample mean difference: [{:.3f}, {
```

Answer: 95% confidence interval for sample mean difference: [5.948, 9.052] The practical interpretation of this interval is that the 95% of the mean thicknesses of the wafers baked at the higher temperatures will fall be somewhere in the above interval less than the wafers baked at the lower temperature. We know that it is less because the t-score was negative.

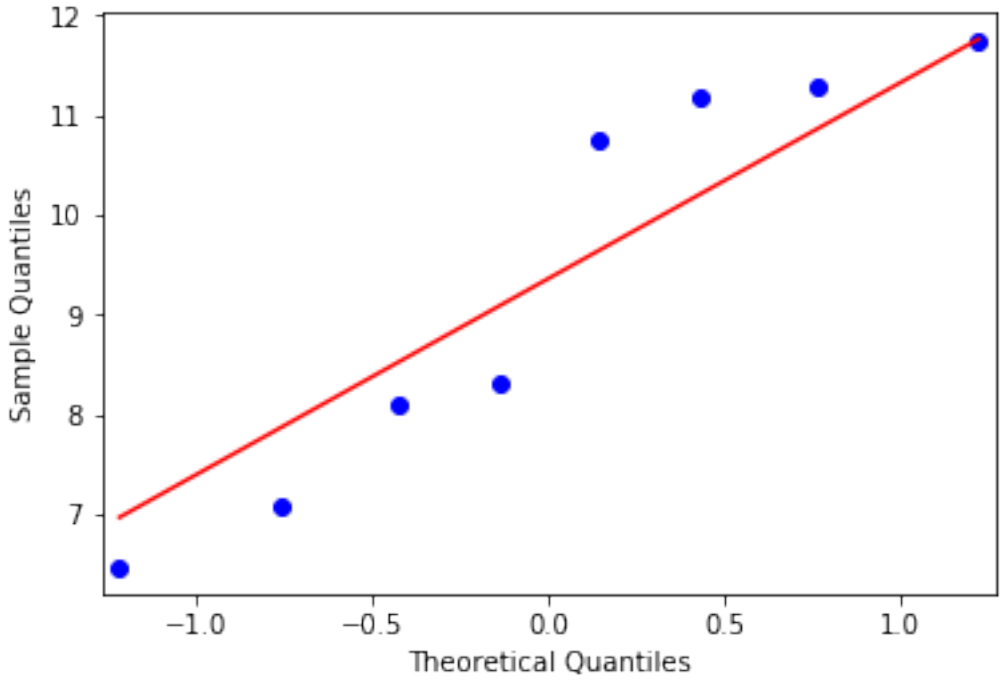
3.8.4 d

```
In [11]: fig = plt.figure(figsize=(10,3))
         ax = plt.subplot()
         y = np.zeros((8,2))
         ax.plot(temp95,y,'bo')
         ax.plot(temp100,y,'r+')

         plt.yticks([])
         plt.xlabel("Wafer thicknesses for 95 (blue) and 100 (red) degrees")

         plt.show()
```

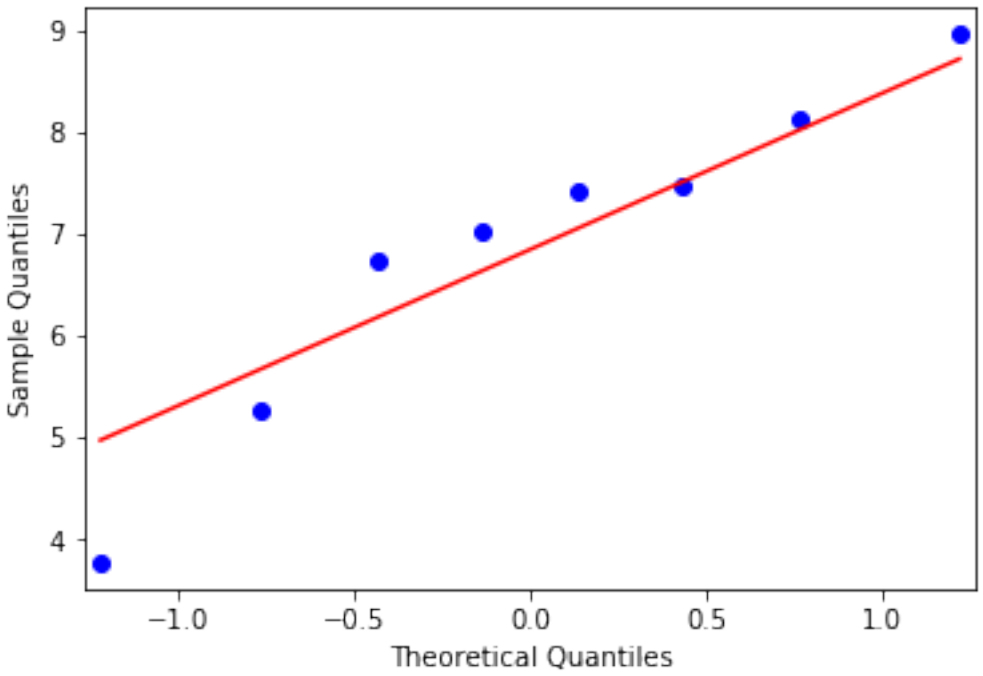


3.9 e

```
In [12]: from statsmodels.graphics.gofplots import qqplot
         qqplot(temp95,line='s')
         plt.show()
```




```
In [13]: qqplot(temp100,line='s')
         plt.show()
```



Answer: The top plot above shows the distribution of the 95 degree values. The 100 degree values are shown below it. The 100 degree values appear to fit the line slightly better and so are more normal than the 95 degree values but both approximately fit the line.

3.9.1 f

Power = $1 - \beta$ = P(reject H_0 | H_0 is false)

To find the power, we use a package from statsmodels called solve_power that provides power given the other 3 parameters. We assume alpha = 0.05

```
In [14]: from statsmodels.stats.power import TTestIndPower
         analysis = TTestIndPower()
         effect_size = 2.5/sp
         alpha = 0.05
         power = analysis.power(effect_size=effect_size, nobs1=n1, alpha=alpha, ratio=1)
         display(md('#### Answer: Power = {:.4f}'.format(power)))
```

Answer: Power = 0.7513

3.9.2 g

To calculate the sample size we use the solve_power function from statsmodels.stats.power. This function returns the parameter for which no value is provided given the thre other parameters listed below. We assume alpha = 0.05

```
In [15]: effect_size = 1.5/sp #Normalized difference in means
         alpha = 0.05
         power = 0.9

         sample_size_1 = analysis.solve_power(effect_size=effect_size, nobs1=None, alpha=alpha,

         sample_size = sample_size_1 * 2 # solve_power gives us n1 and since n1 = n2 we double t

         display(md('#### Answer: Necessary Sample Size = {}'.format(sample_size)))
```

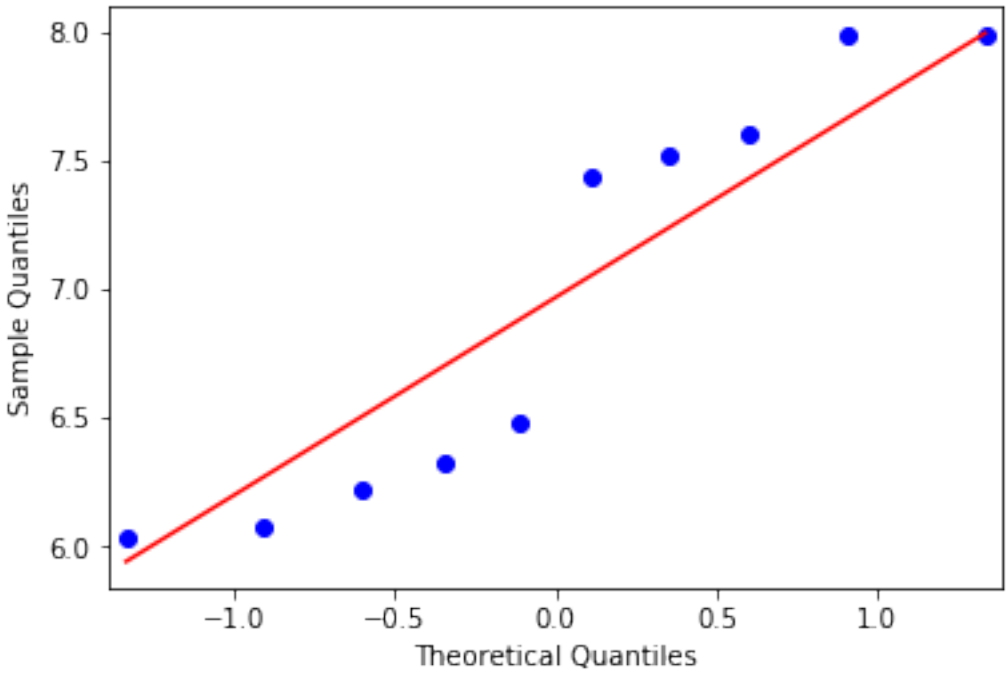
Answer: Necessary Sample Size = 60.004952074251754

3.10 2.35

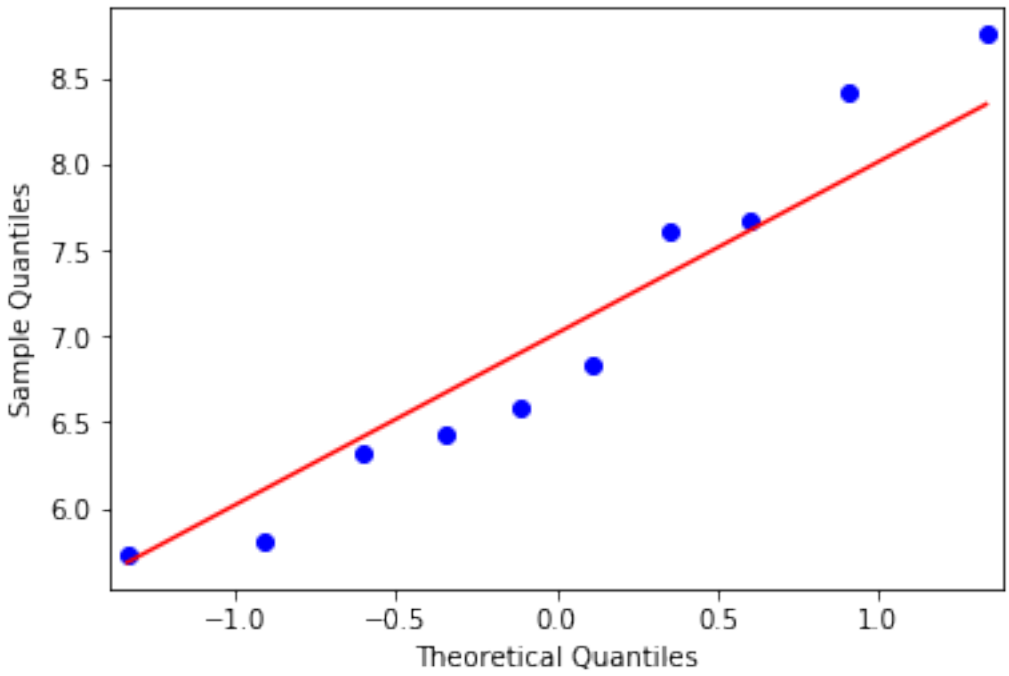
```
In [16]: bo_1 = np.array([6.08, 6.22, 7.99, 7.44, 6.48, 7.99, 6.32, 7.6, 6.03, 7.52])
         bo_2 = np.array([5.73, 5.80, 8.42, 6.84, 6.43, 8.76, 6.32, 7.62, 6.59, 7.67])
         bo_diff = bo_1 - bo_2
```

3.10.1 a

```
In [17]: qqplot(bo_1,line='s')
         plt.show()
```




```
In [18]: qqplot(bo_2, line='s')
         plt.show()
```



Answer: Based on the above plots, it appears that the birth order 2 list is more normally distributed than the birth order 1. However, the birth order 1 is close enough to the line that the normal distribution assumption is okay.

3.10.2 b (and c)

We need to find a 95% percent confidence interval for the difference in means between pairs. To do this we take the two lists and subtract the second from the first to find the difference vector. In this case we are interested in the following hypotheses:

H_0: Mean score does not depend on birth order

H_a: Mean score does depend on birth order

Mathematically we can express this by letting d represent the mean difference in scores between birth orders. This makes the hypotheses the following:

$H_0 : d = 0$

$H_0 : d \neq 0$

In this case, we use a one sample two-tailed t-test, with $\alpha = 0.05$ and 9 degrees of freedom. Examining the t-score table, this make the critical t-score 2.262.

```
In [19]: mean_diff = bo_diff.mean()
         std_diff = bo_diff.std()
         n = bo_diff.size

         tscore_diff = (mean_diff - 0)/(std_diff/np.sqrt(n))
         display(md("T-score: {:.2f}".format(tscore_diff)))
```

```
lower = mean_diff + tscore_diff*(std_diff/np.sqrt(n))
upper = mean_diff - tscore_diff*(std_diff/np.sqrt(n))

display(md("95% confidence interval for difference in scores: [{:.3f}, {:.3f}]".format(
```

T-score: -0.39
95% confidence interval for difference in scores: [-0.102, 0.000]

3.10.3 Answer: Given that 0 is in the 95% percent confidence interval and that the absolute value of the T-score of 0.39 < 2.262, we can say that there is no statistically significant difference in the scores based on birth order. Thus, we fail to reject the null hypothesis.

3.11 2.47

The equation for the paired t-test is given by the following, where $d = y_i - x_i$ for each pair.

$$t_p = \frac{\bar{d}}{\frac{s_d}{\sqrt{n}}}$$

Where s_d is the standard deviation of the differences and $\bar{d}$ is the sample mean of the differences.

On the other hand, the two sample t test equation is given by

$$t_i = \frac{\bar{y_1} - \bar{y_2}}{S_p\sqrt{(\frac{1}{n_1} + \frac{1}{n_2})}}$$

3.11.1 Solution

We want $t_p > t_i$

$$\frac{\bar{d}}{\frac{s_d}{\sqrt{n}}} > \frac{\bar{y_1} - \bar{y_2}}{S_p\sqrt{(\frac{1}{n_1} + \frac{1}{n_2})}}$$

We know that $d = y_1 - y_2$ so $\bar{d} = \bar{y_1} - \bar{y_2}$
We then have

$$\frac{\bar{y_1} - \bar{y_2}}{\frac{s_d}{\sqrt{n}}} > \frac{\bar{y_1} - \bar{y_2}}{S_p\sqrt{(\frac{1}{n_1} + \frac{1}{n_2})}}$$

We know that s_p is an approximation of s_d so we assume that $s_d = s_p \neq 0$. (For ease of calculation and to show the point we assume that $\bar{y_1} - \bar{y_2} \neq 0$). This yields

$$\frac{\sqrt{n}}{1} > \frac{1}{\sqrt{(\frac{1}{n_1} + \frac{1}{n_2})}}$$

From this, for given sample sizes n, n_1, n_2, then the paired t-score will be larger than the pooled t-score.

3.12 2.59

Power is the calculation of how likely the test is to get the true positives P(reject H_0 | H_0 false). Increasing this value is ideal, but along with increasing power comes the risk of increasing false positives as well. Thus, to get an "adequate" power level, we have to decide what our tolerance for false positives is. Generally, the common adequate power level is 80% or 0.8. The higher the power, the less likely we are to make a type II error. One of the main issues to think about when determining power level is the level of significance. The level of significance affects the level of power we are concerned about. A lower significance means we need a higher power test.

3.13 2.60

In the early stages of experimenting, it is more important to avoid the type I error where we reject the null when it is true. This is because if we are experimenting in a new area, committing a type I error and rejecting the null hypothesis may lead to cascading issues down the road because of assumptions made based on the type I error.

In [ ]: