Test a Perceptual Phenomenon

November 24, 2017

(by Eric Sean Turner)

0.1 Variables

The independent variable is the dataset for the congruent task group (C), while the dependent variable is the dataset for the incongruent task group (I).

0.2 Hypotheses & Proposed Test

The null hypothesis (H_0) is that the mean of the congruent task group (C) is the same as the mean of the incongruent task (I) group.

$$H_0 : C = I$$

The alternate hypothesis (H_1) is that there is a difference between the means of the congruent task group (C) and the mean of the incongruent task group (I).

$$H_1 : C \neq I$$

0.2.1 Intuitive prediction

Intuitively, and based on my own experience completing the stroop task, I believe that there will be a significant difference between the sample means of these datasets. This is because it was notably more difficult to complete the incongruent task than it was the congruent one, and this effect should show up in the data.

0.2.2 Test Selection

As a statistical test to evaluate these hypotheses, I propose conducting a **t-test**. The t-test was chosen over other tests for a few reasons: (1) the population standard deviation is not known, (2) the data does not comprise a normal distribution, and (3) there are relatively few data points (n < 30) in each sample. And since the same person was used for each of the tasks, the data is best tested using a **paired t-test**.

0.3 Analysis

The stroop task datasets were analyzed, and their descriptive statistics laid out below:

```
In [1]: import numpy as np
        from scipy import stats

        x = [12.079,16.791,9.564,8.63,14.669,12.238,14.692,8.987,9.401,14.48,22.328,15.298,15.07
        y = [19.278,18.741,21.214,15.687,22.803,20.878,24.572,17.394,20.762,26.282,24.524,18.644

        mean_x = np.mean(x)
        print('Mean of x =', mean_x)

        mean_y = np.mean(y)
        print('Mean of y =', mean_y)

        median_x = np.median(x)
        print('Median of x =', median_x)

        median_y = np.median(y)
        print('Median of y =', median_y)

        variance_x = np.var(x)
        print('Variance of x =', variance_x)

        variance_y = np.var(y)
        print('Variance of y =', variance_y)
```

```
Mean of x = 14.051125
Mean of y = 22.0159166667
Median of x = 14.3565
Median of y = 21.0175
Variance of x = 12.1411528594
Variance of y = 22.0529338264
```

Based on the results of the analysis above, the descriptive statistics of the datasets are as follows:

- x = Congruent data set
- y = Incongruent data set
- N1 = 24 (Number of data points in x)
- N2 = 24 (Number of data points in y)

Measures of Central Tendency

- x = 14.051125 (Mean of x)
- y = 22.01591667 (Mean of y)
- Mode of x = none (No numbers are repeated in the data set)

- Mode of y = none (No numbers are repeated in the data set)
- Median of x = 14.3565
- Median of y = 21.0175

Measures of Variability

- Sx2 = 12.66902907 (Variance of x)
- Sy2 = 23.01175704 (Variance of y)

0.4 Visualizations

Below we have a boxplot depicting the distributions and outliers of each test group. The x-axis shows the congruent task group (1) and the incongruent task group (2). The y-axis shows the time to complete the task in seconds.

0.4.1 Observations

From the boxplot we can see that it takes more time to complete the incongruent task than the congruent task. Several outliers can be seen in the incongruent task group.

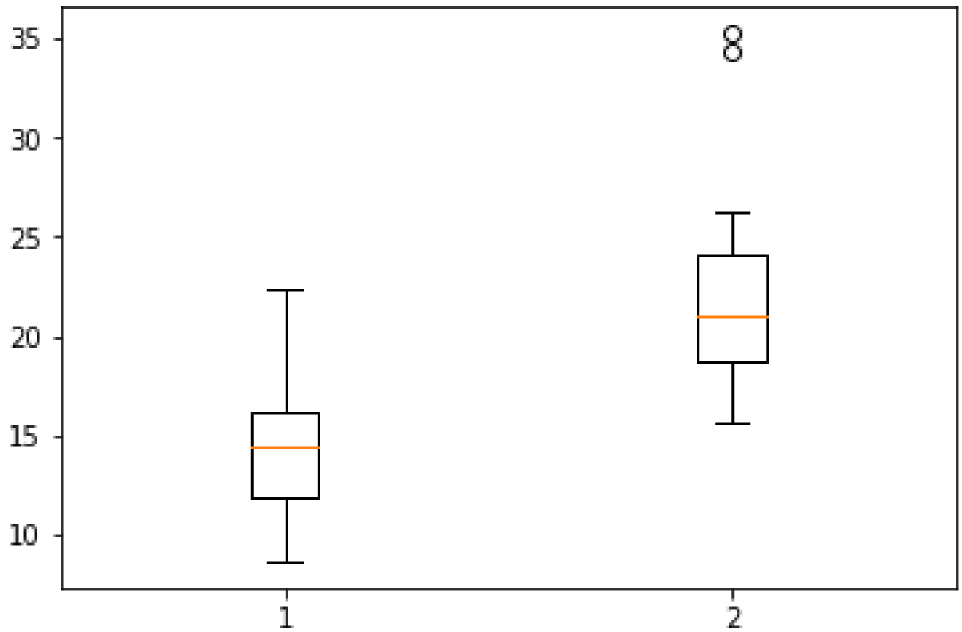
```
In [2]: import matplotlib.pyplot as plt
        import numpy as np

        x = [12.079,16.791,9.564,8.63,14.669,12.238,14.692,8.987,9.401,14.48,22.328,15.298,15.07
        y = [19.278,18.741,21.214,15.687,22.803,20.878,24.572,17.394,20.762,26.282,24.524,18.644

        data_to_plot = [x, y]

        plt.figure()
        plt.boxplot(data_to_plot)

        plt.show()
```



0.5 Statistical Test

A paired t-test was performed to determine whether a statistically significant difference exists between the data.

The formula of the t-test used was:

$$t = \frac{\sum D}{\sqrt{\frac{n \sum D^2 - (\sum D)^2}{n-1}}}$$

The formula for degrees of freedom used was:

$$df = n - 1$$

In [3]:
```
## Perform the statistical test here

import numpy as np
import math
from scipy.stats import t

x = [12.079,16.791,9.564,8.63,14.669,12.238,14.692,8.987,9.401,14.48,22.328,15.298,15.07
y = [19.278,18.741,21.214,15.687,22.803,20.878,24.572,17.394,20.762,26.282,24.524,18.644

n = len(x)

x_mean = np.mean(x)
y_mean = np.mean(y)
```

```
sum_of_differences = 0
sum_of_squared_differences = 0

for i in range(n):
    x_value = x[i]
    y_value = y[i]
    difference = x_value - y_value
    squared_difference = pow(difference, 2)
    sum_of_differences += difference
    sum_of_squared_differences += squared_difference

print('sum_of_differences =', sum_of_differences)
print('sum_of_squared_differences =', sum_of_squared_differences)

sx2 = np.var(x)
sy2 = np.var(y)

df = n - 1
print('df = ', df)

t_stat = sum_of_differences / math.sqrt( ( n * sum_of_squared_differences - pow(sum_of_d
print('t-statistic = ', t_stat)
```

```
sum_of_differences = -191.15499999999994
sum_of_squared_differences = 2066.8401909999993
df =  23
t-statistic =  -8.020706944109953
```

0.6 Results of the Statistical Test

Running the t-test gives us a t-statistic = -8.0207. Referencing the t-table with an alpha value of 0.05 (0.025 for a two-tailed test) and a df of 23 we can see that our critical value t-crit is 2.069. Since the absolute value of our t-statistic is greater than our critical value t-crit, we reject the null hypothesis and accept the alternative hypothesis (H1) which states that there is a difference between the means of the congruent task group (C) and the mean of the incongruent task group (I).

0.7 Conclusion

The results of the test confirmed my original suspicion that the null hypothesis was false – there is a statistically significant difference in the time to complete the stroop task depending on whether you take the congruent task or the incongruent task.

0.8 Hypothesis on Possible Cause of the Stroop Effect

There are a few different theories on the possible cause of the stroop effect. One theory, the speed of processing theory holds that perhaps it takes longer for colors to be uttered than it does for words to be read. Perhaps the reason for this is that when reading colors, first the brain must use a color-recognition region to first detect the color, and second, send that information to the language processing region to actually say the word of the color – essentially two trips to separate regions of the brain. If the task is merely to read the word, perhaps the brain only needs to use the language processing region and that accounts for a faster response time – less trips to other regions in the brain!

0.9 External Resources Used

- **F-distribution table:** http://www.socr.ucla.edu/applets.dir/f_table.html#FTable0.05
- **T-table:** http://www.statisticshowto.com/tables/t-distribution-table/
- **Python sqrt():** https://stackoverflow.com/questions/9595135/how-to-calc-square-root-in-python
- **Python pow():** https://www.tutorialspoint.com/python/number_pow.htm
- **T-Test Formulas:** https://www.youtube.com/watch?v=avixq-YsXv0
- **Paired T-Test:** https://www.youtube.com/watch?v=BPbHujvA9UU
- **Jupyter Plotting:** https://www.youtube.com/watch?v=Hr4yh1_4GlQ
- **Boxplot Code:** https://matplotlib.org/examples/pylab_examples/boxplot_demo.html