

Test a Perceptual Phenomenon

May 13, 2018

0.0.1 Analyzing the Stroop Effect

Perform the analysis in the space below. Remember to follow [the instructions](#) and review the [project rubric](#) before submitting. Once you've completed the analysis and write-up, download this file as a PDF or HTML file, upload that PDF/HTML into the workspace here (click on the orange Jupyter icon in the upper left then Upload), then use the Submit Project button at the bottom of this page. This will create a zip file containing both this .ipynb doc and the PDF/HTML doc that will be submitted for your project.

- (1) What is the independent variable? What is the dependent variable?

Independent variable: Condition (congruent/incongruent)

Dependent variable: Time to name ink colours

- (2) What is an appropriate set of hypotheses for this task? Specify your null and alternative hypotheses, and clearly define any notation used. Justify your choices.

The null hypothesis should be that the mean time for colour recognition for congruent words is equal to or greater than the mean time for incongruent words. The alternative hypothesis should be that the congruent words mean is less than the incongruent words mean.

To achieve this, We can use a two-sided paired student T-test to verify. This is because: one, we need to address the uncertainty in sample standard error resulted from the unknown population standard deviation; two, we are comparing the means of two groups that are dependent; three, the same subject is involved under both conditions.

Below is the hypothesis to test:

$H_0: \mu_C \geq \mu_I$

$H_A: \mu_C < \mu_I$

where μ is a population mean, the subscript "C" represents the congruent words condition, and the subscript "I" represents the incongruent words condition.

- (3) Report some descriptive statistics regarding this dataset. Include at least one measure of central tendency and at least one measure of variability. The name of the data file is 'stroop-data.csv'.

```
In [2]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.patches as mpatches
```

```

from scipy.stats import sem
from scipy.stats import t
from scipy.stats import ttest_rel
%matplotlib inline

```

```

In [3]: # Perform the analysis here
df = pd.read_csv('stroopdata.csv')
df.describe()

```

```

Out[3]:

```

	Congruent	Incongruent
count	24.000000	24.000000
mean	14.051125	22.015917
std	3.559358	4.797057
min	8.630000	15.687000
25%	11.895250	18.716750
50%	14.356500	21.017500
75%	16.200750	24.051500
max	22.328000	35.255000

Central Tendency: Considering all data on the congruent and incongruent groups, we can find mean and standard deviation of both groups as you can see above.

Measure of Variability: To get a better understanding of the general variability of Congruent and Incongruent data, we can determine 1st , 2nd and 3rd quartile as shown above.

- (4) Provide one or two visualizations that show the distribution of the sample data. Write one or two sentences noting what you observe about the plot or plots.

```

In [4]: # Build the visualizations here
def visualize_hist(data1, data2, labels, legend_title, legends, savefig=True, show_mean=
    """
    This function visualizes the two datasets
    Params
    data1: 1D numpy array of first dataset
    data2: 1D numpy array of second dataset
    labels: list of x label and y label
    legend_title: Title of legends
    legends: list of 2 legends
    savefig: if true save figure as png file
    """

    # Visualize data
    plt.figure(0)
    # labels
    plt.xlabel(labels[0])
    plt.ylabel(labels[1])
    # Data to histogram
    plt.hist(incong, color='orange')
    plt.hist(cong, color='red')
    # Show mean
    if show_mean:

```

```

plt.axvline(cong.mean(), color='blue', linestyle='dashed', linewidth=2)
plt.axvline(incong.mean(), color='cyan', linestyle='dashed', linewidth=2)
# Legends
orange_patch = mpatches.Patch(color='orange', label=legends[1])
red_patch = mpatches.Patch(color='red', label=legends[0])
plt.legend(title=legend_title, handles=[red_patch, orange_patch])
# Save Visualization
if savefig:
    plt.savefig('visualize.png', bbox_inches='tight')
# Show
plt.show()

```

```

In [5]: def visualize_box(data1, data2, labels, savefig=True, show_mean=True):
        fig = plt.figure(0)
        ax = fig.add_subplot(111)
        bp = ax.boxplot([data1, data2])
        ax.set_xticklabels(labels)
        if savefig:
            plt.savefig('box_visualize.png', bbox_inches='tight')
plt.show()

```

```

In [6]: cong = np.asarray(df['Congruent'])
        incong = np.asarray(df['Incongruent'])

```

```

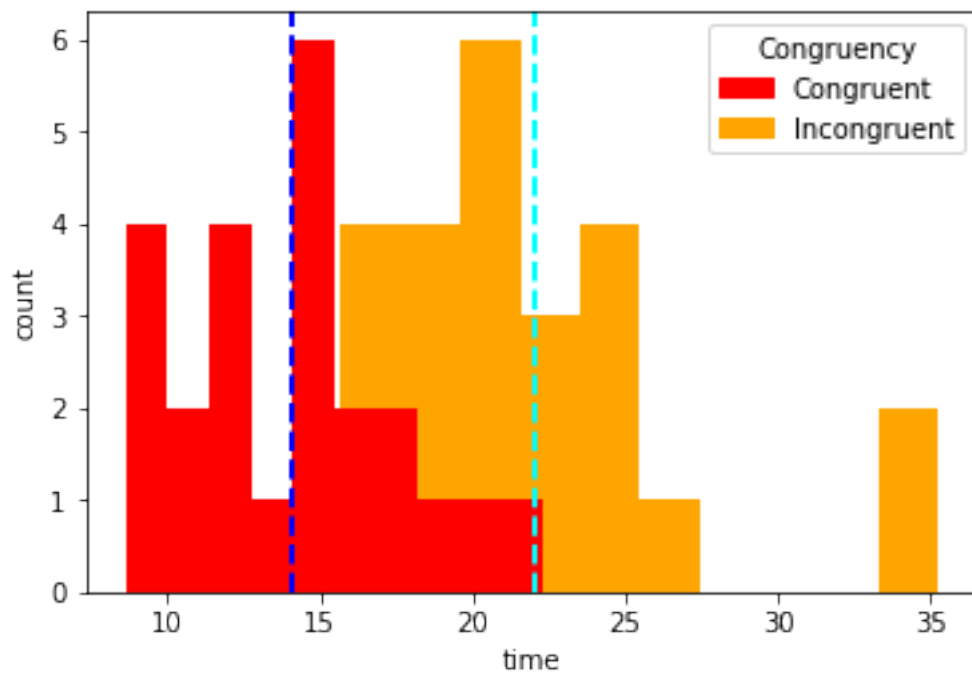
# Visualize plot

```

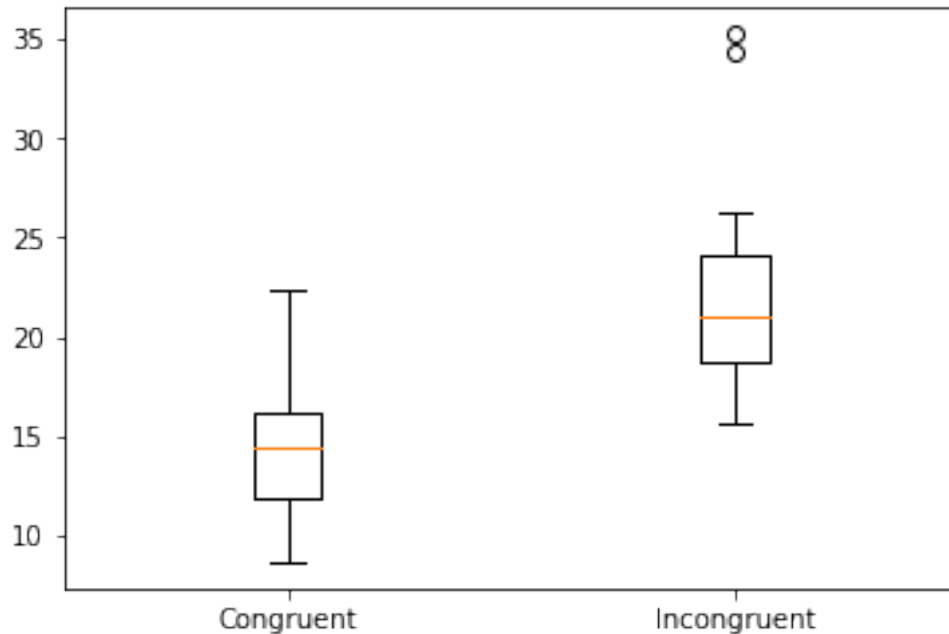
```

visualize_hist(incong, cong, ['time', 'count'], 'Congruency', ['Congruent', 'Incongruent'])

```



```
In [7]: visualize_box(cong, incong, ['Congruent', 'Incongruent'])
```



The histogram shows that the times on the incongruent sample are larger than on the congruent sample. It also shows that both groups have evident outliers.

The boxplot indicates that the two groups have significant difference in median times, and the two groups also have different ranges - with the Incongruent words group presenting much longer times. The box plot is simply a quick top down view of the data from the congruent and incongruent tests.

- (5) Now, perform the statistical test and report your results. What is your confidence level or Type I error associated with your test? What is your conclusion regarding the hypotheses you set up? Did the results match up with your expectations? **Hint:** Think about what is being measured on each individual, and what statistic best captures how an individual reacts in each environment.

```
In [8]: # Perform the statistical test here
```

```
def statistical_test(data1, data2):
    """
    Performs paired t-test on related samples
    params:
    data1: 1D sample 1 numpy array
    data2: 1D sample 2 numpy array
    """
```

```

mean_diff = np.mean(data1) - np.mean(data2)
stand_error = sem(data1 + data2) #np.sqrt(np.std(data1, ddof=1)+np.std(data1, ddof=1))
df = data1.shape[0] - 1
t_critical = t.isf([0.05], [[df]])[0][0]
moe = t_critical * stand_error

print(' t-critical value: ', t_critical)
print(' Mean Difference: ', mean_diff)
print(' Standar Error = %6.3f df = %d' % (stand_error, df))
print(' t-statistic = %6.3f pvalue = %f' % ttest_rel(data1, data2))
#print(' t-statistic = %6.3f pvalue = %f' % ttest_rel(data1, data2))
print(' CI: (%6.4f, %6.4f)' % (mean_diff - moe, mean_diff + moe))

cong = np.asarray(df['Congruent'])
incong = np.asarray(df['Incongruent'])

statistical_test(cong, incong)

t-critical value:  1.71387152775
Mean Difference:  -7.96479166667
Standar Error =  1.410 df = 23
t-statistic = -8.021 pvalue = 0.000000
CI: (-10.3809, -5.5487)

```

Since the p-value is less than 0.05, we reject the null hypothesis and conclude that the difference between congruence and incongruence group time difference is statistically significant, namely, the stroop effect is present. This is in line with my expectation.

Based on the confidence intervals, we're 95% confident that the true difference between the congruence and incongruence group average times is between -10.8810 and -5.0486.

- (6) Optional: What do you think is responsible for the effects observed? Can you think of an alternative or similar task that would result in a similar effect? Some research about the problem will be helpful for thinking about these two questions!

My hypothesis for the effects observed is that the brain dominantly focuses on reading the word rather than recognizing a colour when the eyes are presented with a coloured word. To recognize a colour, one has to override the brain's natural tendency of reading the word. This override takes time and is likely not always successful, which means re-analyzing a word after the error is recognized, which costs more time.

Numerical/Physical size Stroop tasks, where numerical values and physical size are the factors that contribute to congruency/incongruency, results in a similar effect. It takes longer to recognize the number and physical size (two separate tasks) of small numbers that have a large physical size and large numbers that have a small physical size.