Project : Stroop Effect

ASHISH SHARMA

2/5/2018

Objective: Investigate a classic phenomenon from experimental psychology called the Stroop Effect.

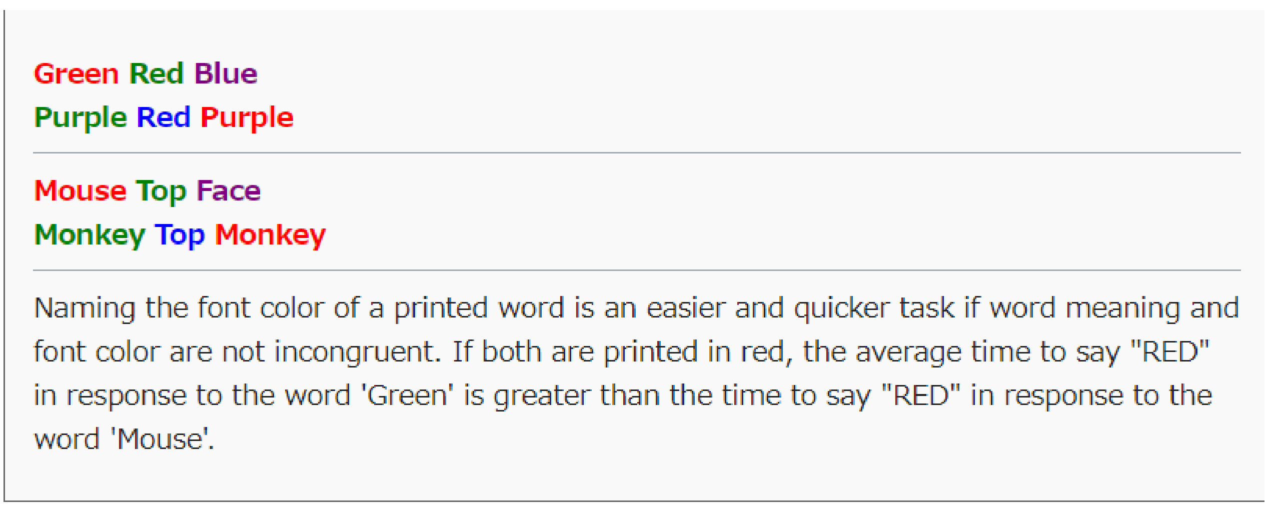
Background information

From Wikipedia, the free encyclopedia

In psychology, the Stroop effect is a demonstration of interference in the reaction time of a task. When the name of a color (e.g., "blue", "green", or "red") is printed in a color that is not denoted by the name (e.g., the word "red" printed in blue ink instead of red ink), naming the color of the word takes longer and is more prone to errors than when the color of the ink matches the name of the color

In [1]:
```
from IPython.display import Image
Image(filename='stroop_effect_01.png')
```

Out[1]:



In [2]:
```
from IPython.display import Image
Image(filename='stroop_effect_02.png')
```

Out[2]:

In a Stroop task, participants are presented with a list of words, with each word displayed in a color of ink. The participant's task is to say out loud the color of the ink in which the word is printed. The task has **two conditions: a congruent words condition, and an incongruent words condition. In the congruent words condition**, the words being displayed are color words whose names match the colors in which they are printed: for example **RED**, **BLUE**. In the incongruent words condition, the words displayed are color words whose names do not match the colors in which they are printed: for example **PURPLE**, **ORANGE**. In each case, we measure the time it takes to name the ink colors in equally-sized lists. Each participant will go through and record a time from each condition.

Questions for Investigation

1. What is our independent variable? What is our dependent variable?

Independent variable is the variable being manipulated in the experiment i.e congruent/incongruent words condition.
Dependent variable is the reaction time it takes to recognize/name the ink/font colors of the congruent/incongruent words.

2. What is an appropriate set of hypotheses for this task? What kind of statistical test do you expect to perform? Justify your choices.

Null Hypothesis, H0 is the hypothesis that there is no significant difference between specified populations, any observed difference being due to sampling or experimental error.
There is no significant difference in the time between congruent and incongruent scores
i.e the mismatch of color to word will have no effect or decrease time to recognize and say the color.
H0: μi ≤ μc (μi - population mean of incongruent values, μc - population mean of congruent values)
Alternate Hypothesis, H1 - There would be significant difference in the time between congruent and incongruent scores
i.e the mismatch of color to word will increase time to recognize and say the color
H1: μi > μc (μi - population mean of incongruent values, μc - population mean of congruent values)

Statistical test choice :

- **95% confidence interval**
- **Paired one tail t-test -> with two tests per participant this test show if the mean of incongruent words is statistically significantly different from the congruent words at an alpha of 0.05.**

Justification:
I 'll be using a t-test instead of a z-test because the sample size is small (n =24 is much less than 30) and the population standard deviation and other population parameters are not known.
The t-test will be a one tailed t-test i.e. my directional alternative hypothesis is that participant's incongruent sample mean will be larger than the participant's congruent sample mean

A paired t-test (or dependent sample test), will be used because the data set is of one group of participants tested twice under different conditions (word/colour congruency). This will also facilitate either rejecting or accepting the null hypothesis.

3. Report some descriptive statistics regarding this dataset. Include at least one measure of central tendency and at least one measure of variability.

In [3]:
```
# Import Libraries
from scipy import stats
# Render our plots inline
%matplotlib inline
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
```

In [4]:
```
stroopDF = pd.read_csv("stroopdata.csv")
print(stroopDF)
```

```
    Congruent  Incongruent
0      12.079       19.278
1      16.791       18.741
2       9.564       21.214
3       8.630       15.687
4      14.669       22.803
5      12.238       20.878
6      14.692       24.572
7       8.987       17.394
8       9.401       20.762
9      14.480       26.282
10     22.328       24.524
11     15.298       18.644
12     15.073       17.510
13     16.929       20.330
14     18.200       35.255
15     12.130       22.158
16     18.495       25.139
17     10.639       20.429
18     11.344       17.425
19     12.369       34.288
20     12.944       23.894
21     14.233       17.960
```

```
22      19.710        22.058
23      16.004        21.157
```

In [5]:
```
# descriptive stats
stroopDF.describe()
```

Out[5]:

	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

Sample size = 24
Mean: $xbar = \Sigma{x}/n$ Congruent: 14.05, Incogruent: 22.02
Median: as the data seems slightly positively skewed, median is a better representation of central tendency
Congruent: 14.36, Incongruent: 21.02
Sample std. deviation: $\sigma = \Sigma{(x - xbar)^2}/n$ Congruent: 3.56, Incongruent: 4.80

In [6]:
```
# median / 50% values
congruent = stroopDF['Congruent']
incongruent = stroopDF['Incongruent']
congruent.median(), incongruent.median()
```

Out[6]:
```
(14.3565, 21.0175)
```

In [7]:
```
# Lets create a Series and DataFrame for the difference between the two co
nditions
differenceSeries = incongruent - congruent
differenceDF = pd.DataFrame({"Difference":differenceSeries})
differenceDF.describe()
```

Out[7]:

	Difference
count	24.000000
mean	7.964792
std	4.864827
min	1.950000
25%	3.645500
50%	7.666500
75%	10.258500

max	21.919000

In [8]:
```
# Perform calculations for a dependent samples t test
ttestResults = stats.ttest_rel(incongruent, congruent)
tstatistic = ttestResults[0]
pvalue = ttestResults[1]
print "t-statistic = " + '%.2f' % tstatistic
print "p-value = " + '%.8f' % pvalue
```

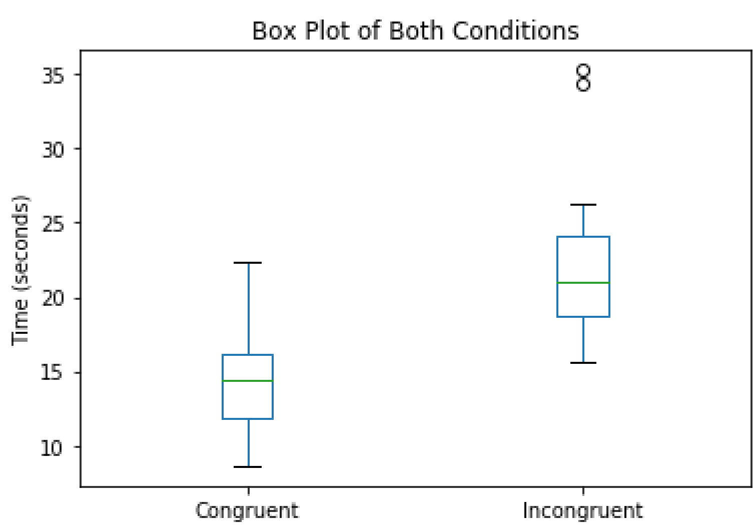
```
t-statistic = 8.02
p-value = 0.00000004
```

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.

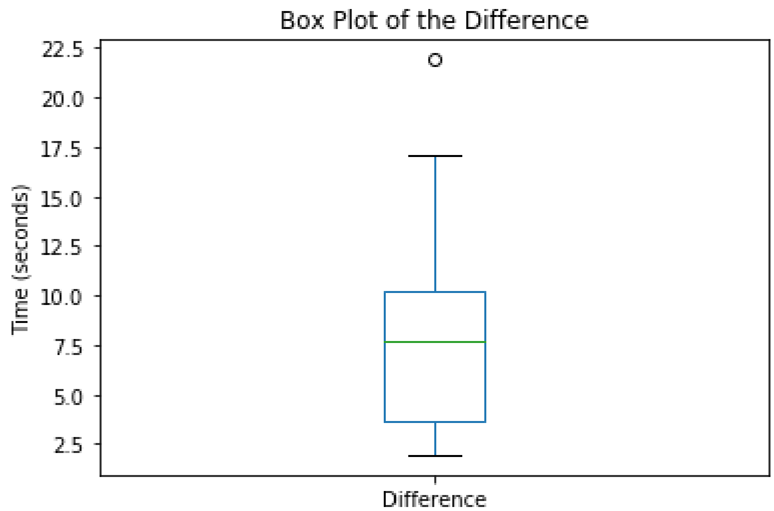
Plots
Please see below a boxplot and histogram which show the distribution of data from both congruent and incongruent conditions.

Observations
From the boxplot, there are two somewhat obvious outliers or extraneous data which would possible skew the true mean of incongruent values. And from the histogram plots, although both graphs visually appear somewhat positively skewed, the mean is pretty close to the peak in both graphs which would indicate a normal distribution. Provided these are samples from the population, the sampling mean would be similar to the population mean.

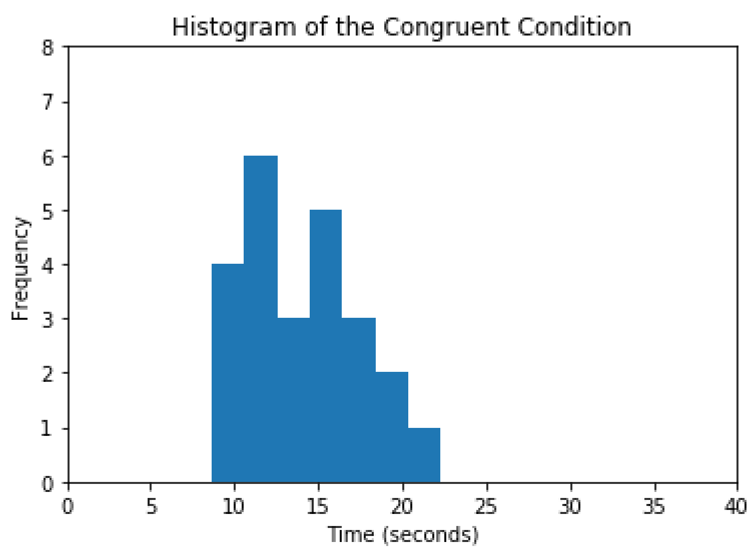
In [9]:
```
# Box plots of the two conditions
title = 'Box Plot of Both Conditions'
kind = 'box'
stroopDF.plot(title=title, kind=kind)
ylabel = plt.ylabel('Time (seconds)')
```



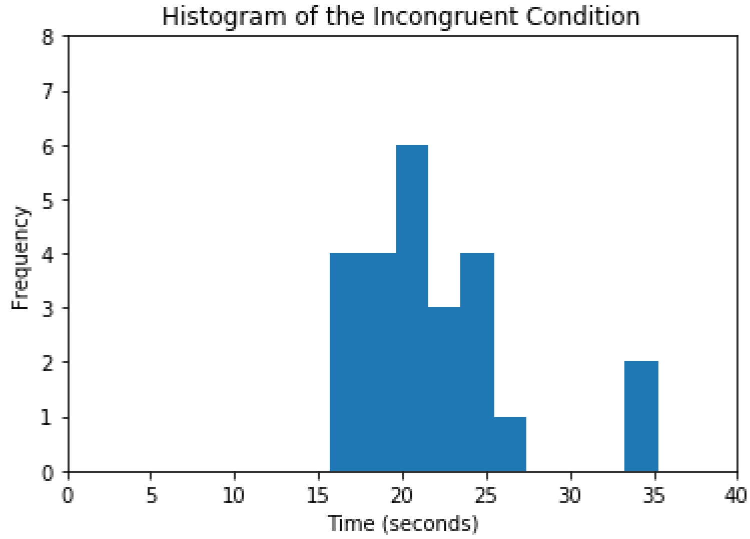
In [10]:
```
# Box plots of the difference between the two conditions
title = 'Box Plot of the Difference'
kind = 'box'
differenceDF.plot(title=title, kind=kind)
ylabel = plt.ylabel('Time (seconds)')
```



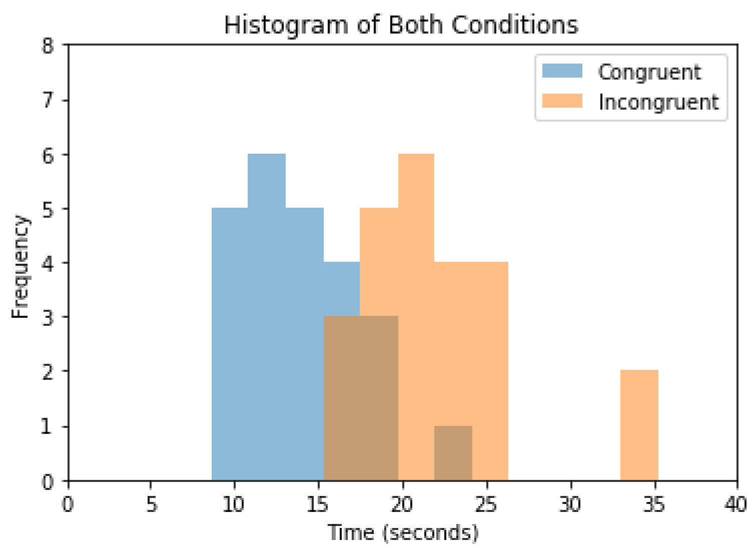
In [11]:
```
# Histogram of the Congruent Condition
title = 'Histogram of the Congruent Condition'
kind = 'hist'
plot = congruent.plot(title=title, kind=kind, bins=7)
xLabel = plt.xlabel('Time (seconds)')
window = plt.axis([0,40,0,8])
```



In [12]:
```
# Histogram of the Incongruent Condition
title = 'Histogram of the Incongruent Condition'
kind = 'hist'
plot = incongruent.plot(title=title, kind=kind)
xLabel = plt.xlabel('Time (seconds)')
window = plt.axis([0,40,0,8])
```



In [13]:
```
# Histogram of Both Conditions
title = 'Histogram of Both Conditions'
kind = 'hist'
alpha = 0.5
plot = stroopDF.plot(title=title, kind=kind, alpha=alpha, bins=12)
xLabel = plt.xlabel('Time (seconds)')
window = plt.axis([0,40,0,8])
```



5. Now, perform the statistical test and report your results. What is your confidence level and your critical statistic value? Do you reject the null hypothesis or fail to reject it? Come to a conclusion in terms of the experiment task. Did the results match up with your expectations?

In [14]:
```
from IPython.display import Image
```

```
Image(filename='stroop_effect_03.png')
```

Out[14]:

Table B *t* distribution critical values

	Tail probability *p*											
df	.25	.20	.15	.10	.05	.025	.02	.01	.005	.0025	.001	.0005
1	1.000	1.376	1.963	3.078	6.314	12.71	15.89	31.82	63.66	127.3	318.3	636.6
2	.816	1.061	1.386	1.886	2.920	4.303	4.849	6.965	9.925	14.09	22.33	31.60
3	.765	.978	1.250	1.638	2.353	3.182	3.482	4.541	5.841	7.453	10.21	12.92
4	.741	.941	1.190	1.533	2.132	2.776	2.999	3.747	4.604	5.598	7.173	8.610
5	.727	.920	1.156	1.476	2.015	2.571	2.757	3.365	4.032	4.773	5.893	6.869
6	.718	.906	1.134	1.440	1.943	2.447	2.612	3.143	3.707	4.317	5.208	5.959
7	.711	.896	1.119	1.415	1.895	2.365	2.517	2.998	3.499	4.029	4.785	5.408
8	.706	.889	1.108	1.397	1.860	2.306	2.449	2.896	3.355	3.833	4.501	5.041
9	.703	.883	1.100	1.383	1.833	2.262	2.398	2.821	3.250	3.690	4.297	4.781
10	.700	.879	1.093	1.372	1.812	2.228	2.359	2.764	3.169	3.581	4.144	4.587
11	.697	.876	1.088	1.363	1.796	2.201	2.328	2.718	3.106	3.497	4.025	4.437
12	.695	.873	1.083	1.356	1.782	2.179	2.303	2.681	3.055	3.428	3.930	4.318
13	.694	.870	1.079	1.350	1.771	2.160	2.282	2.650	3.012	3.372	3.852	4.221
14	.692	.868	1.076	1.345	1.761	2.145	2.264	2.624	2.977	3.326	3.787	4.140
15	.691	.866	1.074	1.341	1.753	2.131	2.249	2.602	2.947	3.286	3.733	4.073
16	.690	.865	1.071	1.337	1.746	2.120	2.235	2.583	2.921	3.252	3.686	4.015
17	.689	.863	1.069	1.333	1.740	2.110	2.224	2.567	2.898	3.222	3.646	3.965
18	.688	.862	1.067	1.330	1.734	2.101	2.214	2.552	2.878	3.197	3.611	3.922
19	.688	.861	1.066	1.328	1.729	2.093	2.205	2.539	2.861	3.174	3.579	3.883
20	.687	.860	1.064	1.325	1.725	2.086	2.197	2.528	2.845	3.153	3.552	3.850
21	.686	.859	1.063	1.323	1.721	2.080	2.189	2.518	2.831	3.135	3.527	3.819
22	.686	.858	1.061	1.321	1.717	2.074	2.183	2.508	2.819	3.119	3.505	3.792
23	.685	.858	1.060	1.319	1.714	2.069	2.177	2.500	2.807	3.104	3.485	3.768
24	.685	.857	1.059	1.318	1.711	2.064	2.172	2.492	2.797	3.091	3.467	3.745

Using $\alpha = .05$, the results indicated that the incongruent condition ($\mu_i = 22.02$, $\sigma_i = 4.80$) took particants significantly longer to identify than the congruent condition ($\mu_c = 14.05$, $\sigma_c = 3.56$), with $t(23) = 8.02$, $p < .01$.

As we can see here, our t-statistic - 8.02 is larger than the critical values for all of our alpha levels highlighted above.
Therefore, we can reject the Null hypothesis at a confidence level greater than 99%. In fact, there doesn't seem to be any value on the t-table that comes very close to our t-statistic. Hence it is extremely likely that the difference in these distributions is not due to chance.

Hypothesis
I reject the null hypothesis i.e the word/colour incongruent does cause a greater time response
Conclusion
The results match my expectations.

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!

One of the prime reason for this observered effect is the strong linkage between words we see, and our interpretation of them. It is very difficult for anyone to "see" a word, without "reading" it.
This link clearly has a profound effect on how we draw conclusions about abstract concepts presented to us, such as the color blue.
I assume replacing letters with numbers in the experiment would yeild the same results.
Investigating outcomes for fair vs unfair coins would yeild similar effects.

References

https://en.wikipedia.org/wiki/Stroop_effect
http://www.statisticshowto.com/when-to-use-a-t-score-vs-z-score/
https://s3.amazonaws.com/udacity-hosted-downloads/t-table.jpg

In [ ]: