

# 363 Final Report: Stroop test

Andrej Glavic, Yuanxi Liu 20688855, Rachel Morton 20925026, Wasam Syed 20746474, Amy

April 18, 2021

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Methods</b>	<b>2</b>
<b>3</b>	<b>Results</b>	<b>2</b>
3.1	1. Collect data into a dataframe: . . . . .	2
3.2	Descriptive statistics . . . . .	12
<b>4</b>	<b>Discussion and Conclusions</b>	<b>13</b>
<b>5</b>	<b>References</b>	<b>13</b>

## 1 Introduction

John Ridley Stroop published the groundbreaking, "Studies of interference in serial verbal ractions" in the Journal of Experimental Psychology in 1935 [4]. Ever since, psychology students worldwide learn about the "Stroop effect", where incongruent stimuli take longer to process than do congruent stimuli. Even though Stroop wasn't the first to publish this effect, his experiments were foundational; his original study is one of the most-cited papers in the history of experimental psychology [1].

In one part of Stroop's classic study, participants had to say the ink colour of the printed word rather than read the word. For instance, if the word 'RED' was printed in purple ink, they were to say, "purple" and not "red". Stroop noticed that subjects took significantly longer to complete this colour naming task than one where they just had to name the colour of coloured squares. This difference in response time, or processing speed, is what is known as the Stroop effect.

## 2 Methods

There have been many variations on Stroop's experiment over the years. Since key presses are easier to time than a person reading aloud, we chose to go with a "manual" Stroop task where reaction time is measured by how long it takes a participant to press a certain key on a keyboard after being shown the stimuli on the computer screen. We used the psychopy library in python to program the experiment [2] and followed the example shown in the demo Stroop task at psytoolkit.org [3].

Computational tools used? R??

## 3 Results

### 3.1 1. Collect data into a dataframe:

```
#graphics :file "StroopPlot.png"
import pandas as pd
import os
import numpy as np
import matplotlib.pyplot as plt

files = os.listdir()

# Identify all csv files in the directory
csv_files = []
for file in files:
    if file[-4:] == ".csv":
        csv_files.append(file)
data = pd.DataFrame()
for file in csv_files:
    data = data.append(pd.read_csv(file))

pd.set_option('display.max_rows', None)
print(data)
```

Python 3.8.5 (default, Jan 27 2021, 15:41:15)  
[GCC 9.3.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.

	Word	Ink	Response Time	Correct
0	g	b	0.902731	1.0
1	g	g	0.757359	1.0
2	b	y	0.914164	1.0
3	r	b	0.928025	1.0
4	b	b	0.566040	1.0
5	g	g	0.834033	1.0
6	y	r	0.811387	1.0

7	r	b	1.237977	1.0
8	b	b	0.583341	1.0
9	b	r	0.914697	1.0
10	b	b	0.824691	1.0
11	y	b	0.978719	1.0
12	y	y	0.622731	1.0
13	b	r	0.880696	1.0
14	r	r	0.560040	1.0
15	y	y	0.676701	1.0
16	r	y	1.164679	1.0
17	g	b	0.952696	1.0
18	b	y	1.005375	1.0
19	r	r	0.687999	1.0
0	b	b	0.650724	1.0
1	y	g	0.875670	1.0
2	y	g	0.658684	1.0
3	r	g	0.922880	1.0
4	y	r	1.088798	1.0
5	y	y	0.949007	1.0
6	r	y	0.546538	1.0
7	r	g	0.935575	1.0
8	g	b	0.869165	1.0
9	g	g	0.659764	1.0
10	y	y	0.788567	1.0
11	b	y	0.839376	1.0
12	r	r	0.742394	1.0
13	r	y	0.699396	0.0
14	g	r	1.058347	1.0
15	y	y	0.969040	1.0
16	b	y	0.693291	1.0
17	g	b	0.963676	0.0
18	b	r	0.903184	1.0
19	r	r	0.790540	1.0
0	b	b	1.172047	1.0
1	g	g	1.376213	1.0
2	b	g	2.060264	1.0
3	b	b	1.380893	1.0
4	g	y	1.528297	1.0
5	g	g	1.526311	1.0
6	y	g	2.332335	1.0
7	b	b	1.609578	1.0
8	r	g	1.940261	1.0
9	b	b	1.545599	1.0
10	b	g	1.622961	0.0
11	r	r	1.744913	1.0
12	g	g	1.846302	1.0

13	g	g	1.655584	1.0
14	b	b	1.968250	1.0
15	b	b	1.412247	1.0
16	b	b	1.318842	1.0
17	g	b	1.468872	1.0
18	y	g	1.748895	0.0
19	y	g	1.825570	1.0
0	y	y	0.977377	1.0
1	y	y	0.757410	1.0
2	b	b	1.050211	1.0
3	b	y	0.986739	1.0
4	b	r	0.902156	1.0
5	g	g	1.229024	1.0
6	r	r	0.990830	1.0
7	g	g	0.986155	1.0
8	g	g	0.781414	1.0
9	y	y	1.080983	1.0
10	r	y	1.088067	1.0
11	r	g	0.967526	1.0
12	g	b	1.107587	1.0
13	r	r	0.914097	1.0
14	r	y	0.904100	1.0
15	g	y	0.910802	1.0
16	b	b	1.301593	1.0
17	g	g	1.052199	1.0
18	y	b	1.260909	1.0
19	b	y	1.021478	1.0
0	b	b	1.315021	1.0
1	g	g	1.865382	1.0
2	g	b	1.617281	0.0
3	r	r	1.515250	1.0
4	b	b	1.177302	1.0
5	b	y	1.861330	0.0
6	y	y	2.373174	1.0
7	r	y	1.865256	1.0
8	r	g	1.649240	1.0
9	y	g	3.073948	1.0
10	r	r	2.361929	1.0
11	r	b	2.534568	1.0
12	b	b	2.459909	1.0
13	g	b	2.012503	1.0
14	y	g	1.647227	0.0
15	g	g	2.199827	1.0
16	b	g	1.970596	1.0
17	y	b	1.998565	1.0
18	r	b	1.657183	1.0

0	g	g	0.983795	1.0
1	b	g	1.017413	1.0
2	g	g	0.515788	1.0
3	r	g	0.586648	1.0
4	y	r	1.277641	1.0
5	r	b	1.068428	1.0
6	y	y	0.857382	1.0
7	r	r	0.977592	1.0
8	b	y	1.500616	1.0
9	b	b	0.804275	1.0
10	g	g	0.998647	1.0
11	b	g	1.281180	1.0
12	g	g	0.552297	1.0
13	r	b	1.132891	1.0
14	y	b	0.770555	1.0
15	r	r	1.678370	1.0
16	g	g	0.773462	1.0
17	y	y	0.889242	1.0
18	b	b	0.814580	1.0
19	y	b	1.240382	1.0
0	y	r	2.033660	1.0
1	g	g	1.467006	1.0
2	y	y	1.594960	1.0
3	y	y	1.526392	1.0
4	r	y	1.500360	1.0
5	g	g	1.611039	1.0
6	g	g	1.677788	1.0
7	y	y	2.797804	1.0
8	r	r	2.456315	1.0
9	r	g	1.737014	1.0
10	g	g	1.990351	1.0
11	g	g	1.929699	1.0
12	y	y	1.727609	1.0
13	r	b	1.981695	1.0
14	r	b	1.896927	0.0
15	y	y	1.807787	1.0
16	y	y	1.874347	1.0
17	y	b	1.882387	1.0
18	r	r	2.201704	1.0
19	y	y	1.644511	1.0
0	y	y	0.910879	1.0
1	y	y	0.469010	1.0
2	r	r	0.620605	1.0
3	r	r	0.550894	1.0
4	g	b	1.267275	1.0
5	r	g	1.176622	1.0

6	r	g	0.843934	1.0
7	g	g	0.555604	1.0
8	r	b	1.070192	1.0
9	r	y	1.318873	1.0
10	r	g	0.852656	1.0
11	b	g	1.292762	1.0
12	b	y	0.963082	1.0
13	b	r	1.271206	1.0
14	y	y	1.197752	1.0
15	g	b	1.199835	0.0
16	g	g	1.226448	1.0
17	y	b	1.218104	1.0
18	r	b	1.140128	1.0
19	g	g	0.912819	1.0
0	b	r	1.558254	1.0
1	y	b	0.722076	1.0
2	b	r	1.286783	1.0
3	y	r	0.693706	1.0
4	y	g	1.460833	1.0
5	r	y	1.355493	1.0
6	y	g	1.470367	1.0
7	y	g	0.686714	1.0
8	g	g	0.687071	1.0
9	r	r	0.702120	1.0
10	r	b	1.218535	1.0
11	y	y	0.837985	0.0
12	b	b	1.590545	1.0
13	b	b	0.641961	1.0
14	y	r	0.905767	1.0
15	b	b	0.613839	1.0
16	y	y	0.970136	1.0
17	b	y	0.989939	1.0
18	g	g	1.008779	1.0
19	b	y	1.127347	1.0
0	g	g	1.481261	1.0
1	y	y	1.440831	1.0
2	g	b	1.561981	1.0
3	y	y	1.254616	1.0
4	r	y	1.575352	1.0
5	b	b	1.507336	1.0
6	g	g	1.417933	1.0
7	b	b	1.476005	1.0
8	r	y	2.204069	1.0
9	y	y	1.718706	1.0
10	r	b	3.724674	1.0
11	g	g	1.490654	1.0

12	y	r	2.954000	1.0
13	y	b	1.766542	1.0
14	g	g	1.602538	1.0
15	y	y	1.645362	1.0
16	g	g	1.494557	1.0
17	b	b	1.757998	1.0
18	r	g	2.146605	1.0
19	y	b	1.671941	1.0
0	b	b	1.117947	1.0
1	r	r	0.593142	1.0
2	r	b	1.021338	1.0
3	b	b	0.654462	1.0
4	y	y	1.116462	1.0
5	y	y	0.713730	1.0
6	b	r	0.936620	1.0
7	b	b	0.579975	1.0
8	r	b	0.967414	1.0
9	g	b	0.811538	1.0
10	r	r	0.638495	1.0
11	r	r	0.604862	1.0
12	y	y	0.846839	1.0
13	b	b	0.599635	1.0
14	g	g	0.931120	1.0
15	b	b	0.624305	1.0
16	r	b	1.969691	1.0
17	b	b	0.741825	1.0
18	g	b	0.891133	1.0
19	r	r	0.644671	1.0
0	b	g	1.067161	1.0
1	y	g	0.644196	1.0
2	g	g	0.492713	1.0
3	g	y	1.009481	1.0
4	y	b	0.881858	1.0
5	r	r	0.666264	1.0
6	y	y	0.796431	1.0
7	y	y	0.605677	1.0
8	y	g	0.934725	1.0
9	g	r	0.859989	1.0
10	g	g	0.660462	1.0
11	b	y	0.868914	1.0
12	y	b	1.692175	1.0
13	b	r	0.819546	1.0
14	b	y	1.029049	1.0
15	r	r	0.671236	1.0
16	g	g	0.934642	1.0
17	r	r	0.835753	1.0

18	y	b	1.387948	1.0
19	b	b	0.605799	1.0
0	r	g	0.907942	1.0
1	b	b	0.881836	1.0
2	b	b	0.552707	1.0
3	g	b	1.173299	1.0
4	y	y	0.864438	1.0
5	y	y	0.742724	1.0
6	y	r	0.739302	1.0
7	y	b	1.113319	1.0
8	r	r	0.767271	1.0
9	g	g	0.730309	1.0
10	r	b	0.950361	1.0
11	y	g	0.947094	1.0
12	r	y	0.939133	1.0
13	r	r	0.734647	1.0
14	y	g	1.193929	1.0
15	g	g	0.685919	1.0
16	y	g	0.588277	1.0
17	g	r	0.944851	1.0
18	b	b	0.820510	1.0
19	y	r	1.093873	1.0
0	b	b	0.327669	1.0
1	b	b	0.781805	1.0
2	y	y	1.280098	1.0
3	g	b	1.103348	1.0
4	y	y	0.963182	1.0
5	r	r	0.888070	1.0
6	r	r	0.757919	1.0
7	y	y	0.808787	1.0
8	b	b	0.787059	1.0
9	y	r	0.972978	1.0
10	y	g	1.136630	1.0
11	r	b	1.742167	1.0
12	y	y	1.067661	1.0
13	b	b	0.843104	1.0
14	y	y	0.857924	1.0
15	g	r	1.147155	1.0
16	y	y	0.918772	1.0
17	y	y	0.685667	1.0
18	r	r	1.067134	1.0
19	g	y	1.795667	1.0
0	g	g	0.992047	1.0
1	y	b	2.107541	1.0
2	b	r	1.306184	1.0
3	g	r	1.824886	1.0



4	y	b	1.266881	1.0
5	g	y	1.210864	1.0
6	r	y	0.845522	1.0
7	g	b	1.214871	1.0
8	y	r	1.432753	1.0
9	r	r	1.738093	1.0
10	r	y	1.519436	1.0
11	r	b	1.141418	1.0
12	b	r	1.770746	1.0
13	b	y	1.190761	1.0
14	g	g	1.626138	1.0
15	b	r	2.068018	1.0
16	g	b	1.619989	1.0
17	y	g	1.614139	1.0
18	b	b	0.912171	1.0
19	y	b	0.878804	0.0
0	g	g	0.889023	1.0
1	b	b	0.744468	1.0
2	g	b	1.315463	1.0
3	r	b	0.800622	1.0
4	g	g	0.956006	1.0
5	r	y	1.014641	1.0
6	y	g	0.873009	1.0
7	y	r	0.911480	1.0
8	r	r	0.654158	1.0
9	b	r	0.766908	1.0
10	r	g	0.978224	1.0
11	r	g	0.741426	1.0
12	g	y	1.202270	1.0
13	b	b	1.173223	1.0
14	b	b	0.680816	1.0
15	r	r	0.679807	1.0
16	r	r	0.535366	1.0
17	r	r	0.420098	1.0
18	r	r	0.437865	1.0
19	r	g	1.335448	1.0
0	y	g	1.131475	1.0
1	y	g	0.479914	1.0
2	y	y	1.094262	1.0
3	b	g	1.436906	1.0
4	b	b	0.860741	1.0
5	b	b	0.531961	1.0
6	b	b	0.489254	1.0
7	b	r	0.855807	1.0
8	y	g	1.015819	1.0
9	b	b	0.713459	1.0

10	y	r	1.719761	1.0
11	b	g	1.166920	1.0
12	r	r	0.627157	1.0
13	r	r	0.491899	1.0
14	r	b	1.569950	1.0
15	r	y	0.620472	0.0
16	b	b	1.047315	1.0
17	r	r	0.952231	1.0
18	r	y	1.101043	1.0
19	b	g	0.987293	1.0
0	g	y	4.495212	1.0
1	b	b	1.742471	1.0
2	r	y	1.995806	1.0
3	b	y	1.783156	0.0
4	r	g	3.298523	1.0
5	g	b	2.345141	1.0
6	r	b	1.654407	1.0
7	r	b	1.567104	1.0
8	b	g	2.091159	1.0
9	r	y	1.754488	1.0
10	r	r	2.095204	1.0
11	g	y	2.726535	0.0
12	y	g	2.532567	1.0
13	y	y	1.769790	1.0
14	r	g	2.037907	1.0
15	r	b	1.847148	1.0
16	r	b	1.775833	1.0
17	g	y	2.003723	1.0
18	b	b	1.502396	1.0
19	r	b	1.629036	1.0
0	g	g	1.003741	1.0
1	g	g	1.782742	1.0
2	b	r	1.602037	0.0
3	r	g	3.276595	1.0
4	y	g	2.180584	1.0
5	r	r	1.607845	0.0
6	r	r	3.002608	1.0
7	r	r	1.261343	1.0
8	b	r	1.718036	1.0
9	y	y	1.419899	1.0
10	r	r	1.493981	1.0
11	g	y	1.989347	1.0
12	g	r	1.888707	1.0
13	y	b	2.231908	1.0
14	g	g	1.873276	1.0
15	r	y	1.873278	1.0

```

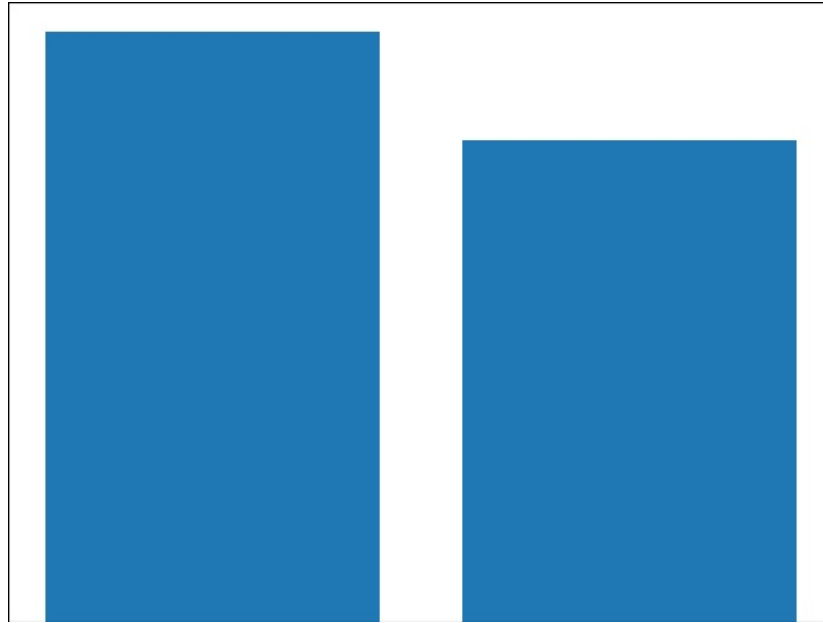
16     y    y        1.865308        1.0
17     y    r        1.841314        1.0
18     b    b        1.498685        1.0
19     y    y        2.081316        1.0
python.el: native completion setup loaded

#output graphics :file "StroopPlot.jpg"
# Average Congruent vs Incongruent Time
congruent = np.array([])
incongruent = np.array([])
for i in data.iterrows():
    if i[1]['Correct']:
if i[1]['Word'] != i[1]['Ink']:
    congruent = np.append(congruent, i[1]['Response Time'])
else:
    incongruent = np.append(incongruent, i[1]['Response Time'])

print(np.average(congruent))
print(np.average(incongruent))
print("DONE")

fig = plt.figure()
ax = fig.add_axes([0,0,1,1])
state = ['Congruent', 'Incongruent']
average = [np.average(congruent)*1000, np.average(incongruent)*1000]
ax.bar(state, average)
#plt.show()
plt.savefig("/home/nobrainleft/Documents/363/Psych363FinalProject/StroopPlot.jpg")
"/home/nobrainleft/Documents/363/Psych363FinalProject/StroopPlot.jpg"

```



### 3.2 Descriptive statistics

[##] participants completed [##] trials.

```
data.mean()
```

```
Response Time    1.249759
Correct          0.957784
dtype: float64
```

[Include code]

Congruent trials are where the colour word and the font colour match. Incongruent trials are where the font colour is different than the colour word.

We calculated the Stroop effect as the average response time for correct, incongruent trials minus correct, congruent trials.

[t-test comparing the means of (correct) congruent vs (correct) incongruent trials.]

[PLOT]

## 4 Discussion and Conclusions

Our experiment takes less than two minutes to complete. It requires pressing the appropriate keyboard key rather than naming the colour aloud as Stroop did originally [4]. In our test, there are only 20 trials and a handful [##?] of participants. For a more reliable measure of the Stroop effect you would want to have considerably more participants along with more trials.

## 5 References

### References

- [1] Colin M. MacLeod. Half a century of research on the stroop effect: An integrative review. *Psychological Bulletin*, 109:163–203, 1991.
- [2] Jonathan Peirce, Jeremy R. Gray, Sol Simpson, Michael MacAskill, Richard Höchenberger, Hiroyuki Sogo, Erik Kastman, and Jonas Kristoffer Lindeløv. Psychopy2: Experiments in behavior made easy. *Behavior Research Methods*, 51:195–203, 2019.
- [3] Gijsbert Stoet. Stroop task demo. <https://www.psychtoolkit.org/experiment-library/stroop.html>. Accessed: 2021-04-12.
- [4] J. Ridley Stroop. Studies of interference in serial verbal reactions. *Journal of Experimental Psychology*, 18:643–662, 1935.