

PSYCH 363 - Stroop Effect: Congruency and Response Time

Keagan McMahon, Brigitta Munds,
Benjamin Brown, & Christina Rachmadita

<2020-12-14 Mon> December 14th, 2020

Contents

1	Introduction	1
2	Methods	1
3	Results	1
4	Conclusions	4
5	References	4
6	Testing Plots here.....	4

This is to test your installation of the files and programs needed to make a simple report. To compile to pdf use `C-c C-e 1 p`.

1 Introduction

Insert introduction text here...

2 Methods

Insert some method text here

This loads an R library

```
library(random)
```

3 Results

Insert some results text here and other content (i.e. code, etc)

Now we will see if we can some source code and a simple plot for our export.

```
x = 1:10
y = rnorm(10)
print(mean(y))
```

```
-0.436099408057575
```

Here is some more R source code!

```
{ a=2
  b=6
  multiply <- function(a,b)
  return(a * b)
  print(paste(a, "multiplied by", b, "is", (print(multiply(a,b)))))
}

{ for(i in seq(1,10))
  if(i%%2==0){
  print(i) }
}
```

Now lets try some Python source code from my loop assignment...

```
letters = ['t', 'r', 'i', 'b', 'q', 'v', 'h', 'p']
position = ['1st', '2nd', '3rd', '4th', '5th', '6th', '7th', '8th']

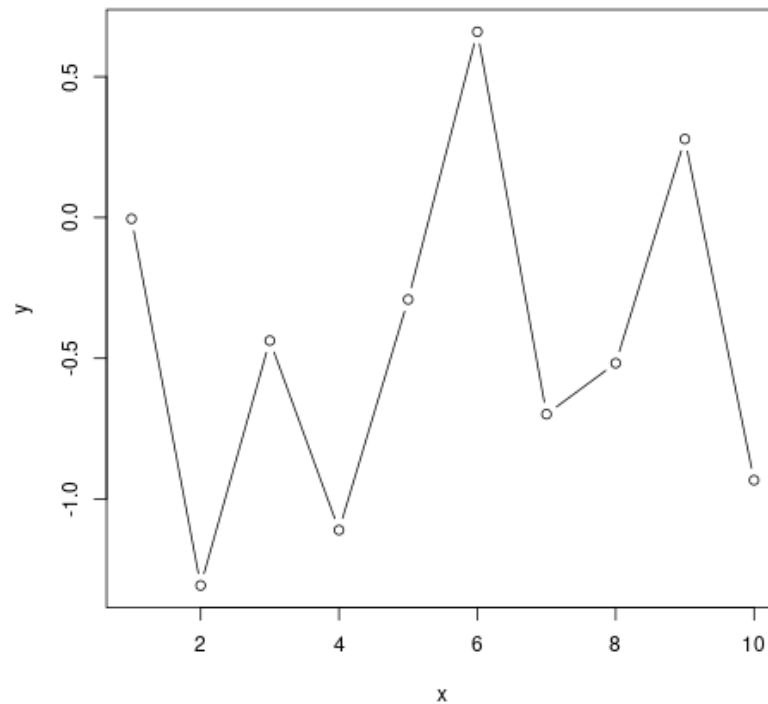
for x in letters:
    print(x)

for i in sorted(letters):
    print(i)

for x in enumerate(zip(letters, position)):
    print("The {0} letter in list 1 is {0}".format(x))
```

Here is a graph of our results for you to see:

```
plot(x,y,type = 'b')
```



Here is some code that produces a table of data for us:

```
d <- data.frame(foo=c('a','b','n'), bar=c(1.0/3.0,22,32))
d
```

foo	bar
a	0.333333333333333
b	22
n	32

Here is an example of an inline piece of code, it will generate 20 random numbers:

```
xinline = rnorm(20)
```

We can use that code in this way:

The mean of 20 mean 0 normally distributed numbers is 0.0581210550728103.

4 Conclusions

Put some type of conclusion content here. . . .

5 References

Insert some references here, such as. . .

This article [1]

References

- [1] Britt Anderson. There is no such thing as attention. *Frontiers in Psychology*, 2:246, 2011.

6 Testing Plots here.

```
library(ggplot2)
```

```
data <- read.csv("/home/keagan/GitRepos/363Stroop/363Stroop_Data_Dec_4.csv")
```

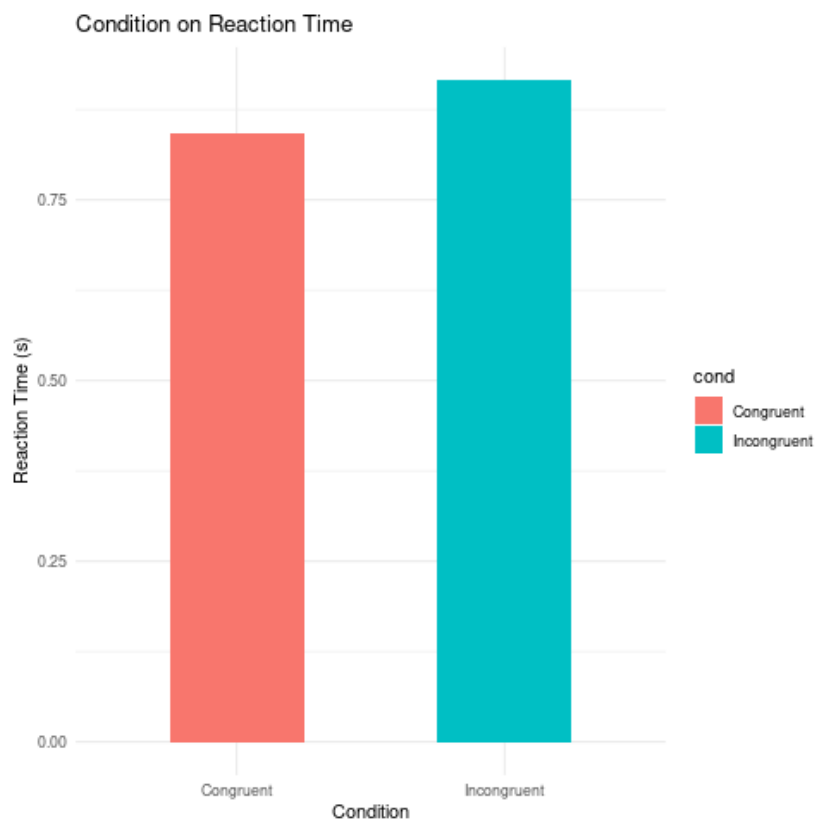
```
incongruent <- data[which(data$Congruent == 0),]$Time
```

```
congruent <- data[which(data$Congruent == 1),]$Time
```

```
df <- data.frame(cond = c("Incongruent", "Congruent"), rt = c(mean(incongruent), mean(congruent)))
```

```
p <- ggplot(df, aes(x = cond, y = rt, fill = cond)) + geom_bar(stat = "identity", width = 0.5)
```

```
p
```



```
library(ggplot2)

data <- read.csv("/home/keagan/GitRepos/363Stroop/363Stroop_Data_Dec_4.csv")

Lincongruent <- c()
counter = 1
while(counter <= 20) {
  T = data[which(data$Trial == counter & data$Congruent == 0),]
  mean_RT = mean(T$Time)
  Lincongruent = append(Lincongruent, mean_RT)
  counter = counter + 1
}

Lcongruent <- c()
counter = 1
while(counter <= 20) {
  T = data[which(data$Trial == counter & data$Congruent == 1),]
  mean_RT = mean(T$Time)
  Lcongruent = append(Lcongruent, mean_RT)
}
```

```

    counter = counter + 1
  }

  cond_rt_df <- data.frame(Condition = rep(c("Congruent", "Incongruent"), each = 20), RT = c(
df <- data.frame(Congruent = Lcongruent, Incongruent = Lincongruent)
df$Interference <- df$Incongruent - df$Congruent

incongruent_mean <- mean(data[which(data$Congruent == 0),]$Time)
congruent_mean <- mean(data[which(data$Congruent == 1),]$Time)
overall <- data.frame(cond = c("Incongruent", "Congruent"), rt = c(incongruent_mean, congruent_mean))

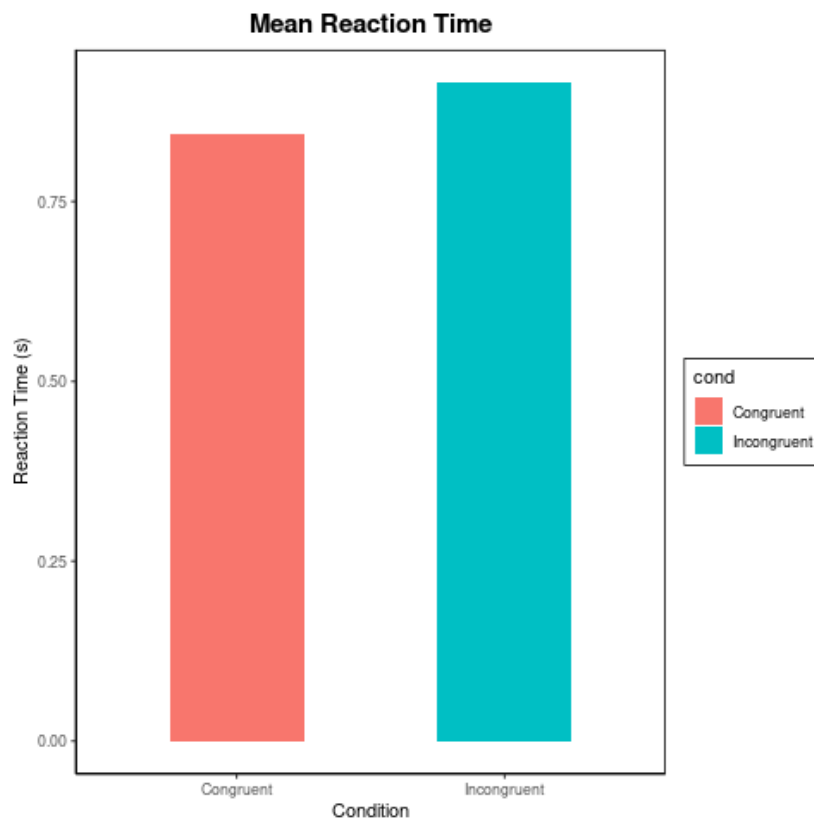
```

Incongruent	0.915385980111821
Congruent	0.843044126528736

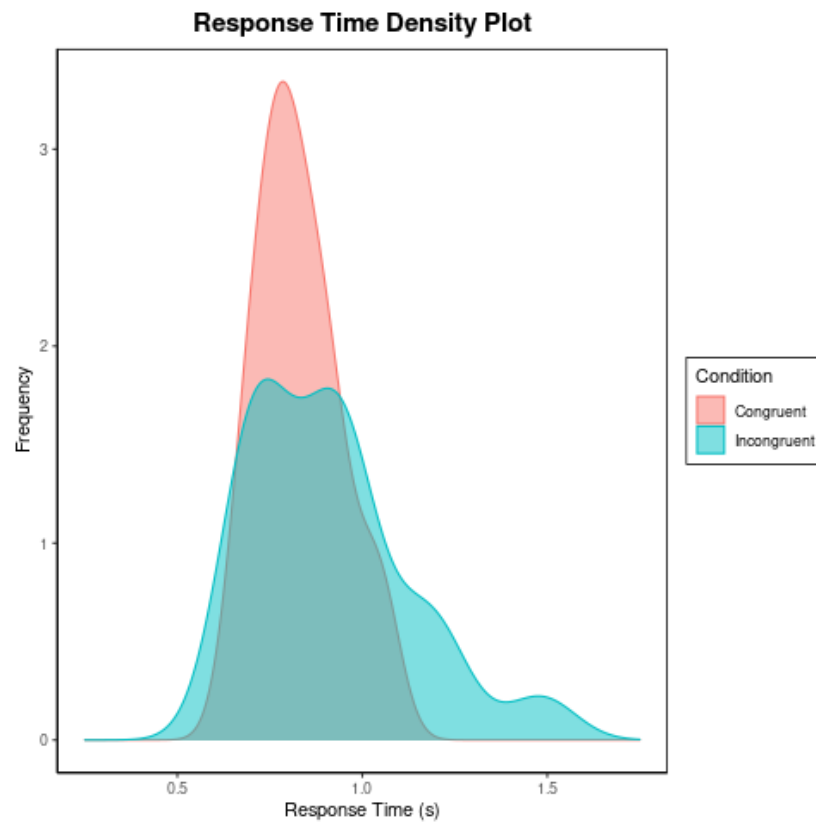
```

p <- ggplot(overall, aes(x = cond, y = rt, fill = cond)) + geom_bar(stat = "identity", width = 0.5)
p

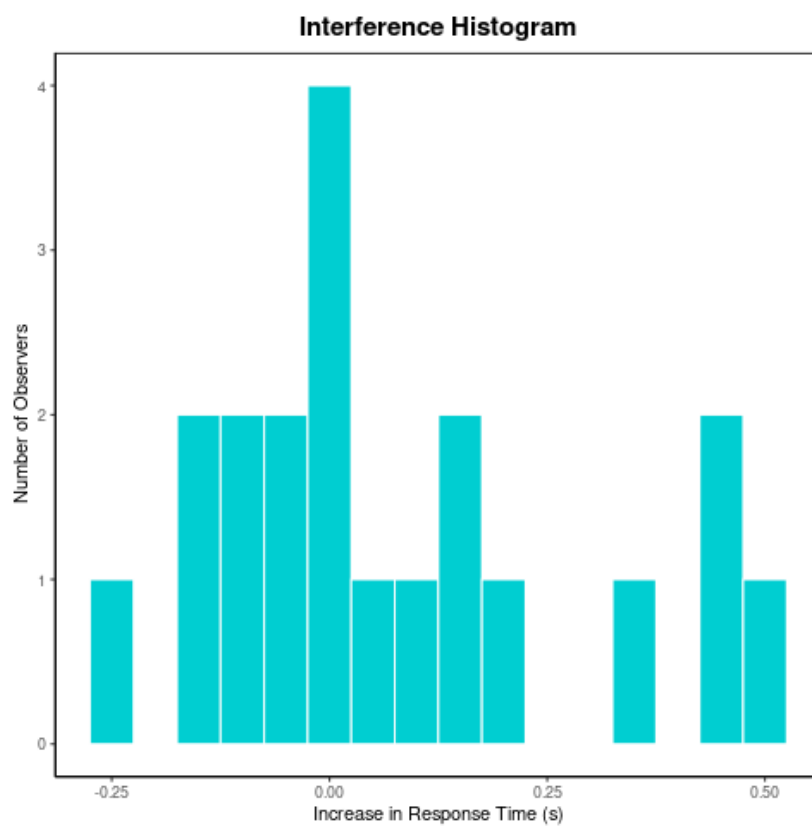
```



```
density_plot <- ggplot(cond_rt_df, aes(x = RT, color = Condition, fill = Condition)) + geom.  
density_plot
```

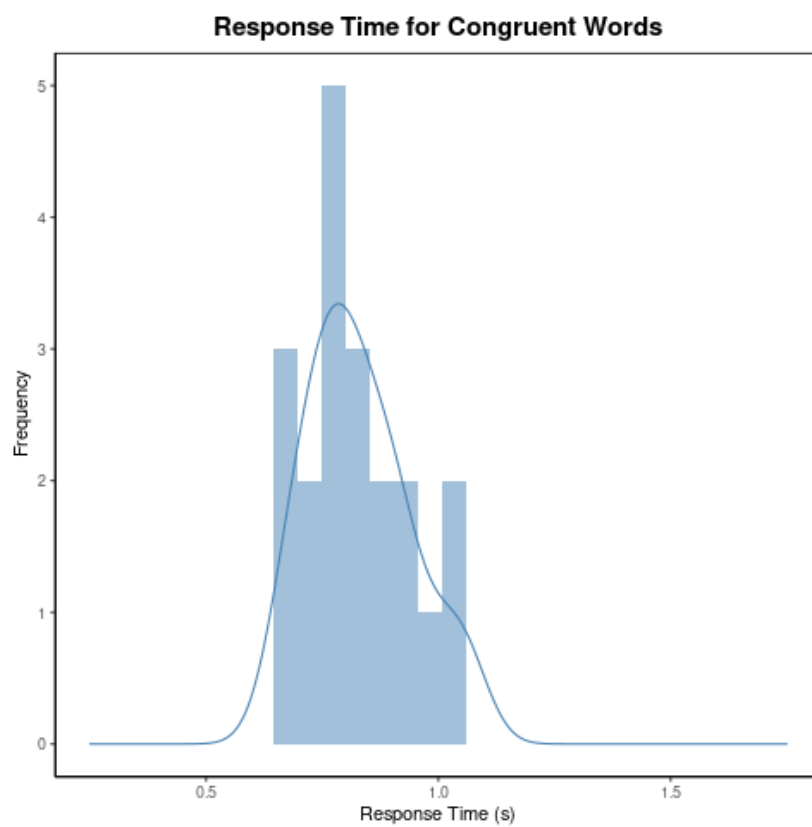


```
interference_hist <- ggplot(df, aes(x = Interference)) + geom_histogram(binwidth = 0.05, col.  
interference_hist
```

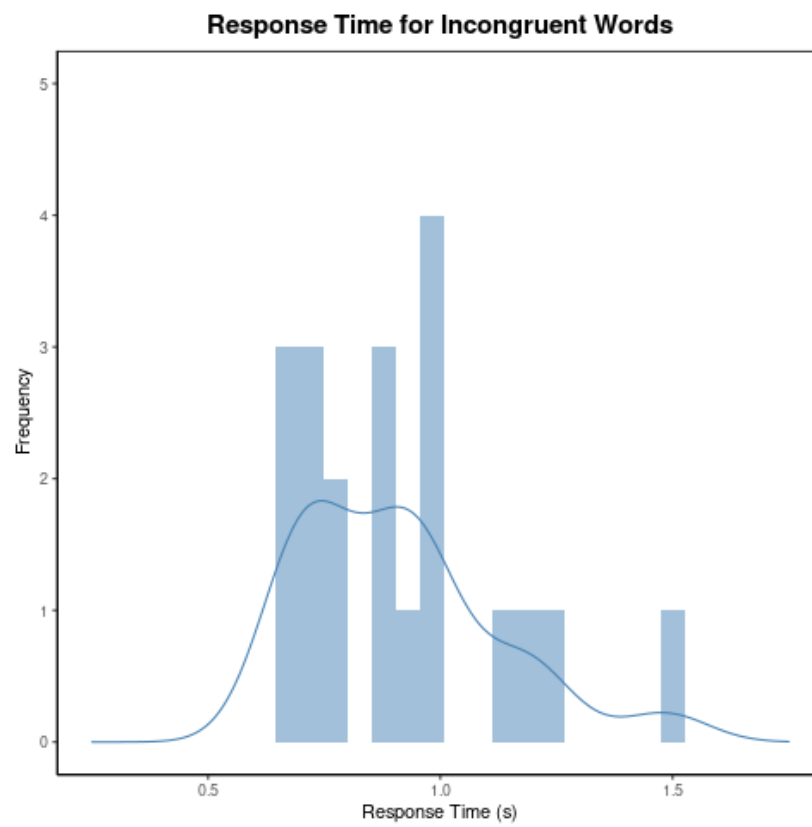


```
RT_congruent <- ggplot(df, aes(x = Congruent)) + geom_histogram(alpha = 0.5, fill = "steelb
```

```
RT_congruent
```

```
RT_incongruent <- ggplot(df, aes(x = Incongruent)) + geom_histogram(alpha = 0.5, fill = "steelblue") +  
  geom_density(alpha = 0.5, fill = "steelblue")  
RT_incongruent
```



```
RT_cond <- ggplot(cond_rt_df, aes(x = RT, color = Condition, fill = Condition)) + geom_histogram()
RT_cond
```

Response Time for Congruent vs. Incongruent Words

