StroopTest

November 14, 2015

1 Statistics: The Science of Decisions

```
In [3]: from pandas import *
        dataFrame = read_csv('stroopdata.csv')

In [4]: dataFrame
```

Out[4]:

	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

1.0.1 Variables

The independent variable in this experiment is the reading condition, specifically whether text of words are written in the colour that they describe (congruent), or in a colour other than the one that they describe.

The dependent variable in this experiment is the time taken to read the list of words.

1.0.2 Null and alternative hypotheses

The null hypothesis is that population mean of congruent times is equal to the population mean of incongruent times. - H_0: $\mu_C = \mu_I$ (where μ_C is the congruent population mean and μ_I is the incongruent population mean.)

The alternative hypothesis is that population mean of congruent times is *less* than the population mean of incongruent times (as there is reason to believe that that cogruent mean is less than the incongruent mean from the exploratory visualisations below). - H_1: $\mu_C < \mu_I$

1.0.3 Statistical test rationale

Using sample data from congruent and incongruent times, we will conduct a *one- tailed, two-sample dependent t-test* of the null hypothesis. We will making the following assumptions about the data

- Participants have recorded a congruent and subsequent incongruent time, i.e. the population is the same and times are paired across the two groups.
- Population of congruent and incongruent times are iid normal.
- Variances of groups *are* assumed to be equal.
- Data distributions that are roughly symmetric/mound (Gaussian) shaped and not skewed.

For this analysis, we will use a signficance value of $\alpha = 0.05$.

1.0.4 Measures of central tendency and variance

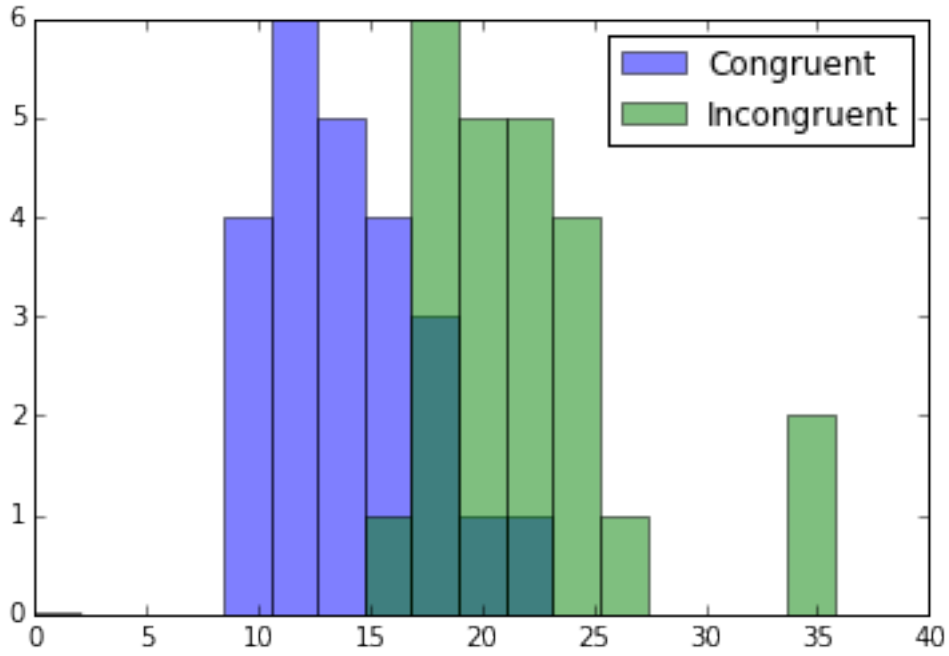
In [5]:
```
#Calculate the mean time to complete the congruent and incongruent exercise
print "The mean time taken for the congruent condition is =", dataFrame["Congruent"].mean()
print "The mean time taken for the incongruent condition is =", dataFrame["Incongruent"].mean()

#Calculate the  the congruent and incongruent variances
print "The congruent time variance =", dataFrame["Congruent"].var()
print "The incongruent time variance =", dataFrame["Incongruent"].var()
```

```
The mean time taken for the congruent condition is = 14.051125
The mean time taken for the incongruent condition is = 22.0159166667
The congruent time variance = 12.6690290707
The incongruent time variance = 23.0117570362
```

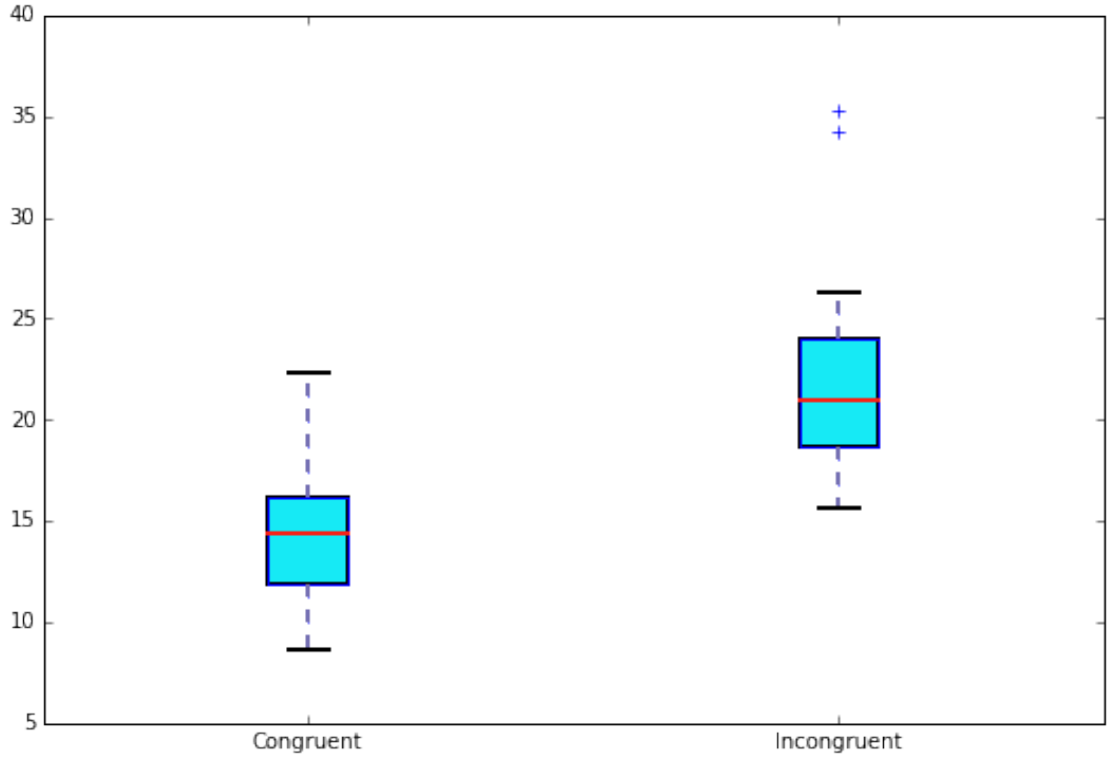
1.0.5 Exploring the data with visualisations

In [6]:
```
#Histogram of congruent and incongruent
#source: http://stackoverflow.com/questions/6871201/plot-two-histograms-at-the-same-time-with-m

import matplotlib as mpl
mpl.use('agg')
import matplotlib.pyplot as plt
%matplotlib inline

import numpy
from pandas import *


dataFrame = read_csv('stroopdata.csv')

x = dataFrame["Congruent"]
y = dataFrame["Incongruent"]

bins = numpy.linspace(0, 40, 20)

plt.hist(x, bins, alpha=0.5, label='Congruent')
plt.hist(y, bins, alpha=0.5, label='Incongruent')
plt.legend(loc='upper right')
```

```
plt.show()

#Box plot of the congruent and incongruent times
#Source: http://blog.bharatbhole.com/creating-boxplots-with-matplotlib/

dataFrame = read_csv('stroopdata.csv')

plotData = [dataFrame["Congruent"], dataFrame["Incongruent"]]

# Create a figure instance
fig = plt.figure(1, figsize=(9, 6))

# Create an axes instance
ax = fig.add_subplot(111)

# Create the boxplot
bp = ax.boxplot(plotData)

# Save the figure
fig.savefig('fig1.png', bbox_inches='tight')

## add patch_artist=True option to ax.boxplot()
## to get fill color
bp = ax.boxplot(plotData, patch_artist=True)

## change outline color, fill color and linewidth of the boxes
for box in bp['boxes']:
    # change outline color
    box.set(facecolor = '#16EAF5', linewidth=2)

## change color and linewidth of the whiskers
for whisker in bp['whiskers']:
    whisker.set(color='#7570b3', linewidth=2)

## change color and linewidth of the caps
for cap in bp['caps']:
    cap.set(linewidth=2)

## change color and linewidth of the medians
for median in bp['medians']:
    median.set(color='#F62217', linewidth=2)

ax.set_xticklabels(['Congruent', 'Incongruent'])
```



Out[6]: [<matplotlib.text.Text at 0x16f26a20>, <matplotlib.text.Text at 0x16f43ba8>]



2 Observations from central tendency measures, variance and exploration of data

- The histogram shows that the distributions of both groups are roughly normal distribution shaped (for a small sample), but both hint at a possible positive skew.
- The means of the two samples are visibly different.
- The variance looks quite similar in the histogram, however the samples do have different variances. In another project I might run an F test to determine the difference.
- The incongruent sample contains outliers.

2.0.6 Performing paired t-test

Since $\alpha = 0.05$, we have a confidence level of 95% for our hypothesis test The critical t-values of the test with $df = 23$ are 1.714 and -1.714

we can reject the null hypothesis $\mu_C = \mu_I$ if - the t-statistic $>$ 1.714 - the t-statistic $<$ -1.714 we would then accept the alternative hypothesis $\mu_C < \mu_I$.

If the t-statistic is between -1.714 and 1.714, we fail to reject the null hypothesis.

```
In [7]: import scipy
        from scipy import stats

        #Calculate t-statistic
        print 'The t-statistic for the test = %6.3f pvalue = %s' %  scipy.stats.ttest_rel(x, y)
        print 'with df = 23'
```

```
The t-statistic for the test = -8.021 pvalue = 4.10300058571e-08
with df = 23
```

2.0.7 Conclusions

The calculated t-statistic for the congruent and incongruent samples is -8.021. This is less than the critical t-value -1.714, therefore we can reject the null hypothesis with more than 95% confidence.

The p-value is very small (4.103e-08) which means that we can reject the null hypothesis with a more than 99.9% confidence.

Experiment conclusions There is a significant difference between the time taken to read a set of words with colours congruent with their meaning, and the time taken to read a set of words with colours incongruent meaning. This matches my personal expectation having completed the Stroop test.

2.0.8 What might be responsible for the effects observed?

The lag that a person experiences when reading words under the incongruent condition compared to the congruent condition could be caused by a few things. Here are some possible causes:

- **Relative speeed of processing** - words are read faster than colours are named.
- **Automaticity** - reading is in some sense *automatic* but naming colours is not. We associate words very strongly with their spoken responses.
- **Response competition** - the naming of incogruent words interferes with the naming of the colour, slowing down the response time.

2.0.9 Finding sImilar effects from other possible experiments

- To test *relative speed of processing*, measure and compare the time it takes to read the number symbols compared to words (e.g. “2” compared to “two”)

- To test *Automaticity*, measure and compare the time it takes to identify common symbols for concrete objects, compared to artistic representations (e.g. a portrait of a man compared to a 'male toilets' man symbol).
- To test *response competition* as a theory, ask students to read colours whilst listening to words read of different colours.

One other interesting extension of the Stroop effect is the *emotional stroop effect.* Compare times it takes to read "neutral" words compared to "emotionally laden words," e.g. "hairy", "crawl" for spider phobic participants.

2.0.10 References

- http://stackoverflow.com/questions/6871201/plot-two-histograms-at-the-same-time-with-matplotlib
- http://blog.bharatbhole.com/creating-boxplots-with-matplotlib/
- http://worthylab.tamu.edu/Courses_files/Stroop%20Effect.pdf