※ ***Created by Ilho Song (ilho.p.song@gmail.com), December 2016.***

- Note: *italics* are the paragraphs I borrowed from Udacity's project descirptition.

Stroop task

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 equallysized lists. Each participant will go through and record a time from each condition.

In [1]:
```
%%html
<style>
table {float:left}
</style>
```

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

- The task consists of two parts: congruent words test, and incongruent words test. Each of these tests can be considered as two conditions test design. In this case our independent variable would be **whether we use congruent words or incongruent words for the test**.
- Also our dependent variable would be **the number of seconds (time measurement) to finish each tests** per each participants.
- As our independent variable is varying as a participant takes the whole test, the experimental design is **repeated measures design with two conditions**.

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

- Let say the mean of the number of seconds it takes for the congruent words test be $\bar{X_c}$ and one for the incongruent words test be $\bar{X_i}$.
- With the independend variable and dependent variable stated above, we can set the null hypothesis H_0 and the alternative hypothesis H_A as follows:

$$H_0 : \bar{X_c} = \bar{X_i}$$
$$H_A : \bar{X_c} \neq \bar{X_i}$$

$$or$$

$$H_0 : \bar{X_c} - \bar{X_i} = 0$$
$$H_A : \bar{X_c} - \bar{X_i} \neq 0$$

, so that H_0 represents there is no time difference between the time for congruent words test and incongruent words test, while H_A represent the opposite. In this case we do not assume any directions as we are interested in differences, not which leads to faster test time than other. If we define $\bar{X}_{c-i} = \bar{X_c} - \bar{X_i}$ then:

$$H_0 : \bar{X}_{c-i} = 0$$
$$H_A : \bar{X}_{c-i} \neq 0$$

With these hypothesis, **two sample dependent two-tailed t-test can be performed**.

It needs to be t-test than z-test as we do not know about the population statistics but only sample statistics. It is two sample test as we have two different samples with each condition (congruent and incongruent). It is dependent since each participants for both test conditions are the same person, doing one test after another; thus they are not independent. It is a two-tail test, as the alternative hypothesis is interested in differences between samples not which one is faster or slower.

Now it's your chance to try out the Stroop task for yourself. Go to this link, which has a Java based applet for performing the Stroop task. Record the times that you received on the task (you do not need to submit your times to the site.) Now, download this dataset which contains results from a number of participants in the task. Each row of the dataset contains the performance for one participant, with the first number their results on the congruent task and the second number their performance on the incongruent task.

Let's read the given data from the link (https://www.google.com/url?q=https://drive.google.com/file/d/0B9Yf01UaIbUgQXpYb2NhZ29yX1U/view?usp%3Dsharing&sa=D&ust=1481322308972000&usg=AFQjCNGki-zewfEuKXItXfKZi3YxE0bBNA) provided on the project description. I downloaded the stroopdata.csv (stroopdata.csv) file and put it on the same directory as this notebook.

In [2]:
```
import pandas
import numpy
from IPython.display import display

data = pandas.read_csv('stroopdata.csv')
print 'total {0} lines read'.format(numpy.shape(data)[0])
data.head()
```

```
total 24 lines read
```

Out[2]:

	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

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

Let's get some descriptive statistics.

In [3]:
```
ns = pandas.Series([len(data), len(data)], list(data.columns))
medians = numpy.median(data, axis=0)
means = numpy.mean(data, axis=0)
stds = numpy.std(data, ddof=1)
meanDiff = means[0] - means[1]

print 'data size:'
print ns
print 'data medians:'
print medians
print 'data means:'
print means
print 'data standard deviations:'
print stds
print 'mean difference'
print meanDiff
```

```
data size:
Congruent      24
Incongruent    24
dtype: int64
data medians:
[ 14.3565  21.0175]
data means:
Congruent      14.051125
Incongruent    22.015917
dtype: float64
data standard deviations:
Congruent      3.559358
Incongruent    4.797057
dtype: float64
mean difference
-7.96479166667
```

Using above code, we can get data size, median values per test, mean values per test, and their standard deviations as follows:

	Congruent	Incongruent
Size (n)	24	24
Median	14.3565	21.0175
Mean ($\bar{X}$)	14.051125	22.015917
Standard Deviations (σ)	3.559358	4.797057

$$Therefore,$$

$$n_c = 24,\ n_i = 24$$
$$\bar{X_c} = 14.051125,\ \bar{X_i} = 22.015917$$
$$\sigma_c = 3.559358,\ \sigma_i = 4.797057$$

$$\bar{X_c} - \bar{X_i} = -7.96479166667$$

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.

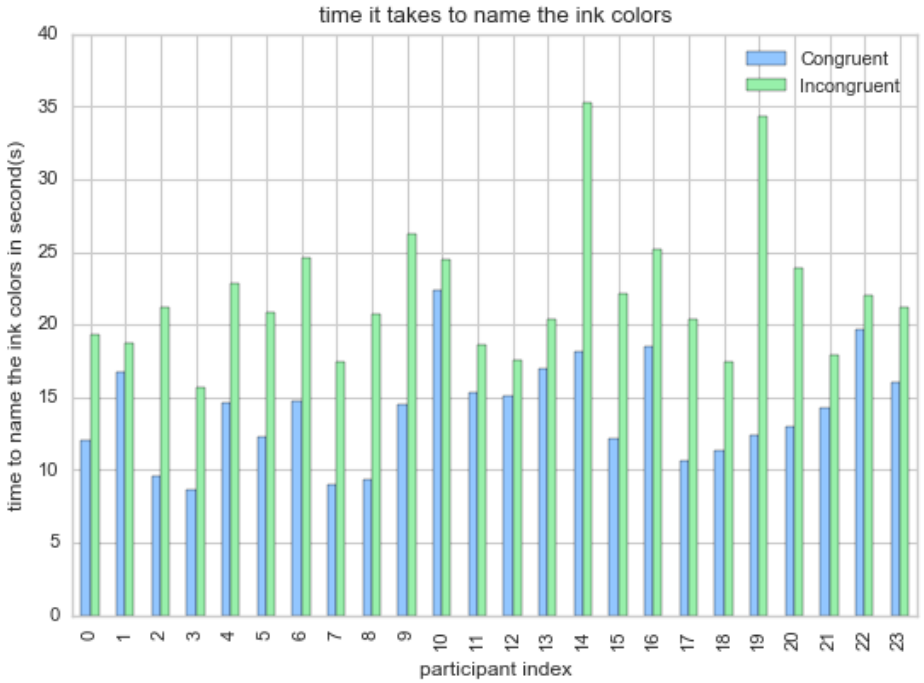
Let's start with a simple plotting.

Graph 1. Time it takes to name the ink colors per test set

In [4]:
```
import matplotlib
import matplotlib.pyplot as plt
%matplotlib inline
```

In [12]:
```
plt.style.use(['seaborn-paper', 'seaborn-whitegrid', 'seaborn-pastel'])
pl = data.plot(kind='bar', title='time it takes to name the ink colors')
pl.set_xlabel('participant index')
pl.set_ylabel('time to name the ink colors in second(s)')
```

Out[12]:
```
<matplotlib.text.Text at 0xc397f98>
```

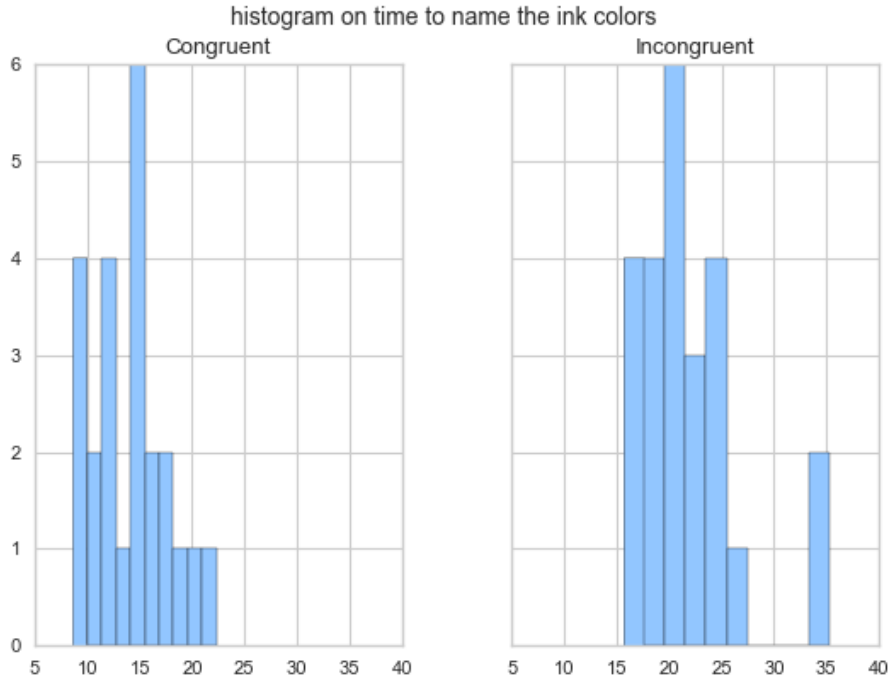


Clearly, this graph shows that for all 24 participants (index 0 to 23) they got finished the test **faster** for the congruent word sets than incongruent word sets, although the differences are vary per participants. This is as expected since the mismatch with the word and the color would make a participant to name its color harder.

Graph 2. Histogram

In [28]:
```
hist = data.hist(sharex=True, sharey=True)
plt.suptitle('histogram on time to name the ink colors')
```

Out[28]:
```
<matplotlib.text.Text at 0xf74f7f0>
```



This graph shows again, as like the above graph 1, that incongruent results are mostly larger than congruent results, which means in general it took more time to name the ink color when the word does not match with it.

Also we can see, from above histogram, that both congruent and incongruent results are **postively skewed**.

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?

Point estimation, degrees of freedom, and the standard deviation

In [39]:
```
import math

pointEstimation = means[0] - means[1]
dof = ns[0] - 1
sd = math.sqrt(stds[0]*stds[0] + stds[1]*stds[1])

print 'Point Estimation: {0}'.format(pointEstimation)
print 'dof: {0}'.format(dof)
print 'sd: {0}'.format(sd)
```

```
Point Estimation: -7.96479166667
dof: 23
sd: 5.97333961088
```

Our point estimation is $\bar{X}_{c-i} = \bar{X}_c - \bar{X}_i = -7.96479166667$, and degrees of freedom is $df = n - 1 = 23$. The standard deviation for the two samples is $SD = \sqrt{S_1^2 + S_2^2} = 5.97333961088$.

t-critical value

Now we need the t-critical value for our test. Let's use alpha level $\alpha = 0.05$.

In [45]:
```
# getting t-critical value for alpha=0.05
import scipy.stats as stats

alpha = 0.05
tcritical = abs(stats.t.ppf(alpha / 2, dof))  # divided by 2, as it is a two-tail test

print 'tcritical: ±{0}'.format(tcritical)   #  ± as it is a two-tail test
```

```
tcritical: ±2.06865761042
```

Although we can get the $t - critical$ value from the t-table (t-table.jpg) from the course, for more accurate value I have used **scipy.stats** package of Python. The obtained value is

$$t - critical = \pm 2.06865761042$$

(for the reference, the linked t-table (t-table.jpg) image will give you 2.069 for $p = \frac{\alpha}{2} = 0.25,\ df = 23$)

t-value

Since we have our values ready, let's get the $t - value$. First we need to get the standard error, SE.

In [62]:
```
se = sd / math.sqrt(ns[0])
print 'se: {0}'.format(se)
```

```
se: 1.21930284225
```

$$SE = \frac{SD}{\sqrt{n}} = \frac{\sqrt{S_1^2 + S_2^2}}{\sqrt{n}} = \frac{5.97333961088}{\sqrt{24}} = 1.21930284225$$

And the $t - value$ could be obtained as follows:

In [49]:
```
tvalue = pointEstimation / se
print 'tvalue: {0}'.format(tvalue)
```

```
tvalue: -6.5322505539
```

$$t = \frac{\bar{X}_c - \bar{X}_i}{\frac{SD}{\sqrt{n}}} = \frac{\bar{X}_{c-i}}{SE} = \frac{-7.96479166667}{1.21930284225} = -6.5322505539$$

p-value

Finally, we can also calculate the $p - value$ using scipy.stats.

In [61]:
```
pvalue = abs(stats.t.sf(abs(tvalue), dof))
print 'pvalue: {0:.9f}'.format(pvalue)  # only 9 decimals as it is really small!
```

```
pvalue: 0.000000576
```

Margin of error and confidence Interval

In [64]:
```
moe = se * tcritical
ci = [pointEstimation-moe, pointEstimation+moe]

print 'Margin of error: {0}'.format(moe)
print '{0}% CI: ({1}, {2})'.format(100* (1-alpha), min(ci), max(ci))
```

```
Margin of error: 2.52232010403
95.0% CI: (-10.4871117707, -5.44247156264)
```

$$Margin\ of\ error\ = SE * (t - critical) = 1.21930284225 * \pm 1.21930284225 = \pm 2.52232010403$$
$$95\%CI : (\bar{X}_{c-i} - MoE,\ \bar{X}_{c-i} + MoE) = (-10.4871117707, -5.44247156264)$$

r^2 value

In [66]:
```
t_sq = tvalue * tvalue
r_sq = t_sq / (t_sq + dof)
print 'r_squared: {0}'.format(r_sq)
```

```
r_squared: 0.649765556941
```

Conclusion

Now let's list out the values we have calculated so far:

$$\bar{X}_{c-i} = -7.96479166667$$
$$SE = 1.21930284225$$
$$t - critical = \pm 2.06865761042$$
$$t - value = -6.5322505539$$
$$p - value \approx 0.000000576$$
$$Margin\ of\ error = 2.52232010403$$
$$95\%CI : (-10.4871117707, -5.44247156264)$$
$$r^2 = 0.649765556941$$

Obviously, t-value is less than the negative t-critical value. Therefore, we can say that the differences in these two samples are statistically significant at $p < 0.05$. In other words, the null hypothesis H_0 **is rejected** or the alternative hypothesis H_A **is accepted** at $p < 0.05$.

From the r^2, we can infer about **65% of the differences are due to the condition chanage**, in this case whether the words are congruent or incongruent.

References

- Udacity courses for STATISTICS, https://classroom.udacity.com/nanodegrees/nd002/syllabus?activePartKey=0021345402 (https://classroom.udacity.com/nanodegrees/nd002/syllabus?activePartKey=0021345402)
- SciPy documentation, https://docs.scipy.org/doc/scipy/reference/ (https://docs.scipy.org/doc/scipy/reference/), v0.18.1, September 19, 2016
- NumPy documentation, https://docs.scipy.org/doc/numpy/ (https://docs.scipy.org/doc/numpy/), v.1.11.0, May 29, 2016
- matplotlib documentation, http://matplotlib.org/contents.html (http://matplotlib.org/contents.html), v.1.5.3, December 5, 2016
- IPython documentation, http://ipython.org/ipython-doc/stable/index.html (http://ipython.org/ipython-doc/stable/index.html), v.3.2.1, September 25, 2015