In [1]:
```
import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
import scipy.stats as stats
import math
import os
%matplotlib inline
```

In [2]:
```
# Sets path for data files and reads them into a dictionary
path = '~/Documents/ME365/Lab4/'
files = ['Lab4_420G4.xlsx','Lab4_440AG4.xlsx','Lab4_Nimark_300G4.xlsx','Lab4_C
dfKeys = ['dat420','dat440','dat455','dat300']

dfs = {}
for df,filename in zip(dfKeys,files):
    dfs[df]=(pd.read_excel(path+filename))
```

In [3]:
```
dfs
```

Out[3]:
```
{'dat300':    Sample  Thickness  %Reduction   1   2   3     4 scale
 0       1      0.091         NaN  30  30  30  30.0     C
 1       2      0.080   12.087912  30  30  31  30.0     C
 2       3      0.072   20.879121  30  29  30  31.0     C
 3       4      0.053   41.758242  31  31  30  32.0     C
 4       5      0.048   47.252747  34  33  34  35.5     C,
 'dat420':    Sample  Thickness  %Reduction     1   2     3   4 scale
 0       1      0.258         NaN  93.5  95  95.0  93     B
 1       2      0.237    8.139535  93.5  99  98.5  99     B
 2       3      0.213   17.441860  21.0  20  21.0  20     C
 3       4      0.154   40.310078  26.0  26  26.0  26     C
 4       5      0.106   58.914729  30.0  30  30.0  30     C,
 'dat440':    Sample  Thickness  %Reduction   1   2   3   4 scale
 0       5      0.084         NaN  96  97  97  97     B
 1       4      0.077    8.333333  22  22  22  22     C
 2       3      0.065   22.619048  26  27  25  25     C
 3       2      0.052   38.095238  28  27  28  31     C
 4       1      0.040   52.380952  29  29  30  29     C,
 'dat455':    Sample  Thickness  %Reduction   1   2   3   4 scale
 0       1      0.080         NaN  32  32  33  31     C
 1       2      0.072       34.00  32  32  33  34     C
 2       3      0.065       18.75  35  34  35  35     C
 3       4      0.057       28.75  32  33  33  32     C
 4       5      0.043       46.25  32  33  34  32     C}
```

In [4]:
```
# Convert all Rockwell B to C for 440 data
dfs['dat440'].iloc[0,3] = 18
dfs['dat440'].iloc[0,4:7] = 20

# Convert all Rockwell B to C for
def interpolate(x1,x2,y1,y2,X):

    return y1 + (y1 - y2)*(X - x1)/(x1 - x2)

x1 = 93
x2 = 94
X = 93.5
y1 = 13
y2 = 15
dfs['dat420'].iloc[0:2,3] = interpolate(x1,x2,y1,y2,X)
x1 = 98
x2 = 99
X = 98.5
y1 = 21
y2 = 22
dfs['dat420'].iloc[1,5] = interpolate(x1,x2,y1,y2,X)
dfs['dat420'].iloc[0,4:6] = 16.
dfs['dat420'].iloc[0,6] = 13.
dfs['dat420'].iloc[1,4] = 22.
dfs['dat420'].iloc[1,6] = 22.
dfs['dat420'].iloc[0:2,7] = 'C'
dfs['dat440'].iloc[0,7] = 'C'
```

In [5]:
```
# Retrieve t-stat value from the "chart" TINV(a,v) 95% confidence interval
alpha = .05
t = stats.t.ppf(1-alpha/2,df=6)
print t
```

```
2.4469118487916806
```

In [6]:
```
for df in dfKeys:
    print dfs[df]
```

```
   Sample  Thickness  %Reduction     1     2     3     4 scale
0       1      0.258         NaN  14.0  16.0  16.0  13.0     C
1       2      0.237    8.139535  14.0  22.0  21.5  22.0     C
2       3      0.213   17.441860  21.0  20.0  21.0  20.0     C
3       4      0.154   40.310078  26.0  26.0  26.0  26.0     C
4       5      0.106   58.914729  30.0  30.0  30.0  30.0     C
   Sample  Thickness  %Reduction   1   2   3   4 scale
0       5      0.084         NaN  18  20  20  20     C
1       4      0.077    8.333333  22  22  22  22     C
2       3      0.065   22.619048  26  27  25  25     C
3       2      0.052   38.095238  28  27  28  31     C
4       1      0.040   52.380952  29  29  30  29     C
   Sample  Thickness  %Reduction   1   2   3   4 scale
0       1      0.080         NaN  32  32  33  31     C
1       2      0.072       34.00  32  32  33  34     C
2       3      0.065       18.75  35  34  35  35     C
3       4      0.057       28.75  32  33  33  32     C
4       5      0.043       46.25  32  33  34  32     C
   Sample  Thickness  %Reduction   1   2   3     4 scale
0       1      0.091         NaN  30  30  30  30.0     C
1       2      0.080   12.087912  30  30  31  30.0     C
2       3      0.072   20.879121  30  29  30  31.0     C
3       4      0.053   41.758242  31  31  30  32.0     C
4       5      0.048   47.252747  34  33  34  35.5     C
```

In [7]:
```
# This mess creates a function that does the rest of the statistcal calculatio
def trueMean(dfs,n1,n2):
    alpha = .05
    t = stats.t.ppf(1.-alpha/2.,df=6)
    #print t
    v1 = n1-1
    v2 = n2-1
    df = n1+n2-2
    v = v1+v2
    for df in dfs.keys():
        df = dfs[df]
        df['mean_hardness'] = 0.
        df['std'] = 0.
        df['Sx'] = 0.
        df['true_mean+'] = 0.
        df['true_mean-'] = 0.

        for i in range(5):
            df['mean_hardness'].loc[i] = sum(df.iloc[i,3:7])/4.
            df['std'].iloc[i] = df.iloc[i,3:7].std()
            df.Sx.iloc[i] = (v1*df['std'].iloc[0]+v2*df['std'].iloc[i])/(v)
    for df in dfs.keys():
        df = dfs[df]

        for i in range(5):

            if i >= 1:
                df['true_mean+'].iloc[i] = df['mean_hardness'].iloc[0]-df['mea
                df['true_mean-'].iloc[i] = df['mean_hardness'].iloc[0]-df['mea

            else:
                df['true_mean+'].iloc[i] = 'NaN'
                df['true_mean-'].iloc[i] = 'NaN'

    return('Statiscal Analysis Successful\n\n')
```

In [8]:
```
# This
#dfs = [dat420,dat440,dat455,dat300]
print trueMean(dfs,5.,5.)
#print trueMean(dfs,5.,5.)
print 'dat420\n\n',dfs['dat420']
print '\ndat440\n\n',dfs['dat440']
print '\ndat455\n\n',dfs['dat455']
print '\ndat300\n\n',dfs['dat300']
```

```
/home/scott/.local/lib/python2.7/site-packages/pandas/core/indexing.py:190: S
ettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame

See the caveats in the documentation: http://pandas.pydata.org/pandas-docs/st
able/indexing.html#indexing-view-versus-copy (http://pandas.pydata.org/pandas
-docs/stable/indexing.html#indexing-view-versus-copy)
  self._setitem_with_indexer(indexer, value)

Statiscal Analysis Successful


dat420

   Sample  Thickness  %Reduction     1     2     3     4 scale  mean_hardness
\
0       1      0.258         NaN  14.0  16.0  16.0  13.0     C          14.750
1       2      0.237    8.139535  14.0  22.0  21.5  22.0     C          19.875
2       3      0.213   17.441860  21.0  20.0  21.0  20.0     C          20.500
3       4      0.154   40.310078  26.0  26.0  26.0  26.0     C          26.000
4       5      0.106   58.914729  30.0  30.0  30.0  30.0     C          30.000

        std        Sx true_mean+ true_mean-
0  1.500000  1.500000        NaN        NaN
1  3.923752  2.711876  -0.928201    -9.3218
2  0.577350  1.038675   -4.14258   -7.35742
3  0.000000  0.750000   -10.0893   -12.4107
4  0.000000  0.750000   -14.0893   -16.4107

dat440

   Sample  Thickness  %Reduction   1   2   3   4 scale  mean_hardness  \
0       5      0.084         NaN  18  20  20  20     C          19.50
1       4      0.077    8.333333  22  22  22  22     C          22.00
2       3      0.065   22.619048  26  27  25  25     C          25.75
3       2      0.052   38.095238  28  27  28  31     C          28.50
4       1      0.040   52.380952  29  29  30  29     C          29.25

        std        Sx true_mean+ true_mean-
0  1.000000  1.000000        NaN        NaN
1  0.000000  0.500000   -1.72622   -3.27378
2  0.957427  0.978714   -4.73538   -7.76462
3  1.732051  1.366025   -6.88599    -11.114
4  0.500000  0.750000   -8.58933   -10.9107

dat455

   Sample  Thickness  %Reduction   1   2   3   4 scale  mean_hardness  \
0       1      0.080         NaN  32  32  33  31     C          32.00
1       2      0.072       34.00  32  32  33  34     C          32.75
2       3      0.065       18.75  35  34  35  35     C          34.75
3       4      0.057       28.75  32  33  33  32     C          32.50
4       5      0.043       46.25  32  33  34  32     C          32.75
```

In [9]:
```
path = '~/Documents/ME365/Lab4/'
files = ['Lab4_420G4_stat.xlsx','Lab4_440AG4_stat.xlsx','Lab4_Custom_455G4_sta
for df,fileName in zip(dfs.keys(),files):
    dfs[df].to_excel(path + fileName)
```