1 Multiple comparisons between groups

If an ANOVA test has identified that not all groups belong to the same population, then methods may be used to identify which groups are significantly different to each other.

Below are two commonly used methods: Tukey's and Holm-Bonferroni.

These two methods assuming data that is approximately normally distributed.

1.1 Setting up the data, and running an ANOVA

```
import numpy as np
import scipy.stats as stats

# Create four random groups of data with a mean difference of 1

mu, sigma = 10, 3 # mean and standard deviation
group1 = np.random.normal(mu, sigma, 50)

mu, sigma = 11, 3 # mean and standard deviation
group2 = np.random.normal(mu, sigma, 50)

mu, sigma = 12, 3 # mean and standard deviation
group3 = np.random.normal(mu, sigma, 50)

mu, sigma = 13, 3 # mean and standard deviation
group4 = np.random.normal(mu, sigma, 50)

# Show the results for Anova

F_statistic, pVal = stats.f_oneway(group1, group2, group3, group4)

print ('P value:')
print (pVal)

OUT:

P value:
1.6462001201818463e-08
```

For the multicomparison tests we will put the data into a dataframe. And then reshape it to a stacked dataframe

```
# Put into dataframe

df = pd.DataFrame()
df['treatment1'] = group1
df['treatment2'] = group2
df['treatment3'] = group3
df['treatment4'] = group4

# Stack the data (and rename columns):

stacked_data = df.stack().reset_index()
stacked_data = stacked_data.rename(columns={'level_0': 'id',
                                            'level_1': 'treatment',
                                            0:'result'})
# Show the first 8 rows:
```

```
print (stacked_data.head(8))

OUT:

   id   treatment     result
0   0  treatment1  12.980445
1   0  treatment2   8.444603
2   0  treatment3  10.713692
3   0  treatment4  10.777762
4   1  treatment1  14.350560
5   1  treatment2   9.436072
6   1  treatment3  12.715509
7   1  treatment4  15.016419
```

1.2 Tukey's multi-comparison method

See https://en.wikipedia.org/wiki/Tukey's_range_test

This method tests at P¡0.05 (correcting for the fact that multiple comparisons are being made which would normally increase the probability of a significant difference being identified). A results of 'reject = True' means that a significant difference has been observed.

```
from statsmodels.stats.multicomp import (pairwise_tukeyhsd,
                                         MultiComparison)

# Set up the data for comparison (creates a specialised object)
MultiComp = MultiComparison(stacked_data['result'],
                            stacked_data['treatment'])

# Show all pair-wise comparisons:

# Print the comparisons

print(MultiComp.tukeyhsd().summary())

OUT:

 Multiple Comparison of Means - Tukey HSD,FWER=0.05
===================================================
  group1     group2   meandiff  lower  upper  reject
---------------------------------------------------
treatment1 treatment2  1.5021  -0.0392 3.0435 False
treatment1 treatment3   1.47   -0.0714 3.0113 False
treatment1 treatment4  3.8572   2.3159 5.3985  True
treatment2 treatment3 -0.0322  -1.5735 1.5091 False
treatment2 treatment4  2.355    0.8137 3.8963  True
treatment3 treatment4  2.3872   0.8459 3.9285  True
```

1.3 Holm-Bonferroni Method

See: https://en.wikipedia.org/wiki/Holm%E2%80%93Bonferroni_method

The Holm-Bonferroni method is an alterantive method.

```
comp = MultiComp.allpairtest(stats.ttest_rel, method='Holm')
print (comp[0])

OUT:
```

```
Test Multiple Comparison ttest_rel
FWER=0.05 method=Holm
alphacSidak=0.01, alphacBonf=0.008
```

group1	group2	stat	pval	pval_corr	reject
treatment1	treatment2	-2.1234	0.0388	0.0776	False
treatment1	treatment3	-2.4304	0.0188	0.0564	False
treatment1	treatment4	-6.4443	0.0	0.0	True
treatment2	treatment3	0.0457	0.9637	0.9637	False
treatment2	treatment4	-3.7878	0.0004	0.0017	True
treatment3	treatment4	-5.0246	0.0	0.0	True