Normalized Opus -- Functional Level Comparision

This notebook computes the significance of the normalized opus score for the following:

1. Electric Hook Type TD
   - vs. all types of Electric Hand
   - vs. Electric single grip hand
   - vs. body-powered TD
   - vs. Electric multi-articulating hand
2. Motion Control ETD
   - vs. electric hooks
3. Multi-articulating hands
   - vs. Electric single grip hand
   - vs. Body-powered TD
4. Check T-test
   - Anova
   - Tukey Tests

In [31]:

```
import pandas as pd
import numpy as np
from scipy import stats
import matplotlib.pyplot as plt
from driverCleanData import cleanData
%matplotlib inline
[sht,_] = cleanData();
```

Let's look at the summary stats for all data on the normalized opus

In [32]:

```
sht.normalizedOpus.dropna().describe()
```

Out[32]:

```
count    1221.000000
mean        3.179006
std         0.416217
min         2.166667
25%         2.833333
50%         3.111111
75%         3.437500
max         4.000000
Name: normalizedOpus, dtype: float64
```

Since the opus is normalized, this means that only the activities the respondents use the prosthesis on are counted. Thus the maximum score possible is 4 and the lowest is 0. Here we see a mean of 3.17 and a standard deviation of 0.41, which may be interpreted as: The tasks which the respondents use their prosthesis for, they typically find the task slightly difficult.

Electric Hook TD

vs. all types of Electric Hand

In [2]:
```
from vectorSubsets import electricHooks, electricHandMA, electricHandSG
print("Electric hooks are: {0}".format(electricHooks))
print()
print("Electric hands - multi articulating are {0}".format(electricHandMA))
print()
print("Electric hands - single grip are {0}".format(electricHandSG))
```

```
Electric hooks are: ['Motion Control (MC) ETD', 'Otto Bock Axon Hook', 'Otto Bock Greif
er']

Electric hands - multi articulating are ['bebionic Hand', 'i-limb Hand', 'Michelangelo
 Hand']

Electric hands - single grip are ['Motion Control (MC) Hand', 'Otto Bock Hand']
```

In [3]:
```
sht["opusEHo"] = sht.normalizedOpus[sht["PrimaryTerminalDevice"].isin(electricHooks)]
sht["opusAllEHa"] = sht.normalizedOpus[sht["PrimaryTerminalDevice"].isin(electricHandMA
or electricHandSG)].dropna()
```

Summary Stats

First up the electric hooks

In [4]:
```
sht.opusEHo.dropna().describe()
```

Out[4]:
```
count    117.000000
mean       3.058773
std        0.372775
min        2.363636
25%        2.777778
50%        3.000000
75%        3.250000
max        4.000000
Name: opusEHo, dtype: float64
```

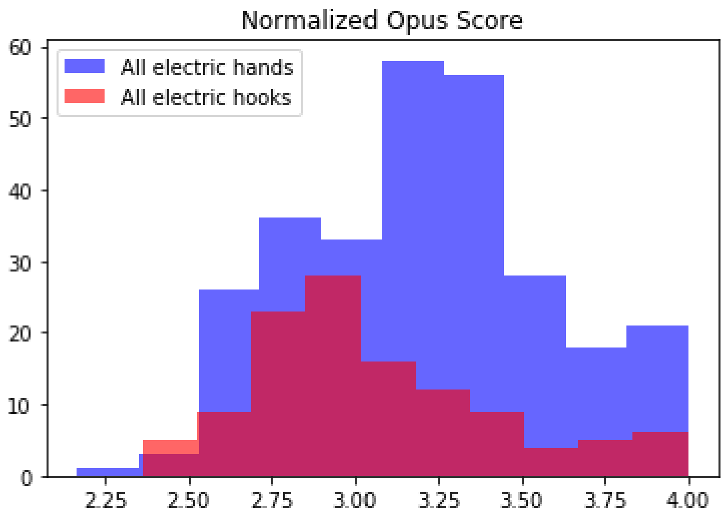
Now for the electric hands

In [5]:
```
sht.opusAllEHa.dropna().describe()
```

Out[5]:
```
count    280.000000
mean       3.211556
std        0.378506
min        2.166667
25%        2.900000
50%        3.242647
75%        3.437500
max        4.000000
Name: opusAllEHa, dtype: float64
```

Histogram

In [36]:
```
plt.hist(sht.opusAllEHa.dropna(), alpha=0.6, color='b', label = "All electric hands")
plt.hist(sht.opusEHo.dropna(), alpha=0.6, color='r', label = "All electric hooks")
plt.title("Normalized Opus Score")
plt.legend()
plt.show()
```



In [7]:
```
stats.ttest_ind(sht.opusAllEHa.dropna(), sht.opusEHo.dropna())
```

Out[7]:
```
Ttest_indResult(statistic=3.6830158986463086, pvalue=0.00026258021363330305)
```

The higher mean score on Normalized Opus of electric hands compared to electric hooks is significant.

vs. Electric single grip hand

In [8]:
```
sht["opusElecSG"] =
sht.normalizedOpus[sht["PrimaryTerminalDevice"].isin(electricHandSG)]
print("The types of devices we are considering as electric hands are {0}".format(electri
cHandSG))
print()
print("The mean of the electric hooks opus is {0}".format(sht.opusEHo.dropna().mean()))
print()
print("The summary stats for the single grip are")
print(sht.opusElecSG.dropna().describe())
```

```
The types of devices we are considering as electric hands are ['Motion Control (MC) Han
d', 'Otto Bock Hand']

The mean of the electric hooks opus is 3.058772974443373

The summary stats for the single grip are
count    607.000000
mean       3.214464
std        0.437060
min        2.333333
25%        2.888889
50%        3.100000
75%        3.535885
max        4.000000
Name: opusElecSG, dtype: float64
```

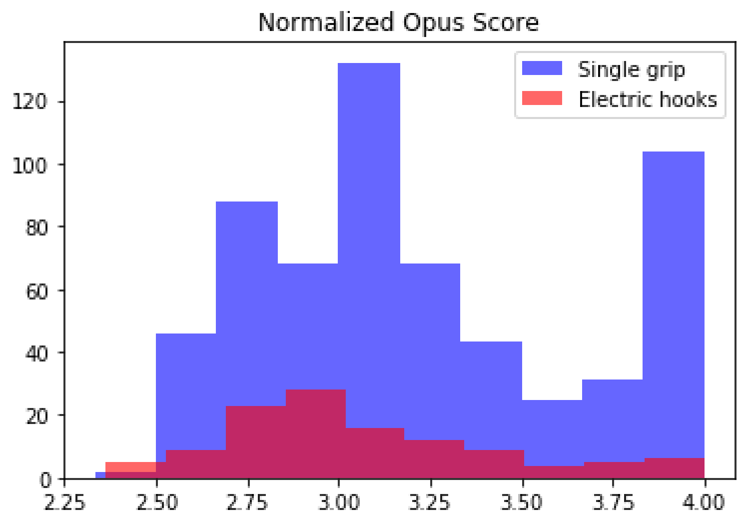
In [9]:
```
stats.ttest_ind(sht.opusEHo.dropna(), sht.opusElecSG.dropna())
```

Out[9]:
```
Ttest_indResult(statistic=-3.6079645954487964, pvalue=0.0003298957487412504)
```

The higher mean score for single grip hands is signficant.

Histograms

In [35]:
```
plt.hist(sht.opusElecSG.dropna(), color='b', alpha=0.6, label = "Single grip")
plt.hist(sht.opusEHo.dropna(), color='r', alpha=0.6, label = "Electric hooks")
plt.title("Normalized Opus Score")
plt.legend()
plt.show()
```



vs. Body Powered Hooks

In [11]:
```
from vectorSubsets import bodyPoweredHooks
sht["opusBPHook"] = sht.normalizedOpus[sht["PrimaryTerminalDevice"].isin(bodyPoweredHook
s)]
sht.opusBPHook.dropna().describe()
```

Out[11]:
```
count    185.000000
mean       3.064431
std        0.401842
min        2.375000
25%        2.750000
50%        2.900000
75%        3.350000
max        4.000000
Name: opusBPHook, dtype: float64
```

In [12]:
```
stats.ttest_ind(sht.opusEHo.dropna(), sht.opusBPHook.dropna())
```

Out[12]:
```
Ttest_indResult(statistic=-0.12256137888705929, pvalue=0.90253650099781646)
```

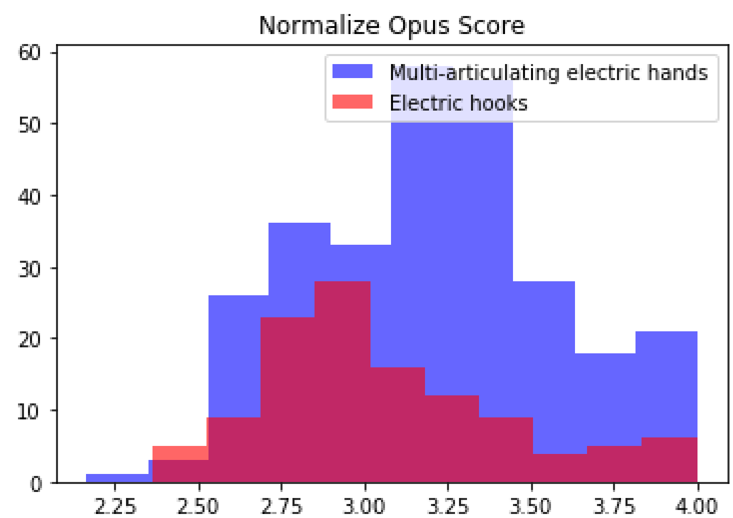
Can't reject the null at alpha=0.05, so there might not be a significant difference between body powered and electric hooks on the normalized opus score

vs. Multi-articulating Hands

In [13]:
```
from vectorSubsets import electricHandMA
sht["opusMA"] = sht.normalizedOpus[sht["PrimaryTerminalDevice"].isin(electricHandMA)]
print("The summary stats for multi-articulating are:")
print(sht.opusMA.dropna().describe())
```

```
The summary stats for multi-articulating are:
count    280.000000
mean       3.211556
std        0.378506
min        2.166667
25%        2.900000
50%        3.242647
75%        3.437500
max        4.000000
Name: opusMA, dtype: float64
```

In [37]:
```
plt.hist(sht.opusMA.dropna(), alpha=0.6, color='b', label = "Multi-articulating electric
 hands")
plt.hist(sht.opusEHo.dropna(), alpha=0.6, color='r', label = "Electric hooks")
plt.title("Normalize Opus Score")
plt.legend()
plt.show()
```



In [15]:
```
stats.ttest_ind(sht.opusEHo.dropna(),sht.opusMA.dropna())
```

Out[15]:
```
Ttest_indResult(statistic=-3.6830158986463086, pvalue=0.00026258021363330305)
```

The higher mean score of the multi-articulating is significant compared to electric hooks.

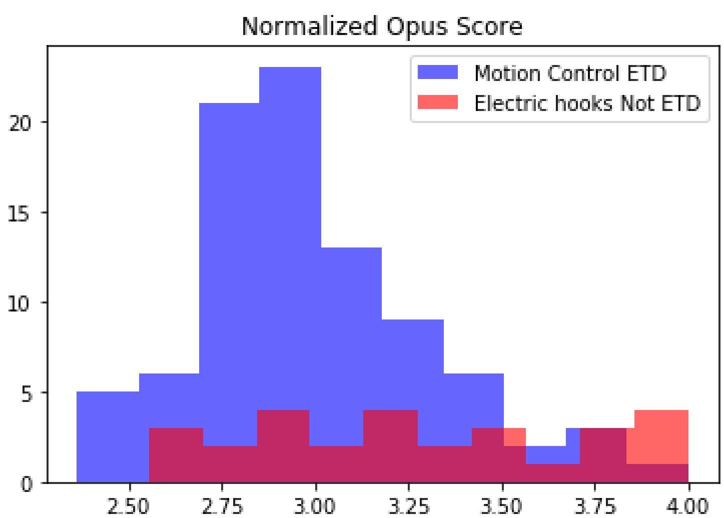
Motion Control ETD vs. All Electric Hooks

```
In [16]: sht["opusETD"] = sht.normalizedOpus[sht["PrimaryTerminalDevice"]=="Motion Control (MC) E
TD"]
from vectorSubsets import electricHooksNoETD
sht["opusElecNoETD"] = sht.normalizedOpus[sht["PrimaryTerminalDevice"].isin(electricHook
sNoETD)]
print("The electric hooks we are considering are: {0}".format(electricHooksNoETD))
print("The number of participants are {0}".format(sht.opusElecNoETD.dropna().count()))
print()
print("The average score on opus for the ETD is
{0}".format(sht.opusETD.dropna().mean()))
print("The average score on opus for the electric hooks is
{0}".format(sht.opusElecNoETD.dropna().mean()))
print()
print(stats.ttest_ind(sht.opusETD.dropna(), sht.opusElecNoETD.dropna()))
```

```
The electric hooks we are considering are: ['Otto Bock Greifer', 'Otto Bock Axon Hook']
The number of participants are 28

The average score on opus for the ETD is 2.9885097542192205
The average score on opus for the electric hooks is 3.282109638727286

Ttest_indResult(statistic=-3.8447600819128045, pvalue=0.00019834505869393984)
```

This result is significant and unfortunately unfavorable for the ETD. However when we look at the histogram we do not see that the non etd hooks distribution is normal so that assumption falls away and we can neither accept nor reject the null hypothesis.

```
In [39]: plt.hist(sht.opusETD.dropna(), alpha=0.6, color='b', label = "Motion Control ETD")
plt.hist(sht.opusElecNoETD.dropna(), alpha=0.6, color='r', label = "Electric hooks Not E
TD")
plt.title("Normalized Opus Score")
plt.legend()
plt.show()
```



Multi-articulating hands

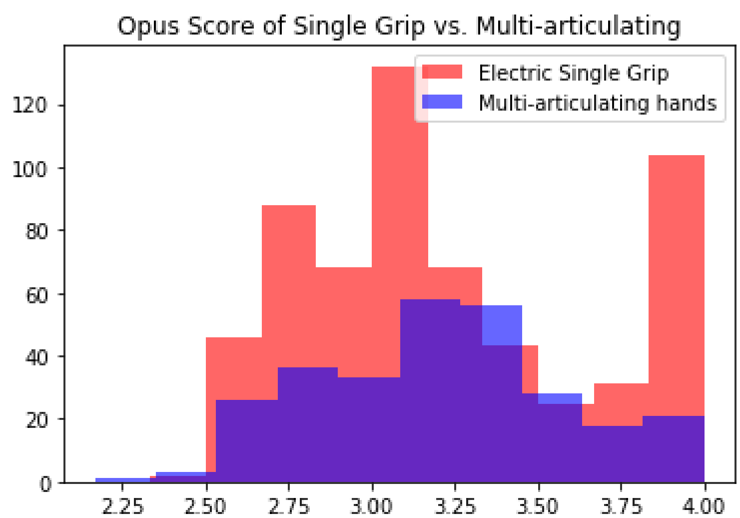
vs. Single Grip hands

In [18]:
```
print("The hands in the multi-articulating group are {0}".format(electricHandMA))
print()
print("The hands in the single grip group are {0}".format(electricHandSG))
print()
print("The average score on opus for the multi-articulating is {0}".format(sht.opusMA.dr
opna().mean()))
print("The average score on opus for the single grip is {0}".format(sht.opusElecSG.dropn
a().mean()))
print()
print(stats.ttest_ind(sht.opusMA.dropna(), sht.opusElecSG.dropna()))
```

```
The hands in the multi-articulating group are ['bebionic Hand', 'i-limb Hand', 'Michela
ngelo Hand']

The hands in the single grip group are ['Motion Control (MC) Hand', 'Otto Bock Hand']

The average score on opus for the multi-articulating is 3.211555785883242
The average score on opus for the single grip is 3.214463690408739

Ttest_indResult(statistic=-0.095956965997671603, pvalue=0.9235764745847912)
```

The higher score on the single grip hands is not significant.

In [19]:
```
plt.hist(sht.opusElecSG.dropna(), alpha=0.6, color='r', label = "Electric Single Grip")
plt.hist(sht.opusMA.dropna(), alpha=0.6, color='b', label = "Multi-articulating hands")
plt.title("Opus Score of Single Grip vs. Multi-articulating")
plt.legend()
plt.show()
```



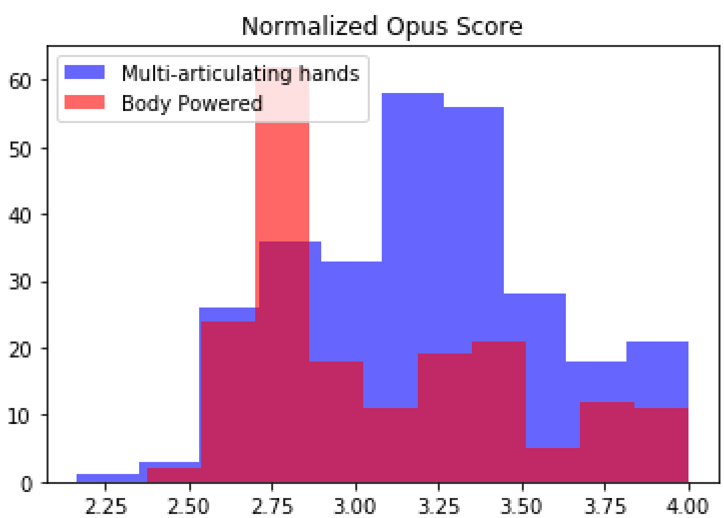
vs. Body Powered

In [20]:
```
print("The hands in the multi-articulating group are {0}".format(electricHandMA))
print()
print("The hands in the body powered group are {0}".format(bodyPoweredHooks))
print()
print("The average score on opus for the multi-articulating is {0}".format(sht.opusMA.dr
opna().mean()))
print("The average score on opus for body powered is
{0}".format(sht.opusBPHook.dropna().mean()))
print()
print(stats.ttest_ind(sht.opusMA.dropna(), sht.opusBPHook.dropna()))
```

```
The hands in the multi-articulating group are ['bebionic Hand', 'i-limb Hand', 'Michela
ngelo Hand']

The hands in the body powered group are ['TRS Hook', 'Hosmer Hook', 'Body-powered Han
d']

The average score on opus for the multi-articulating is 3.211555785883242
The average score on opus for body powered is 3.0644314336729175

Ttest_indResult(statistic=4.0026683585178757, pvalue=7.2923558005955942e-05)
```

The higher score on the normalized opus by the multi-articulating hands is significant.

In [40]:
```
plt.hist(sht.opusMA.dropna(), alpha=0.6, color='b', label = "Multi-articulating hands")
plt.hist(sht.opusBPHook.dropna(), alpha=0.6, color='r', label = "Body Powered")
plt.title("Normalized Opus Score")
plt.legend()
plt.show()
```



Check t-test analysis

The results from t-tests can sometimes lead to too many false positives, anova and Tukey's test accomadate for multiple comparisons. So the below should be a validation of what was observed above.

To recall the T-tests for the following were significant (with the grouping with the higher mean score coming first)

- Electric Hand vs. Electric Hooks
- Single Grip vs. Electric Hooks
- Multi-articulating vs. Electric Hooks
- Multi-articulating vs. Body-powered

ANOVA

Use anova to make sure there is a difference in the means of at least one of the opus scores between the body powered, multi-articulating, single grip, and electric hooks

In [22]:
```
print("The body powered hooks are {0}".format(bodyPoweredHooks))
print()
print("The multi-articulating hands are {0}".format(electricHandMA))
print()
print("The single grip hands are {0}".format(electricHandSG))
print()
print("The electric hooks are {0}".format(electricHooks))
```

```
The body powered hooks are ['TRS Hook', 'Hosmer Hook', 'Body-powered Hand']

The multi-articulating hands are ['bebionic Hand', 'i-limb Hand', 'Michelangelo Hand']

The single grip hands are ['Motion Control (MC) Hand', 'Otto Bock Hand']

The electric hooks are ['Motion Control (MC) ETD', 'Otto Bock Axon Hook', 'Otto Bock Gr
eifer']
```

Now we'll run anova to check that there is a difference among the mean scores on opus of these 4 groups.

In [23]:
```
stats.f_oneway(sht.opusBPHook.dropna(), sht.opusMA.dropna(), sht.opusElecSG.dropna(), sh
t.opusEHo.dropna())
```

Out[23]:
```
F_onewayResult(statistic=10.115388615804502, pvalue=1.3957021019213132e-06)
```

We definetely have have signficance at 0.05.

Tukey

The below will display a graph and a table the table will tell us the mean difference, confidence intervals and if we can reject the null hypothesis at a 95% confidence level

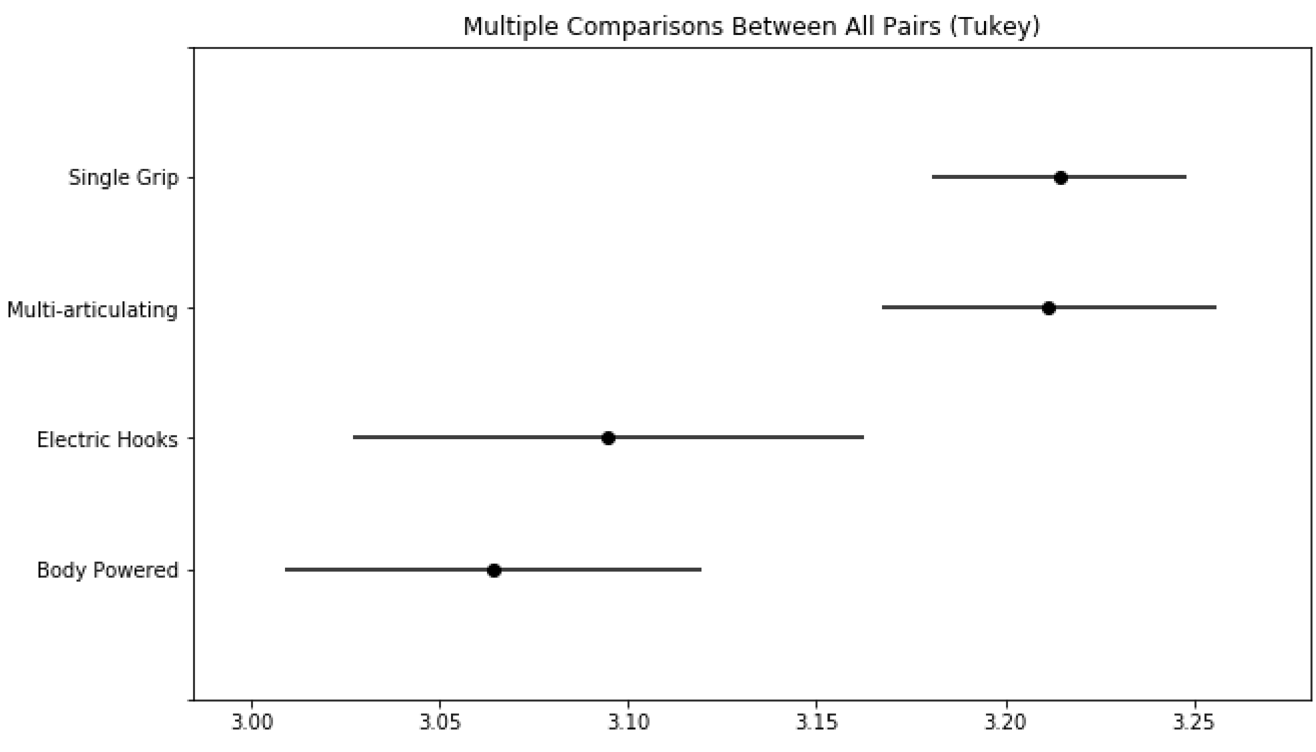
In [24]:
```
opusDf = pd.concat([sht['PrimaryTerminalDevice'], sht['normalizedOpus']], axis=1, keys=
['deviceCategory','normalizedOpus'])
opusDf["deviceCategory"] = opusDf["deviceCategory"].replace([bodyPoweredHooks],"Body Pow
ered")
opusDf["deviceCategory"] = opusDf["deviceCategory"].replace([electricHandMA], "Multi-art
iculating")
opusDf["deviceCategory"] = opusDf["deviceCategory"].replace([electricHandSG], "Single Gr
ip")
opusDf["deviceCategory"] = opusDf["deviceCategory"].replace([electricHooks], "Electric H
ooks")
opusDf["deviceCategory"] = opusDf["deviceCategory"].replace('Otto.*', "Electric Hooks",
regex=True)
opusDf["deviceCategory"] = opusDf["deviceCategory"].replace('Other.*', np.nan, regex=Tru
e)
opusDf["deviceCategory"] = opusDf["deviceCategory"].replace('Pass.*', np.nan,
regex=True)
#groups = opusDf.groupby(opusDf['deviceCategory']).groups
#opusDf['deviceCategory'] = str(opusDf['deviceCategory'])
opusDf = opusDf.dropna()
df = opusDf.as_matrix()
dC = df[:,0].astype(str)
oS = df[:,1].astype(float)
```

In [25]:
```
from ipykernel import kernelapp as app
from statsmodels.stats.multicomp import pairwise_tukeyhsd
tukey = pairwise_tukeyhsd(endog = oS,
                          groups = dC,
                          alpha = 0.05)
tukey.plot_simultaneous()    # Plot group confidence intervals
plt.vlines(x=49.57,ymin=-0.5,ymax=4.5, color="red")

tukey.summary()
```

Out[25]:

Multiple Comparison of Means - Tukey HSD,FWER=0.05

group1	group2	meandiff	lower	upper	reject
Body Powered	Electric Hooks	0.0302	-0.0913	0.1517	False
Body Powered	Multi-articulating	0.1471	0.0463	0.2479	True
Body Powered	Single Grip	0.15	0.0607	0.2394	True
Electric Hooks	Multi-articulating	0.1169	0.0043	0.2295	True
Electric Hooks	Single Grip	0.1198	0.0173	0.2223	True
Multi-articulating	Single Grip	0.0029	-0.074	0.0798	False



At a significance level of 0.05 we are beginning to see that the hands are outperforming electric hooks and body powered hooks on the normalized opus. This confirms what we saw in the T-tests.

Significance among brands

I'm curious if we'll get something by looking at the specific brands

In [26]:
```
opusDf = pd.concat([sht['PrimaryTerminalDevice'], sht['normalizedOpus']], axis=1, keys=
['deviceCategory','normalizedOpus'])
opusDf = opusDf.dropna()
df = opusDf.as_matrix()
dC = df[:,0].astype(str)
oS = df[:,1].astype(float)
tukey = pairwise_tukeyhsd(endog = oS,
                          groups = dC,
                          alpha = 0.05)
tukey.plot_simultaneous()    # Plot group confidence intervals
tukey.summary()
```

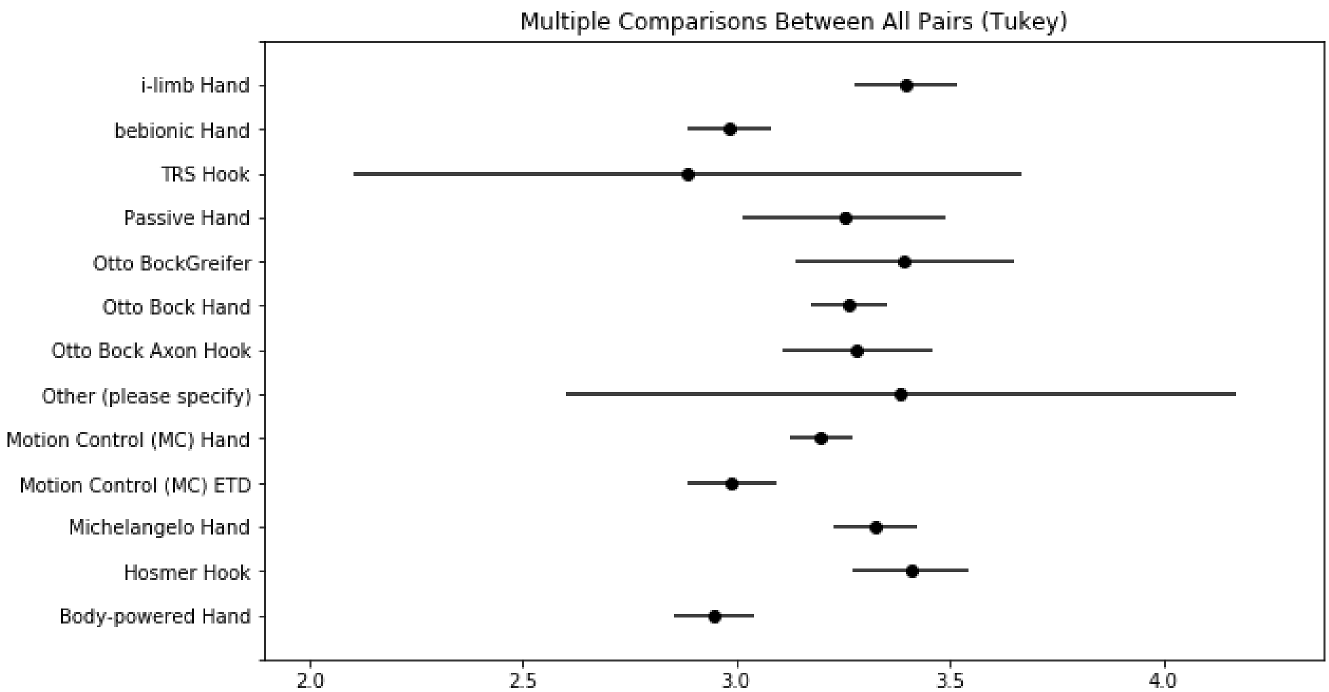
Out[26]:

Multiple Comparison of Means - Tukey HSD,FWER=0.05

group1	group2	meandiff	lower	upper	reject
Body-powered Hand	Hosmer Hook	0.4603	0.2396	0.6809	True
Body-powered Hand	Michelangelo Hand	0.3782	0.2122	0.5442	True
Body-powered Hand	Motion Control (MC) ETD	0.0403	-0.1375	0.2181	False
Body-powered Hand	Motion Control (MC) Hand	0.2509	0.1236	0.3782	True
Body-powered Hand	Other (please specify)	0.4364	-0.4925	1.3654	False
Body-powered Hand	Otto Bock Axon Hook	0.3339	0.0633	0.6046	True
Body-powered Hand	Otto Bock Hand	0.3144	0.1592	0.4695	True
Body-powered Hand	Otto BockGreifer	0.4464	0.0804	0.8125	True
Body-powered Hand	Passive Hand	0.3056	-0.0391	0.6503	False
Body-powered Hand	TRS Hook	-0.0628	-0.9917	0.8662	False
Body-powered Hand	bebionic Hand	0.0343	-0.1347	0.2033	False
Body-powered Hand	i-limb Hand	0.4488	0.2479	0.6498	True
Hosmer Hook	Michelangelo Hand	-0.0821	-0.3085	0.1442	False
Hosmer Hook	Motion Control (MC) ETD	-0.4199	-0.6551	-0.1848	True
Hosmer Hook	Motion Control (MC) Hand	-0.2094	-0.4091	-0.0097	True
Hosmer Hook	Other (please specify)	-0.0238	-0.9654	0.9177	False
Hosmer Hook	Otto Bock Axon Hook	-0.1263	-0.4377	0.185	False
Hosmer Hook	Otto Bock Hand	-0.1459	-0.3644	0.0726	False
Hosmer Hook	Otto BockGreifer	-0.0139	-0.4109	0.3832	False
Hosmer Hook	Passive Hand	-0.1547	-0.5321	0.2228	False
Hosmer Hook	TRS Hook	-0.523	-1.4646	0.4185	False
Hosmer Hook	bebionic Hand	-0.426	-0.6545	-0.1974	True
Hosmer Hook	i-limb Hand	-0.0114	-0.2646	0.2417	False
Michelangelo Hand	Motion Control (MC) ETD	-0.3378	-0.5226	-0.153	True
Michelangelo Hand	Motion Control (MC) Hand	-0.1272	-0.2642	0.0097	False
Michelangelo Hand	Other (please specify)	0.0583	-0.872	0.9886	False
Michelangelo Hand	Otto Bock Axon Hook	-0.0442	-0.3195	0.2311	False
Michelangelo Hand	Otto Bock Hand	-0.0638	-0.2269	0.0994	False
Michelangelo Hand	Otto BockGreifer	0.0683	-0.3012	0.4378	False
Michelangelo Hand	Passive Hand	-0.0725	-0.4209	0.2758	False
Michelangelo Hand	TRS Hook	-0.4409	-1.3712	0.4894	False
Michelangelo Hand	bebionic Hand	-0.3438	-0.5202	-0.1675	True
Michelangelo Hand	i-limb Hand	0.0707	-0.1365	0.2779	False

Motion Control (MC) ETD	Motion Control (MC) Hand	0.2106	0.0596	0.3616	True
Motion Control (MC) ETD	Other (please specify)	0.3961	-0.5364	1.3286	False
Motion Control (MC) ETD	Otto Bock Axon Hook	0.2936	0.011	0.5762	True
Motion Control (MC) ETD	Otto Bock Hand	0.274	0.0989	0.4492	True
Motion Control (MC) ETD	Otto BockGreifer	0.4061	0.0311	0.781	True
Motion Control (MC) ETD	Passive Hand	0.2653	-0.0888	0.6194	False
Motion Control (MC) ETD	TRS Hook	-0.1031	-1.0356	0.8294	False
Motion Control (MC) ETD	bebionic Hand	-0.006	-0.1935	0.1815	False
Motion Control (MC) ETD	i-limb Hand	0.4085	0.1917	0.6253	True
Motion Control (MC) Hand	Other (please specify)	0.1855	-0.7386	1.1097	False
Motion Control (MC) Hand	Otto Bock Axon Hook	0.083	-0.1708	0.3369	False
Motion Control (MC) Hand	Otto Bock Hand	0.0635	-0.0601	0.187	False
Motion Control (MC) Hand	Otto BockGreifer	0.1955	-0.1583	0.5493	False
Motion Control (MC) Hand	Passive Hand	0.0547	-0.2769	0.3864	False
Motion Control (MC) Hand	TRS Hook	-0.3137	-1.2378	0.6105	False
Motion Control (MC) Hand	bebionic Hand	-0.2166	-0.3571	-0.0761	True
Motion Control (MC) Hand	i-limb Hand	0.1979	0.0202	0.3756	True
Other (please specify)	Otto Bock Axon Hook	-0.1025	-1.057	0.852	False
Other (please specify)	Otto Bock Hand	-0.1221	-1.0505	0.8064	False
Other (please specify)	Otto BockGreifer	0.01	-0.9758	0.9958	False
Other (please specify)	Passive Hand	-0.1308	-1.1089	0.8473	False
Other (please specify)	TRS Hook	-0.4992	-1.8033	0.8049	False
Other (please specify)	bebionic Hand	-0.4021	-1.333	0.5287	False
Other (please specify)	i-limb Hand	0.0124	-0.9248	0.9496	False
Otto Bock Axon Hook	Otto Bock Hand	-0.0196	-0.2885	0.2494	False
Otto Bock Axon Hook	Otto BockGreifer	0.1125	-0.3144	0.5394	False
Otto Bock Axon Hook	Passive Hand	-0.0283	-0.437	0.3804	False
Otto Bock Axon Hook	TRS Hook	-0.3967	-1.3512	0.5578	False
Otto Bock Axon Hook	bebionic Hand	-0.2996	-0.5767	-0.0225	True
Otto Bock Axon Hook	i-limb Hand	0.1149	-0.1828	0.4126	False
Otto Bock Hand	Otto BockGreifer	0.132	-0.2327	0.4968	False
Otto Bock Hand	Passive Hand	-0.0088	-0.3521	0.3346	False
Otto Bock Hand	TRS Hook	-0.3771	-1.3056	0.5513	False
Otto Bock Hand	bebionic Hand	-0.2801	-0.4463	-0.1139	True
Otto Bock Hand	i-limb Hand	0.1345	-0.0642	0.3331	False
Otto BockGreifer	Passive Hand	-0.1408	-0.6181	0.3365	False

Otto BockGreifer	TRS Hook	-0.5092	-1.495	0.4766	False
Otto BockGreifer	bebionic Hand	-0.4121	-0.783	-0.0413	True
Otto BockGreifer	i-limb Hand	0.0024	-0.3841	0.3889	False
Passive Hand	TRS Hook	-0.3684	-1.3465	0.6097	False
Passive Hand	bebionic Hand	-0.2713	-0.6211	0.0785	False
Passive Hand	i-limb Hand	0.1432	-0.2231	0.5095	False
TRS Hook	bebionic Hand	0.0971	-0.8338	1.0279	False
TRS Hook	i-limb Hand	0.5116	-0.4256	1.4488	False
bebionic Hand	i-limb Hand	0.4145	0.205	0.6241	True



The TRS Hook and "other" are mucking up the scale, let's remove those

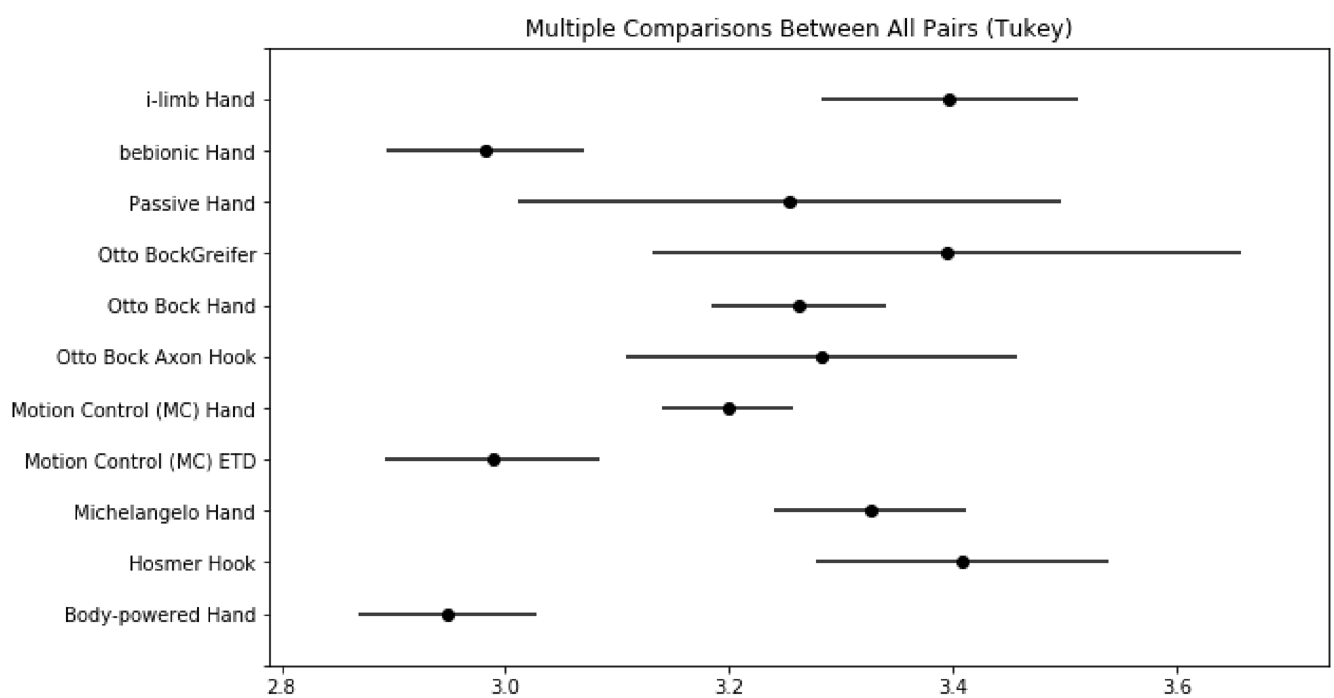
In [27]:
```
opusDf = pd.concat([sht['PrimaryTerminalDevice'], sht['normalizedOpus']], axis=1, keys=
['primaryTD','normalizedOpus'])
opusDf["primaryTD"] = opusDf["primaryTD"].replace('Other.*', np.nan, regex=True)
opusDf["primaryTD"] = opusDf["primaryTD"].replace('TRS.*', np.nan, regex=True)
opusDf = opusDf.dropna()
df = opusDf.as_matrix()
dC = df[:,0].astype(str)
oS = df[:,1].astype(float)
tukey = pairwise_tukeyhsd(endog = oS,
                          groups = dC,
                          alpha = 0.05)
tukey.plot_simultaneous()    # Plot group confidence intervals
tukey.summary()
```

Out[27]:

Multiple Comparison of Means - Tukey HSD,FWER=0.05

group1	group2	meandiff	lower	upper	reject
Body-powered Hand	Hosmer Hook	0.4603	0.2459	0.6747	True
Body-powered Hand	Michelangelo Hand	0.3782	0.2169	0.5394	True
Body-powered Hand	Motion Control (MC) ETD	0.0403	-0.1324	0.2131	False
Body-powered Hand	Motion Control (MC) Hand	0.2509	0.1273	0.3746	True
Body-powered Hand	Otto Bock Axon Hook	0.3339	0.071	0.5969	True
Body-powered Hand	Otto Bock Hand	0.3144	0.1636	0.4651	True
Body-powered Hand	Otto BockGreifer	0.4464	0.0908	0.802	True
Body-powered Hand	Passive Hand	0.3056	-0.0292	0.6405	False
Body-powered Hand	bebionic Hand	0.0343	-0.1299	0.1985	False
Body-powered Hand	i-limb Hand	0.4488	0.2536	0.6441	True
Hosmer Hook	Michelangelo Hand	-0.0821	-0.302	0.1378	False
Hosmer Hook	Motion Control (MC) ETD	-0.4199	-0.6484	-0.1915	True
Hosmer Hook	Motion Control (MC) Hand	-0.2094	-0.4034	-0.0153	True
Hosmer Hook	Otto Bock Axon Hook	-0.1263	-0.4288	0.1761	False
Hosmer Hook	Otto Bock Hand	-0.1459	-0.3582	0.0664	False
Hosmer Hook	Otto BockGreifer	-0.0139	-0.3996	0.3719	False
Hosmer Hook	Passive Hand	-0.1547	-0.5214	0.2121	False
Hosmer Hook	bebionic Hand	-0.426	-0.648	-0.2039	True
Hosmer Hook	i-limb Hand	-0.0114	-0.2573	0.2345	False
Michelangelo Hand	Motion Control (MC) ETD	-0.3378	-0.5174	-0.1583	True
Michelangelo Hand	Motion Control (MC) Hand	-0.1272	-0.2603	0.0058	False
Michelangelo Hand	Otto Bock Axon Hook	-0.0442	-0.3117	0.2232	False
Michelangelo Hand	Otto Bock Hand	-0.0638	-0.2223	0.0947	False
Michelangelo Hand	Otto BockGreifer	0.0683	-0.2907	0.4273	False
Michelangelo Hand	Passive Hand	-0.0725	-0.411	0.2659	False
Michelangelo Hand	bebionic Hand	-0.3438	-0.5152	-0.1725	True
Michelangelo Hand	i-limb Hand	0.0707	-0.1306	0.272	False
Motion Control (MC) ETD	Motion Control (MC) Hand	0.2106	0.0639	0.3573	True
Motion Control (MC) ETD	Otto Bock Axon Hook	0.2936	0.0191	0.5681	True
Motion Control (MC) ETD	Otto Bock Hand	0.274	0.1039	0.4442	True
Motion Control (MC) ETD	Otto BockGreifer	0.4061	0.0418	0.7704	True
Motion Control (MC) ETD	Passive Hand	0.2653	-0.0788	0.6093	False
Motion Control (MC) ETD	bebionic Hand	-0.006	-0.1882	0.1761	False

Motion Control (MC) ETD	i-limb Hand	0.4085	0.1979	0.6191	True
Motion Control (MC) Hand	Otto Bock Axon Hook	0.083	-0.1636	0.3296	False
Motion Control (MC) Hand	Otto Bock Hand	0.0635	-0.0566	0.1835	False
Motion Control (MC) Hand	Otto BockGreifer	0.1955	-0.1482	0.5392	False
Motion Control (MC) Hand	Passive Hand	0.0547	-0.2675	0.3769	False
Motion Control (MC) Hand	bebionic Hand	-0.2166	-0.3531	-0.0801	True
Motion Control (MC) Hand	i-limb Hand	0.1979	0.0253	0.3706	True
Otto Bock Axon Hook	Otto Bock Hand	-0.0196	-0.2808	0.2417	False
Otto Bock Axon Hook	Otto BockGreifer	0.1125	-0.3022	0.5272	False
Otto Bock Axon Hook	Passive Hand	-0.0283	-0.4254	0.3688	False
Otto Bock Axon Hook	bebionic Hand	-0.2996	-0.5688	-0.0304	True
Otto Bock Axon Hook	i-limb Hand	0.1149	-0.1743	0.4041	False
Otto Bock Hand	Otto BockGreifer	0.132	-0.2223	0.4864	False
Otto Bock Hand	Passive Hand	-0.0088	-0.3423	0.3248	False
Otto Bock Hand	bebionic Hand	-0.2801	-0.4415	-0.1186	True
Otto Bock Hand	i-limb Hand	0.1345	-0.0585	0.3274	False
Otto BockGreifer	Passive Hand	-0.1408	-0.6045	0.3229	False
Otto BockGreifer	bebionic Hand	-0.4121	-0.7724	-0.0518	True
Otto BockGreifer	i-limb Hand	0.0024	-0.3731	0.3779	False
Passive Hand	bebionic Hand	-0.2713	-0.6111	0.0685	False
Passive Hand	i-limb Hand	0.1432	-0.2127	0.4991	False
bebionic Hand	i-limb Hand	0.4145	0.2109	0.6182	True



The brands with wide confidence intervals don't have many respondents. The brands with narrow confidence intervals have many respondents. This graph helps us see where everything lands.

```
In [ ]:
```