

### 1 Sample T-Test

```
from scipy.stats import ttest_1samp
tstat, pval = ttest_1samp(example_distribution,
expected_mean)
-> Generates two outputs
-> tstat:t-statistic
-> pval: p-value
```

### 2 Sample T-Test

```
from scipy.stats import ttest_ind
tstat, pval = ttest_ind(data1, data2)
```

### ANOVA

```
from scipy.stats import f_oneway
fstat, pval = f_oneway(data1, data2, data3)
```

### Tukey's Range Test (not SciPy)

```
from statsmodels.stats.multicomp import
pairwise_tukeyhsd
# All Data has to be unioned to one List
movie_scores = np.concatenate([drama_scores,
comedy_scores, documentary_scores])
labels = ['drama'] * len(drama_scores) +
['comedy'] * len(comedy_scores) + ['documentary']
* len(documentary_scores)
tukey_results = pairwise_tukeyhsd(movie_scores,
labels, 0.05)
-> 0.05 represents the significance level
```

### Binomial Test

```
from scipy.stats import binom_test
pval = binom_test(successes, n, p)
successes: Number observed successes
n: Number of Trials
p: Expected probability for success
-> if pval < 0.05 we can reject the Null
Hypothesis
```

### Chi Square Test

```
from scipy.stats import chi2_contingency
_, pval, _, _ = chi2_contingency(iron_contingency-
_table)
```

### Point Distance Functions

Import: `from scipy.spatial import distance`

Euclidean **`distance.euclidean(pt1, pt2)`**

Manhattan **`distance.cityblock(pt1, pt2)`**

Hamming **`distance.hamming(pt1, pt2)`**

**Manhattan Distance:** like calculating how many blocks are between two points

**Hamming Distance:** will always return a number between 0 and 1  
-> The Hamming distance between [1, 2, 3] and [7, 2, -10] would be 2. In scipy's version, it would be 2/3

