**FORMULAS**:

If the date columns are a timestamp instead of a date, you may want to use pd.DatetimeIndex to get just the date.

experiment\_data['date'] = pd.DatetimeIndex(experiment\_data['timestamp']).date

print(experiment\_data.date.unique())

how large?

**print(**'test sample size:', **len**(experiment\_data[experiment\_data.test == 1]))

**print**('test proportion:', experiment\_data.test**.mean())**

what’s the mean, rate?

**print(**'test signup rate:', experiment\_data[experiment\_data.test ==1].**is\_signed\_up.mean())**

Are the test and control groups significantly different?

**stats.ttest\_ind**(experiment\_data[experiment\_data.test == 1].is\_signed\_up,

experiment\_data[experiment\_data.test == 0].is\_signed\_up)

**plt.legend**(['test','control']) #shows on the right upper corner color test, orange control

**plt.title**('Comparing Signup Rates by Treatment Date')

**plt.show()**

We can get a record count for each group by chaining the `groupby()` and `agg()` methods:

**iris.groupby**(['species']).**agg([**'count'])agg – aggregate – total, total of species here.