

```
import numpy as np
import statsmodels.api as sm
import pandas as pd
```

```
/usr/local/lib/python3.7/dist-packages/statsmodels/tools/_testing.py:19: FutureWarning
import pandas.util.testing as tm
```

```
# One population proportion
n = 1018
p = 0.52
phat = 0.56
sm.stats.proportions_ztest(phat * n , n, value = p, alternative = 'larger', prop_var = 0)

(2.571067795759113, 0.005069273865860533)
```

```
# Difference in population proportions
```

```
# Total parents
n1 = 247
n2= 308
```

```
# Parents in support
y1= 91
y2= 120
```

```
# Estimates of the population proportions
p1 = round(y1 / n1, 2)
p2 = round(y2 / n2, 2)
```

```
population1 = np.random.binomial(1, p1, n1)
population2 = np.random.binomial(1, p2, n2)
```

```
sm.stats.ttest_ind(population1, population2)
```

```
↳ (-1.5804959633985163, 0.11456509991112795, 553.0)
```

```
# One Population Mean
```

```
df = pd.read_csv('Cartwheeldata.csv')
n = len(df)
mean = df["CWDistance"].mean()
sd = df["CWDistance"].std()
(n, mean, sd)
```

```
(25, 82.48, 15.058552387264852)
```

```
sm.stats.ztest(df["CWDistance"], value = 80, alternative = "larger")
```

```
(0.8234523266982029, 0.20512540845395266)
```

```
# Difference in Population means
url = "nhanes_2015_2016.csv"
da = pd.read_csv(url)
da.head()
```

	SEQN	ALQ101	ALQ110	ALQ130	SMQ020	RIAGENDR	RIDAGEYR	RIDRETH1	DMDCITZN	DMI
0	83732	1.0	NaN	1.0	1	1	62	3	1.0	
1	83733	1.0	NaN	6.0	1	1	53	3	2.0	
2	83734	1.0	NaN	NaN	1	1	78	3	1.0	
3	83735	2.0	1.0	1.0	2	2	56	3	1.0	
4	83736	2.0	1.0	1.0	2	2	42	4	1.0	

5 rows × 28 columns

```
females = da[da["RIAGENDR"] == 2]
males = da[da["RIAGENDR"] == 1]
```

```
n1 = len(females)
n2 = len(males)
```

```
m1 = females['BMXBMI'].mean()
s1 = females['BMXBMI'].std()
```

```
m2 = males['BMXBMI'].mean()
s2 = males['BMXBMI'].std()
```

```
[(m1,s1), (m2,s2)]
```

```
[(29.93994565217392, 7.753318809545674),
 (28.778072111846942, 6.2525676168014614)]
```

```
sm.stats.ztest(females['BMXBMI'].dropna(), males['BMXBMI'].dropna(), value = 0, alternative = 'two-sided')
# NOTE: .dropna() ensures that missing values are not being considered as they result in e
```

```
(6.1755933531383205, 6.591544431126401e-10)
```