In [1]:
```
%matplotlib inline
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from numpy.random import seed
from numpy.random import randn
from statsmodels.graphics.gofplots import qqplot
import numpy as np
from scipy import stats
```

In [2]:
```
from math import sqrt
from numpy.random import seed
from numpy.random import randn
from numpy import mean
from scipy.stats import sem
from scipy.stats import t

# function for calculating the t-test for two independent samples
def independent_ttest(data1, data2, alpha):
    # calculate means
    mean1, mean2 = mean(data1), mean(data2)
    # calculate standard errors
    se1, se2 = sem(data1), sem(data2)
    # standard error on the difference between the samples
    sed = sqrt(se1**2.0 + se2**2.0)
    # calculate the t statistic
    t_stat = (mean1 - mean2) / sed
    # degrees of freedom
    df = len(data1) + len(data2) - 2
    # calculate the critical value
    cv = t.ppf(1.0 - alpha, df)
    # calculate the p-value
    p = (1.0 - t.cdf(abs(t_stat), df)) * 2.0
    # return everything
    return t_stat, df, cv, p
```

Intro

Each distribution is a total of eggs layed during 6 days for each female individual

In [3]:
```
FH = [2,0,0,0,0,0,0,0,0,0]
FMH = [38,37,23,33,34,30,38,46,43,47]
FH2 = [16,45,27,3,8,10,5,7,21,7,]
F =  [0,0,0,0,0,0,0,0,0,0,]
FM = [0,0,0,0,0,0,0,1,1,1,]
```

Each distribution is the total emergence for each female indivual layed eggs during 6 days

In [4]:
```
eFH = [0,7,0,0,27,0,0,0,0,0,0,11,2,5,7,2,0,3,8,2]
eFMH = [25,6,9,13,7,11,3,19,16,16,5,7,0,4,11,19,9,14,18,2]
```

Sanity test

In [5]:
```python
mu, sigma = 30, 5 # mean and standard deviation
s = np.random.normal(mu, sigma, 5000)
stats.kstest(s, 'norm')
```

Out[5]:
```
KstestResult(statistic=1.0, pvalue=0.0)
```

In [6]:
```python
mu2, sigma2 = 25, 5 # mean and standard deviation
s2 = np.random.normal(mu2, sigma2, 5000)
stats.kstest(s2, 'norm')
```

Out[6]:
```
KstestResult(statistic=0.9999999999999998, pvalue=0.0)
```

In [7]:
```python
zz = [0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
stats.kstest(zz, 'norm')
```

Out[7]:
```
KstestResult(statistic=0.5, pvalue=3.787595240539032e-05)
```

In [8]:
```python
stats.ttest_ind(s2, s)
```

Out[8]:
```
Ttest_indResult(statistic=-49.84449053216533, pvalue=0.0)
```

In [9]:
```python
def interpret_ttest(a, b, alpha=0.05):
    t_stat, df, cv, p = independent_ttest(a, b, alpha)
    print('t=%.3f, df=%d, cv=%.5f, p=%.8f' % (t_stat, df, cv, p))
    # interpret via critical value
    if abs(t_stat) <= cv:
        print('Accept null hypothesis that the means are equal.')
    else:
        print('Reject the null hypothesis that the means are equal.')
    # interpret via p-value
    if p > alpha:
        print('Accept null hypothesis that the means are equal.')
    else:
        print('Reject the null hypothesis that the means are equal.')
```

In [10]:
```python
interpret_ttest(s,s2)
```

```
t=49.844, df=9998, cv=1.64501, p=0.00000000
Reject the null hypothesis that the means are equal.
Reject the null hypothesis that the means are equal.
```

In [11]:
```python
def interpret_kwtest(a, b, alpha = 0.05):
    stat, p = stats.kruskal(a, b)
    print('Statistics=%.3f, p=%.8f' % (stat, p))
    # interpret

    if p > alpha:
        print('Same distributions (fail to reject H0)')
    else:
        print('Different distributions (reject H0)')
```

In [12]:
```python
interpret_kwtest(s,s2)
```

```
Statistics=2017.332, p=0.00000000
Different distributions (reject H0)
```

In [13]:
```python
def interpret_kstest(a, alpha = 0.05):
    ks, p = stats.kstest(a, 'norm')
    print(f'Statistics={ks:.3f} p={p:0.8f}' )
    # interpret
    if abs(ks) <= 0.5 or p > alpha:
        print('We cannot reject the hypothesis that the distributions of the t
    else:
        print(f'Accept the hypothesis that the distribution is normal with the
    # interpret via p-value
#     If the K-S statistic is small or the p-value is high,
#     then we cannot reject the hypothesis that the distributions of the two s
# Cannot reject doesn't mean we confirm.

interpret_kstest(s)
```

```
Statistics=1.000 p=0.00000000
Accept the hypothesis that the distribution is normal with the probabililty o
f 100.00%.
```

Validate if those are normal distributions

Using kolmogorov-smirnov tests

In [14]:
```python
distributions = {
    'F': F,
    'FH2': FH2,
    'FH': FH,
    'FMH': FMH,
    'FM': FM,

}
```

In [15]:
```python
for k, v in distributions.items():
    print(f'Distributions {k}\n')
    interpret_kstest(v)
    print('\n\n')
```

```
Distributions F

Statistics=0.500 p=0.00777741
We cannot reject the hypothesis that the distributions of the two samples are
the same (Normal).



Distributions FH2

Statistics=0.999 p=0.00000000
Accept the hypothesis that the distribution is normal with the probabililty o
f 99.87%.



Distributions FH

Statistics=0.500 p=0.00777741
We cannot reject the hypothesis that the distributions of the two samples are
the same (Normal).



Distributions FMH

Statistics=1.000 p=0.00000000
Accept the hypothesis that the distribution is normal with the probabililty o
f 100.00%.



Distributions FM

Statistics=0.500 p=0.00777741
We cannot reject the hypothesis that the distributions of the two samples are
the same (Normal).
```

Considering normal distributions

In [16]:
```
scenarios = {
    'F-FH': (F,FH),
    'F-FM': (F, FM),
    'F-FMH': (F, FMH),
    'F-FH2': (F, FH2),

    'FH-FM': (FH, FM),
    'FH-FMH': (FH, FMH),
    'FH-FH2': (FH, FH2),

    'FM-FMH': (FM, FMH),
    'FM-FH2': (FM, FH2),

    'FMH-FH2': (FMH, FH2),

}
```

In [17]:
```
# for k, v in scenarios.items():
#     a, b = v
#     print(f'Scenario {k}\n')
#     interpret_ttest(a,b)
#     print('\n\n')
```

Non parametric test using Kruskal Wallis Test

In [18]:
```
scenarios_dict = []
for k, v in scenarios.items():
    a, b = v
    print(f'Scenario {k}\n')
    interpret_kwtest(a,b)
    print('\n\n')
```

```
Scenario F-FH

Statistics=1.000, p=0.31731051
Same distributions (fail to reject H0)



Scenario F-FM

Statistics=3.353, p=0.06708505
Same distributions (fail to reject H0)



Scenario F-FMH

Statistics=16.323, p=0.00005341
Different distributions (reject H0)



Scenario F-FH2

Statistics=16.323, p=0.00005341
Different distributions (reject H0)



Scenario FH-FM

Statistics=0.850, p=0.35655234
Same distributions (fail to reject H0)



Scenario FH-FMH

Statistics=15.715, p=0.00007362
Different distributions (reject H0)



Scenario FH-FH2

Statistics=15.715, p=0.00007362
Different distributions (reject H0)



Scenario FM-FMH

Statistics=14.972, p=0.00010909
Different distributions (reject H0)



Scenario FM-FH2
```

```
Statistics=14.972, p=0.00010909
Different distributions (reject H0)



Scenario FMH-FH2

Statistics=9.620, p=0.00192451
Different distributions (reject H0)
```