1. MLE

a) $\mathcal{L}(\lambda, x) = P_\lambda(X=x) \quad P_\lambda(X=x_i) = \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}$

$$\mathcal{L}(\lambda, x) = \prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}$$

The x_i s are independent & therefore can all be just multiplied together

b) $$\mathcal{L}(\lambda, x) = \prod_{i=1}^{n} \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}$$

$$\log \mathcal{L}(\lambda, x) = \sum_{i=1}^{n} \log \frac{\lambda^{x_i} e^{-\lambda}}{x_i!}$$

$$= \sum_{i=1}^{n} \left(x_i \log \lambda - \lambda \log e - \log(x_i!) \right)$$

ln instead of log doesn't make a difference

$\Updownarrow$

$$\ln \mathcal{L}(\lambda, x) = \sum_{i=1}^{n} \left(x_i \ln \lambda - \lambda - \ln(x_i!) \right)$$

$$\hat{\lambda}_{MLE} = \underset{\lambda}{\operatorname{argmax}}\, \mathcal{L}(\lambda, x) = \underset{\lambda}{\operatorname{argmax}}\, \ln \mathcal{L}(\lambda, x)$$

Take derivative, set to 0

$$\frac{d}{d\lambda}\left(\sum_{i=1}^{n} \left(x_i \ln \hat{\lambda} - \hat{\lambda} - \ln(x_i!) \right) \right) = 0$$

$$\sum_{i=1}^{n} \left(\frac{x_i}{\hat{\lambda}} - 1 \right) = 0$$

$$\sum_{i=1}^{n} \frac{x_i}{\hat{\lambda}} = n$$

$$\hat{\lambda} = \frac{1}{n} \sum_{i=1}^{n} x_i$$

2. Poisson Counts

a) $H_0: \quad \lambda_0 = \frac{1}{2n} \sum_{i=1}^{n} (x_i + y_i)$

$H_a: \quad \lambda_x = \frac{1}{n} \sum_{i=1}^{n} x_i$

$\lambda_y = \frac{1}{n} \sum_{i=1}^{n} y_i$

b) Jupyter Notebook

c) $$\mathcal{L}_0 = \prod_{i=1}^{n} \frac{\lambda_0^{x_i} e^{-\lambda_0}}{x_i!} \cdot \frac{\lambda_0^{y_i} e^{-\lambda_0}}{y_i!}$$

$$\mathcal{L}_a = \prod_{i=1}^{n} \frac{\lambda_x^{x_i} e^{-\lambda_x}}{x_i!} \cdot \frac{\lambda_y^{y_i} e^{-\lambda_y}}{y_i!}$$

d) Jupyter Notebook

e) χ^2 value for 1 dof: 3.84

Reject if ratio > 3.84

Ratio = 0.296

so we can't reject the null hypothesis

In [0]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import poisson
```

In [0]:
```
two = pd.read_csv("q2_set_1.tsv", sep='\t', index_col=0)
three_1 = pd.read_csv("q3_set_1.tsv", sep='\t')
three_2 = pd.read_csv("q3_set_2.tsv", sep='\t')
```

2. Poisson Counts

2b. Compute Lambdas

In [0]:
```
two_x = two.iloc[0,0:5].to_numpy()
two_y = two.iloc[0,5:].to_numpy()
two_all = two.iloc[0,:].to_numpy()
```

In [4]:
```
lambda_0 = np.mean(two_all)
lambda_x = np.mean(two_x)
lambda_y = np.mean(two_y)
print("lambda_0: " + str(lambda_0))
print("lambda_x: " + str(lambda_x))
print("lambda_y: " + str(lambda_y))
```

```
lambda_0: 5889.4
lambda_x: 5902.6
lambda_y: 5876.2
```

2d. Compute Likelihoods

In [5]:
```
likelihood_0 = np.prod(poisson.pmf(two_all, lambda_0))
likelihood_a = np.prod(poisson.pmf(two_x, lambda_x))*np.prod(poisson.pmf(two_y,
                                                                          lambda_
y))
print("likelihood_0: " + str(likelihood_0))
print("likelihood_a: " + str(likelihood_a))
```

```
likelihood_0: 2.2655243807774852e-24
likelihood_a: 2.626712736561282e-24
```

2e. Compute Likelihood Ratio

In [6]:
```
ratio = -2*np.log(likelihood_0/likelihood_a)
print("Likelihood ratio: " + str(ratio))
```

```
Likelihood ratio: 0.29585381503238745
```

3. Comparing Distributions

In [0]:
```
set_1 = {'cond1': three_1.iloc[:,:5].to_numpy(),
'cond2': three_1.iloc[:,5:].to_numpy()}

set_2 = {'cond1': three_2.iloc[:,:5].to_numpy(),
'cond2': three_2.iloc[:,5:].to_numpy()}
```

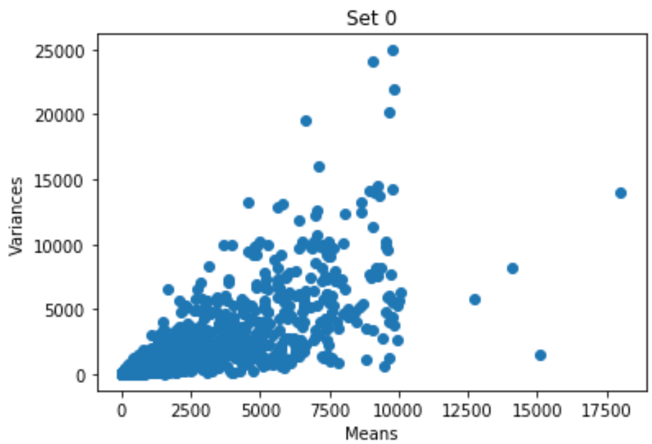
3a. Mean and Variance

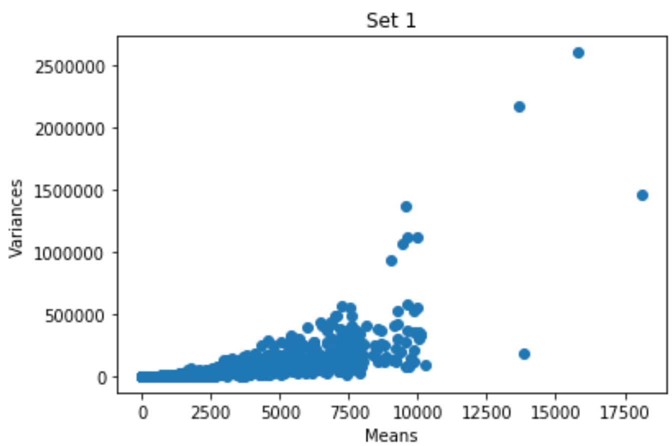
In [0]:
```
set_1_stats = {}
set_2_stats = {}

for data, stat in zip([set_1, set_2], [set_1_stats, set_2_stats]):
    for cond in data.keys():
        stat['mean'+'_'+cond] = np.mean(data[cond], axis = 1)
        stat['var'+'_'+cond] = np.var(data[cond], axis = 1)
```

3b. Plot Means and Variances

In [9]:
```
for i, stat in enumerate([set_1_stats, set_2_stats]):
    plt.figure()
    means = np.concatenate((stat['mean_cond1'], stat['mean_cond2']))
    variances = np.concatenate((stat['var_cond1'], stat['var_cond2']))
    plt.scatter(means, variances)
    plt.title("Set "+str(i))
    plt.xlabel("Means")
    plt.ylabel("Variances")
    plt.plot()
```





3c.i. Determine statistical difference in expression counts

In [10]:
```
significant = {"set1": 0, "set2": 0}

n = len(three_1)

for j, (data, stat) in enumerate(zip([set_1, set_2],
                                     [set_1_stats, set_2_stats])):
    for i in range(n):
        both_conds = np.concatenate((data['cond1'][i], data['cond2'][i]))

        lambda_0 = np.mean(both_conds)
        likelihood_0 = np.prod(poisson.pmf(both_conds, lambda_0))

        lambda_cond1 = stat['mean_cond1'][i]
        lambda_cond2 = stat['mean_cond2'][i]
        likelihood_a = np.prod(poisson.pmf(data['cond1'][i],
                        lambda_cond1))*np.prod(poisson.pmf(data['cond2'][i],
                                                           lambda_cond2))

        ratio = -2*np.log(likelihood_0/likelihood_a)
        if ratio > 3.84:
            significant['set'+str(j+1)] += 1
```

```
/usr/local/lib/python3.6/dist-packages/ipykernel_launcher.py:19: RuntimeWarning:
divide by zero encountered in log
```

In [11]:
```
print(significant)
```

```
{'set1': 139, 'set2': 376}
```

3.c.ii. Determine underlying distribution

Using Poisson on the negative binomial gives us an inflated number of signficant differences. Based on the plots in 3b and the counts of significant differences, we can coonclude that set 1 is the Poisson distribution and set 2 is the negative binomial distribution.