MLE_with_same_distributions

September 16, 2018

In [12]:
```
import numpy as np
import matplotlib.pyplot as plt
```

In [13]:
```
#Number of initialized data points
N_points = 100000
#Number of columns in histogram
n_bins = 100
```

In [32]:
```
print("Select distribution by number")
print("1. Binomial")
print("2. Poisson")
print("3. Exponential")
print("4. Gaussian")
print("5. Laplacian")
distri = int(input())
```

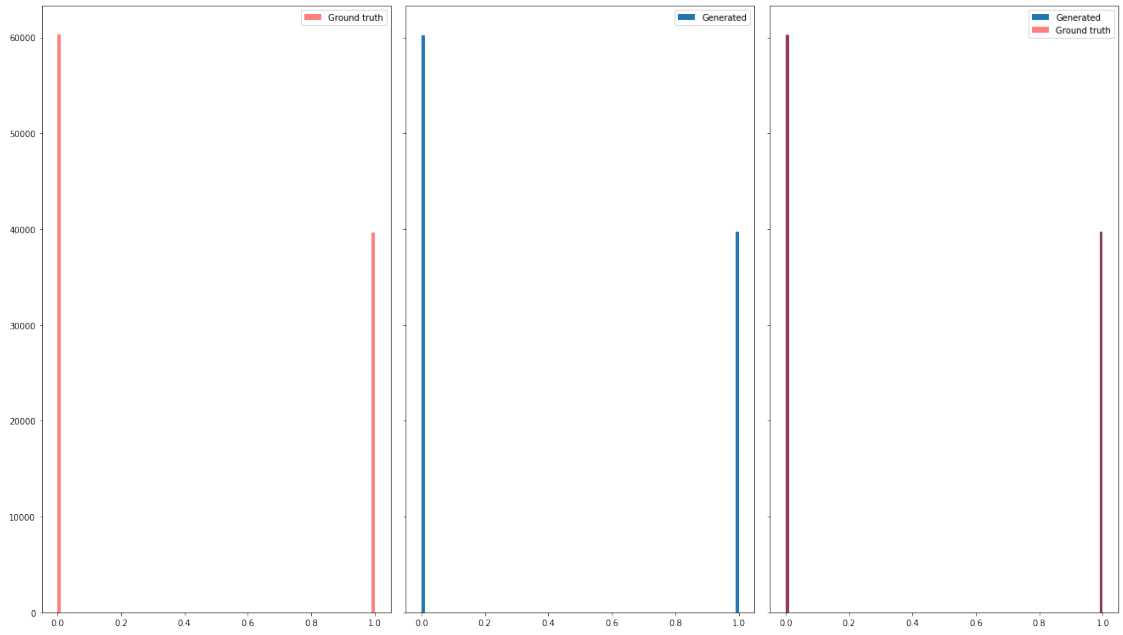
```
Select distribution by number
1. Binomial
2. Poisson
3. Exponential
4. Gaussian
5. Laplacian
5
```

In [22]:
```
if(distri == 1):
    print("Give probability factor for Binomial distribution generation")
    p = float(input())
    while p>1 or p<0 :
        print("Re enter p with proper bounds")
        p = float(input())
    Ground_truth = np.random.binomial(1,p,N_points)
    TH = np.mean(Ground_truth)
    Generated_bin_1 = np.random.binomial(1,TH,N_points)

    fig, axs = plt.subplots(1, 3, sharey=True, tight_layout=True)
    axs[0].hist(Ground_truth, bins=n_bins,alpha=0.5,color='r',label='Ground truth')
    axs[0].legend()
```

```
            axs[1].hist(Generated_bin_1, bins=n_bins,label="Generated")
            axs[1].legend()

            axs[2].hist(Generated_bin_1, bins=n_bins,label='Generated')
            axs[2].hist(Ground_truth, bins=n_bins,alpha=0.5,color='r',label='Ground truth')
            axs[2].legend()
            fig.set_size_inches(18.5, 10.5, forward=True)
        #     plt.savefig("Bin_Vs_Bin.png")
```

```
Give probability factor for Binomial distribution generation
0.4
```

```
/home/legion/.local/lib/python3.6/site-packages/matplotlib/figure.py:2299: UserWarning: This fi
  warnings.warn("This figure includes Axes that are not compatible "
```

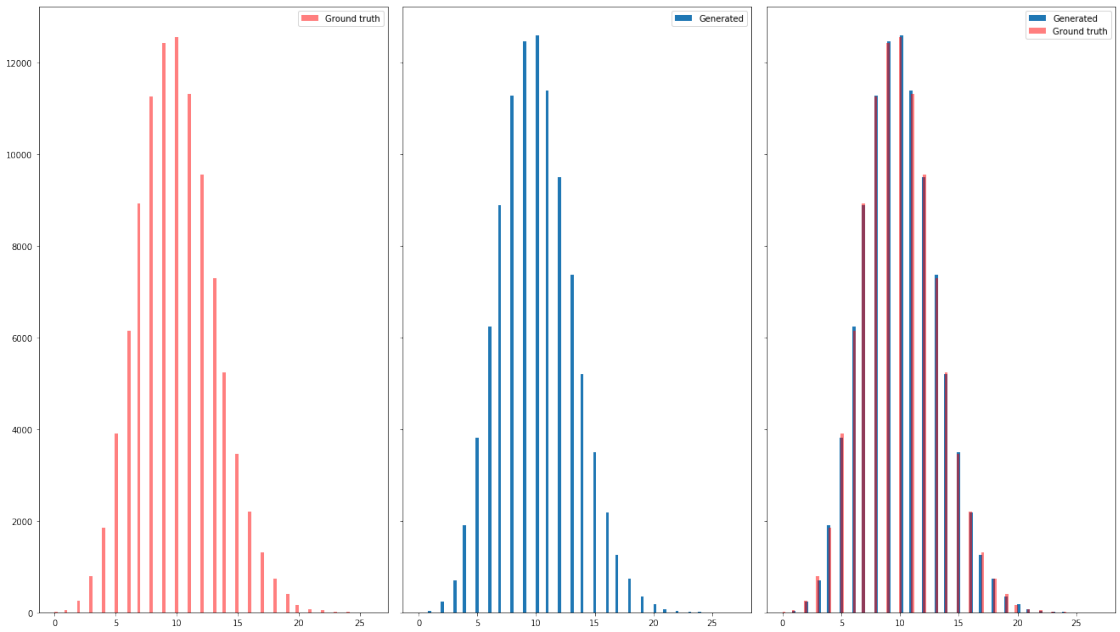



```
In [24]: if(distri == 2):
            print("Enter lambda for poisson")
            Lambda_1 = float(input())
            while Lambda_1<0 :
                print("Enter proper lambda (Lambda >= 0)")
                Lambda_1 = float(input())

            Ground_truth = np.random.poisson(Lambda_1,N_points)
            Lambda = np.mean(Ground_truth)
```

```
            Generated = np.random.poisson(Lambda,N_points)
            fig, axs = plt.subplots(1, 3, sharey=True, tight_layout=True)
            axs[1].hist(Generated, bins=n_bins,label='Generated')
            axs[1].legend()

            axs[0].hist(Ground_truth, bins=n_bins,alpha=0.5,color='r',label='Ground truth')
            axs[0].legend()

            axs[2].hist(Generated, bins=n_bins,label='Generated')
            axs[2].hist(Ground_truth, bins=n_bins,alpha=0.5,color='r',label='Ground truth')
            axs[2].legend()
            fig.set_size_inches(18.5, 10.5, forward=True)
        #     plt.savefig("Lambda_Vs_Lambda.png")
```

```
Enter lambda for poisson
10
```

```
/home/legion/.local/lib/python3.6/site-packages/matplotlib/figure.py:2299: UserWarning: This fi
  warnings.warn("This figure includes Axes that are not compatible "
```

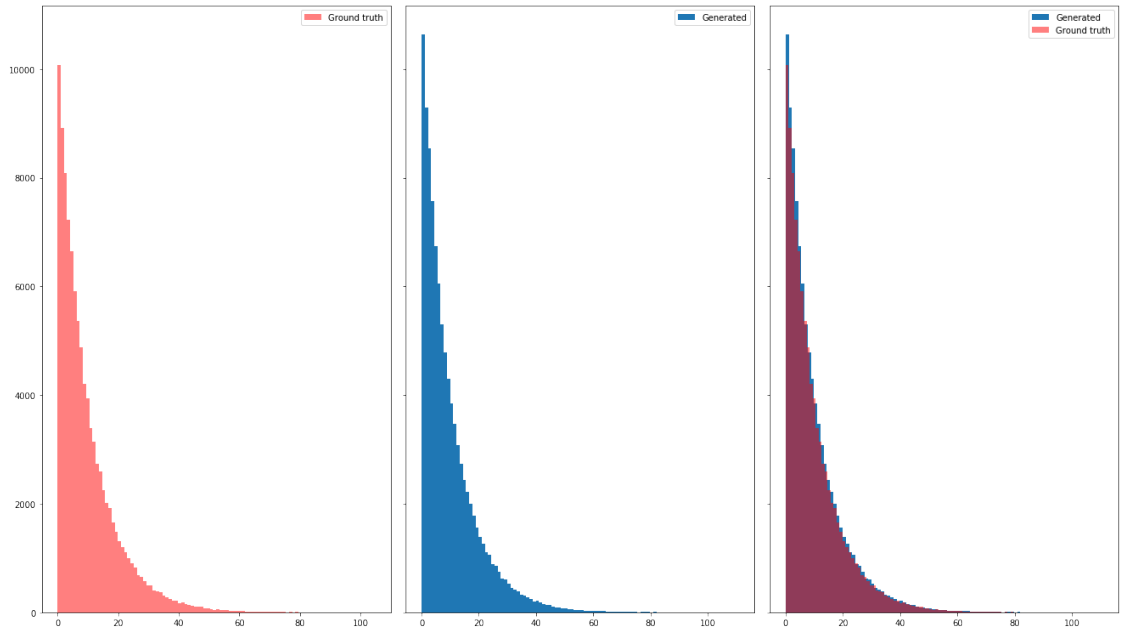



```
In [29]: if(distri == 3):
            print("Enter beta (1/lambda) for Exponential distribution")
            beta = float(input())
            while beta<0 :
                print("Enter proper beta (beta >= 0)")
```

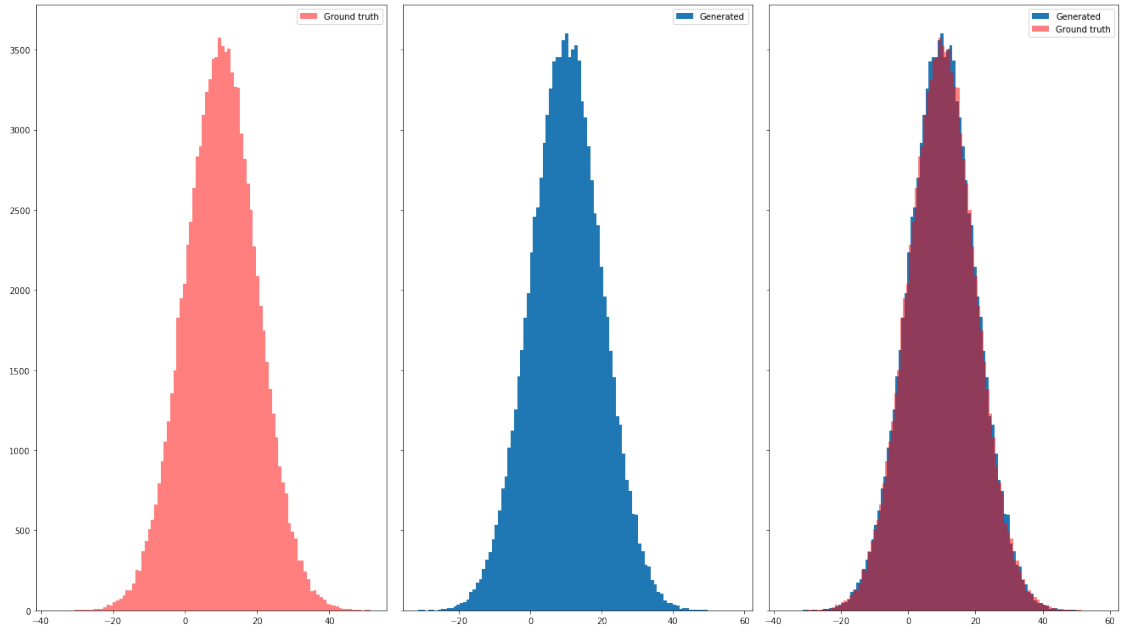
```
        beta = float(input())
    Ground_truth = np.random.exponential(beta,N_points)
    beta1 = np.mean(Ground_truth)
    Generated = np.random.exponential(beta1,N_points)
    fig, axs = plt.subplots(1, 3, sharey=True, tight_layout=True)
    axs[0].hist(Ground_truth, bins=n_bins,label='Ground truth',color='r',alpha=0.5)
    axs[0].legend()

    axs[1].hist(Generated, bins=n_bins,label='Generated')
    axs[1].legend()

    axs[2].hist(Generated, bins=n_bins,label='Generated')
    axs[2].hist(Ground_truth, bins=n_bins,label='Ground truth',color='r',alpha=0.5)
    axs[2].legend()

    fig.set_size_inches(18.5, 10.5, forward=True)

#     plt.savefig("Exp_Vs_Exp.png")
```

```
Enter beta (1/lambda) for Exponential distribution
10
```

```
/home/legion/.local/lib/python3.6/site-packages/matplotlib/figure.py:2299: UserWarning: This f
  warnings.warn("This figure includes Axes that are not compatible "
```



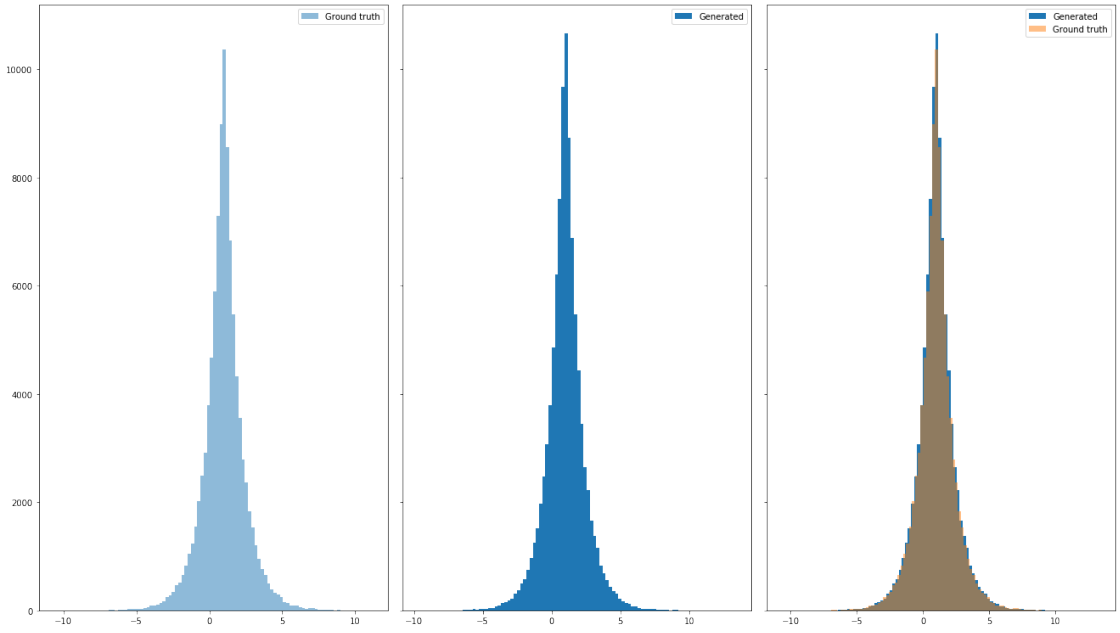
```
In [31]: if(distri == 4):
             print("Enter the mean for ground truth gaussian")
             mean = float(input())
             print("Give std dev for the ground truth gaussian")
             std = float(input())
             while std<0:
                 print("Give std dev for the ground truth properly (>0)")
                 std = float(input())
             Ground_truth = np.random.normal(mean,std,N_points)
             Mean = np.mean(Ground_truth)
             Std_dev = np.std(Ground_truth)
             Generated = np.random.normal(Mean,Std_dev,N_points)
             fig, axs = plt.subplots(1, 3, sharey=True, tight_layout=True)
             axs[0].hist(Ground_truth, bins=n_bins,color='r',alpha=0.5,label="Ground truth")
             axs[0].legend()

             axs[1].hist(Generated, bins=n_bins,label="Generated")
             axs[1].legend()

             axs[2].hist(Generated, bins=n_bins,label="Generated")
             axs[2].hist(Ground_truth, bins=n_bins,color='r',alpha=0.5,label="Ground truth")
             axs[2].legend()

             fig.set_size_inches(18.5, 10.5, forward=True)

         #     plt.savefig("Gauss_Vs_Gauss.png")

Enter the mean for ground truth gaussian
10
Give std dev for the ground truth gaussian
10


/home/legion/.local/lib/python3.6/site-packages/matplotlib/figure.py:2299: UserWarning: This fi
  warnings.warn("This figure includes Axes that are not compatible "
```




```
In [41]: if(distri == 5):
             print("Give Mu for Laplace")
             Mu1 = float(input())
             print("Give Beta for Laplace")
             Beta1 = float(input())
             Ground_truth = np.random.laplace(Mu1,Beta1,N_points)
             Mu = np.median(Ground_truth)
             Beta = np.mean(np.abs(Ground_truth-Mu))
             Generated = np.random.laplace(Mu,Beta,N_points)
             fig, axs = plt.subplots(1, 3, sharey=True, tight_layout=True)
             axs[0].hist(Ground_truth, bins=n_bins,label="Ground truth",alpha=0.5)
             axs[0].legend()

             axs[1].hist(Generated, bins=n_bins, label="Generated")
             axs[1].legend()

             axs[2].hist(Generated, bins=n_bins, label="Generated")
             axs[2].hist(Ground_truth, bins=n_bins,label="Ground truth",alpha=0.5)
             axs[2].legend()

             fig.set_size_inches(18.5, 10.5, forward=True)

         #     plt.savefig("Laplace_Vs_Laplace.png")

Give Mu for Laplace
1
Give Beta for Laplace
```

```
1
```

```
/home/legion/.local/lib/python3.6/site-packages/matplotlib/figure.py:2299: UserWarning: This fi
  warnings.warn("This figure includes Axes that are not compatible "
```



In [ ]: