Problem 1

a.)

Expected Value/Mean:

$$\mu = \int_{-\infty}^{\infty} x f(x)\, dx$$

$$= \int_0^a \frac{x}{a}\, dx$$

$$= \frac{1}{2a}\left[x^2\right]_0^a$$

$$\boxed{\mu = \frac{a}{2}}$$

Variance:

$$\sigma^2 = E[(X - \mu)^2]$$

$$= \int_{-\infty}^{\infty} (x - \mu)^2 f(x)\, dx$$

$$= \int_0^a \left(x - \frac{a}{2}\right)^2 \left(\frac{1}{a}\right) dx$$

$$= \int_0^a \left(x^2 - ax + \frac{a^2}{4}\right)\left(\frac{1}{a}\right) dx$$

$$= \int_0^a \frac{x^2}{a}\, dx - \int_0^a x\, dx + \int_0^a \frac{a}{4}\, dx$$

$$= \left[\frac{x^3}{3a}\right]_0^a - \left[\frac{x^2}{2}\right]_0^a + \left[\frac{ax}{4}\right]_0^a$$

$$= \frac{a^2}{3} - \frac{a^2}{2} + \frac{a^2}{4}$$

$$\boxed{\sigma^2 = \frac{a^2}{12}}$$

Cumulative Distribution Function:

$$F(x) = \int_{-\infty}^{x} f(x')dx'$$

$$= \int_0^x \frac{1}{a}\, dx'$$

$$= \frac{1}{a}\left[x'\right]_0^x$$

$$= \frac{x}{a}$$

$$\boxed{F(x) = \begin{cases} 0, & x < 0, x > a \\ \frac{x}{a}, & 0 < x < a \end{cases}}$$

_b.)_

Expected Value/Mean:

$$\mu = \int_{-\infty}^{\infty} x f(x) \, dx$$

$$= \int_{0}^{\infty} \lambda x e^{-\lambda x} \, dx$$

$$(\text{let } u = \lambda x, du = \lambda \, dx, dv = e^{-\lambda x}, v = -e^{-\lambda x}/\lambda)$$

$$= uv|_0^\infty - \int_0^\infty v \, du$$

$$= -xe^{-\lambda x}\Big|_0^\infty - \int_0^\infty \frac{-\lambda e^{-\lambda x}}{\lambda} \, dx$$

$$= 0 + \left[\frac{-e^{-\lambda x}}{\lambda}\right]_0^\infty$$

$$\boxed{\mu = \frac{1}{\lambda}}$$

Variance:

$$\sigma^2 = E[(X - \mu)^2]$$

$$= E[X^2] - \mu^2$$

$$= \int_0^\infty x^2 f(x) \, dx - \frac{1}{\lambda^2}$$

$$= \int_0^\infty \lambda x^2 e^{-\lambda x} \, dx - \frac{1}{\lambda^2}$$

$$(\text{let } u = \lambda x^2, du = 2\lambda x \, dx, dv = e^{-\lambda x}, v = -e^{-\lambda x}/\lambda)$$

$$= uv|_0^\infty - \int_0^\infty v \, du - \frac{1}{\lambda^2}$$

$$= \left[-x^2 e^{-\lambda x}\right]_0^\infty + \int_0^\infty 2x e^{-\lambda x} \, dx - \frac{1}{\lambda^2}$$

$$= 0 + 2\int_0^\infty x e^{-\lambda x} \, dx - \frac{1}{\lambda^2}$$

$$= \frac{2}{\lambda^2} \left[e^{-\lambda x}(-\lambda x - 1)\right]_0^\infty - \frac{1}{\lambda^2}$$

$$= \frac{2}{\lambda^2} - \frac{1}{\lambda^2}$$

$$\boxed{\sigma^2 = \frac{1}{\lambda^2}}$$

Cumulative Distribution Function:

$$F(x) = \int_{-\infty}^{x} f(x') dx'$$

$$= \int_0^x \lambda e^{-\lambda x'} \, dx'$$

$$= \left[-e^{-\lambda x'}\right]_0^x$$

$$= \left[1 - e^{-\lambda x}\right]$$

$$\boxed{F(x) = 1 - e^{-\lambda x}}$$

Problem 2

(See Jupyter notebook, attached)

Problem 3

a.)

b.)

c.)

d.)

e.)

f.)

g.)

Problem 4

NE250_HW06_mnegus-prob2

November 26, 2017

1 NE 250 – Homework 6

1.1 Problem 1

12/1/2017

```
In [1]: %matplotlib inline
        import numpy as np
        import seaborn as sns
        from matplotlib import pyplot as plt
```

a.)

To sample from a distribution with an exponentially decaying probability density function

$$p(x) = e^{-x},$$

we must first determine the cumulative distribution function (CDF). This is

$$P(x) = \int_0^x e^{-x'} dx'$$

$$P(x) = \left[-e^{-x'} \right]_0^x$$

$$P(x) = 1 - e^{-x}$$

If we take a random sample from this CDF, ξ, then we can invert the function to find a random x corresponding to ξ.

$$\xi = 1 - e^{-x}$$

$$x = -\ln(1 - \xi)$$

Since ξ is chosen uniformly and ranges from 0 to 1, we can say that $1 - \xi$ is a random number with identical distribution to ξ. Then

$$x = -\ln \xi$$

```
In [2]: def sample_exp_decay(num_samples):
            '''Randomly sample from an exponential distribution, and return the sam
            xi = np.random.random(num_samples)
            x = -np.log(xi)
            return x,x.mean(),x.var()
```

1

Now, we want to estimate the mean and variance of samples of size 10, 40, and 160.

```python
In [3]: for num_samples in [10,40,160,10_000]:
            x,mu,sig = sample_exp_decay(num_samples)
            mu = round(mu,4)
            sig = round(sig,4)
            print('{} samples \t  Mean: {} \t Variance: {} '.format(num_samples,mu,
```

```
10 samples          Mean: 1.1785        Variance: 1.1298
40 samples          Mean: 1.0652        Variance: 1.2882
160 samples          Mean: 1.1874         Variance: 1.2137
10000 samples          Mean: 1.0046          Variance: 0.9755
```

We can see that the values for the required 10, 40, and 160 are reasonably close to the values predicted analytically (We can use Problem 1 part b, with $\lambda = 1$ to see that the mean and variance of the distribution should both equal 1). Our precision increases considerably with more histories, and with 10,000 histories, we are quite accurate on both mean and variance.

b.)

Next, we will run several batches of runs, each with a given number of histories. To show that the batch average tends to follow the central limit theorem when several batches are considered, we will plot the averages as a histogram

```python
In [4]: def batch_sampling(num_batches,num_samples):
            '''Run a given number of batches, each with the given number of samples
            batch_means = [sample_exp_decay(num_samples)[1] for i in range(num_batc
            return batch_means
```

```python
In [5]: def plot_hists(data,title,overlay=False,var=1,n=100):
            '''Plot a histogram with from a given dataset.'''
            binnings = 2
            fig, ax = plt.subplots(ncols=binnings,figsize=(16,10))
            bin_count = [10+40*i for i in range(binnings)]
            data=np.array(data)
            for b in range(binnings):
                ax[b].hist(data,bins=bin_count[b],normed=True,alpha=0.7)
                ax[b].set_title('{} with {} bins'.format(title,bin_count[b]),fontsi
                # Add Gaussian overlay (with mean 1 and variance 1)
                if overlay:
                    x = np.arange(0.5,1.5,0.01)
                    g = np.exp(-(x-1)**2/(2*var/n))/np.sqrt(2*np.pi*var/n)
                    ax[b].plot(x,g,'k')
            return ax
```

We will first run 100 batches of the sampling algorithm, each batch with 100 histories.

```python
In [6]: batch_means = batch_sampling(100,100)
        mean_plots = plot_hists(batch_means,'Batch Means')
```

2

Batch Means with 10 bins   Batch Means with 50 bins
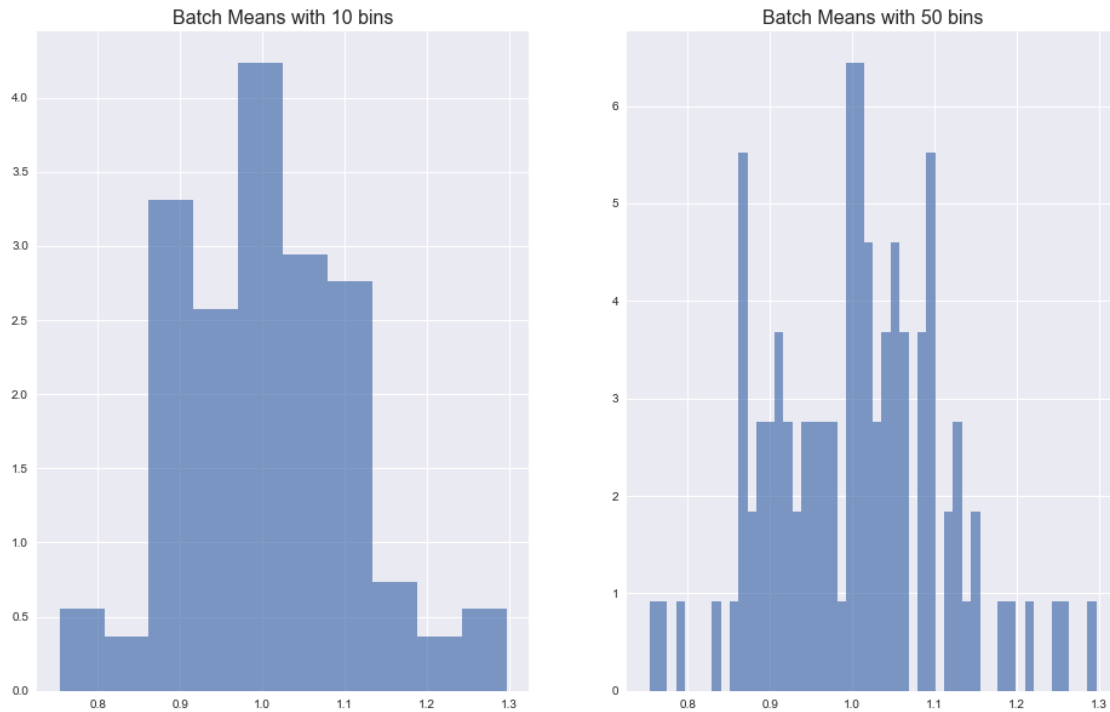
The Gaussian behavior is fairly clear in these graphics, but we can make it more apparent if we increase the number of batches. Let's try 100,000 batches instead, but each with still 100 histories. We can overlay the predicted Gaussian distribution (with mean $\mu = 1$ and variance $\frac{\sigma^2}{n} = 0.01$) given by the central limit theorem on the plots.

```
In [7]: batch_means = batch_sampling(100_000,100)
        mean_plots = plot_hists(batch_means,'Batch Means',overlay=True)
```

Batch Means with 10 bins   Batch Means with 50 bins