0.1 Question 1

The probability density function for a *normal distribution with mean μ and SD σ* is defined by

$$f(x) \;=\; \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}, \quad -\infty < x < \infty$$

The following picture depicts a much-often spouted fact in statistics classes that roughly 68% of the probability for a normal distribution falls within 1 standard deviation of the mean, roughly 95% falls within two standard deviations of the mean, etc:

Use Calculus to prove that the inflection point(s) of the probability density function of the normal distribution occur at $x = \mu \pm \sigma$

Show all steps using LaTeX in the Markdown cell below:

$f'(x) = \frac{d}{dx}(\frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2)}{2\sigma^2}})$

$\frac{d}{dx}(\frac{-(x-\mu)^2)}{2\sigma^2}) = \frac{-(x-\mu)}{\sigma^2}$

$f'(x) = (\frac{1}{\sigma\sqrt{2\pi}})(\frac{-(x-\mu)}{\sigma^2})(e^{\frac{-(x-\mu)^2)}{2\sigma^2}})$

$f''(x) = \frac{d}{dx}((\frac{1}{\sigma\sqrt{2\pi}})(\frac{-(x-\mu)}{\sigma^2})(e^{\frac{-(x-\mu)^2)}{2\sigma^2}}))$

Using the product rule, $f''(x) = (\frac{1}{\sigma\sqrt{2\pi}}\frac{-(x-\mu)}{\sigma^2})'(e^{\frac{-(x-\mu)^2)}{2\sigma^2}}) + (\frac{1}{\sigma\sqrt{2\pi}}\frac{-(x-\mu)}{\sigma^2})(e^{\frac{-(x-\mu)^2)}{2\sigma^2}})'$

$(\frac{1}{\sigma\sqrt{2\pi}}\frac{-(x-\mu)}{\sigma^2})' = (\frac{1}{\sigma\sqrt{2\pi}})(\frac{-1}{\sigma^2})$

$(e^{\frac{-(x-\mu)^2)}{2\sigma^2}})' = (\frac{-(x-\mu)}{\sigma^2})(e^{\frac{-(x-\mu)^2)}{2\sigma^2}})$

$f''(x) = (\frac{1}{\sigma\sqrt{2\pi}})(\frac{-1}{\sigma^2})(e^{\frac{-(x-\mu)^2)}{2\sigma^2}}) + (\frac{1}{\sigma\sqrt{2\pi}}\frac{-(x-\mu)}{\sigma^2})(\frac{-(x-\mu)}{\sigma^2})(e^{\frac{-(x-\mu)^2)}{2\sigma^2}})$

$f''(x) = (\frac{-1}{\sigma^3\sqrt{2\pi}})(e^{\frac{-(x-\mu)^2}{2\sigma^2}}) + (\frac{(x-\mu)^2}{\sigma^5\sqrt{2\pi}})(e^{\frac{-(x-\mu)^2}{2\sigma^2}})$

$$(\frac{-1}{\sigma^3\sqrt{2\pi}})(e^{\frac{-(x-\mu)^2}{2\sigma^2}}) + (\frac{(x-\mu)^2}{\sigma^5\sqrt{2\pi}})(e^{\frac{-(x-\mu)^2}{2\sigma^2}}) = 0$$

$$(\frac{-1}{\sigma^3\sqrt{2\pi}}) + (\frac{(x-\mu)^2}{\sigma^5\sqrt{2\pi}}) = 0$$

$$\frac{-\sigma^2}{\sigma^5\sqrt{2\pi}} + \frac{(x-\mu)^2}{\sigma^5\sqrt{2\pi}} = 0$$

$$-\sigma^2 + (x-\mu)^2 = 0$$

$$(x-\mu)^2 = \sigma^2$$

$$(x-\mu) = \pm\sigma$$

$$x = \mu \pm \sigma$$

0.2 Question 2 :

A hardware store receives a shipment of 10,000 bolts that are supposed to be 12 cm long. The mean of this shipment of 10,000 bolts is indeed 12 cm, and the standard deviation is 0.2 cm.

For the following questions, determine if you have enough information to answer. If you do, then show all steps calculating the answer in the Markdown cell directly below. If you don't, explain what additional information you would need. If you use any theorems, cite the theorem and specify which assumptions are necessary for the theorem to hold.

Question 2a). What is the probability that a randomly chosen bolt is less than 10 cm long?

Question 2b). For quality control, the hardware store chooses 100 bolts at random to measure. They will declare the shipment defective and return it to the manufacturer if the average length of 100 bolts is less than 11.97 cm or greater than 12.04 cm. Find the probability that the shipment is found satisfactory (i.e. not defective).

Question 2a).

There is not enough information to solve. In order to find the probablity that a random chosen bolt is less than 10 cm long, we would need to know it's distribution because from the given info we cannot assume the distribution.

Question 2b).

Using the Central Limit Theorem, we know that the average has a normal distribution.

Thus we know the mean of the distribution is 12 and the standard error is $\frac{.2}{\sqrt{100}} = \frac{.2}{10} = .02$.

Now we can calculate $P(11.97 < X < 12.04)$ by finding the area under the pdf.

The pdf of a normal distribution is $f(x) = (\frac{1}{\sigma\sqrt{2\pi}})(e^{\frac{-(x-\mu)^2}{2\sigma^2}})$.

Thus, we can caluclate $\int_{11.97}^{12.04} f(x)dx$ by using the python function `stats.norm.cdf`.

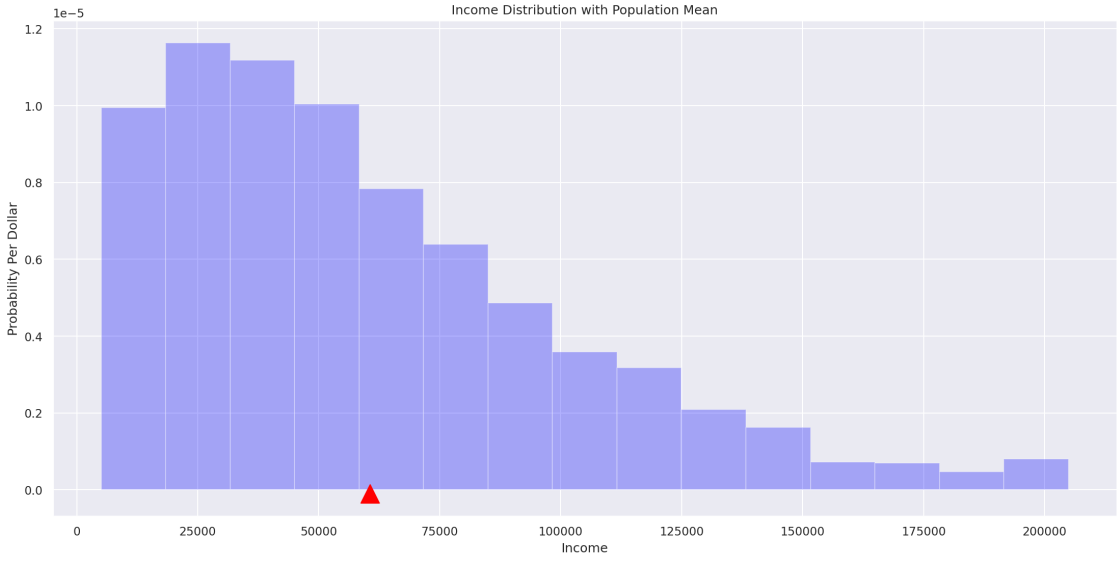
$\int_{11.97}^{12.04} f(x)dx =$ `stats.norm.cdf(12.04, 12, .02) - stats.norm.cdf(11.97, 12, .02)` $= 0.910$

QUESTION 3A: Load the data into a pandas DataFrame called `dfIncome`, calculate the population mean of Income and then make a **density histogram** of the Distribution of the Income data **with 15 bins**. Include a title for your plot and label the x-axis (we have provided a label for the y-axis). Note we have included code to mark where the population mean lies on the histogram.

In [5]:
```
dfIncome = pd.read_csv("income_data.csv")

mean_income = dfIncome["Income"].mean()

print("Population income mean is", mean_income)

income = dfIncome["Income"]

plt.hist(income, bins = 15, density=True, alpha = .3, color='blue')
# write code to plot density histogram above this line

plt.title("Income Distribution with Population Mean")
plt.xlabel("Income")
plt.ylabel("Probability Per Dollar") # Since this is a density histogram, the y-units are proba
#Add a triangle marker to indicate where the population mean is
plt.scatter(mean_income, -.0000001, marker='^', color='red', s=300)
```

```
Population income mean is 60613.8492
```

Out[5]:
```
<matplotlib.collections.PathCollection at 0x7f84157df970>
```



QUESTION 3B: Describe the shape of the Income distribution (i.e. comment on modality and skew)

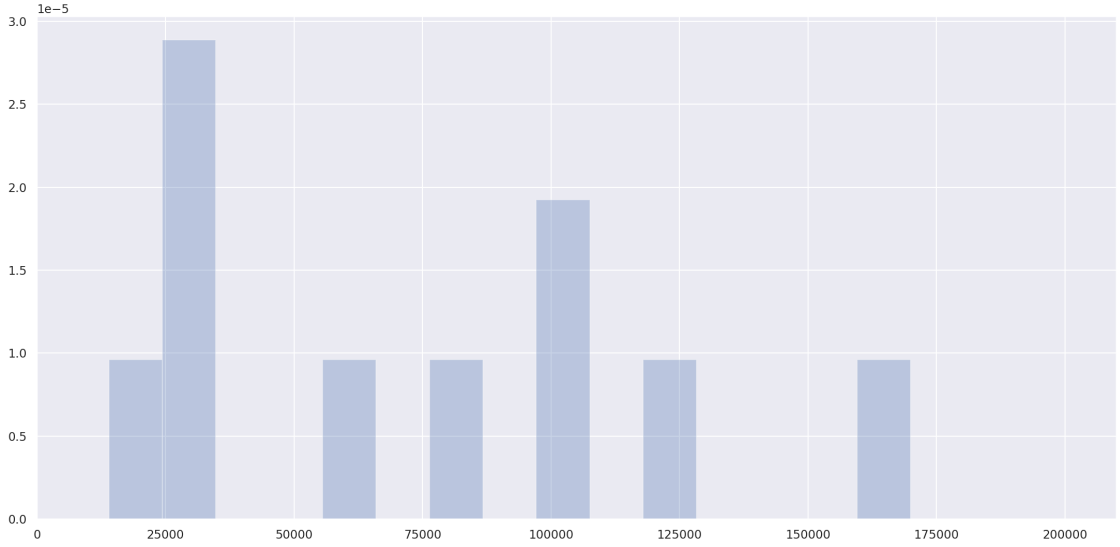
The income distribution is unimodal and is skewed right.

QUESTION 3C:

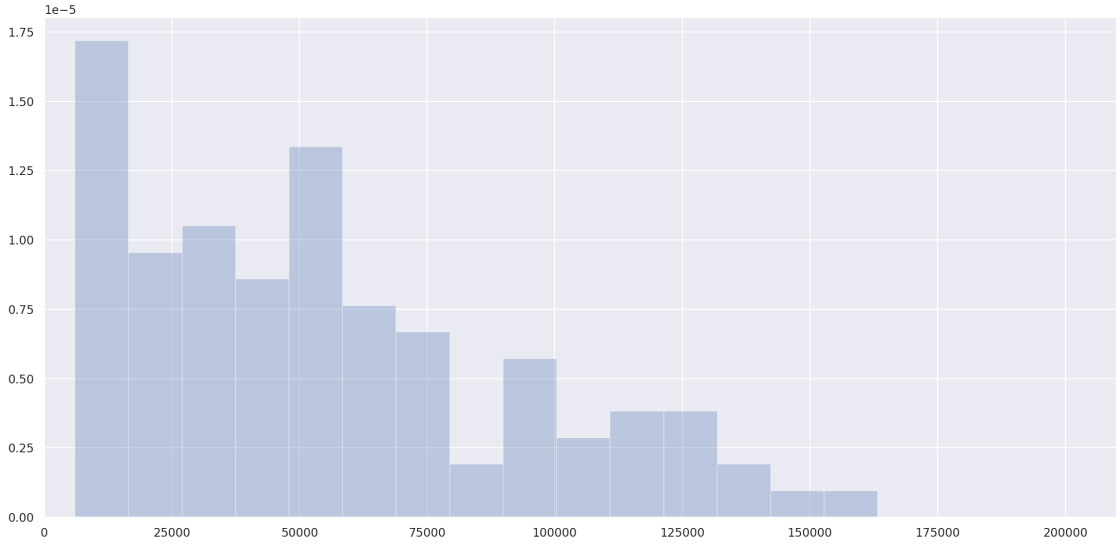
i). Write a function to collect a random sample of size `sample_size` with replacement from `dfIncome` and plot the **density histogram** of the empirical distribution of the income for your sample. Use 15 bins for your histogram and set the x-axis range to be from (0,210000). (Hint: use the dataframe method `.sample()`). Include a title and label for both axes.
Then run the cells provided below to output 3 separate distributions for sample sizes of 10, 100 and 1000.

```
In [6]: def income_sample(df, sample_size):
            sample = df.sample(sample_size, replace=True)
            plt.hist(sample["Income"], bins = 15, density=True, alpha = .3)
            plt.xlim(0,210000)
        # rest of code above this line
```

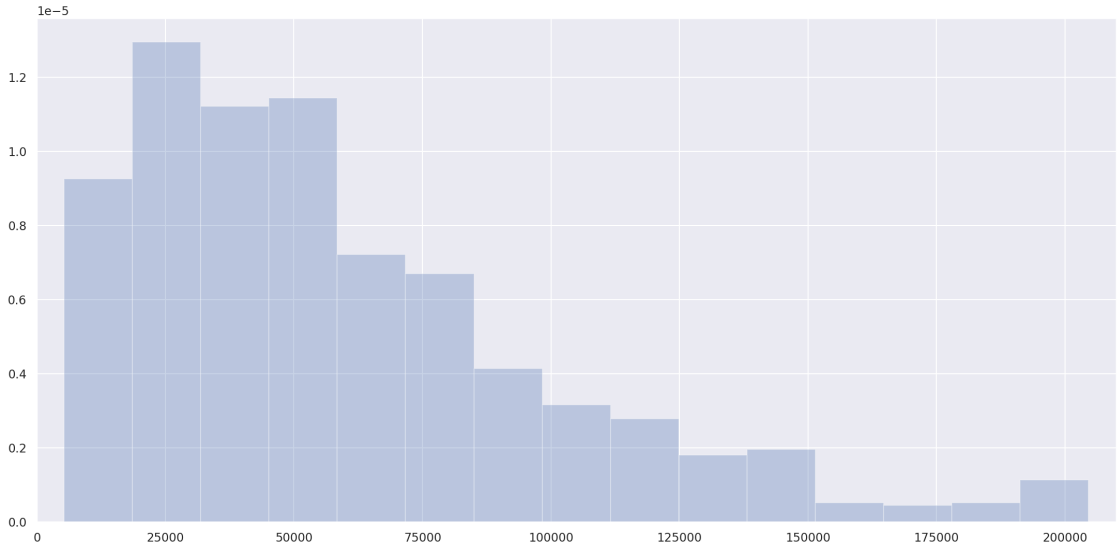
```
In [7]: income_sample(dfIncome,10)
```




```
In [8]: income_sample(dfIncome,100)
```



In [9]: `income_sample(dfIncome,1000)`



Part 3cii). What happens to the shape of the empirical sample distributions of income as you increase the sample size?

The shape of the empirical sample distributions of income starts to look more like the actual population's distribution as you increase the sample size.

QUESTION 3D:

If we want to estimate the **mean** of the population we can draw a sample from the population and compute the sample mean. As we learned in class, since samples can vary, the sample mean can vary and thus it is a random variable and has its own distribution.

i). Complete the function `income_sample_mean` below to randomly sample `sample_size` rows from `dfIncome` with replacement and return the sample mean of income for that sample.

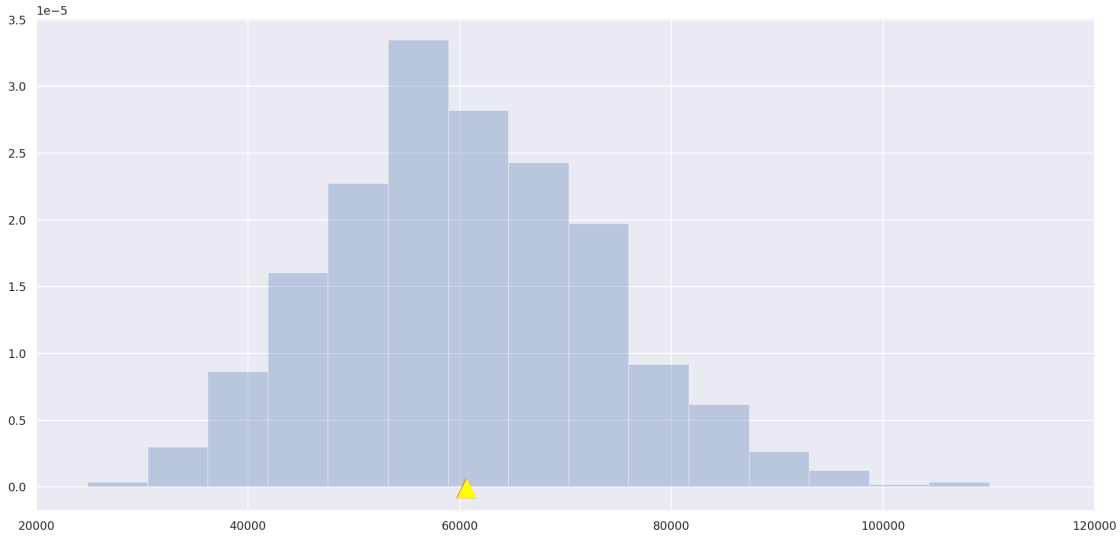
ii). Complete the function `income_sample_dist` below to simulate `num_simulations` of randomly sampling `sample_size` rows from `dfIncome` with replacement and calculate the sample mean of income for each sample. Store the sample means in an np.array called `means`. The function should output a **density** histogram of the empirical sample mean income distribution. On the histogram, include two markers on the histogram: A red one for the population mean (that you calculated in part 3A) and a yellow one for the mean of the `num_simulations` sample mean estimates. Include a title and labels for the x and y-axis.

Then run the cells provided below to output 3 separate distributions for `num_simulations`=1000 and `sample_size` = 10, 100 and 1000

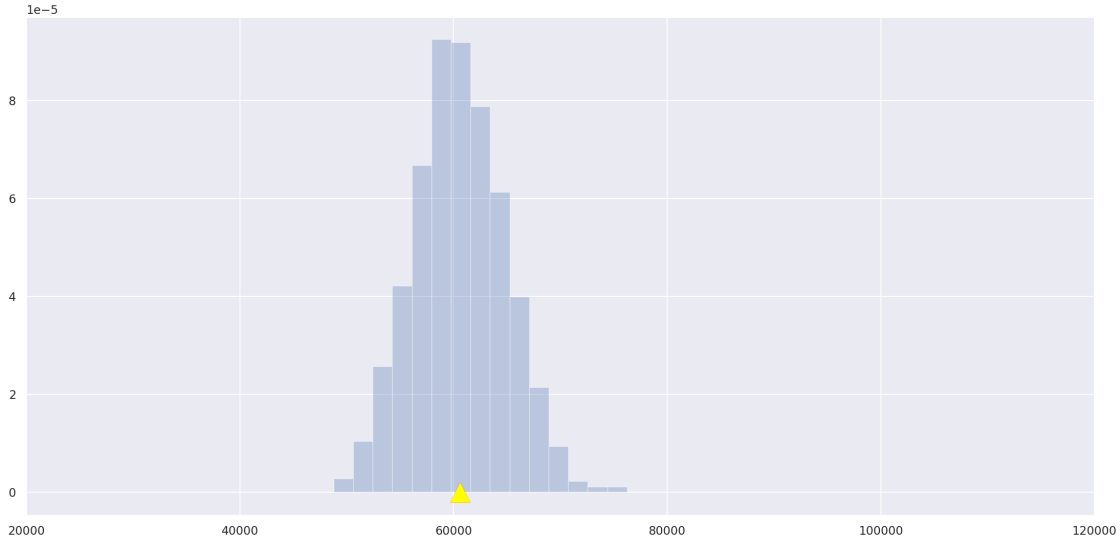
```
In [10]: def income_sample_mean(df, sample_size):
             sample = df.sample(sample_size, replace=True)
             return sample["Income"].mean()
```

```
In [11]: def income_sample_dist(df, sample_size, num_simulations):
             means = np.zeros(num_simulations)
               # 'means' stores "num_simulations" means from samples of size "sample_size"
             for i in range(num_simulations):
                 means[i]= income_sample_mean(df, sample_size)

             plt.xlim([20000,120000])
             plt.hist(means, bins = 15, density=True, alpha = .3)
             plt.scatter(mean_income, -.0000001, marker='^', color='red', s=300)
             plt.scatter(means.mean(), -.0000001, marker='^', color='yellow', s=300)
         # Your code for part (ii) above
```

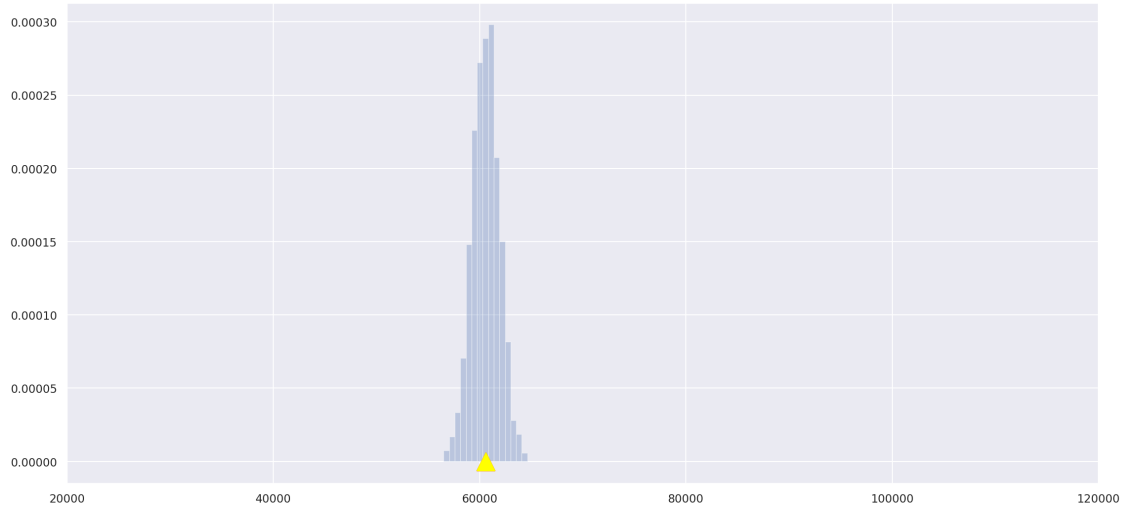
```
In [12]: income_sample_dist(dfIncome, 10, 1000)
```



In [13]: income_sample_dist(dfIncome, 100, 1000)



In [14]: income_sample_dist(dfIncome, 1000, 1000)


0.00030
0.00025
0.00020
0.00015
0.00010
0.00005
0.00000
20000
40000
60000
80000
100000
120000

QUESTION 3E:

Describe the shapes of the empirical sample mean distributions (comment on their modality and skew compared to the modality and skew of the population distribution). What happens to the mean and standard deviations of these distributions as you increase the sample size? What is the name of the theorem that explains what you are observing?

The sample mean distribution is noraml, meaning it is unimodal and uniform (no skew). In comparisonl, the population's distrubution is also unimodal, but is skewed right. Therefore, their distributions are different. As I increase the sample size, the mean becomes more aligned with the population's mean, and the standard deviation decreases. The Central Limit Theorem explains what I am observing.

QUESTION 4H.

Create an array called `simulated_statistics` that contains 50,000 simulated values of the test statistic under the null hypothesis. Assume that the original sample consisted of 210 experiments.

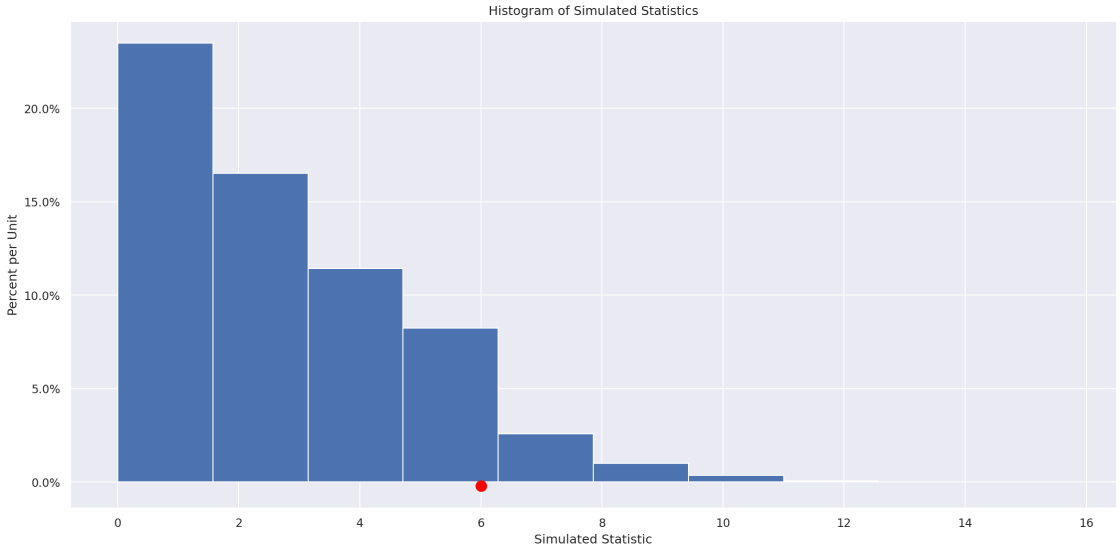
As usual, start by defining a function `one_simulated_statistic()` that simulates one value of the statistic. Your function should use `np.random.DISTRIBUTION` where `DISTRIBUTION` is the distribution you chose in part 4g. Your function should also use your `statistic` function from part 4e.

We have included the code that plots the distribution of the simulated values. The red dot represents the observed statistic you found in Question 4f.

In [43]:
```
def one_simulated_statistic():
    return statistic(np.random.binomial(210,.5)/210, .5)


num_simulations = 50000

simulated_statistics = np.array([])

for i in range(num_simulations):
   simulated_statistics = np.append(simulated_statistics, one_simulated_statistic())



# Run the this cell a few times to see how the simulated statistic changes
one_simulated_statistic()
```

Out[43]:
```
2.857142857142858
```

In [44]:
```
# Run this cell to produce a histogram of the simulated statistics
plt.hist(simulated_statistics, density = True, ec= "white")
plt.xlabel('Simulated Statistic')
plt.ylabel('Percent per Unit')
plt.title('Histogram of Simulated Statistics')
plt.gca().yaxis.set_major_formatter(PercentFormatter(1))
plt.scatter(observed_statistic, -0.002, color='red', s=100);
plt.show()
```


Histogram of Simulated Statistics
Percent per Unit
20.0%
15.0%
10.0%
5.0%
0.0%
0
2
4
6
8
10
12
14
16
Simulated Statistic