Statistics 572
Homework 2

Alejandro Robles

September 13th, 2018

1 Exercise 3.1

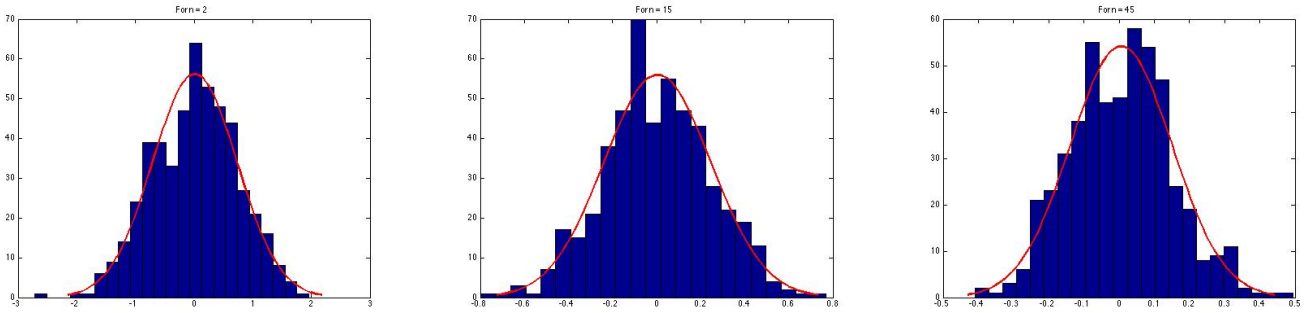
Generate 500 random samples from the standard normal distribution for sample sizes of n = 2, 15, and 45. At each sample size, calculate the sample mean for all 500 samples. How are the means distributed as n gets large? Look at a histogram of the sample means to help answer this question. What is the mean and variance of the sample means for each n? Is this what you would expect from the Central Limit Theorem?

MATLAB Code:

```
title_string = 'For n = %d\n';
mean_string = 'the mean of sample means is %0.5f\n';
var_string = 'the variance of sample means is %0.5f\n';

for n = [2, 15, 45];
    % Generate 500 random samples of size n:
    x = randn(n, 500);
    % Get the mean of each sample:
    xbar = mean(x);

    % Do a histogram with superimposed normal density.
    figure;
    histfit(xbar);
    title(sprintf(title_string,n));

    % Print out statistics from sample means
    fprintf(title_string, n);
    fprintf(mean_string, mean(xbar));
    fprintf(var_string, var(xbar));
end
```

Result:

```
For n = 2
the mean of sample means is -0.03084
the variance of sample means is 0.49394
For n = 15
the mean of sample means is 0.01100
the variance of sample means is 0.07224
For n = 45
the mean of sample means is 0.00563
the variance of sample means is 0.02250
```



(a) N = 2 (b) N = 15 (c) N = 45

Figure 1: Sample Mean Distributions

Discussion: As n increases, the distribution of the sample means converges to a standard normal distribution, which is what we expect from the Central Limit Theorem. Also, the mean of the sample means converges to zero, which shows how accuracy increases as n gets large. Similarly, the variance of the sample means decreases, which shows how precision increases as n gets large.

2 Exercise 3.4

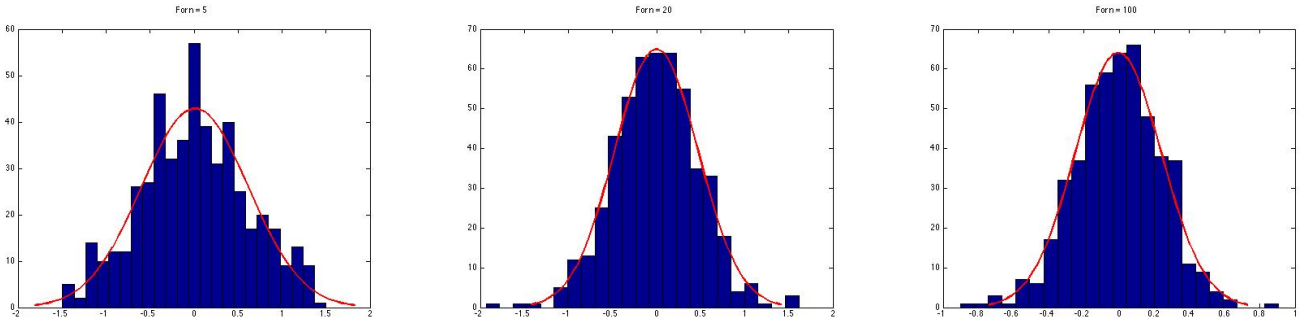
Repeat example 3.1 using different sample sizes. What happens to the coefficient of skewness and kurtosis as the sample size gets large?

MATLAB Code:

```
function gaussian_stats(x, n, statistic, stat_name)
    title_string = 'For n = %d\n';
    mean_string = 'the mean of sample %s is %0.5f\n';
    var_string = 'the variance of sample %s is %0.5f\n';

    % Get Statistic from data
    stat = statistic(x);

    % Do a histogram with superimposed normal density.
    figure;
    histfit(stat);
    title(sprintf(title_string,n));

    % Print out statistics from sample statistic
    fprintf(title_string, n);
    fprintf(mean_string, stat_name, mean(stat));
    fprintf(var_string, stat_name, var(stat));

for n = [5, 20, 100];
    % Generate 500 random samples of size n:
    x = randn(n, 500);

    % Use user defined function for coefficient of skewness and kurtosis
    gaussian_stats(x, n, @skewness, 'coefficient of skewness')
    gaussian_stats(x, n, @kurtosis, 'kurtosis')
end
```

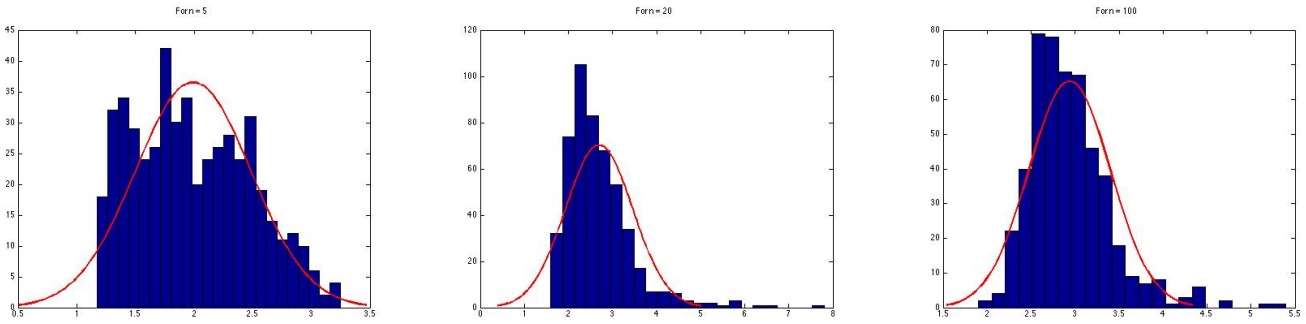
Result:

```
>> exercise_3_4
For n = 5
the mean of sample coefficient of skewness is 0.00998
the variance of sample coefficient of skewness is 0.36682
For n = 5
the mean of sample kurtosis is 1.99214
the variance of sample kurtosis is 0.24284
For n = 20
the mean of sample coefficient of skewness is -0.00266
the variance of sample coefficient of skewness is 0.22430
For n = 20
the mean of sample kurtosis is 2.69534
the variance of sample kurtosis is 0.58848
For n = 100
the mean of sample coefficient of skewness is -0.00451
the variance of sample coefficient of skewness is 0.05980
For n = 100
the mean of sample kurtosis is 2.93805
the variance of sample kurtosis is 0.21556
```



(a) N = 5 (b) N = 20 (c) N = 100

Figure 2: Sample Skewness Distributions



(a) N = 5 (b) N = 20 (c) N = 100

Figure 3: Sample Kurtosis Distributions

Discussion: The same interpretation can be made as in previous exercise. In general, we can say that the distribution of any sample statistic from a standard gaussian will converge to a normal distribution with accuracy and precision increasing as sample size gets large. One thing to note is that kurtosis is initially heavily skewed, which may be due to its cubic nature.

I have a suspicion this is a numerical approach to justify the theorem stated below:

Theorem 6.2.2. *Assume $X_1, \ldots, X_n$ are iid with pdf $f(x;\theta_0)$ for $\theta_0 \in \Omega$ such that the regularity conditions (R0)–(R5) are satisfied. Suppose further that the Fisher information satisfies $0 < I(\theta_0) < \infty$. Then any consistent sequence of solutions of the mle equations satisfies*

$$\sqrt{n}(\widehat{\theta} - \theta_0) \xrightarrow{D} N\left(0, \frac{1}{I(\theta_0)}\right). \tag{6.2.18}$$

Figure 4: Theorem

3 Exercise 3.7

Generate a random sample of size 100 from a normal distribution with mean 10 and variance of 2.

Plot the empirical cumulative distribution function. What is the value of the empirical distribution function evaluated at a point less than the smallest observation in your random sample? What is the value of the empirical cumulative distribution function evaluated at a point that is greater than the largest observation in your random sample?

MATLAB Code:

```
x=randn(1,100)*sqrt(2)+10;
ecdf(x);
[F,y]=ecdf(x);
```

Result:

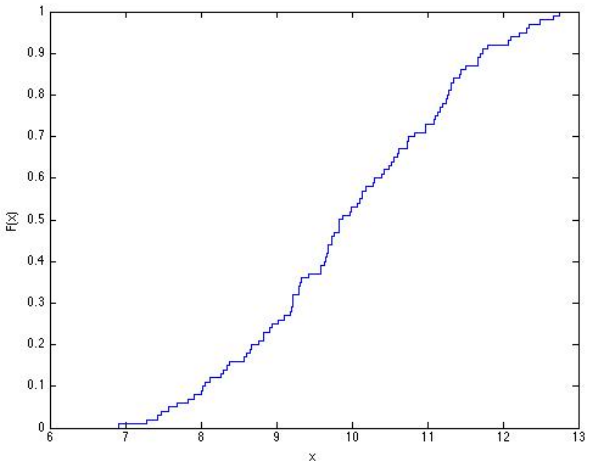


Figure 5: Empirical CDF

Discussion: We know the the empirical distribution function is denoted as the following:

$$\mathrm{F}_n(X) = \begin{cases} 0 & \text{if } X \leq x_1 \\ \frac{j}{n} & \text{if } x_j \leq X \leq x_{j+1} \\ 1 & \text{if } X \geq x_n \end{cases}$$

for $x_1 \leq x_2 \leq ... \leq x_n$

Thus when the empirical distribution is evaluated at a point less than the smallest observation in the random sample $\hat{F}_n(X \leq X_1) = 0$ similarly the value of the empirical distribution evaluated at a point that is greater than the largest observation in the random sample is $\hat{F}_n(X > X_1) = 1$

4 Exercise 3.10

Write a MATLAB function that will return the sample quartiles based on the general definition given for sample quantiles.

MATLAB Code:

```
function q = samplequantiles(x, p)
    x_sorted = sort(x);
    n = length(x);
    j = round(n*p + 0.5);
    q = x_sorted(j);

% Sample size and pth quantile to test
n = 100000;
p = 0.5;

% Generating random sample from standard gaussian
x=randn(n,1);

% Generating quantile using user defined function
q_myfunc = samplequantiles(x, p);

% Generating quantile using using built in function for comparison
q_builtin=quantile(x,p);

% Display Message
message = 'Using %s method, q = %0.5f\n';

fprintf(message, 'samplequantiles', q_myfunc);
fprintf(message, 'build-in', q_builtin);
```

Result:

```
>> exercise_3_10
Using samplequantiles method, q = 0.00009
Using build-in method, q = 0.00009
```

Discussion: To test my function, I sampled from a standard normal distribution with sample size of 100,000. I wanted to find the 0.50th quantile, which we know should be 0 based on our choice of distribution. From this trial, we see that it not only does it do a great job at approximating the 0.50th sample quantile, but it also matches the result from a built in method.

5 Exercise 3.16

Investigate the bias in the maximum likelihood estimate of the variance that is given in equation 3.28. Generate a random sample from the standard normal distribution. You can use the randn function that is available in the standard MATLAB package.

Calculate $\hat{\sigma^2}$ using equation 3.28 and record the value in a vector. Repeat this process (generate a random sample from the standard normal distribution, estimate the variance, save the value) many times. Once you are done with this procedure, you should have many estimates for the variance. Take the mean of these estimates to get an estimate of the expected value of $\hat{\sigma^2}$.

How does this compare with the known value of $\sigma^2 = 1$? Does this indicate that the maximum likelihood estimate for the variance is biased? What is the estimated bias from this procedure?

MATLAB Code:

```
n=1000;
max_iterations = 10000;
var_estimates = zeros(1,max_iterations);

for i = 1:max_iterations
    x=randn(1,n);
    x_bar = mean(x);
    var_estimates(i) = (sum(power((x - x_bar), 2)))/n;
end

histfit(var_estimates);
fprintf( 'For n = %d\n', n);
fprintf('the mean of sample %s is %0.5f\n', 'MLE Estimator', mean(var_estimates));
fprintf('the variance of sample %s is %0.5f\n', 'MLE Estimator', var(var_estimates));
```

Result:

```
>> exercise_3_16
For n = 1000
the mean of sample MLE Estimator is 0.99871
the variance of sample MLE Estimator is 0.00207
```

Discussion:

The simulation indicates that the MLE estimator, $\sigma^2_{MLE} = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2$, is an unbiased estimator for σ^2 for a large sample size. The estimate of the expected value of σ^2_{MLE} is very close to 1 (0.99871) and the variance is small, which indicates high accuracy and precision. The estimated bias is 0.0013.

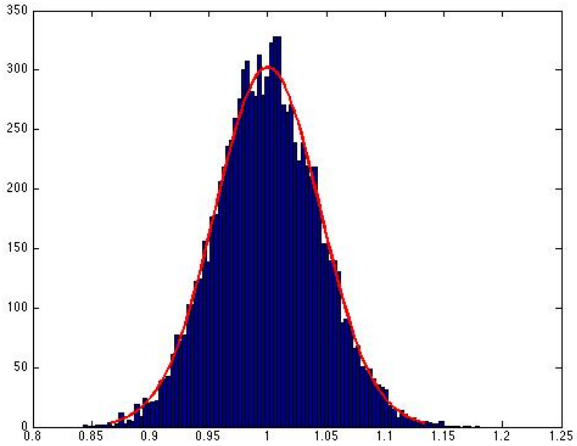


Figure 6: σ^2_{MLE} Distribition for n = 1,000

6 Inverse Transformation with Uniform Distribution

Generate random sample of size 1,000 from Gamma(3,2) using the alternative approach discussed in class.

MATLAB Code:

```
% Parameters
alpha = 3;
beta = 2;
n = 1000;
x = 0:0.01:35;
x_hat = zeros(1,n);


% Approximate CDF values with user defined function
F = zeros(1,length(x));
for i = 1:length(x)
    F(i) = gamma_cdf(x(i), alpha, beta);
end

% Alternative Approach using Uniform Distribution
for i = 1:n
    u = rand;
    index = min(find(F >= u, 1 ));
    x_hat(i) = x(index);
end

% Checking fit
histfit(x_hat, 30, 'gamma')
```

Result:

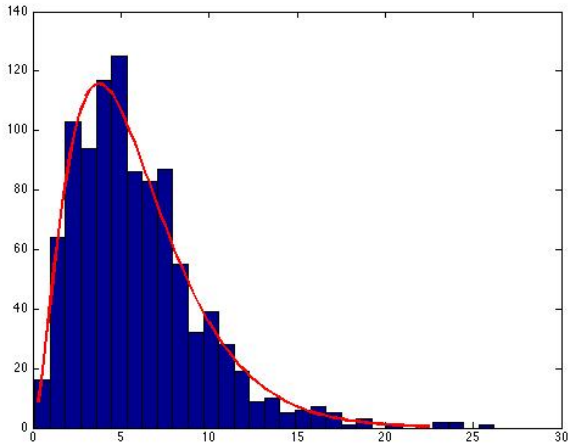


Figure 7: Gamma(3,2) Distribition for n = 1,000

Discussion:

We can see from the plot that it does a good job at randomly sampling from Gamma(3,2). The red curve is the theoretical Gamma probability distribution function, which fits nicely with the sampled data.