Examining the Distribution of Experimental Averages of Samples of the Exponential Distribution

Sean Dunagan

September 27, 2015

Project Overview

For this project, we generated samples consisting of 40 random number generations (hereby referred to as observations) distributed via the exponential distribution. We then took the mean value among the 40 observations in each sample. It is these 40-observation-means (hereby referred to as averages) that we examined, calculating the mean and variance of the distribution of these averages. We then compared the distribution of these averages to a normal distribution whose mean and variance matched the theoretical mean and variance of our set of averages.

The distribution of the individual observations

Attached in the appendix are plots showing the experimental observations with the theoretical exponential distribution overlayed for comparison. You can see above that the set of observations more closely approximates the theoretical distribution curve as the number of observations increases. This is an expected consequence of the law of large numbers. As the number of observations increases, the more closely we expect the distribution of the observations to match the theoretical distribution.

The distribution of the sample averages of the observations

Function generateBatchesOfObservations() was the function which actually executed the random number generation. Function rexp() was used to generate the observations. The mean() function was used to evaluate the averages of each set of 40 observations. mean() was also used to generate the expiremental mean of the averages; sd() was used to compute the standard deviation of the averages (and therefore the variance as well). Function produceStatisticsReport() output the theoretical and experimental values associated with the generated observation averages. Function produceSampleDensityHistogramAgainstNormalCurve() created a histogram overlaying the theoretical normal distribution of observation averages over our set of experimental observation averages. In the histogram, our experimental observation averages are displayed as densities, not as frequencies; this allows us to easily use dnorm() to generate the theoretical normal curve necessary for comparison. The code is commented to explain step-by-step how these plots are generated.

The Theoretical mean of the Sample Averages is the population mean, which is 1/lambda for the exponential distribution. As such, with lambda fixed to .2 for our simulations, the population mean was 5, resulting in the theoretical sample average to also have an expected mean of 5.

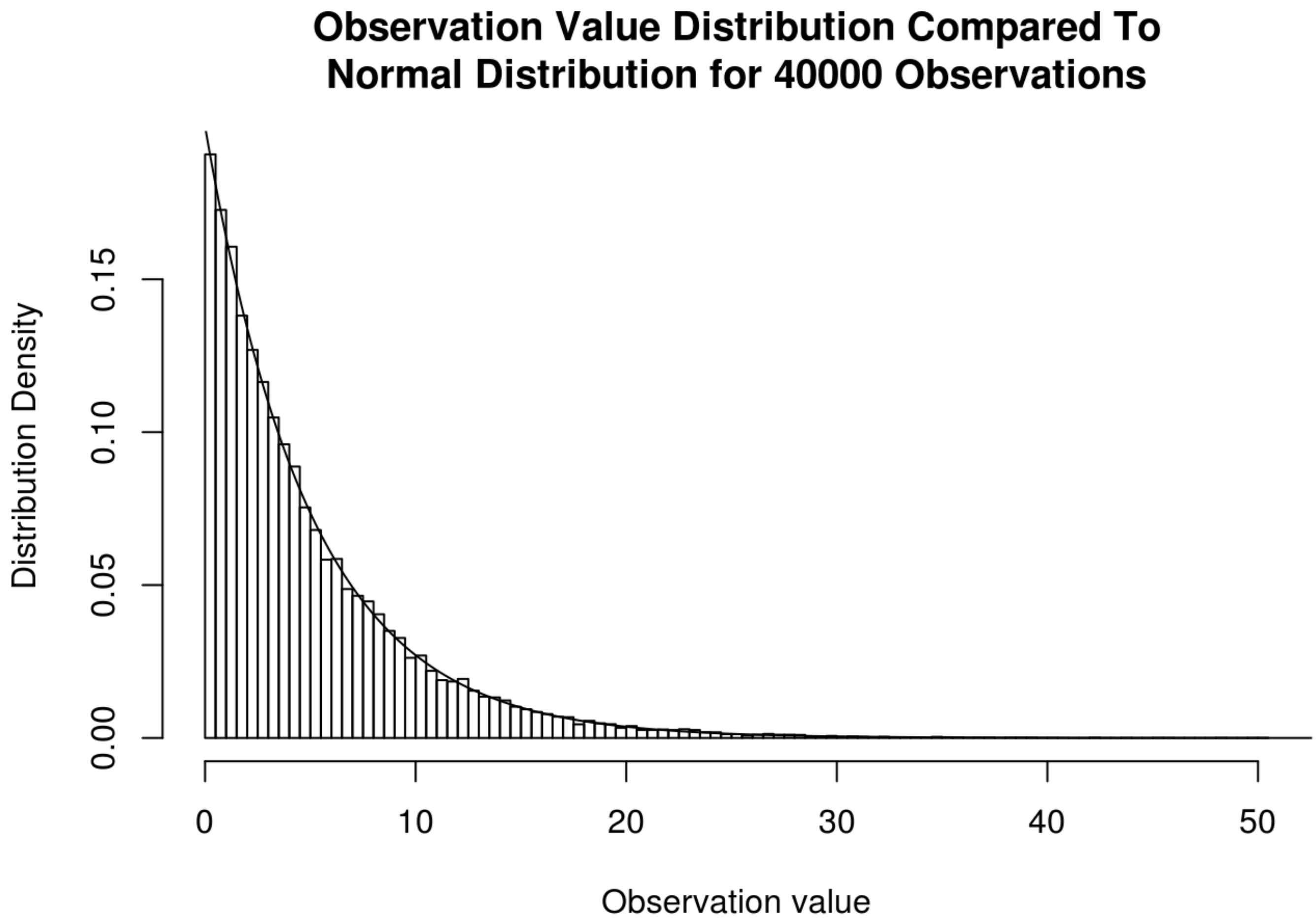
The Theoretical Population Variance refers to the amount of variation we expect in a random variable. This value is independent of the number of observations which may be sampled from a population. The Theoretical Variance of the Sample Averages is the Population Variance divided by the number of observations n which comprise each Average's sample size. As such, the more observations we have comprising each sample average, the less we expect the distribution of the sample averages to vary; this is due to the law of large numbers taking effect. This differs from the population variance because it refers to the distribution of experimental sample AVERAGES, not the distribution of individual members of the population. Examining the distribution of averages allows us to smooth away outliers in the data population.

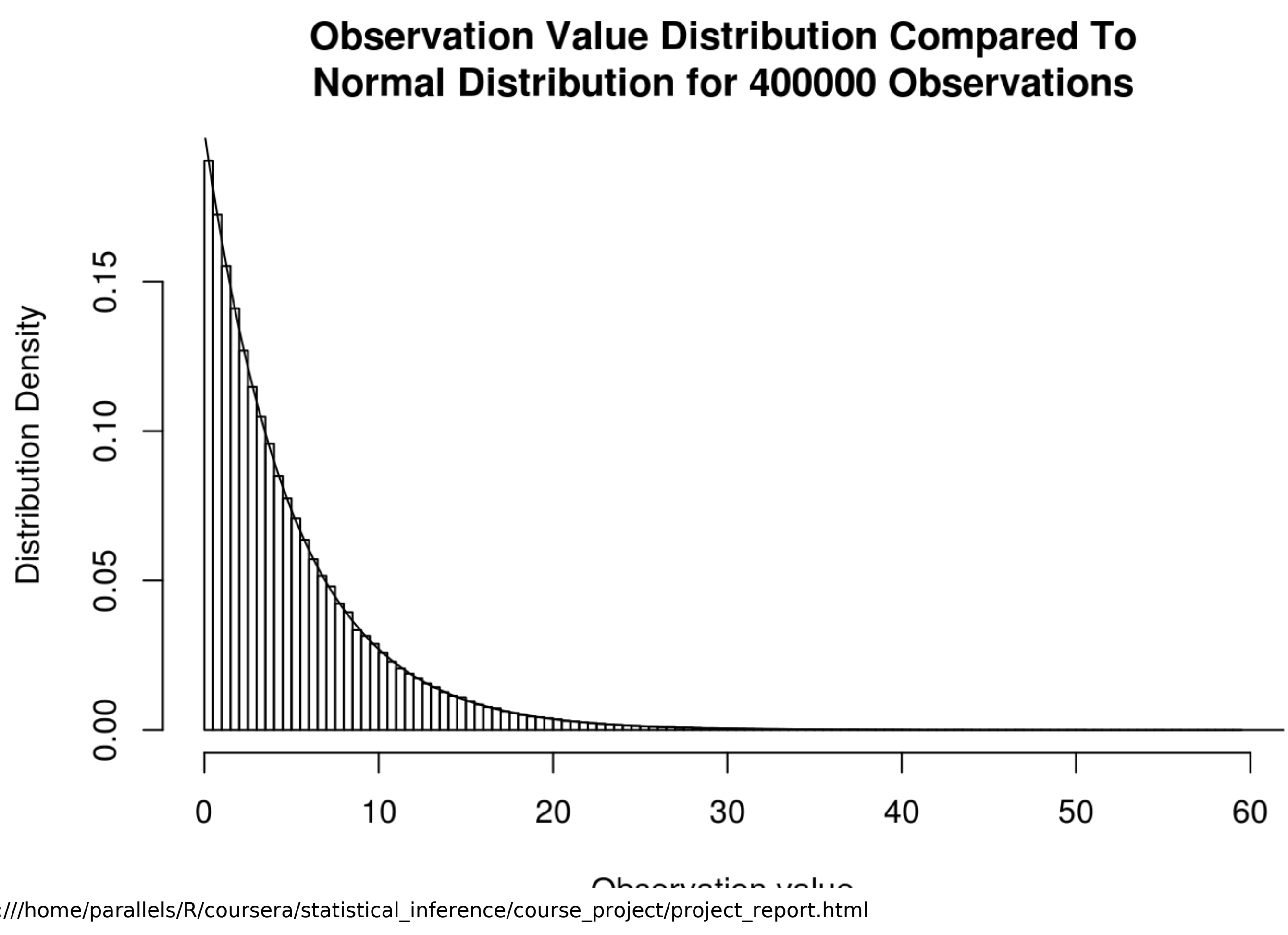
As you can see below, while the experimental mean approximated the population mean for all sample sizes, the variance decreased as the size of the samples increased. Theoretically this is expected as the theoretical variance of a sample mean is the population mean divided by the number of observations per sample, mentioned above. Intuitively this makes sense as the law of large numbers implies that as we take more samples, we expect that the average observation should approach the expected value. This is why the expected value is expected in the first place. The numbers in the table below are compared to their theoretical values in the appendix

```
##          1000 Samples 10000 Samples 100000 Samples
## Mean         4.990898     4.9953912       4.998901
## Variance     0.634041     0.6231779       0.621723
```

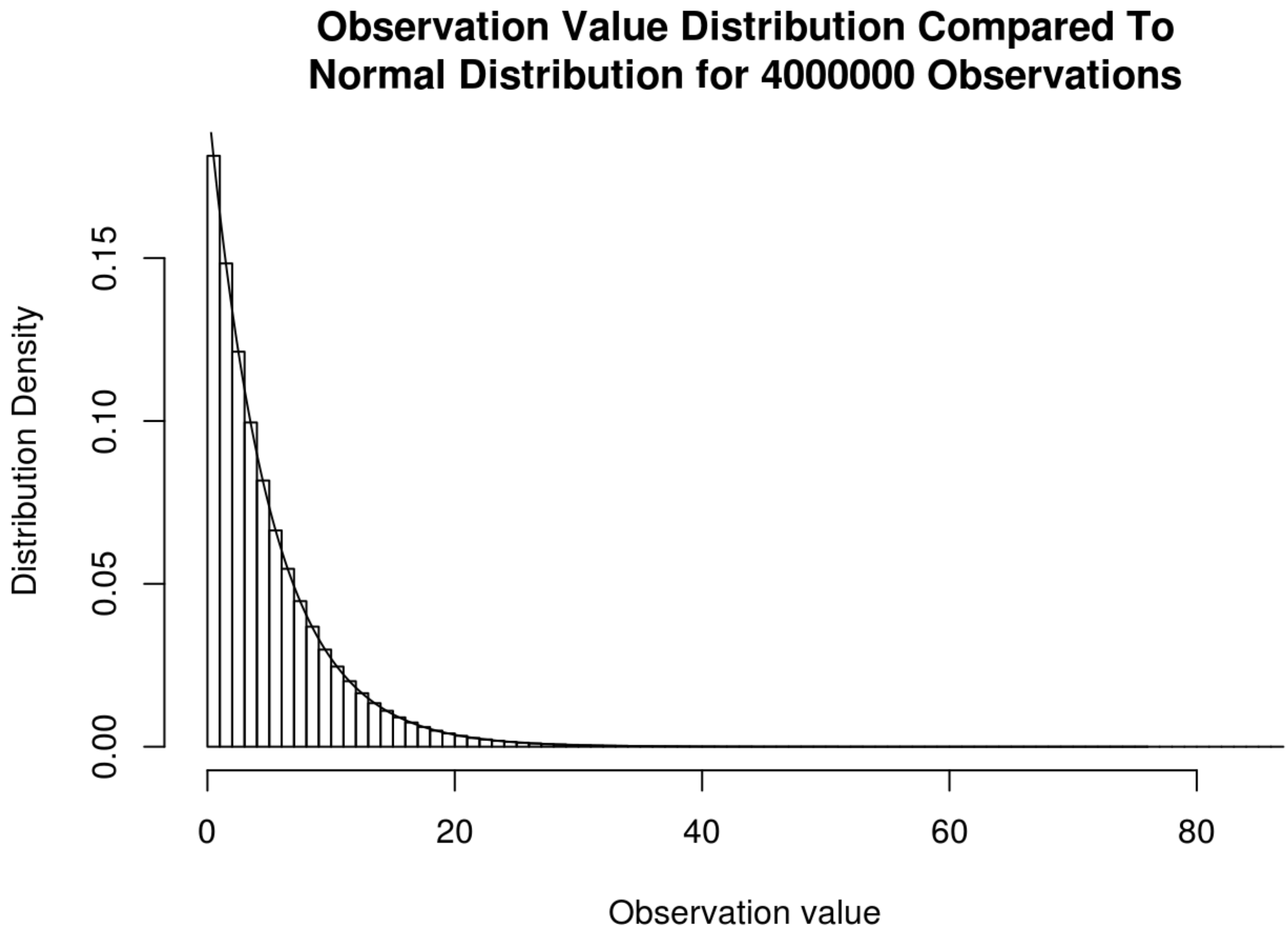
In the histograms shown in the appendix, you can see the distribution of the sample averages more closely approximating a normal distribution as the number of sample averages increases. This is expected as per the Central Limit Theorem and Law of Large Numbers

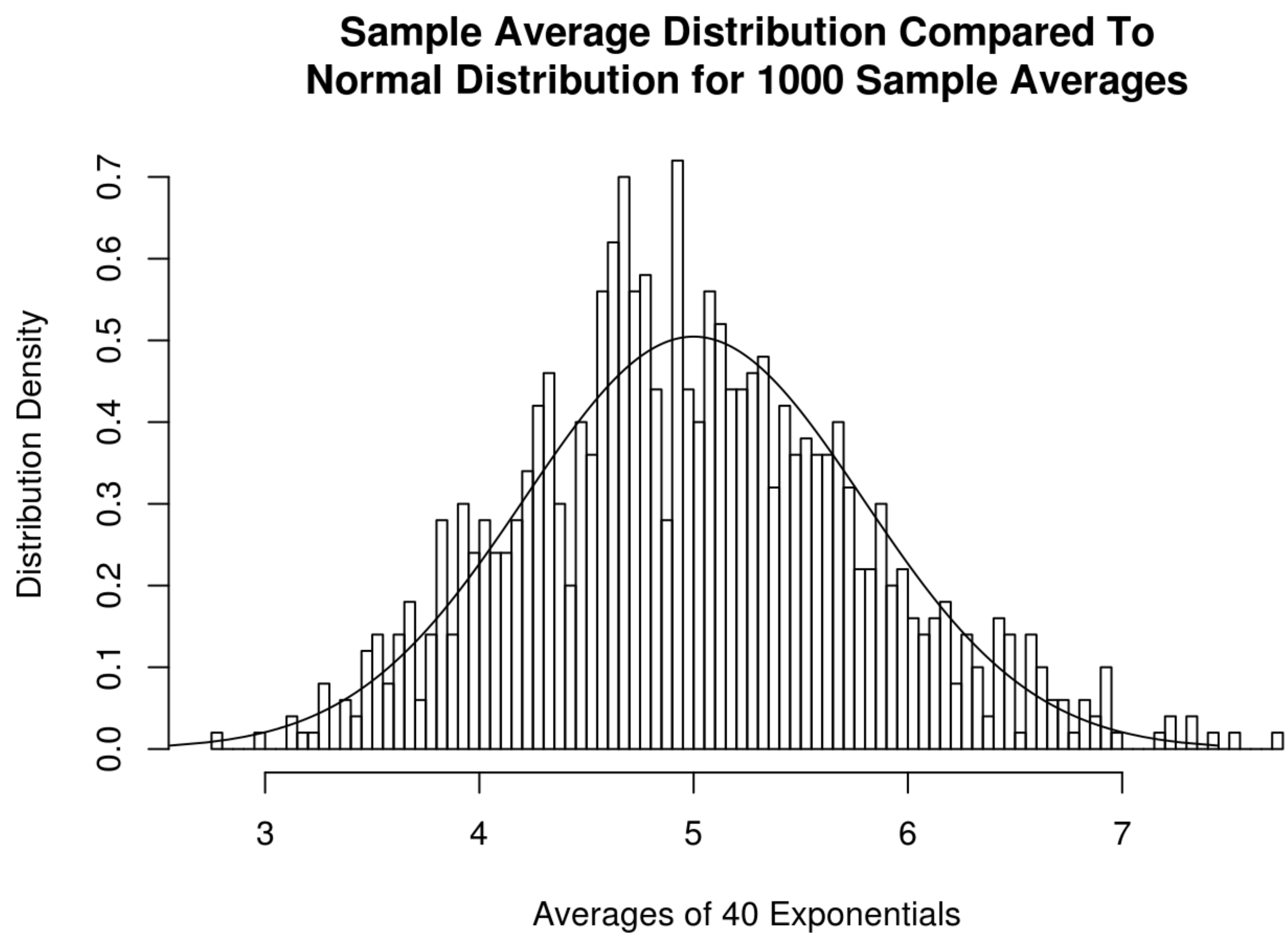
Appendix

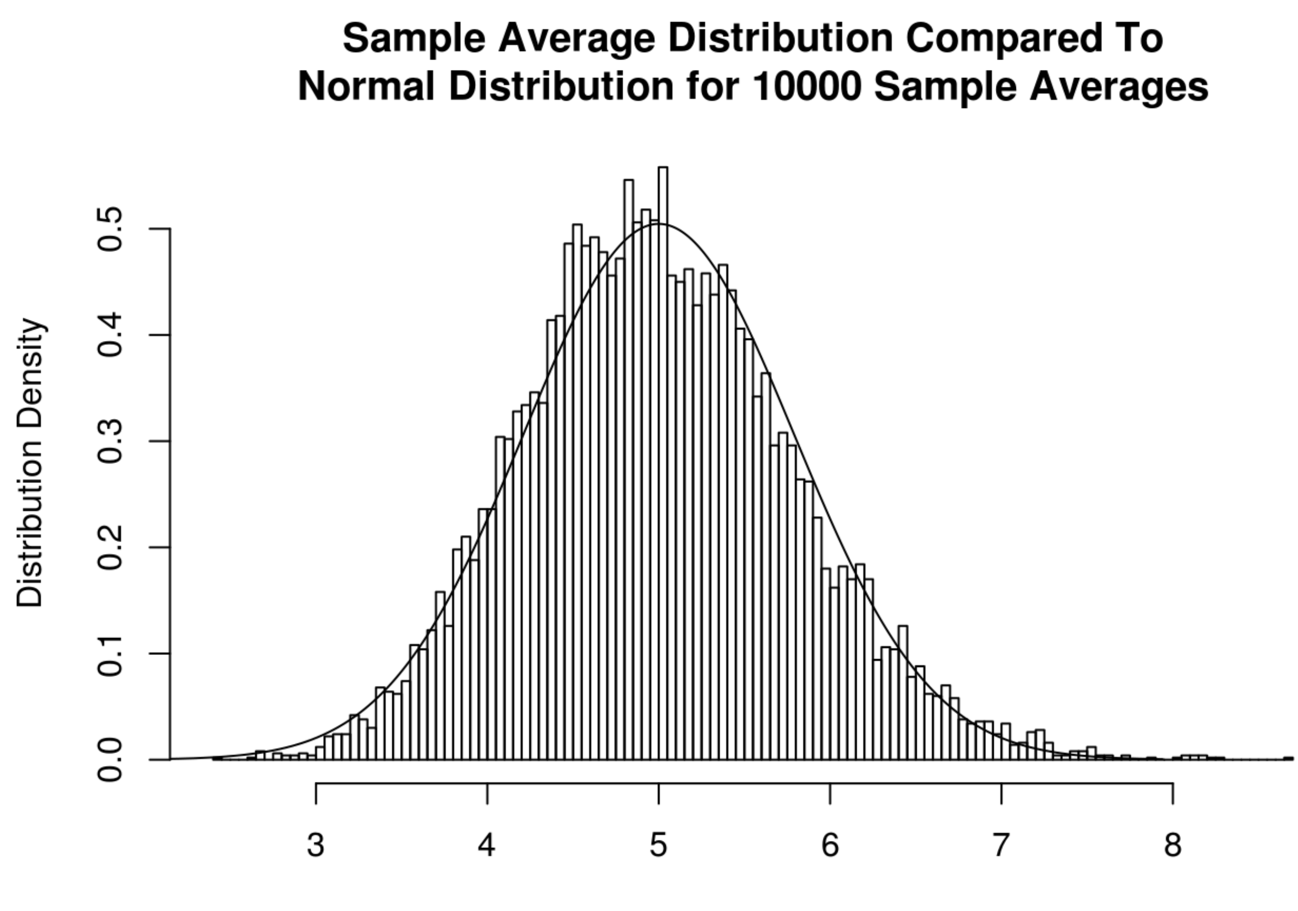

Observation Value Distribution Compared To
Normal Distribution for 40000 Observations
Distribution Density
0.00
0.05
0.10
0.15
0
10
20
30
40
50
Observation value


Observation Value Distribution Compared To
Normal Distribution for 400000 Observations
Distribution Density
0.00
0.05
0.10
0.15
0
10
20
30
40
50
60

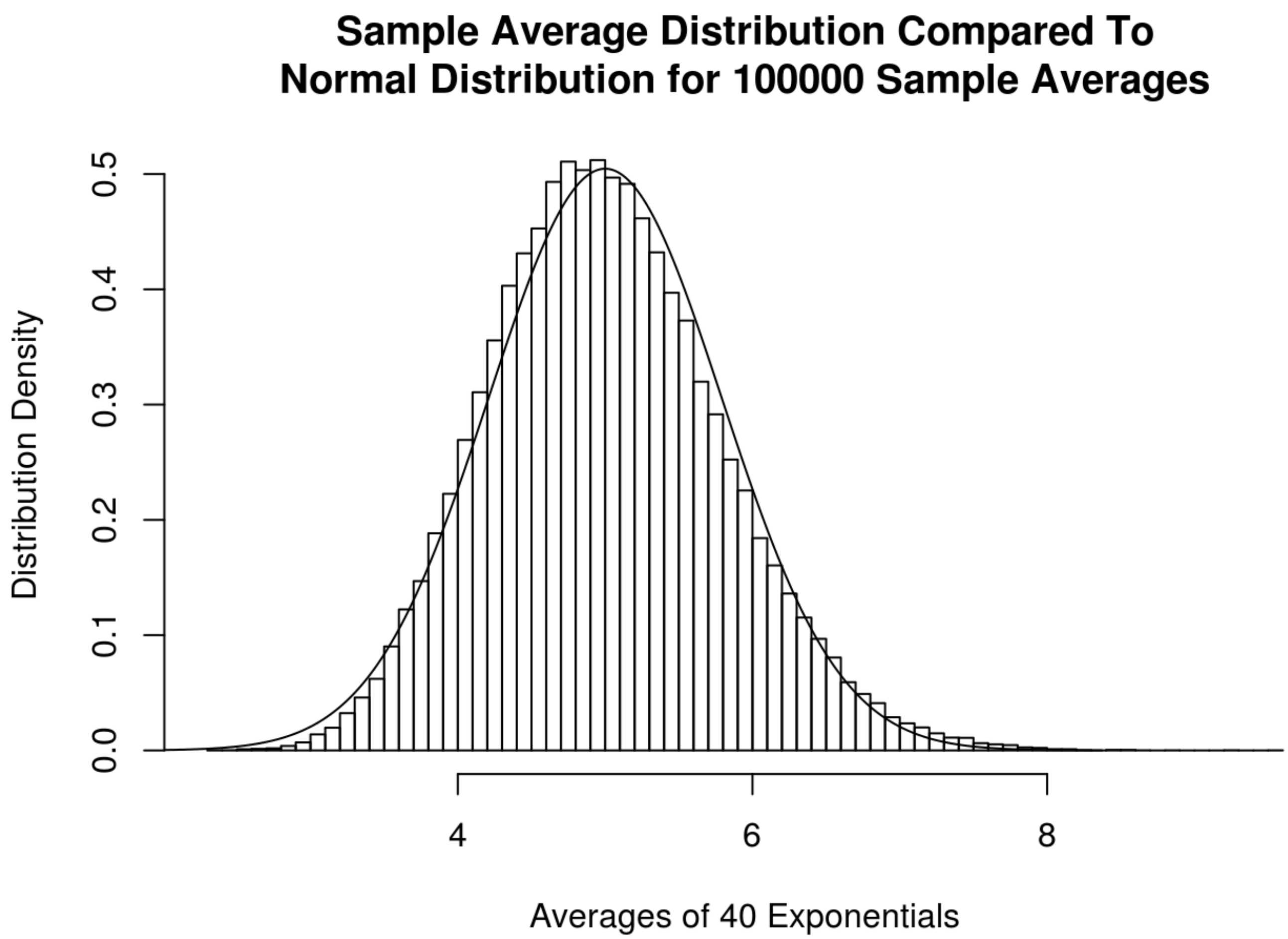
Observation value




Sample Average Distribution Compared To
Normal Distribution for 1000 Sample Averages
Distribution Density
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
3
4
5
6
7
Averages of 40 Exponentials


Sample Average Distribution Compared To
Normal Distribution for 10000 Sample Averages
Distribution Density
0.0
0.1
0.2
0.3
0.4
0.5
3
4
5
6
7
8

Averages of 40 Exponentials




```
##  [1] "1000 samples of the average of 40 exponentials with lambda=0.2 were take
n:"
##  [2] ""
##  [3] "Population Average: 5"
##  [4] "Theoretical Mean of Averages of 40 exponentials: 5"
##  [5] "Experimental Mean of Averages of 40 exponentials: 4.99089829499993"
##  [6] ""
##  [7] "Theoretical Variance of Averages of 40 exponentials: 0.625"
##  [8] "Experimental Sample Variance of Averages of 40 exponentials: 0.6340410159
73402"
##  [9] ""
## [10] "Theoretical Standard Devation of Averages of 40 exponentials: 0.790569415
042095"
## [11] "Experimental Standard Devation of Averages of 40 exponentials: 0.79626692
5078143"
```

```
##  [1] "10000 samples of the average of 40 exponentials with lambda=0.2 were take
n:"
##  [2] ""
##  [3] "Population Average: 5"
##  [4] "Theoretical Mean of Averages of 40 exponentials: 5"
##  [5] "Experimental Mean of Averages of 40 exponentials: 4.9953911570227"
##  [6] ""
##  [7] "Theoretical Variance of Averages of 40 exponentials: 0.625"
##  [8] "Experimental Sample Variance of Averages of 40 exponentials: 0.6231778782
7566"
##  [9] ""
## [10] "Theoretical Standard Devation of Averages of 40 exponentials: 0.790569415
042095"
## [11] "Experimental Standard Devation of Averages of 40 exponentials: 0.78941616
2917671"
```

```
##  [1] "100000 samples of the average of 40 exponentials with lambda=0.2 were tak
en:"
##  [2] ""
##  [3] "Population Average: 5"
##  [4] "Theoretical Mean of Averages of 40 exponentials: 5"
##  [5] "Experimental Mean of Averages of 40 exponentials: 4.99890111950532"
##  [6] ""
##  [7] "Theoretical Variance of Averages of 40 exponentials: 0.625"
##  [8] "Experimental Sample Variance of Averages of 40 exponentials: 0.6217230417
42492"
##  [9] ""
## [10] "Theoretical Standard Devation of Averages of 40 exponentials: 0.790569415
042095"
## [11] "Experimental Standard Devation of Averages of 40 exponentials: 0.78849416
0880404"
```

Below is the R code used to generate this report

```
exponentialObservationsGenerator <- function(number_of_batches_to_construct_with =
1000)
{
    lambda <- 0.2;
    observations_per_batch <- 40;
    number_of_batches_to_observe <- number_of_batches_to_construct_with;
    averages_of_batches <- NULL;
    sample_mean <- NULL;
    sample_standard_deviation <- NULL;
    matrix_of_observation_batches <- NULL;
    vector_of_all_observations <- NULL;
    absolute_filepath_to_export_to <- '/home/parallels/R/coursera/statistical_infe
rence/course_project/';

    setNumberOfBatchesToObserve <- function(number_of_batches_to_set)
    {
        number_of_batches_to_observe <<- number_of_batches_to_set;
    }

    initializeBatchData <- function()
    {
        sample_mean <<- NULL;
        sample_standard_deviation <<- NULL;
        matrix_of_observation_batches <<- NULL;
        averages_of_batches <<- NULL;
        vector_of_all_observations <- NULL;
    }

    getNumberOfObservationsPerBatch <- function()
    {
        observations_per_batch;
    }

    getNumberOfBatchesToObserve <- function()
    {
        number_of_batches_to_observe;
    }

    getTotalObservationsCount <- function()
    {
        total_observations_count <- getNumberOfObservationsPerBatch() * getNumberO
fBatchesToObserve();
        total_observations_count;
    }

    getAllObservations <- function()
    {
        vector_of_all_observations
    }
```

```
    generateBatchesOfObservations <- function()
    {
        initializeBatchData();
        total_number_of_observations <- number_of_batches_to_observe * observation
s_per_batch;
        vector_of_all_observations <<- rexp(total_number_of_observations, lambda);
        # We will make each row of the matrix be a batch of observations
        matrix_of_observation_batches <<- matrix(data = vector_of_all_observation
s, nrow = number_of_batches_to_observe,
                                                 ncol = observations_per_batch, by
row = TRUE);
        # Compute the averages of the batches
        averages_of_batches <<- apply(matrix_of_observation_batches, 1, mean);

        computeSampleMean();
        computeSampleStandardDeviation();

        invisible(matrix_of_observation_batches);
    }
    # Output the theoretical sample average mean/variance and the experimental sam
ple average mean/variance
    produceStatisticsReport <- function()
    {
        #statistics_report_file <- getAbsoluteFilePathForFile('statistics_report.l
og');

        observations_per_batch <- getNumberOfObservationsPerBatch();
        number_of_batches_to_observe <- getNumberOfBatchesToObserve();

        lines_to_write <- c(
            paste0(as.integer(number_of_batches_to_observe), " samples of the aver
age of ", observations_per_batch ," exponentials with lambda=", lambda, " were tak
en:", collapse=""),
            '',
            # Compare Mean of Averages with Theoretical Value
            paste0('Population Average: ', getTheoreticalPopulationMean()),
            paste0('Theoretical Mean of Averages of ', observations_per_batch ,' e
xponentials: ', getTheoreticalSampleMean(), collapse=""),
            paste0('Experimental Mean of Averages of ', observations_per_batch ,'
exponentials: ', getSampleMean(), collapse=""),
            '',
            # Compare Variance of Averages with Theoretical Value
            paste0('Theoretical Variance of Averages of ', observations_per_batch
,' exponentials: ', getTheoreticalSampleVariance(), collapse=""),
            paste0('Experimental Sample Variance of Averages of ', observations_pe
r_batch ,' exponentials: ', getSampleVariance(), collapse=""),
            '',
            # Compare Standard Deviation of Averages with Theoretical Value
            paste0('Theoretical Standard Devation of Averages of ', observations_p
er_batch ,' exponentials: ', getTheoreticalSampleStandardDeviation(), collaps
```

```
e=""),
            paste0('Experimental Standard Devation of Averages of ', observation
s_per_batch ,' exponentials: ', getSampleStandardDeviation(), collapse="")

        );

        #writeLines(lines_to_write, statistics_report_file);
        print(lines_to_write);
    }
    # Show a histogram of the experimental sample averages with the theoretical no
rmal distribution the sample averages
    #   should be distributed by overlayed as a curve
    produceSampleDensityHistogramAgainstNormalCurve <- function()
    {
        #density_comparison_histogram_filepath <- getAbsoluteFilePathForFile('samp
le_density_comparison_histogram.jpg');
        #jpeg(density_comparison_histogram_filepath);
        # Produce the histogram representing the sample data.
        x_axis_label <- paste0("Averages of ", observations_per_batch ," Exponenti
als", collapse="");
        main_label <- paste0("Sample Average Distribution Compared To\nNormal Dist
ribution for ", as.integer(number_of_batches_to_observe), " Sample Averages");
        # Display the samples as DENSITIES, NOT AS FREQUENCIES to allow for compar
ison to the normal density curve
        hist(x=averages_of_batches, breaks=100, xlab=x_axis_label, freq=FALSE, mai
n=main_label, ylab="Distribution Density");
        # Produce the points which will comprise the expected normal distributon c
urve
        # Get a vector of probability steps, with one step for every sample averag
e
        probability_steps_for_quantile_calculation <- getProbabilitySteps(number_o
f_batches_to_observe);
        theoretical_sample_mean <- getTheoreticalSampleMean();
        theoretical_sample_standard_deviation <- getTheoreticalSampleStandardDevia
tion();
        # Compute the theoretical quantile for every sample average
        normal_quantiles <- qnorm(probability_steps_for_quantile_calculation, mea
n=theoretical_sample_mean, sd=theoretical_sample_standard_deviation);
        # Compute the theoretical density at every theoretical quanitle
        normal_densities <- dnorm(normal_quantiles, mean=theoretical_sample_mean,
sd=theoretical_sample_standard_deviation);
        # Plot the theoretical sample average density distribution, which is appro
ximated as a normal distribution curve
        points(x=normal_quantiles, y=normal_densities, type="l");
        #dev.off();
        #invisible(density_comparison_histogram_filepath);
    }

    getProbabilitySteps <- function(number_of_steps)
    {
```

```
        probability_steps_for_quantile_calculation <- seq((1/number_of_steps), 1,
(1/number_of_steps));
        probability_steps_for_quantile_calculation;
    }

    drawTheoreticalExponentialDistributionOverSampleDistribution <- function()
    {
        #exponential_observation_distribution_plot_filepath <- getAbsoluteFilePath
ForFile('exponential_observation_distribution_plot_filepath.jpg');
        #jpeg(exponential_observation_distribution_plot_filepath);
        total_number_of_observations <- getTotalObservationsCount();
        probability_steps_for_quantile_calculation <- getProbabilitySteps(total_nu
mber_of_observations);

        main_label <- paste0("Observation Value Distribution Compared To\nNormal D
istribution for ", as.integer(total_number_of_observations), " Observations");
        hist(x=getAllObservations(), breaks=100, freq=FALSE, main=main_label, xla
b="Observation value", ylab="Distribution Density");

        exp_quantiles <- qexp(probability_steps_for_quantile_calculation, rate=lam
bda);
        exp_densities <- dexp(exp_quantiles, rate=lambda);
        points(x=exp_quantiles, y=exp_densities, type="l");
        #dev.off();
        #invisible(exponential_observation_distribution_plot_filepath);
    }

    getAveragesOfBatches <- function()
    {
        averages_of_batches;
    }

    # The sample mean is the mean of the batch averages
    computeSampleMean <- function()
    {
        sample_mean <<- mean(averages_of_batches);
        sample_mean;
    }

    getSampleMean <- function()
    {
        sample_mean;
    }

    computeSampleStandardDeviation <- function()
    {
        sample_standard_deviation <<- sd(averages_of_batches);
        sample_standard_deviation;
    }
    getSampleStandardDeviation <- function()
```

```
    {
        sample_standard_deviation;
    }
    # Var(sample_mean) = standard_deviation(sample_mean)^2
    getSampleVariance <- function()
    {
        sample_variance <- (sample_standard_deviation^2);
        sample_variance;
    }

    getTheoreticalSampleMean <- function()
    {
        getTheoreticalPopulationMean();
    }
    # Var(sample_mean) = Var(X) / n
    getTheoreticalSampleVariance <- function()
    {
        theoretical_variance_of_population <- getTheoreticalPopulationVariance();
        theoretical_variance_of_sample_mean <- theoretical_variance_of_population
/ observations_per_batch;
        theoretical_variance_of_sample_mean;
    }
    # standard_deviation(sample_mean) = sqrt(Var(sample_mean))
    getTheoreticalSampleStandardDeviation <- function()
    {
        theoretical_variance_of_sample_mean <- getTheoreticalSampleVariance();
        theoretical_standard_deviation_of_sample_mean = sqrt(theoretical_varianc
e_of_sample_mean);
        theoretical_standard_deviation_of_sample_mean;
    }

    # For exponential distribution, E[X] = 1 / lambda = mu
    getTheoreticalPopulationMean <- function()
    {
        theoretical_mean <- (1.0 / lambda);
        theoretical_mean
    }
    # For exponential distribution, Var(X) = 1 / lambda^2
    getTheoreticalPopulationVariance <- function()
    {
        theoretical_variance <- (1.0 / (lambda^2));
        theoretical_variance;
    }
    # For exponential distribution, sd(X) = 1 / lambda
    getTheoreticalPopulationStandardDeviation <- function()
    {
        theoretical_standard_deviation <- (1.0 / lambda);
        theoretical_standard_deviation
    }
    # Functions for outputting the data and the density comparison histogram
```

```
    getAbsoluteFilePathForFile <- function(file_name)
    {
        absolute_file_path <- paste0(absolute_filepath_to_export_to, as.integer(nu
mber_of_batches_to_observe), "_samples_", file_name, collapse = '');
        absolute_file_path;
    }

    list(generateBatchesOfObservations = generateBatchesOfObservations,
         produceStatisticsReport = produceStatisticsReport,
         produceSampleDensityHistogramAgainstNormalCurve = produceSampleDensityHis
togramAgainstNormalCurve,
         drawTheoreticalExponentialDistributionOverSampleDistribution = drawTheore
ticalExponentialDistributionOverSampleDistribution,
         getSampleMean = getSampleMean,
         getSampleVariance = getSampleVariance
     );
}
```