Project 2
STAT 355

Sabbir Ahmed

April 11, 2017

1 Part 1

1000 random samples of size 40 were generated from normal distribution with mean $\mu = 3$ and standard deviation $\sigma = 2$.

```r
# initialize parameters for normal distribution
N <- 40 # size
mu <- 3 # mean
sigma <- 2 # standard deviation
sampMeans <- rep(0, times=NUMSAMPS) # initialize empty array
# generate 1000 samples
for (i in 1:NUMSAMPS){
    generatedData <- rnorm(N, mu, sigma)
    # store the sample means in vector
    sampMeans[i] = mean(generatedData)

    if (i == 1) {
        firstMean = mean(generatedData)
        firstStd = sd(generatedData)
    }

}
```

1.1 Output

The first sample mean and standard deviation were computed:

$$E(\overline{X}) = 2.37843, \ \sigma_{\overline{X}} = 0.31623$$

All the samples were then used to find the sample mean and standard deviation. The theoretical values were also computed based on the relationships:

$$\mu = \mu$$

$$E(\overline{X}) = \mu$$

$$\sigma = \sigma$$

$$\sigma_{\overline{X}} = \frac{\sigma}{\sqrt{n}}$$

	Actual	Theoretical
μ	3.00000	3.00000
$E(\overline{X})$	3.00689	3.00000
σ	2.00000	2.00000
$\sigma_{\overline{X}}$	0.30675	0.31623

1.2 Distribution

Distribution of the data was plotted with a histogram using ggplot2 in Figure 1.

```
ggplot() + aes(sampMeans) +
    geom_histogram(binwidth=0.1, color="black", fill="white") +
    labs(y="Count", x="Sample Means")
```

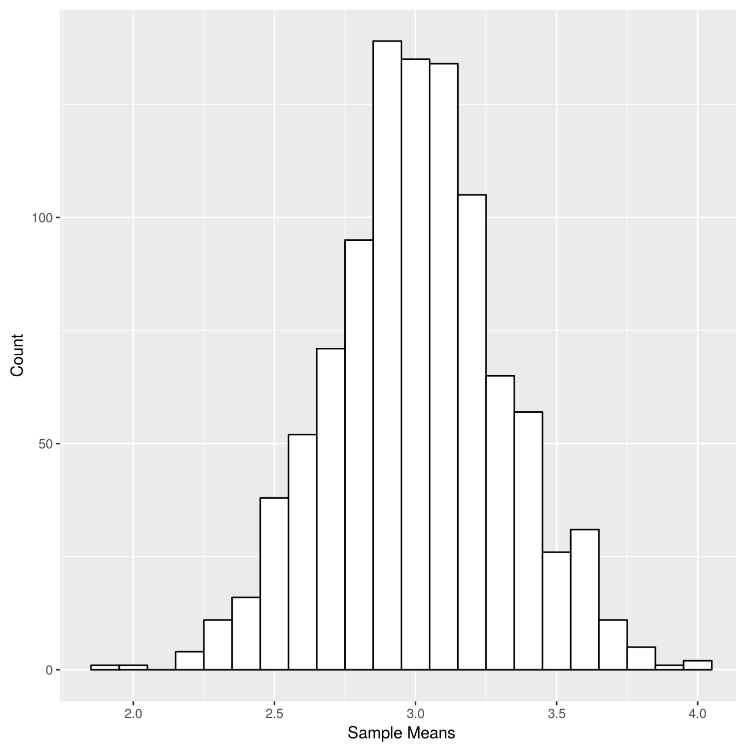

Figure 1: Histogram of the Generated Data

2 Part 2

1000 random samples of size 15 were generated from a binomial distribution with n = 10 and standard deviation p = 0.15.

```
# initialize parameters for binomial distribution
N <- 15
n <- 10
p <- 0.15
sampMeans <- rep(0, times=NUMSAMPS) # initialize empty array
for (i in 1:NUMSAMPS){
```

```
    generatedData <- rbinom(N, n, p)
    sampMeans[i] = mean(generatedData)

    if (i == 1) {
        firstMean = mean(generatedData)
        firstStd = sd(generatedData)
    }

}
```

2.1 Output

The first sample mean and standard deviation were computed:

$$E(\overline{X}) = 1.20000, \ \ \sigma_{\overline{X}} = 0.29155$$

All the samples were then used to find the sample mean and standard deviation. The theoretical values were also computed based on the relationships:

$$\mu = np$$
$$E(\overline{X}) = np$$
$$\sigma = \sqrt{np(1-p)}$$
$$\sigma_{\overline{X}} = \sqrt{\frac{np(1-p)}{N}}$$

	Actual	**Theoretical**
μ	1.50000	1.50000
E($\overline{X}$)	1.51100	1.50000
σ	1.12916	1.12916
$\sigma_{\overline{X}}$	0.29739	0.29155

2.2 Distribution

Distribution of the data was plotted with a histogram using ggplot2 in Figure 2.

```
# plot a histogram of the data
ggplot() + aes(sampMeans) +
    geom_histogram(binwidth=0.2, color="black", fill="white") +
    labs(y="Count", x="Sample Means")
```

3 Part 3

1000 random samples of size 120 were generated from a binomial distribution with n = 10 and standard deviation p = 0.15.

```
# initialize parameters for binomial distribution
N <- 120
n <- 10
p <- 0.15
sampMeans <- rep(0, times=NUMSAMPS) # initialize empty array
for (i in 1:NUMSAMPS){
```

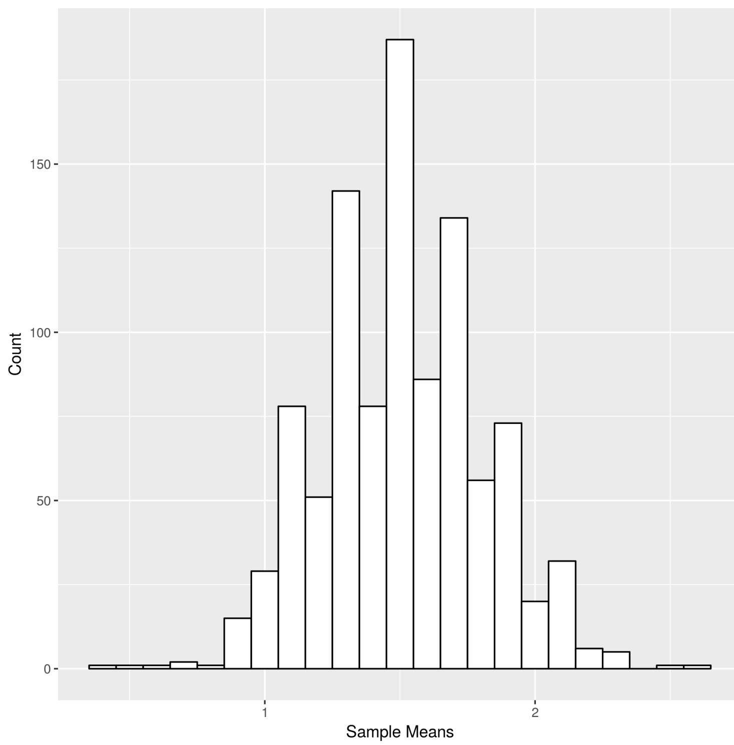

Figure 2: Histogram of the Generated Data

```
        generatedData <- rbinom(N, n, p)
        sampMeans[i] = mean(generatedData)

        if (i == 1) {
            firstMean = mean(generatedData)
            firstStd = sd(generatedData)
        }

    }
```

3.1 Output

The first sample mean and standard deviation were computed:

$$E(\overline{X}) = 1.40833, \ \sigma_{\overline{X}} = 0.10308$$

All the samples were then used to find the sample mean and standard deviation. The theoretical values were also computed based on the relationships:

$$\mu = np$$
$$E(\overline{X}) = np$$
$$\sigma = \sqrt{np(1-p)}$$
$$\sigma_{\overline{X}} = \sqrt{\frac{np(1-p)}{N}}$$

	Actual	Theoretical
μ	1.50000	1.50000
$\mathrm{E}(\overline{X})$	1.50789	1.50000
σ	1.12916	1.12916
$\sigma_{\overline{X}}$	0.10311	0.10308

3.2 Distribution

Distribution of the data was plotted with a histogram using ggplot2 in Figure 3.

```
# plot a histogram of the data
ggplot() + aes(sampMeans) +
    geom_histogram(binwidth=0.1, color="black", fill="white") +
    labs(y="Count", x="Sample Means")
```

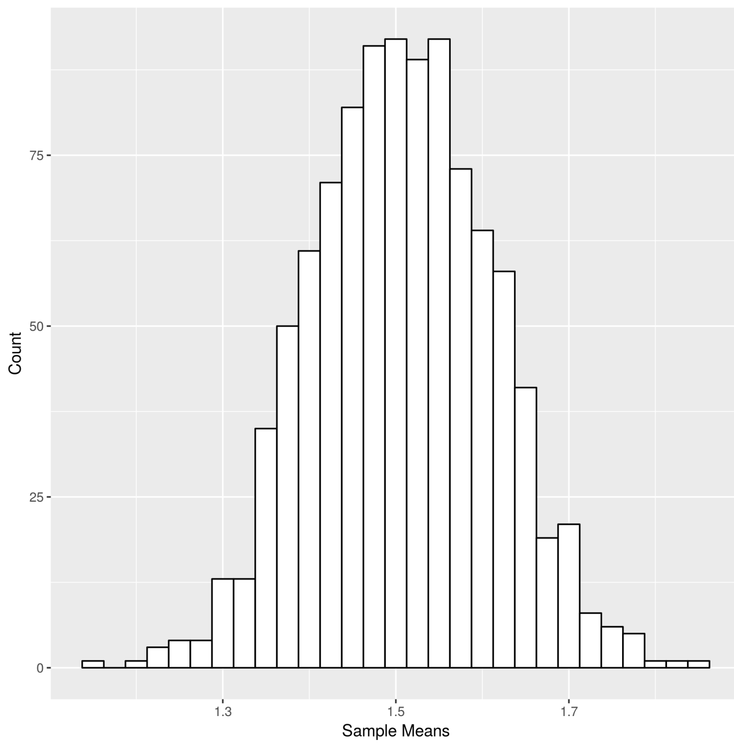

Figure 3: Histogram of the Generated Data

4 Conclusion

```
# main.R
# This file contains the implementation of the functions in the Project 2
# NOTE: THIS SCRIPT WAS COMPILED ON A LINUX MACHINE - SOME STATEMENTS MAY THROW
# WARNINGS OR ERRORS IN OTHER SYSTEMS

library(ggplot2)  # for generating high quality plots
# set.seed(124)  # seed the random generators

# LaTex template for the output
outputTemplate <- "\\subsection{Output}

    The first sample mean and standard deviation were computed:

    \\[ E(\\overline{X}) = %.5f, \\ \\sigma_{\\overline{X}} = %.5f \\]

    All the samples were then used to find the sample mean and standard
    deviation. The theoretical values were also computed based on the
    relationships:

    \\[ \\mu = %s \\]
    \\[ E(\\overline{X}) = %s \\]
    \\[ \\sigma = %s \\]
    \\[ \\sigma_{\\overline{X}} = %s \\]

    \\begin{table}[h]
        \\centering
        \\begin{tabular*}{200pt}{@{\\extracolsep{\\fill}} c c c}

        & \\textbf{Actual} & \\textbf{Theoretical} \\\\
        \\hline
        $\\mu$ & %.5f  & %.5f \\\\
        E($\\overline{X}$) & %.5f & %.5f \\\\
        $\\sigma$ & %.5f & %.5f \\\\
        $\\sigma$\\textsubscript{$\\overline{X}$} & %.5f & %.5f \\\\

        \\end{tabular*}
    \\end{table}
"

# global variables
NUMSAMPS <- 1000  # number of random samples per distribution

randDist <- function(N, a, b, distType, outputFile) {

    # initialize variables to hold data for the first sample
    firstMean <- firstStd <- 0
    mu <- sigma <- n <- p <- 0
    sampMeans <- generatedData <- rep(0, times=NUMSAMPS)  # initialize empty array

    if (distType == "normal") {
        mu <- a
        sigma <- b
    } else if (distType == "binomial") {
        n <- a
        p <- b
    }

    # generate 1000 samples
    for (i in 1:NUMSAMPS) {
        if (distType == "normal") {
            generatedData <- rnorm(N, mu, sigma)
        } else if (distType == "binomial") {
            generatedData <- rbinom(N, n, p)
        }
        # store the sample means in vector
```

```
        sampMeans[i] = sum(generatedData)/N

        if (i == 1) {
            firstMean = sum(generatedData)/N
            if (distType == "normal") {
                firstStd = sigma/sqrt(N)
            } else if (distType == "binomial") {
                firstStd = sqrt(n*p*(1-p)/N)
            }
        }

    }

    outputData <- ''

    if (distType == "normal") {
        outputData <- sprintf(
                outputTemplate,
                firstMean, firstStd,
                "\\mu", "\\mu", "\\sigma", "\\frac{\\sigma}{\\sqrt{n}}",
                mu, mu,
                sum(sampMeans)/NUMSAMPS, mu,
                sigma, sigma,
                sd(sampMeans), sigma/sqrt(N)
        )
    } else if (distType == "binomial") {
        outputData <- sprintf(
                outputTemplate,
                firstMean, firstStd,
                "np", "np", "\\sqrt{np(1-p)}", "\\sqrt{\\frac{np(1-p)}{N}}",
                n*p, n*p,
                sum(sampMeans)/NUMSAMPS, n*p,
                sqrt(n*p*(1-p)), sqrt(n*p*(1-p)),
                sd(sampMeans), sqrt(n*p*(1-p)/N)
        )
    }

    # dump output to LaTex modules
    sink(outputFile, append=FALSE, split=FALSE)
    cat(outputData)
    sink()

    return(sampMeans)

}


plotHist <- function(sampMeans, figureFile, binwidth) {

    # plot a histogram of the data
    histPlot <- ggplot() + aes(sampMeans) +
        geom_histogram(binwidth=binwidth, color="black", fill="white") +
        labs(y="Count", x="Sample Means")

    ggsave(filename=paste0("figures/", figureFile), plot=histPlot)

}

# ------------------------------ Part 1 ------------------------------

# initialize parameters for normal distribution
N <- 40  # size
mu <- 3  # mean
sigma <- 2  # standard deviation

sampMeans <- randDist(N, mu, sigma, "normal", "part1.tex")
plotHist(sampMeans, "hist1.png", 0.1)
```

```
# ------------------------------ Part 2 ------------------------------

# initialize parameters for binomial distribution
N <- 15
n <- 10
p <- 0.15

sampMeans <- randDist(N, n, p, "binomial", "part2.tex")
plotHist(sampMeans, "hist2.png", 0.1)


# ------------------------------ Part 3 ------------------------------

# initialize parameters for binomial distribution
N <- 120
n <- 10
p <- 0.15

sampMeans <- randDist(N, n, p, "binomial", "part3.tex")
plotHist(sampMeans, "hist3.png", 0.025)
```
