Project 2
STAT 355

Sabbir Ahmed

April 9, 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

samp_means <- c() # initialize empty array
# generate 1000 samples
for (i in 1:1000){
    generatedData <- rnorm(N, mu, sigma)
    # store the sample means in vector
    sampMeans[i] = mean(generatedData)
}
```

1.1 Output

Actual:	**Theoretical**:
μ: 3	μ: 3
$E(\overline{X})$: 2.9903	$E(\overline{X})$: 2.9903
σ: 2	σ: 2
$\sigma_{\overline{X}}$: 0.311	$\sigma_{\overline{X}}$: 0.316

1.2 Distribution

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

```r
# plot a histogram
ggplot() + aes(generatedData) +
    geom_histogram(binwidth=1, color="black", fill="white") + labs(y="Count")
```

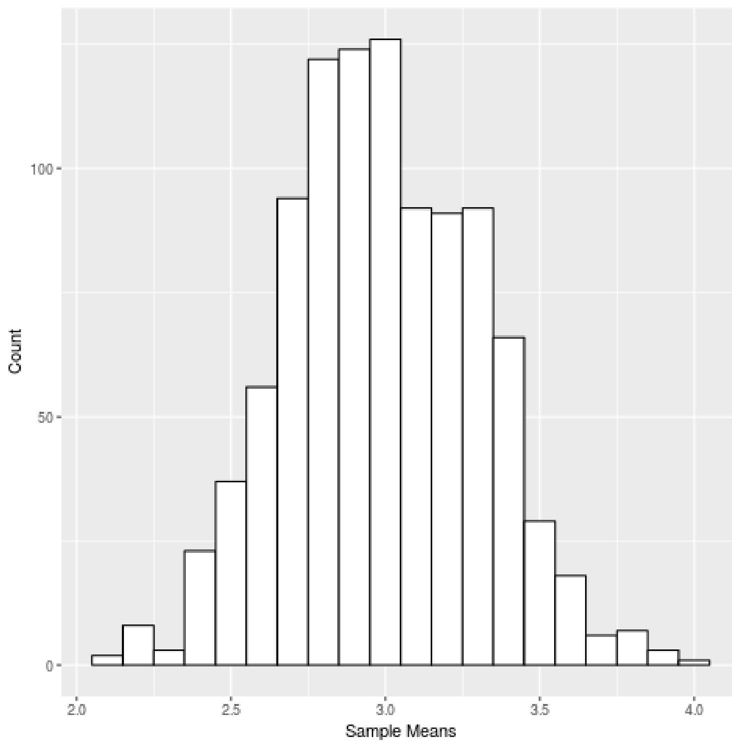

Figure 1: Histogram of the Generated Data

```r
# 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)

outputTemplate <- "\\subsection{Output}

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

        \\textbf{Actual}: & \\textbf{Theoretical}: \\\\
        $\\mu$: %1.0f  & $\\mu$: %1.0f \\\\
        E($\\overline{X}$): %.4f & E($\\overline{X}$): %.4f \\\\
        $\\sigma$: %1.0f & $\\sigma$: %1.0f \\\\
        $\\sigma$\\textsubscript{$\\overline{X}$}: %.3f & $\\sigma$\\textsubscript{$\\overline{X}$}: %.3f \\\\

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

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

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

sampMeans <- c()  # initialize empty array
# generate 1000 samples
for (i in 1:1000){
    generatedData <- rnorm(N, mu, sigma)
    # store the sample means in vector
    sampMeans[i] = mean(generatedData)
}


# save output
sink("part1.tex", append=FALSE, split=FALSE)
cat(
    sprintf(
        outputTemplate,
        mu, mu,
        mean(sampMeans), mean(sampMeans),
        sigma, sigma,
        sd(sampMeans), sigma/sqrt(N)
    )
)
sink()

png(filename="figures/hist1.png")
# plot a histogram of the data
ggplot() + aes(sampMeans) +
    geom_histogram(binwidth=0.1, color="black", fill="white") +
    labs(y="Count", x="Sample Means")

dev.off()


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

# initialize parameters for binomial distribution
```

```
N <- 15
n <- 10
p <- 0.15
sampMeans <- c()
for (i in 1:1000){
    generatedData <- rbinom(N, n, p)
    sampMeans[i] = mean(generatedData)
}


# save output
sink("part2.out", append=FALSE, split=FALSE)

# save output
sink("part2.tex", append=FALSE, split=FALSE)
cat(
    sprintf(
        outputTemplate,
        n*p, n*p,
        mean(sampMeans), mean(sampMeans),
        n*p*(1-p), n*p*(1-p),
        sd(sampMeans), sd(sampMeans)
    )
)
sink()

png(filename="figures/hist2.png")

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

dev.off()


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

# initialize parameters for binomial distribution
N <- 120
n <- 10
p <- 0.15
sampMeans <- c()
for (i in 1:1000){
    generatedData <- rbinom(N, n, p)
    sampMeans[i] = mean(generatedData)
}


# save output
sink("part3.out", append=FALSE, split=FALSE)

# save output
sink("part3.tex", append=FALSE, split=FALSE)
cat(
    sprintf(
        outputTemplate,
        n*p, n*p,
        mean(sampMeans), mean(sampMeans),
        n*p*(1-p), n*p*(1-p),
        sd(sampMeans), sd(sampMeans)
    )
)
sink()

png(filename="figures/hist3.png")

# plot a histogram of the data
```

```
ggplot() + aes(sampMeans) +
    geom_histogram(binwidth=0.3, color="black", fill="white") +
    labs(y="Count", x="Sample Means")

dev.off()
```
