

Project 1

STAT 355

Sabbir AHMED

March 13, 2017

1 Part 1

1000 random numbers were generated using a Bernoulli random variable with $n = 20$, $p = 0.4$

```
# initialize parameters for binomial distribution
x <- 20
N <- 1000
p <- 0.4
generatedData <- rbinom(N, x, p)
```

1.1 Distribution

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

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

1.2 Mean, Variance, and Standard Deviation

The mean, variance and standard deviation of the data were computed with the following snippet to generate Table 1:

```
# print out the mean, variance and standard deviation
paste("Mean:", signif(mean(generatedData), digits=4),
      "| Variance:", signif(var(generatedData), digits=4),
      "| Standard Deviation:", signif(sqrt(var(generatedData)), digits=4))
```

Table 1: Statistics		
Mean	Variance	Standard Deviation
7.953	4.832	2.198

1.3 Summary

Summary statistics were generated with the following snippet to generate Table 2:

```
# print out the summary statistics
summary(generatedData)
```

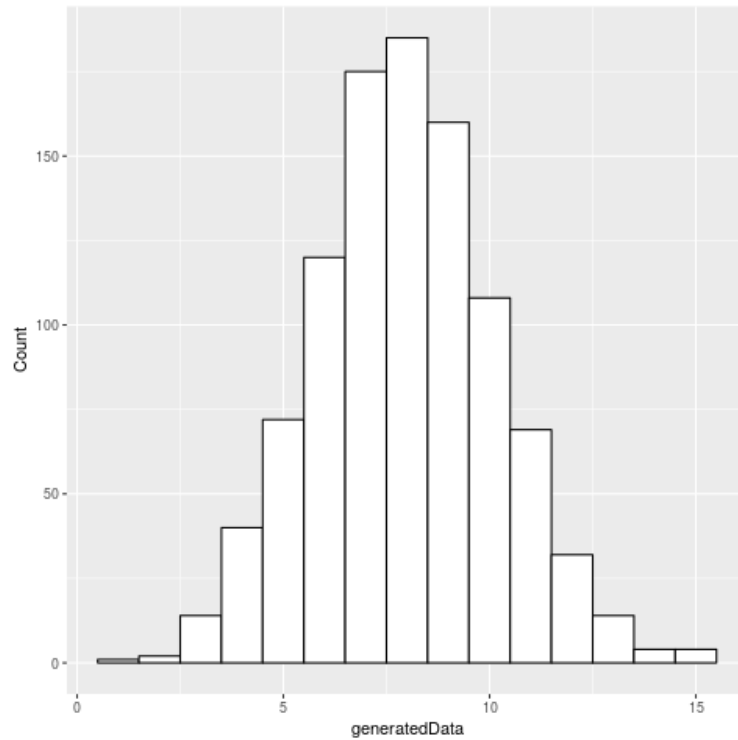


Figure 1: Histogram of the Generated Data

Table 2: Summary Statistics					
Minimum	1st Quar.	Median	Mean	3rd Quar.	Maximum
1.000	7.000	8.000	7.953	9.000	15.000

1.4 Relative Frequencies

The frequencies and relative frequencies of the generate data were computed with the following snippet:

```
# frequencies
ftable(generatedData)

# relative frequencies
ftable(generatedData)/N
```

The relative frequencies were then plotted in a barplot as in Figure 2:

```
# generate barplot of the relative frequencies
ggplot(data=relfreq, aes(x=generatedData, y=Freq)) +
  geom_bar(color="black", fill="white", stat="identity") +
  labs(y="Relative Frequency")
```

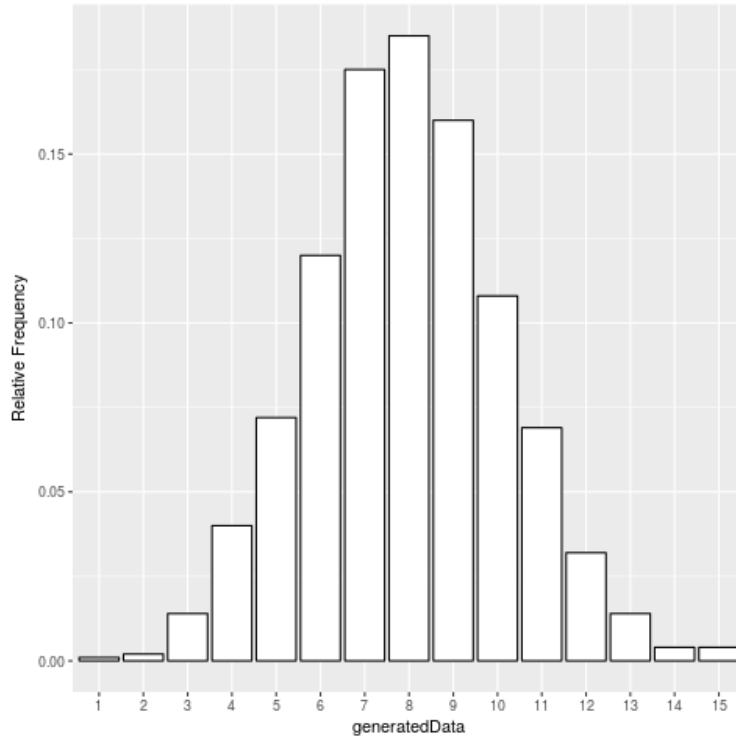


Figure 2: Relative Frequencies of the Generated Data

1.5 Binomial Table

The cumulative probability $P(X = x)$ of the relative frequencies were computed with the following snippet to generate Figure 4:

```
# print out the cumulative probability to compare with the binomial
# distribution table
cumProb <- 0 # initialize binomial cumulative probability

# iterate through the entire dataframe
for (i in 1:(nrow(relfreq))) {
  cumProb=cumProb+relfreq[i, "Freq"]
  print(
    paste0(
      "b(", relfreq[i, "generatedData"], ', ', x, ', ', p, ") -> ",
      cumProb
    )
  )
}
```

2 Part 2

A very large batch of components has arrived at a distributor. The batch can be characterized as acceptable only if the proportion of defective components is at most 0.10. The distributor decides to randomly select 10 components and to accept the batch only if the number of defective components in the sample is at most 2.

Table A.1 Cumulative Binomial Probabilities (cont.)

$$B(x; n, p) = \sum_{y=0}^x b(y; n, p)$$

d. $n = 20$

	p														
	0.01	0.05	0.10	0.20	0.25	0.30	0.40	0.50	0.60	0.70	0.75	0.80	0.90	0.95	0.99
0	.818	.358	.122	.012	.003	.001	.000	.000	.000	.000	.000	.000	.000	.000	.000
1	.983	.736	.392	.069	.024	.008	.001	.000	.000	.000	.000	.000	.000	.000	.000
2	.999	.925	.677	.206	.091	.035	.004	.000	.000	.000	.000	.000	.000	.000	.000
3	1.000	.984	.867	.411	.225	.107	.016	.001	.000	.000	.000	.000	.000	.000	.000
4	1.000	.997	.957	.630	.415	.238	.051	.006	.000	.000	.000	.000	.000	.000	.000
5	1.000	1.000	.989	.804	.617	.416	.126	.021	.002	.000	.000	.000	.000	.000	.000
6	1.000	1.000	.998	.913	.786	.608	.250	.058	.006	.000	.000	.000	.000	.000	.000
7	1.000	1.000	1.000	.968	.898	.772	.416	.132	.021	.001	.000	.000	.000	.000	.000
8	1.000	1.000	1.000	.990	.959	.887	.596	.252	.057	.005	.001	.000	.000	.000	.000
9	1.000	1.000	1.000	.997	.986	.952	.755	.412	.128	.017	.004	.001	.000	.000	.000
10	1.000	1.000	1.000	.999	.996	.983	.872	.588	.245	.048	.014	.003	.000	.000	.000
11	1.000	1.000	1.000	1.000	.999	.995	.943	.748	.404	.113	.041	.010	.000	.000	.000
12	1.000	1.000	1.000	1.000	1.000	.999	.979	.868	.584	.228	.102	.032	.000	.000	.000
13	1.000	1.000	1.000	1.000	1.000	1.000	.994	.942	.750	.392	.214	.087	.002	.000	.000
14	1.000	1.000	1.000	1.000	1.000	1.000	.998	.979	.874	.584	.383	.196	.011	.000	.000
15	1.000	1.000	1.000	1.000	1.000	1.000	1.000	.994	.949	.762	.585	.370	.043	.003	.000
16	1.000	1.000	1.000	1.000	1.000	1.000	1.000	.999	.984	.893	.775	.589	.133	.016	.000
17	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	.996	.965	.909	.794	.323	.075	.001
18	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	.999	.992	.976	.931	.608	.264	.017
19	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	.999	.997	.988	.878	.642	.182

Figure 3: Cumulative Binomial Probability Table from the Textbook

2.1 Probability

The probabilities of accepting the lot were computed with the proportion of defective components as $p = 0.01, 0.05, 0.10, 0.20, 0.25$.

```
# initialize variables
probs <- c(0.01, 0.05, 0.10, 0.20, 0.25)
proportions <- rep(0, times=length(probs))

# generate binomial probabilities with b(2, 10, p)
for (i in 1:length(proportions)) {
  proportions[i] = signif(pbinom(2, size=10, prob=probs[i]), digits=4)
}
bnomDF <- data.frame(p=probs, px=proportions)
```

2.2 Characteristic Curve

The characteristic curve for the acceptance sampling plan was generated as Figure 5 with the following snippet:

```
# plot the characteristic curve
ggplot(bnomDF, aes(x=p, y=px)) + geom_line() + geom_point(color="red") +
  labs(x="p", y="P(Acceptance)")
```

2.3 Probability with Different Maximum Acceptable Sample

The probabilities of accepting the lot were computed with the same proportions of defective components with the maximum sample changed to 1 and its characteristic curve was plotted as Figure 6

```
# generate binomial probabilities with b(1, 10, p)
for (i in 1:length(proportions)) {
  proportions[i] = signif(pbinom(1, size=10, prob=probs[i]), digits=4)
}
bnomDF <- data.frame(p=probs, px=proportions)
```

```

"b(1, 20, 0.4) -> 0.001"
"b(2, 20, 0.4) -> 0.003"
"b(3, 20, 0.4) -> 0.017"
"b(4, 20, 0.4) -> 0.057"
"b(5, 20, 0.4) -> 0.129"
"b(6, 20, 0.4) -> 0.249"
"b(7, 20, 0.4) -> 0.424"
"b(8, 20, 0.4) -> 0.609"
"b(9, 20, 0.4) -> 0.769"
"b(10, 20, 0.4) -> 0.877"
"b(11, 20, 0.4) -> 0.946"
"b(12, 20, 0.4) -> 0.978"
"b(13, 20, 0.4) -> 0.992"
"b(14, 20, 0.4) -> 0.996"
"b(15, 20, 0.4) -> 1"

```

Figure 4: Cumulative Binomial Probability Generated

2.4 Probability with Different Sample Size

The probabilities of accepting the lot were computed with the same proportions of defective components with the sample size changed to 15 and its characteristic curve was plotted as Figure 7

```

# generate binomial probabilities with b(2, 15, p)
for (i in 1:length(proportions)) {
  proportions[i] = signif(pbinom(2, size=15, prob=probs[i]), digits=4)
}
bnomDF <- data.frame(p=probs, px=proportions)

```

2.5 Outcomes

The second characteristic curve (Part C, Figure 6) seems to be the most satisfactory because of its near linear relationship between the proportion and their probability.

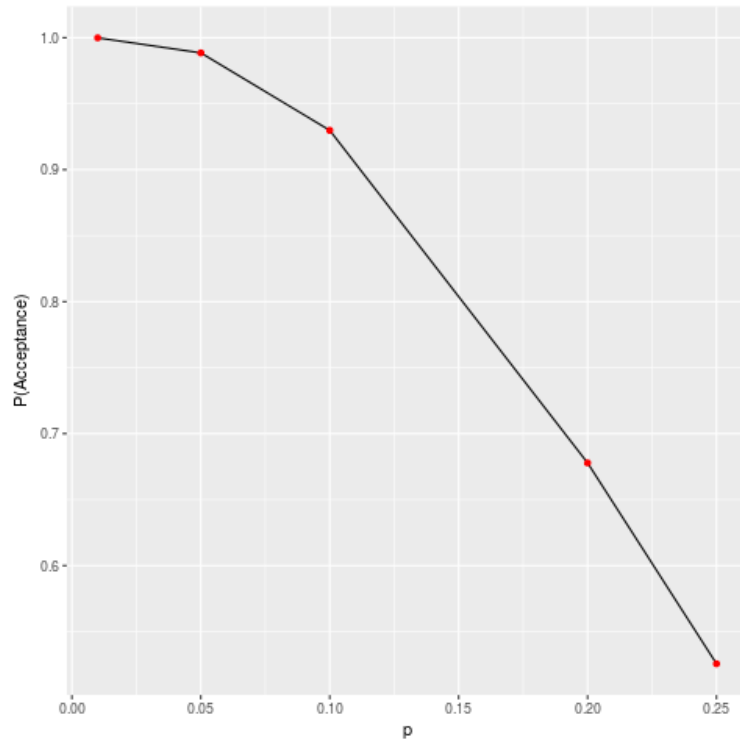


Figure 5: Relative Frequencies of the Generated Data (Part A)

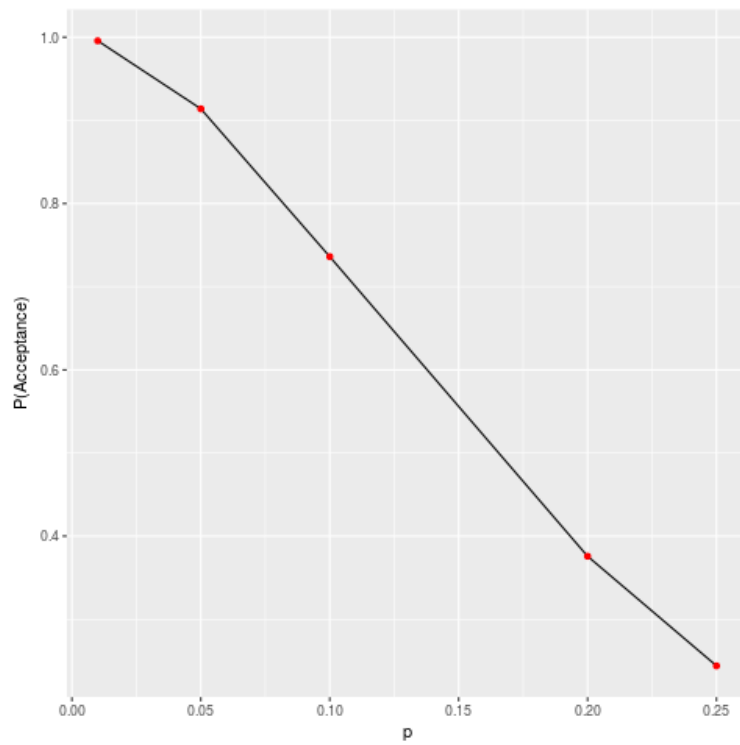


Figure 6: Relative Frequencies of the Generated Data (Part C)

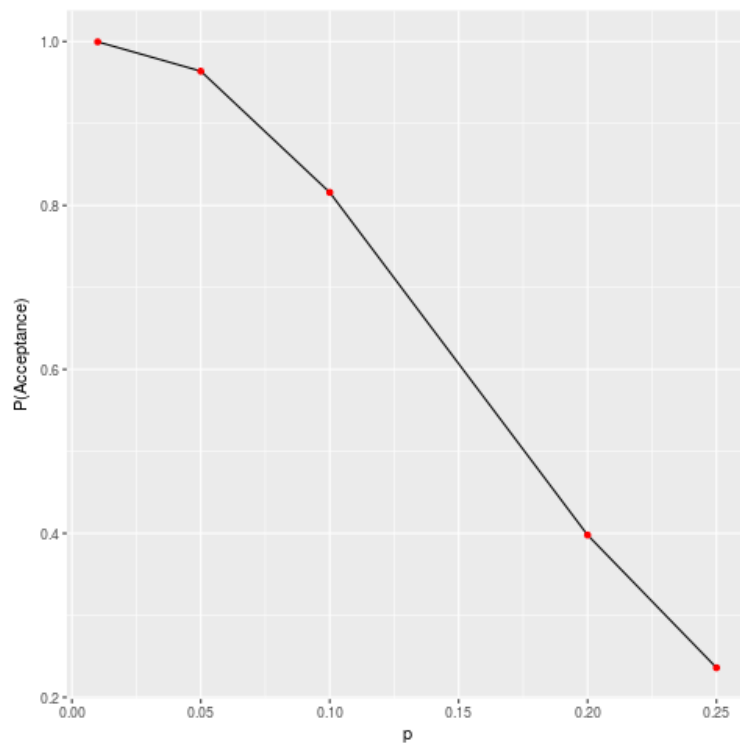


Figure 7: Relative Frequencies of the Generated Data (Part D)

```

# main.R
# This file contains the implementation of the functions in the Project 1
# 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

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

# initialize parameters for binomial distribution
x <- 20
N <- 1000
p <- 0.4
generatedData <- rbinom(N, x, p)

# ----- a -----

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

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

dev.off()

# ----- b -----

# print out the mean, variance and standard deviation
paste("Mean:", signif(mean(generatedData), digits=4),
      "| Variance:", signif(var(generatedData), digits=4),
      "| Standard Deviation:", signif(sqrt(var(generatedData)), digits=4)
)

# ----- c -----

# print out the summary statistics
summary(generatedData)

# ----- d -----

# generate a frequency table of the data and divide by the sample size
relfreq <- ftable(generatedData)/N

# convert the table to a dataframe for convenience
relfreq <- as.data.frame(relfreq)

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

# plot a bar plot of the relative frequencies
ggplot(data=relfreq, aes(x=generatedData, y=Freq)) +
  geom_bar(color="black", fill="white", stat="identity") +
  labs(y="Relative Frequency")

dev.off()

# ----- e -----

# print out the cumulative probability to compare with the binomial
# distribution table
cumProb <- 0 # initialize binomial cumulative probability

# iterate through the entire dataframe
for (i in 1:(nrow(relfreq))) {
  cumProb=cumProb+relfreq[i, "Freq"]
}

```



```

    print(
      paste0(
        "b(", relfreq[i, "generatedData"], ', ', x, ', ', p, ") -> ",
        cumProb
      )
    )
  }
}

# ----- Part 2: 3.58 -----

# initialize variables
probs <- c(0.01, 0.05, 0.10, 0.20, 0.25)
proportions <- rep(0, times=length(probs))

# ----- a -----

# generate binomial probabilities with b(2, 10, p)
for (i in 1:length(proportions)) {
  proportions[i] = signif(pbinom(2, size=10, prob=probs[i]), digits=4)
}
bnomDF <- data.frame(p=probs, px=proportions)
print(bnomDF)

# ----- b -----

png(filename="figures/charcurve1.png")

# plot the characteristic curve
ggplot(bnomDF, aes(x=p, y=px)) + geom_line() + geom_point(color="red") +
  labs(x="p", y="P(Acceptance)")

dev.off()

# ----- c -----

# generate binomial probabilities with b(1, 10, p)
for (i in 1:length(proportions)) {
  proportions[i] = signif(pbinom(1, size=10, prob=probs[i]), digits=4)
}
bnomDF <- data.frame(p=probs, px=proportions)
print(bnomDF)

png(filename="figures/charcurve2.png")

# plot the characteristic curve
ggplot(bnomDF, aes(x=p, y=px)) + geom_line() + geom_point(color="red") +
  labs(x="p", y="P(Acceptance)")

dev.off()

# ----- d -----

# generate binomial probabilities with b(2, 15, p)
for (i in 1:length(proportions)) {
  proportions[i] = signif(pbinom(2, size=15, prob=probs[i]), digits=4)
}
bnomDF <- data.frame(p=probs, px=proportions)
print(bnomDF)

png(filename="figures/charcurve3.png")

# plot the characteristic curve
ggplot(bnomDF, aes(x=p, y=px)) + geom_line() + geom_point(color="red") +
  labs(x="p", y="P(Acceptance)")

dev.off()

```