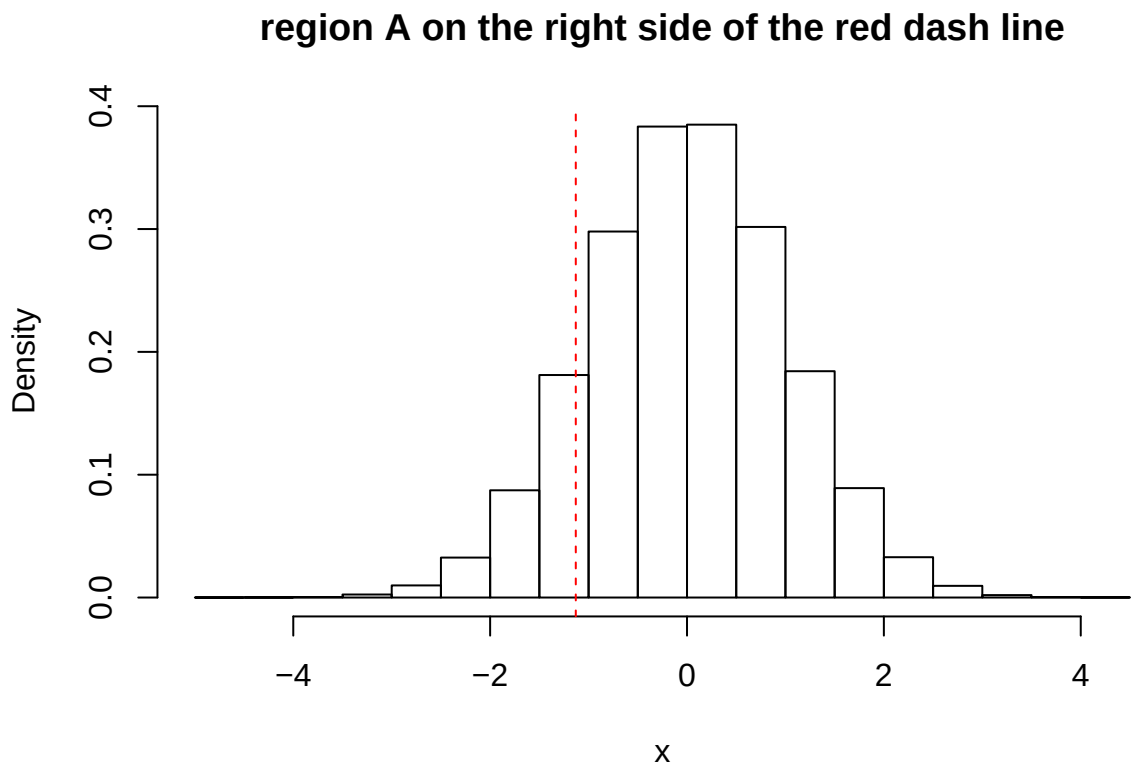
homework_ch3_JimmyNg

Jimmy Ng

October 2, 2018

Question 3.2.a

```
# what percent of a standard normal distribution N(mu = 0, sigma = 1) is found in each region?
# (a) Z > -1.13
mu = 0
sigma = 1
x <- rnorm(100000, mu, sigma)
hist(x, freq = F, main = "region A on the right side of the red dash line")
abline(v = -1.13, col = "red", lty = 2)
```



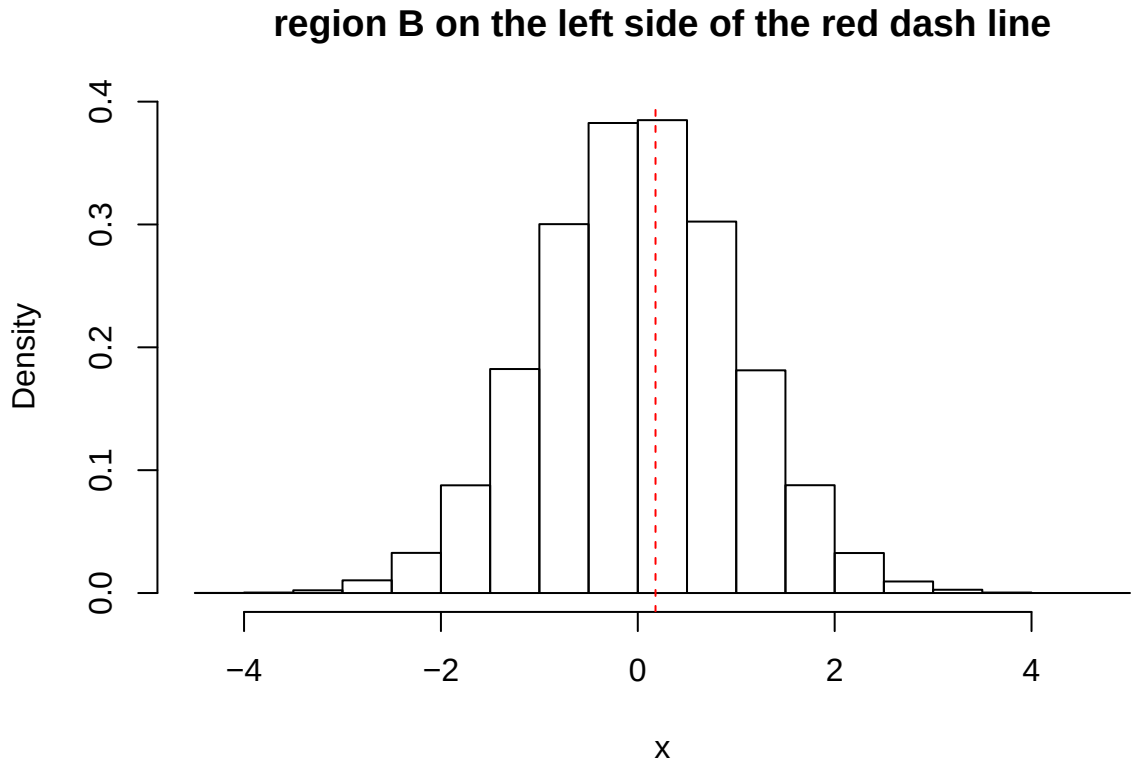

```
print( paste(
        round( (1 - pnorm(q = -1.13, mean = mu, sd = sigma)) * 100, 1 ),
        "% is found in this region"
) )
```

```
## [1] "87.1 % is found in this region"
```

```
# [1] "87.1 % is found in this region"
```

Question 3.2.b

```
# what percent of a standard normal distribution N(mu = 0, sigma = 1) is found in each region?
# (b) Z < 0.18
mu = 0
sigma = 1
x <- rnorm(100000, mu, sigma)
hist(x, freq = F, main = "region B on the left side of the red dash line")
abline(v = 0.18, col = "red", lty = 2)
```




```
print( paste(
        round( (pnorm(q = 0.18, mean = mu, sd = sigma)) * 100, 1 ),
        "% is found in this region"
) )
```

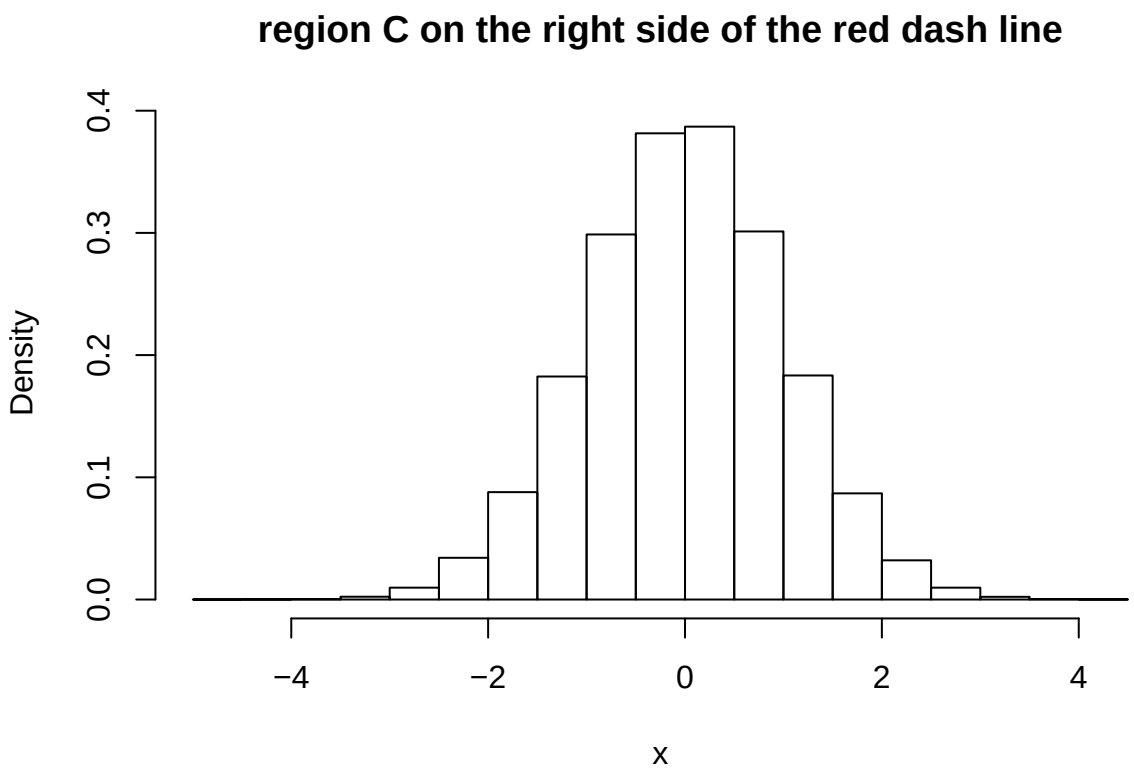
```
## [1] "57.1 % is found in this region"
```

```
# [1] "57.1 % is found in this region"
```

Question 3.2.c

```
# what percent of a standard normal distribution N(mu = 0, sigma = 1) is found in each region?
# (c) Z >8
mu = 0
sigma = 1
```

```r
x <- rnorm(100000, mu, sigma)
hist(x, freq = F, main = "region C on the right side of the red dash line")
abline(v = 8, col = "red", lty = 2)
```



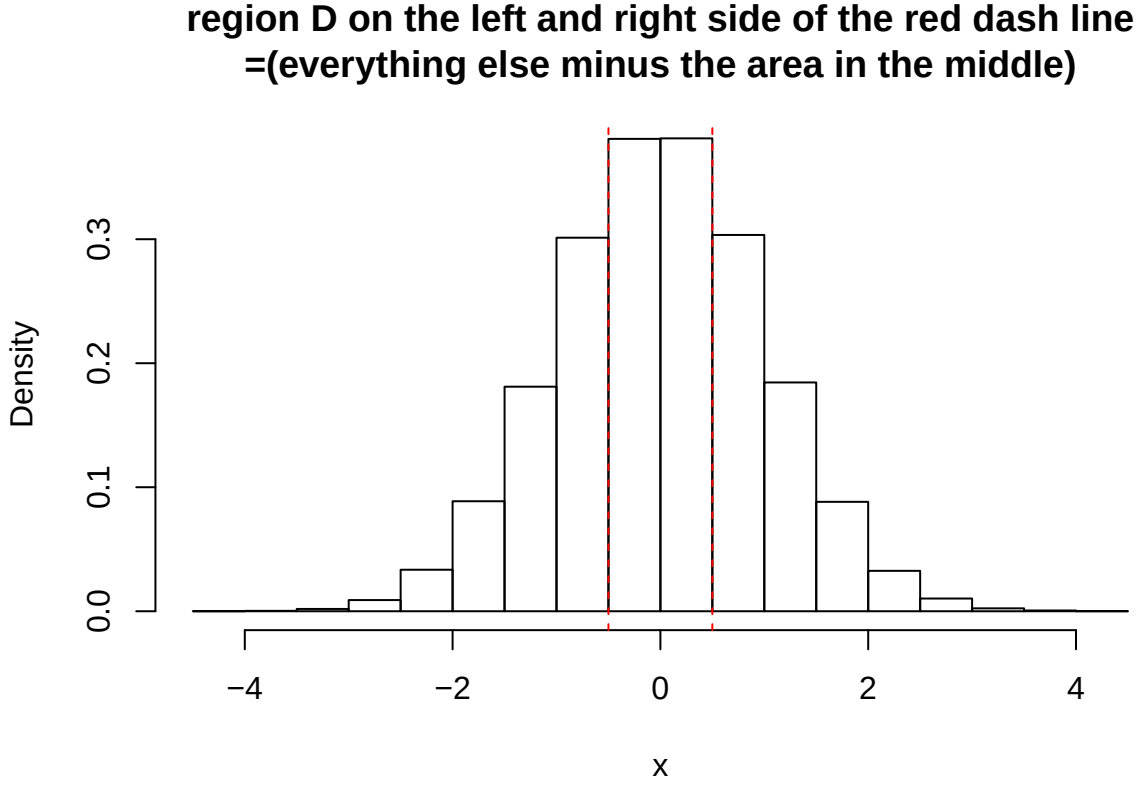

```r
print( paste(
        round( (1 - pnorm(q = 8, mean = mu, sd = sigma)) * 100, 15 ),
        "% is found in this region"
) )
```

```
## [1] "6.7e-14 % is found in this region"
```

```r
# [1] "6.7e-14 % is found in this region"
# virtually close to 0% and the red dash line cannot even be seen as it is too far on the right
# (outside the margin of the chart)
```

Question 3.2.d

```r
# what percent of a standard normal distribution N(mu = 0, sigma = 1) is found in each region?
# (d) |Z| >0.5
mu = 0
sigma = 1
x <- rnorm(100000, mu, sigma)
hist(x, freq = F, main = "region D on the left and right side of the red dash line\n=(everything else m
abline(v = 0.5, col = "red", lty = 2)
abline(v = -0.5, col = "red", lty = 2)
```




```
print( paste(
        round( (1 - (pnorm(q = 0.5, mean = mu, sd = sigma) - pnorm(q = -0.5, mean = mu, sd = sigma))) *
        "% is found in this region"
) )
```

```
## [1] "61.7 % is found in this region"
```

```
# [1] "61.7 % is found in this region"
```

Question 3.4

(a) men: N(mu = 4313, sigma = 583); women: N(mu = 5261, sigma = 807)

```
leo <- 4948
mary <- 5513

leo.z <- (4313 - 4948) / 583
# [1] -1.089194
mary.z <- (5261 - 5513) / 807
# [1] -0.3122677
```

(b) Leo has a z-score of -1.089 whereas Mary fares better with -0.312. Both have poorer performance in their respective group/below mean; however, Mary does better than Leo with a higher z-score (-0.312 > -1.089).

```
print(paste("Mary ranked ",
            round(100 * (pnorm(mary.z, 0, 1)), 1),
```

```
            " in her group, whereas Leo ranked ",
            round(100 * (pnorm(leo.z, 0, 1)), 1),
            ".", sep = ""))
```

```
## [1] "Mary ranked 37.7 in her group, whereas Leo ranked 13.8."
```

(c) Mary does better in her group.
(d) round(pnorm(leo.z, 0, 1), 2), i.e. Leo is approximately faster than 14% of his group.
(e) round(pnorm(mary.z, 0, 1), 2), i.e. Mary is approximately faster than 38% of her group.
(f) Yes I would have changed my answer if the distribution is not near normal. It's because the distribution calculated based on the z-score is not applicable for non-normal distribution. I would have transformed the data (such as square-root or log-transformation) before fitting a normal distribution to the dataset.

Question 3.18

```
female_height <- c(54, 55, 56, 56, 57, 58, 58, 59, 60, 60, 60, 61, 62, 62, 63, 63, 63, 64, 65, 66, 67, 
qqnorm(female_height, main = "Normal Q-Q plot of Female College Students")
qqline(female_height)

# alternatively, we can use the qqnormsim() function from the DATA606 package
library(DATA606)
```

```
## Loading required package: shiny
```

```
## Loading required package: openintro
```

```
## Please visit openintro.org for free statistics materials
```

```
##
## Attaching package: 'openintro'
```

```
## The following objects are masked from 'package:datasets':
##
##     cars, trees
```

```
## Loading required package: OIdata
```

```
## Loading required package: RCurl
```

```
## Loading required package: bitops
```

```
## Loading required package: maps
```

```
## Loading required package: ggplot2
```

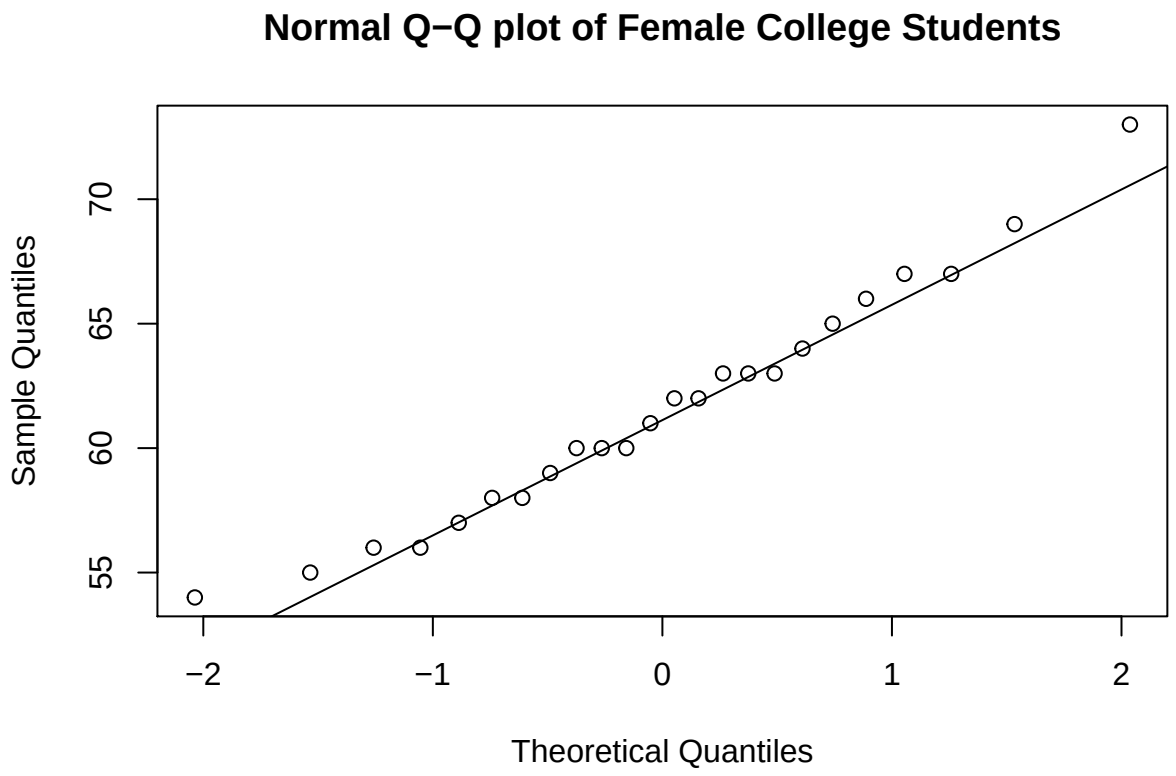
```
##
## Attaching package: 'ggplot2'
```

```
## The following object is masked from 'package:openintro':
##
##     diamonds
```

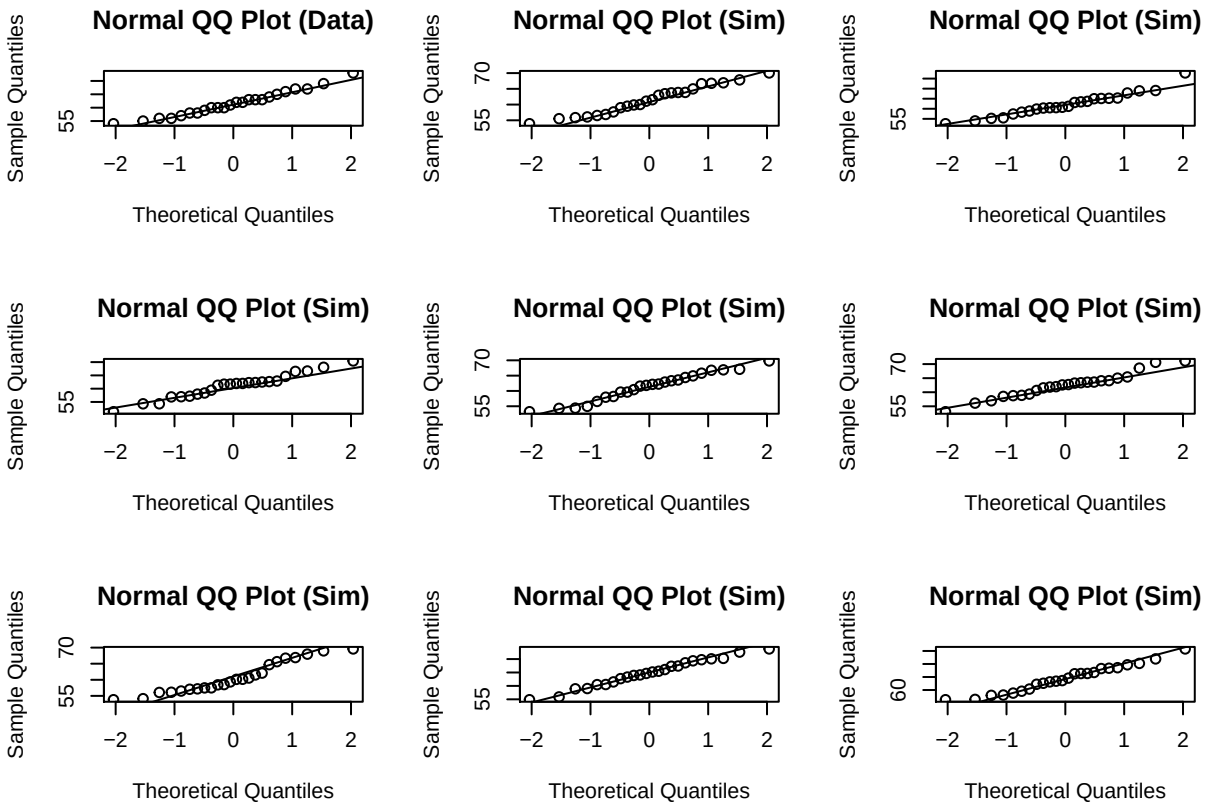
```
## Loading required package: markdown
```

```
##
## Welcome to CUNY DATA606 Statistics and Probability for Data Analytics
## This package is designed to support this course. The text book used
## is OpenIntro Statistics, 3rd Edition. You can read this by typing
## vignette('os3') or visit www.OpenIntro.org.
##
```

```
## The getLabs() function will return a list of the labs available.
##
## The demo(package='DATA606') will list the demos that are available.
##
## Attaching package: 'DATA606'
## The following object is masked from 'package:utils':
##
##     demo
```




```
qqnormsim(female_height)
```



(a) Yes, the qqplot shown above does support a normal distribution of the female college heights, which determines the 68-95-99.7% rule.

(b) From the histogram on the left and qqplot on the right, the data follows a normal distribution. The qqplot is using theoretical quantile (normal dist) plotting against real data and it displays a straight line as a strong indication of normal distribution.

Question 3.22

```
# (a) geometric distribution
defect = 0.02
print(paste("(a) The probability that exactly the 10th transistor is the first to defect is approximatel
            round(100 * dgeom(x = 10, prob = defect), 1),
            "%.",
            sep = ""))
```

```
## [1] "(a) The probability that exactly the 10th transistor is the first to defect is approximately 1.
```

```
# (b) binomial distribution
print(paste("(b) The probability that the machine produces no defective transistor in a batch of 100 is
            round(100 * pbinom(q = 0, size = 100, prob = defect), 1),
            "%.",
            sep = ""))
```

```
## [1] "(b) The probability that the machine produces no defective transistor in a batch of 100 is appr
```

```r
# the question can also be solved using poisson distribution, i.e. ppois(q = 0, lambda = 2) and that wo

# (c) mean and sd of geometric distribution
x = 1 / defect
sd = sqrt( ((1 - defect) / defect^2) )
print(paste("(c) The expected number of transistor to be produced before the first with a defect is ",
            x,
            " whereas the standard deviation is ",
            round(sd, 2),
            sep = ""))
```

```
## [1] "(c) The expected number of transistor to be produced before the first with a defect is 50 whera
```

```r
# (d) mean and sd of geometric distribution
new.defect = 0.05
new.x = 1 / new.defect
new.sd = sqrt( ((1 - new.defect) / new.defect^2) )
print(paste("(d) The expected number of transistor to be produced for this another machine before the fi
            new.x,
            " whereas the standard deviation is ",
            round(new.sd, 2),
            sep = ""))
```

```
## [1] "(d) The expected number of transistor to be produced for this another machine before the first w
```

(a) The probability that exactly the 10th transistor is the first to defect is approximately 1.6%.
(b) The probability that the machine produces no defective transistor in a batch of 100 is approximately 13.3%.
(c) The expected number of transistor to be produced before the first with a defect is 50 whereas the standard deviation is 49.5
(d) The expected number of transistor to be produced for this another machine before the first with a defect is 20 whereas the standard deviation is 19.49
(e) Increasing the probabily will decrease the mean and sd for geometric distribution. In this machine-manufacturing case, that makes absolute sense. If the probability of producing defect increases (from 2% to 5%), the expected number of transistor to be produced until seeing a first defect certainly would decrease. Or, put it this way, if the probability of success increases, the wait time until success would certainly decrease. You would expect to see a success (or transistor defect in this case) sooner with a higher probability (of success/defect).

Question 3.38

```r
# binomial distribution
# 3.38.a
m = 0.51
print(paste("(a) The probability that two of them will be boys is approximately ",
            round( 100 * dbinom(x = 2, size = 3, prob = 0.51), 1 ),
            "%.",
            sep = ""))
```

```
## [1] "(a) The probability that two of them will be boys is approximately 38.2%."
```

```r
# 3.38.b
# possible combo for two boys out of three children
# nCk, essentially this is the binomial coefficient
```

```r
combo <- choose(3, 2)  # [boy girl boy]  [girl boy boy]  [boy boy girl]

# probability for each combo
p <- .49 * .51 * .51

# addition rule for disjoint outcomes
p.combo <- round(p + p + p, 2)  # or, round(p * combo, 2)

print(paste("(b) It is ",
            p.combo == round(dbinom(x = 2, size = 3, prob = 0.51), 2),
            " that the answers from part(a) match with part(b).",
            sep = ""))
```

```
## [1] "(b) It is TRUE that the answers from part(a) match with part(b)."
```

(a) The probability that two of them will be boys is approximately 38.2%.
(b) It is TRUE that the answers from part(a) match with part(b).
(c) It is more tedious because we need to first figure out the number of combinations (binomial coefficient), and then the probability of each combo, and finally add each of them together. It would be a lengthy process and more tedious than approach (a), where we can simply do this <- dbinom(x = 3, size = 8, prob = 0.51).

Question 3.42

```r
# negative binomial distribution
p = 0.15
print(paste("(a) The probability that on the 10th try she will make her 3rd successful serve is approxi
            round( 100 * dnbinom(x = 7, size = 3, prob = p), 1 ),
            "%.",
            sep = ""))
```

```
## [1] "(a) The probability that on the 10th try she will make her 3rd successful serve is approximatel
```

(a) The probability that on the 10th try she will make her 3rd successful serve is approximately 3.9%.
(b) 15%. Each serve is independent.
(c) Part a applied the negative binomal distribution to look at the exact chance of getting the 10th try as her 3rd successful serve, whereas part b is asking the probability of getting the next (which is the 10th) serve successful. Each serve is independent and therefore the success rate for the 10th serve is the same/consistent as other trials.