Assignment 6

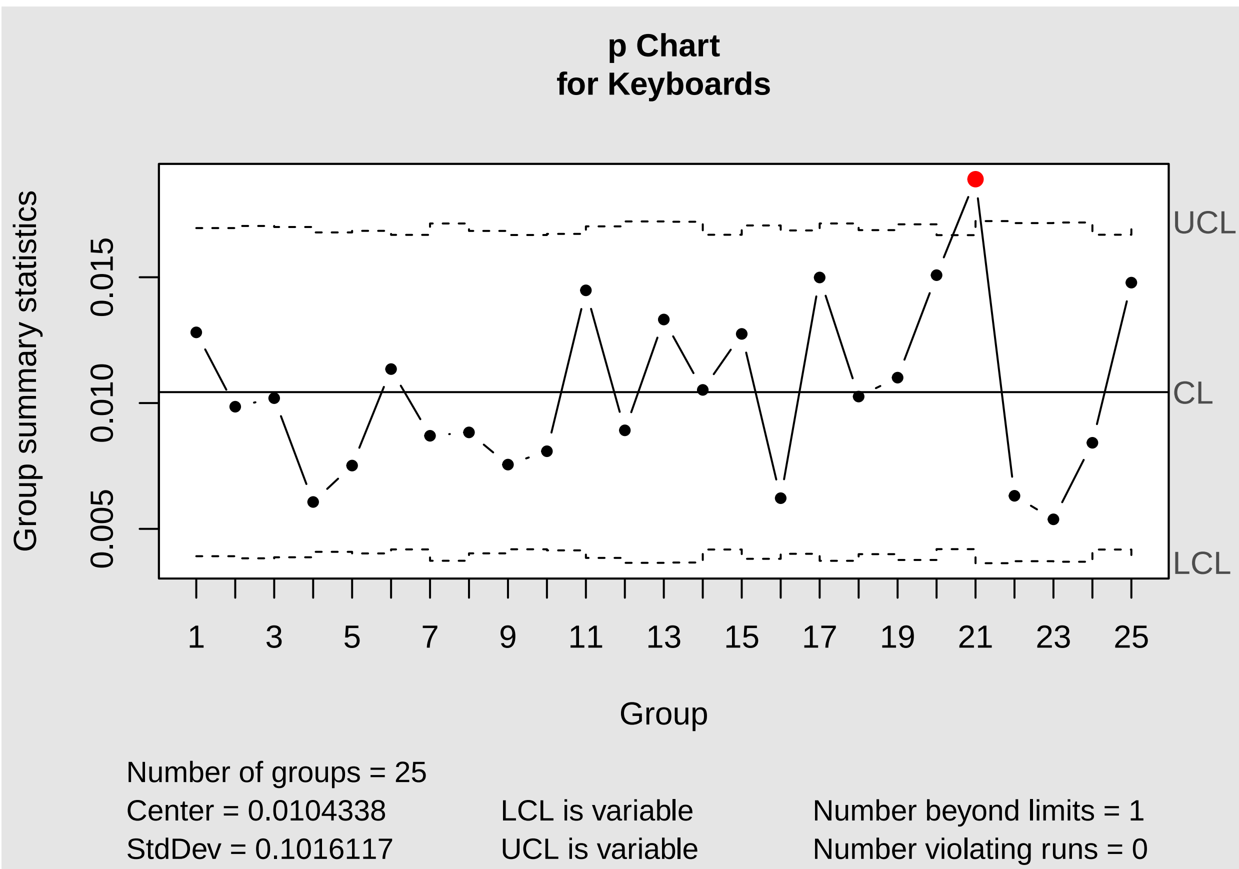
Evan Hodges

2025-04-24

1. After assembly and wiring of the individual keys, computer keyboards are tested by an automated test station that pushes each key several times. Daily records are kept of the number of keyboards inspected and the number that fail inspection. Data from 25 successive manufacturing days is in the file hw6q1.csv.

   a. Calculate the centerline and the control limits for a p chart for this data.

```
keys <- read.csv("hw6q1.csv", header = TRUE)
p_chart <- qcc(keys$Failed, keys$Tested, type = "p", data.name = "Keyboards")
```




```
p_center <- p_chart$center
p_lcl   <- p_chart$limits[ ,1]
p_ucl   <- p_chart$limits[ ,2]
p_center; range(p_lcl); range(p_ucl)
```

```
## [1] 0.0104338

## [1] 0.003636147 0.004194457

## [1] 0.01667314 0.01723145
```

```
# Overall defect proportion = 0.01043. LCL range = 0.00364 - 0.00419 and UCL = 0.01667 - 0.01723.
```

   b. Are there any signs of out-of-control conditions in this data?

```
p_chart$violations
```

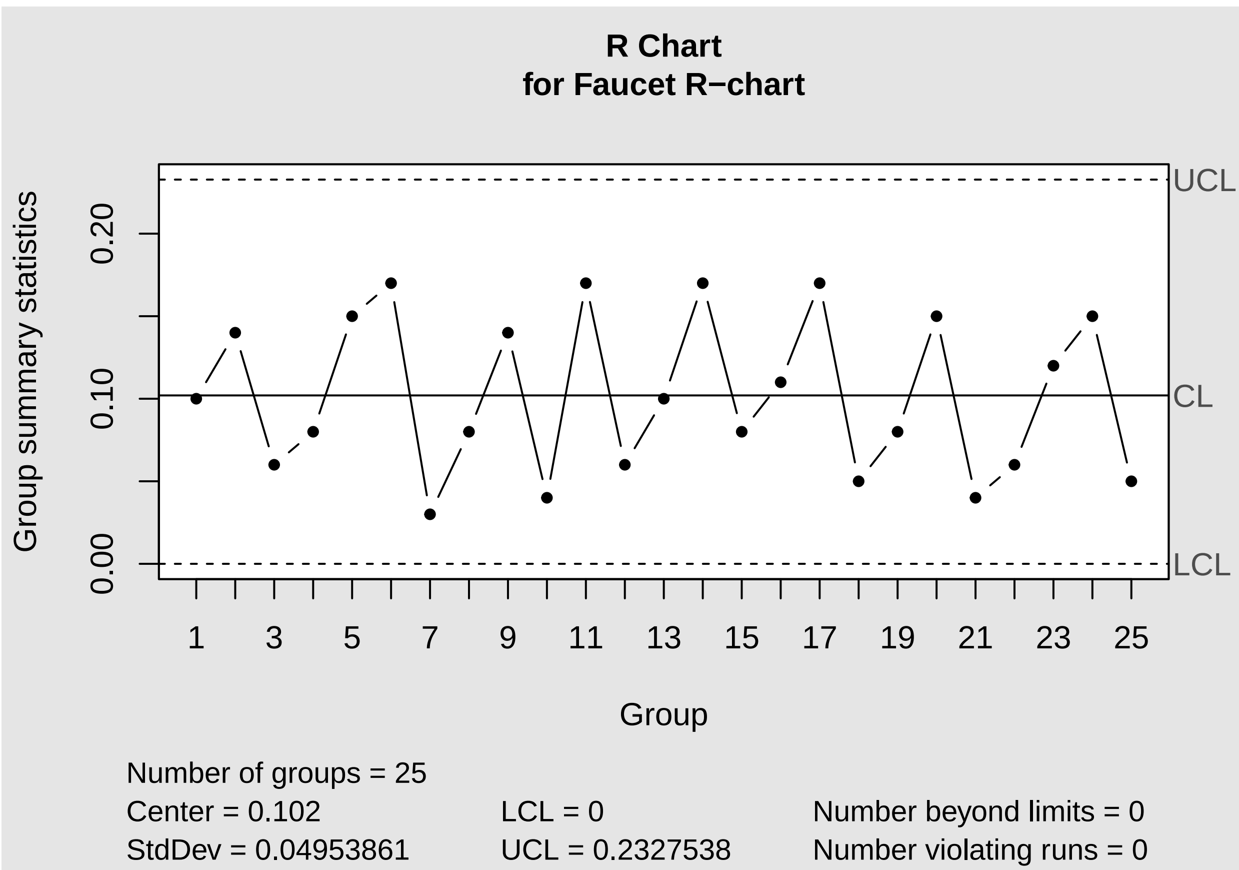
```
## $beyond.limits
## [1] 21
##
## $violating.runs
## numeric(0)
```

```
# Yes, there is one point beyond the upper control limit (day 21).
```

2. When installing a bath faucet, it is important to properly fasten the threaded end of the faucet stem to the water-supply line. The threaded stem dimensions must meet product specifications, otherwise malfunction and leakage may occur. Researchers investigated the production process of a particular bath faucet manufactured in India. They reported the threaded stem diameter (target value being 13mm) of each faucet in 25 samples of size 4 as given in the dataset hw6q2.csv.

   a. Construct an R chart for this data. Are there any out-of-control signals present?

```
faucet <- read.csv("hw6q2.csv", header = TRUE)
measurements <- faucet[, 2:5]
RChart <- qcc(measurements, type = "R", data.name = "Faucet R-chart")
```




```
RChart
```

```
## List of 11
##  $ call      : language qcc(data = measurements, type = "R", data.name = "Faucet R-chart")
##  $ type      : chr "R"
##  $ data.name : chr "Faucet R-chart"
##  $ data      : num [1:25, 1:4] 13 13 13 13 13 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:25] 0.1 0.14 0.06 0.08 0.15 ...
##   ..- attr(*, "names")= chr [1:25] "1" "2" "3" "4" ...
##  $ sizes     : int [1:25] 4 4 4 4 4 4 4 4 4 4 ...
##  $ center    : num 0.102
```

```
##  $ std.dev   : num 0.0495
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 0 0.233
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

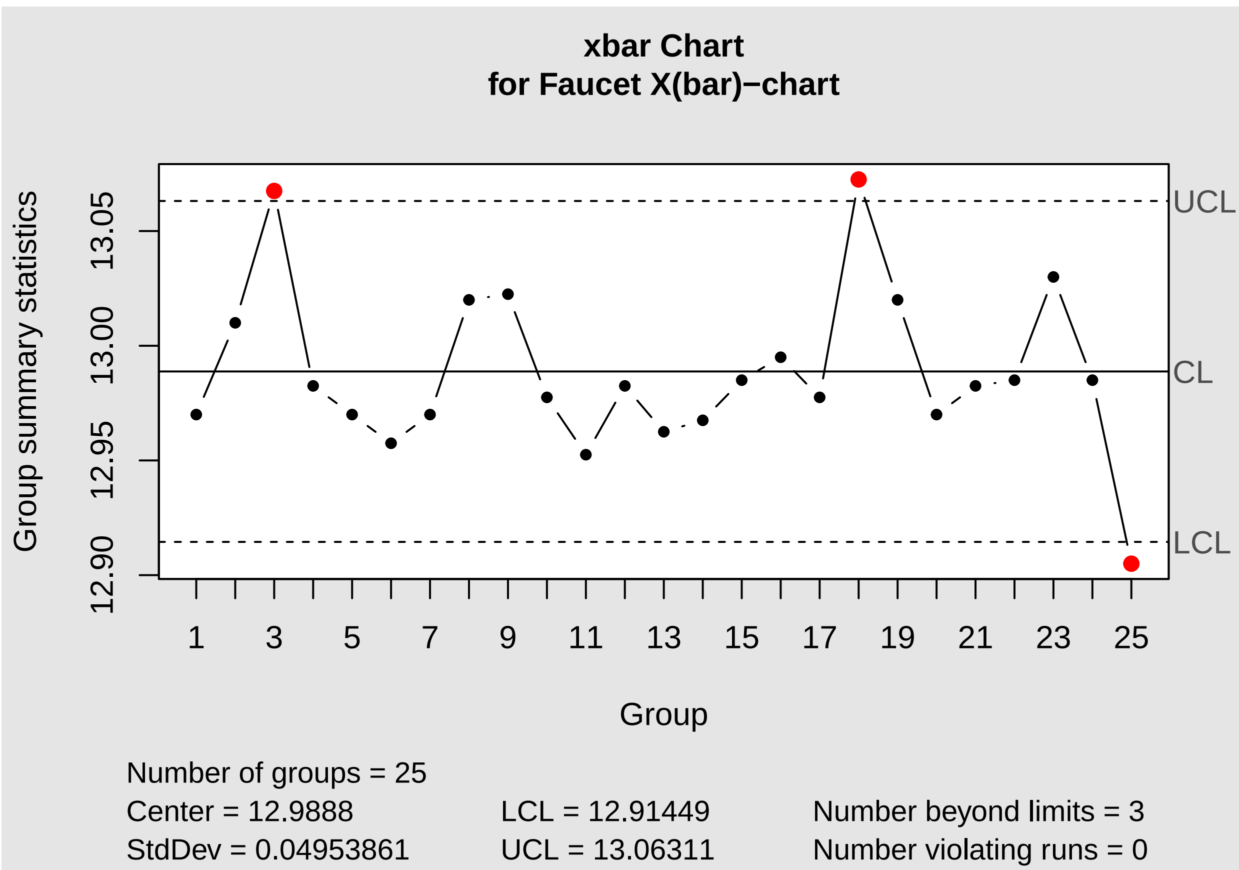
```
RChart$violations
```

```
## $beyond.limits
## integer(0)
##
## $violating.runs
## numeric(0)
```

```
# No, there are no out-of-control signals present.
```

b. Construct an X(bar) chart for this data. Are there any out-of-control signals present?

```
XbarChart <- qcc(measurements, type = "xbar", data.name = "Faucet X(bar)-chart")
```




```
XbarChart
```

```
## List of 11
##  $ call      : language qcc(data = measurements, type = "xbar", data.name = "Faucet X(bar)-chart")
##  $ type      : chr "xbar"
##  $ data.name : chr "Faucet X(bar)-chart"
##  $ data      : num [1:25, 1:4] 13 13 13 13 13 ...
##   ..- attr(*, "dimnames")=List of 2
##  $ statistics: Named num [1:25] 13 13 13.1 13 13 ...
##   ..- attr(*, "names")= chr [1:25] "1" "2" "3" "4" ...
##  $ sizes     : int [1:25] 4 4 4 4 4 4 4 4 4 4 ...
##  $ center    : num 13
##  $ std.dev   : num 0.0495
```

```
##  $ nsigmas   : num 3
##  $ limits    : num [1, 1:2] 12.9 13.1
##   ..- attr(*, "dimnames")=List of 2
##  $ violations:List of 2
##  - attr(*, "class")= chr "qcc"
```

```
XbarChart$violations
```

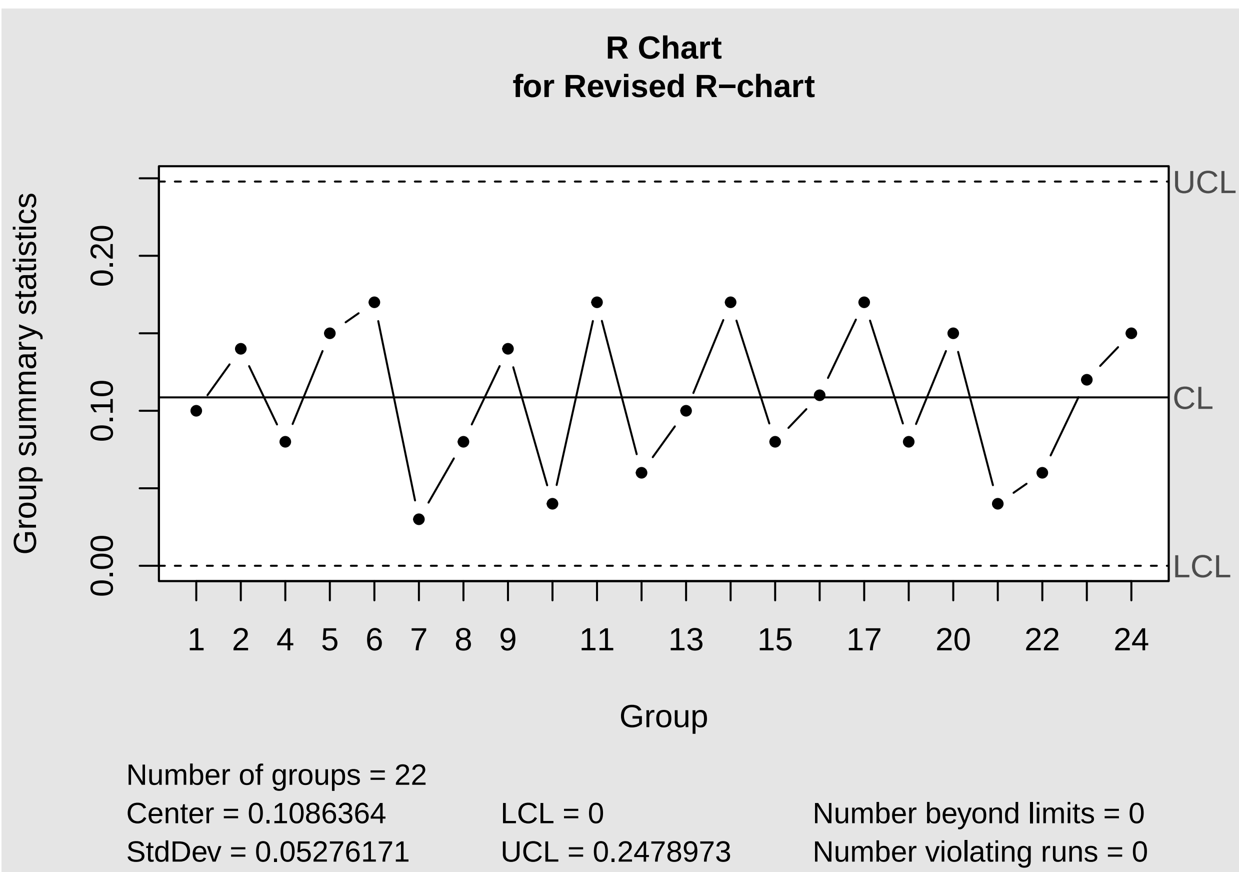
```
## $beyond.limits
## [1]  3 18 25
##
## $violating.runs
## numeric(0)
```

```
# Yes, subgroups 3, 18, and 25 fall outside the LCL/UCL limits.
```

c. If there are any out-of-control conditions in parts (a) or (b), reconstruct and interpret the revised X(bar) and R charts after eliminating these subgroups.

```
violating <- sort(unique(c(unlist(RChart$violations), unlist(XbarChart$violations))))
measurements_rev <- measurements
if(length(violating)>0) measurements_rev <- measurements[-violating, ]

RChart_rev <- qcc(measurements_rev, type = "R", data.name = "Revised R-chart")
```

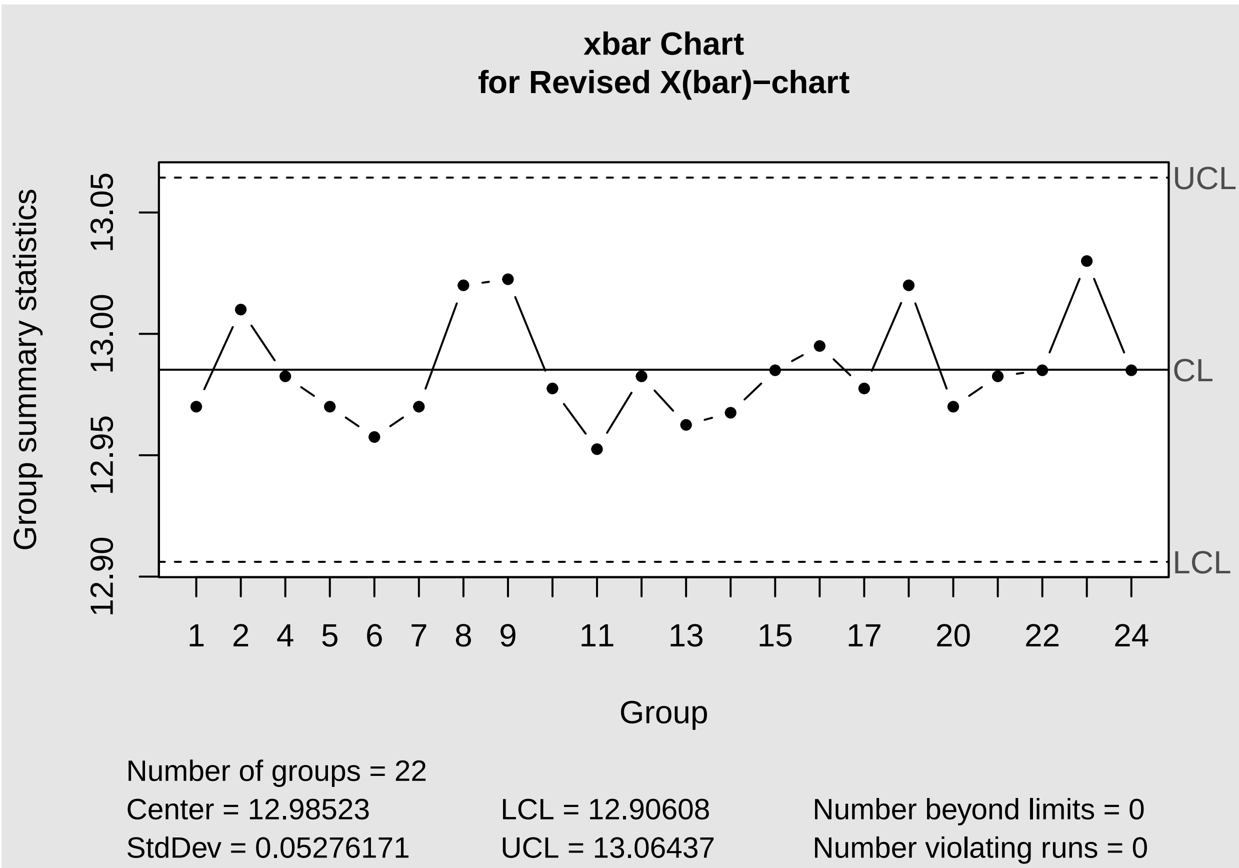



```
RChart_rev
```

```
## List of 11
##  $ call      : language qcc(data = measurements_rev, type = "R", data.name = "Revised R-chart")
##  $ type      : chr "R"
##  $ data.name : chr "Revised R-chart"
##  $ data      : num [1:22, 1:4] 13 13 13 13 12.9 ...
##   ..- attr(*, "dimnames")=List of 2
```

```
##   $ statistics: Named num [1:22] 0.1 0.14 0.08 0.15 0.17 ...
##    ..- attr(*, "names")= chr [1:22] "1" "2" "4" "5" ...
##   $ sizes     : Named int [1:22] 4 4 4 4 4 4 4 4 4 4 ...
##    ..- attr(*, "names")= chr [1:22] "1" "2" "4" "5" ...
##   $ center    : num 0.109
##   $ std.dev   : num 0.0528
##   $ nsigmas   : num 3
##   $ limits    : num [1, 1:2] 0 0.248
##    ..- attr(*, "dimnames")=List of 2
##   $ violations:List of 2
##   - attr(*, "class")= chr "qcc"
```

```
XbarChart_rev <- qcc(measurements_rev, type = "xbar", data.name = "Revised X(bar)-chart")
```




```
XbarChart_rev
```

```
## List of 11
##   $ call      : language qcc(data = measurements_rev, type = "xbar", data.name = "Revised X(bar)-chart")
##   $ type      : chr "xbar"
##   $ data.name : chr "Revised X(bar)-chart"
##   $ data      : num [1:22, 1:4] 13 13 13 13 12.9 ...
##    ..- attr(*, "dimnames")=List of 2
##   $ statistics: Named num [1:22] 13 13 13 13 13 ...
##    ..- attr(*, "names")= chr [1:22] "1" "2" "4" "5" ...
##   $ sizes     : Named int [1:22] 4 4 4 4 4 4 4 4 4 4 ...
##    ..- attr(*, "names")= chr [1:22] "1" "2" "4" "5" ...
##   $ center    : num 13
##   $ std.dev   : num 0.0528
##   $ nsigmas   : num 3
##   $ limits    : num [1, 1:2] 12.9 13.1
##    ..- attr(*, "dimnames")=List of 2
##   $ violations:List of 2
##   - attr(*, "class")= chr "qcc"
```

```
# After removing the out-of-control conditions, the revised R-chart shows no further violations.
```

3. Random samples of size n are taken from a normal population whose standard deviation is known to be 5.

   a. For random samples of size n=10, calculate the area under the sampling distribution curve for X(bar) between the values (population mean)-1 and (population mean)+1. That is, find the probability that the sample mean lies within ±1 unit of the population mean.

```
sigma <- 5
n1 <- 10
se1 <- sigma / sqrt(n1)
prob1 <- pnorm(1 / se1) - pnorm(-1 / se1)
prob1
```

```
## [1] 0.4729107
```

```
# For n=10, the probability that the sample mean lies within ±1 unit of the population mean = 0.47.
```

   b. Repeat the probability calculation in part (a) for samples of size n=50, n=100, and n=1000.

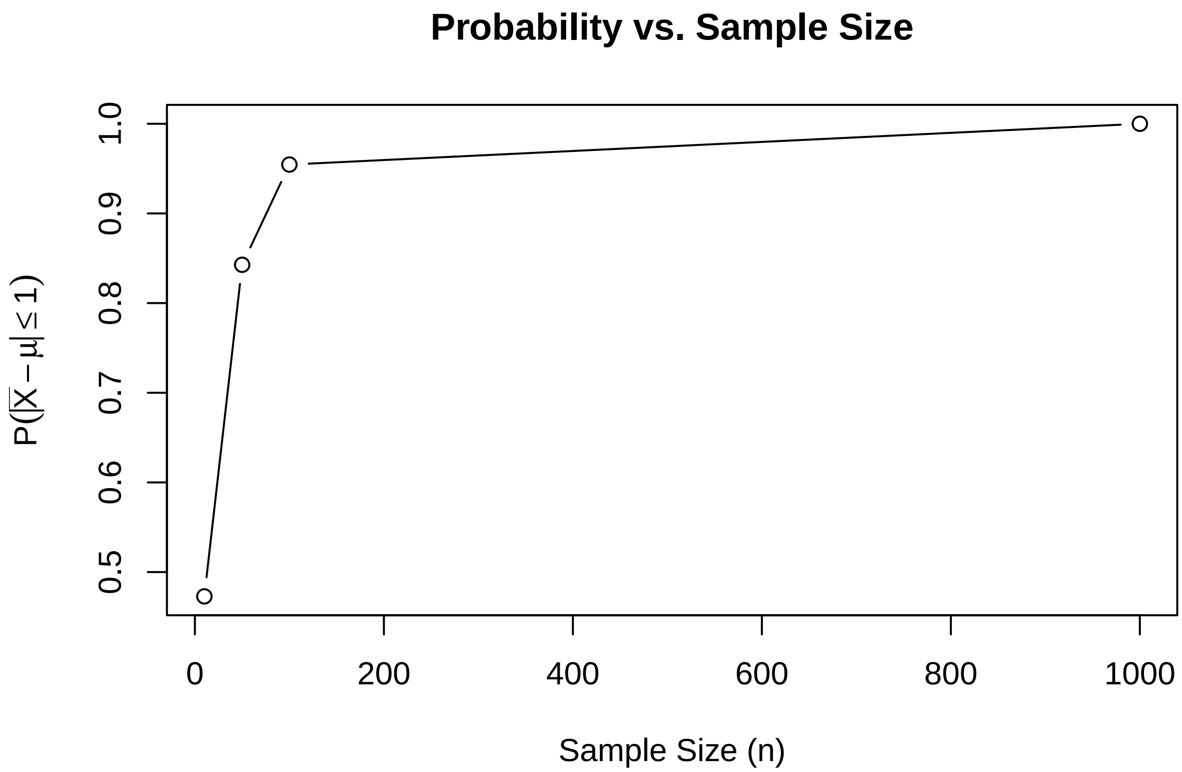
```
n_values <- c(10, 50, 100, 1000)
probs    <- sapply(n_values, function(n) pnorm(1/(sigma/sqrt(n))) - pnorm(-1/(sigma/sqrt(n))))
results  <- data.frame(n = n_values, Probability = probs)
results
```

```
##      n Probability
## 1   10   0.4729107
## 2   50   0.8427008
## 3  100   0.9544997
## 4 1000   1.0000000
```

```
# As n increases, the chance that the sample mean lies within ±1 unit of the population mean increases.
```

   c. Graph the probabilities you found in parts (a) and (b) versus their corresponding sample sizes, n. What can you conlude from this graph.

```
plot(results$n, results$Probability, type = "b",
 xlab = "Sample Size (n)",
 ylab = expression(P(abs(bar(X)- mu) <= 1)),
 main = "Probability vs. Sample Size")
```




```
# The sampling distribution narrows as n grows, so the probability approaches 1.
```