```
> #This script was used to do the problems on
> #   a particular worksheet7
> ##################
> ## C A S E:  1  ##
> ##################
> n_1 <- 112  # the sample size
> x_1 <- 71  # the number of 'good' items
> c_level <- 0.8950 # set the confidence level
>
> p <- x_1/n_1  # get the proportion
> p
[1] 0.6339286
> #       compute the standard deviation estimate
> stddev <- sqrt( p*(1-p)/n_1 )
> stddev
[1] 0.04551914
> #   using this as an approximately normal distribution
> #   we can use qnorm to get a z value
> z <- qnorm( (1-c_level)/2, lower.tail=FALSE)
> z
[1] 1.621082
> moe <- z*stddev   # compute the margin of error
> moe
[1] 0.07379027
> # make sure that the function is loaded
> source("../ci_prop.R")
> #    run the function
> ci_prop( n_1, x_1, c_level)
      lower      upper      p hat    z-score   p hat sd
0.56013830 0.70771884 0.63392857 1.62108225 0.04551914
>
> ##################
> ## C A S E:  2  ##
> ##################
> n_1 <- 107  # the sample size
> x_1 <-  44 # the number of 'good' items
> c_level <- 0.9950
>
> p <- x_1/n_1  # get the proportion
> p
[1] 0.411215
> #       compute the standard deviation estimate
> stddev <- sqrt( p*(1-p)/n_1 )
> stddev
[1] 0.04756866
> #   using this as an approximately normal distribution
> #   we can use qnorm to get a z value
> z <- qnorm( (1-c_level)/2, lower.tail=FALSE)
> z
[1] 2.807034
> moe <- z*stddev   # compute the margin of error
> moe
[1] 0.1335268
> #    run the function
> ci_prop( n_1, x_1, c_level)
      lower      upper      p hat    z-score   p hat sd
0.27768811 0.54474180 0.41121495 2.80703377 0.04756866
>
>
> ##################
```

```
> ## C A S E:  3  ##
> ##################
> n_1 <- 114  # the sample size
> x_1 <-  90 # the number of 'good' items
> c_level <- 0.9800
>
> p <- x_1/n_1  # get the proportion
> p
[1] 0.7894737
> #       compute the standard deviation estimate
> stddev <- sqrt( p*(1-p)/n_1 )
> stddev
[1] 0.03818296
> #   using this as an approximately normal distribution
> #   we can use qnorm to get a z value
> z <- qnorm( (1-c_level)/2, lower.tail=FALSE)
> z
[1] 2.326348
> moe <- z*stddev   # compute the margin of error
> moe
[1] 0.08882685
> #    run the function
> ci_prop( n_1, x_1, c_level)
      lower      upper      p hat    z-score   p hat sd
0.70064684 0.87830053 0.78947368 2.32634787 0.03818296
>
>
> ##################
> ## C A S E:  4  ##
> ##################
> #
> #             generate the values
> source("../gnrnd4.R")
> gnrnd4(373788907,4768986)
style= 7   size= 90    seed= 37378   num digits= 0   alt_sign= 1
8 9 8 6 7 4
1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 4 4 4 4 4 4 5 5 5 5 5 5 5 6
6 6 6
[1] "DONE "
> #             verify the values
> head( L1 )
[1] 2 3 2 6 1 2
> tail( L1 )
[1] 1 2 3 3 4 1
> n_1 <- length( L1 )  # get the number of values
> n_1
[1] 90
> hold_table <- table( L1 ) # how many of each
> hold_table
L1
 1  2  3  4  5  6
18 19 22 10  8 13
> pick <- 1     # set the item that is of interest
> #              now find how many of those items we have
> x_1 <- as.numeric( hold_table[ pick ])
> x_1
[1] 18
>
> c_level <- 0.9850  # set the confidence level
>
```

```
> p <- x_1/n_1  # get the proportion
> p
[1] 0.2
> stddev <- sqrt( p*(1-p)/n_1 ) # get the std. dev.
> stddev
[1] 0.0421637
> z <- qnorm( (1-c_level)/2, lower.tail=FALSE)
> z
[1] 2.432379
> moe <- z*stddev
> moe
[1] 0.1025581
> #    run the function
> ci_prop( n_1, x_1, c_level)
      lower      upper      p hat    z-score   p hat sd
0.09744189 0.30255811 0.20000000 2.43237906 0.04216370
>
>
> ##################
> ## C A S E:  5  ##
> ##################
> #
> #             generate the values
> source("../gnrnd4.R")
> gnrnd4(187659407,895695)
style= 7   size= 95    seed= 18765   num digits= 0   alt_sign= 1
9 6 5 9 8
1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 4 4 4 4 4 4 4 4 4 5 5 5 5 5 5 5 5
[1] "DONE "
> #             verify the values
> head( L1 )
[1] 1 2 5 5 3 5
> tail( L1 )
[1] 4 3 3 4 5 1
> n_1 <- length( L1 )  # get the number of values
> n_1
[1] 95
> hold_table <- table( L1 ) # how many of each
> hold_table
L1
 1  2  3  4  5
18 11 15 20 31
> pick <- 1     # set the item that is of interest
> #              now find how many of those items we have
> x_1 <- as.numeric( hold_table[ pick ])
> x_1
[1] 18
>
> c_level <- 0.9225  # set the confidence level
>
> p <- x_1/n_1  # get the proportion
> p
[1] 0.1894737
> stddev <- sqrt( p*(1-p)/n_1 ) # get the std. dev.
> stddev
[1] 0.04020649
> z <- qnorm( (1-c_level)/2, lower.tail=FALSE)
> z
[1] 1.76538
> moe <- z*stddev
```

```
> moe
[1] 0.07097972
> #    run the function
> ci_prop( n_1, x_1, c_level)
      lower      upper      p hat    z-score   p hat sd
0.11849396 0.26045341 0.18947368 1.76537954 0.04020649
>
```