

Worksheet9

STAT414

2024-11-20

Module09 deals with two important topics: obtaining prediction intervals for future observations from a process, and determining tolerance intervals that are expected to capture certain proportion of individuals from the population. These concepts are closely related to the concept of quantiles introduced earlier. Prediction intervals are used widely in many applications, especially in the area of statistical quality control. Tolerance intervals are used in biomedical fields, especially in determining acceptable ranges of biomarkers (such as level of sugar in blood samples, typical range of cholesterol levels in certain populations etc).

```
#1. Suppose a population is well approximated by a Normal distribution with  
#mu = 10 and sig = 2.  
#2. Suppose a population is well-modeled by a lognormal distribution with theta = 4  
#and tau = 1.  
  
mu = 10  
sig = 2  
theta = 4  
tau = 1  
  
#(a) Obtain the quantiles for this normal population for p = 0.025, 0.05, 0.95, 0.975  
  
p = c(0.025, 0.05, 0.95, 0.975)  
  
quantiles <- data.frame (  
  population = "N(mu=10, sig=2)",  
  quantiles = p,  
  values = qnorm(p, mean = mu, sd = sig)  
)  
  
#(a) Obtain the quantiles for this lognormal population for p = 0.025, 0.05, 0.95, 0.975  
  
cat("A percentile is a quantile corresponding to p=[0,1], with Xp corresponding to  
the .p quantiles. For any continuous distribution, the CDF F can be used to calculate  
a quantile:  $X_p = F^{-1}(p)$ ")  
  
## A percentile is a quantile corresponding to p=[0,1], with Xp corresponding to  
## the .p quantiles. For any continuous distribution, the CDF F can be used to calculate  
## a quantile:  $X_p = F^{-1}(p)$   
  
append <- data.frame (  
  population = "LN(theta=4, tau=1)",  
  quantiles = p,  
  values = qlnorm(p, meanlog = theta, sdlog = tau)
```

```
)

#combine both populations and the calculated quantiles into 1 df
quantiles <- rbind(quantiles, append)
quantiles

##           population quantiles      values
## 1    N(mu=10, sig=2)      0.025  6.080072
## 2    N(mu=10, sig=2)      0.050  6.710293
## 3    N(mu=10, sig=2)      0.950 13.289707
## 4    N(mu=10, sig=2)      0.975 13.919928
## 5 LN(theta=4, tau=1)      0.025  7.690886
## 6 LN(theta=4, tau=1)      0.050 10.539671
## 7 LN(theta=4, tau=1)      0.950 282.832154
## 8 LN(theta=4, tau=1)      0.975 387.596165

#(b) Determine a tolerance interval with 90% coverage. Determine a TI with a
#95% coverage.

cat("A tolerance interval for a population contains the pre-specified proportion
of the population. A Beta * 100% TI is called the coverage of the interval.
Create 2 samples using the normal and lognormal parameters, since there is
no actual dataset provided, and a vector x is a required input for the EnvStats
function to generate TI's tolIntNorm and tolIntLnorm. Choose a high sample size
since we are trying to calculate theoretical values.
Choose alpha=.05")

## A tolerance interval for a population contains the pre-specified proportion
## of the population. A Beta * 100% TI is called the coverage of the interval.
## Create 2 samples using the normal and lognormal parameters, since there is
## no actual dataset provided, and a vector x is a required input for the EnvStats
## function to generate TI's tolIntNorm and tolIntLnorm. Choose a high sample size
## since we are trying to calculate theoretical values.
## Choose alpha=.05

# norm_sample <- rnorm(10000, mean=mu, sd=sig)
# lnorm_sample <- rlnorm(10000, meanlog=theta, sdlog=tau)
#
# #Normal distribution 90% TI & 95% TI
# tolIntNorm(norm_sample, coverage = 0.90, ti.type = "two-sided", conf.level = 0.95,
#             method = "exact")
# tolIntNorm(norm_sample, coverage = 0.95, ti.type = "two-sided", conf.level = 0.95,
#             method = "exact")
#
# #Lognormal distribution 90% TI & 95% TI
# tolIntLnorm(lnorm_sample, coverage = 0.90, ti.type = "two-sided", conf.level = 0.95,
#             method = "exact")
# tolIntLnorm(lnorm_sample, coverage = 0.95, ti.type = "two-sided", conf.level = 0.95,
#             method = "exact")

cat("There is a simpler method to calculate tolerance intervals using quantiles,
by simply using quantiles containing the desired proportions. Using .5 +/- Beta/2
as the quantiles of the lower and upper bounds for the TI.")
```

```
## There is a simpler method to calculate tolerance intervals using quantiles,
##     by simply using quantiles containing the desired proportions. Using .5 +/- Beta/2
##     as the quantiles of the lower and upper bounds for the TI.
```

```
lower_90 <- .5 - (.9/2)
upper_90 <- .5 + (.9/2)
lower_95 <- .5 - (.95/2)
upper_95 <- .5 + (.95/2)

cat(sprintf("90%% Normal TI: (%.2f, %.2f)\n",
            qnorm(lower_90, mean = mu, sd = sig),
            qnorm(upper_90, mean = mu, sd = sig)))
```

```
## 90% Normal TI: (6.71, 13.29)
```

```
cat(sprintf("95%% Normal TI: (%.2f, %.2f)\n",
            qnorm(lower_95, mean = mu, sd = sig),
            qnorm(upper_95, mean = mu, sd = sig)))
```

```
## 95% Normal TI: (6.08, 13.92)
```

```
cat(sprintf("90%% Lognormal TI: (%.2f, %.2f)\n",
            qlnorm(lower_90, meanlog = theta, sdlog = tau),
            qlnorm(upper_90, meanlog = theta, sdlog = tau)))
```

```
## 90% Lognormal TI: (10.54, 282.83)
```

```
cat(sprintf("95%% Lognormal TI: (%.2f, %.2f)\n",
            qlnorm(lower_95, meanlog = theta, sdlog = tau),
            qlnorm(upper_95, meanlog = theta, sdlog = tau)))
```

```
## 95% Lognormal TI: (7.69, 387.60)
```

```
##(c) For each coverage level, provide several other intervals of the form
#[xdelta, xbeta+delta] with the same coverage by taking different values of delta.
```

```
##For any delta < 1-beta, [Xdelta, Xdelta+beta] is a B 100% TI
#delta + beta < 1
```

```
##for b=90, deltamin = .01, deltamax = .1
delta_90 <- seq(.01, .1, by=.01)
delta_90 <- delta_90[delta_90 + 0.90 < 1]
```

```
##for b=95, deltamin = .01, deltamax = .05
delta_95 <- seq(.01, .05, by=.01)
delta_95 <- delta_95[delta_95 + 0.95 < 1]
```

```
for(i in delta_90){
  cat(sprintf("Delta: %.2f | 90%% Normal TI: (%.2f, %.2f)\n",
            i,
```

```

        qnorm(i, mean = mu, sd = sig),
        qnorm(i + .90, mean = mu, sd = sig)))

cat(sprintf("Delta: %.2f | 90%% Lognormal TI: (%.2f, %.2f)\n",
    i,
    qlnorm(i, meanlog = theta, sdlog = tau),
    qlnorm(i + .90, meanlog = theta, sdlog = tau)))
}

```

```

## Delta: 0.01 | 90% Normal TI: (5.35, 12.68)
## Delta: 0.01 | 90% Lognormal TI: (5.33, 208.67)
## Delta: 0.02 | 90% Normal TI: (5.89, 12.81)
## Delta: 0.02 | 90% Lognormal TI: (7.00, 222.53)
## Delta: 0.03 | 90% Normal TI: (6.24, 12.95)
## Delta: 0.03 | 90% Lognormal TI: (8.32, 238.84)
## Delta: 0.04 | 90% Normal TI: (6.50, 13.11)
## Delta: 0.04 | 90% Lognormal TI: (9.48, 258.47)
## Delta: 0.05 | 90% Normal TI: (6.71, 13.29)
## Delta: 0.05 | 90% Lognormal TI: (10.54, 282.83)
## Delta: 0.06 | 90% Normal TI: (6.89, 13.50)
## Delta: 0.06 | 90% Lognormal TI: (11.53, 314.41)
## Delta: 0.07 | 90% Normal TI: (7.05, 13.76)
## Delta: 0.07 | 90% Lognormal TI: (12.48, 358.09)
## Delta: 0.08 | 90% Normal TI: (7.19, 14.11)
## Delta: 0.08 | 90% Lognormal TI: (13.40, 425.71)
## Delta: 0.09 | 90% Normal TI: (7.32, 14.65)
## Delta: 0.09 | 90% Lognormal TI: (14.29, 559.11)

```

```

for(i in delta_95){
  cat(sprintf("Delta: %.2f | 95%% Normal TI: (%.2f, %.2f)\n",
    i,
    qnorm(i, mean = mu, sd = sig),
    qnorm(i + .95, mean = mu, sd = sig)))

  cat(sprintf("Delta: %.2f | 95%% Lognormal TI: (%.2f, %.2f)\n",
    i,
    qlnorm(i, meanlog = theta, sdlog = tau),
    qlnorm(i + .95, meanlog = theta, sdlog = tau)))
}

```

```

## Delta: 0.01 | 95% Normal TI: (5.35, 13.50)
## Delta: 0.01 | 95% Lognormal TI: (5.33, 314.41)
## Delta: 0.02 | 95% Normal TI: (5.89, 13.76)
## Delta: 0.02 | 95% Lognormal TI: (7.00, 358.09)
## Delta: 0.03 | 95% Normal TI: (6.24, 14.11)
## Delta: 0.03 | 95% Lognormal TI: (8.32, 425.71)
## Delta: 0.04 | 95% Normal TI: (6.50, 14.65)
## Delta: 0.04 | 95% Lognormal TI: (9.48, 559.11)

```

#(d) Plot the length of these intervals vs delta. Comment.

#create dataframes to store the lengths of the TIs for different deltas

```

delta_90_lengths <- data.frame(delta = delta_90)
delta_95_lengths <- data.frame(delta = delta_95)

#For each coverage value, calculate the LB and UB for normal and lognormal
#populations, for each value of delta for the corresponding tolerance levels,
#and store in the corresponding dataframes
delta_90_lengths$normal_lower <- qnorm(delta_90, mean = mu, sd = sig)
delta_90_lengths$normal_upper <- qnorm(delta_90 + .90, mean = mu, sd = sig)

delta_90_lengths$lnormal_lower <- qlnorm(delta_90, meanlog = theta, sdlog = tau)
delta_90_lengths$lnormal_upper <- qlnorm(delta_90 + .90, meanlog = theta, sdlog = tau)

delta_95_lengths$normal_lower <- qnorm(delta_95, mean = mu, sd = sig)
delta_95_lengths$normal_upper <- qnorm(delta_95 + .95, mean = mu, sd = sig)

delta_95_lengths$lnormal_lower <- qlnorm(delta_95, meanlog = theta, sdlog = tau)
delta_95_lengths$lnormal_upper <- qlnorm(delta_95 + .95, meanlog = theta, sdlog = tau)

#Now for each normal and lognormal population, calculate the lengths of the found
#tolerance intervals for B=.9 and B=.95
delta_90_lengths$normal_length <- delta_90_lengths$normal_upper - delta_90_lengths$normal_lower
delta_90_lengths$lnormal_length <- delta_90_lengths$lnormal_upper - delta_90_lengths$lnormal_lower

delta_95_lengths$normal_length <- delta_95_lengths$normal_upper - delta_95_lengths$normal_lower
delta_95_lengths$lnormal_length <- delta_95_lengths$lnormal_upper - delta_95_lengths$lnormal_lower

# #print lengths
delta_90_lengths$normal_length

```

```

## [1] 7.334206 6.917641 6.713169 6.610919 6.579415 6.610919 6.713169 6.917641
## [9] 7.334206

```

```
delta_90_lengths$lnormal_length
```

```

## [1] 203.3386 215.5298 230.5148 248.9872 272.2925 302.8731 345.6123 412.3103
## [9] 544.8254

```

```
delta_95_lengths$normal_length
```

```
## [1] 8.154068 7.869085 7.869085 8.154068
```

```
delta_95_lengths$lnormal_length
```

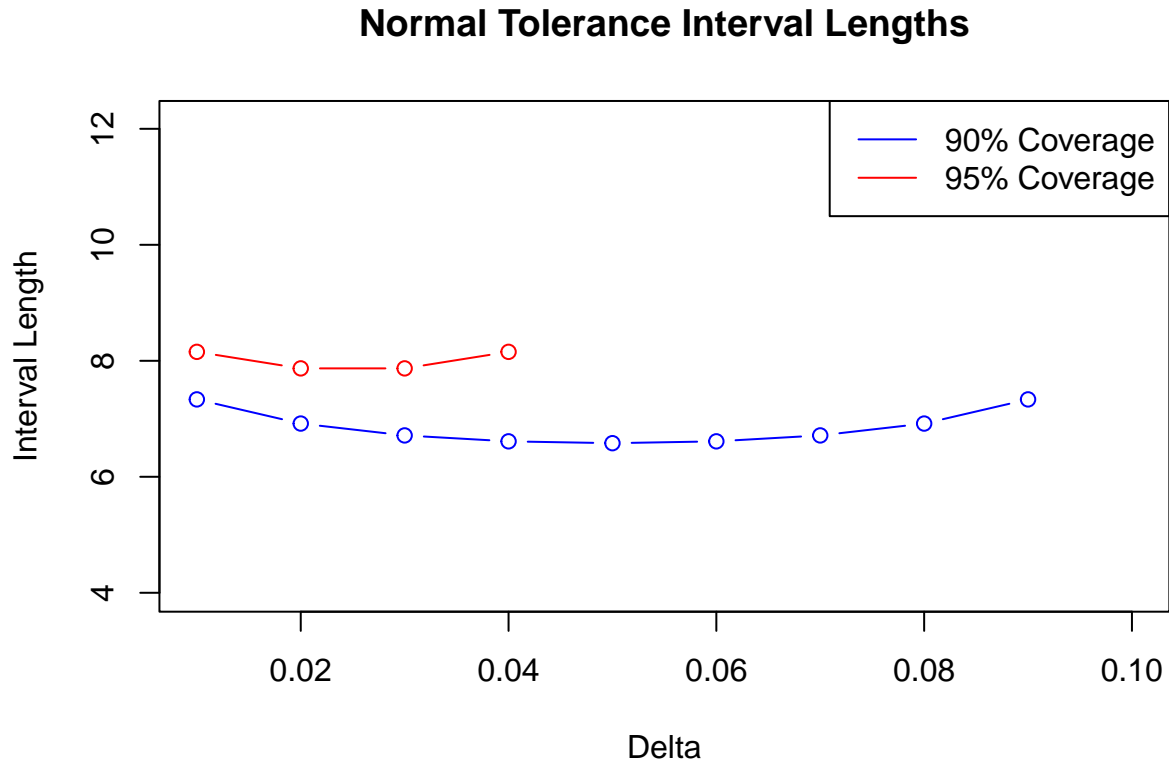
```
## [1] 309.0747 351.0909 417.3814 549.6297
```

```

# Plot lengths for normal population
plot(delta_90_lengths$delta, delta_90_lengths$normal_length, type = "b",
     col = "blue", ylim = c(4, max(delta_95_lengths$normal_length)+4),
     xlim = c(.01, .1), xlab = "Delta", ylab = "Interval Length",
     main = "Normal Tolerance Interval Lengths")

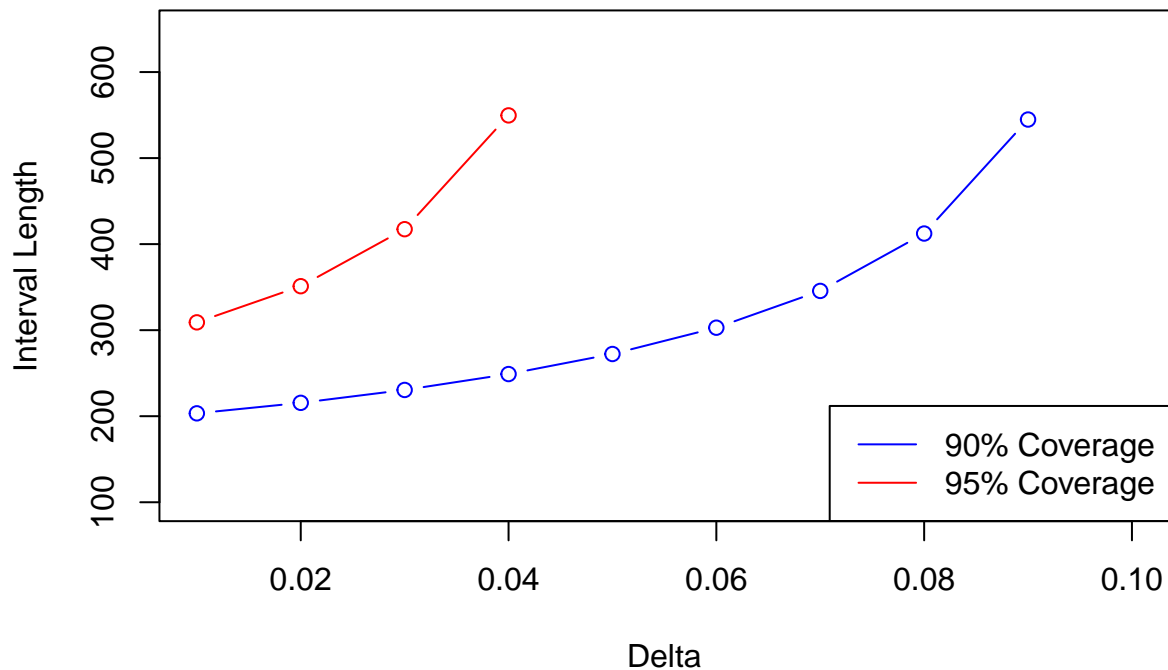
```

```
lines(delta_95_lengths$delta, delta_95_lengths$normal_length, type = "b", col = "red")
legend("topright", legend = c("90% Coverage", "95% Coverage"), col = c("blue", "red"),
      lty = 1)
```



```
# Plot lengths for lognormal population
plot(delta_90_lengths$delta, delta_90_lengths$lnormal_length, type = "b",
      col = "blue", ylim = c(100,max(delta_95_lengths$lnormal_length)+100),
      xlim = c(.01,.1), xlab = "Delta", ylab = "Interval Length",
      main = "Lognormal Tolerance Interval Lengths")
lines(delta_95_lengths$delta, delta_95_lengths$lnormal_length, type = "b", col = "red")
legend("bottomright", legend = c("90% Coverage", "95% Coverage"), col = c("blue", "red"),
      lty = 1)
```

Lognormal Tolerance Interval Lengths



```
cat("From the plots we can see that as delta increases, the lengths of the intervals
    increase for lognormal distributions; there is a positive correlation. However for
    a normal distribution the delta affects the interval length parabolically, meaning that
    extreme values being the min/max of delta are where the TI interval lengths are
    maximized.")
```

```
## From the plots we can see that as delta increases, the lengths of the intervals
##     increase for lognormal distributions; there is a positive correlation. However for
##     a normal distribution the delta affects the interval length parabolically, meaning that
##     extreme values being the min/max of delta are where the TI interval lengths are
##     maximized.
```

```
 #(e) Obtain a random sample of size n = 25 from this population. Obtain 90% TI
    #from your sample. Compare the TI based on exact TI obtained in part (a).
```

```
library(EnvStats)
```

```
##
## Attaching package: 'EnvStats'
```

```
## The following objects are masked from 'package:stats':
##
##     predict, predict.lm
```

```

set.seed(1)

#n=25 samples for normal and lognormal populations
norm_sample <- rnorm(25, mean=mu, sd=sig)
lnorm_sample <- rlnorm(25, meanlog=theta, sdlog=tau)

#Normal distribution 90% TI
sample_norm <- tolIntNorm(norm_sample, coverage = 0.90, ti.type = "two-sided", conf.level = 0.95,
                          method = "exact")
sample_norm

##
## Results of Distribution Parameter Estimation
## -----
##
## Assumed Distribution:          Normal
##
## Estimated Parameter(s):       mean = 10.337330
##                               sd   = 1.900216
##
## Estimation Method:            mvue
##
## Data:                         norm_sample
##
## Sample Size:                  25
##
## Tolerance Interval Coverage:   90%
##
## Coverage Type:                content
##
## Tolerance Interval Method:     Exact
##
## Tolerance Interval Type:       two-sided
##
## Confidence Level:              95%
##
## Tolerance Interval:            LTL = 6.128191
##                               UTL = 14.546470
##

#Lognormal distribution 90% TI
sample_lnorm <- tolIntLnorm(lnorm_sample, coverage = 0.90, ti.type = "two-sided", conf.level = 0.95,
                           method = "exact")
sample_lnorm

##
## Results of Distribution Parameter Estimation
## -----
##
## Assumed Distribution:          Lognormal
##
## Estimated Parameter(s):       meanlog = 4.0322314
##                               sdlog   = 0.7062806
##

```



```
##
## Estimation Method:          mvue
##
## Data:                      lnorm_sample
##
## Sample Size:               25
##
## Tolerance Interval Coverage: 90%
##
## Coverage Type:             content
##
## Tolerance Interval Method:   Exact
##
## Tolerance Interval Type:     two-sided
##
## Confidence Level:           95%
##
## Tolerance Interval:         LTL = 11.79599
##                             UTL = 269.53627
```

```
cat(sprintf("n=25 Normal sample 90%% TI: (%.2f, %.2f)\n",
            sample_norm$interval$limits[1], sample_norm$interval$limits[2]))
```

```
## n=25 Normal sample 90% TI: (6.13, 14.55)
```

```
cat(sprintf("90%% Normal TI: (%.2f, %.2f)\n",
            qlnorm(lower_90, mean = mu, sd = sig),
            qlnorm(upper_90, mean = mu, sd = sig)))
```

```
## 90% Normal TI: (6.71, 13.29)
```

```
cat(sprintf("n=25 Lognormal sample 90%% TI: (%.2f, %.2f)\n",
            sample_lnorm$interval$limits[1], sample_lnorm$interval$limits[2]))
```

```
## n=25 Lognormal sample 90% TI: (11.80, 269.54)
```

```
cat(sprintf("90%% Lognormal TI: (%.2f, %.2f)\n",
            qlnorm(lower_90, meanlog = theta, sdlog = tau),
            qlnorm(upper_90, meanlog = theta, sdlog = tau)))
```

```
## 90% Lognormal TI: (10.54, 282.83)
```

```
cat("For normal and lognormal populations, it is expected that the sample calculation
of the tolerance intervals be different than the actual tolerance intervals, when
using a sample of n=25. This is because the sample size is small, and is less
likely to accurately represent the population parameters, especially
for lognormal since it is inherently skewed. From the output above we can see
that the normal sample calculation does differ slightly from the actual
tolerance interval, and the lognormal sample calculation is significantly different
from the actual TI.")
```

```
## For normal and lognormal populations, it is expected that the sample calculation
## of the tolerance intervals be different than the actual tolerance intervals, when
## using a sample of n=25. This is because the sample size is small, and is less
## likely to accurately represent the population parameters, especially
## for lognormal since it is inherently skewed. From the output above we can see
## that the normal sample calculation does differ slightly from the actual
## tolerance interval, and the lognormal sample calculation is significantly different
## from the actual TI.
```