Technical University of Košice

Faculty of Electrical Engineering and Informatics

Applied Statistics

Exercises 3-4

Worked Solutions in R

Jana Fortes

Random Sampling and Estimation Theory

Exercise 3.1

Sample mean, variance, and standard deviation

Problem statement (textbook, p. 44)

Twenty one-kilogram bags of sugar were weighed. Their weights, in kg, were:

1.00; 1.01; 1.05; 0.99; 0.95; 1.00; 0.98; 0.99; 1.04; 1.06; 0.93; 1.00; 1.03; 0.97; 1.00; 0.99; 1.05; 1.01; 0.94; 1.00.

For this sample, calculate:
a) the sample arithmetic mean,
b) the sample variance,
c) the sample standard deviation.

R solution

```
# EXERCISE 3.1

x = c(1.00, 1.01, 1.05, 0.99, 0.95, 1.00, 0.98, 0.99,
      1.04, 1.06, 0.93, 1.00, 1.03, 0.97, 1.00, 0.99,
      1.05, 1.01, 0.94, 1.00)

# a) Sample arithmetic mean
sample_mean = mean(x)

# b) Sample variance
sample_variance = var(x)

# c) Sample standard deviation
sample_sd = sd(x)
```

Exercise 4.1

Normal mean: population variance known

Problem statement (textbook, p. 48)

A random sample comes from a normal population with variance 7.4.

xj	9	11	12	14	15	16	17	18	20	21
nj	1	2	3	4	7	5	4	3	2	1

Using the sample values xj and absolute frequencies nj above, determine:

a) the 99% two-sided confidence interval for the unknown mean,

b) the 90% one-sided confidence interval for the unknown mean with a lower bound,

c) the 95% one-sided confidence interval for the unknown mean with an upper bound.

R solution

```
# EXERCISE 4.1

values = c(9, 11, 12, 14, 15, 16, 17, 18, 20, 21)
frequencies = c(1, 2, 3, 4, 7, 5, 4, 3, 2, 1)
sigma = sqrt(7.4)

n = sum(frequencies)
# Repeat each value according to its frequency to form the data vector.
x = rep(values, frequencies)

# Point estimate of the population mean
sample_mean = mean(x)

# a) 99% two-sided confidence interval for the population mean
alpha1 = 0.01                     # Significance level
q1 = qnorm(1 - alpha1 / 2, 0, 1)   # Standard normal quantile
lower = sample_mean - q1 * sigma / sqrt(n)
upper = sample_mean + q1 * sigma / sqrt(n)
cat("[", lower, ",", upper, "]")   # Print the confidence interval

# Alternatively, using the TeachingDemos package
TeachingDemos::z.test(x, mu = sample_mean, sd = sigma,
                      alternative = "two.sided", conf.level = 0.99)

# b) 90% one-sided confidence interval with a lower bound
alpha2 = 0.1
q2 = qnorm(1 - alpha2, 0, 1)
lower = sample_mean - q2 * sigma / sqrt(n)
cat("[", lower, ", Inf)")

# Alternatively, using the TeachingDemos package
TeachingDemos::z.test(x, mu = sample_mean, sd = sigma,
                      alternative = "greater", conf.level = 0.9)

# c) 95% one-sided confidence interval with an upper bound
alpha3 = 0.05
q3 = qnorm(1 - alpha3, 0, 1)
upper = sample_mean + q3 * sigma / sqrt(n)
cat("(-Inf,", upper, "]")

# Alternatively, using the TeachingDemos package
TeachingDemos::z.test(x, mu = sample_mean, sd = sigma,
                      alternative = "less", conf.level = 0.95)
```

Exercise 4.2

Normal mean: population variance unknown

Problem statement (textbook, p. 50)

A random sample comes from a normal population.

xj	9	11	12	14	15	16	17	18	20	21
nj	1	2	3	3	6	4	4	3	2	1

Using the sample values xj and absolute frequencies nj above, determine:

a) the 90% two-sided confidence interval for the unknown mean,

b) the 99% one-sided confidence interval for the unknown mean with a lower bound,

c) the 95% one-sided confidence interval for the unknown mean with an upper bound.

R solution

```
# EXERCISE 4.2

values = c(9, 11, 12, 14, 15, 16, 17, 18, 20, 21)
frequencies = c(1, 2, 3, 3, 6, 4, 4, 3, 2, 1)

n = sum(frequencies)
# Repeat each value according to its frequency to form the data vector.
x = rep(values, frequencies)

# Point estimate of the population mean
sample_mean = mean(x)

# a) 90% two-sided confidence interval for the population mean
alpha1 = 0.1                          # Significance level
q1 = qt(1 - alpha1 / 2, n - 1)        # Student's t quantile
sample_sd = sd(x)                     # Sample standard deviation
lower = sample_mean - q1 * sample_sd / sqrt(n)
upper = sample_mean + q1 * sample_sd / sqrt(n)
cat("[", lower, ",", upper, "]")      # Print the confidence interval

# Alternatively, using the stats package
stats::t.test(x, mu = sample_mean,
              alternative = "two.sided", conf.level = 0.9)

# b) 99% one-sided confidence interval with a lower bound
alpha2 = 0.01
q2 = qt(1 - alpha2, n - 1)
lower = sample_mean - q2 * sample_sd / sqrt(n)
cat("[", lower, ", Inf)")

# Alternatively, using the stats package
stats::t.test(x, mu = sample_mean,
              alternative = "greater", conf.level = 0.99)

# c) 95% one-sided confidence interval with an upper bound
alpha3 = 0.05
q3 = qt(1 - alpha3, n - 1)
upper = sample_mean + q3 * sample_sd / sqrt(n)
cat("(-Inf,", upper, "]")

# Alternatively, using the stats package
stats::t.test(x, mu = sample_mean,
              alternative = "less", conf.level = 0.95)
```

Exercise 4.3

Normal variance: population mean known

Problem statement (textbook, p. 52)

A random sample comes from a normal population with mean 15.

xj	9	11	12	14	15	16	17	18	20	21
nj	1	2	3	4	7	5	4	3	2	1

Using the sample values xj and absolute frequencies nj above, determine:

a) the 99% two-sided confidence interval for the variance,

b) the 95% one-sided confidence interval for the variance with a lower bound,

c) the 95% one-sided confidence interval for the variance with an upper bound.

R solution

```
# EXERCISE 4.3

values = c(9, 11, 12, 14, 15, 16, 17, 18, 20, 21)
frequencies = c(1, 2, 3, 4, 7, 5, 4, 3, 2, 1)

n = sum(frequencies)
# Repeat each value according to its frequency to form the data vector.
x = rep(values, frequencies)

# Point estimate of the population variance; known mean mu = 15
variance_known_mean = sum(frequencies * (values - 15)^2) / n

# a) 99% two-sided confidence interval for the population variance
alpha1 = 0.01
q1 = qchisq(alpha1 / 2, n)       # Chi-square quantile, n degrees of freedom
q2 = qchisq(1 - alpha1 / 2, n)
lower = n * variance_known_mean / q2
upper = n * variance_known_mean / q1
cat("[", lower, ",", upper, "]") # Print the confidence interval

# b) 95% one-sided confidence interval with a lower bound
alpha = 0.05                     # Significance level
q22 = qchisq(1 - alpha, n)
lower = n * variance_known_mean / q22
cat("[", lower, ", Inf)")

# c) 95% one-sided confidence interval with an upper bound
q11 = qchisq(alpha, n)
upper = n * variance_known_mean / q11
cat("(0,", upper, "]")
```

Exercise 4.4

Normal variance: population mean unknown

Problem statement (textbook, p. 54)

A random sample comes from a normal population.

xj	9	11	12	14	15	16	17	18	20	21
nj	1	2	3	4	7	5	4	3	2	1

Using the sample values xj and absolute frequencies nj above, determine:

a) the 99% two-sided confidence interval for the variance,

b) the 95% one-sided confidence interval for the variance with a lower bound,

c) the 95% one-sided confidence interval for the variance with an upper bound.

R solution

```
# EXERCISE 4.4

values = c(9, 11, 12, 14, 15, 16, 17, 18, 20, 21)
frequencies = c(1, 2, 3, 4, 7, 5, 4, 3, 2, 1)

n = sum(frequencies)
# Repeat each value according to its frequency to form the data vector.
x = rep(values, frequencies)

# Point estimate of the population variance
sample_variance = var(x)

# a) 99% two-sided confidence interval for the population variance
alpha1 = 0.01
q1 = qchisq(alpha1 / 2, n - 1)    # Chi-square quantile, n-1 df
q2 = qchisq(1 - alpha1 / 2, n - 1)
lower = (n - 1) * sample_variance / q2
upper = (n - 1) * sample_variance / q1
cat("[", lower, ",", upper, "]")  # Print the confidence interval

# Alternatively, using the DescTools package
DescTools::VarTest(x, sigma.squared = sample_variance,
                   conf.level = 0.99)

# b) 95% one-sided confidence interval with a lower bound
alpha = 0.05                      # Significance level
q22 = qchisq(1 - alpha, n - 1)
lower = (n - 1) * sample_variance / q22
cat("[", lower, ", Inf)")

# Alternatively, using the DescTools package
DescTools::VarTest(x, sigma.squared = sample_variance,
                   conf.level = 0.95, alternative = "greater")

# c) 95% one-sided confidence interval with an upper bound
q11 = qchisq(alpha, n - 1)
upper = (n - 1) * sample_variance / q11
cat("(0,", upper, "]")

# Alternatively, using the DescTools package
DescTools::VarTest(x, sigma.squared = sample_variance,
                   conf.level = 0.95, alternative = "less")
```

Exercise 4.5

Exponential distribution: estimating the mean lambda

Problem statement (textbook, p. 56)

For 100 randomly selected passengers, bus waiting times (in minutes) were recorded.

xj	0.5	2.5	4.5	6.5	8.5
nj	60	20	11	5	4

Assume an exponential distribution with mean parameter lambda. Determine:

a) the 95% two-sided confidence interval for lambda,
b) the 99% one-sided confidence interval for lambda with a lower bound,
c) the 99% one-sided confidence interval for lambda with an upper bound.

R solution

```
# EXERCISE 4.5

values = c(0.5, 2.5, 4.5, 6.5, 8.5)
frequencies = c(60, 20, 11, 5, 4)

n = sum(frequencies)
# Repeat each value according to its frequency to form the data vector.
x = rep(values, frequencies)

# Point estimate of lambda (the exponential mean)
sample_mean = mean(x)

# a) 95% two-sided confidence interval for lambda
alpha = 0.05                        # Significance level
q1 = qchisq(alpha / 2, 2 * n)       # Chi-square quantile, 2n df
q2 = qchisq(1 - alpha / 2, 2 * n)
lower = (2 * n * sample_mean) / q2
upper = (2 * n * sample_mean) / q1
cat("[", lower, ",", upper, "]")    # Print the confidence interval

# b) 99% one-sided confidence interval with a lower bound
alpha1 = 0.01
q11 = qchisq(1 - alpha1, 2 * n)
lower = (2 * n * sample_mean) / q11
cat("[", lower, ", Inf)")

# c) 99% one-sided confidence interval with an upper bound
q22 = qchisq(alpha1, 2 * n)
upper = (2 * n * sample_mean) / q22
cat("(0,", upper, "]")
```