

Technical University of Košice
Faculty of Electrical Engineering and Informatics



Applied Statistics

The 1st and 2nd Lesson - Solved Problems in R

Jana Fortes, Mária Švecová

EXERCISE 2.1

```
x = c(17, 18, 18, 22, 18, 17, 16, 17, 18, 18, 19, 18, 19, 17, 19, 19, 17, 18, 17, 18, 21, 17, 19, 18, 17, 20, 19,
      17, 18, 20, 19, 15, 19, 19, 19, 19, 18, 21, 18, 18)

# a)-----
ordered_sample = sort(x)                                # Ordered sample

# Frequency table (frequency distribution table)
frequency_counts = table(x)                             # Table of absolute frequencies
absolute_frequencies = as.numeric(frequency_counts)
distinct_values = unique(x)
ordered_values = sort(distinct_values)
n = length(x)
relative_frequencies = absolute_frequencies / n
cumulative_absolute_frequencies = cumsum(absolute_frequencies)
cumulative_relative_frequencies = cumsum(relative_frequencies)
frequency_table = cbind(ordered_values, absolute_frequencies,
                        cumulative_absolute_frequencies, relative_frequencies,
                        cumulative_relative_frequencies)

# b)-----
# Numerical characteristics
summary_statistics = summary(x)                          # Median, first quartile, and third quartile

# Mode from the frequency table
arithmetic_mean = mean(x)
variance = ((n - 1) / n) * var(x)
standard_deviation = sqrt(variance)
mean_deviation = (1 / n) * sum(abs(x - arithmetic_mean))
coefficient_of_variation = standard_deviation / arithmetic_mean
data_range = max(x) - min(x)
median = as.numeric(summary_statistics[3])
first_quartile = as.numeric(summary_statistics[2])
third_quartile = as.numeric(summary_statistics[5])
interquartile_range = third_quartile - first_quartile
coefficient_of_skewness = (1 / ((standard_deviation^3) * n)) * sum((x - arithmetic_mean)^3)
coefficient_of_kurtosis = (1 / ((standard_deviation^4) * n)) * sum((x - arithmetic_mean)^4) - 3

# Alternatively, using the e1071 package
coefficient_of_skewness_e1071 = skewness(x)
coefficient_of_kurtosis_e1071 = kurtosis(x)

statistics = data.frame(
  arithmetic_mean,
  variance,
  standard_deviation,
  mean_deviation,
  coefficient_of_variation,
  data_range,
  median,
  first_quartile,
  third_quartile,
  interquartile_range,
  coefficient_of_skewness,
  coefficient_of_kurtosis
)
round(statistics, digits = 3)                            # Display values rounded to three decimal places

# c)-----
# Different types of plots
plot(ordered_values, absolute_frequencies, col = "blue", xlab = "x_i", pch = 19) # Scatter plot of n_i
plot(ordered_values, absolute_frequencies, type = "o", col = "blue", xlab = "x_i", pch = 19) # Frequency polygon
barplot(cumulative_absolute_frequencies, xlab = "x_i", col = "blue", ylab = "N_i") # Bar chart of N_i
hist(x, main = NULL, col = "blue", border = "white", xlab = "x_i") # Histogram
pie(ordered_values, labels = ordered_values, col = gray(seq(from = 0, to = 1, length = 5))) # Pie chart
boxplot(x)                                                # Box plot

# Bonus-----
# Define a function that finds the mode
get_mode = function(v) {
  unique_values = unique(v)
  unique_values[which.max(tabulate(match(v, unique_values)))]
}
mode = get_mode(x)
```

EXERCISE 2.2

```
x = c(4.25, 4.26, 4.25, 4.07, 4.20, 4.40, 4.05, 4.00, 4.17, 3.96, 4.30, 4.22, 4.25, 4.15, 4.25, 4.25, 3.97, 4.10,
      4.47, 4.10, 4.17, 4.05, 4.25, 4.18, 4.05, 4.27, 4.03, 4.05, 4.15, 4.15, 4.40, 4.27, 4.25, 4.12, 4.30, 4.17,
      4.20, 4.17, 4.22, 4.32, 4.20, 4.05, 4.20, 4.37, 4.20, 4.25, 4.25, 4.15, 4.33, 4.10, 4.15, 4.25, 4.25, 4.05,
      4.32, 4.27, 4.12, 4.27, 4.50, 4.38, 4.27, 4.17, 4.17, 4.12, 4.25, 4.12, 4.30, 4.33, 4.33, 4.21, 4.12, 4.15,
      4.22, 4.20, 4.25, 4.50, 4.27, 4.10, 4.15, 4.31, 4.32, 4.10, 4.10, 4.17, 4.32, 4.02, 4.20, 4.17, 4.30, 4.44,
      4.17, 4.20, 4.45, 4.07, 4.27, 4.15, 4.12, 4.42, 4.21, 4.35)

# a)-----
# Grouping into class intervals
minimum = min(x)
maximum = max(x)
n = length(x)
estimated_number_of_intervals = round(sqrt(n))
estimated_interval_width = (maximum - minimum) / estimated_number_of_intervals
number_of_intervals = 11
interval_width = 0.05

# Determine the class boundaries and the corresponding absolute frequencies
# One option
class_boundaries = seq(3.95, 4.5, by = 0.05) # From 3.95 to 4.50 in increments of 0.05
histogram_data = hist(x, seq(3.95, 4.5, by = 0.05), prob = FALSE)
absolute_frequencies = histogram_data[["counts"]] # Absolute frequencies from the histogram

# Alternatively
histogram_data = hist(x, seq(3.95, 4.5, by = 0.05), plot = FALSE)
class_boundaries = histogram_data$breaks # Class boundaries from the histogram
absolute_frequencies = histogram_data$counts # Absolute frequencies from the histogram
cat(class_boundaries, "\n", absolute_frequencies, "\n")

class_midpoints = (class_boundaries[1:11] + class_boundaries[2:12]) / 2

# Frequency table (frequency distribution table)

relative_frequencies = absolute_frequencies / n
cumulative_absolute_frequencies = cumsum(absolute_frequencies)
cumulative_relative_frequencies = cumsum(relative_frequencies)
frequency_table = cbind(
  class_midpoints,
  absolute_frequencies,
  cumulative_absolute_frequencies,
  relative_frequencies,
  cumulative_relative_frequencies
)

# b)-----
# Numerical characteristics

arithmetic_mean = mean(x)
variance = ((n - 1) / n) * var(x)
standard_deviation = sqrt(variance)
mean_deviation = (1 / n) * sum(abs(x - arithmetic_mean))
coefficient_of_variation = standard_deviation / arithmetic_mean
range = max(x) - min(x)
median = 4.15 + interval_width * (((n + 1) / 2) - 33) / 18
first_quartile = 4.10 + interval_width * ((n / 4) - 19) / 14
third_quartile = 4.25 + interval_width * (((3 * n) / 4) - 69) / 12
interquartile_range = third_quartile - first_quartile
coefficient_of_skewness = (1 / ((standard_deviation^3) * n)) * sum((x - arithmetic_mean)^3)
coefficient_of_kurtosis = (1 / ((standard_deviation^4) * n)) * sum((x - arithmetic_mean)^4) - 3

# Alternatively, using the e1071 package
coefficient_of_skewness = skewness(x)
coefficient_of_kurtosis = kurtosis(x)

summary_statistics = data.frame(
  arithmetic_mean,
  variance,
  standard_deviation,
  mean_deviation,
  coefficient_of_variation,
  range,
  median,
  first_quartile,
  third_quartile,
  interquartile_range,
  coefficient_of_skewness,
  coefficient_of_kurtosis
)
round(summary_statistics, digits = 3) # Display the values rounded to three decimal places

# Modal class from the frequency table
# Mode
delta_11 = absolute_frequencies[5] - absolute_frequencies[4]
delta_21 = absolute_frequencies[5] - absolute_frequencies[6]
mode_1 = class_boundaries[5] + interval_width * (delta_11 / (delta_11 + delta_21))
delta_12 = absolute_frequencies[6] - absolute_frequencies[5]
delta_22 = absolute_frequencies[5] - absolute_frequencies[7]
mode_2 = class_boundaries[6] + interval_width * (delta_12 / (delta_12 + delta_22))
```

```
# c)-----  
# Different types of plots  
  
plot(class_midpoints, absolute_frequencies, type = "o", col = "blue",  
      xlab = "z_j", ylab = "n_j", pch = 19)          # Frequency polygon  
hist(x, main = NULL, col = "blue", border = "white", xlab = "x") # Histogram  
boxplot(x)                                           # Box plot
```