Worksheet_#5

Carl Louie A. Quillo

2023-12-11

1. Create a data frame for the table below. Show your solution.

```
student <- c(1,2,3,4,5,6,7,8,9,10)
pre_test_scores <- c(55, 54, 47, 57, 51, 61, 57, 54, 63, 58)
post_test_scores <- c(61, 60, 56, 63, 56, 63, 59, 56, 62, 61)

studenttest <- data.frame(student, pre_test_scores, post_test_scores)
```

a. Compute the descriptive statistics using different packages (Hmisc and pastecs). Write the codes and its result.

```
library(Hmisc)
```

```
##
## Attaching package: 'Hmisc'
```

```
## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
library(pastecs)

sumofstudents <- summary(student)
startifstudents <- stat.desc(student)
```

2. The Department of Agriculture was studying the effects of several levels of a fertilizer on the growth of a plant. For some analyses, it might be useful to convert the fertilizer levels to an ordered factor.

- The data were 10,10,10, 20,20,50,10,20,10,50,20,50,20,10. a. Write the codes and describe the result.

```
fertilizerlevel <- c(10,10,10,20,20,50,10,20,10,50,20,50,20,10)
sortedFert <- sort(fertilizerlevel)
orderedFert <- order(fertilizerlevel)
```

3. Abdul Hassan, president of Floor Coverings Unlimited, has asked you to study the ex- ercise levels undertaken by 10 subjects were “l”, “n”, “n”, “i”, “l” , “l”, “n”,

“n”, “i”, “l” ; n=none, l=light, i=intense

```
exercise_levels <- c("l", "n", "i")

subject_exercise_levels <- c("l", "n", "n", "i", "l", "l", "n", "n", "i", "l")

ordered_exercise_levels <- factor(subject_exercise_levels, levels = exercise_levels)

summary(ordered_exercise_levels)
```

```
## l n i
## 4 4 2
```

4. Sample of 30 tax accountants from all the states and territories of Australia and their individual state of origin is specified by a character vector of state mnemonics as:

```
state <- c("tas", "sa", "qld", "nsw", "nsw", "nt", "wa", "wa", "qld",
"vic", "nsw", "vic", "qld", "qld", "sa", "tas", "sa", "nt",
"wa", "vic", "qld", "nsw", "nsw", "wa", "sa", "act", "nsw",
"vic", "vic", "act")
```

a. Apply the factor function and factor level. Describe the results.

```
state_factor <- factor(state)
state_factor_level <- levels(state_factor)
```

5. From #4 - continuation:

- Suppose we have the incomes of the same tax accountants in another vector (in suitably large units of money) incomes <- c(60, 49, 40, 61, 64, 60, 59, 54, 62, 69, 70, 42, 56, 61, 61, 61, 58, 51, 48, 65, 49, 49, 41, 48, 52, 46, 59, 46, 58, 43)

```
incomes <- c(60, 49, 40, 61, 64, 60, 59, 54,
62, 69, 70, 42, 56, 61, 61, 61, 58, 51, 48,
65, 49, 49, 41, 48, 52, 46, 59, 46, 58, 43)
```

a. Calculate the sample mean income for each state we can now use the special function

```
mean_factor <- tapply(incomes, state_factor, mean)
```

b. Copy the results and interpret.

act nsw nt qld sa tas vic wa 44.50000 57.33333 55.50000 53.60000 55.00000 60.50000 56.00000 52.25000

6. Calculate the standard errors of the state income means (refer again to number 3)

```
state_counts <- table(state_factor)

stdError <- sqrt(mean_factor^2/state_counts)
```

a. What is the standard error? Write the codes.

```
state_se <- sqrt(sum(incomes^2) / length(incomes))
```

b. Interpret the result.

```
#a state with a wide range of income values may have a higher standard error compared to a state with a
```

7. Use the titanic dataset.

a. subset the titatic dataset of those who survived and not survived. Show the codes and its result.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:pastecs':
##
##     first, last

## The following objects are masked from 'package:Hmisc':
##
##     src, summarize
## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
data("Titanic")

Titanic <- data.frame(Titanic)

survived <- Titanic %>%
 filter(Survived == "Yes")

survived
```

```
##    Class    Sex   Age Survived Freq
## 1    1st   Male Child      Yes    5
## 2    2nd   Male Child      Yes   11
## 3    3rd   Male Child      Yes   13
## 4   Crew   Male Child      Yes    0
## 5    1st Female Child      Yes    1
## 6    2nd Female Child      Yes   13
## 7    3rd Female Child      Yes   14
## 8   Crew Female Child      Yes    0
## 9    1st   Male Adult      Yes   57
## 10   2nd   Male Adult      Yes   14
## 11   3rd   Male Adult      Yes   75
## 12  Crew   Male Adult      Yes  192
## 13   1st Female Adult      Yes  140
## 14   2nd Female Adult      Yes   80
## 15   3rd Female Adult      Yes   76
## 16  Crew Female Adult      Yes   20
```

```
not_survived <- Titanic %>%
 filter(Survived == "No")

not_survived
```

```
##    Class    Sex   Age Survived Freq
## 1    1st   Male Child       No    0
## 2    2nd   Male Child       No    0
## 3    3rd   Male Child       No   35
## 4   Crew   Male Child       No    0
## 5    1st Female Child       No    0
## 6    2nd Female Child       No    0
## 7    3rd Female Child       No   17
## 8   Crew Female Child       No    0
## 9    1st   Male Adult       No  118
## 10   2nd   Male Adult       No  154
## 11   3rd   Male Adult       No  387
## 12  Crew   Male Adult       No  670
## 13   1st Female Adult       No    4
## 14   2nd Female Adult       No   13
## 15   3rd Female Adult       No   89
## 16  Crew Female Adult       No    3
```

8. The data sets are about the breast cancer Wisconsin. The samples arrive periodically as Dr. Wolberg reports his clinical cases. The database therefore reflects this

chronologihttps://drive.google.com/file/d/16MFLoehCgx2MJuNSAuB2CsBy6eDIIr- u/view?usp=drive_link)

```r
library(readr)
breastcancer_wisconsin <- read_csv("breastcancer_wisconsin.csv")
```

```
## Rows: 699 Columns: 11
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr  (1): bare_nucleoli
## dbl (10): id, clump_thickness, size_uniformity, shape_uniformity, marginal_a...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

a. describe what is the dataset all about.

```r
breastcancer_wisconsin
```

```
## # A tibble: 699 x 11
##         id clump_thickness size_uniformity shape_uniformity marginal_adhesion
##      <dbl>           <dbl>           <dbl>            <dbl>             <dbl>
##  1 1000025               5               1                1                 1
##  2 1002945               5               4                4                 5
##  3 1015425               3               1                1                 1
##  4 1016277               6               8                8                 1
##  5 1017023               4               1                1                 3
##  6 1017122               8              10               10                 8
##  7 1018099               1               1                1                 1
##  8 1018561               2               1                2                 1
##  9 1033078               2               1                1                 1
## 10 1033078               4               2                1                 1
## # i 689 more rows
## # i 6 more variables: epithelial_size <dbl>, bare_nucleoli <chr>,
## #   bland_chromatin <dbl>, normal_nucleoli <dbl>, mitoses <dbl>, class <dbl>
```

```r
# Its all about breast cancer
```

d. Compute the descriptive statistics using different packages. Find the values of: d.1 Standard error of the mean for clump thickness. d.2 Coefficient of variability for Marginal Adhesion. d.3 Number of null values of Bare Nuclei. d.4 Mean and standard deviation for Bland Chromatin d.5 Confidence interval of the mean for Uniformity of Cell Shape

```r
#d.1 Standard error of the mean for clump thickness.

clump_thickness <- as.numeric(breastcancer_wisconsin$clump_thickness)
mean_clump_thickness <- mean(clump_thickness)
std_dev_clump_thickness <- sd(clump_thickness)
n <- length(clump_thickness)
standard_error <- std_dev_clump_thickness / sqrt(n)
standard_error
```

```
## [1] 0.1065011
```

```
#d.2 Coefficient of variability for Marginal Adhesion.

marginal_adhesion <- as.numeric(breastcancer_wisconsin$marginal_adhesion)
mean_marginal_adhesion <- mean(marginal_adhesion)
std_dev_marginal_adhesion <- sd(marginal_adhesion)
coefficient_of_variability <- std_dev_marginal_adhesion / mean_marginal_adhesion
coefficient_of_variability
```

```
## [1] 1.017283
```

```
#d.3 Number of null values of Bare Nuclei.

bare_nuclei <- as.numeric(breastcancer_wisconsin$bare_nucleoli)
```

```
## Warning: NAs introduced by coercion
```

```
number_of_null_values <- sum(is.na(bare_nuclei))
number_of_null_values
```

```
## [1] 16
```

```
#d.4 Mean and standard deviation for Bland Chromatin

bland_chromatin <- as.numeric(breastcancer_wisconsin$bland_chromatin)
mean_bland_chromatin <- mean(bland_chromatin)
std_dev_bland_chromatin <- sd(bland_chromatin)
list(mean = mean_bland_chromatin, sd = std_dev_bland_chromatin)
```

```
## $mean
## [1] 3.437768
##
## $sd
## [1] 2.438364
```

```
#d.5 Confidence interval of the mean for Uniformity of Cell Shape

uniformity_of_cell_shape <- as.numeric(breastcancer_wisconsin$shape_uniformity)
mean_uniformity_of_cell_shape <- mean(uniformity_of_cell_shape)
std_dev_uniformity_of_cell_shape <- sd(uniformity_of_cell_shape)
n <- length(uniformity_of_cell_shape)
confidence_level <- 0.95
z_score <- qnorm(1 - (1 - confidence_level) / 2)
margin_of_error <- z_score * (std_dev_uniformity_of_cell_shape / sqrt(n))
confidence_interval <- c(mean_uniformity_of_cell_shape - margin_of_error, mean_uniformity_of_cell_shape
confidence_interval
```

```
## [1] 2.987123 3.427755
```

d. How many attributes?

```
# d.1
attribute_mean <- mean(breastcancer_wisconsin$clump_thickness)
attribute_se <- sqrt(var(breastcancer_wisconsin$clump_thickness) / length(breastcancer_wisconsin$clump_

d.1 <- attribute_se

# d.2
attribute_mean <- mean(breastcancer_wisconsin$marginal_adhesion)
```

```
attribute_cv <- sqrt(var(breastcancer_wisconsin$marginal_adhesion) / mean(breastcancer_wisconsin$margina

d.2 <- attribute_cv

# d.3
d.3 <- sum(is.na(breastcancer_wisconsin$bare_nuclei))
```

```
## Warning: Unknown or uninitialised column: `bare_nuclei`.
```

```
# d.4
attribute_mean <- mean(breastcancer_wisconsin$bland_chromatin)
attribute_std_dev <- sqrt(var(breastcancer_wisconsin$bland_chromatin))

d.4 <- c(mean = attribute_mean, std_dev = attribute_std_dev)

# d.5
attribute_mean <- mean(breastcancer_wisconsin$uniformity_of_cell_shape)
```

```
## Warning: Unknown or uninitialised column: `uniformity_of_cell_shape`.
```

```
## Warning in mean.default(breastcancer_wisconsin$uniformity_of_cell_shape):
## argument is not numeric or logical: returning NA
```

```
margin_of_error <- qt(0.975, df = length(breastcancer_wisconsin$uniformity_of_cell_shape) - 1) * (attri
```

```
## Warning: Unknown or uninitialised column: `uniformity_of_cell_shape`.
```

```
## Warning in qt(0.975, df =
## length(breastcancer_wisconsin$uniformity_of_cell_shape) - : NaNs produced
```

```
## Warning: Unknown or uninitialised column: `uniformity_of_cell_shape`.
```

```
d.5 <- c(mean = attribute_mean,
         lower_bound = attribute_mean - margin_of_error,
         upper_bound = attribute_mean + margin_of_error)
```

e. Find the percentage of respondents who are malignant. Interpret the results.

```
breastcancer_wisconsin$clump_thickness <- as.numeric(breastcancer_wisconsin$clump_thickness)
breastcancer_wisconsin$size_uniformity <- as.numeric(breastcancer_wisconsin$size_uniformity)
breastcancer_wisconsin$bare_nucleoli <- as.numeric(breastcancer_wisconsin$bare_nucleoli)
```

```
## Warning: NAs introduced by coercion
```

```
breastcancer_wisconsin$bare_nucleoli <- as.numeric(breastcancer_wisconsin$shape_uniformity)
breastcancer_wisconsin$bare_nucleoli <- as.numeric(breastcancer_wisconsin$marginal_adhesion)
breastcancer_wisconsin$bare_nucleoli <- as.numeric(breastcancer_wisconsin$epithelial_size)
breastcancer_wisconsin$bare_nucleoli <- as.numeric(breastcancer_wisconsin$bland_chromatin)
breastcancer_wisconsin$bare_nucleoli <- as.numeric(breastcancer_wisconsin$normal_nucleoli)
breastcancer_wisconsin$bare_nucleoli <- as.numeric(breastcancer_wisconsin$mitoses)



mean_values <- mean(breastcancer_wisconsin[,2:10])
```

```
## Warning in mean.default(breastcancer_wisconsin[, 2:10]): argument is not
## numeric or logical: returning NA
```

```
breastcancer_wisconsin <- rbind(breastcancer_wisconsin, mean_values)
```

```
percentage_malignant <- 100 * breastcancer_wisconsin$class[breastcancer_wisconsin$class == "malignant"]

print(paste("Percentage of respondents who are malignant:", percentage_malignant))
```

```
## [1] "Percentage of respondents who are malignant: NA"
```

```
correlation_matrix <- cor(breastcancer_wisconsin[, c("clump_thickness", "size_uniformity", "shape_unifo
```

9. Export the data abalone to the Microsoft excel file. Copy the codes. install.packages("AppliedPredictiveModeling") library("AppliedPredictiveModeling") view(abalone) head(abalone) summary(abalone)

```
#install.packages("AppliedPredictiveModeling")
#library("AppliedPredictiveModeling")

#View(abalone)
#head(abalone)
#summary(abalone)
```