rWorksheet_5

Rey Angelo Calopez BSIT 2-C

2023-12-13

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

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

```
StudentScore <- data.frame(Student = c(1,2,3,4,5,6,7,8,9,10),
                           PreTest = c(55,54,47,57,51,61,57,54,63,58),
                           PostTest = c(61,60,56,63,56,63,59,56,62,61))

StudentScore
```

```
##    Student PreTest PostTest
## 1        1      55       61
## 2        2      54       60
## 3        3      47       56
## 4        4      57       63
## 5        5      51       56
## 6        6      61       63
## 7        7      57       59
## 8        8      54       56
## 9        9      63       62
## 10      10      58       61
```

```
#a.Compute the descriptive statistics using different packages (Hmisc and pastecs).
library(Hmisc)
```

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

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

```
library(pastecs)


HmiscStats <- describe(StudentScore[,c("PreTest","PostTest")])
HmiscStats
```

```
## StudentScore[, c("PreTest", "PostTest")]
##
##  2  Variables      10  Observations
## ---------------------------------------------------------------------------
## PreTest
##        n  missing distinct     Info     Mean      Gmd
##       10        0        8    0.988     55.7    5.444
```

```
##
## Value       47  51  54  55  57  58  61  63
## Frequency    1   1   2   1   2   1   1   1
## Proportion 0.1 0.1 0.2 0.1 0.2 0.1 0.1 0.1
##
## For the frequency table, variable is rounded to the nearest 0
## ----------------------------------------------------------------------------
## PostTest
##        n  missing distinct     Info     Mean      Gmd
##       10        0        6    0.964     59.7    3.311
##
## Value       56  59  60  61  62  63
## Frequency    3   1   1   2   1   2
## Proportion 0.3 0.1 0.1 0.2 0.1 0.2
##
## For the frequency table, variable is rounded to the nearest 0
## ----------------------------------------------------------------------------
```

```
# Calculate descriptive statistics using pastecs
pastecsStats <- apply(StudentScore[,c('PreTest','PostTest')], 2, function(x) summary(x))
pastecsStats
```

```
##         PreTest PostTest
## Min.      47.00    56.00
## 1st Qu.   54.00    56.75
## Median    56.00    60.50
## Mean      55.70    59.70
## 3rd Qu.   57.75    61.75
## Max.      63.00    63.00
```

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.

```
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
```

```
fertilizerLevels <- c(10,10,10, 20,20,50,10,20,10,50,20,50,20,10)

orderedFactor <- factor(fertilizerLevels, levels = unique(fertilizerLevels))
```

```
basicStats <- summary(orderedFactor)
basicStats
```

```
## 10 20 50
##  6  5  3
```

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 a. What is the best way to represent this in R?

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

ExerciseFactor <- factor(excerciseLevels, levels = c("n","l","i"))


basic_stats <- summary(ExerciseFactor)
basic_stats
```

```
## n l i
## 5 4 1
```

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 <- 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")
stateFactor <- factor(state)
stateFactor
```

```
##  [1] tas sa  qld nsw nsw nt  wa  wa  qld vic nsw vic qld qld sa  tas sa  nt  wa
## [20] vic qld nsw nsw wa  sa  act nsw vic vic act
## Levels: act nsw nt qld sa tas vic wa
```

```
summaryState <- summary(stateFactor)
summaryState
```

```
## act nsw  nt qld  sa tas vic  wa
##   2   6   2   5   4   2   5   4
```

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)

a. Calculate the sample mean income for each state we can now use the special function tapply(): Example: giving a means vector with the components labelled by the levels incmeans <- tapply(incomes, statef, mean) Note: The function tapply() is used to apply a function, here mean(), to each group of components of the first argument, here incomes, defined by the levels of the second component, here state 2

```
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)
```

```r
meanIncome <- tapply(incomes, stateFactor, mean)
meanIncome
```

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

b. 6.Calculate the standard errors of the state income means (refer again to number 3) stdError <- function(x) sqrt(var(x)/length(x)) Note: After this assignment, the standard errors are calculated by: incster <- tapply(incomes, statef, stdError)

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

```r
stdError <- function(x) sqrt(var(x)/length(x))
incster <- tapply(incomes, state, stdError)
standardError <- tapply(incomes, stateFactor, stdError)
standardError
```

```
##      act      nsw       nt      qld       sa      tas      vic       wa
## 1.500000 4.310195 4.500000 4.106093 2.738613 0.500000 5.244044 2.657536
```

7. Use the titanic dataset.

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

```r
# Install and load the 'titanic' package if not already installed
install.packages("titanic")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.3'
## (as 'lib' is unspecified)
```

```r
library(titanic)

# Load the Titanic dataset
data("titanic_train")
titanic_data <- titanic_train

# Subset the dataset into those who survived and those who did not survive
survived_data <- subset(titanic_data, Survived == 1)
not_survived_data <- subset(titanic_data, Survived == 0)

# Show the first few rows of the resulting datasets
head(survived_data)
```

```
##    PassengerId Survived Pclass
## 2            2        1      1
## 3            3        1      3
## 4            4        1      1
## 9            9        1      3
## 10          10        1      2
## 11          11        1      3
##                                                  Name    Sex Age SibSp Parch
## 2  Cumings, Mrs. John Bradley (Florence Briggs Thayer) female  38     1     0
## 3                             Heikkinen, Miss. Laina female  26     0     0
## 4        Futrelle, Mrs. Jacques Heath (Lily May Peel) female  35     1     0
## 9    Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg) female  27     0     2
## 10                Nasser, Mrs. Nicholas (Adele Achem) female  14     1     0
## 11                    Sandstrom, Miss. Marguerite Rut female   4     1     1
##              Ticket    Fare Cabin Embarked
## 2         PC 17599 71.2833   C85        C
```

```
## 3  STON/O2. 3101282  7.9250             S
## 4            113803 53.1000  C123       S
## 9            347742 11.1333             S
## 10           237736 30.0708             C
## 11          PP 9549 16.7000    G6       S
```

```
head(not_survived_data)
```

```
##    PassengerId Survived Pclass                          Name  Sex Age SibSp
## 1            1        0      3       Braund, Mr. Owen Harris male  22     1
## 5            5        0      3      Allen, Mr. William Henry male  35     0
## 6            6        0      3              Moran, Mr. James male  NA     0
## 7            7        0      1       McCarthy, Mr. Timothy J male  54     0
## 8            8        0      3 Palsson, Master. Gosta Leonard male   2     3
## 13          13        0      3 Saundercock, Mr. William Henry male  20     0
##    Parch    Ticket    Fare Cabin Embarked
## 1      0 A/5 21171  7.2500              S
## 5      0    373450  8.0500              S
## 6      0    330877  8.4583              Q
## 7      0     17463 51.8625   E46        S
## 8      1    349909 21.0750              S
## 13     0 A/5. 2151  8.0500              S
```

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

a. describe what is the dataset all about. Ans. This data set shows all levels related to breast cancer like clump thickness, size, shape and many more.

b. 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

```
library(readr)
library(rcompanion)
breastcancer_wisconsin <- read_csv("breastcancer_wisconsin.csv",col_types = cols(
  id = col_double(),
  clump_thickness = col_double(),
  size_uniformity = col_double(),
  shape_uniformity = col_double(),
  marginal_adhesion = col_double(),
  epithelial_size = col_double(),
  bare_nucleoli = col_character(),
  bland_chromatin = col_double(),
  normal_nucleoli = col_double(),
  mitoses = col_double(),
  class = col_double()
))
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>
```

```
#d.1 Standard error of the mean for clump thickness.
clump_thickness <- sd(breastcancer_wisconsin$clump_thickness) / sqrt(length(breastcancer_wisconsin$clump
cat("d.1 Standard Error of the Mean for Clump Thickness:", clump_thickness, "\n")
```

```
## d.1 Standard Error of the Mean for Clump Thickness: 0.1065011
```

```
# d.2 Coefficient of variability for Marginal Adhesion.
marginal_adhesion <- sd(breastcancer_wisconsin$marginal_adhesion) / mean(breastcancer_wisconsin$marginal
cat("d.2 Coefficient of Variability for Marginal Adhesion:", marginal_adhesion, "%\n")
```

```
## d.2 Coefficient of Variability for Marginal Adhesion: 1.017283 %
```

```
# d.3 Number of null values of Bare Nuclei.
values_bare_nuclei <- sum(is.na(breastcancer_wisconsin$bare_nucleoli))
cat("d.3 Number of Null Values in Bare Nuclei:", values_bare_nuclei, "\n")
```

```
## d.3 Number of Null Values in Bare Nuclei: 15
```

```
# d.4 Mean and standard deviation for Bland Chromatin.
mean_bland_chromatin <- mean(breastcancer_wisconsin$bland_chromatin)
mean_bland_chromatin
```

```
## [1] 3.437768
```

```
sd_bland_chromatin <- sd(breastcancer_wisconsin$bland_chromatin)
cat("d.4 Mean for Bland Chromatin:", mean_bland_chromatin, "\n")
```

```
## d.4 Mean for Bland Chromatin: 3.437768
```

```
cat("   Standard Deviation for Bland Chromatin:", sd_bland_chromatin, "\n")
```

```
##    Standard Deviation for Bland Chromatin: 2.438364
```

```
# d.5 Confidence interval of the mean for Uniformity of Cell Shape.
uniformity_of_cell_shape <- t.test(breastcancer_wisconsin$shape_uniformity, conf.level = 0.95)$conf.int
cat("d.5 Confidence Interval for the Mean of Uniformity of Cell Shape:", uniformity_of_cell_shape, "\n"
```

```
## d.5 Confidence Interval for the Mean of Uniformity of Cell Shape: 2.986741 3.428138
```

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")

data(abalone)
#View(abalone)
head(abalone)
```

```
##   Type LongestShell Diameter Height WholeWeight ShuckedWeight VisceraWeight
## 1    M        0.455    0.365  0.095      0.5140        0.2245        0.1010
## 2    M        0.350    0.265  0.090      0.2255        0.0995        0.0485
## 3    F        0.530    0.420  0.135      0.6770        0.2565        0.1415
## 4    M        0.440    0.365  0.125      0.5160        0.2155        0.1140
## 5    I        0.330    0.255  0.080      0.2050        0.0895        0.0395
## 6    I        0.425    0.300  0.095      0.3515        0.1410        0.0775
##   ShellWeight Rings
## 1       0.150    15
## 2       0.070     7
## 3       0.210     9
## 4       0.155    10
## 5       0.055     7
## 6       0.120     8
```

```
summary(abalone)
```

```
##  Type       LongestShell      Diameter          Height        WholeWeight
##  F:1307   Min.   :0.075   Min.   :0.0550   Min.   :0.0000   Min.   :0.0020
##  I:1342   1st Qu.:0.450   1st Qu.:0.3500   1st Qu.:0.1150   1st Qu.:0.4415
##  M:1528   Median :0.545   Median :0.4250   Median :0.1400   Median :0.7995
##           Mean   :0.524   Mean   :0.4079   Mean   :0.1395   Mean   :0.8287
##           3rd Qu.:0.615   3rd Qu.:0.4800   3rd Qu.:0.1650   3rd Qu.:1.1530
##           Max.   :0.815   Max.   :0.6500   Max.   :1.1300   Max.   :2.8255
##  ShuckedWeight    VisceraWeight     ShellWeight         Rings
##  Min.   :0.0010   Min.   :0.0005   Min.   :0.0015   Min.   : 1.000
##  1st Qu.:0.1860   1st Qu.:0.0935   1st Qu.:0.1300   1st Qu.: 8.000
##  Median :0.3360   Median :0.1710   Median :0.2340   Median : 9.000
##  Mean   :0.3594   Mean   :0.1806   Mean   :0.2388   Mean   : 9.934
##  3rd Qu.:0.5020   3rd Qu.:0.2530   3rd Qu.:0.3290   3rd Qu.:11.000
##  Max.   :1.4880   Max.   :0.7600   Max.   :1.0050   Max.   :29.000
```

```
#export

library(xlsx)

#write.xlsx(abalone, "abalone.xlsx")
```