Rworksheet_Ulgasan6

2023-12-21

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
```

```
install.packages("pastecs")
```

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

```
library(pastecs)



pastecsStats <- stat.desc(StudentScore[, c('PreTest', 'PostTest')])
pastecsStats
```

```
##                   PreTest     PostTest
## nbr.val       10.00000000  10.00000000
## nbr.null       0.00000000   0.00000000
## nbr.na         0.00000000   0.00000000
## min           47.00000000  56.00000000
## max           63.00000000  63.00000000
## range         16.00000000   7.00000000
## sum          557.00000000 597.00000000
## median        56.00000000  60.50000000
## mean          55.70000000  59.70000000
## SE.mean        1.46855938   0.89504811
## CI.mean.0.95   3.32211213   2.02473948
## var           21.56666667   8.01111111
## std.dev        4.64399254   2.83039063
## coef.var       0.08337509   0.04741023
```

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.

```r
library(dplyr)
```

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

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

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

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

```r
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?


```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.

```r
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
```

```r
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

```r
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)

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) a. 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.packages("titanic")
```

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

```r
library(titanic)

data("titanic_train")
titanic_data <- titanic_train

survived_data <- subset(titanic_data, Survived == 1)

not_survived_data <- subset(titanic_data, Survived == 0)

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
```

```r
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
```

```r
survived_data <- titanic_data[titanic_data$Survived == 1, ]

not_survived_data <- titanic_data[titanic_data$Survived == 0, ]
```

```r
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
```

```r
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

chronologihttps://drive.google.com/file/d/16MFLoehCgx2MJuNSAuB2CsBy6eDIIr- u/view?usp=drive_link)
a. describe what is the dataset all about.

```r
#The dataset consists of cytological features of breast cancer cell samples, such as clump thickness, s
```

d. Compute the descriptive statistics using different packages. Find the values of:

d.1 Standard error of the mean for clump thickness.

```r
library(readr)

breastcancer_wisconsin <- read_csv("/cloud/project/Worksheet_6/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.
```

```
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>
```

```
clump_thickness_mean <- mean(breastcancer_wisconsin$clump_thickness)
clump_thickness_sd <- sd(breastcancer_wisconsin$clump_thickness)
clump_thickness_sem <- clump_thickness_sd / sqrt(length(breastcancer_wisconsin$clump_thickness))

clump_thickness_mean
```

```
## [1] 4.41774
```

```
clump_thickness_sd
```

```
## [1] 2.815741
```

```
clump_thickness_sem
```

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

d.2 Coefficient of variability for Marginal Adhesion.

```
colnames(breastcancer_wisconsin)
```

```
##  [1] "id"                "clump_thickness"   "size_uniformity"
##  [4] "shape_uniformity"  "marginal_adhesion" "epithelial_size"
##  [7] "bare_nucleoli"     "bland_chromatin"   "normal_nucleoli"
## [10] "mitoses"           "class"
```

```
marginal_adhesion_cv <- sd(breastcancer_wisconsin$`Marginal Adhesion`) / mean(breastcancer_wisconsin$`Ma
```

```
## Warning: Unknown or uninitialised column: `Marginal Adhesion`.
## Unknown or uninitialised column: `Marginal Adhesion`.
```

```
## Warning in mean.default(breastcancer_wisconsin$`Marginal Adhesion`, na.rm =
## TRUE): argument is not numeric or logical: returning NA
```

```
marginal_adhesion_cv
```

```
## [1] NA
```

d.3 Number of null values of Bare Nuclei.

```
bare_nuclei_null_count <- sum(is.na(breastcancer_wisconsin$`Bare Nuclei`))
```

```
## Warning: Unknown or uninitialised column: `Bare Nuclei`.
```

```
bare_nuclei_null_count
```

```
## [1] 0
```

d.4 Mean and standard deviation for Bland Chromatin

```
# Check column names
colnames(breastcancer_wisconsin)
```

```
##  [1] "id"                "clump_thickness"   "size_uniformity"  
##  [4] "shape_uniformity"  "marginal_adhesion" "epithelial_size"  
##  [7] "bare_nucleoli"     "bland_chromatin"   "normal_nucleoli"  
## [10] "mitoses"           "class"
```

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

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

```
col_index <- grep("Bland Chromatin", colnames(breastcancer_wisconsin))


bland_chromatin_mean <- mean(as.numeric(breastcancer_wisconsin[, col_index]), na.rm = TRUE)
bland_chromatin_sd <- sd(as.numeric(breastcancer_wisconsin[, col_index]), na.rm = TRUE)

bland_chromatin_mean
```

```
## [1] NaN
```

```
bland_chromatin_sd
```

```
## [1] NA
```

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

```
if ("Uniformity of Cell Shape" %in% names(breastcancer_wisconsin) && !all(is.na(breastcancer_wisconsin$

  pop_mean <- 10  # Replace this with your actual population mean


  uniformity_cell_shape_ci <- t.test(breastcancer_wisconsin$`Uniformity of Cell Shape`, mu = pop_mean)$


  uniformity_cell_shape_ci
} else {
  cat("Error: 'Uniformity of Cell Shape' column is missing or contains only missing values.\n")
}
```

```
## Error: 'Uniformity of Cell Shape' column is missing or contains only missing values.
```

9.Export the data abalone to the Microsoft excel file. Copy the codes.

```r
install.packages("AppliedPredictiveModeling")
```

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

```r
install.packages("MASS")
```

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

```r
install.packages("openxlsx")
```

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

```r
library("AppliedPredictiveModeling")
library(MASS)
```

```
##
## Attaching package: 'MASS'
```

```
## The following object is masked from 'package:dplyr':
##
##     select
```

```r
library(openxlsx)

data(abalone)
str(abalone)
```

```
## 'data.frame':    4177 obs. of  9 variables:
##  $ Type         : Factor w/ 3 levels "F","I","M": 3 3 1 3 2 2 1 1 3 1 ...
##  $ LongestShell : num  0.455 0.35 0.53 0.44 0.33 0.425 0.53 0.545 0.475 0.55 ...
##  $ Diameter     : num  0.365 0.265 0.42 0.365 0.255 0.3 0.415 0.425 0.37 0.44 ...
##  $ Height       : num  0.095 0.09 0.135 0.125 0.08 0.095 0.15 0.125 0.125 0.15 ...
##  $ WholeWeight  : num  0.514 0.226 0.677 0.516 0.205 ...
##  $ ShuckedWeight: num  0.2245 0.0995 0.2565 0.2155 0.0895 ...
##  $ VisceraWeight: num  0.101 0.0485 0.1415 0.114 0.0395 ...
##  $ ShellWeight  : num  0.15 0.07 0.21 0.155 0.055 0.12 0.33 0.26 0.165 0.32 ...
##  $ Rings        : int  15 7 9 10 7 8 20 16 9 19 ...
```

```r
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  
```

```
openxlsx::write.xlsx(abalone, "/cloud/project/RWorksheet_Ulgasan#4.xlsx", sheetName = "AbaloneData", ro
```