Worksheet 7a

Kirby Clarence Alegoria

2022-12-10

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

```
#1. Create a data frame for the table below. Figure 1: Student Scores
```

```
student <- seq(1:10)
    preTest <- c(55,54,47,57,51,61,57,54,63,58)
    postTest <- c(61,60,56,63,56,63,59,56,62,61)

    DataFr <- data.frame(student,preTest,postTest)
    DataFr
```

```
##    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). Write the codes and its result.
```

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.2.2
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.2.2
```

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

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

```
library(pastecs)
```

```
## Warning: package 'pastecs' was built under R version 4.2.2
```

```
    #describe(DF)

    #stat.desc(DF)
```

```
#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.
```

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

    #a. Write the codes and describe the result.

    Order <- sort(DepartmentofAgriculture, decreasing = FALSE)
    Order
```

```
##  [1] 10 10 10 10 10 10 20 20 20 20 20 50 50 50
```

```
   #Based on the result, the fertilizer levels were sorted from the lowest
   #to the highest value.
```

```
#3. Abdul Hassan, president of Floor Coverings Unlimited, has asked you to study
#   the exercise 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?
```

```
#   The best way to represent this is through dataframe.

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

    exerciselvl <- data.frame(exercise_levels)
    exerciselvl
```

```
##    exercise_levels
## 1                l
## 2                n
```

```
## 3                n
## 4                i
## 5                l
## 6                l
## 7                n
## 8                n
## 9                i
## 10               l
```

```
    exerlvl <- data.frame(
      none = c("n", "n", "n", "n"),
      light = c("l", "l", "l", "l"),
      intense = c("i", "i", "NA", "NA")
    )
    exerlvl
```

```
##   none light intense
## 1    n     l       i
## 2    n     l       i
## 3    n     l      NA
## 4    n     l      NA
```

```
#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")
    state
```

```
##  [1] "tas" "sa"  "qld" "nsw" "nsw" "nt"  "wa"  "wa"  "qld" "vic" "nsw" "vic"
## [13] "qld" "qld" "sa"  "tas" "sa"  "nt"  "wa"  "vic" "qld" "nsw" "nsw" "wa"
## [25] "sa"  "act" "nsw" "vic" "vic" "act"
```

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

    factorfunc <- function(state)
    factorfunc
```

```
#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)
```

```r
#a. Calculate the sample mean income for each state we can now use the special
 #    function tapply():
```

```r
cal <- tapply(state, incomes, mean)
```

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

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

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

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

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

## Warning in mean.default(X[[i]], ...): argument is not numeric or logical:
## returning NA

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

```
    cal
```

```
## 40 41 42 43 46 48 49 51 52 54 56 58 59 60 61 62 64 65 69 70
## NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

```
  #b. Copy the results and interpret.
  #    40 41 42 43 46 48 49 51 52 54 56 58 59 60 61 62 64 65 69 70
  #    NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

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

```
    calcustd.n <- length(cal)
    calculate.sd <- sd(cal)
    calstd.se <- calculate.sd/sqrt(calcustd.n)
    calstd.se
```

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

```
    #a. What is the standard error? Write the codes.
    #   The result of the code above is NA.
```

```
   #b. Interpret the result.
   #   The result is not available for the reason that the other objects are
   #   classified as character type, therefore it won't be able to get
   #   the standard error.
```

```
#7. Use the titanic dataset.
```

```
    data("Titanic")

    titanset<- data.frame(Titanic)
    titanset
```

```
##    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
## 17   1st   Male Child      Yes    5
## 18   2nd   Male Child      Yes   11
## 19   3rd   Male Child      Yes   13
## 20  Crew   Male Child      Yes    0
## 21   1st Female Child      Yes    1
## 22   2nd Female Child      Yes   13
## 23   3rd Female Child      Yes   14
## 24  Crew Female Child      Yes    0
## 25   1st   Male Adult      Yes   57
## 26   2nd   Male Adult      Yes   14
## 27   3rd   Male Adult      Yes   75
## 28  Crew   Male Adult      Yes  192
## 29   1st Female Adult      Yes  140
## 30   2nd Female Adult      Yes   80
## 31   3rd Female Adult      Yes   76
## 32  Crew Female Adult      Yes   20
```

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

```
        Sbset <- subset(titanset, select = "Survived")
        Sbset
```

```
##    Survived
## 1        No
## 2        No
## 3        No
## 4        No
## 5        No
## 6        No
## 7        No
## 8        No
## 9        No
## 10       No
## 11       No
## 12       No
## 13       No
## 14       No
## 15       No
## 16       No
## 17      Yes
## 18      Yes
```

```
## 19      Yes
## 20      Yes
## 21      Yes
## 22      Yes
## 23      Yes
## 24      Yes
## 25      Yes
## 26      Yes
## 27      Yes
## 28      Yes
## 29      Yes
## 30      Yes
## 31      Yes
## 32      Yes
```

```
#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 chronological grouping of the data. You can create
#   this dataset in Microsoft Excel.
```

```
  #a. describe what is the dataset all about.
  #   The dataset is all about Breast Cancer Wisconsin.
```

```
  #b. Import the data from MS Excel. Copy the codes.
```

```
library("readxl")
```

```
## Warning: package 'readxl' was built under R version 4.2.2
```

```
  DATA <- read_excel("C:/Users/acer/Documents/BSIT-2A/WorkSheet#7a/Breast_Cancer.xlsx")
  DATA
```

```
## # A tibble: 49 x 11
##         Id CL. thickne~1 Cell ~2 Cell ~3 Marg.~4 Epith~5 Bare.~6 Bl. C~7 Norma~8
##      <dbl>         <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>     <dbl>   <dbl>
##  1 1000025             5       1       1       1       2 1             3       1
##  2 1002945             5       4       4       5       7 10            3       2
##  3 1015425             3       1       1       1       2 2             3       1
##  4 1016277             6       8       8       1       3 4             3       7
##  5 1017023             4       1       1       3       2 1             3       1
##  6 1017122             8      10      10       8       7 10            9       7
##  7 1018099             1       1       1       1       2 10            3       1
##  8 1018561             2       1       2       1       2 1             3       1
##  9 1033078             2       1       1       1       2 1             1       1
## 10 1033078             4       2       1       1       2 1             2       1
## # ... with 39 more rows, 2 more variables: Mitoses <dbl>, Class <chr>, and
## #   abbreviated variable names 1: 'CL. thickness', 2: 'Cell size',
## #   3: 'Cell Shape', 4: 'Marg. Adhesion', 5: 'Epith. C.size',
## #   6: 'Bare. Nuclei', 7: 'Bl. Cromatin', 8: 'Normal nucleoli'
```

```r
#c. Compute the descriptive statistics using different packages.
#   Find the values of:
```

```r
#c.1 Standard error of the mean for clump thickness.
     clump <- length(DATA$`CL. thickness`)
     clump1 <- sd(DATA$`CL. thickness`)
     clump2 <- clump1/sqrt(DATA$`CL. thickness`)
     clump2
```

```
##  [1] 1.2812754 1.2812754 1.6541194 1.1696391 1.4325095 1.0129371 2.8650189
##  [8] 2.0258743 2.0258743 1.4325095 2.8650189 2.0258743 1.2812754 2.8650189
## [15] 1.0129371 1.0828754 1.4325095 1.4325095 0.9059985 1.1696391 1.0828754
## [22] 0.9059985 1.6541194 1.0129371 2.8650189 1.2812754 1.6541194 1.2812754
## [29] 2.0258743 2.8650189 1.6541194 2.0258743 0.9059985 2.0258743 1.6541194
## [36] 2.0258743 0.9059985 1.1696391 1.2812754 2.0258743 1.1696391 0.9059985
## [43] 1.1696391 1.2812754 0.9059985 2.8650189 1.6541194 2.8650189 1.4325095
```

```r
#c.2 Coefficient of variability for Marginal Adhesion.
     COV <- sd(DATA$`Marg. Adhesion`) / mean(DATA$`Marg. Adhesion`)* 100
     COV
```

```
## [1] 97.67235
```

```r
#c.3 Number of null values of Bare Nuclei.
     null_values <- subset(DATA,`Bare. Nuclei` == "NA")
```

```r
#c.4 Mean and standard deviation for Bland Chromatin
     mean(DATA$`Bl. Cromatin`)
```

```
## [1] 3.836735
```

```r
     sd(DATA$`Bl. Cromatin`)
```

```
## [1] 2.085135
```

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

   #Calculate the mean
     confInterMean <- mean(DATA$`Cell Shape`)
     confInterMean
```

```
## [1] 3.163265
```

```r
   #Calculate the standard error of the mean
     stdErrMean <- length(DATA$`Cell Shape`)
     stdB <- sd(DATA$`Cell Shape`)
     result <- stdB/sqrt(stdErrMean)
     result
```

```
## [1] 0.4158294
```

```r
    #Find the t-score that corresponds to the confidence level
      tscore = 0.05
      num1 = stdErrMean - 1
      num2 = qt(p = tscore/ 2, df = num1,lower.tail = F)
      num2
```

```
## [1] 2.010635
```

```r
    #Constructing the confidence interval
      num3 <- num2 * num1

    #LOWER
      num4 <- stdErrMean - num3

    #UPPER
      num5 <- stdErrMean + num3

      num6 <- c(num4,num5)
      num6
```

```
## [1] -47.51047 145.51047
```

```r
    #d. How many attributes?
        attributes(DATA)
```

```
## $class
## [1] "tbl_df"     "tbl"        "data.frame"
##
## $row.names
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
## [26] 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
##
## $names
##  [1] "Id"              "CL. thickness"   "Cell size"       "Cell Shape"
##  [5] "Marg. Adhesion"  "Epith. C.size"   "Bare. Nuclei"    "Bl. Cromatin"
##  [9] "Normal nucleoli" "Mitoses"         "Class"
```

```r
    #e. Find the percentage of respondents who are malignant.
    #   Interpret the results.
        percResp <- subset(DATA, Class == "maligant")
        percResp
```

```
## # A tibble: 16 x 11
##         Id CL. thickne~1 Cell ~2 Cell ~3 Marg.~4 Epith~5 Bare.~6 Bl. C~7 Norma~8
##      <dbl>         <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>     <dbl>   <dbl>
##  1 1041801             5       3       3       3       2 3             4       4
##  2 1044572             8       7       5      10       7 9             5       5
##  3 1047630             7       4       6       4       6 1             4       3
##  4 1050670            10       7       7       6       4 10            4       1
##  5 1054590             7       3       2      10       5 10            5       4
##  6 1054593            10       5       5       3       6 7             7      10
```

```
##  7 1057013             8       4       5       1       2 NA            7       3
##  8 1065726             5       2       3       4       2 7             3       6
##  9 1072179            10       7       7       3       8 5             7       4
## 10 1080185            10      10      10       8       6 1             8       9
## 11 1084584             5       4       4       9       2 10            5       6
## 12 1091262             2       5       3       3       6 7             7       5
## 13 1099510            10       4       3       1       3 3             6       5
## 14 1102573             5       6       5       6      10 1             3       1
## 15 1103608            10      10      10       4       8 1             8      10
## 16 1105257             3       7       7       4       4 9             4       8
## # ... with 2 more variables: Mitoses <dbl>, Class <chr>, and abbreviated
## #   variable names 1: ‘CL. thickness‘, 2: ‘Cell size‘, 3: ‘Cell Shape‘,
## #   4: ‘Marg. Adhesion‘, 5: ‘Epith. C.size‘, 6: ‘Bare. Nuclei‘,
## #   7: ‘Bl. Cromatin‘, 8: ‘Normal nucleoli‘
```

```
    #There 17 respondents who are malignant.
    #And there are total of 49 respondent.
```

```
    #Getting the percentage
        percenTage <- 17 / 49 * 100
        percenTage
```

```
## [1] 34.69388
```

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

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

```
library("AppliedPredictiveModeling")
```

```
## Warning: package 'AppliedPredictiveModeling' was built under R version 4.2.2
```

```
        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
```

```
        #Exporting the data abalone to the Microsoft excel file.
          #install.packages("xlsxjars")
```

```
library(xlsx)
```

```
## Warning: package 'xlsx' was built under R version 4.2.2
```

```
  write.xlsx("abalone","C:/Users/acer/Documents/BSIT-2A/WorkSheet#7a/abalone.xlsx")
```