RWorksheet_Salvador#6

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

```
scores <- data.frame(
  Student = c(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)
)
scores
```

```
##    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
```

#1a.Compute the descriptive statistics using different packages (Hmisc and pastecs). Write the codes and its result.

```
# Install and load the Hmisc package
install.packages("Hmisc")
```

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

```
library(Hmisc)
```

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

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

```
# I create a data frame named "scores".
scores <- data.frame(
  Student = c(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)
)

#compute descriptive statistics using hmisc
summary_hmisc <- describe(scores)
summary_hmisc
```

```
## scores
```

```
##
##  3  Variables      10  Observations
## --------------------------------------------------------------------------------
## Student
##        n  missing distinct     Info     Mean      Gmd      .05      .10
##       10        0       10        1      5.5    3.667     1.45     1.90
##      .25      .50      .75      .90      .95
##     3.25     5.50     7.75     9.10     9.55
##
## Value        1   2   3   4   5   6   7   8   9  10
## Frequency    1   1   1   1   1   1   1   1   1   1
## Proportion 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1 0.1
##
## For the frequency table, variable is rounded to the nearest 0
## --------------------------------------------------------------------------------
## 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
## --------------------------------------------------------------------------------
```

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

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

```
library(pastecs)

# Compute descriptive statistics using pastecs
summary_pastecs <- stat.desc(scores)
summary_pastecs
```

```
##                 Student      PreTest     PostTest
## nbr.val      10.0000000  10.00000000  10.00000000
## nbr.null      0.0000000   0.00000000   0.00000000
## nbr.na        0.0000000   0.00000000   0.00000000
## min           1.0000000  47.00000000  56.00000000
## max          10.0000000  63.00000000  63.00000000
## range         9.0000000  16.00000000   7.00000000
```

```
## sum          55.0000000 557.00000000 597.00000000
## median        5.5000000  56.00000000  60.50000000
## mean          5.5000000  55.70000000  59.70000000
## SE.mean       0.9574271   1.46855938   0.89504811
## CI.mean.0.95  2.1658506   3.32211213   2.02473948
## var           9.1666667  21.56666667   8.01111111
## std.dev       3.0276504   4.64399254   2.83039063
## coef.var      0.5504819   0.08337509   0.04741023
```

#2a. 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.

```
# this is the sample data given.
data <- c(10, 10, 10, 20, 20, 50, 10, 20, 10, 50, 20, 50, 20, 10)

# this is were i convert fertilizer levels to an ordered factor.
ferti_lvls <- ordered(data, levels = c(10, 20, 50))

# this is were i display the factor
ferti_lvls
```

```
##  [1] 10 10 10 20 20 50 10 20 10 50 20 50 20 10
## Levels: 10 < 20 < 50
```

```
#This code shows the different levels of fertilizer used in a plant study and provides a summary of the
```

#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?

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

exer_lvls <- factor(exercise_levels, levels = c("n", "l", "i"), labels = c("none", "light", "intense"))

# Display the factor variable
print(exer_lvls)
```

```
## [1] none    light   intense
## Levels: none light intense
```

#4a. 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.

```
# this is the sample data given
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")

# this is a factor variable
state_factor <- factor(state)

# display the factor levels
levels(state_factor)
```

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

```
#The result you see when using levels(state_factor) shows the specific categories or states that are pa
```

#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) #5a. 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 # • 2 #that tapply() also works in this case when its second argument is not a factor, # • e.g., 'tapply(incomes, state)', and this is true for quite a few other functions, since #arguments are coerced to factors when necessary (using as.factor()).

```
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)


# this line of my code calculate the sample mean income for each state using tapply.
inc_means <- tapply(incomes, state_factor, mean)

# Display the results
inc_means
```

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

#5b

#this is the output that shows the average income of each state. #act nsw nt qld sa tas vic wa #44.50000 57.33333 55.50000 53.60000 55.00000 60.50000 56.00000 52.25000

#The tapply function helps us in determining the average income of tax accountants in each state. The output provides us with the average income for tax accountants in different states.

#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)

#6a.What is the standard error? Write the codes.

```
stdError <- function(x) sqrt(var(x) / length(x))

inc_std_errors <- tapply(incomes, state_factor, stdError)

print(inc_std_errors)
```

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

#6b. Interpret the result. #These numbers show how much the average incomes in each place might change. If the numbers are higher, we're not so sure about the average incomes. The "stdError function" figures out these numbers, helps us know how sure we can be about the average incomes for tax accountants in different places.

#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")

survived <- subset(titanic_train, Survived == 1)

not_survived <- subset(titanic_train, Survived == 0)

head(survived)
```

```
##    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)
```

```
##    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
#These commands helps examine the data for passengers who either survived or did not survive the Titani
```

#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/16MFLoehCgx2MJuNSA#u/view?usp=drive_link)

#8a. the data set is about the data of breast cancer.

```
breastcancer_data <- read.csv("breastcancer_wisconsin.csv")

str(breastcancer_data)
```

```
## 'data.frame':    699 obs. of  11 variables:
##  $ id               : int  1000025 1002945 1015425 1016277 1017023 1017122 1018099 1018561 1033078 10
##  $ clump_thickness  : int  5 5 3 6 4 8 1 2 2 4 ...
##  $ size_uniformity  : int  1 4 1 8 1 10 1 1 1 2 ...
##  $ shape_uniformity : int  1 4 1 8 1 10 1 2 1 1 ...
##  $ marginal_adhesion: int  1 5 1 1 3 8 1 1 1 1 ...
##  $ epithelial_size  : int  2 7 2 3 2 7 2 2 2 2 ...
##  $ bare_nucleoli    : chr  "1" "10" "2" "4" ...
##  $ bland_chromatin  : int  3 3 3 3 3 9 3 3 1 2 ...
##  $ normal_nucleoli  : int  1 2 1 7 1 7 1 1 1 1 ...
##  $ mitoses          : int  1 1 1 1 1 1 1 1 5 1 ...
##  $ class            : int  2 2 2 2 2 4 2 2 2 2 ...
```

```
head(breastcancer_data)
```

```
##        id clump_thickness size_uniformity shape_uniformity marginal_adhesion
## 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
##   epithelial_size bare_nucleoli bland_chromatin normal_nucleoli mitoses class
## 1               2             1               3                1       1     2
## 2               7            10               3                2       1     2
## 3               2             2               3                1       1     2
## 4               3             4               3                7       1     2
## 5               2             1               3                1       1     2
## 6               7            10               9                7       1     4
```

```
summary(breastcancer_data)
```

```
##        id           clump_thickness  size_uniformity  shape_uniformity
##  Min.   :   61634   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
##  1st Qu.:  870688   1st Qu.: 2.000   1st Qu.: 1.000   1st Qu.: 1.000
##  Median : 1171710   Median : 4.000   Median : 1.000   Median : 1.000
##  Mean   : 1071704   Mean   : 4.418   Mean   : 3.134   Mean   : 3.207
##  3rd Qu.: 1238298   3rd Qu.: 6.000   3rd Qu.: 5.000   3rd Qu.: 5.000
##  Max.   :13454352   Max.   :10.000   Max.   :10.000   Max.   :10.000
##  marginal_adhesion epithelial_size  bare_nucleoli      bland_chromatin
##  Min.   : 1.000    Min.   : 1.000   Length:699         Min.   : 1.000
##  1st Qu.: 1.000    1st Qu.: 2.000   Class :character   1st Qu.: 2.000
##  Median : 1.000    Median : 2.000   Mode  :character   Median : 3.000
##  Mean   : 2.807    Mean   : 3.216                      Mean   : 3.438
##  3rd Qu.: 4.000    3rd Qu.: 4.000                      3rd Qu.: 5.000
##  Max.   :10.000    Max.   :10.000                      Max.   :10.000
##  normal_nucleoli     mitoses           class
```

```
##  Min.   : 1.000   Min.   : 1.000   Min.   :2.00
##  1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.:2.00
##  Median : 1.000   Median : 1.000   Median :2.00
##  Mean   : 2.867   Mean   : 1.589   Mean   :2.69
##  3rd Qu.: 4.000   3rd Qu.: 1.000   3rd Qu.:4.00
##  Max.   :10.000   Max.   :10.000   Max.   :4.00
```

#8d1.d.1 Standard error of the mean for clump thickness.

```
install.packages("psych")
```

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

```
library(psych)
```

```
##
## Attaching package: 'psych'
```

```
## The following object is masked from 'package:Hmisc':
##
##     describe
```

```
clump_thickness <- breastcancer_data$ClumpThickness
marginal_adhesion <- breastcancer_data$MarginalAdhesion
bare_nuclei <- breastcancer_data$BareNuclei
bland_chromatin <- breastcancer_data$BlandChromatin
uniformity_cell_shape <- breastcancer_data$UniformityCellShape

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

SE_clumpthickness <- sd(clump_thickness) / sqrt(length(clump_thickness))
SE_clumpthickness
```

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

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

CV_marginaladhesion <- sd(marginal_adhesion) / mean(marginal_adhesion)
```

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

```
CV_marginaladhesion
```

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

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

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

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

```
#d.4 Mean and standard deviation for Bland Chromatin
mean_blandchromatin <- mean(breastcancer_data$bland_chromatin)
sd_blandchromatin <- sd(breastcancer_data$bland_chromatin)
mean_blandchromatin
```

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

```
sd_blandchromatin
```

```
## [1] 2.438364
```

```
#d.5 Confidence interval of the mean for Uniformity of Cell Shape
ci_uniformitycellshape <- tryCatch(
  t.test(breastcancer_data$`uniformity_cell_shape`)$conf.int,
  error = function(e) NULL
)
```

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

```
ci_uniformitycellshape
```

```
## NULL
```

#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")
```

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

```
library(AppliedPredictiveModeling)

data("abalone")

install.packages("openxlsx")
```

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

```
library(openxlsx)

write.xlsx(abalone, file = "abalone.xlsx")

View(abalone)
```

```
## Warning in View(abalone): unable to open display
```

```
## Error in .External2(C_dataviewer, x, title): unable to start data viewer
```

```
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
```

```r
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
```