WORKSHEET 7A

Kylene Joy Yanguas

2022-12-10

#1. Create a data frame for the table below.

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

Students_ScoresDF <- data.frame(Student, Pre_test, Post_test )
```

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

```
library(Hmisc)
```

```
## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

## Loading required package: ggplot2

##
## Attaching package: 'Hmisc'

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

```
library(pastecs)
describe(Students_ScoresDF)
```

```
## Students_ScoresDF
##
##  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
##
## lowest :  1  2  3  4  5, highest:  6  7  8  9 10
##
## 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
## --------------------------------------------------------------------------------
## Pre_test
##        n  missing distinct     Info     Mean      Gmd
##       10        0        8    0.988     55.7    5.444
##
## lowest : 47 51 54 55 57, highest: 55 57 58 61 63
##
## 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
## --------------------------------------------------------------------------------
## Post_test
##        n  missing distinct     Info     Mean      Gmd
##       10        0        6    0.964     59.7    3.311
##
## lowest : 56 59 60 61 62, highest: 59 60 61 62 63
##
## 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
## --------------------------------------------------------------------------------
```

```
stat.desc(Students_ScoresDF)
```

```
##                  Student     Pre_test    Post_test
## 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
```

#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.

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

#a. Write the codes and describe the result.

```
  Orders <- factor(levels_of_fert, ordered = TRUE)
  Orders
```

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

#The result shows the ascending order of fertilizer levels.

#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

```
  subjects <- c("l","n","n","i","l","l","n","n","i","l")
  #a. What is the best way to represent this in R?
  subjectDF <- data.frame(subjects)
```

#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
```

```
##  [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
```

```
  levels(state_factor)
```

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

#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():

```
  incmeans <- tapply(incomes, state, mean)
  incmeans
```

```
##      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. 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)

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

#a. What is the standard error? Write the codes.
  incster <- tapply(incomes, state, stdError)
  incster
```

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

#b. Interpret the result. #the result of data shows the standard errors of the state income means.

#7. Use the titanic dataset.

```
  data("Titanic")

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

```
##    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.

```
Survives <- subset(Titanic, Survived == "Yes")
Survives
```

```
##    Class    Sex   Age Survived Freq
## 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
```

```
  Died <- subset(Titanic, Survived == "No")
  Died
```

```
##    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 #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.

```
getwd()
```

```
## [1] "/cloud/project/WORKSHEET 7A"
```

```
br_cancer <- read.table("/cloud/project/WORKSHEET 7A/Breast_Cancer.csv", header = FALSE, sep = "," )
br_cancer
```

```
##         V1            V2        V3         V4             V5            V6
## 1       Id CL. thickness Cell size Cell Shape Marg. Adhesion Epith. C.size
## 2  1000025             5         1          1              1             2
## 3  1002945             5         4          4              5             7
## 4  1015425             3         1          1              1             2
## 5  1016277             6         8          8              1             3
## 6  1017023             4         1          1              3             2
## 7  1017122             8        10         10              8             7
## 8  1018099             1         1          1              1             2
## 9  1018561             2         1          2              1             2
## 10 1033078             2         1          1              1             2
## 11 1033078             4         2          1              1             2
## 12 1035283             1         1          1              1             1
## 13 1036172             2         1          1              1             2
## 14 1041801             5         3          3              3             2
## 15 1043999             1         1          1              1             2
## 16 1044572             8         7          5             10             7
## 17 1047630             7         4          6              4             6
## 18 1048672             4         1          1              1             2
## 19 1049815             4         1          1              1             2
## 20 1050670            10         7          7              6             4
## 21 1050718             6         1          1              1             2
## 22 1054590             7         3          2             10             5
## 23 1054593            10         5          5              3             6
## 24 1056784             3         1          1              1             2
## 25 1057013             8         4          5              1             2
## 26 1059552             1         1          1              1             2
## 27 1065726             5         2          3              4             2
## 28 1066373             3         2          1              1             1
## 29 1066979             5         1          1              1             2
## 30 1067444             2         1          1              1             2
## 31 1070935             1         1          3              1             2
## 32 1070935             3         1          1              1             1
## 33 1071760             2         1          1              1             2
## 34 1072179            10         7          7              3             8
## 35 1074610             2         1          1              2             2
## 36 1075123             3         1          2              1             2
## 37 1079304             2         1          1              1             2
## 38 1080185            10        10         10              8             6
## 39 1081791             6         2          1              1             1
## 40 1084584             5         4          4              9             2
## 41 1091262             2         5          3              3             6
## 42 1096800             6         6          6              9             6
## 43 1099510            10         4          3              1             3
## 44 1100524             6        10         10              2             8
## 45 1102573             5         6          5              6            10
## 46 1103608            10        10         10              4             8
## 47 1103722             1         1          1              1             2
## 48 1105257             3         7          7              4             4
## 49 1105524             1         1          1              1             2
## 50 1106095             4         1          1              3             2
## 51
## 52
## 53
## 54
## 55
## 56
## 57
## 58
## 59
## 60
## 61
## 62
## 63
## 64
## 65
## 66
## 67
## 68
## 69
## 70
## 71
## 72
## 73
## 74
## 75
## 76
## 77
## 78
## 79
## 80
## 81
## 82
## 83
## 84
## 85
## 86
## 87
## 88
## 89
## 90
## 91
##              V7           V8              V9     V10       V11
## 1  Bare. Nuclei Bl. Cromatin Normal nucleoli Mitoses     Class
## 2             1            3               1       1    benign
## 3            10            3               2       1    benign
## 4             2            3               1       1    benign
## 5             4            3               7       1    benign
## 6             1            3               1       1    benign
## 7            10            9               7       1 malignant
## 8            10            3               1       1    benign
## 9             1            3               1       1    benign
## 10            1            1               1       5    benign
## 11            1            2               1       1    benign
## 12            1            3               1       1    benign
## 13            1            2               1       1    benign
## 14            3            4               4       2  maligant
## 15            3            3               1       1    benign
## 16            9            5               5       4  maligant
## 17            1            4               3       1  maligant
## 18            1            2               1       1    benign
## 19            1            3               1       1    benign
## 20           10            4               1       2  maligant
## 21            1            3               1       1    benign
## 22           10            5               4       4  maligant
## 23            7            7              10       1  maligant
## 24            1            2               1       1    benign
## 25         <NA>            7               3       1  maligant
## 26            1            3               1       1    benign
## 27            7            3               6       1  maligant
## 28            1            2               1       1    benign
## 29            1            2               1       1    benign
## 30            1            2               1       1    benign
## 31            1            1               1       1    benign
## 32            1            2               1       1    benign
## 33            1            3               1       1    benign
## 34            5            7               4       3  maligant
## 35            1            3               1       1    benign
## 36            1            2               1       1    benign
## 37            1            2               1       1    benign
## 38            1            8               9       1  maligant
## 39            1            7               1       1    benign
## 40           10            5               6       1  maligant
## 41            7            7               5       1  maligant
## 42         <NA>            7               8       1    benign
## 43            3            6               5       2  maligant
## 44           10            7               3       3  malugant
## 45            1            3               1       1  maligant
## 46            1            8              10       1  maligant
## 47            1            2               1       2    benign
## 48            9            4               8       1  maligant
## 49            1            2               1       1    benign
## 50            1            3               1       2    benign
## 51
## 52
## 53
## 54
## 55
## 56
## 57
## 58
## 59
## 60
## 61
## 62
## 63
## 64
## 65
## 66
## 67
## 68
## 69
## 70
## 71
## 72
## 73
## 74
## 75
## 76
## 77
## 78
## 79
## 80
## 81
## 82
## 83
## 84
## 85
## 86
## 87
## 88
## 89
## 90
## 91
```

#c. Compute the descriptive statistics using different packages. Find the values of: #c.1 Standard error of the mean for clump thickness.

```
 Clump<- as.numeric(br_cancer$V2)
```

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

```
num8.n <- length(Clump)
num8.sd <- sd(Clump)
num8.se <- num8.sd /sqrt(Clump)
num8.se
```

```
##  [1] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [26] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [51] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
## [76] NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

#c.2 Coefficient of variability for Marginal Adhesion.

```
   Marginal_Adhesion <- as.numeric(br_cancer$V5)
```

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

```
   stat.desc(Marginal_Adhesion)
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range 
##   49.0000000    0.0000000   42.0000000    1.0000000   10.0000000    9.0000000 
##          sum       median         mean      SE.mean CI.mean.0.95          var 
##  137.0000000    1.0000000    2.7959184    0.3901199    0.7843886    7.4574830 
##      std.dev     coef.var 
##    2.7308392    0.9767235
```

#The result is 0.9767235

#c.3 Number of null values of Bare Nuclei.

```
   Bare_Nuclei <- as.numeric(br_cancer$V7)
```

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

```
   stat.desc( Bare_Nuclei)
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range 
##   47.0000000    0.0000000   44.0000000    1.0000000   10.0000000    9.0000000 
##          sum       median         mean      SE.mean CI.mean.0.95          var 
##  158.0000000    1.0000000    3.3617021    0.5174347    1.0415421   12.5837188 
##      std.dev     coef.var 
##    3.5473538    1.0552255
```

#The result is 0.0000000

#c.4 Mean and standard deviation for Bland Chromatin

```
  Bland_Chromatin <- as.numeric(br_cancer$V8)
```

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

```
  mean(Bland_Chromatin , na.rm = TRUE)
```

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

```
  sd(Bland_Chromatin , na.rm = TRUE)
```

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

```
  stat.desc( Bland_Chromatin)
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range
##   49.0000000    0.0000000   42.0000000    1.0000000    9.0000000    8.0000000
##          sum       median         mean      SE.mean CI.mean.0.95          var
##  188.0000000    3.0000000    3.8367347    0.2978765    0.5989208    4.3477891
##      std.dev     coef.var
##    2.0851353    0.5434661
```

#The mean is 3.836735 #The standard Deviation is 2.085135

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

```
 cell_shape <- as.numeric(br_cancer$V4)
```

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

```
 stat.desc(cell_shape )
```

```
##      nbr.val     nbr.null       nbr.na          min          max        range
##   49.0000000    0.0000000   42.0000000    1.0000000   10.0000000    9.0000000
##          sum       median         mean      SE.mean CI.mean.0.95          var
##  155.0000000    1.0000000    3.1632653    0.4158294    0.8360810    8.4727891
##      std.dev     coef.var
##    2.9108056    0.9201902
```

#The result is 0.8360810

#d. How many attributes?

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

```
describe(br_cancer$V11, na.rm =TRUE)
```

```
## br_cancer$V11
##        n  missing distinct
##       50       41        5
##
## lowest : benign    Class     maligant  malignant malugant
## highest: benign    Class     maligant  malignant malugant
##
## Value         benign     Class  maligant malignant  malugant
## Frequency         31         1        16         1         1
## Proportion      0.62      0.02      0.32      0.02      0.02
```

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

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

```r
#Exporting the data abalone to the Microsoft excel file
library(xlsx)
write.xlsx("abalone","/cloud/project/WORKSHEET 7A/abalone.xlsx")
```