

WORKSHEET -7

Mark Wryzel Macarobo BSIT-2A

12/23/2022

1.

```
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)

DF <- data.frame(Student,PreTest,PostTest)
DF
```

##	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

1.a

```
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)
```

```
## DF
##
## 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
## -----
## PreTest
##      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
## -----
## PostTest
##      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(DF)
```

```
##           Student      PreTest      PostTest
## nbr.val    10.0000000  10.0000000  10.0000000
## nbr.null    0.0000000   0.0000000   0.0000000
```

```
## 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.

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

2.a

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

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

3.

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

3.a

```
out <- data.frame(subjects)
out
```

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

4.

```
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"
```

4.a

```
fctor <- function(state)
  fctor
```

5.

```
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)
```

5.a

```
calcu <- 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
```

```
calcu
```

```
## 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
```

5.b

```
#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
```

6.

```
calcu_ST.n <- length(calcu)
calcu_1.sd <- sd(calcu)
calcu_Final.se <- calcu_1.sd/sqrt(calcu_ST.n)
calcu_Final.se
```

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

6.a

```
#NA
```

6.b

#Results are not available due to some objects are character type, as a result it won't able to get the

7.

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

7.a

```
head_subset <- subset(head, select = "Survived")  
head_subset
```

```
##      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.a

#The dataset s all about Breast Cancer.

8.b

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

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

```
DATA <- read_excel("D:/BSIT2A-CS101/Macarobo_Repo/WORKSHEET -7//breast_cancer.xlsx")
DATA
```

```
## # A tibble: 49 x 11
##       ID CL. thickness~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'
```

```
#8.c
```

```
#8.c.1
```

```
clump <- length(DATA$`CL. thickness`)
clump_A <- sd(DATA$`CL. thickness`)
clump_B <- clump_A/sqrt(DATA$`CL. thickness`)
clump_B
```

```
## [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
```

```
#8.c.2
```

```
coeff <- sd(DATA$`Marg. Adhesion`) / mean(DATA$`Marg. Adhesion`)* 100
coeff
```

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

```
#8.c.4
```

```
mean(DATA$`Bl. Cromatin`)
```

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

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

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

```
#8.c.5
```

```
#Calculate the mean
```

```
calmean <- mean(DATA$`Cell Shape`)  
calmean
```

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

```
#Calculate the standard error of the mean
```

```
SE_M <- length(DATA$`Cell Shape`)
```

```
SD_B <- sd(DATA$`Cell Shape`)
```

```
Ans_1 <- SD_B/sqrt(SE_M)
```

```
Ans_1
```

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

```
#Find the t-score that corresponds to the confidence level
```

```
D = 0.05
```

```
numE = SE_M - 1
```

```
numF = qt(p = D/ 2, df = numE, lower.tail = F)
```

```
numF
```

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

```
#Constructing the confidence interval
```

```
numG <- numF * numE
```

```
#Lower
```

```
numH <- calmean - numG
```

```
#Upper
```

```
numI <- calmean + numG
```

```
c(numH,numI)
```

```
## [1] -93.34720 99.67373
```

```
#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"
```

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

```
P_R <- subset(DATA, Class == "malignant")
P_R
```

```
## # A tibble: 17 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 1100524           6        10        10         2         8 10        7         3
## 15 1102573           5         6         5         6        10 1         3         1
## 16 1103608          10        10        10         4         8 1         8        10
## 17 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

```
17 / 49 * 100
```

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

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

```
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
```

```
library(xlsx)
```

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

```
write.xlsx("abalone", "D:/BSIT2A-CS101/Macarobo_Repo/WORKSHEET -7/abalone.xlsx")
```