

# RWorksheet5

Ariane Soldevilla BSIT 2-B

2023-12-12

#1

```
student_score <- data.frame(  
  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)  
)  
print(student_score)
```

```
##      Student Pre_test Post_test  
## 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

```
library(Hmisc)
```

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

```
library(pastecs)  
summary(student_score)
```

```
##      Student      Pre_test      Post_test  
## Min.   : 1.00   Min.   :47.00   Min.   :56.00  
## 1st Qu.: 3.25   1st Qu.:54.00   1st Qu.:56.75  
## Median : 5.50   Median :56.00   Median :60.50  
## Mean   : 5.50   Mean   :55.70   Mean   :59.70  
## 3rd Qu.: 7.75   3rd Qu.:57.75   3rd Qu.:61.75  
## Max.   :10.00   Max.   :63.00   Max.   :63.00
```

```
stat.desc(student_score)
```

```
##              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

```
fertilizer_levels <- c(10, 10, 10, 20, 20, 50, 10, 20, 10, 50, 20, 50, 20, 10)
ordered_factor_Fertilizer <- factor(fertilizer_levels, levels = c(10, 20, 50), ordered = TRUE)
ordered_factor_Fertilizer
```

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

*#The ordered\_factor\_Fertilizer variable will contain the converted fertilizer levels as an ordered factor*

#3

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

```
## [1] "l" "n" "n" "i" "l" "l" "n" "n" "i" "l"
```

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

#4

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

```
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_factor <- factor(state)
print(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
```

*#The factor function changed the list of state names into a new thing called a factor with seven parts*

#5a

```
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)
```

```
state_factor <- factor(state, levels = c("act", "nsw", "nt", "qld", "sa", "tas", "va"), labels = c("ACT", "NSW", "NT", "QLD", "SA", "TAS", "VA"))
```

```
incmeans <- tapply(incomes, state_factor, mean)
print(incmeans)
```

```
##      ACT      NSW      NT      QLD      SA      TAS      VIC
## 44.50000 57.33333 55.50000 53.60000 55.00000 60.50000      NA
```

#5b

*#The results gives the average income for each state which is inside the "incmeans" vector.*

#6a

```
stdError <- function(x) sqrt(var(x)/length(x))
incster <- tapply(incomes, state_factor, stdError)
incster
```

```
##      ACT      NSW      NT      QLD      SA      TAS      VIC
## 1.500000 4.310195 4.500000 4.106093 2.738613 0.500000      NA
```

#6b

*#Some states have smaller standard errors (like NSW and ACT) have more precise and reliable mean income.*

#7

```
data (Titanic)
Titanic
```

```
## , , Age = Child, Survived = No
```

```
##
```

```
##      Sex
```

```
## Class  Male Female
```

```
## 1st      0      0
```

```
## 2nd      0      0
```

```
## 3rd     35     17
```

```
## Crew     0      0
```

```
##
```

```
## , , Age = Adult, Survived = No
```

```
##
```

```
##      Sex
```

```
## Class  Male Female
```

```
## 1st    118      4
```

```
## 2nd    154     13
```

```
## 3rd    387     89
```

```
## Crew   670      3
```

```
##
```

```
## , , Age = Child, Survived = Yes
```

```
##
```

```
##      Sex
```

```
## Class  Male Female
```

```
## 1st      5      1
```

```
## 2nd     11     13
```

```
## 3rd     13     14
```

```
## Crew     0      0
```

```
##
```

```
## , , Age = Adult, Survived = Yes
```

```
##
```

```
##      Sex
```

```
## Class   Male Female
##   1st    57    140
##   2nd    14     80
##   3rd    75     76
##   Crew  192     20
```

#7a

```
library(dplyr)
```

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

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

## The following objects are masked from 'package:Hmisc':
##
##   src, summarize

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

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

```
titanic <- as.data.frame(Titanic)
survivors <- titanic %>%filter(Survived == "Yes")
head(survivors)
```

```
##   Class   Sex   Age Survived Freq
## 1   1st  Male Child     Yes    5
## 2   2nd  Male Child     Yes   11
## 3   3rd  Male Child     Yes   13
## 4  Crew  Male Child     Yes    0
## 5   1st Female Child     Yes    1
## 6   2nd Female Child     Yes   13
```

```
tail(survivors)
```

```
##   Class   Sex   Age Survived Freq
## 11  3rd  Male Adult     Yes   75
## 12  Crew  Male Adult     Yes  192
## 13  1st Female Adult     Yes  140
## 14  2nd Female Adult     Yes   80
## 15  3rd Female Adult     Yes   76
## 16  Crew Female Adult     Yes   20
```

```
non_survivors <- titanic %>%filter(Survived == "No")
head(non_survivors)
```

```
##   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
```

```
tail(survivors)
```

```
##      Class    Sex    Age Survived Freq
## 11    3rd     Male Adult      Yes   75
## 12   Crew     Male Adult      Yes  192
## 13   1st Female Adult      Yes  140
## 14   2nd Female Adult      Yes   80
## 15   3rd Female Adult      Yes   76
## 16   Crew Female Adult      Yes   20
```

```
#8
```

```
install.packages("readr")
```

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

```
library(readr)
```

```
breastcancer_wisconsin <- read_csv("/cloud/project/RWorksheet#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>
```

```
#8a
```

```
#The dataset is a collection of information about breast cancer tumors reported by Dr. Wolberg. It incl
```

```
#8d.1
```

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

```
sd_clump_thickness <- sd(breastcancer_wisconsin$clump_thickness)
```

```
sample_size <- length(breastcancer_wisconsin$clump_thickness)
```

```
sem_clump_thickness <- sd_clump_thickness / sqrt(sample_size)
cat("Standard Error of the Mean for Clump Thickness:", sem_clump_thickness, "\n")
```

```
## Standard Error of the Mean for Clump Thickness: 0.1065011
```

```
#8d.2
```

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

```
mean_marginal_adhesion <- mean(breastcancer_wisconsin$marginal_adhesion)
sd_marginal_adhesion <- sd(breastcancer_wisconsin$marginal_adhesion)
cv_marginal_adhesion <- (sd_marginal_adhesion / mean_marginal_adhesion) * 100
cat("Coefficient of Variation for Marginal Adhesion:", cv_marginal_adhesion, "%\n")
```

```
## Coefficient of Variation for Marginal Adhesion: 101.7283 %
```

```
#8d.3
```

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

```
num_null_bare_nuclei <- sum(is.na(breastcancer_wisconsin$bare_nucleoli))
cat("Number of null values for Bare Nuclei:", num_null_bare_nuclei, "\n")
```

```
## Number of null values for Bare Nuclei: 15
```

```
#8d.4
```

```
#d.4 Mean and standard deviation for Bland Chromatin
```

```
mean_bland_chromatin <- mean(breastcancer_wisconsin$bland_chromatin, na.rm = TRUE)
sd_bland_chromatin <- sd(breastcancer_wisconsin$bland_chromatin, na.rm = TRUE)
cat("Mean for Bland Chromatin:", mean_bland_chromatin, "\n")
```

```
## Mean for Bland Chromatin: 3.437768
```

```
cat("Standard Deviation for Bland Chromatin:", sd_bland_chromatin, "\n")
```

```
## Standard Deviation for Bland Chromatin: 2.438364
```

```
#8d.5
```

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

```
mean_value <- mean(breastcancer_wisconsin$shape_uniformity, na.rm = TRUE)
se <- sd(breastcancer_wisconsin$shape_uniformity, na.rm = TRUE) / sqrt(length(breastcancer_wisconsin$shape_uniformity))
confidence_level <- 0.95
```

```
margin_of_error <- qt((1 + confidence_level) / 2, df = length(breastcancer_wisconsin$shape_uniformity))
```

```
confidence_interval <- c(mean_value - margin_of_error, mean_value + margin_of_error)
cat("Confidence Interval (", confidence_level * 100, "%) :", confidence_interval, "\n")
```

```
## Confidence Interval ( 95 %) : 2.986741 3.428138
```

```
#8d
```

```
attributes <- names(breastcancer_wisconsin)
attributes
```

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

```
#8e
```

```

malignant_count <- sum(breastcancer_wisconsin$class == 4)
malignant_percentage <- (malignant_count / nrow(breastcancer_wisconsin)) * 100
cat("Percentage of respondents who are malignant:", malignant_percentage, "%\n")

## Percentage of respondents who are malignant: 34.47783 %

#9

install.packages("AppliedPredictiveModeling")

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

library("AppliedPredictiveModeling")
library(readr)
abalone <- read_csv("/cloud/project/RWorksheet#6/abalone.csv")

## Rows: 4177 Columns: 9

## -- Column specification -----
## Delimiter: ","
## chr (1): Sex
## dbl (8): Length, Diameter, Height, Whole weight, Shucked weight, Viscera wei...
##
## 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.

head(abalone)

## # A tibble: 6 x 9
##   Sex   Length Diameter Height `Whole weight` `Shucked weight` `Viscera weight`
##   <chr>   <dbl>   <dbl>   <dbl>         <dbl>         <dbl>         <dbl>
## 1 M      0.455    0.365  0.095         0.514         0.224         0.101
## 2 M      0.35    0.265  0.09         0.226         0.0995        0.0485
## 3 F      0.53    0.42   0.135        0.677         0.256         0.142
## 4 M      0.44    0.365  0.125        0.516         0.216         0.114
## 5 I      0.33    0.255  0.08         0.205         0.0895        0.0395
## 6 I      0.425    0.3    0.095        0.352         0.141         0.0775
## # i 2 more variables: `Shell weight` <dbl>, Rings <dbl>

summary(abalone)

##           Sex           Length           Diameter           Height
## Length:4177   Min.    :0.075   Min.    :0.0550   Min.    :0.0000
## Class :character 1st Qu.:0.450   1st Qu.:0.3500   1st Qu.:0.1150
## Mode  :character Median :0.545   Median :0.4250   Median :0.1400
##              Mean  :0.524   Mean  :0.4079   Mean  :0.1395
##              3rd Qu.:0.615   3rd Qu.:0.4800   3rd Qu.:0.1650
##              Max.  :0.815   Max.  :0.6500   Max.  :1.1300
## Whole weight Shucked weight Viscera weight Shell weight
## Min.    :0.0020 Min.    :0.0010 Min.    :0.0005 Min.    :0.0015
## 1st Qu.:0.4415 1st Qu.:0.1860 1st Qu.:0.0935 1st Qu.:0.1300
## Median :0.7995 Median :0.3360 Median :0.1710 Median :0.2340
## Mean  :0.8287 Mean  :0.3594 Mean  :0.1806 Mean  :0.2388
## 3rd Qu.:1.1530 3rd Qu.:0.5020 3rd Qu.:0.2530 3rd Qu.:0.3290
## Max.  :2.8255 Max.  :1.4880 Max.  :0.7600 Max.  :1.0050
##           Rings
## Min.    : 1.000

```

```
## 1st Qu.: 8.000
## Median : 9.000
## Mean   : 9.934
## 3rd Qu.:11.000
## Max.    :29.000
```