

Rworksheet_Esmalla#6

Janessa Marie Esmalla

2023-12-17

```
library(Hmisc)

##
## Attaching package: 'Hmisc'
## The following objects are masked from 'package:base':
##
##      format.pval, units
library(pastecs)
library(readr)
library("AppliedPredictiveModeling")
library(xlsx)
```

1. Create a data frame for the table below

```
Student <- seq(1:10)
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)

studDF <- data.frame(Student, Pre_Test, Post_Test)
studDF
```

```
##      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
```

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

```
describe(studDF)

## studDF
##
## 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
## -----
## Pre_Test
##      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
## -----
## Post_Test
##      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
## -----
```

```
stat.desc(studDF)
```

```
##      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.

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

- a. Write the codes and describe the result.

```
agriculture <- sort(fertilizer, decreasing = FALSE)
agriculture
```

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

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

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

- a. What is the best way to represent this in R? #Data frame

```
study <- data.frame(unli)
study
```

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

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.

```
fx3 <- factor(state)
fx3
```

```
## [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
```

5. From 4 - continuation: • Suppose we have the incomes of the same tax accountants in another vector (insuitably 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()`:

```
revenue <- 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
```

```
revenue
```

```
## 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
```

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
```

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

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

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

```
## Warning in var(x): NAs introduced by coercion
```

```
##      unli  
## unli   NA
```

```
incster <- tapply(incomes, state, stdError)
```

b. Interpret the result. #It is not available due to character type variables that are in the data.

7. Use the titanic dataset.

```
data("Titanic")
```

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

a. subset the titanic dataset of those who survived and not survived. Show the codes and its result.

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

```
##      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 data is about breast cancer in Wisconsin. The dataset contains the sample of reports from Dr. Wolberg's clinical cases.

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

```
library("readxl")

df <- read_csv("BreastCancer.csv")

## Rows: 90 Columns: 11
## -- Column specification -----
## Delimiter: ","
## chr (1): Class
## dbl (10): Id, CL. thickness, Cell size, Cell Shape, Marg. Adhesion, Epith. C...
##
## 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.
df

## # A tibble: 90 x 11
##       Id `CL. thickness` `Cell size` `Cell Shape` `Marg. 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 80 more rows
## # i 6 more variables: `Epith. C.size` <dbl>, `Bare. Nuclei` <dbl>,
## #   `Bl. Chromatin` <dbl>, `Normal nucleoli` <dbl>, Mitoses <dbl>, Class <chr>
```

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

```
thick <- length(df$`CL. thickness`)
thic1 <- sd(df$`CL. thickness`)
thic2 <- thic1/sqrt(df$`CL. thickness`)
thic2
```

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

c.2 Coefficient of variability for Marginal Adhesion.

```
variants <- sd(df$`Marg. Adhesion`) / mean(df$`Marg. Adhesion`)* 100
variants
```

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

c.3 Number of null values of Bare Nuclei.

```
nuclei <- subset(df,`Bare. Nuclei` == "NA")
nuclei
```

```
## # A tibble: 0 x 11
## # i 11 variables: Id <dbl>, CL. thickness <dbl>, Cell size <dbl>,
## #   Cell Shape <dbl>, Marg. Adhesion <dbl>, Epith. C.size <dbl>,
## #   Bare. Nuclei <dbl>, Bl. Cromatin <dbl>, Normal nucleoli <dbl>,
## #   Mitoses <dbl>, Class <chr>
```

c.4 Mean and standard deviation for Bland Chromatin

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

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

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

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

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

```
uni <- mean(df$`Cell Shape`)
uni
```

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

Calculate the standard error of the mean

```
standardE <- length(df$`Cell Shape`)
standardE1 <- sd(df$`Cell Shape`)
standardE2 <- standardE1/sqrt(standardE)
standardE2
```

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

Find the t-score that corresponds to the confidence level

```
tscore = 0.05
tsc = standardE1 - 1
confi = qt(p = tscore/ 2, df = tsc,lower.tail = F)
confi
```

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

Constructing the confidence interval

```
confiE <- confi * standardE2
```

Lower & Upper

```
low <- uni - confiE
```

```
up <- uni + confiE
```

```
LowUp <- c(low, up)
```

d. How many attributes?

```
attri <- attributes(df)
```

```
attri
```

```
## $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 50
## [51] 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] 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90
##
## $names
## [1] "Id" "CL. thickness" "Cell size" "Cell Shape"
## [5] "Marg. Adhesion" "Epith. C.size" "Bare. Nuclei" "Bl. Cromatin"
## [9] "Normal nucleoli" "Mitoses" "Class"
##
## $spec
## cols(
## Id = col_double(),
## `CL. thickness` = col_double(),
## `Cell size` = col_double(),
## `Cell Shape` = col_double(),
## `Marg. Adhesion` = col_double(),
## `Epith. C.size` = col_double(),
## `Bare. Nuclei` = col_double(),
## `Bl. Cromatin` = col_double(),
## `Normal nucleoli` = col_double(),
## Mitoses = col_double(),
## Class = col_character()
## )
##
## $problems
## <pointer: 0x558e3258b150>
##
## $class
## [1] "spec_tbl_df" "tbl_df" "tbl" "data.frame"
```

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

```
perce <- subset(df, Class == "malignant")
perce
```

```
## # A tibble: 18 x 11
## Id `CL. thickness` `Cell size` `Cell Shape` `Marg. Adhesion`
## <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 1017122 8 10 10 8
## 2 1041801 5 3 3 3
```



```
## 3 1044572      8      7      5      10
## 4 1047630      7      4      6      4
## 5 1050670     10      7      7      6
## 6 1054590      7      3      2     10
## 7 1054593     10      5      5      3
## 8 1057013      8      4      5      1
## 9 1065726      5      2      3      4
## 10 1072179     10      7      7      3
## 11 1080185     10     10     10      8
## 12 1084584      5      4      4      9
## 13 1091262      2      5      3      3
## 14 1099510     10      4      3      1
## 15 1100524      6     10     10      2
## 16 1102573      5      6      5      6
## 17 1103608     10     10     10      4
## 18 1105257      3      7      7      4
```

```
## # i 6 more variables: `Epith. C.size` <dbl>, `Bare. Nuclei` <dbl>,
## #   `Bl. Cromatin` <dbl>, `Normal nucleoli` <dbl>, Mitoses <dbl>, Class <chr>
```

#In 49 respondents, there are 18 that are malignant.

```
malig <- 17 / 49 * 100
malig
```

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

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

```
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
```

Export abalone

```
install.packages("xlsxjars")
```

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

```
library(xlsx)
```

```
write.xlsx(abalone, "abalone.xlsx")
```