

RWorksheet7a

Frias, Dazzele Mae

2022-12-31

```

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)

#1. Create a data frame for the table below

Student <- seq(1:10)
Pre_Test <- c(ss,54,47,57,51,61,57,54,43,58)
Post_Test <- c(61,60,56,63,56,63,59,56,62,61)

record <- data.frame(Student,Pre_Test,Post_Test)

record



| Student | Pre_Test | Post_Test |
|---------|----------|-----------|
| 1       | 55       | 61        |
| 2       | 54       | 60        |
| 3       | 47       | 56        |
| 4       | 57       | 63        |
| 5       | 51       | 56        |
| 6       | 61       | 63        |
| 7       | 57       | 59        |
| 8       | 54       | 56        |
| 9       | 63       | 62        |
| 10      | 58       | 61        |



1-10 of 10 rows

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

describe(record)

## record
##
## 3 Variables      10 Observations
## -----
## Student
##      n missing distinct   Info    Mean    Gnd      .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    Gnd
## 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 3 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     3.111
##
## 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(record)
```

	Student <dbl>	Pre_Test <dbl>	Post_Test <dbl>
nbr.val	10.00000000	10.00000000	10.00000000
nbr.null	0.00000000	0.00000000	0.00000000
nbr.na	0.00000000	0.00000000	0.00000000
min	1.00000000	47.00000000	56.00000000
max	10.00000000	63.00000000	63.00000000
range	9.00000000	16.00000000	7.00000000
sum	55.00000000	557.00000000	597.00000000
median	5.50000000	56.00000000	60.50000000
mean	5.50000000	55.70000000	59.70000000
SE.mean	0.9574271	1.46855938	0.89504811

1-10 of 14 rows

[Previous](#) [1](#) [2](#) [Next](#)

```
#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, 20,20,50,10,20,10,50,20,10,20,10.
```

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

```
##a. Write the codes and describe the result.

ordr <- factor(Fert_Lvl, ordered = TRUE)

ordr

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

```
#3. Abdul hassan, president of Floor Coverings Unlimited, has asked you to study
#the exercise levels undertaken by 10 subjects were â-1â , â-4â , â-4â , â-1â ,
#â-1â , â-4â , â-4â , â-1â , â-1â , =none, i=light, i=intense

subs <- c("1","n","n","1","1","1","n","n","1","1")

#a. What is the best way to represent this in R?
#Dataframe
```

```
mine <- data.frame(subs)

mine
```

```
subs
<data>
```

```
#The individual state of origin is specified by a character vector of state mnemonics
#BS:

state_organ <- c("tas", "sa", "qld", "nsw", "nt", "wa", "wa", "qld",
"vic", "nsw", "vic", "qld", "qld", "nsw", "act", "nsw",
"wa", "vic", "qld", "nsw", "nsw", "wa", "sa", "act", "nsw",
"vic", "vic", "act")

state_organ
```

```
## [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"
```

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

```
han <- factor(state.orgn)
```

```
han

## [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(han)
```

```
# [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

income <- c(50, 49, 40, 61, 54, 60, 59, 54,
            62, 69, 78, 42, 56, 81, 61, 61, 58, 51, 48
```

```
65, 49, 49, 41, 48, 52, 46, 59, 46, 58, 43)

##. Calculate the sample mean income for each state we can now use the special
#function tapply():

    stt_income <- tapply(income,state_organ, mean )

    stt_income
```

```
##      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
#44.50000 57.33333 55.50000
#      qld      tas
#53.60000 55.00000 60.50000
```

```
# vic      md
#56.00000 52.25000
#these are the mean of Income

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

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

```
##9. What is the standard error? Write the codes.
##10. Interpret the result. It tells the sample mean of income

istdr <- tapply(income, state_organ, stdError)

istdr

##          act          hsw          nt          qld          sa          tas          vic          wa
## 1.5000000 4.3101095 4.5000000 4.1060903 2.738613 0.5000000 5.244044 2.657536
```

```
#7. Use the Titanic dataset.

data("Titanic")

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

```
#8. Subset the titanic dataset of those who survived and not survived. Show the
#codes and its result.
```

```

  srvv <- subset(titanic, Survived == "yes")

  srvv

```

	Class <fct>	Sex <fct>	Age <dbl>	Survived <fct>	Freq <dbl>
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

1-10 of 16 rows

Previous 1 2 Next

```
yours <- subset(titanic, Survived == "No")
```

yours					
	Class <fct>	Sex <fct>	Age <fct>	Survived <fct>	Freq <dbl>
1	1st	Male	Child	No	
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
1-10 of 10 rows					Previous 1 2 Next

#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 s all about Breast Cancer.

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

library("readxl")

brst_cncr <- read_excel("C:/Users/User/Downloads/Breast_Cancer.xlsx")
```

brst_cncr							
Id	CL. thickness	Cell size	Cell Shape	Marg. Adhesion	Epith. C.size	Bare. Nuclei	Bl. Cromatin
100025	5	1	1	1	2	1	3
1002945	5	4	4	5	7	10	3
1000195	5	4	4	5	7	10	3

1015425	3	1	1	1	2 2	3
1016277	6	8	8	1	3 4	3
1017023	4	1	1	3	2 1	3
1017122	8	10	10	8	7 10	9
1018099	1	1	1	1	2 10	3
1018561	2	1	2	1	2 1	3

```
1033078      2      1      1      1      2 1      1
1033078      4      2      1      1      2 1      2
```

1-10 of 49 rows | 1-8 of 11 columns

Previous [1](#) [2](#) [3](#) [4](#) [5](#) [Next](#)

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

sumSep <- c(length(best.one$Cl), thickness)
```

```
numBc1.sd <- sd(brst_cncr$CL_thickness')
numBc1.se <- sqrt(numBc1.sd/sqrt(brst_cncr$CL_thickness'))
numBc1.se
```

```
## [1] 2.1812754 2.1812754 1.6541194 1.1696391 1.4325995 1.0129371 2.8658189
## [2] 0.0258743 0.0258743 1.4325995 2.8658189 0.0258743 2.1812754 2.8658189
## [5] 1.0129371 1.0828754 1.4325995 1.4325995 0.9659895 1.1696391 1.0828754
## [22] 0.9659895 1.6541194 1.0129371 2.8658189 1.2812754 1.6541194 1.2812754
```

```
## [29] 2.0258743 2.0650818 1.9549275 2.0258743 0.9059985 2.0258743 1.9549275  
## [30] 2.0258743 0.9059985 1.9063919 1.2812754 2.0258743 1.9063919 0.9059985  
## [43] 1.1696391 1.2812754 0.9059985 2.0650818 1.6541194 2.0650818 1.4325695
```

#C.2 Coefficient of variability for Marginal Adhesion.

```
Marginal_Adh <- as.numeric(brst_cmc$V5)
```

Warning: Warning re uninitialized column 'V5':
#> Warning: some elements are NA

```
## Warning: unknown or uninitialised column: vs .

stat.desc(Marginal_Adh)

## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf
```

```
## Warning in qt((0.5 + p/2), (Nbrval - 1)): NANS produced
```

##	nbr.val	nbr.null	nbr.na	min	max	range
##	sum	0	0	Inf	-Inf	-Inf
##	0	median	mean	SE, mean	CI, mean	0.95 var
##	0	NA	NaN	NA	NaN	NA
##	std.dev	coef.var				
##		NA				

```
#C: 3 Number of null values of Bare Nuclei.

Br_Nuclei <- as.numeric(brst_cncr$V7)

## Warning: Unknown or uninitialised column: 'V7'.

plot(brst_cncr$Br_Nuclei)
```

```
## Warning in min(x): no non-missing arguments to min; returning Inf
```

```
## Warning in max(x): no non-missing arguments to max; returning -Inf
```

```
## Warning in qt((0.5 + p/2), (Nbrval - 1)): NaNs produced
```

##	nbr.val	nbr.muli	nbr.na	min	max	range
##			0	Inf	-Inf	
##	sum	median	mean	SE.mean	CI.mean.0.95	var
##	0	NA	NaN	NA	NaN	NA
##	std.dev	coef.var				
##	NA	NA				

#C.4 Mean and standard deviation for Bland Chromatin

```
b1_Chromatin <- as.numeric(brst_cncr[V8])

## Warning: Unknown or uninitialised column: `V8`.

mean(b1_Chromatin , na.rm = TRUE)
```

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

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

```
stat_desc( bi_Chromatin)
```

```
## Warning in min(x): no non-missing arguments to min; returning Inf

## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in qt((0.5 + p/2), (Nbrval - 1)): NaNs produced
```

##	nbr_val	nbr_null	nbr_na	min	max	range
##	1	0	0	0.000000	0.000000	0.000000
##	2	0	0	0.000000	0.000000	0.000000
##	3	0	0	0.000000	0.000000	0.000000
##	4	0	0	0.000000	0.000000	0.000000
##	5	0	0	0.000000	0.000000	0.000000
##	6	0	0	0.000000	0.000000	0.000000
##	7	0	0	0.000000	0.000000	0.000000
##	8	0	0	0.000000	0.000000	0.000000
##	9	0	0	0.000000	0.000000	0.000000
##	10	0	0	0.000000	0.000000	0.000000
##	11	0	0	0.000000	0.000000	0.000000
##	12	0	0	0.000000	0.000000	0.000000
##	13	0	0	0.000000	0.000000	0.000000
##	14	0	0	0.000000	0.000000	0.000000
##	15	0	0	0.000000	0.000000	0.000000
##	16	0	0	0.000000	0.000000	0.000000
##	17	0	0	0.000000	0.000000	0.000000
##	18	0	0	0.000000	0.000000	0.000000
##	19	0	0	0.000000	0.000000	0.000000
##	20	0	0	0.000000	0.000000	0.000000
##	21	0	0	0.000000	0.000000	0.000000
##	22	0	0	0.000000	0.000000	0.000000
##	23	0	0	0.000000	0.000000	0.000000
##	24	0	0	0.000000	0.000000	0.000000
##	25	0	0	0.000000	0.000000	0.000000
##	26	0	0	0.000000	0.000000	0.000000
##	27	0	0	0.000000	0.000000	0.000000
##	28	0	0	0.000000	0.000000	0.000000
##	29	0	0	0.000000	0.000000	0.000000
##	30	0	0	0.000000	0.000000	0.000000
##	31	0	0	0.000000	0.000000	0.000000
##	32	0	0	0.000000	0.000000	0.000000
##	33	0	0	0.000000	0.000000	0.000000
##	34	0	0	0.000000	0.000000	0.000000
##	35	0	0	0.000000	0.000000	0.000000
##	36	0	0	0.000000	0.000000	0.000000
##	37	0	0	0.000000	0.000000	0.000000
##	38	0	0	0.000000	0.000000	0.000000
##	39	0	0	0.000000	0.000000	0.000000
##	40	0	0	0.000000	0.000000	0.000000
##	41	0	0	0.000000	0.000000	0.000000
##	42	0	0	0.000000	0.000000	0.000000
##	43	0	0	0.000000	0.000000	0.000000
##	44	0	0	0.000000	0.000000	0.000000
##	45	0	0	0.000000	0.000000	0.000000
##	46	0	0	0.000000	0.000000	0.000000
##	47	0	0	0.000000	0.000000	0.000000
##	48	0	0	0.000000	0.000000	0.000000
##	49	0	0	0.000000	0.000000	0.000000
##	50	0	0	0.000000	0.000000	0.000000
##	51	0	0	0.000000	0.000000	0.0

```
##          sum          0          0          Inf          -Inf          -Inf
##          sum          median      mean      SE.mean CI.mean.0.95      var
##          0          NA          NaN          NA          NaN          NA
##          std.dev      coef.var
##          NA          NA
```

```
c11_shape <- as.numeric(nrow(cmc11a47))

## Warning: Unknown or uninitialised column: 'V4'.

stat_desc(c11_shape)

## Warning in min(x): no non-missing arguments to min; returning Inf
```

```
## Warning in max(x): no non-missing arguments to max; returning -Inf

## Warning in qt((0.5 + p/2), (Nbrval - 1)): NaNs produced
```

##	nbr.val	nbr.null	nbr.na	min	max	range
##	0	0	0	Inf	-Inf	-Inf
##	sum	median	mean	SE.mean	CI.mean.0.95	var

```
##          std.dev      coef.var      rsn      na      rsn      na
##          NA          NA          NA          NA          NA          NA

#d. How many attributes?
attributes(brst_cncr)

## $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"

#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	ShellWeight	Rings
1 M	0.455	0.365	0.095	0.5140	0.2245	0.1010	0.150	15
2 M	0.350	0.265	0.090	0.2255	0.0995	0.0485	0.070	7
3 F	0.530	0.420	0.135	0.6770	0.2565	0.1415	0.210	9
4 M	0.440	0.365	0.125	0.5160	0.2155	0.1140	0.155	10

51	0.330	0.255	0.080	0.2050	0.0895	0.0395	0.055	7
61	0.425	0.300	0.095	0.3515	0.1410	0.0775	0.120	8

6 rows

summary(abalone)

#	Type	LongestShell	Diameter	Height	WholeWeight
---	------	--------------	----------	--------	-------------

```
## #1367 Min. -0.675   1st Qu. -0.6550   Median -0.6000   Mean -0.6020
## #1368 Max. -0.450   3rd Qu. -0.3500   1st Qu. -0.3500   1st Qu. -0.4415
## #W1528 Median -0.545   Median -0.4920   Median -0.4390   Median -0.7995
##
##      Mean -0.524   Mean -0.4079   Mean -0.3395   Mean -0.8287
##
##      3rd Qu. -0.615   3rd Qu. -0.4800   3rd Qu. -0.3650   3rd Qu. -1.1530
##
##      Max. -0.815   Max. -0.6500   Max. -0.1300   Max. -1.8255
##
## ShuckedWeight Visceraweight Shellweight Rings
##
## Min. -0.9930   Min. -0.0905   Min. -0.0915   Min. 1.0000
## 1st Qu. -0.2660   1st Qu. -0.0935   1st Qu. -0.1000   1st Qu. 8.0000
##
## Median -0.3360   Median -0.0700   Median -0.2340   Median -8.0000
```

```
## 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.:111.000
## Max. :1.4880 Max. :0.7600 Max. :1.0050 Max. :29.000
```

```
library(xlsx)
library(latexpdf)
```