RWorksheet_5.rmd

Darlene Erl Lapso

2023-12-15

---

1. Create a data frame for the table below. Show your solution.

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

```
library(Hmisc)
```

```
##
## Attaching package: 'Hmisc'
```

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

```
library(pastecs)
studentData<- data.frame(
  Student <- c(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))

colnames(studentData) <- c("Student", "Pre-Test","Post-Test")
studentData
```

```
##    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
```

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.

```
fertLevels <- c(10,10,10, 20,20,50,10,20,10,50,20,50,20,10.)
ordered(fertLevels)
```

```
##  [1] 10 10 10 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 "l", "n", "n", "i", "l" , "l", "n", "n", "i", "l" ; n=none, l=light, i=intense.

```
exLevels <- c("l", "n", "n", "i", "l", "l", "n", "n", "i", "l")
factor_exLevels<- factor(exLevels)
levels(factor_exLevels) <- c("none","light","intense")
factor_exLevels
```

```
##  [1] light   intense intense none    light   light   intense intense none
## [10] light
## Levels: none light intense
```

4a.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")
statef<-factor(state)
levels(statef)
```

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

```
statef
```

```
##  [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 (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)

incmeans <- tapply(incomes, statef, 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
```

```
data <- data.frame(State = statef, Income = incomes)
data
```

```
##    State Income
## 1    tas     60
## 2     sa     49
## 3    qld     40
## 4    nsw     61
## 5    nsw     64
## 6     nt     60
## 7     wa     59
## 8     wa     54
## 9    qld     62
## 10   vic     69
## 11   nsw     70
## 12   vic     42
## 13   qld     56
## 14   qld     61
## 15    sa     61
## 16   tas     61
## 17    sa     58
## 18    nt     51
## 19    wa     48
## 20   vic     65
## 21   qld     49
## 22   nsw     49
## 23   nsw     41
## 24    wa     48
## 25    sa     52
## 26   act     46
## 27   nsw     59
## 28   vic     46
## 29   vic     58
## 30   act     43
```

b. Copy the results and interpret act nsw nt qld sa tas 44.50000 57.33333 55.50000 53.60000 55.00000 60.50000 vic wa 56.00000 52.25000

This result shows the average income of the different/individual states in Australia.

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

```
stdError <- function(x) sqrt(var(x) / length(x))
incster <- tapply(incomes, statef, 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 standard error of the state incomes mean of different states,

#On this dataset, the "Titanic" data is not available on the version of my RStudio, #therefore I downloaded a .csv file from the internet. Link where I get the 'Titanic.csv': https://github.com/datasciencedojo/datasets/blob/master/titanic.csv

7. Use the titanic dataset.

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

```
library(readr)
titanic <- read_csv("C:/Users/steve/Documents/lapso-worksheetactivity/worksheet#5/titanic.csv")
```

```
## Rows: 891 Columns: 12
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (5): Name, Sex, Ticket, Cabin, Embarked
## dbl (7): PassengerId, Survived, Pclass, Age, SibSp, Parch, Fare
##
## 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.
```

```
View(titanic)
```

```
survived_passengers <- subset(titanic, Survived == "1")
head(survived_passengers)
```

```
## # A tibble: 6 x 12
##   PassengerId Survived Pclass Name    Sex     Age SibSp Parch Ticket  Fare Cabin
##         <dbl>    <dbl>  <dbl> <chr>   <chr> <dbl> <dbl> <dbl> <chr>  <dbl> <chr>
## 1           2        1      1 Cuming~ fema~    38     1     0 PC 17~ 71.3  C85
## 2           3        1      3 Heikki~ fema~    26     0     0 STON/~  7.92 <NA>
## 3           4        1      1 Futrel~ fema~    35     1     0 113803 53.1  C123
## 4           9        1      3 Johnso~ fema~    27     0     2 347742 11.1  <NA>
## 5          10        1      2 Nasser~ fema~    14     1     0 237736 30.1  <NA>
## 6          11        1      3 Sandst~ fema~     4     1     1 PP 95~ 16.7  G6
## # i 1 more variable: Embarked <chr>
```

```
not_survived_passengers <- subset(titanic, Survived == 0)
head(not_survived_passengers)
```

```
## # A tibble: 6 x 12
##   PassengerId Survived Pclass Name    Sex     Age SibSp Parch Ticket  Fare Cabin
##         <dbl>    <dbl>  <dbl> <chr>   <chr> <dbl> <dbl> <dbl> <chr>  <dbl> <chr>
## 1           1        0      3 Braund~ male     22     1     0 A/5 2~  7.25 <NA>
## 2           5        0      3 Allen,~ male     35     0     0 373450  8.05 <NA>
## 3           6        0      3 Moran,~ male     NA     0     0 330877  8.46 <NA>
## 4           7        0      1 McCart~ male     54     0     0 17463  51.9  E46
## 5           8        0      3 Palsso~ male      2     3     1 349909 21.1  <NA>
## 6          13        0      3 Saunde~ male     20     0     0 A/5. ~  8.05 <NA>
## # i 1 more variable: Embarked <chr>
```

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 chronologihttps://drive.google.com/file/d/16MFLoehCgx2M

```
breastCancer <- read_csv("C:/Users/steve/Documents/lapso-worksheetactivity/worksheet#5/breastcancer_wis
```

```
## 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
```

```
## # 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>
```

```
str(breastCancer)
```

```
## spc_tbl_ [699 x 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id               : num [1:699] 1000025 1002945 1015425 1016277 1017023 ...
##  $ clump_thickness  : num [1:699] 5 5 3 6 4 8 1 2 2 4 ...
##  $ size_uniformity  : num [1:699] 1 4 1 8 1 10 1 1 1 2 ...
##  $ shape_uniformity : num [1:699] 1 4 1 8 1 10 1 2 1 1 ...
##  $ marginal_adhesion: num [1:699] 1 5 1 1 3 8 1 1 1 1 ...
##  $ epithelial_size  : num [1:699] 2 7 2 3 2 7 2 2 2 2 ...
##  $ bare_nucleoli    : chr [1:699] "1" "10" "2" "4" ...
##  $ bland_chromatin  : num [1:699] 3 3 3 3 3 9 3 3 1 2 ...
##  $ normal_nucleoli  : num [1:699] 1 2 1 7 1 7 1 1 1 1 ...
##  $ mitoses          : num [1:699] 1 1 1 1 1 1 1 1 5 1 ...
##  $ class            : num [1:699] 2 2 2 2 2 4 2 2 2 2 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_double(),
##   ..   clump_thickness = col_double(),
##   ..   size_uniformity = col_double(),
```

```
##   ..   shape_uniformity = col_double(),
##   ..   marginal_adhesion = col_double(),
##   ..   epithelial_size = col_double(),
##   ..   bare_nucleoli = col_character(),
##   ..   bland_chromatin = col_double(),
##   ..   normal_nucleoli = col_double(),
##   ..   mitoses = col_double(),
##   ..   class = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

a. describe what is the dataset all about.

The 'BreastCancer.csv' file shows the different quantity form and what can be found on a cyst of a cancer patient.

d. Compute the descriptive statistics using different packages. Find the values of:

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

```
clump_thickness <- breastCancer$clump_thickness

stEclump_thickness <- sd(clump_thickness) / sqrt(length(clump_thickness))
stEclump_thickness
```

```
## [1] 0.1065011
```

d.2 Coefficient of variability for Marginal Adhesion.

```
marginal_adhesion <- breastCancer$marginal_adhesion

coefVar_margAd <- (sd(marginal_adhesion) / mean(marginal_adhesion)) * 100
coefVar_margAd
```

```
## [1] 101.7283
```

d.3 Number of null values of Bare Nuclei.

```
bareNucNULLS <- sum(is.na(breastCancer$bare_nucleoli))
bareNucNULLS
```

```
## [1] 15
```

#[1] 15 I do not know why the result is 15, when I check the data, the 'bare_nucleoli' does #not contain any NULL values.

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

```
blandChrom <- breastCancer$bland_chromatin
Mean_blandChrom <- mean(blandChrom)
Mean_blandChrom
```

```
## [1] 3.437768
```

#standard dev

```
stdDev_blandChrom <- sd(blandChrom)
stdDev_blandChrom
```

```
## [1] 2.438364
```

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

```
shapeUniformity <- breastCancer$shape_uniformity

confidenceInt_shapeUniformity <- t.test(shapeUniformity)$conf.int
confidenceInt_shapeUniformity
```

```
## [1] 2.986741 3.428138
## attr(,"conf.level")
## [1] 0.95
```

d. How many attributes?

```
str(breastCancer) #one of my knowledge to get the attributes is by using the str() function.
```

```
## spc_tbl_ [699 x 11] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ id               : num [1:699] 1000025 1002945 1015425 1016277 1017023 ...
##  $ clump_thickness  : num [1:699] 5 5 3 6 4 8 1 2 2 4 ...
##  $ size_uniformity  : num [1:699] 1 4 1 8 1 10 1 1 1 2 ...
##  $ shape_uniformity : num [1:699] 1 4 1 8 1 10 1 2 1 1 ...
##  $ marginal_adhesion: num [1:699] 1 5 1 1 3 8 1 1 1 1 ...
##  $ epithelial_size  : num [1:699] 2 7 2 3 2 7 2 2 2 2 ...
##  $ bare_nucleoli    : chr [1:699] "1" "10" "2" "4" ...
##  $ bland_chromatin  : num [1:699] 3 3 3 3 3 9 3 3 1 2 ...
##  $ normal_nucleoli  : num [1:699] 1 2 1 7 1 7 1 1 1 1 ...
##  $ mitoses          : num [1:699] 1 1 1 1 1 1 1 1 5 1 ...
##  $ class            : num [1:699] 2 2 2 2 2 4 2 2 2 2 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   id = col_double(),
##   ..   clump_thickness = col_double(),
##   ..   size_uniformity = col_double(),
##   ..   shape_uniformity = col_double(),
##   ..   marginal_adhesion = col_double(),
##   ..   epithelial_size = col_double(),
##   ..   bare_nucleoli = col_character(),
##   ..   bland_chromatin = col_double(),
##   ..   normal_nucleoli = col_double(),
##   ..   mitoses = col_double(),
##   ..   class = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

#or

```
BC_attributes <- ncol(breastCancer)
BC_attributes
```

```
## [1] 11
```

9. Export the data abalone to the Microsoft excel file. Copy the codes. install.packages("AppliedPredictiveModeling") library("AppliedPredictiveModeling") view(abalone) head(abalone) summary(abalone)

#As I install the 'install.packages("AppliedPredictingModeling")' it says that #the package is not available on the R Version I have. Therefore, once again, I downloaded #a file from the internet so I can gather data, as what the worksheet asking for our activity.

```
library(readr)
abalone <- read_csv("C:/Users/steve/Documents/lapso-worksheetactivity/worksheet#5/abalone.csv")
```

```
## Rows: 4176 Columns: 9
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (1): M
## dbl (8): 0.455, 0.365, 0.095, 0.514, 0.2245, 0.101, 0.15, 15
##
## 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
##   M     `0.455` `0.365` `0.095` `0.514` `0.2245` `0.101` `0.15`  `15`
##   <chr>   <dbl>   <dbl>   <dbl>   <dbl>    <dbl>   <dbl>  <dbl> <dbl>
## 1 M       0.35    0.265   0.09    0.226   0.0995  0.0485  0.07      7
## 2 F       0.53    0.42    0.135   0.677   0.256   0.142   0.21      9
## 3 M       0.44    0.365   0.125   0.516   0.216   0.114   0.155    10
## 4 I       0.33    0.255   0.08    0.205   0.0895  0.0395  0.055     7
## 5 I       0.425   0.3     0.095   0.352   0.141   0.0775  0.12      8
## 6 F       0.53    0.415   0.15    0.778   0.237   0.142   0.33     20
```

```
summary(abalone)
```

```
##       M                 0.455           0.365            0.095
##  Length:4176        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
##      0.514            0.2245           0.101             0.15
##  Min.   :0.0020   Min.   :0.0010   Min.   :0.00050   Min.   :0.0015
##  1st Qu.:0.4415   1st Qu.:0.1860   1st Qu.:0.09337   1st Qu.:0.1300
##  Median :0.7997   Median :0.3360   Median :0.17100   Median :0.2340
```

```
##  Mean   :0.8288   Mean   :0.3594   Mean   :0.18061   Mean   :0.2389
##  3rd Qu.:1.1533   3rd Qu.:0.5020   3rd Qu.:0.25300   3rd Qu.:0.3290
##  Max.   :2.8255   Max.   :1.4880   Max.   :0.76000   Max.   :1.0050
##        15
##  Min.   : 1.000
##  1st Qu.: 8.000
##  Median : 9.000
##  Mean   : 9.932
##  3rd Qu.:11.000
##  Max.   :29.000
```