RWorksheet #7a

Jacklord Espanola

2022-12-09

```
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
```

```
#1. Create a data frame for the table below.
exam_result <- data.frame(
  Student = c(1:10),
  Pretest = c(55, 54, 57, 47, 51, 61, 57, 54, 63, 58),
  Posttest = c(61, 60, 56, 63, 56, 63, 59, 56, 62, 61)
)
exam_result
```

```
##    Student Pretest Posttest
## 1        1      55       61
## 2        2      54       60
## 3        3      57       56
## 4        4      47       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.

#Using the Hmisc package
describe(exam_result)
```

```
## exam_result
##
##  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
## ---------------------------------------------------------------------------
```

```
#Using the pastecs package
library(pastecs)
stat.desc(exam_result)
```

```
##                 Student      Pretest     Posttest
## 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 gr
#• The data were 10,10,10, 20,20,50,10,20,10,50,20,50,20,10.
```

```r
#a. Write the codes and describe the result.
fertilizer_levels <- c(10,10,10,20,20,50,10,20,10,50,20,50,20,10)

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

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

```r
#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

#a. What is the best way to represent this in R?
exercise_levels <- data.frame(
  subject_exercise_levels = c("l", "n", "n", "i", "l", "l", "n", "n", "i", "l")
)
exercise_levels
```

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

```r
#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"
```

```r
  str(state)
```

```
##  chr [1:30] "tas" "sa" "qld" "nsw" "nsw" "nt" "wa" "wa" "qld" "vic" "nsw" ...
```

```r
#a. Apply the factor function and factor level. Describe the results.
state_factor <- factor(c(state))
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
```

```
#factor level
levels(state_factor)
```

```
## [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
#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)
incomes
```

```
##  [1] 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
## [26] 46 59 46 58 43
```

```
#a. Calculate the sample mean income for each state we can now use the special
#function tapply():
income_mean <- tapply(incomes, state, mean)
income_mean
```

```
##      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.

#The result shows the income mean of each state and its level as well. It #shows "tas" has the highest
```

```
#6. Calculate the standard errors of the state income means (refer again to
#number 3)
stdError <- sd(income_mean)/sqrt(length(income_mean))
stdError
```

```
## [1] 1.653911
```

```
#b. Interpret the result.
#The result simply shows the standard errors of the state income means through
#dividing the standard deviation of the income mean by the square root of the
#length of the income mean. Having larger sample size, the standard errors tend
#decreases which mean the lesser spread out values are around the mean in a
#dataset.
```

```
#7. Use the titanic dataset.
data(Titanic)
Titanic <- data.frame(Titanic)
#a. subset the titatic dataset of those who survived and not survived. Show the
#codes and its result.
survived <- subset(Titanic, Survived == "Yes")
survived
```

```
##    Class    Sex   Age Survived Freq
## 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
## 27   3rd   Male Adult      Yes   75
## 28  Crew   Male Adult      Yes  192
## 29   1st Female Adult      Yes  140
## 30   2nd Female Adult      Yes   80
## 31   3rd Female Adult      Yes   76
## 32  Crew Female Adult      Yes   20
```

```
unsurvived <- subset(Titanic, Survived == "No")
unsurvived
```

```
##    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
## 7    3rd Female Child       No   17
## 8   Crew Female Child       No    0
## 9    1st   Male Adult       No  118
## 10   2nd   Male Adult       No  154
## 11   3rd   Male Adult       No  387
## 12  Crew   Male Adult       No  670
## 13   1st Female Adult       No    4
## 14   2nd Female Adult       No   13
## 15   3rd Female Adult       No   89
## 16  Crew Female Adult       No    3
```

```
#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 M

#a. describe what is the dataset all about.
#The data sets are about the breast cancer Wisconsin. The samples arrive
#periodically as Dr. Wolberg reports his clinical cases.

#b. Import the data from MS Excel. Copy the codes.
library(readxl)
dataSet <- read_excel("Breast_Cancer.xlsx")


#c. Compute the descriptive statistics using different packages. Find the
#values of:
#c.1 Standard error of the mean for clump thickness.
error.n <- length(dataSet$`CL. thickness`)
error.sd <- sd(dataSet$`CL. thickness`)
error.se <- error.sd/sqrt(dataSet$`CL. thickness`)
error.se
```

```
##  [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
```

```
#c.2 Coefficient of variability for Marginal Adhesion.
coe_var <- sd(dataSet$`Marg. Adhesion`) / mean(dataSet$`Marg. Adhesion`) * 100
coe_var
```

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

```
#c.3 Number of null values of Bare Nuclei.
bare_nuclei <- subset(dataSet, `Bare. Nuclei` == "NA")
bare_nuclei
```

```
## # A tibble: 2 x 11
##       ID CL. t~1 Cell ~2 Cell ~3 Marg.~4 Epith~5 Bare.~6 Bl. C~7 Norma~8 Mitoses
##    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>     <dbl>   <dbl>   <dbl>
## 1 1.06e6       8       4       5       1       2 NA            7       3       1
## 2 1.10e6       6       6       6       9       6 NA            7       8       1
## # ... with 1 more variable: 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`
```

```
#c.4 Mean and standard deviation for Bland Chromatin
mean(dataSet$`Bl. Cromatin`)
```

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

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

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

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

#Calculating the mean
mean_cshape <- mean(dataSet$`Cell Shape`)
mean_cshape
```

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

```
#Calculating the standard error of the mean
cshape.n <- length(dataSet$`Cell Shape`)
cshape.sd <- sd(dataSet$`Cell Shape`)
cshape.se <- cshape.sd/sqrt(cshape.n)
cshape.se
```

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

```
#Step 3: Find the t-score that corresponds to the confidence level
alpha = 0.05
df_cshape = cshape.n - 1
t.score = qt(p=alpha/2, df=df_cshape,lower.tail=F)
t.score
```

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

```
#Constructing the confidence interval
cInterval <- t.score * cshape.se
cInterval
```

```
## [1] 0.836081
```

```
#Lower
lower_cInterval <- mean_cshape - cInterval

#Upper
upper_cInterval <- mean_cshape + cInterval

c(lower_cInterval,upper_cInterval)
```

```
## [1] 2.327184 3.999346
```

```
#d. How many attributes?
attributes(dataSet)
```

```
## $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.
mrespondents <- subset(dataSet, Class == "malignant")
mrespondents
```

```
## # A tibble: 1 x 11
##       ID CL. t~1 Cell ~2 Cell ~3 Marg.~4 Epith~5 Bare.~6 Bl. C~7 Norma~8 Mitoses
##    <dbl>   <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>     <dbl>   <dbl>   <dbl>
## 1 1.02e6       8      10      10       8       7 10            9       7       1
## # ... with 1 more variable: 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 from the total of 49 respondent.

#Getting the percentage
17  / 49 * 100
```

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

```
#There are 34.69388 or 35% of respondents who are malignant.
```

```
#9. Export the data abalone to the Microsoft excel file. Copy the codes.
install.packages("AppliedPredictiveModeling")
```

```
## Installing package into '/cloud/lib/x86_64-pc-linux-gnu-library/4.2'
## (as 'lib' is unspecified)
```

```
library("AppliedPredictiveModeling")
data(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)
write.xlsx("abalone", "C:\\Abalone\\abalone.xlsx")
```

#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 sets are about the breast cancer Wisconsin. The samples arrive #periodically as Dr. Wolberg reports his clinical cases.

#b. Import the data from MS Excel. Copy the codes. library(readxl) dataSet <- read_excel("Breast_Cancer.xlsx") View(dataSet)

#c. Compute the descriptive statistics using different packages. Find the #values of: #c.1 Standard error of the mean for clump thickness. error.n <- length(dataSet$CL.thickness$)error.sd < −sd(dataSetCL. thickness) error.se <- error.sd/sqrt(dataSet$CL. thickness) error.se

#c.2 Coefficient of variability for Marginal Adhesion. coe_var <- sd(dataSet$Marg.Adhesion$)/mean(dataSetMarg. Adhesion) * 100 coe_var

#c.3 Number of null values of Bare Nuclei. bare_nuclei <- subset(dataSet, Bare. Nuclei == "NA") bare_nuclei

#c.4 Mean and standard deviation for Bland Chromatin mean(dataSet$Bl.Cromatin$)sd(dataSetBl. Cromatin)

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

#Calculating the mean mean_cshape <- mean(dataSet$`Cell Shape`) mean_cshape

#Calculating the standard error of the mean cshape.n <- length(dataSet‘*CellShape*‘)*cshape.sd* <− *sd*(*dataSet*`Cell Shape`) cshape.se <- cshape.sd/sqrt(cshape.n) cshape.se

#Step 3: Find the t-score that corresponds to the confidence level alpha = 0.05 df_cshape = cshape.n - 1 t.score = qt(p=alpha/2, df=df_cshape,lower.tail=F) t.score

#Constructing the confidence interval cInterval <- t.score * cshape.se cInterval

#Lower lower_cInterval <- mean_cshape - cInterval

#Upper upper_cInterval <- mean_cshape + cInterval

c(lower_cInterval,upper_cInterval)

#d. How many attributes? attributes(dataSet)

#e. Find the percentage of respondents who are malignant. Interpret the results. mrespondents <- subset(dataSet, Class == “malignant”) mrespondents

#There 17 respondents who are malignant from the total of 49 respondent.

#Getting the percentage 17 / 49 * 100 #There are 34.69388 or 35% of respondents who are malignant.

#9. Export the data abalone to the Microsoft excel file. Copy the codes. install.packages(“AppliedPredictiveModeling”) library(“AppliedPredictiveModeling”) data(abalone) View(abalone) head(abalone) summary(abalone)

#Exporting the data abalone to the Microsoft excel file library(xlsx) write.xlsx(“abalone”, “C:\Abalone\abalone.xlsx”)