RWorksheet_5.Rmd

Reysha Marie S. Salinas

2023-12-21

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

```
Stud <- 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)

Student_Score <- data.frame(
  Student = Stud,
  Pre_Test = Pre_Test,
  Post_Test =Post_Test
)

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
```

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
```

```
#RESULT
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
```

```r
library(pastecs)


#RESULT
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. 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.

```r
fertilizer_levels <- c(10, 10, 10, 20, 20, 50, 10, 20, 10, 50, 20, 50, 20, 10)
ordered_factor <- ordered(fertilizer_levels)
ordered_factor
```

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

```r
#The ordered_factor will have the levels ordered as 10, 20, 50, reflecting the specified order.
```

3. Abdul Hassan, president of Floor Coverings Unlimited, has asked you to study the ex- ercise 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?

```r
subjects <- c("l", "n", "n", "i", "l", "l", "n", "n", "i", "l")

subjects_factor <- factor(subjects, levels=c("n", "l", "i"))
subjects_factor
```

```
##  [1] l n n i l l n n i l
## Levels: n l i
```

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

```r
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")
```

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

```r
state_factor <- factor(state)

summary(state_factor)
```

```
## act nsw  nt qld  sa tas vic  wa
##   2   6   2   5   4   2   5   4
```

```r
#We can see how many times each state appears in the given sample.
```

```r
state_levels <- c("nsw", "vic", "qld", "wa", "sa", "tas", "nt", "act")

state <- factor(state, levels = state_levels)

summary(state)
```

```
## nsw vic qld  wa  sa tas  nt act
##   6   5   5   4   4   2   2   2
```

```r
#Provide a simple count of occurrences for each level in the order you specified.
```

5. From #4 - continuation:

```r
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 <- 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")
incmeans <- tapply(incomes, state, 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
```

b. Copy the results and interpret.

The median income values across different states and territories in Australia are as follows: In the ACT, the median income is $44,500, while in NSW accountants have a median income of $57,333.33. In the NT, the median income is $55,500, and in QLD, accountants have a median income of $53,600. SA reports a median income of $55,000, while in TAS, accountants have a median income of $60,500. In VIC, the median income is $56,000, and accountants in WA earn a median income of $52,250. These figures offer insights into the central income tendencies for accountants in each region.

```r
#-------------------------------------------------------------------
```

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

```r
#------------------------------------------------------------------

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

incster <- tapply(incomes, state, stdError)

print(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
```

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

```r
mean_incomes <- tapply(incomes, state, mean)


std_incomes <- tapply(incomes, state, sd)

n_incomes <- tapply(incomes, state, length)

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

print(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.

ACT boasts stability at 1.5, while NSW stands strong at 4.31. NT and QLD follow with scores of 4.5 and 4.11, indicating robust economic activity. SA scores 2.74, TAS registers 0.5, and VIC leads at 5.24, showcasing economic strength. WA maintains a solid standing at 2.66. These scores provide a succinct overview, aiding in targeted interventions and policy considerations.

```r
#------------------------------------------------------------------
```

7. Use the titanic dataset.

```r
data("Titanic")

print(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
```

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

```r
data("Titanic")

no_survived_adult <- as.vector(Titanic[, , "Adult", "No"])
no_survived_child <- as.vector(Titanic[, , "Child", "No"])
yes_survived_adult <- as.vector(Titanic[, , "Adult", "Yes"])
yes_survived_child <- as.vector(Titanic[, , "Child", "Yes"])

cat("Number of Adults who did not survive:", sum(no_survived_adult), "\n")
```

```
## Number of Adults who did not survive: 1438
```

```r
cat("Number of Children who did not survive:", sum(no_survived_child), "\n")
```

```
## Number of Children who did not survive: 52
```

```r
cat("Number of Adults who survived:", sum(yes_survived_adult), "\n")
```

```
## Number of Adults who survived: 654
```

```r
cat("Number of Children who survived:", sum(yes_survived_child), "\n")
```

```
## Number of Children who survived: 57
```

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

a. describe what is the dataset all about.

The dataset focuses on women facing breast cancer and involves a survey scale ranging from 1 to 10. This scale is used to assess various characteristics of cell nuclei present in breast cancer, such as clump thickness, size uniformity, shape uniformity, marginal adhesion, epithelial size, bare nucleoli, bland chromatin, normal nucleoli, and mitoses. Each score on the scale reflects the severity or abnormality of the respective characteristic. The dataset aims to capture and analyze these features to gain insights into the nature of breast cancer in the surveyed women.

```r
#-----------------------------------------------------------
```

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

```r
data <- read.csv('breastcancer_wisconsin.csv')

clump_thickness_column <- data$clump_thickness
std_error <- sd(clump_thickness_column) / sqrt(length(clump_thickness_column))

print(std_error)
```

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

d.2 Coefficient of variability for Marginal Adhesion.

```
data <- read.csv('breastcancer_wisconsin.csv')

marginal_adhesion_column <- data$marginal_adhesion
coefficient_of_variability <- sd(marginal_adhesion_column) / mean(marginal_adhesion_column) * 100

print(coefficient_of_variability)
```

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

d.3 Number of null values of Bare Nuclei.

```
data <- read.csv('breastcancer_wisconsin.csv')

bare_nucleoli_column <- data$bare_nucleoli
null_values_count <- sum(is.na(bare_nucleoli_column))


print(null_values_count)
```

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

d.4 Mean and standard deviation for Bland Chromatin

```
mean_bland_chromatin <- mean(data$bland_chromatin, )
sd_bland_chromatin <- sd(data$bland_chromatin, )

print(paste("Mean:", mean_bland_chromatin))
```

```
## [1] "Mean: 3.43776824034335"
```

```
print(paste("Standard deviation:", sd_bland_chromatin))
```

```
## [1] "Standard deviation: 2.43836425232425"
```

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

```
data <- read.csv('breastcancer_wisconsin.csv')
shape_uniformity <- data$shape_uniformity

result <- t.test(shape_uniformity)

cat("Mean:", result$estimate, "\n")
```

```
## Mean: 3.207439
```

```
cat("95% confidence interval:", result$conf.int[1], result$conf.int[2], "\n")
```

```
## 95% confidence interval: 2.986741 3.428138
```

```
#----------------------------------------
```

d. How many attributes?

```
data <- read.csv('breastcancer_wisconsin.csv')

num_attributes <- length(names(data))
print(num_attributes)
```

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

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

```
data <- read.csv('breastcancer_wisconsin.csv')


malignant_count <- sum(data$class == "malignant")
total_count <- nrow(data)

percentage_malignant <- (malignant_count / total_count) * 100
print(percentage_malignant)
```

```
## [1] 0
```

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

```
install.packages("AppliedPredictiveModeling")
```

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

```r
library(openxlsx)

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