

# RWorksheet #7

Grace Anne Capanang

2022-12-13

## ##Basic Statistics

1. Create a data frame for the table below.

```
Students <- seq (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)

studentsGrades <- data.frame(
  Students,
  PreTest,
  PostTest)
studentsGrades
```

##	Students	PreTest	PostTest
## 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

- 1a. Compute the descriptive statistics using different packages (Hmisc and pastecs). Write the codes and its result.

```
library(Hmisc)
```

```
## Warning: package 'Hmisc' was built under R version 4.2.2
```

```
## Loading required package: lattice
```

```
## Loading required package: survival
```

```
## Loading required package: Formula
```

```
## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.2.2

##
## Attaching package: 'Hmisc'

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

```
library(pastecs)
```

```
## Warning: package 'pastecs' was built under R version 4.2.2
```

```
#descriptive statistic using Hmisc package
```

```
hmiscStat <- describe(studentsGrades)
```

```
hmiscStat
```

```
## studentsGrades
```

```
##
```

```
## 3 Variables      10 Observations
```

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

```
## Students
```

	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
## -----
```

*#descriptive statistic using the pastecs package*

```
pastecsStat <- stat.desc(studentsGrades)
pastecsStat
```

```
##           Students      PreTest      PostTest
## nbr.val      10.0000000    10.0000000    10.0000000
## nbr.null      0.0000000     0.0000000     0.0000000
## nbr.na        0.0000000     0.0000000     0.0000000
## min           1.0000000    47.0000000    56.0000000
## max          10.0000000    63.0000000    63.0000000
## range         9.0000000    16.0000000     7.0000000
## sum          55.0000000   557.0000000   597.0000000
## median        5.5000000    56.0000000    60.5000000
## mean          5.5000000    55.7000000    59.7000000
## 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.

```
fertilizerLevel <- factor(c(10, 10, 10, 20, 20, 50, 10, 20, 10, 50, 20, 50, 20, 10),
                          ordered = TRUE)
fertilizerLevel
```

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

```
class(fertilizerLevel)
```

```
## [1] "ordered" "factor"
```

```
orderFertilizerLevel <- sort (fertilizerLevel)
orderFertilizerLevel
```

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

#describe the result: #the result outputs the values in ascending order, where the levels shows the inequality (<) of the values.

- 3a. 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. What is the best way to represent this in R?

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

```
## [1] "l" "n" "n" "i" "l" "l" "n" "n" "i" "l"
```

```
factorLevels <- factor(exerciseLevels)
factorLevels
```

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

```
levels(factorLevels) <- c("intense", "light", "none")
factorLevels
```

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

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")`

4a. 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")
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"
```

```
factoredDataStateAccountants <- factor(state)
factoredDataStateAccountants
```

```
## [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
```

```
#describe the result
#the factor function in this item arranged the values of the variable state alphabetically and
#sorted them or leveled them.
```

```
levelfactoredStateAccountants <- levels(factoredDataStateAccountants)
levelfactoredStateAccountants
```

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

```
#describe the result
#the result outputs the sorted values from previous number in character form because of the quotation
#marks (""), and the 'Levels' that can be seen in the output from factor function is now gone.
```

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
```

- 5a. Calculate the sample mean income for each state we can now use the special function `tapply()`:

```
incmeans <- tapply(incomes, factoredDataStateAccountants, 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 `tapply` function in this item calculated the mean of the factored State accountants' income.

6. Calculate the standard errors of the state income means (refer again to number 3). What is the standard error? Write the codes.

```
stdError <- sqrt(var(incmeans)/length(incmeans))
stdError
```

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

*#another method for #6*

```
incMeansSD <- sqrt(var(incmeans)) #mean
incMeansSD
```

```
## [1] 4.677966
```

```
sd(incmeans)
```

```
## [1] 4.677966
```

```
length(incmeans) #size
```

```
## [1] 8
```

```
standardError <- incMeansSD / sqrt(8) #standard error
standardError
```

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

After this assignment, the standard errors are calculated by:

```
#incster <- tapply(incomes, factoredDataStateAccountants, stdError) #incster
```

6b. Interpret the result. #in this item there is an error, but the standard error of the factored State accountants' income is produce in the output in Rfile. #output is: #act nsw nt qld sa tas vic wa #1.500000 4.310195 4.500000 4.106093 2.738613 0.500000 5.244044 2.657536

7. Use the titanic dataset.

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

```
data(Titanic)

titanicDataFrame <- data.frame(Titanic)

titanicSurvived <- subset(titanicDataFrame, Survived == "Yes")
titanicSurvived
```

```
##      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
```

```
titanicDidntSurvived <- subset(titanicDataFrame, Survived == "No")
titanicDidntSurvived
```

```
##      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 Microsoft Excel.

8a. describe what is the dataset all about. #the dataset is about a database of Dr. Wolberg's breast cancer clinical cases samples.

8b. Import the data from MS Excel. Copy the codes.

```
getwd()
```

```
## [1] "C:/ANOTHER/NEECHAN/PracticeGit/#3b, 4, 5/Worksheet 7"
```

```
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.2.2
```

```
library(plotrix)
```

```
brstCancerData <- read_excel('C:\\ANOTHER\\NEECHAN\\PracticeGit\\#3b, 4, 5\\Worksheet 7\\breastCancer.xls')
brstCancerData
```

```
## # A tibble: 49 x 11
##       Id CL. thickne~1 Cell ~2 Cell ~3 Marg.~4 Epith~5 Bare.~6 Bl. C~7 Norma~8
##       <dbl>         <dbl>   <dbl>   <dbl>   <dbl>   <dbl> <chr>     <dbl>   <dbl>
## 1 1000025           5         1         1         1         2 1         3         1
## 2 1002945           5         4         4         5         7 10         3         2
## 3 1015425           3         1         1         1         2 2         3         1
## 4 1016277           6         8         8         1         3 4         3         7
## 5 1017023           4         1         1         3         2 1         3         1
## 6 1017122           8        10        10         8         7 10         9         7
## 7 1018099           1         1         1         1         2 10         3         1
## 8 1018561           2         1         2         1         2 1         3         1
## 9 1033078           2         1         1         1         2 1         1         1
## 10 1033078          4         2         1         1         2 1         2         1
## # ... with 39 more rows, 2 more variables: Mitoses <dbl>, 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'
```

```
brstCancerDataFrame <- data.frame(brstCancerData)
brstCancerDataFrame
```

##	Id	CL..thickness	Cell.size	Cell.Shape	Marg..Adhesion	Epith..C.size
## 1	1000025	5	1	1	1	2
## 2	1002945	5	4	4	5	7
## 3	1015425	3	1	1	1	2
## 4	1016277	6	8	8	1	3
## 5	1017023	4	1	1	3	2
## 6	1017122	8	10	10	8	7
## 7	1018099	1	1	1	1	2
## 8	1018561	2	1	2	1	2
## 9	1033078	2	1	1	1	2
## 10	1033078	4	2	1	1	2
## 11	1035283	1	1	1	1	1
## 12	1036172	2	1	1	1	2
## 13	1041801	5	3	3	3	2
## 14	1043999	1	1	1	1	2
## 15	1044572	8	7	5	10	7
## 16	1047630	7	4	6	4	6
## 17	1048672	4	1	1	1	2
## 18	1049815	4	1	1	1	2
## 19	1050670	10	7	7	6	4
## 20	1050718	6	1	1	1	2
## 21	1054590	7	3	2	10	5
## 22	1054593	10	5	5	3	6
## 23	1056784	3	1	1	1	2
## 24	1057013	8	4	5	1	2
## 25	1059552	1	1	1	1	2
## 26	1065726	5	2	3	4	2
## 27	1066373	3	2	1	1	1
## 28	1066979	5	1	1	1	2
## 29	1067444	2	1	1	1	2
## 30	1070935	1	1	3	1	2
## 31	1070935	3	1	1	1	1
## 32	1071760	2	1	1	1	2
## 33	1072179	10	7	7	3	8
## 34	1074610	2	1	1	2	2
## 35	1075123	3	1	2	1	2
## 36	1079304	2	1	1	1	2
## 37	1080185	10	10	10	8	6
## 38	1081791	6	2	1	1	1
## 39	1084584	5	4	4	9	2
## 40	1091262	2	5	3	3	6
## 41	1096800	6	6	6	9	6
## 42	1099510	10	4	3	1	3
## 43	1100524	6	10	10	2	8
## 44	1102573	5	6	5	6	10
## 45	1103608	10	10	10	4	8
## 46	1103722	1	1	1	1	2
## 47	1105257	3	7	7	4	4
## 48	1105524	1	1	1	1	2
## 49	1106095	4	1	1	3	2



##	Bare..Nuclei	Bl..Cromatin	Normal.nucleoli	Mitoses	Class
## 1	1	3	1	1	benign
## 2	10	3	2	1	benign
## 3	2	3	1	1	benign
## 4	4	3	7	1	benign
## 5	1	3	1	1	benign
## 6	10	9	7	1	malignant
## 7	10	3	1	1	benign
## 8	1	3	1	1	benign
## 9	1	1	1	5	benign
## 10	1	2	1	1	benign
## 11	1	3	1	1	benign
## 12	1	2	1	1	benign
## 13	3	4	4	2	malignant
## 14	3	3	1	1	benign
## 15	9	5	5	4	malignant
## 16	1	4	3	1	malignant
## 17	1	2	1	1	benign
## 18	1	3	1	1	benign
## 19	10	4	1	2	malignant
## 20	1	3	1	1	benign
## 21	10	5	4	4	malignant
## 22	7	7	10	1	malignant
## 23	1	2	1	1	benign
## 24	NA	7	3	1	malignant
## 25	1	3	1	1	benign
## 26	7	3	6	1	malignant
## 27	1	2	1	1	benign
## 28	1	2	1	1	benign
## 29	1	2	1	1	benign
## 30	1	1	1	1	benign
## 31	1	2	1	1	benign
## 32	1	3	1	1	benign
## 33	5	7	4	3	malignant
## 34	1	3	1	1	benign
## 35	1	2	1	1	benign
## 36	1	2	1	1	benign
## 37	1	8	9	1	malignant
## 38	1	7	1	1	benign
## 39	10	5	6	1	malignant
## 40	7	7	5	1	malignant
## 41	NA	7	8	1	benign
## 42	3	6	5	2	malignant
## 43	10	7	3	3	malignant
## 44	1	3	1	1	malignant
## 45	1	8	10	1	malignant
## 46	1	2	1	2	benign
## 47	9	4	8	1	malignant
## 48	1	2	1	1	benign
## 49	1	3	1	1	benign

8c1. Standard error of the mean for clump thickness.

```
clumpThickness <- subset(brstCancerDataFrame, select = CL..thickness)
clumpThickness
```

```
##      CL..thickness
## 1                5
## 2                5
## 3                3
## 4                6
## 5                4
## 6                8
## 7                1
## 8                2
## 9                2
## 10               4
## 11               1
## 12               2
## 13               5
## 14               1
## 15               8
## 16               7
## 17               4
## 18               4
## 19              10
## 20               6
## 21               7
## 22              10
## 23               3
## 24               8
## 25               1
## 26               5
## 27               3
## 28               5
## 29               2
## 30               1
## 31               3
## 32               2
## 33              10
## 34               2
## 35               3
## 36               2
## 37              10
## 38               6
## 39               5
## 40               2
## 41               6
## 42              10
## 43               6
## 44               5
## 45              10
## 46               1
## 47               3
## 48               1
## 49               4
```

```
std.error(clumpThickness)
```

```
## CL..thickness  
##      0.4092884
```

```
std.error(brstCancerDataFrame$CL..thickness) #trying another method
```

```
## [1] 0.4092884
```

8c.2 Coefficient of variability for Marginal Adhesion.

```
marginalAdhesionCV <- sd(brstCancerDataFrame$Marg..Adhesion) / mean(brstCancerDataFrame$Marg..Adhesion)  
marginalAdhesionCV
```

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

8c.3 Number of null values of Bare Nuclei.

```
var <- subset(brstCancerDataFrame, select = Bare..Nuclei)  
var
```

```
##      Bare..Nuclei  
## 1                1  
## 2               10  
## 3                2  
## 4                4  
## 5                1  
## 6               10  
## 7               10  
## 8                1  
## 9                1  
## 10               1  
## 11               1  
## 12               1  
## 13               3  
## 14               3  
## 15               9  
## 16               1  
## 17               1  
## 18               1  
## 19              10  
## 20               1  
## 21              10  
## 22               7  
## 23               1  
## 24              NA  
## 25               1  
## 26               7  
## 27               1  
## 28               1  
## 29               1
```

```
## 30      1
## 31      1
## 32      1
## 33      5
## 34      1
## 35      1
## 36      1
## 37      1
## 38      1
## 39     10
## 40      7
## 41     NA
## 42      3
## 43     10
## 44      1
## 45      1
## 46      1
## 47      9
## 48      1
## 49      1
```

```
sum(is.na(var))
```

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

8c.4 Mean and standard deviation for Bland Chromatin.

```
blChromatinSD <- sqrt(var(brstCancerDataFrame$Bl..Cromatin))
blChromatinSD
```

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

```
blChromatinMean <- mean(brstCancerDataFrame$Bl..Cromatin)
blChromatinMean
```

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

8c.5 Confidence interval of the mean for Uniformity of Cell Shape.

```
s = length(brstCancerDataFrame$Cell.Shape) #size
s
```

```
## [1] 49
```

```
meanCS <- mean(brstCancerDataFrame$Cell.Shape) #mean
meanCS
```

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

```
csSD <- sqrt(var(brstCancerDataFrame$Cell.Shape)) #standard deviation  
csSD
```

```
## [1] 2.910806
```

```
sd(brstCancerDataFrame$Cell.Shape) #another method for sd
```

```
## [1] 2.910806
```

```
standardError <- csSD / sqrt(s) #standard error  
standardError
```

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

```
alpha = 0.05  
degrees_of_freedom = s - 1  
t_score = qt(p=alpha/2, df=degrees_of_freedom, lower.tail=F)  
print(t_score)
```

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

```
marginError <- t_score * standardError #margin error  
marginError
```

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

```
lowerBound <- meanCS - marginError  
lowerBound
```

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

```
upperBound <- meanCS + marginError  
upperBound
```

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

```
#lower and upper bounds of confidence interval  
lowerBound
```

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

```
upperBound
```

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

8d. How many attributes?

```
attributesNumber <- attributes(brstCancerData)
attributesNumber
```

```
## $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"
```

8e. Find the percentage of respondents who are malignant. Interpret the results.

```
subsetting <- subset(brstCancerDataFrame, Class=="malignant")
subsetting
```

```
##      Id CL..thickness Cell.size Cell.Shape Marg..Adhesion Epith..C.size
## 6  1017122           8         10         10              8              7
## 13 1041801           5          3          3              3              2
## 15 1044572           8          7          5             10              7
## 16 1047630           7          4          6              4              6
## 19 1050670          10          7          7              6              4
## 21 1054590           7          3          2             10              5
## 22 1054593          10          5          5              3              6
## 24 1057013           8          4          5              1              2
## 26 1065726           5          2          3              4              2
## 33 1072179          10          7          7              3              8
## 37 1080185          10         10         10              8              6
## 39 1084584           5          4          4              9              2
## 40 1091262           2          5          3              3              6
## 42 1099510          10          4          3              1              3
## 43 1100524           6         10         10              2              8
## 44 1102573           5          6          5              6             10
## 45 1103608          10         10         10              4              8
## 47 1105257           3          7          7              4              4
##      Bare..Nuclei Bl..Cromatin Normal.nucleoli Mitoses      Class
## 6              10              9              7          1 malignant
## 13              3              4              4          2 malignant
## 15              9              5              5          4 malignant
## 16              1              4              3          1 malignant
## 19             10              4              1          2 malignant
## 21             10              5              4          4 malignant
## 22              7              7             10          1 malignant
## 24             NA              7              3          1 malignant
## 26              7              3              6          1 malignant
## 33              5              7              4          3 malignant
## 37              1              8              9          1 malignant
## 39             10              5              6          1 malignant
```

```
## 40      7      7      5      1 malignant
## 42      3      6      5      2 malignant
## 43     10      7      3      3 malignant
## 44      1      3      1      1 malignant
## 45      1      8     10      1 malignant
## 47      9      4      8      1 malignant
```

```
classVar <- (brstCancerDataFrame$Class)
classVar
```

```
## [1] "benign" "benign" "benign" "benign" "benign" "malignant"
## [7] "benign" "benign" "benign" "benign" "benign" "benign"
## [13] "malignant" "benign" "malignant" "malignant" "benign" "benign"
## [19] "malignant" "benign" "malignant" "malignant" "benign" "malignant"
## [25] "benign" "malignant" "benign" "benign" "benign" "benign"
## [31] "benign" "benign" "malignant" "benign" "benign" "benign"
## [37] "malignant" "benign" "malignant" "malignant" "benign" "malignant"
## [43] "malignant" "malignant" "malignant" "benign" "malignant" "benign"
## [49] "benign"
```

```
malignant <- length(subsetting)
malignant
```

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

```
variableClass <- length(classVar)
variableClass
```

```
## [1] 49
```

```
options (digits = 4)
percentage <- (malignant * 100) / variableClass
percentage
```

```
## [1] 22.45
```

#In this item we got the number of patients whose their type of cancer is malignant by subsetting. #And then we got the percentage of the malignant patients from the subsetted variable multiply to 100 and divided by the overall number of respondents.

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

```
library(AppliedPredictiveModeling)
```

```
## Warning: package 'AppliedPredictiveModeling' was built under R version 4.2.2
```

```
data(abalone)
library(xlsx)
```

```
## Warning: package 'xlsx' was built under R version 4.2.2
```

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

```
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.   :0.055    Min.   :0.000    Min.   :0.002
## I:1342  1st Qu.:0.450    1st Qu.:0.350    1st Qu.:0.115    1st Qu.:0.442
## M:1528  Median :0.545    Median :0.425    Median :0.140    Median :0.799
##        Mean   :0.524    Mean   :0.408    Mean   :0.140    Mean   :0.829
##        3rd Qu.:0.615    3rd Qu.:0.480    3rd Qu.:0.165    3rd Qu.:1.153
##        Max.   :0.815    Max.   :0.650    Max.   :1.130    Max.   :2.825
## ShuckedWeight VisceraWeight      ShellWeight      Rings
## Min.   :0.001    Min.   :0.0005    Min.   :0.0015    Min.   : 1.00
## 1st Qu.:0.186    1st Qu.:0.0935    1st Qu.:0.1300    1st Qu.: 8.00
## Median :0.336    Median :0.1710    Median :0.2340    Median : 9.00
## Mean   :0.359    Mean   :0.1806    Mean   :0.2388    Mean   : 9.93
## 3rd Qu.:0.502    3rd Qu.:0.2530    3rd Qu.:0.3290    3rd Qu.:11.00
## Max.   :1.488    Max.   :0.7600    Max.   :1.0050    Max.   :29.00
```