

RWorksheet-7a

Daisy Ann Pajarillo

2022-12-8

1. Create a data frame for the table below.

```
Student <- c(1:10)
Student

## [1] 1 2 3 4 5 6 7 8 9 10

Pre_Test <- c(55, 54, 47, 57, 51, 61, 57, 54, 63, 58)
Pre_Test

## [1] 55 54 47 57 51 61 57 54 63 58

Post_Test <- c(61, 60, 56, 63, 56, 63, 59, 56, 62, 61)
Post_Test

## [1] 61 60 56 63 56 63 59 56 62 61

data1 <- data.frame(Student, Pre_Test, Post_Test)
data1

##      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.

```
install.packages("Hmisc")

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

install.packages("pastecs")

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

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)
```

```
a <- describe(data1)
```

```
a
```

```
## data1
##
## 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
## -----
## Pre_Test
##      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
## -----
## Post_Test
##      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
## -----
```

```
b <-stat.desc(data1)
b
```

```
##           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. • The data were 10,10,10, 20,20,50,10,20,10,50,20,50,20,10.

a. Write the codes and describe the result. Answer: categorize the data and display the factor's values

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

```
## [1] 10 10 10 20 20 50 10 20 10 50 20 50 10
```

```
fdata <- factor(fertilizerData)
fdata
```

```
## [1] 10 10 10 20 20 50 10 20 10 50 20 50 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

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

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

a. What is the best way to represent this in R? Answer: factor

```
fdata1 <- factor(Sub)
fdata1
```

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

```
dat = levels(fdata1) =c("none","light","intense")
dat
```

```
## [1] "none" "light" "intense"
```

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 <- 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"
```

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

Answer: categorize the data and display the factor's values

```
state_data <- factor(state)
state_data

## [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(state_data)

## [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 <- 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()`:

```
incomes_means <- tapply(incomes, state_data, mean)
incomes_means

##      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.

Answer: RResult:

```
act nsw nt qld sa tas vic wa 44.50000 57.33333 55.50000 53.60000 55.00000 60.50000 56.00000 52.25000
```

It give the mean of each group

6. Calculate the standard errors of the state income means (refer again to number 3) `stdError <- function(x) sqrt(var(x)/length(x))` Note: After this assignment, the standard errors are calculated by: `incster <- tapply(incomes, statef, stdError)`

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

```
incster <- tapply(Sub, fdata1, stdError)
```

```
## Warning in var(x): NAs introduced by coercion
```

```
## Warning in var(x): NAs introduced by coercion
```

```
## Warning in var(x): NAs introduced by coercion
```

```
incster
```

```
##      none    light intense
```

```
##      NA      NA      NA
```

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

```
incster <- tapply(Sub, fdata1, stdError) incster
```

b. Interpret the result. The Result: none light intense NA NA NA

The result is NA because the data inside the object in #3 is Character

7. Use the titanic dataset.

```
data("Titanic")
```

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

```
titanic_data
```

##	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
## 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
```

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

```
Tdata<- subset(titanic_data, select = "Survived")
```

Tdata

```
## Survived
## 1 No
## 2 No
## 3 No
## 4 No
## 5 No
## 6 No
## 7 No
## 8 No
## 9 No
## 10 No
## 11 No
## 12 No
## 13 No
## 14 No
## 15 No
## 16 No
## 17 Yes
## 18 Yes
## 19 Yes
## 20 Yes
## 21 Yes
## 22 Yes
## 23 Yes
## 24 Yes
## 25 Yes
## 26 Yes
## 27 Yes
## 28 Yes
## 29 Yes
## 30 Yes
## 31 Yes
## 32 Yes
```

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 is all about Breast cancer sample

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

```
getwd()
```

```
## [1] "/cloud/project/pajarillo_repo/RWorksheet-7a"
```

```
importData <- read.table("/cloud/project/pajarillo_repo/RWorksheet-7a/BreastCanser - Sheet1.csv", header=
importData
```

##	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	1	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	3	malignant
## 33	5	7	4	1	benign
## 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

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

- c.1 Standard error of the mean for clump thickness. c.2 Coefficient of variability for Marginal Adhesion.
- c.3 Number of null values of Bare Nuclei. c.4 Mean and standard deviation for Bland Chromatin c.5
- Confidence interval of the mean for Uniformity of Cell Shape

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

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

```
## Warning in var(x): NAs introduced by coercion
```

```
## Warning in sqrt(var(x)/length(x)): NaNs produced
```

```
##              Id CL..thickness Cell.size Cell.Shape Marg..Adhesion
## Id          8962.144155    22.7631913 40.4555800 37.1417338    28.1723159
## CL..thickness 22.763191    0.8638357  0.7414739  0.7261727    0.6072702
## Cell.size     40.455580    0.7414739  0.8897131  0.8696997    0.6643532
## Cell.Shape    37.141734    0.7261727  0.8696997  0.8776409    0.6301044
## Marg..Adhesion 28.172316    0.6072702  0.6643532  0.6301044    0.8233790
## Epith..C.size 35.142251    0.5987451  0.7104877  0.6870343    0.5974526
## Bare..Nuclei   NA          NA          NA          NA          NA
## Bl..Cromatin   35.524148    0.6285075  0.6565992  0.6326144    0.5210845
## Normal.nucleoli 40.834699    0.6721903  0.7520586  0.7415000    0.6281999
## Mitoses        3.394036    0.1525398  0.1600817  0.0894479    0.2402232
## Class          NA          NA          NA          NA          NA
##              Epith..C.size Bare..Nuclei Bl..Cromatin Normal.nucleoli
## Id          35.1422511          NA    35.52414809    40.8346994
## CL..thickness 0.5987451          NA    0.62850747    0.6721903
## Cell.size     0.7104877          NA    0.65659915    0.7520586
## Cell.Shape    0.6870343          NA    0.63261441    0.7415000
## Marg..Adhesion 0.5974526          NA    0.52108451    0.6281999
## Epith..C.size 0.7032151          NA    0.52544265    0.5687165
## Bare..Nuclei   NA          NA          NA          NA
## Bl..Cromatin   0.5254427          NA    0.62869194    0.6180582
## Normal.nucleoli 0.5687165          NA    0.61805823    0.8356338
## Mitoses        0.1746319          NA    0.03832453    NaN
## Class          NA          NA          NA          NA
##              Mitoses Class
## Id          3.39403602    NA
## CL..thickness 0.15253983    NA
## Cell.size     0.16008173    NA
## Cell.Shape    0.08944790    NA
## Marg..Adhesion 0.24022317    NA
## Epith..C.size 0.17463187    NA
## Bare..Nuclei   NA          NA
## Bl..Cromatin   0.03832453    NA
## Normal.nucleoli NaN          NA
## Mitoses        0.27213397    NA
## Class          NA          NA
```

```
dat1 <- (importData$CL..thickness)
dat1
```

```
## [1] 5 5 3 6 4 8 1 2 2 4 1 2 5 1 8 7 4 4 10 6 7 10 3 8 1
## [26] 5 3 5 2 1 3 2 10 2 3 2 10 6 5 2 6 10 6 5 10 1 3 1 4
```

```
dat2 <- factor(dat1)
dat2
```

```
## [1] 5 5 3 6 4 8 1 2 2 4 1 2 5 1 8 7 4 4 10 6 7 10 3 8 1
## [26] 5 3 5 2 1 3 2 10 2 3 2 10 6 5 2 6 10 6 5 10 1 3 1 4
```

```
## Levels: 1 2 3 4 5 6 7 8 10
inc <- tapply(dat1, dat2, stdError)
inc

## 1 2 3 4 5 6 7 8 10
## 0 0 0 0 0 0 0 0 0

clump <- mean(dat1)
clump

## [1] 4.571429

c.2 Coefficient of variability for Marginal Adhesion.
CV <-sapply(importData, function(x) sd(x) / mean(x) * 100)

## Warning in var(if (is.vector(x) || is.factor(x)) x else as.double(x), na.rm =
## na.rm): NAs introduced by coercion
## Warning in mean.default(x): argument is not numeric or logical: returning NA
CV

##           Id    CL..thickness    Cell.size    Cell.Shape    Marg..Adhesion
##      2.803817    62.672289    92.096419    92.019015    97.672353
## Epith..C.size    Bare..Nuclei    Bl..Cromatin    Normal.nucleoli    Mitoses
##      69.684590             NA    54.346611    97.699789    67.008707
##           Class
##             NA

c.3 Number of null values of Bare Nuclei.
NV <- (importData$Bare..Nuclei)
NV

## [1] 1 10 2 4 1 10 10 1 1 1 1 1 3 3 9 1 1 1 10 1 10 7 1 NA 1
## [26] 7 1 1 1 1 1 1 5 1 1 1 1 1 10 7 NA 3 10 1 1 1 9 1 1

num <-sum(NV %in% c(NA))
num

## [1] 2

nn <-is.null(list(NV))
nn

## [1] FALSE

c.4 Mean and standard deviation for Bland Chromatin
ndata <- (importData$Bl..Cromatin)
ndata

## [1] 3 3 3 3 3 9 3 3 1 2 3 2 4 3 5 4 2 3 4 3 5 7 2 7 3 3 2 2 2 1 2 3 7 3 2 2 8 7
## [39] 5 7 7 6 7 3 8 2 4 2 3

BC <- mean(ndata)
BC

## [1] 3.836735

SD <- sd(ndata)
SD
```

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

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

```
Calc_Mean <- mean(importData$Cell.Shape)
Calc_Mean
```

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

d. How many attributes?

```
AT <- attributes(importData)
AT
```

```
## $names
## [1] "Id" "CL..thickness" "Cell.size" "Cell.Shape"
## [5] "Marg..Adhesion" "Epith..C.size" "Bare..Nuclei" "Bl..Cromatin"
## [9] "Normal.nucleoli" "Mitoses" "Class"
##
## $class
## [1] "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
```

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

```
a1 <- (importData$Class)
a1
```

```
## [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" "malignant" "benign" "benign" "benign" "benign"
## [37] "malignant" "benign" "malignant" "malignant" "benign" "malignant"
## [43] "malignant" "malignant" "malignant" "benign" "malignant" "benign"
## [49] "benign"
```

```
nr <- nrow(importData)
nr
```

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

```
pp <- sum(c(a1 == "malignant")/nr*100)
pp
```

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

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/4.2'
## (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
```

```
if (interactive()) View(abalone)
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
```

Export Data abalone to xlsx

```
install.packages("readxl")
```

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

```
install.packages("xlsxjars")
```

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

```
install.packages("xlsx")
```

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

```
library(readxl)
```

```
library(xlsx)
```

```
write.xlsx(abalone, "C:\\daisy\\pajarillo_repo\\RWorksheet7a\\abalone.xlsx")
```