Ch8. Input and output

Jongrak Jeong

2023-01-17

```
setwd("~/Library/Mobile Documents/com~apple~CloudDocs/Study/2_Data Science/Practice/R Programming by He
```

1. read.table() writh.table()

read.table()

ex1. read exams.txt

```
exams <- read.table("exams.txt", header = T)
str(exams)
```

```
## 'data.frame':    26 obs. of  3 variables:
##  $ course.id: int  1 2 3 4 5 6 7 8 9 10 ...
##  $ mid      : int  8 22 25 25 21 12 12 29 40 25 ...
##  $ final    : int  11 24 31 13 34 26 6 36 34 38 ...
```

```
exams_2 <- read.table("exams_2.txt", header = T)
str(exams_2)
```

```
## 'data.frame':    26 obs. of  2 variables:
##  $ mid  : int  8 22 25 25 21 12 12 29 40 25 ...
##  $ final: int  11 24 31 13 34 26 6 36 34 38 ...
```

ex2. add new variable and write table

```
# add new variable
exams$total <- exams$mid + exams$final

# output
write.table(exams, file = "exams_t.txt", row.names = F)
```

2. read.table() and stringsAsFactors = F (???)

prevent string variables from being converted as factors

- probably in present r studio, it's different. it seems that 'stringsAsFactors = F' is defualt.

```
survey <- read.table("survey.txt", header = T)
str(survey)
```

```
## 'data.frame':    5 obs. of  3 variables:
##  $ gender   : chr  "male" "female" "male" "male" ...
##  $ age      : int  12 62 23 22 34
```

```
##  $ preference: int  4 4 3 2 1
```

```
survey.1 <- read.table("survey.txt", header = T, stringsAsFactors = F)
str(survey.1)
```

```
## 'data.frame':    5 obs. of  3 variables:
##  $ gender    : chr  "male" "female" "male" "male" ...
##  $ age       : int  12 62 23 22 34
##  $ preference: int  4 4 3 2 1
```

```
survey.1[1, 1] <- "child"
survey.1
```

```
##   gender age preference
## 1  child  12          4
## 2 female  62          4
## 3   male  23          3
## 4   male  22          2
## 5 female  34          1
```

3. read.csv() and write.csv

- header = T is default no header csv? > header = F

4. scan()

useful for reading unstructured string data

```
# seperator is " "
lyrics <- scan("yesterday.txt", what = "")
str(lyrics)
```

```
##  chr [1:126] "Yesterday," "all" "my" "troubles" "seemed" "so" "far" "away." ...
```

```
head(lyrics, 10)
```

```
##  [1] "Yesterday," "all"        "my"         "troubles"   "seemed"
##  [6] "so"         "far"        "away."      "Now"        "it"
```

```
# seperator is "\n"
lyrics.2 <- scan("yesterday.txt", what = "", sep = "\n")
str(lyrics.2)
```

```
##  chr [1:20] "Yesterday, all my troubles seemed so far away." ...
```

```
head(lyrics.2, 10)
```

```
##  [1] "Yesterday, all my troubles seemed so far away."
##  [2] "Now it looks as though they're here to stay."
##  [3] "Oh, I believe in yesterday."
##  [4] "Suddenly, I'm not half the man I used to be."
##  [5] "There's a shadow hanging over me."
##  [6] "Oh, yesterday came suddenly."
##  [7] "Why she had to go?"
##  [8] "I don't know, she wouldn't say."
##  [9] "I said something wrong."
## [10] "Now I long for yesterday."
```

5. load data from web url

```
uci.abalone <- "https://archive.ics.uci.edu/ml/machine-learning-databases/abalone/abalone.data"
abalone <- read.csv(uci.abalone, header = F)
colnames(abalone) <- c("Sex", "Length", "Diameter", "Height", "Whole", "Shucked", "Viscera", "Shell", "
```

```
head(abalone)
```

```
##   Sex Length Diameter Height  Whole Shucked Viscera Shell Rings
## 1   M  0.455    0.365  0.095 0.5140  0.2245  0.1010 0.150    15
## 2   M  0.350    0.265  0.090 0.2255  0.0995  0.0485 0.070     7
## 3   F  0.530    0.420  0.135 0.6770  0.2565  0.1415 0.210     9
## 4   M  0.440    0.365  0.125 0.5160  0.2155  0.1140 0.155    10
## 5   I  0.330    0.255  0.080 0.2050  0.0895  0.0395 0.055     7
## 6   I  0.425    0.300  0.095 0.3515  0.1410  0.0775 0.120     8
```

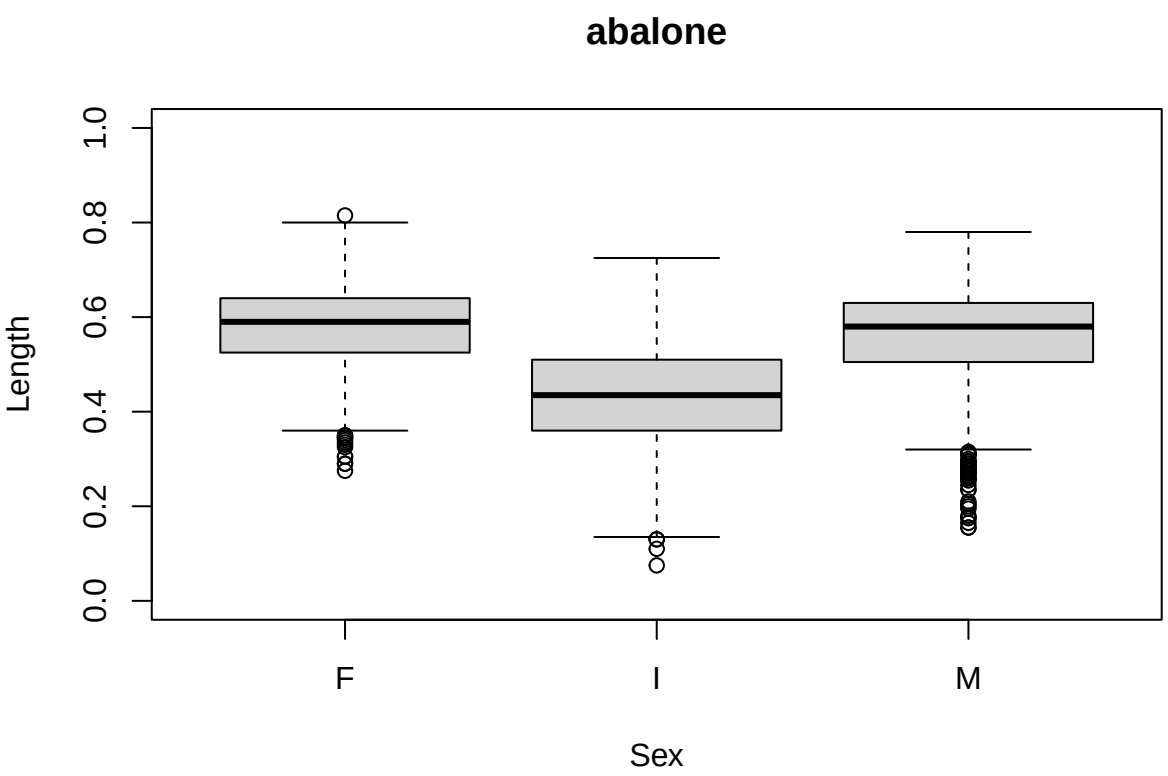
```
with(abalone, table(Sex))
```

```
## Sex
##    F    I    M
## 1307 1342 1528
```

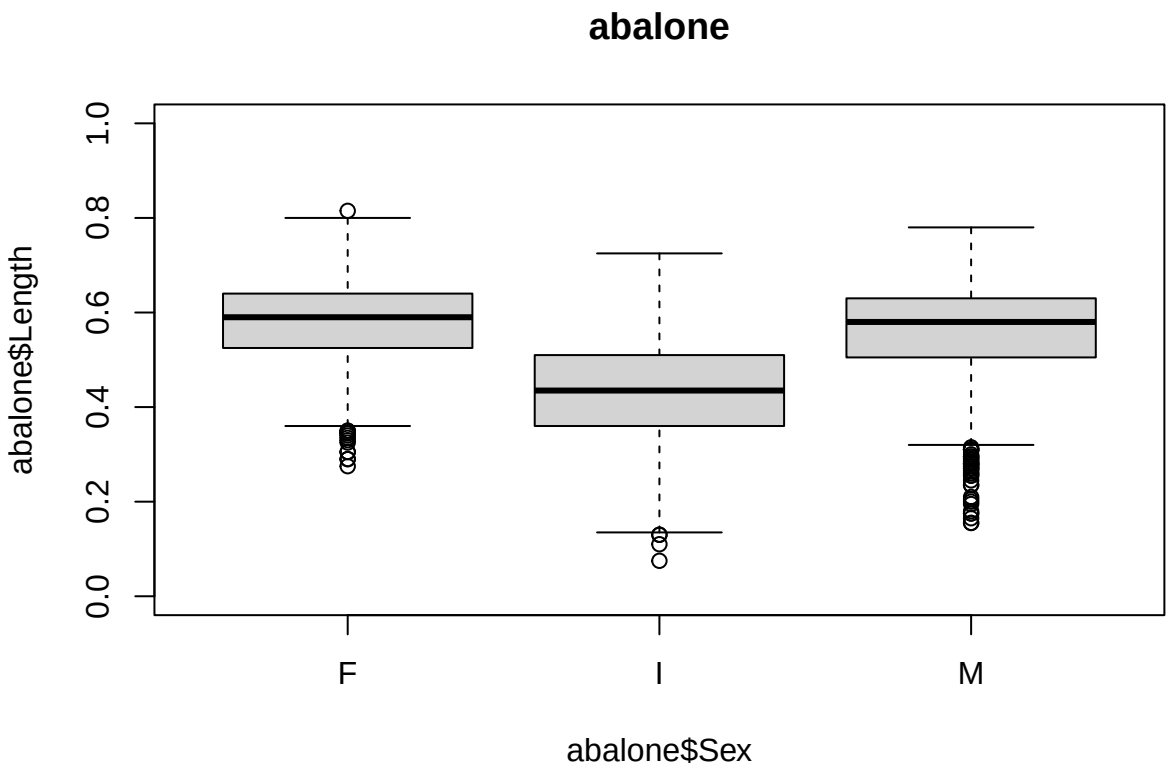
```
table(abalone$Sex)
```

```
##
##    F    I    M
## 1307 1342 1528
```

```
with(abalone, boxplot(Length ~ Sex, main = "abalone", ylim = c(0, 1)))
```




```
boxplot(abalone$Length ~ abalone$Sex, main = "abalone", ylim = c(0, 1))
```



6. load SPSS and Excel dataset

With 'foreign' library, read.spss() and read.xlsx can be used to load SPSS and Excel dataset

7. saving textual data output: sink()

sink("filename") ~ output record ~ sink()

```
sink("output summaries of abalone data.txt") # start recording
summary(abalone[abalone$Sex == "M", 2:4])
```

```
##      Length          Diameter          Height
##  Min.   :0.1550   Min.   :0.1100   Min.   :0.0250
##  1st Qu.:0.5050   1st Qu.:0.3950   1st Qu.:0.1300
##  Median :0.5800   Median :0.4550   Median :0.1550
##  Mean   :0.5614   Mean   :0.4393   Mean   :0.1514
##  3rd Qu.:0.6300   3rd Qu.:0.5000   3rd Qu.:0.1750
##  Max.   :0.7800   Max.   :0.6300   Max.   :0.5150
```

```
summary(abalone[abalone$Sex == "F", 2:4])
```

```
##      Length          Diameter          Height
##  Min.   :0.2750   Min.   :0.1950   Min.   :0.015
##  1st Qu.:0.5250   1st Qu.:0.4100   1st Qu.:0.140
##  Median :0.5900   Median :0.4650   Median :0.160
##  Mean   :0.5791   Mean   :0.4547   Mean   :0.158
##  3rd Qu.:0.6400   3rd Qu.:0.5050   3rd Qu.:0.175
##  Max.   :0.8150   Max.   :0.6500   Max.   :1.130
```

```
sink() # stop recording
```

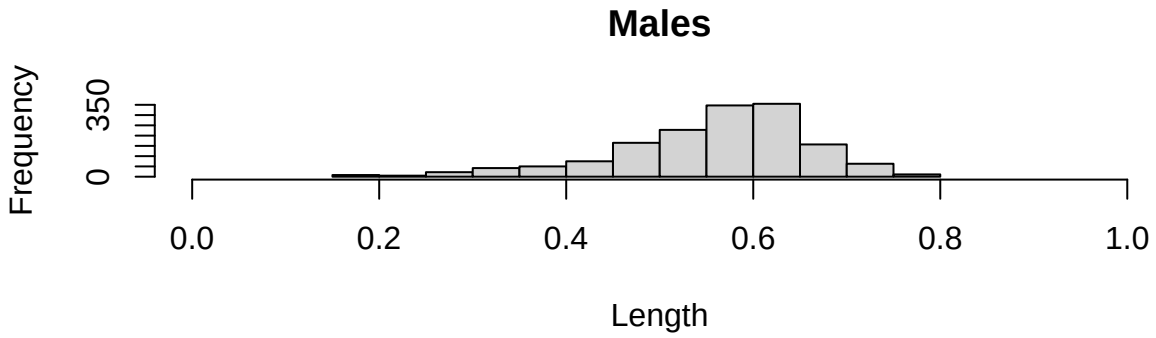
8. saving graphical data output: pdf() ~ dev.off()

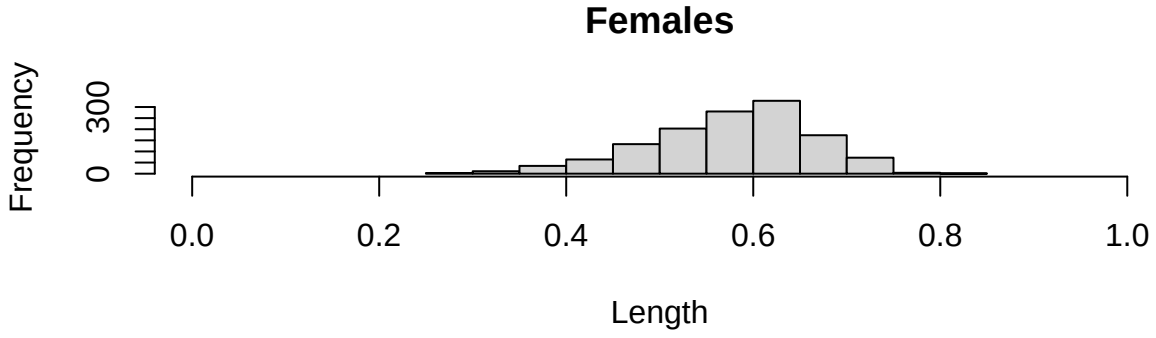
```
pdf(file = "output histograms of abalone data.pdf")
par(mfrow = c(2,1))
hist(abalone[abalone$Sex == "M", 2], xlim = c(0, 1), xlab = "Length",
     main = "Males")
hist(abalone[abalone$Sex == "F", 2], xlim = c(0, 1), xlab = "Length",
     main = "Females")
dev.off()
```

```
## pdf
##   2
```

9. savePlot()

```
par(mfrow = c(2,1))
hist(abalone[abalone$Sex == "M", 2], xlim = c(0, 1), xlab = "Length",
     main = "Males")
hist(abalone[abalone$Sex == "F", 2], xlim = c(0, 1), xlab = "Length",
     main = "Females")
```






```
# savePlot("two histograms.png", type = "png")
```