Chapter 16 Answers

Highlights

- Review these functions
  - data.frame
  - head
  - tail
  - summary
  - transform
  - subset

16.1 What is returned by the follwing R commands? (Waking hours from wikipedia.)

```
creatures = c("dog","cat","armadillo","human")
friendly = c(TRUE,TRUE,FALSE,TRUE)
diet = c("cats","mice","termites","Twinkies(tm)")
waking.hours = c(13.9, 11.5, 5.9, 16.0)
creature.data = data.frame (friendly, diet, waking.hours, row.names=creatures)
creature.data
```

```
##           friendly         diet waking.hours
## dog           TRUE         cats         13.9
## cat           TRUE         mice         11.5
## armadillo    FALSE     termites          5.9
## human         TRUE Twinkies(tm)         16.0
```

```
creatures[creature.data$waking.hours < 12 & !creature.data$friendly]
```

```
## [1] "armadillo"
```

16.2 Write a single R command that alphabetizes the rows of the data frame `creature.data` from Exercise 16.1 by creature name.

```
creature.data[sort(creatures),]
```

```
##           friendly         diet waking.hours
## armadillo    FALSE     termites          5.9
## cat           TRUE         mice         11.5
## dog           TRUE         cats         13.9
## human         TRUE Twinkies(tm)         16.0
```

```
# also (perhaps the separate object creatures was not created)
 creature.data[sort(row.names(creature.data)),]
```

```
##           friendly         diet waking.hours
## armadillo    FALSE     termites          5.9
## cat           TRUE         mice         11.5
## dog           TRUE         cats         13.9
## human         TRUE Twinkies(tm)         16.0
```

```
# for fun:
summary(creature.data)
```

```
##   friendly                diet    waking.hours
##  Mode :logical   cats        :1   Min.   : 5.90
##  FALSE:1         mice        :1   1st Qu.:10.10
##  TRUE :3         termites    :1   Median :12.70
```

```
##                   Twinkies(tm):1   Mean   :11.82
##                                    3rd Qu.:14.43
##                                    Max.   :16.00
```

16.3 Use the R `subset` function to create a data frame consisting of just the creature name and diet associated with friendly creatures who are awake more than 12 hours a day from the data drame `creature.data` from Exercise 16.1.

```
subset(creature.data,friendly & (waking.hours > 12),c(diet))
```

```
##               diet
## dog           cats
## human Twinkies(tm)
```

16.4 Consider the data frame ‘`creature.data` from Exercise 16.1.

- Extract the waking hours for a dog using two different R commands

```
creature.data["dog","waking.hours"]
```

```
## [1] 13.9
```

```
# subset extracts a new data frame,
# adding the [1,1] indexes extracts the cell value
subset(creature.data,"dog" == row.names(creature.data),c(waking.hours))[1,1]
```

```
## [1] 13.9
```

- Extract the waking hours for all creatures using two different R commands

```
creature.data$waking.hours
```

```
## [1] 13.9 11.5  5.9 16.0
```

```
creature.data[["waking.hours"]]
```

```
## [1] 13.9 11.5  5.9 16.0
```

```
subset(creature.data,TRUE,c(waking.hours))[,1]
```

```
## [1] 13.9 11.5  5.9 16.0
```

- Create a data frame that consists only of the rows for dogs and armadillos using two different R commands

```
subset(creature.data,"dog" == row.names(creature.data) | "armadillo" == row.names(creature.data),colnam
```

```
##           friendly     diet waking.hours
## dog           TRUE     cats         13.9
## armadillo    FALSE termites          5.9
```

```
subset(creature.data,"dog" == row.names(creature.data) | "armadillo" == row.names(creature.data),)
```

```
##           friendly     diet waking.hours
## dog           TRUE     cats         13.9
## armadillo    FALSE termites          5.9
```

```
subset(creature.data,(waking.hours < 16 & waking.hours>12) | waking.hours<6,)
```

```
##           friendly     diet waking.hours
## dog           TRUE     cats         13.9
## armadillo    FALSE termites          5.9
```

- Create a data frame that consists only of friendly creatures

```
subset(creature.data,friendly,)
```

```
##       friendly         diet waking.hours
## dog       TRUE         cats         13.9
## cat       TRUE         mice         11.5
## human     TRUE Twinkies(tm)         16.0
```

16.5 Consider a data frame named `a` with 4 rows and 3 columns populated with positive integers and -1 where -1 denotes a missing value. Write an R command that replaces each -1 with NA.

```
a = data.frame(col1=1:4,col2=seq(10,40,by=10),col3=seq(100,400,by=100))
a
```

```
##   col1 col2 col3
## 1    1   10  100
## 2    2   20  200
## 3    3   30  300
## 4    4   40  400
```

```
a[1,1] = a[2,2] = a[3,3] = a$col2[4] = -1
a
```

```
##   col1 col2 col3
## 1   -1   10  100
## 2    2   -1  200
## 3    3   30   -1
## 4    4   -1  400
```

```
a[a==-1] <- NA
a
```

```
##   col1 col2 col3
## 1   NA   10  100
## 2    2   NA  200
## 3    3   30   NA
## 4    4   NA  400
```

16.6 Create the objects `v`, `m`, `a`, `l`, and `d` as a vector, matrix, array, list, and data.frame. Apply the functions `class`, `typeof`, and `mode`to each of the objects and summarize the results in a table.

```
v = 1:12
m = matrix(v,3,4)
a = array(v,c(2,2,3))
l = list(v,m,a)
d = subset(creature.data,friendly,)

st = data.frame(
    class  =    c(  class(v),   class(m),   class(a),   class(l),   class(d)),
    typeof =    c(  typeof(v),  typeof(m),  typeof(a),  typeof(l),  typeof(d)),
    mode   =    c(  mode(v),    mode(m),    mode(a),    mode(l),    mode(d)),
    row.names = c(  "v",        "m",        "a",        "l",        "d")
)
st
```

```
##        class  typeof    mode
## v    integer integer numeric
## m     matrix integer numeric
## a      array integer numeric
```

```
## l       list    list    list
## d data.frame    list    list
```

16.7 Create a data frame named `w` that consists of the following three named columns:

- `x`, the first four positve integers,
- `y`, the abbreviations of the first four months,
- `z`, the first names of the Beatles.

```
w = data.frame( x = 1:4,
                y = month.abb[1:4],
                z = sort(c("John","Paul","Ringo","George"))
)
w
```

```
##   x   y      z
## 1 1 Jan George
## 2 2 Feb   John
## 3 3 Mar   Paul
## 4 4 Apr  Ringo
```

Remove the second column of `w` in the following two fashions:

- set `w$y` to `NULL`, that is, `w$y = NULL`,
- subset the columns to keep, that is, `w[c("x","z")]`.

```
w_ = w
w_$y = NULL

w__ = w
w__[c("x","z")]
```

```
##   x      z
## 1 1 George
## 2 2   John
## 3 3   Paul
## 4 4  Ringo
```

Show that the first technique alters the data frame `w`but the second technique does not alter the data frame `w`.

```
# the first method was an assignment operation, resulting in a change
w_
```

```
##   x      z
## 1 1 George
## 2 2   John
## 3 3   Paul
## 4 4  Ringo
```

```
# the second method was an extraction operation, with no permanent damage done to the data frame
w__
```

```
##   x   y      z
## 1 1 Jan George
## 2 2 Feb   John
## 3 3 Mar   Paul
## 4 4 Apr  Ringo
```