

Intro to R - Part II

Data sampling

`sample()` function takes a sample of the specified size from the elements of `x`.

```
# create a vector x with values from 1 to 10
x <- 1:10

# create a sample of size 5 from the vector x
sample(x, size = 5)

## [1] 4 6 9 1 8
```

If we want a sample with a size greater than the original vector, we need to pass argument `replace=TRUE`.

```
# create a sample of size 20 from the vector x, where duplicates are allowed
sample(x, size = 20, replace = TRUE)

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

Matrices

Matrices are objects with elements arranged in a two-dimensional rectangular layout. They contain elements of the same type. A matrix is created with a function `matrix()`. Similar to vectors, elements are indexed and a specific element can be retrieved by its index. `nrow()` returns number of rows, and `ncol()` returns number of columns in a matrix.

```
# create a 2 x 4 matrix with values from 8 to 1, filled by rows
a <- matrix(8:1, nrow = 2, ncol = 4, byrow = TRUE)
a

##      [,1] [,2] [,3] [,4]
## [1,]    8    7    6    5
## [2,]    4    3    2    1

# get the first row
a[1, ]

## [1] 8 7 6 5

# get the element from the row 1, column 2
a[1,2]

## [1] 7
```

```
# get number of rows
nrow(a)

## [1] 2

# get number of columns
ncol(a)

## [1] 4
```

All matrix operations can be applied.

```
# create two matrices of the same dimension
matrix1 <- matrix(c(3, 9, -1, 4, 2, 6), nrow = 2)
matrix1

##      [,1] [,2] [,3]
## [1,]    3   -1    2
## [2,]    9    4    6

matrix2 <- matrix(c(5, 2, 0, 9, 3, 4), nrow = 2)
matrix2

##      [,1] [,2] [,3]
## [1,]    5    0    3
## [2,]    2    9    4

# add matrix2 to matrix1
matrix1 + matrix2

##      [,1] [,2] [,3]
## [1,]    8   -1    5
## [2,]   11   13   10
```

Transposing a matrix can be achieved via the `t()` function.

```
# transpose a matrix
t(matrix1)

##      [,1] [,2]
## [1,]    3    9
## [2,]   -1    4
## [3,]    2    6
```

Lists

A list is an ordered collection of elements that can be of different types, such as vectors, functions and even another lists. A list is created by using the function `list()`. Specific element can be accessed by its index or its name. `length()` returns number of elements in a list.

```

# create a new list with attributes: adult, passport, age
traveler1 <- list(adult = TRUE, passport = "P212123", age = 34)
traveler1

## $adult
## [1] TRUE
##
## $passport
## [1] "P212123"
##
## $age
## [1] 34

# get the 3rd element
traveler1[3]

## $age
## [1] 34

# get value of the 3rd element
traveler1[[3]]

## [1] 34

# get value of the age element
traveler1$age

## [1] 34

# get the list length
length(traveler1)

## [1] 3

```

When the concatenation function `c()` is given list arguments, the result is also a list containing all elements from the passed lists joined in a sequence.

```

# concatenate two lists
traveler2 <- list(adult = FALSE, passport = "P4567756", age = 14)

travelers <- c(traveler1, traveler2)
travelers

## $adult
## [1] TRUE
##
## $passport
## [1] "P212123"
##
## $age
## [1] 34
##

```

```
## $adult
## [1] FALSE
##
## $passport
## [1] "P4567756"
##
## $age
## [1] 14
```

`append()` function is similar to the `c()` function. But `append()` is different in the sense that it allows for values to be inserted into a vector after a certain position.

```
# add new List after the 2nd element
traveler1 <- append(traveler1, list(country = "AUS"), after=2)
length(traveler1)

## [1] 4

traveler1

## $adult
## [1] TRUE
##
## $passport
## [1] "P212123"
##
## $country
## [1] "AUS"
##
## $age
## [1] 34
```

`is.list()` returns TRUE if an object is of type list.

```
# check if travelers is a list
is.list(travelers)

## [1] TRUE
```

`names()` function retrieves names of all list elements.

```
# get names of all list elements
names(travelers)

## [1] "adult"      "passport" "age"      "adult"      "passport" "age"

# get elements with name 'age'
travelers[names(travelers) == "age"]

## $age
## [1] 34
##
```

```
## $age
## [1] 14
```

An element is deleted by assigning NULL to it.

```
# delete 3rd element
traveler1[[3]] <- NULL
length(traveler1)

## [1] 3

traveler1

## $adult
## [1] TRUE
##
## $passport
## [1] "P212123"
##
## $age
## [1] 34
```

Loops and branching

For each loop

Iterates through each element of the provided vector. *break* stops the loop, while *next* stops the current iteration.

```
# print all odd numbers from 1 to 10 using for each loop
for (i in 1:10) {
  if (i %% 2 == 1) {
    print(paste(i, "is odd number"))
  }
}

## [1] "1 is odd number"
## [1] "3 is odd number"
## [1] "5 is odd number"
## [1] "7 is odd number"
## [1] "9 is odd number"
```

While loop

```
# print all odd numbers from 1 to 10 using while loop
i <- 1
while (i <= 10) {
  if (i %% 2 == 1) {
    print(paste(i, "is odd number"))
  }
}
```

```

    i <- i + 1
  }

## [1] "1 is odd number"
## [1] "3 is odd number"
## [1] "5 is odd number"
## [1] "7 is odd number"
## [1] "9 is odd number"

```

if-else

use ifelse function to assign a new attribute to traveler1 called 'request' with the value 'assistance required' if a traveler is younger than 5 years, and the value 'no special requests' otherwise

```

traveler1$request <- ifelse(test = traveler1$age < 5,
                             yes = "assistance required",
                             no = "no special requests")

```

```

traveler1

## $adult
## [1] TRUE
##
## $passport
## [1] "P212123"
##
## $age
## [1] 34
##
## $request
## [1] "no special requests"

```

Task 1

From a 2 x 3 matrix with the following elements: 3, 9, -1, 4, 2, 6, print only values from the first row that are positive.

Answer:

```

matrix1 <- matrix(c(3, 9, -1, 4, 2, 6), nrow = 2)

for (i in matrix1[1,]) {
  if (i > 0) {
    print(i)
  }
}

## [1] 3
## [1] 2

```

User-defined functions and apply

The structure of a function is given below.

```
myfunction <- function(arg1, arg2, ... ){  
  statements  
  return(object)  
}
```

The last expression evaluated in a function is a return value.

```
# create a function that adds two numbers. The default value for the second argument is 1  
add <- function(x, y = 1){  
  x + y  
}  
  
add(2)  
## [1] 3  
  
add(2, 3)  
## [1] 5
```

`return(value)` stops the execution of a function and returns a value.

```
# create a function returning an absolute value of x. Return the result using the return() function  
my_abs <- function(x) {  
  if (x > 0) {  
    return(x)  
  }  
  return(-x)  
}  
  
my_abs(5)  
## [1] 5  
  
my_abs(-5)  
## [1] 5
```

Applying a function over rows and columns in data frame

We can apply a custom function to a data frame (similar can be done with vectors, lists and matrices). `apply()` function accepts data as the first argument. The second argument is called `MARGIN` and it defines how the function is applied. If `MARGIN=1`, it applies over rows, whereas with `MARGIN=2`, it works over columns. When `MARGIN=c(1,2)`, it applies to both rows and columns.

```

# Load the data "The Beatles songs dataset, v1.csv"
the.beatles.songs <- read.csv("The Beatles songs dataset, v1.csv", stringsAsFactors = FALSE)

# call print() function over titles of the first 10 songs
apply(the.beatles.songs[1:10, 1, drop=FALSE], 1, print)

##           Title
## "12-Bar Original"
##           Title
## "A Day in the Life"
##           Title
## "A Hard Day's Night"
##           Title
## "A Shot of Rhythm and Blues"
##           Title
## "A Taste of Honey"
##           Title
## "Across the Universe"
##           Title
## "Act Naturally"
##           Title
## "Ain't She Sweet"
##           Title
## "All I've Got to Do"
##           Title
## "All My Loving"

##           1           2
## "12-Bar Original" "A Day in the Life"
##           3           4
## "A Hard Day's Night" "A Shot of Rhythm and Blues"
##           5           6
## "A Taste of Honey" "Across the Universe"
##           7           8
## "Act Naturally" "Ain't She Sweet"
##           9          10
## "All I've Got to Do" "All My Loving"

# calculate mean value of the duration and Top.50.Billboard values of all songs from 1963
apply(the.beatles.songs[the.beatles.songs$Year == 1963, c(4,9)], 2, mean)

##           Duration Top.50.Billboard
##           NA           NA

# calculate mean value of the duration and Top.50.Billboard values that are not NAs of all songs from 1963
mean.with.na <- function(x) {
  mean(x, na.rm = TRUE)
}

```

```
apply(the.beatles.songs[the.beatles.songs$Year == 1963, c(4,9)], 2, mean.with
.na)
```

```
##          Duration Top.50.Billboard
##          134.9016          21.0000
```

Working with tables

`table()` builds a contingency table of the counts at each attribute value.

```
# create a contingency table of column Year values
```

```
table(the.beatles.songs$Year)
```

```
##
##      1958      1960      1961      1962      1963      1964      1965
##        2        4        3       20       66       41       37
##      1966      1967      1968      1969      196?      1970 1977/1994
##       19       27       45       42        1        1        1
## 1980/1995
##        1
```

```
# get the 4th element from the table
```

```
table(the.beatles.songs$Year)[4]
```

```
## 1962
```

```
##    20
```

```
# store the 4th element from the table in a variable
```

```
x <- table(the.beatles.songs$Year)[4]
```

```
x
```

```
## 1962
```

```
##    20
```

```
# convert the variable to numeric
```

```
y <- as.numeric(x)
```

```
y
```

```
## [1] 20
```

Table of proportions can be obtained by using the `prop.table()` function.

```
# get the proportions table for the values of the Year column
```

```
prop.table(table(the.beatles.songs$Year))
```

```
##
##      1958      1960      1961      1962      1963      1964
## 0.006451613 0.012903226 0.009677419 0.064516129 0.212903226 0.132258065
##      1965      1966      1967      1968      1969      196?
## 0.119354839 0.061290323 0.087096774 0.145161290 0.135483871 0.003225806
```

```
##      1970      1977/1994      1980/1995
## 0.003225806 0.003225806 0.003225806

# get the proportions table for the values of the Year column, but limiting number of digits to 2
round(prop.table(table(the.beatles.songs$Year)), digits = 2)

##
##      1958      1960      1961      1962      1963      1964      1965
##      0.01      0.01      0.01      0.06      0.21      0.13      0.12
##      1966      1967      1968      1969      196?      1970 1977/1994
##      0.06      0.09      0.15      0.14      0.00      0.00      0.00
## 1980/1995
##      0.00
```

`xtabs()` creates a contingency table using formula style input.

```
# create a contingency table Top.50.Billboard vs. Year
xtabs(~Top.50.Billboard + Year, the.beatles.songs)

##
##      Year
## Top.50.Billboard 1961 1962 1963 1964 1965 1966 1967 1968 1969 1977/1994
##      1      0      0      0      0      0      0      0      1      0      0
##      2      0      0      1      0      0      0      0      0      0      0
##      3      0      0      1      0      0      0      0      0      0      0
##      4      0      0      0      0      0      0      0      0      1      0
##      5      0      0      0      0      0      0      0      0      1      0
##      6      0      0      0      0      0      0      0      0      1      0
##      7      0      0      0      0      0      0      1      0      0      0
##      8      0      0      0      1      0      0      0      0      0      0
##      9      0      0      0      0      1      0      0      0      0      0
##     10      0      0      0      1      0      0      0      0      0      0
##     11      0      0      0      1      0      0      0      0      0      0
##     12      0      0      0      0      1      0      0      0      0      0
##     13      0      0      1      0      0      0      0      0      0      0
##     14      0      0      0      0      1      0      0      0      0      0
##     15      0      0      0      0      0      0      1      0      0      0
##     16      0      1      0      0      0      0      0      0      0      0
##     17      0      0      0      0      1      0      0      0      0      0
##     18      0      1      0      0      0      0      0      0      0      0
##     19      0      0      0      0      0      1      0      0      0      0
##     20      0      0      0      0      0      0      0      0      1      0
##     21      0      0      0      1      0      0      0      0      0      0
##     22      0      0      0      0      0      0      0      1      0      0
##     23      0      0      0      0      0      1      0      0      0      0
##     24      0      0      0      0      0      1      0      0      0      0
##     25      0      0      0      0      0      1      0      0      0      0
##     26      0      0      1      0      0      0      0      0      0      0
##     27      0      0      0      0      1      0      0      0      0      0
##     28      0      0      0      0      0      0      0      0      1      0
##     29      0      0      0      0      1      0      0      0      0      0
```

##	30	0	0	0	0	0	0	0	0	1	0
##	31	0	0	0	1	0	0	0	0	0	0
##	32	0	0	0	0	0	0	0	1	0	0
##	33	0	0	0	0	0	1	0	0	0	0
##	34	0	1	0	0	0	0	0	0	0	0
##	36	0	0	1	0	0	0	0	0	0	0
##	37	0	0	0	1	0	0	0	0	0	0
##	38	0	0	0	0	0	1	0	0	0	0
##	39	0	0	0	0	0	0	0	0	0	1
##	40	0	0	0	1	0	0	0	0	0	0
##	41	1	0	0	0	0	0	0	0	0	0
##	42	0	0	0	0	0	1	0	0	0	0
##	43	0	0	0	1	0	0	0	0	0	0
##	44	0	0	0	1	0	0	0	0	0	0
##	45	1	0	0	0	0	0	0	0	0	0
##	46	0	0	1	0	0	0	0	0	0	0
##	47	0	0	0	0	0	0	0	0	0	0
##	48	0	0	0	0	0	0	1	0	0	0
##	49	0	0	0	1	0	0	0	0	0	0
##	50	0	0	0	0	1	0	0	0	0	0

Year

Top.50.Billboard 1980/1995

##	1	0
##	2	0
##	3	0
##	4	0
##	5	0
##	6	0
##	7	0
##	8	0
##	9	0
##	10	0
##	11	0
##	12	0
##	13	0
##	14	0
##	15	0
##	16	0
##	17	0
##	18	0
##	19	0
##	20	0
##	21	0
##	22	0
##	23	0
##	24	0
##	25	0
##	26	0
##	27	0
##	28	0

```
##          29          0
##          30          0
##          31          0
##          32          0
##          33          0
##          34          0
##          36          0
##          37          0
##          38          0
##          39          0
##          40          0
##          41          0
##          42          0
##          43          0
##          44          0
##          45          0
##          46          0
##          47          1
##          48          0
##          49          0
##          50          0
```

Manipulating data frames

Adding new rows and columns

A column can be added by assigning values to a new column name in the data frame.

```
# set On.album attribute for all songs to FALSE
the.beatles.songs$On.album <- FALSE
```

```
head(the.beatles.songs)
```

```
##          Title Year
## 1      12-Bar Original 1965
## 2      A Day in the Life 1967
## 3      A Hard Day's Night 1964
## 4 A Shot of Rhythm and Blues 1963
## 5      A Taste of Honey 1963
## 6      Across the Universe 1968
##
##          Album.debut Duration Other.releases
## 1              Anthology 2      174          NA
## 2  Sgt. Pepper's Lonely Hearts Club Band    335          12
## 3      UK: A Hard Day's Night US: 1962-1966    152          35
## 4              Live at the BBC     104          NA
## 5  UK: Please Please Me US: The Early Beatles    163          29
## 6              Let It Be     230          19
##
##          Genre
## 1          Blues
```

```
## 2 Psychedelic Rock, Art Rock, Pop/Rock
## 3          Rock, Electronic, Pop/Rock
## 4          R&B, Pop/Rock
## 5          Pop/Rock, Jazz, Stage&Screen
## 6          Psychedelic folk, Pop/Rock
##          Songwriter          Lead.vocal
## 1 Lennon, McCartney, Harrison and Starkey
## 2          Lennon and McCartney Lennon and McCartney
## 3          Lennon Lennon, with McCartney
## 4          Thompson          Lennon
## 5          Scott, Marlow          McCartney
## 6          Lennon          Lennon
## Top.50.Billboard On.album
## 1          NA FALSE
## 2          NA FALSE
## 3          8 FALSE
## 4          NA FALSE
## 5          NA FALSE
## 6          NA FALSE
```

By using the `cbind()` function, you can join two data frames by columns

```
# create a data frame with one column and 310 rows, filled with NAs
na.matrix <- matrix(NA, nrow = 310, ncol = 1)
publishers <- data.frame(Publisher = na.matrix)
```

```
# combine two data frames
```

```
the.beatles.songs <- cbind(the.beatles.songs, publishers)
head(the.beatles.songs)
```

```
##          Title Year
## 1      12-Bar Original 1965
## 2      A Day in the Life 1967
## 3      A Hard Day's Night 1964
## 4 A Shot of Rhythm and Blues 1963
## 5      A Taste of Honey 1963
## 6      Across the Universe 1968
##          Album.debut Duration Other.releases
## 1          Anthology 2      174          NA
## 2      Sgt. Pepper's Lonely Hearts Club Band      335      12
## 3      UK: A Hard Day's Night US: 1962-1966      152      35
## 4          Live at the BBC      104          NA
## 5 UK: Please Please Me US: The Early Beatles      163      29
## 6          Let It Be      230      19
##          Genre
## 1          Blues
## 2 Psychedelic Rock, Art Rock, Pop/Rock
## 3          Rock, Electronic, Pop/Rock
## 4          R&B, Pop/Rock
## 5          Pop/Rock, Jazz, Stage&Screen
```

```
## 6          Psychedelic folk, Pop/Rock
##                               Songwriter          Lead.vocal
## 1 Lennon, McCartney, Harrison and Starkey
## 2          Lennon and McCartney    Lennon and McCartney
## 3                               Lennon Lennon, with McCartney
## 4                               Thompson          Lennon
## 5          Scott, Marlow          McCartney
## 6          Lennon          Lennon
## Top.50.Billboard On.album Publisher
## 1          NA      FALSE      NA
## 2          NA      FALSE      NA
## 3          8       FALSE      NA
## 4          NA      FALSE      NA
## 5          NA      FALSE      NA
## 6          NA      FALSE      NA
```

Rows are added by using the `rbind()` function.

```
# get the first song
new.song <- data.frame(the.beatles.songs[1, ])

# add the song to the end of the data frame
the.beatles.songs <- rbind(the.beatles.songs, new.song)
tail(the.beatles.songs)

##                               Title Year
## 306   You're Going to Lose That Girl 1965
## 307 You've Got to Hide Your Love Away 1965
## 308   You've Really Got a Hold on Me 1963
## 309                               Young Blood 1963
## 310   Your Mother Should Know 1967
## 311   12-Bar Original 1965
##                               Album.debut Duration
## 306                               Help!      140
## 307                               Help!      131
## 308 UK: With the Beatles US: The Beatles Second Album      182
## 309                               Live at the BBC      116
## 310                               Magical Mystery Tour      149
## 311                               Anthology 2      174
##   Other.releases                               Genre
## 306              6                               Rock, Pop/Rock
## 307             12                               FolkPop/Rock
## 308              2                               Soul, Pop/Rock
## 309             NA                               Pop/Rock
## 310             13 Music Hall, Vaudeville Rock, Psychedelic Pop, Pop/Rock
## 311             NA                               Blues
##                               Songwriter          Lead.vocal
## 306                               Lennon          Lennon
## 307                               Lennon          Lennon
## 308                               Robinson Lennon and Harrison
```

```

## 309                Leiber, Stoller                Harrison
## 310                McCartney                McCartney
## 311 Lennon, McCartney, Harrison and Starkey
##      Top.50.Billboard On.album Publisher
## 306                NA      FALSE      NA
## 307                NA      FALSE      NA
## 308                NA      FALSE      NA
## 309                NA      FALSE      NA
## 310                NA      FALSE      NA
## 311                NA      FALSE      NA

# add the song after the 3rd song in the data frame
the.beatles.songs <- rbind(the.beatles.songs[1:3, ],
                           new.song,
                           the.beatles.songs[4:nrow(the.beatles.songs), ])
head(the.beatles.songs)

##                Title Year
## 1          12-Bar Original 1965
## 2          A Day in the Life 1967
## 3          A Hard Day's Night 1964
## 4          12-Bar Original 1965
## 410 A Shot of Rhythm and Blues 1963
## 5          A Taste of Honey 1963
##                Album.debut Duration Other.releases
## 1                Anthology 2      174             NA
## 2          Sgt. Pepper's Lonely Hearts Club Band      335             12
## 3          UK: A Hard Day's Night US: 1962-1966      152             35
## 4                Anthology 2      174             NA
## 410                Live at the BBC      104             NA
## 5          UK: Please Please Me US: The Early Beatles      163             29
##                Genre
## 1                Blues
## 2          Psychedelic Rock, Art Rock, Pop/Rock
## 3                Rock, Electronic, Pop/Rock
## 4                Blues
## 410                R&B, Pop/Rock
## 5          Pop/Rock, Jazz, Stage&Screen
##                Songwriter                Lead.vocal
## 1          Lennon, McCartney, Harrison and Starkey
## 2                Lennon and McCartney          Lennon and McCartney
## 3                Lennon          Lennon, with McCartney
## 4          Lennon, McCartney, Harrison and Starkey
## 410                Thompson                Lennon
## 5                Scott, Marlow                McCartney
##      Top.50.Billboard On.album Publisher
## 1                NA      FALSE      NA
## 2                NA      FALSE      NA
## 3                8      FALSE      NA
## 4                NA      FALSE      NA

```

```
## 410      NA      FALSE      NA
## 5        NA      FALSE      NA
```

Removing columns and rows

A column is removed by assigning a NULL to it.

```
# remove the attribute On.album
the.beatles.songs$On.album <- NULL
names(the.beatles.songs)

## [1] "Title"      "Year"      "Album.debut"
## [4] "Duration"   "Other.releases" "Genre"
## [7] "Songwriter" "Lead.vocal" "Top.50.Billboard"
## [10] "Publisher"
```

Another way of removing columns is to form a set of the columns you want to remove, and keep the complement of that set. The complement of a set is given by the '-' operator.

```
# remove columns Top.50.Billboard (at index 9) and Publisher (at index 10)
the.beatles.songs1 <- the.beatles.songs[,-c(9, 10)]
names(the.beatles.songs1)

## [1] "Title"      "Year"      "Album.debut" "Duration"
## [5] "Other.releases" "Genre"     "Songwriter"  "Lead.vocal"
```

Using the same method, rows can be removed.

```
# create a subset of the data frame without songs in rows 2, 4 and 6
the.beatles.songs2 <- the.beatles.songs[-c(2, 4, 6), ]
head(the.beatles.songs2)

##           Title Year      Album.debut
## 1      12-Bar Original 1965 Anthology 2
## 3      A Hard Day's Night 1964 UK: A Hard Day's Night US: 1962-1966
## 410 A Shot of Rhythm and Blues 1963 Live at the BBC
## 6      Across the Universe 1968 Let It Be
## 7      Act Naturally 1965 UK: Help! US: Yesterday and Today
## 8      Ain't She Sweet 1961 Anthology 1
##      Duration Other.releases      Genre
## 1          174           NA      Blues
## 3          152           35 Rock, Electronic, Pop/Rock
## 410         104           NA      R&B, Pop/Rock
## 6          230           19 Psychedelic folk, Pop/Rock
## 7          139           14      Country, Pop/Rock
## 8           NA           9      Pop/Rock
##           Songwriter      Lead.vocal
## 1 Lennon, McCartney, Harrison and Starkey
## 3           Lennon Lennon, with McCartney
## 410           Thompson      Lennon
## 6           Lennon      Lennon
```

```
## 7 Russell, Morrison Starkey
## 8 Yellen, Ager Lennon
## Top.50.Billboard Publisher
## 1 NA NA
## 3 8 NA
## 410 NA NA
## 6 NA NA
## 7 50 NA
## 8 41 NA

# create a subset of the data frame without songs in rows from 1 to 8
the.beatles.songs3 <- the.beatles.songs[-(1:8), ]
head(the.beatles.songs3)

## Title Year Album.debut
## 8 Ain't She Sweet 1961 Anthology 1
## 9 All I've Got to Do 1963 UK: With the Beatles US: Meet The Beatles!
## 10 All My Loving 1963 UK: With the Beatles US: Meet The Beatles!
## 11 All Things Must Pass 1969 Anthology 3
## 12 All Together Now 1967 Yellow Submarine
## 13 All You Need Is Love 1967 Magical Mystery Tour
## Duration Other.releases Genre Songwriter
## 8 NA 9 Pop/Rock Yellen, Ager
## 9 124 9 Pop/Rock Lennon
## 10 124 32 Pop/Rock McCartney
## 11 227 NA Folk Rock, Pop/Rock Harrison
## 12 130 8 Skiffle, Pop/Rock McCartney, with Lennon
## 13 237 25 Pop/Rock Lennon
## Lead.vocal Top.50.Billboard Publisher
## 8 Lennon 41 NA
## 9 Lennon NA NA
## 10 McCartney NA NA
## 11 Harrison NA NA
## 12 McCartney, with Lennon NA NA
## 13 Lennon 15 NA
```

Updating column and row names

`colnames()` function returns all column names. A column name is changed by assigning a new name to it.

```
# get column names
colnames(the.beatles.songs)

## [1] "Title" "Year" "Album.debut"
## [4] "Duration" "Other.releases" "Genre"
## [7] "Songwriter" "Lead.vocal" "Top.50.Billboard"
## [10] "Publisher"

# change name of the column 'Genre' to 'Song.genre'
genreIndex <- which(colnames(the.beatles.songs) == "Genre")
```

```

colnames(the.beatles.songs)[genreIndex] <- "Song.genre"
colnames(the.beatles.songs)

## [1] "Title"          "Year"          "Album.debut"
## [4] "Duration"       "Other.releases" "Song.genre"
## [7] "Songwriter"     "Lead.vocal"    "Top.50.Billboard"
## [10] "Publisher"

# change name of the column at the index 6 to 'Genre'
colnames(the.beatles.songs)[6] <- "Genre"
colnames(the.beatles.songs)

## [1] "Title"          "Year"          "Album.debut"
## [4] "Duration"       "Other.releases" "Genre"
## [7] "Songwriter"     "Lead.vocal"    "Top.50.Billboard"
## [10] "Publisher"

```

`rownames()` function returns all row names. A row name is changed by assigning a new name to it.

```

# change row names to a string containing word 'song' and a song order number
rownames(the.beatles.songs) <- paste("song", 1:nrow(the.beatles.songs))
head(the.beatles.songs)

##              Title Year
## song 1      12-Bar Original 1965
## song 2      A Day in the Life 1967
## song 3      A Hard Day's Night 1964
## song 4      12-Bar Original 1965
## song 5 A Shot of Rhythm and Blues 1963
## song 6      A Taste of Honey 1963
##              Album.debut Duration Other.releases
## song 1              Anthology 2      174          NA
## song 2      Sgt. Pepper's Lonely Hearts Club Band      335          12
## song 3      UK: A Hard Day's Night US: 1962-1966      152          35
## song 4              Anthology 2      174          NA
## song 5              Live at the BBC      104          NA
## song 6      UK: Please Please Me US: The Early Beatles      163          29
##              Genre
## song 1              Blues
## song 2      Psychedelic Rock, Art Rock, Pop/Rock
## song 3              Rock, Electronic, Pop/Rock
## song 4              Blues
## song 5              R&B, Pop/Rock
## song 6      Pop/Rock, Jazz, Stage&Screen
##              Songwriter              Lead.vocal
## song 1      Lennon, McCartney, Harrison and Starkey
## song 2              Lennon and McCartney      Lennon and McCartney
## song 3              Lennon      Lennon, with McCartney
## song 4      Lennon, McCartney, Harrison and Starkey
## song 5              Thompson              Lennon

```

```
## song 6 Scott, Marlow McCartney
## Top.50.Billboard Publisher
## song 1 NA NA
## song 2 NA NA
## song 3 8 NA
## song 4 NA NA
## song 5 NA NA
## song 6 NA NA
```

change row names to a string containing order number

```
rownames(the.beatles.songs) <- c(1:nrow(the.beatles.songs))
head(the.beatles.songs)
```

```
## Title Year
## 1 12-Bar Original 1965
## 2 A Day in the Life 1967
## 3 A Hard Day's Night 1964
## 4 12-Bar Original 1965
## 5 A Shot of Rhythm and Blues 1963
## 6 A Taste of Honey 1963
## Album.debut Duration Other.releases
## 1 Anthology 2 174 NA
## 2 Sgt. Pepper's Lonely Hearts Club Band 335 12
## 3 UK: A Hard Day's Night US: 1962-1966 152 35
## 4 Anthology 2 174 NA
## 5 Live at the BBC 104 NA
## 6 UK: Please Please Me US: The Early Beatles 163 29
## Genre
## 1 Blues
## 2 Psychedelic Rock, Art Rock, Pop/Rock
## 3 Rock, Electronic, Pop/Rock
## 4 Blues
## 5 R&B, Pop/Rock
## 6 Pop/Rock, Jazz, Stage&Screen
## Songwriter Lead.vocal
## 1 Lennon, McCartney, Harrison and Starkey
## 2 Lennon and McCartney Lennon and McCartney
## 3 Lennon Lennon, with McCartney
## 4 Lennon, McCartney, Harrison and Starkey
## 5 Thompson Lennon
## 6 Scott, Marlow McCartney
## Top.50.Billboard Publisher
## 1 NA NA
## 2 NA NA
## 3 8 NA
## 4 NA NA
## 5 NA NA
## 6 NA NA
```

Retrieving and changing values

Parts of a data frame can be selected in different ways.

```
# get songs in rows from 1 to 5, but only attributes Title and Album.debut
first.songs <- the.beatles.songs[1:5, c("Title", "Album.debut")]
first.songs
```

	Title	Album.debut
## 1	12-Bar Original	Anthology 2
## 2	A Day in the Life	Sgt. Pepper's Lonely Hearts Club Band
## 3	A Hard Day's Night	UK: A Hard Day's Night US: 1962-1966
## 4	12-Bar Original	Anthology 2
## 5	A Shot of Rhythm and Blues	Live at the BBC

```
# get the songs from year 1964 not having McCartney as a Lead vocal
indexes <- which((the.beatles.songs$Year == "1964") & (the.beatles.songs$Lead.vocal != "McCartney"))
selected.songs <- the.beatles.songs[indexes, ]
head(selected.songs)
```

	Title	Year	Album.debut
## 3	A Hard Day's Night	1964	UK: A Hard Day's Night US: 1962-1966
## 19	Any Time at All	1964	UK: A Hard Day's Night US: Something New
## 22	Baby's in Black	1964	UK: Beatles for Sale US: Beatles '65
## 36	Can't Buy Me Love	1964	UK: A Hard Day's Night US: Hey Jude
## 64	Eight Days a Week	1964	UK: Beatles for Sale US: Beatles VI
## 67	Every Little Thing	1964	UK: Beatles for Sale US: Beatles VI

	Duration	Other.releases	Genre
## 3	152	35	Rock, Electronic, Pop/Rock
## 19	133	15	Pop/Rock
## 22	122	20	Folk Rock, Pop/Rock
## 36	131	42	Pop/Rock
## 64	164	23	Rock, Pop/Rock
## 67	121	9	Rock, Pop/Rock

	Songwriter	Lead.vocal	Top.50.Billboard
## 3	Lennon Lennon, with McCartney		8
## 19	Lennon, with McCartney Lennon, with McCartney		NA
## 22	Lennon and McCartney Lennon and McCartney		NA
## 36	McCartney McCartney, with Lennon		10
## 64	McCartney, with Lennon Lennon, with McCartney		21
## 67	McCartney Lennon, with McCartney		NA

	Publisher
## 3	NA
## 19	NA
## 22	NA
## 36	NA
## 64	NA
## 67	NA

```
# get the songs from year 1958, but only attributes Title and Album.debut
songs.1958 <- subset(the.beatles.songs, Year == 1958, c("Title", "Album.debut"))
head(songs.1958)

##              Title Album.debut
## 146 In Spite of All the Danger Anthology 1
## 254      That'll Be the Day Anthology 1
```

Values of specific columns/rows can be changed by assigning new values to them.

```
# create a vector of logical values denoting whether there the attribute Album.debut has a value or not
empty.album.debut <- the.beatles.songs$Album.debut == ""

# songs at indexes of all TRUE value will have their Album.debut attribute set to 'empty'
the.beatles.songs$Album.debut[empty.album.debut] <- "empty"

# set the value back to empty string
the.beatles.songs$Album.debut[empty.album.debut] <- ""
```

Saving dataset

```
# save dataset to a CSV file, but without the row names (row numbers) column
write.csv(the.beatles.songs, "p2.csv", row.names = F)

# save R object for the next session into file "p2.RData"
saveRDS(the.beatles.songs, "p2.RData")

# restore R object from the file "p2.RData" in the next session
p2 <- readRDS("p2.RData")
```

Task 2

Create a new column in the *the.beatles.songs* data frame called *Billboard.hit* having TRUE for all songs that were in the Top 50 Billboard (songs that have the *Top.50.Billboard* defined), and FALSE for all other songs (not having this value set).

Answer:

```
the.beatles.songs$Billboard.hit <- FALSE
the.beatles.songs$Billboard.hit[!is.na(the.beatles.songs$Top.50.Billboard)] <- TRUE
head(the.beatles.songs)

##              Title Year
## 1      12-Bar Original 1965
## 2      A Day in the Life 1967
## 3      A Hard Day's Night 1964
## 4      12-Bar Original 1965
```

```

## 5 A Shot of Rhythm and Blues 1963
## 6          A Taste of Honey 1963
##          Album.debut Duration Other.releases
## 1          Anthology 2          174          NA
## 2      Sgt. Pepper's Lonely Hearts Club Band          335          12
## 3          UK: A Hard Day's Night US: 1962-1966          152          35
## 4          Anthology 2          174          NA
## 5          Live at the BBC          104          NA
## 6 UK: Please Please Me US: The Early Beatles          163          29
##          Genre
## 1          Blues
## 2 Psychedelic Rock, Art Rock, Pop/Rock
## 3          Rock, Electronic, Pop/Rock
## 4          Blues
## 5          R&B, Pop/Rock
## 6          Pop/Rock, Jazz, Stage&Screen
##          Songwriter          Lead.vocal
## 1 Lennon, McCartney, Harrison and Starkey
## 2          Lennon and McCartney      Lennon and McCartney
## 3          Lennon      Lennon, with McCartney
## 4 Lennon, McCartney, Harrison and Starkey
## 5          Thompson          Lennon
## 6          Scott, Marlow          McCartney
## Top.50.Billboard Publisher Billboard.hit
## 1          NA          NA          FALSE
## 2          NA          NA          FALSE
## 3          8          NA          TRUE
## 4          NA          NA          FALSE
## 5          NA          NA          FALSE
## 6          NA          NA          FALSE

```