Intro to R - Part II

Data sampling

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

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

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

## [1]  1  7 10  9  8
```

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

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

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

If the previous code is ran multiple times, it will always produce different result. If we want to have the same sample each time, we need to specify the seed number (used by the random number generator).

```
# set seed and create two sample of size 20 from the vector, where duplicates
are allowed
set.seed(10)
sample(x, size = 20, replace = TRUE)

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

set.seed(10)
sample(x, size = 20, replace = TRUE)

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

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.

```r
# 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
```

```r
# get the first row
a[1, ]
```

```
## [1] 8 7 6 5
```

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

```
## [1] 7
```

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

```
## [1] 2
```

```r
# get number of columns
ncol(a)
```

```
## [1] 4
```

All matrix operations can be applied.

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

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

```r
matrix2 <- matrix(c(5, 2, 0, 9), nrow = 2)
matrix2
```

```
##      [,1] [,2]
## [1,]    5    0
## [2,]    2    9
```

```r
# add matrix2 to matrix1
matrix1 + matrix2
```

```
##      [,1] [,2]
## [1,]    8   -1
## [2,]   11   13
```

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

```r
# transpose a matrix
t(matrix1)
```

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

Lists

Lists are objects which contain elements of different types, such as numbers, strings, vectors, and even functions and other lists. A list is created by using the function *list()*. A specific element can be accessed by its index or its name. *length()* returns the number of elements in a list.

```r
# create a new list with attributes: passport, age, diplomatic
traveler1 <- list(passport = "P123123", age = 34, diplomatic = TRUE)
traveler1
```

```
## $passport
## [1] "P123123"
##
## $age
## [1] 34
##
## $diplomatic
## [1] TRUE
```

```r
# get the 2nd element
traveler1[2]
```

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

```r
# get the value of the 2nd element
traveler1[[2]]
```

```
## [1] 34
```

```r
# get the value of the age element
traveler1$age
```

```
## [1] 34
```

```r
# get the list length
length(traveler1)
```

```
## [1] 3
```

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

## $passport
## [1] "P123123"
##
## $age
## [1] 34
##
## $country
## [1] "AUS"
##
## $diplomatic
## [1] TRUE
```

An element is deleted by assigning NULL to it.

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

## [1] 3

traveler1

## $passport
## [1] "P123123"
##
## $age
## [1] 34
##
## $diplomatic
## [1] TRUE
```

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.

```
# concatinate two lists
traveler2 <- list(passport = "P456456", age = 14, diplomatic = FALSE)
travelers <- c(traveler1, traveler2)
travelers

## $passport
## [1] "P123123"
##
## $age
## [1] 34
```

```
##
## $diplomatic
## [1] TRUE
##
## $passport
## [1] "P456456"
##
## $age
## [1] 14
##
## $diplomatic
## [1] FALSE
```

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] "passport"   "age"        "diplomatic" "passport"   "age"
## [6] "diplomatic"

# get elements with 'age' in their name
travelers[grepl('age', names(travelers))]

## $age
## [1] 34
##
## $age
## [1] 14
```

Note: *grepl()* returns a logical vector indicating match or not match with the given pattern (1st argument) for each element of the vector or list that is passed as the 2nd argument.

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"
```

Task 1

Create a 2 x 3 matrix with the following elements: 3, 9, -1, 4, 2, 6 (by row). Print only the positive values from the first row.

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
```

if-else

```
# use ifelse function to create a new attribute called 'request' with the
value 'assistance required' if a traveler is younger than 10 years, and the
value 'no special requests' otherwise
traveler1$request <- ifelse(test = traveler1$age < 10,
                            yes = "assistance required",
                            no = "no special requests")
traveler1
```

```
## $passport
## [1] "P123123"
##
## $age
## [1] 34
##
## $diplomatic
## [1] TRUE
##
## $request
## [1] "no special requests"
```

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
```

```r
my_abs <- function(x) {
  if (x > 0) {
    return(x)
  }
  return(-x)
}

my_abs(5)
```

```
## [1] 5
```

```r
my_abs(-5)
```

```
## [1] 5
```

Applying a function over rows and columns of a data frame

We can apply a custom function to a vector, list, matrix or data frame.

The *sapply()* function accepts a vector or a list as the first argument. The 2nd argument is a function to be applied to each element of the vector / list given as the 1st argument. The result is a vector of the computed values.

```r
# load the data "data/beatles_v2.csv"
beatles <- read.csv("data/beatles_v2.csv")

# get the number of characters in the song title "Yellow Submarine"
nchar("Yellow Submarine")
```

```
## [1] 16
```

```r
# get the number of characters of the first 10 songs
sapply(beatles$Title[1:10], nchar)
```

```
##            12-Bar Original          A Day in the Life
##                         15                         17
##         A Hard Day's Night A Shot of Rhythm and Blues
##                         18                         26
##           A Taste of Honey        Across the Universe
##                         16                         19
##              Act Naturally           Ain't She Sweet
##                         13                         15
##         All I've Got to Do             All My Loving
##                         18                         13
```

The *apply()* function accepts a data frame or a matrix as the first argument. The second argument is called MARGIN and it defines how the function (3rd argument) 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.

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

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

```r
# calculate the 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(beatles[beatles$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.

```r
# create a contingency table of column Year values
year.counts <- table(beatles$Year)
year.counts
```

```
##
## 1958 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1977 1980
##    2    4    3   20   66   41   37   19   27   45   43    1    1    1
```

```r
# get the 4th element from the table
year.counts[4]
```

```
## 1962
##   20
```

```r
# store the 4th element from the table in a variable
x <- year.counts[4]
x
```

```
## 1962
##   20
```

```r
# convert the variable to numeric
y <- as.numeric(x)
y
```

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

Sort the table by the count value.

```
# sort the table in the descending order
sort(year.counts, decreasing = T)

##
## 1963 1968 1969 1964 1965 1967 1962 1966 1960 1961 1958 1970 1977 1980
##   66   45   43   41   37   27   20   19    4    3    2    1    1    1
```

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

```
# get the proportions table for the values of the Year column
year.counts.prop <- prop.table(year.counts)
year.counts.prop

##
##        1958        1960        1961        1962        1963        1964
## 0.006451613 0.012903226 0.009677419 0.064516129 0.212903226 0.132258065
##        1965        1966        1967        1968        1969        1970
## 0.119354839 0.061290323 0.087096774 0.145161290 0.138709677 0.003225806
##        1977        1980
## 0.003225806 0.003225806

# sort the proportions table in the descending order
sort(year.counts.prop, decreasing = T)

##
##        1963        1968        1969        1964        1965        1967
## 0.212903226 0.145161290 0.138709677 0.132258065 0.119354839 0.087096774
##        1962        1966        1960        1961        1958        1970
## 0.064516129 0.061290323 0.012903226 0.009677419 0.006451613 0.003225806
##        1977        1980
## 0.003225806 0.003225806

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

##
## 1958 1960 1961 1962 1963 1964 1965 1966 1967 1968 1969 1970 1977 1980
## 0.01 0.01 0.01 0.06 0.21 0.13 0.12 0.06 0.09 0.15 0.14 0.00 0.00 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, beatles)

##                 Year
## Top.50.Billboard 1961 1962 1963 1964 1965 1966 1967 1968 1969 1977 1980
##                1    0    0    0    0    0    0    0    1    0    0    0
##                2    0    0    1    0    0    0    0    0    0    0    0
##                3    0    0    1    0    0    0    0    0    0    0    0
##                4    0    0    0    0    0    0    0    0    1    0    0
##                5    0    0    0    0    0    0    0    0    1    0    0
```

```
##                6     0     0     0     0     0     0     0     0     1     0     0
##                7     0     0     0     0     0     0     1     0     0     0     0
##                8     0     0     0     1     0     0     0     0     0     0     0
##                9     0     0     0     0     1     0     0     0     0     0     0
##                10    0     0     0     1     0     0     0     0     0     0     0
##                11    0     0     0     1     0     0     0     0     0     0     0
##                12    0     0     0     0     1     0     0     0     0     0     0
##                13    0     0     1     0     0     0     0     0     0     0     0
##                14    0     0     0     0     1     0     0     0     0     0     0
##                15    0     0     0     0     0     0     1     0     0     0     0
##                16    0     1     0     0     0     0     0     0     0     0     0
##                17    0     0     0     0     1     0     0     0     0     0     0
##                18    0     1     0     0     0     0     0     0     0     0     0
##                19    0     0     0     0     0     1     0     0     0     0     0
##                20    0     0     0     0     0     0     0     0     1     0     0
##                21    0     0     0     1     0     0     0     0     0     0     0
##                22    0     0     0     0     0     0     0     1     0     0     0
##                23    0     0     0     0     0     1     0     0     0     0     0
##                24    0     0     0     0     0     1     0     0     0     0     0
##                25    0     0     0     0     0     1     0     0     0     0     0
##                26    0     0     1     0     0     0     0     0     0     0     0
##                27    0     0     0     0     1     0     0     0     0     0     0
##                28    0     0     0     0     0     0     0     0     1     0     0
##                29    0     0     0     0     1     0     0     0     0     0     0
##                30    0     0     0     0     0     0     0     0     1     0     0
##                31    0     0     0     1     0     0     0     0     0     0     0
##                32    0     0     0     0     0     0     0     1     0     0     0
##                33    0     0     0     0     0     1     0     0     0     0     0
##                34    0     1     0     0     0     0     0     0     0     0     0
##                36    0     0     1     0     0     0     0     0     0     0     0
##                37    0     0     0     1     0     0     0     0     0     0     0
##                38    0     0     0     0     0     1     0     0     0     0     0
##                39    0     0     0     0     0     0     0     0     0     1     0
##                40    0     0     0     1     0     0     0     0     0     0     0
##                41    1     0     0     0     0     0     0     0     0     0     0
##                42    0     0     0     0     0     1     0     0     0     0     0
##                43    0     0     0     1     0     0     0     0     0     0     0
##                44    0     0     0     1     0     0     0     0     0     0     0
##                45    1     0     0     0     0     0     0     0     0     0     0
##                46    0     0     1     0     0     0     0     0     0     0     0
##                47    0     0     0     0     0     0     0     0     0     0     1
##                48    0     0     0     0     0     0     1     0     0     0     0
##                49    0     0     0     1     0     0     0     0     0     0     0
##                50    0     0     0     0     1     0     0     0     0     0     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.

```
# create a new column On.album and set FALSE for all songs
beatles$On.album <- FALSE

head(beatles)
```

```
##                        Title Year
Album.debut
## 1            12-Bar Original 1965                                  Anthology
2
## 2          A Day in the Life 1967      Sgt. Pepper's Lonely Hearts Club
Band
## 3         A Hard Day's Night 1964       UK: A Hard Day's Night US: 1962-
1966
## 4 A Shot of Rhythm and Blues 1963                            Live at the
BBC
## 5           A Taste of Honey 1963 UK: Please Please Me US: The Early
Beatles
## 6        Across the Universe 1968                                  Let It
Be
##   Duration Other.releases                                Genre
## 1      174             NA                                Blues
## 2      335             12 Psychedelic Rock, Art Rock, Pop/Rock
## 3      152             35           Rock, Electronic, Pop/Rock
## 4      104             NA                       R&B, Pop/Rock
## 5      163             29         Pop/Rock, Jazz, Stage&Screen
## 6      230             19           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 new data frame with two columns (with sample data)
additional.columns <- data.frame(
  Platinum = sample(c(TRUE, FALSE), 310, replace = TRUE),
  Score = sample(5:10, 310, replace = TRUE)
)

# combine two data frames
beatles <- cbind(beatles, additional.columns)
head(beatles)
```

```
##                         Title Year
Album.debut
## 1            12-Bar Original 1965                                  Anthology
2
## 2          A Day in the Life 1967      Sgt. Pepper's Lonely Hearts Club
Band
## 3         A Hard Day's Night 1964       UK: A Hard Day's Night US: 1962-
1966
## 4 A Shot of Rhythm and Blues 1963                               Live at the
BBC
## 5           A Taste of Honey 1963 UK: Please Please Me US: The Early
Beatles
## 6        Across the Universe 1968                                    Let It
Be
##   Duration Other.releases                                  Genre
## 1      174             NA                                  Blues
## 2      335             12 Psychedelic Rock, Art Rock, Pop/Rock
## 3      152             35           Rock, Electronic, Pop/Rock
## 4      104             NA                        R&B, Pop/Rock
## 5      163             29         Pop/Rock, Jazz, Stage&Screen
## 6      230             19           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 Platinum Score
## 1               NA    FALSE     TRUE    10
## 2               NA    FALSE    FALSE     7
## 3                8    FALSE    FALSE     7
## 4               NA    FALSE     TRUE     8
## 5               NA    FALSE    FALSE     5
## 6               NA    FALSE     TRUE     6
```

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

```
# get the first song
new.song <- beatles[1, ]

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

```
##                                 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
Other.releases
## 306                                           Help!      140
6
## 307                                           Help!      131
12
## 308 UK: With the Beatles US: The Beatles Second Album      182
2
## 309                                  Live at the BBC      116
NA
## 310                              Magical Mystery Tour      149
13
## 311                                       Anthology 2      174
NA
##                                                  Genre
## 306                                   Rock, Pop/Rock
## 307                                     FolkPop/Rock
## 308                                   Soul, Pop/Rock
## 309                                         Pop/Rock
## 310 Music Hall, Vaudeville Rock, Psychedelic Pop, Pop/Rock
## 311                                             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 Platinum Score
## 306               NA    FALSE    FALSE     7
## 307               NA    FALSE     TRUE     8
## 308               NA    FALSE    FALSE     7
## 309               NA    FALSE     TRUE     9
```

```
## 310               NA    FALSE    FALSE     9
## 311               NA    FALSE     TRUE    10

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

##                          Title Year
Album.debut
## 1              12-Bar Original 1965
Anthology 2
## 2            A Day in the Life 1967      Sgt. Pepper's Lonely Hearts Club
Band
## 3          A Hard Day's Night 1964       UK: A Hard Day's Night US: 1962-
1966
## 4              12-Bar Original 1965
Anthology 2
## 410 A Shot of Rhythm and Blues 1963                            Live at the
BBC
## 5             A Taste of Honey 1963 UK: Please Please Me US: The Early
Beatles
##     Duration Other.releases                                  Genre
## 1        174             NA                                  Blues
## 2        335             12 Psychedelic Rock, Art Rock, Pop/Rock
## 3        152             35           Rock, Electronic, Pop/Rock
## 4        174             NA                                  Blues
## 410      104             NA                        R&B, Pop/Rock
## 5        163             29         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 Platinum Score
## 1                  NA    FALSE     TRUE    10
## 2                  NA    FALSE    FALSE     7
## 3                   8    FALSE    FALSE     7
## 4                  NA    FALSE     TRUE    10
## 410                NA    FALSE     TRUE     8
## 5                  NA    FALSE    FALSE     5
```

Removing columns and rows

A column is removed by assigning a NULL to it.

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

```
##  [1] "Title"            "Year"             "Album.debut"      "Duration"
##  [5] "Other.releases"   "Genre"            "Songwriter"       "Lead.vocal"
##  [9] "Top.50.Billboard" "Platinum"         "Score"
```

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 Platinum (at index 10) and Score (at index 11)
beatles <- beatles[,-c(10, 11)]
names(beatles)
```

```
## [1] "Title"            "Year"             "Album.debut"      "Duration"
## [5] "Other.releases"   "Genre"            "Songwriter"       "Lead.vocal"
## [9] "Top.50.Billboard"
```

Using the same method, rows can be removed.

```
# create a subset of the data frame without songs in rows 2, 4 and 6
beatles1 <- beatles[-c(2, 4, 6), ]
head(beatles1)
```

```
##                          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
## 1                 NA
## 3                  8
## 410               NA
## 6                 NA
```

```
## 7                 50
## 8                 41

# create a subset of the data frame without songs in rows from 1 to 8
beatles2 <- beatles[-(1:8), ]
head(beatles2)

##                   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
## 8                  Lennon               41
## 9                  Lennon               NA
## 10              McCartney               NA
## 11               Harrison               NA
## 12 McCartney, with Lennon               NA
## 13                 Lennon               15
```

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(beatles)

## [1] "Title"            "Year"             "Album.debut"      "Duration"
## [5] "Other.releases"   "Genre"            "Songwriter"       "Lead.vocal"
## [9] "Top.50.Billboard"

# change name of the column that starts with 'Genre' to 'Song.genre'
genreIndex <- which(startsWith(colnames(beatles), "Genre"))
colnames(beatles)[genreIndex] <- "Song.genre"
colnames(beatles)

## [1] "Title"            "Year"             "Album.debut"      "Duration"
## [5] "Other.releases"   "Song.genre"       "Songwriter"       "Lead.vocal"
## [9] "Top.50.Billboard"
```

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

```
## [1] "Title"            "Year"             "Album.debut"      "Duration"
## [5] "Other.releases"   "Genre"            "Songwriter"       "Lead.vocal"
## [9] "Top.50.Billboard"
```

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(beatles) <- paste("song", 1:nrow(beatles))
head(beatles)
```

```
##                             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
## song 1               NA
## song 2               NA
## song 3                8
## song 4               NA
## song 5               NA
## song 6               NA
```

```r
# change row names to a string containing order number
rownames(beatles) <- c(1:nrow(beatles))
head(beatles)
```

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

Retrieving and changing values

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

```r
# get songs in rows from 1 to 5, but only attributes Title and Album.debut
first.songs <- beatles[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((beatles$Year == "1964") & (!grepl('McCartney',
beatles$Lead.vocal)))
selected.songs <- beatles[indexes, ]
head(selected.songs)

##                                Title Year
## 69  Everybody's Trying to Be My Baby 1964
## 103                      Honey Don't 1964
## 107                  I Call Your Name 1964
## 108  I Don't Want to Spoil the Party 1964
## 109                      I Feel Fine 1964
## 110   I Forgot to Remember to Forget 1964
##                                                 Album.debut Duration
## 69                    UK: Beatles for Sale US: Beatles '65      143
## 103                   UK: Beatles for Sale US: Beatles '65      173
## 107 UK: Past Masters Volume 1 US: The Beatles Second Album      129
## 108                    UK: Beatles for Sale US: Beatles VI      153
## 109     UK: A Collection of Beatles Oldies US: Beatles '65      145
## 110                                       Live at the BBC      148
##     Other.releases                                 Genre
## 69              21 Rock and Roll, Rockabilly, Pop/Rock
## 103             14                Rockabilly, Pop/Rock
## 107             16                      Rock, Pop/Rock
## 108             11              Country Rock, Pop/Rock
## 109             35                      Rock, Pop/Rock
## 110             NA                   Country, Pop/Rock
##                           Songwriter Lead.vocal Top.50.Billboard
## 69                           Perkins   Harrison               NA
## 103                          Perkins    Starkey               NA
## 107                           Lennon     Lennon               NA
## 108                           Lennon     Lennon               49
## 109                           Lennon     Lennon               11
## 110 Stan Kesler and Charlie Feathers   Harrison               NA

# get the songs from year 1958, but only attributes Title and Album.debut
songs.1958 <- subset(beatles, 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 the attribute
Album.debut has a value or not
empty.album.debut <- beatles$Album.debut == ""

# compute how many songs lack the data about the debut album
sum(empty.album.debut)

## [1] 22

# for songs without debut album data, set the value of the Album.debut
attribute to 'empty'
beatles$Album.debut[empty.album.debut] <- "empty"

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

Saving dataset

```
# save dataset to a CSV file, but without the row names (row numbers) column
write.csv(beatles, "data/beatles_v3.csv", row.names = F)

# save R object for the next session into file "data/beatles_v3.RData"
saveRDS(beatles, "data/beatles_v3.RData")

# restore R object from the file "data/beatles_v3.RData" in the next session
b3 <- readRDS("data/beatles_v3.RData")
```

Task 2

Create a new column in the *beatles* 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:

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

##                         Title Year
Album.debut
## 1            12-Bar Original 1965                                Anthology
2
## 2          A Day in the Life 1967      Sgt. Pepper's Lonely Hearts Club
Band
## 3         A Hard Day's Night 1964       UK: A Hard Day's Night US: 1962-
1966
## 4            12-Bar Original 1965                                Anthology
2
## 5 A Shot of Rhythm and Blues 1963                             Live at the
BBC
```

```
## 6           A Taste of Honey 1963 UK: Please Please Me US: The Early
Beatles
##   Duration Other.releases                                  Genre
## 1      174             NA                                  Blues
## 2      335             12 Psychedelic Rock, Art Rock, Pop/Rock
## 3      152             35           Rock, Electronic, Pop/Rock
## 4      174             NA                                  Blues
## 5      104             NA                        R&B, Pop/Rock
## 6      163             29         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 Billboard.hit
## 1               NA         FALSE
## 2               NA         FALSE
## 3                8          TRUE
## 4               NA         FALSE
## 5               NA         FALSE
## 6               NA         FALSE
```