Fachhochschule Südwestfalen
University of Applied Sciences

Programmierung R - Exercise

Data manipulation

April 12, SS 2022 || Hannah Behrens

Wir geben Impulse

Task 1

Your turn

In your R environment the data set `datasets::airquality` (R Core Team 2021) is already available.

1.1 Make yourself familiar with this data set by examining

- its data type
- its structure (and dimensions) and
- its variables.

1.2 Where do and how many NAs occur in the data set? Remove those observations (full rows) and save this data set as `airquality2`. How many rows have been deleted?

Task 1.1 - answer

1.1 Make yourself familiar with the data set `datasets::airquality` (R Core Team 2021)...

```
typeof(airquality) # its data type
```

```
## [1] "list"
```

```
str(airquality) # its structure and its variables
```

```
## 'data.frame':    153 obs. of  6 variables:
##  $ Ozone  : int  41 36 12 18 NA 28 23 19 8 NA ...
##  $ Solar.R: int  190 118 149 313 NA NA 299 99 19 194 ...
##  $ Wind   : num  7.4 8 12.6 11.5 14.3 14.9 8.6 13.8 20.1 8.6 ...
##  $ Temp   : int  67 72 74 62 56 66 65 59 61 69 ...
##  $ Month  : int  5 5 5 5 5 5 5 5 5 5 ...
##  $ Day    : int  1 2 3 4 5 6 7 8 9 10 ...
```

```
?airquality # description of the data set
```

→ This data set is a list, more precisely a data frame.

→ 153 observations (153 rows) and 6 variables (6 columns)

Task 1.2 - answer

→ 37 NAs are present in the column `Ozone` and 7 NAs in `Solar.R`.

```
summary(airquality) # to see where and how many NAs occur
```

```
##      Ozone           Solar.R         Wind             Temp           Month
##  Min.   :  1.0   Min.   :  7   Min.   : 1.70   Min.   :56.0   Min.   :5.00
##  1st Qu.: 18.0   1st Qu.:116   1st Qu.: 7.40   1st Qu.:72.0   1st Qu.:6.00
##  Median : 31.5   Median :205   Median : 9.70   Median :79.0   Median :7.00
##  Mean   : 42.1   Mean   :186   Mean   : 9.96   Mean   :77.9   Mean   :6.99
##  3rd Qu.: 63.2   3rd Qu.:259   3rd Qu.:11.50   3rd Qu.:85.0   3rd Qu.:8.00
##  Max.   :168.0   Max.   :334   Max.   :20.70   Max.   :97.0   Max.   :9.00
##  NA's   :37      NA's   :7
##       Day
##  Min.   : 1.0
##  1st Qu.: 8.0
##  Median :16.0
##  Mean   :15.8
##  3rd Qu.:23.0
##  Max.   :31.0
##
```

Which return value would you expect if `summary()` were applied to a categorical variable?

Task 1.2 - answer

```r
# remove those observations (full rows), where NAs occur:
airquality2 <- na.omit(airquality)
nrow(airquality) - nrow(airquality2) # 42 rows have been removed
```

```
## [1] 42
```

```r
# consequently, 153 - 42 = 111 rows left
```

Task 2

Your turn

As we have already seen, the variable `Month` is numeric.

2.1) Add a further variable named `Month_name` that saves the name of the month (as character) to the *NAs-free* `airquality2` data set.

2.2) Apply `summary()` to `Month_name` and check your previous assumption. Assign the result to the variable `summary_air`.

Task 2.1 - answer

As we have already seen, the variable `Month` is numeric.

Add a further variable named `Month_name` that saves the name of the month (as character) to the *NAs-free* `airquality2` data set.

```
levels_month <- unique(airquality2$Month) # get unique values of Month
# add the names of the months as a factor to the data set:
airquality2$Month_name <- factor(x = airquality2$Month,
                                 levels = levels_month, # alternatively: c(5, 6, 7, 8, 9)
                                 # labels: how to name the levels,
                                 # make use of the predefined  constant named month.name
                                   labels = month.name[levels_month])

head(airquality2) # check the modification
```

```
##   Ozone Solar.R Wind Temp Month Day Month_name
## 1    41     190  7.4   67     5   1        May
## 2    36     118  8.0   72     5   2        May
## 3    12     149 12.6   74     5   3        May
## 4    18     313 11.5   62     5   4        May
## 7    23     299  8.6   65     5   7        May
## 8    19      99 13.8   59     5   8        May
```

Task 2.2 - answer

```r
summary_air <- summary(airquality2$Month_name)
summary_air # a vector results
```

```
##       May      June      July    August September
##        24         9        26        23        29
```

```r
typeof(summary_air)
```

```
## [1] "integer"
```

Task 3

Your turn

Apply the following tasks to the modified `airquality2` data set.

The temperature of the `airquality` data set is given in degrees F (Fahrenheit).
3.1 Convert these values into degrees Celsius [C] by using the following formula given by National Institute of Standards and Technology (NIST) (2021): $C = \frac{(F-32)}{1.8}$.

3.2 You are unhappy with the variables' names of the `airquality2` data set. Get the column names and change `Temp` to `Temperature`.

Tasks 3.1 and 3.2 - answers

3.1

```r
# convert temperature from degrees Fahrenheit to degrees Celsius
# first option:
airquality2 <- transform(airquality2,
          Temp = (Temp - 32)/1.8)
# second option:
# airquality2$Temp <- (airquality2$Temp - 32)/1.8
```

3.2

```r
colnames_airqu <- colnames(airquality2) # get column names of airquality2
colnames_airqu
```

```
## [1] "Ozone"      "Solar.R"    "Wind"       "Temp"       "Month"
## [6] "Day"        "Month_name"
```

```r
colnames_airqu[4] <- "Temperature" # change "Temp" to "Temperature"
colnames_airqu # check the modified vector
```

```
## [1] "Ozone"       "Solar.R"     "Wind"        "Temperature" "Month"
## [6] "Day"         "Month_name"
```

Task 3.2 - answer

Check your modifications!

```
colnames(airquality2) <- colnames_airqu # assign modified column names to data set
colnames(airquality2) # check the modification
```

```
## [1] "Ozone"       "Solar.R"     "Wind"        "Temperature" "Month"
## [6] "Day"         "Month_name"
```

Task 4

Your turn

4.1 Imagine the following observation `new_obs` has misleadingly not been integrated in the `airquality` data set. Add this observation to the modified data set, assign it to the variable `airquality3` and check the dimensions of the data set afterwards.

```
new_obs <- data.frame(Ozone = 20, Solar.R = 300, Wind = 8.7,
                      Temperature = 20, Month = 6, Day = 6, Month_name = "June")
```

4.2 Add a date column to the modified `airquality3` data set:

- First paste the entries of the `Month` and `Day` column with the year of the observations by separating the single values by "/."
- Then convert these strings to POSIX objects.

4.3 Save the modified `airquality3` data set as CSV-file. Afterwards, control your created CSV-file by reading it in your console.

Tasks 4.1 and 4.2 - answers

4.1

```
airquality3 <- rbind(airquality2, new_obs) # add observation new_obs to the
# modified airquality data set (airquality2)
dim(airquality3) # check the dimensions (+ 1 row)
```

```
## [1] 112   7
```

4.2

```
# paste Month, Day and year together in form of Month/Day/Year e.g. 5/2/1973
airquality3$Date <- paste(airquality3$Month, airquality3$Day, "1973", sep = "/")
# convert these strings to POSIX objects based on strptime()
airquality3$Date <- strptime(airquality3$Date, format = "%m/%d/%Y")

head(airquality3) # check the modifications
```

```
##   Ozone Solar.R Wind Temperature Month Day Month_name       Date
## 1    41     190  7.4       19.44     5   1        May 1973-05-01
## 2    36     118  8.0       22.22     5   2        May 1973-05-02
## 3    12     149 12.6       23.33     5   3        May 1973-05-03
## 4    18     313 11.5       16.67     5   4        May 1973-05-04
## 7    23     299  8.6       18.33     5   7        May 1973-05-07
## 8    19      99 13.8       15.00     5   8        May 1973-05-08
```

Task 4.3 - answer

We will use this modified data set in a future exercise.

```r
# save the modified data set as a csv-file:
write.table(airquality3, file = "Airquality3.csv", sep = ",")
getwd() # get working directory
```

```r
airquality3_csv <- read.table(file = "Airquality3.csv", sep = ",") # read csv-file
head(airquality3_csv)
```

```
##   Ozone Solar.R Wind Temperature Month Day Month_name       Date
## 1    41     190  7.4       19.44     5   1        May 1973-05-01
## 2    36     118  8.0       22.22     5   2        May 1973-05-02
## 3    12     149 12.6       23.33     5   3        May 1973-05-03
## 4    18     313 11.5       16.67     5   4        May 1973-05-04
## 7    23     299  8.6       18.33     5   7        May 1973-05-07
## 8    19      99 13.8       15.00     5   8        May 1973-05-08
```

DataCamp course to importing data in R will follow!

Task 5

Your turn

Answer the following questions by using `airquality3`.

5.1 How many observations with temperatures above 30°C exist in the data set?

5.2 Sort the modified `airquality3` data set by temperature in descending order and only return the ordered values of the variable `Wind`.

5.3 Split `airquality3` into sub populations by `Month_name` and return the minimum of each variable. Does it work properly or are some changes necessary?

5.4 Return only observations of those months that include a “u” in their names.

Tasks 5.1 and 5.2 - answers

5.1

```
sum(airquality3$Temperature > 30) # Temperatures above 30°C have been measured 20 times
```

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

5.2

```
# sort by temperature in descending order and return the values of Wind:
airquality3$Wind[sort(airquality3$Temperature, decreasing = TRUE)]
```

```
##   [1]  9.2  9.2  4.1  4.1 10.3 10.3 10.3 10.3 10.3 11.5 11.5 11.5 11.5  9.2  9.2
##  [16]  9.2  9.2 20.7 20.7 20.7 20.7 20.7 20.7 20.7 20.7 14.9 14.9 14.9  8.0  8.0
##  [31]  8.0  8.0  8.0  8.0 11.5 11.5 11.5 11.5 11.5 11.5 11.5 11.5 11.5 11.5 11.5
##  [46] 11.5 11.5 11.5 11.5 11.5 11.5 13.8 13.8 13.8 13.8 13.8 13.8  9.7  9.7  9.7
##  [61]  9.7  9.7  9.7  9.7  9.7  7.4  7.4  7.4  7.4  7.4  7.4  5.7  5.7  5.7  5.7
##  [76] 14.9 14.9 14.9 14.9 14.9 14.9 14.9 12.0 12.0 12.0 12.0 12.0 12.0 12.0 12.0
##  [91] 12.0 12.0 12.0  9.7  9.7  9.7 16.6 16.6 16.6 16.6  9.7  9.7  9.7  9.7  9.7
## [106]  9.7  9.7  9.7 11.5 11.5 18.4 12.0
```

Task 5.3 - answers

5.3

```
aggregate(x = airquality3, by = list(Month_name = airquality3$Month_name), FUN = min)
# error results
```

→ It does not work since `Month_name` is an unordered factor, so order the levels of `Month_name` and apply `aggregate()` again.

```
airquality3$Month_name <- factor(x = airquality3$Month_name, ordered = TRUE)
# remove seventh column, otherwise Month_name would be listed twice
aggregate(x = airquality3[,-7], by = list(Month_name = airquality3$Month_name),
          FUN = min)
```

```
##   Month_name Ozone Solar.R Wind Temperature Month Day       Date
## 1        May     1       8  5.7       13.89     5   1 1973-05-01
## 2       June    12      37  8.0       18.33     6   6 1973-06-06
## 3       July     7       7  4.1       22.78     7   1 1973-07-01
## 4     August     9      24  2.3       22.22     8   1 1973-08-01
## 5  September     7      14  2.8       17.22     9   1 1973-09-01
```

Task 5.4 - answers

5.4

```
# search for "u" in Month_name and only return the corresponding observations
airquality4 <- airquality3[grep(pattern = "u", x = airquality3$Month_name),]

head(airquality4)
```

```
##    Ozone Solar.R Wind Temperature Month Day Month_name       Date
## 38    29     127  9.7       27.78     6   7       June 1973-06-07
## 40    71     291 13.8       32.22     6   9       June 1973-06-09
## 41    39     323 11.5       30.56     6  10       June 1973-06-10
## 44    23     148  8.0       27.78     6  13       June 1973-06-13
## 47    21     191 14.9       25.00     6  16       June 1973-06-16
## 48    37     284 20.7       22.22     6  17       June 1973-06-17
```

```
dim(airquality4)
```

```
## [1] 59  8
```

Task 6

Your turn

6.1 Create a data frame `month_seasons` with two columns: In the first column called `Month_name` the names of all (12) months are listed. The second column named `Season` consists of the seasons (Spring, Summer, Autumn, Winter) corresponding to the months.

6.2 Then, merge `month_seasons` and `airquality3` by `Month_name` and save this data frame as `airquality3`. Check the dimensions of the resulting data set.

Task 6.1 - answer

6.1

```r
month_seasons <- data.frame(Month_name = factor(x = month.name, levels = month.name,
                            ordered = TRUE), # order months
                            Seasons = factor(x = c("Winter", "Winter", "Spring",
                            "Spring", "Spring", "Summer", "Summer", "Summer", "Autumn",
                            "Autumn", "Autumn", "Winter"),
                            levels = c("Spring", "Summer", "Autumn", "Winter"),
                            ordered = TRUE)) # order seasons

head(month_seasons, n = 7)
```

```
##   Month_name Seasons
## 1    January  Winter
## 2   February  Winter
## 3      March  Spring
## 4      April  Spring
## 5        May  Spring
## 6       June  Summer
## 7       July  Summer
```

Task 6.2 - answer

6.2

```
airquality3 <- merge(x = airquality3, y = month_seasons, by = "Month_name")
dim(airquality3) # + 1 column since column Seasons has been added
```

```
## [1] 112   9
```

Task 7

Do not run the following code on your console!

```
1  x <- c(1:10, 23:19, NA)
2  X <- 3
3  x <- x[x < 23]
4  x <- x[x <= 22]
5  x + X
6  x[X] <- x[X] + X
7  x <- x[X] + X
8  x <- x %% 3
9  y <- as.logical(x)
```

Your turn

What is the result of

- `x` after executing lines 1–6?
- `x` after executing lines 1–8?
- `y`?

Task 7 - answer

What is the result of `x` after executing lines 1–6?

```
#>R [1]  1  2  6  4  5  6  7  8  9 10 22 21 20 19 NA
```

What is the result of `x` after executing lines 1–8?

```
#>R [1] 0
```

What is the result of `y`?

```
#>R [1] FALSE
```

Task 8

Do not run the following code on your console!

```
1  L <- list(money = c(250, 124, 360, 720, 340, 340),
2            hours = c(19, 12, 30, 48, 26, 25),
3            idx = c(1:6),
4            name = c("Paul", "Emma", "Mia", "John", "Kim"))
5  z <- 2
6  L[[4]][[6]] <- "Maxi"
7  L$hours <- L$hours * 1.3
8  L[[1]] <- L$money + 25
9  L[1] <- 1250
10 L[[4]][[2]] <- L[[4]][[6]]
11 L[[z]] <- 150
```

Your turn

What is the result of

- L after executing lines 1–8?
- L after executing lines 1–9?
- L after executing lines 1–11?

Task 8 - answers

What is the result of `L` after executing lines 1–8?

```
1  #>R
2  #>R $money
3  #>R [1] 275 149 385 745 365 365
4
5  #>R $hours
6  #>R [1] 24.7 15.6 39.0 62.4 33.8 32.5
7
8  #>R $idx
9  #>R [1] 1 2 3 4 5 6
10
11 #>R $name
12 #>R [1] "Paul" "Emma" "Mia"  "John" "Kim"  "Maxi"
13
14 # since "Maxi" has been added to the name list,
15 # since each element of the hours list has been multiplied by 1.3 and
16 # since to each element of the money list 25 has been added
```

Task 8 - answers

What is the result of `L` after executing lines 1–9?

```
#>R $money
#>R [1] 1250

#>R $hours
#>R [1] 24.7 15.6 39.0 62.4 33.8 32.5

#>R $idx
#>R [1] 1 2 3 4 5 6

#>R $name
#>R [1] "Paul" "Emma" "Mia"  "John" "Kim"  "Maxi"

# in addition to the previous steps, the value 1250 has been assigned as the first list
# (money) of L
```

Task 8 - answers

What is the result of `L` after executing lines 1–11?

```
1  #>R $money
2  #>R [1] 1250
3
4  #>R $hours
5  #>R [1] 150
6
7  #>R $idx
8  #>R [1] 1 2 3 4 5 6
9
10 #>R $name
11 #>R [1] "Paul" "Maxi" "Mia"  "John" "Kim"  "Maxi"
12
13 # in addition to the previous steps, the second element of the name list has been
14 # overwritten by the sixth name and
15 # the value 150 has been assigned as the second list (hours) of L
```

References

Buchwitz, B. 2021. *Computational Statistics*. https://bchwtz.github.io/bchwtz-cswr/.

National Institute of Standards and Technology (NIST). 2021. *SI Units – Temperature*. USA, Gaithersburg. https://www.nist.gov/pml/weights-and-measures/si-units-temperature.

R Core Team. 2021. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.