R-Coding Lab

Dr. Matteo Tanadini

R-bootcamp

Contents

1 Coding

1.1 Piping ("%>%")

```
library(magrittr)
## ("ceci n'est pas un pipe")
```

```
round(mean(iris$Sepal.Length), digits = 2)
```

```
[1] 5.8
```

```
## OR
## with "pipe" operator
iris$Sepal.Length %>% mean %>% round(digits = 2)
```

```
[1] 5.8
```

Same example, but with "exposition pipe" operator.

```
iris %$% ## note the two types of pipe here
  mean(Sepal.Length) %>%
  round(digits = 2)
```

```
[1] 5.8
```

A more complex example.

```
mean(scale(iris$Petal.Length[iris$Species == "setosa"],
     scale = TRUE, center = FALSE))
```

```
[1] 0.98
```

```
## OR
## with "pipe" and "exposition pipe" operator
iris %>%
  subset(Species == "setosa") %$%
  scale(Petal.Length, scale = TRUE, center = FALSE) %>%
  mean
```

```
[1] 0.98
```

1.2 Reshaping data sets

```
# d.sport <- read.table("DataSport.txt", header = TRUE,
#                       nrows = 7)
# ## OR
d.sport <-read.table(
  "http://stat.ethz.ch/Teaching/Datasets/WBL/sport.dat",
  header = TRUE, nrows = 7)
##
d.sport
```

```
           weit kugel hoch disc stab speer punkte
OBRIEN      7.6    16  207   49  500    67   8824
BUSEMANN    8.1    14  204   45  480    67   8706
DVORAK      7.6    16  198   46  470    70   8664
FRITZ       7.8    15  204   50  510    66   8644
HAMALAINEN  7.5    16  198   50  500    58   8613
NOOL        7.9    14  201   43  540    65   8543
ZMELIK      7.6    14  195   43  540    67   8422
```

```
str(d.sport)
```

```
'data.frame':   7 obs. of  7 variables:
 $ weit  : num  7.57 8.07 7.6 7.77 7.48 7.88 7.64
 $ kugel : num  15.7 13.6 15.8 15.3 16.3 ...
 $ hoch  : int  207 204 198 204 198 201 195
 $ disc  : num  48.8 45 46.3 49.8 49.6 ...
 $ stab  : int  500 480 470 510 500 540 540
 $ speer : num  66.9 66.9 70.2 65.7 57.7 ...
 $ punkte: int  8824 8706 8664 8644 8613 8543 8422
```

First take rownames as an actual variable ("there is no such a thing as 'metadata' ").

```
library(tibble) ## for rownames_to_column()
d.sport <- d.sport %>%
  rownames_to_column(var = "Athlete")
d.sport
```

```
     Athlete weit kugel hoch disc stab speer punkte
1     OBRIEN  7.6    16  207   49  500    67   8824
2   BUSEMANN  8.1    14  204   45  480    67   8706
3     DVORAK  7.6    16  198   46  470    70   8664
4      FRITZ  7.8    15  204   50  510    66   8644
5 HAMALAINEN  7.5    16  198   50  500    58   8613
6       NOOL  7.9    14  201   43  540    65   8543
7     ZMELIK  7.6    14  195   43  540    67   8422
```

In this data set each row represents an athlete. Each athlete has a record for each discipline. This data is said to be in wide-format.

Let's turn this data set into a long-format data set. Each row will then represent the performance of an athlete in a given discipline (e.g. weit for OBRIEN).

```
library(tidyr) ## for gather()
d.sport.long <- gather(d.sport,
```

```
                key = "discipline", ## new column with name of the discipline
                value = "result", ## new column with value (unquoted also works)
                -Athlete) ## variable(s) that is(are) not to put as results
##
head(d.sport.long)
```

```
     Athlete discipline result
1     OBRIEN       weit    7.6
2   BUSEMANN       weit    8.1
3     DVORAK       weit    7.6
4      FRITZ       weit    7.8
5 HAMALAINEN       weit    7.5
6       NOOL       weit    7.9
```

```
d.sport.long %>%
  subset(subset = Athlete == "OBRIEN")
```

```
   Athlete discipline result
1   OBRIEN       weit    7.6
8   OBRIEN      kugel   15.7
15  OBRIEN       hoch  207.0
22  OBRIEN       disc   48.8
29  OBRIEN       stab  500.0
36  OBRIEN      speer   66.9
43  OBRIEN     punkte 8824.0
```

```
## OR
# library(dplyr)
# d.long %>%
#   filter(Athlete == "OBRIEN")
```

Let's get back to a wide data set.

```
d.sport.wide.again <- spread(d.sport.long,
                             key = "discipline", ## name from long df
                             value = "result") ## name from long df
head(d.sport.wide.again)
```

```
     Athlete disc hoch kugel punkte speer stab weit
1   BUSEMANN   45  204    14   8706    67  480  8.1
2     DVORAK   46  198    16   8664    70  470  7.6
3      FRITZ   50  204    15   8644    66  510  7.8
4 HAMALAINEN   50  198    16   8613    58  500  7.5
5       NOOL   43  201    14   8543    65  540  7.9
6     OBRIEN   49  207    16   8824    67  500  7.6
```

Note: the data format needed for supervised problems is long-format (each row represents a single observation). Whereas, for unsupervised problems, wide-format is required (i.e. one represents several measurements). For example, a sample of wine where several aspects where quantified.

1.3 Joining data sets

```
set.seed(14)
d.age <- data.frame(age = runif(n = 7, min = 19, max = 34),
```

```
                    Athlete = d.sport$Athlete,
                    Gender = c("M","M","F","M","F","F","F"))
d.age
```

```
  age    Athlete Gender
1  23     OBRIEN      M
2  29   BUSEMANN      M
3  33     DVORAK      F
4  27      FRITZ      M
5  34 HAMALAINEN      F
6  27       NOOL      F
7  33     ZMELIK      F
```

Let's add this information to the long-format data set.

```
library(dplyr)
d.sport.long.age <- left_join(d.sport.long, d.age, by = "Athlete")
```

```
Warning: Column `Athlete` joining character vector and factor, coercing into
character vector
```

```
head(d.sport.long.age)
```

```
     Athlete discipline result age Gender
1     OBRIEN       weit    7.6  23      M
2   BUSEMANN       weit    8.1  29      M
3     DVORAK       weit    7.6  33      F
4      FRITZ       weit    7.8  27      M
5 HAMALAINEN       weit    7.5  34      F
6       NOOL       weit    7.9  27      F
```

```
class(d.sport.long$Athlete)
```

```
[1] "character"
```

```
class(d.age$Athlete)
```

```
[1] "factor"
```

If names of the shared variable differs between the two data sets.

```
set.seed(14)
d.age.2 <- data.frame(age = runif(n = 7, min = 19, max = 34),
                      person = d.sport$Athlete)
```

```
d.sport.long.age.2 <- left_join(d.sport.long, d.age.2,
                                by = c("Athlete" = "person"))
```

```
Warning: Column `Athlete`/`person` joining character vector and factor, coercing
into character vector
```

```
head(d.sport.long.age.2)
```

```
     Athlete discipline result age
1     OBRIEN       weit    7.6  23
2   BUSEMANN       weit    8.1  29
```

```
3     DVORAK       weit    7.6  33
4      FRITZ       weit    7.8  27
5 HAMALAINEN       weit    7.5  34
6       NOOL       weit    7.9  27
```

see *?join* for further function to join two data sets together.

1.4 Warning about *{dplyr}* and loops

Avoid using *{dplyr}* functions within loops (e.g. *for* loops).