

Reading and data wrangling R

Andrea Marcela Huerfano Barbosa

August 14, 2019

Description

In this file you can find some tips to:

- Reading data from different formats (txt,csv,excel...)
- Cleaning data
- Creation of new variables
- Merging datasets
- Dealing with NA

All of the task above are related with how to clean and tidy our data, that is an inevitable phase when you work with data. Some terms for these activities are: data cleaning, data wrangling and data manipulation.

1. Reading data

There are many ways to import datasets depending on the file characteristics as separator, decimals, head, etc. The easy way is using the button Import Dataset in the R-Studio enviroment, however you have to copy the code into your script because the lines just run in the console. To know some of the fuctions that appear throw the bottom you are going to find some examples.

- read.csv: comma separated values with period as decimal separator.
- read.csv2: semicolon separated values with comma as decimal separator.
- read.delim: tab-delimited files with period as decimal separator.
- read.delim2 tab-delimited files with comma as decimal separator.
- read.fwf data with a predetermined number of bytes per column.

```
pigeon <- read.delim("C:/Users/Andrea/Desktop/pigeon-racing.txt")
str(pigeon)
```

```
## 'data.frame': 400 obs. of 11 variables:
## $ Pos : int 1 2 3 4 5 6 7 8 9 10 ...
## $ Breeder : Factor w/ 90 levels "4-Birds","7-11 Syndicate",...: 83 49 47 4 40 24 40 64 9 83 ...
## $ Pigeon : Factor w/ 400 levels "0001-AU15-RTEX",...: 272 99 101 283 381 40 383 184 191 271 ...
## $ Name : Factor w/ 21 levels "", "\"the Duck\"",...: 1 1 18 1 1 1 1 1 1 1 ...
## $ Color : Factor w/ 29 levels "BB","BBPD","BBPI",...: 9 26 1 4 6 6 5 6 1 6 ...
## $ Sex : Factor w/ 2 levels "C","H": 2 2 2 2 2 1 2 2 2 ...
## $ Ent : int 1 1 1 1 1 1 2 1 1 2 ...
## $ Arrival : Factor w/ 355 levels "00:03.0","00:04.0",...: 166 183 184 185 186 188 189 190 191 192 ..
## $ Speed : num 172 164 163 163 163 ...
## $ To.Win : Factor w/ 365 levels "0:00:00","0:05:21",...: 1 2 3 4 5 6 7 8 9 10 ...
## $ Eligible: Factor w/ 1 level "Yes": 1 1 1 1 1 1 1 1 1 1 ...
```

excel

The functions explained above don't require intallation of any library because they are in the R core, however to read excel files it is necessary to load the library readxl

```
library(readxl)
spanish_silver <- read_excel("C:/Users/Andrea/Desktop/spanish-silver.xls",
  sheet = "spanish-silver")
```

website

Subsets

Tibble

In all of the examples above the data were loaded as `data_frame`. However to display a sample of them and their visualization is more easy when the data is convert into a tibble

```
library(tibble)
pigeon_tb <- as_data_frame(pigeon)
pigeon_tb
```

```
## # A tibble: 400 x 11
##   Pos Breeder Pigeon Name Color Sex Ent Arrival Speed To.Win
##   <int> <fct>  <fct>  <fct> <fct> <fct> <int> <fct>  <dbl> <fct>
## 1     1     Texas ~ 19633~ ""    BCWF H     1 42:14.0 172. 0:00:~
## 2     2   Junior~ 0402~~ ""    SIWF H     1 47:36.0 164. 0:05:~
## 3     3   Jerry ~ 0404~~ Perc~ BB   H     1 47:41.0 163. 0:05:~
## 4     4   Alias~~ 2013~~ ""    BBSP H     1 47:43.0 163. 0:05:~
## 5     5   Greg G~ 5749~~ ""    BC   H     1 47:44.0 163. 0:05:~
## 6     6   Dal-Te~ 0032~~ ""    BC   H     1 47:51.0 163. 0:05:~
## 7     7   Greg G~ 5768~~ ""    BBWF C     2 47:53.0 163. 0:05:~
## 8     8   N C Sy~ 1067~~ ""    BC   H     1 47:57.0 163. 0:05:~
## 9     9   Baldwi~ 1194~~ ""    BB   H     1 48:02.0 163. 0:05:~
## 10    10   Texas ~ 19632~ ""    BC   H     2 48:03.0 163. 0:05:~
## # ... with 390 more rows, and 1 more variable: Eligible <fct>
```

This sort of view is obtained directly into the original dataframe with the function `head`.

```
head(pigeon, n=4)
```

```
##   Pos      Breeder      Pigeon      Name Color Sex Ent Arrival
## 1   1      Texas Outlaws 19633-AU15-FOYS      BCWF  H   1 42:14.0
## 2   2      Junior Juanich 0402-AU15-JRL      SIWF  H   1 47:36.0
## 3   3   Jerry Allensworth 0404-AU15-VITA Perch Potato  BB   H   1 47:41.0
## 4   4      Alias-Alias 2013-AU15-ALIA      BBSP   H   1 47:43.0
##   Speed To.Win Eligible
## 1 172.155 0:00:00      Yes
## 2 163.569 0:05:21      Yes
## 3 163.442 0:05:27      Yes
## 4 163.392 0:05:28      Yes
```

In this script most of the data will be used in tibbles.

Sampling

After loaded the dataset is useful sampling to know their data and identify steps to clean them.

```
library(dplyr)
pigeon_tb%>%sample_n(4)
```

```
## # A tibble: 4 x 11
##   Pos Breeder Pigeon Name Color Sex Ent Arrival Speed To.Win
##   <int> <fct>  <fct>  <fct> <fct> <fct> <int> <fct>  <dbl> <fct>
## 1   233 Family~ 0152~~ ""    BB   H     9 26:55.0 120. 0:44:~
```

```
## 2 374 Captai~ 1684~ "" BLK H 5 12:27.0 91.5 1:30:~
## 3 26 Proteg~ 0022~ "" BC H 1 48:39.0 162. 0:06:~
## 4 77 Jb & D 1207~ "" WHGR H 4 55:24.0 153. 0:13:~
## # ... with 1 more variable: Eligible <fct>
```

Extracting a percentage in the data set

```
pigeon_tb%>%sample_frac(0.03, replace=FALSE)
```

```
## # A tibble: 12 x 11
##   Pos Breeder Pigeon Name Color Sex Ent Arrival Speed To.Win
##   <int> <fct> <fct> <fct> <fct> <fct> <int> <fct> <dbl> <fct>
## 1 33 Silver~ 0036~ "" BC H 2 49:03.0 161. 0:06:~
## 2 101 Woodse~ 1502~ "" RC H 3 56:27.0 151. 0:14:~
## 3 215 Centme~ 1663~ "" BBWF H 2 19:04.0 127. 0:36:~
## 4 261 Skip's~ 5315~ "" BB H 5 36:05.0 113. 0:53:~
## 5 311 Kiko A~ 57087~ "" BBPI H 3 50:59.0 103. 1:08:~
## 6 249 Alonso~ 0313~ "" RED H 2 33:20.0 115. 0:51:~
## 7 74 Jb & D 1204~ "" BBPI H 3 54:56.0 153. 0:12:~
## 8 175 Dave D~ 15065~ "" BB H 3 06:09.0 140. 0:23:~
## 9 8 N C Sy~ 1067~ "" BC H 1 47:57.0 163. 0:05:~
## 10 264 Non Pa~ 0005~ "" BB H 3 39:43.0 110. 0:57:~
## 11 204 The Br~ 0566~ "" BC H 4 14:15.0 131. 0:32:~
## 12 17 Alias~ 2020~ "" BBSP H 2 48:16.0 163. 0:06:~
## # ... with 1 more variable: Eligible <fct>
```

Selecting columns

```
pigeon_tb%>%select(Pigeon, Color, Sex)
```

```
## # A tibble: 400 x 3
##   Pigeon Color Sex
##   <fct> <fct> <fct>
## 1 19633-AU15-FOYS BCWF H
## 2 0402-AU15-JRL SIWF H
## 3 0404-AU15-VITA BB H
## 4 2013-AU15-ALIA BBSP H
## 5 5749-AU15-SLI BC H
## 6 0032-AU15-DRPC BC H
## 7 5768-AU15-SLI BBWF C
## 8 1067-AU15-TXHC BC H
## 9 1194-AU15-TENT BB H
## 10 19632-AU15-FOYS BC H
## # ... with 390 more rows
```

Filters

- And &
- Or |

```
pigeon_tb%>%filter(Color=='BB' | Sex=='H')
```

```
## # A tibble: 396 x 11
##   Pos Breeder Pigeon Name Color Sex Ent Arrival Speed To.Win
##   <int> <fct> <fct> <fct> <fct> <fct> <int> <fct> <dbl> <fct>
## 1 1 Texas ~ 19633~ "" BCWF H 1 42:14.0 172. 0:00:~
## 2 2 Junior~ 0402~ "" SIWF H 1 47:36.0 164. 0:05:~
```

```
## 3      3 Jerry ~ 0404~ Perc~ BB      H      1 47:41.0 163. 0:05:~
## 4      4 Alias~ 2013~ ""      BBSP    H      1 47:43.0 163. 0:05:~
## 5      5 Greg G~ 5749~ ""      BC      H      1 47:44.0 163. 0:05:~
## 6      6 Dal-Te~ 0032~ ""      BC      H      1 47:51.0 163. 0:05:~
## 7      8 N C Sy~ 1067~ ""      BC      H      1 47:57.0 163. 0:05:~
## 8      9 Baldwi~ 1194~ ""      BB      H      1 48:02.0 163. 0:05:~
## 9     10 Texas ~ 19632~ ""      BC      H      2 48:03.0 163. 0:05:~
## 10    10 Redtex 0024~ ""      RED     H      1 48:03.0 163. 0:05:~
## # ... with 386 more rows, and 1 more variable: Eligible <fct>
```

```
pigeon_tb%>%filter(Color=='BB' & Sex=='H')
```

```
## # A tibble: 172 x 11
##       Pos Breeder Pigeon Name Color Sex      Ent Arrival Speed To.Win
##   <int> <fct>    <fct>    <fct> <fct> <fct> <int> <fct>    <dbl> <fct>
## 1      3 Jerry ~ 0404~ Perc~ BB      H      1 47:41.0 163. 0:05:~
## 2      9 Baldwi~ 1194~ ""      BB      H      1 48:02.0 163. 0:05:~
## 3     14 Goshen~ 5834~ ""      BB      H      1 48:12.0 163. 0:05:~
## 4     16 Flyhom~ 1531~ ""      BB      H      1 48:15.0 163. 0:06:~
## 5     24 Jb & D 1214~ ""      BB      H      1 48:36.0 162. 0:06:~
## 6     30 Churn ~ 9216~ ""      BB      H      1 48:48.0 162. 0:06:~
## 7     32 Alias~ 2049~ ""      BB      H      3 48:56.0 162. 0:06:~
## 8     35 Clear ~ 0263~ ""      BB      H      1 49:06.0 161. 0:06:~
## 9     38 Clear ~ 0235~ ""      BB      H      2 49:17.0 161. 0:07:~
## 10    40 Skip's~ 5302~ ""      BB      H      2 49:28.0 161. 0:07:~
## # ... with 162 more rows, and 1 more variable: Eligible <fct>
```

Order by

The “-” makes the order from the greatest to the shortest.

```
pigeon_tb%>%arrange(-Speed)
```

```
## # A tibble: 400 x 11
##       Pos Breeder Pigeon Name Color Sex      Ent Arrival Speed To.Win
##   <int> <fct>    <fct>    <fct> <fct> <fct> <int> <fct>    <dbl> <fct>
## 1      1 Texas ~ 19633~ ""      BCWF    H      1 42:14.0 172. 0:00:~
## 2      2 Junior~ 0402~ ""      SIWF    H      1 47:36.0 164. 0:05:~
## 3      3 Jerry ~ 0404~ Perc~ BB      H      1 47:41.0 163. 0:05:~
## 4      4 Alias~ 2013~ ""      BBSP    H      1 47:43.0 163. 0:05:~
## 5      5 Greg G~ 5749~ ""      BC      H      1 47:44.0 163. 0:05:~
## 6      6 Dal-Te~ 0032~ ""      BC      H      1 47:51.0 163. 0:05:~
## 7      7 Greg G~ 5768~ ""      BBWF    C      2 47:53.0 163. 0:05:~
## 8      8 N C Sy~ 1067~ ""      BC      H      1 47:57.0 163. 0:05:~
## 9      9 Baldwi~ 1194~ ""      BB      H      1 48:02.0 163. 0:05:~
## 10    10 Texas ~ 19632~ ""      BC      H      2 48:03.0 163. 0:05:~
## # ... with 390 more rows, and 1 more variable: Eligible <fct>
```

2.Cleaning data

Creation of new variables

Split

Split a string by an specific separator

```
library(dplyr)
library(tidyr)
```

```
## Warning: package 'tidyr' was built under R version 3.5.3
```

```
pigeon_tb%>%separate(Pigeon, sep='-', c('Num', 'id', 'det'))
```

```
## # A tibble: 400 x 13
##   Pos Breeder Num id det Name Color Sex Ent Arrival Speed
##   <int> <fct> <chr> <chr> <chr> <fct> <fct> <fct> <int> <fct> <dbl>
## 1 1 Texas ~ 19633 AU15 FOYS "" BCWF H 1 42:14.0 172.
## 2 2 Junior~ 0402 AU15 JRL "" SIWF H 1 47:36.0 164.
## 3 3 Jerry ~ 0404 AU15 VITA Perc~ BB H 1 47:41.0 163.
## 4 4 Alias~ 2013 AU15 ALIA "" BBSP H 1 47:43.0 163.
## 5 5 Greg G~ 5749 AU15 SLI "" BC H 1 47:44.0 163.
## 6 6 Dal-Te~ 0032 AU15 DRPC "" BC H 1 47:51.0 163.
## 7 7 Greg G~ 5768 AU15 SLI "" BBWF C 2 47:53.0 163.
## 8 8 N C Sy~ 1067 AU15 TXHC "" BC H 1 47:57.0 163.
## 9 9 Baldwi~ 1194 AU15 TENT "" BB H 1 48:02.0 163.
## 10 10 Texas ~ 19632 AU15 FOYS "" BC H 2 48:03.0 163.
## # ... with 390 more rows, and 2 more variables: To.Win <fct>,
## # Eligible <fct>
```

Concatenate

```
pigeon_tb%>%unite_('new', c('Pos', 'Sex'), sep = '-')
```

```
## # A tibble: 400 x 10
##   new Breeder Pigeon Name Color Ent Arrival Speed To.Win Eligible
##   <chr> <fct> <fct> <fct> <fct> <int> <fct> <dbl> <fct> <fct>
## 1 1-H Texas Ou~ 19633~ "" BCWF 1 42:14.0 172. 0:00:~ Yes
## 2 2-H Junior J~ 0402-A~ "" SIWF 1 47:36.0 164. 0:05:~ Yes
## 3 3-H Jerry Al~ 0404-A~ Perch~ BB 1 47:41.0 163. 0:05:~ Yes
## 4 4-H Alias-Al~ 2013-A~ "" BBSP 1 47:43.0 163. 0:05:~ Yes
## 5 5-H Greg Gla~ 5749-A~ "" BC 1 47:44.0 163. 0:05:~ Yes
## 6 6-H Dal-Tex ~ 0032-A~ "" BC 1 47:51.0 163. 0:05:~ Yes
## 7 7-C Greg Gla~ 5768-A~ "" BBWF 2 47:53.0 163. 0:05:~ Yes
## 8 8-H N C Synd~ 1067-A~ "" BC 1 47:57.0 163. 0:05:~ Yes
## 9 9-H Baldwin ~ 1194-A~ "" BB 1 48:02.0 163. 0:05:~ Yes
## 10 10-H Texas Ou~ 19632~ "" BC 2 48:03.0 163. 0:05:~ Yes
## # ... with 390 more rows
```

Variable type conversion

Suppose that Ent is a factor variable not a numeric one.

```
pigeon_tb$Ent<- as.factor(pigeon_tb$Ent)
pigeon_tb
```

```
## # A tibble: 400 x 11
##   Pos Breeder Pigeon Name Color Sex Ent Arrival Speed To.Win
##   <int> <fct> <fct> <fct> <fct> <fct> <fct> <fct> <dbl> <fct>
## 1 1 Texas ~ 19633~ "" BCWF H 1 42:14.0 172. 0:00:~
## 2 2 Junior~ 0402~ "" SIWF H 1 47:36.0 164. 0:05:~
## 3 3 Jerry ~ 0404~ Perc~ BB H 1 47:41.0 163. 0:05:~
## 4 4 Alias~ 2013~ "" BBSP H 1 47:43.0 163. 0:05:~
```

```
## 5      5 Greg G~ 5749~~ ""      BC      H      1      47:44.0 163. 0:05:~
## 6      6 Dal-Te~ 0032~~ ""      BC      H      1      47:51.0 163. 0:05:~
## 7      7 Greg G~ 5768~~ ""      BBWF     C      2      47:53.0 163. 0:05:~
## 8      8 N C Sy~ 1067~~ ""      BC      H      1      47:57.0 163. 0:05:~
## 9      9 Baldwi~ 1194~~ ""      BB      H      1      48:02.0 163. 0:05:~
## 10     10 Texas ~ 19632~~ ""      BC      H      2      48:03.0 163. 0:05:~
## # ... with 390 more rows, and 1 more variable: Eligible <fct>
```

if the variable is as string to convert them type into numeric the function is `as.numeric()`

4. Merging datasets

It's common that you have to merge many files to obtain your final dataset. In R at the same that Python you need to have the same colname in the key variable.

Joins

R has the SQL functions to join files, the key to join the data sets must have the same name in the files.

```
library(readxl)
athlete_country <- read_excel("C:/Users/Andrea/Desktop/python-ml-course-master/datasets/athletes/athlete_
  sheet = "Athelete_Country_Map")

athlete_sport <- read_excel("C:/Users/Andrea/Desktop/python-ml-course-master/datasets/athletes/athlete_
  sheet = "Athelete")

athlete_country
```

```
## # A tibble: 6,970 x 2
##   Athlete      Country
##   <chr>      <chr>
## 1 Michael Phelps United States
## 2 Natalie Coughlin United States
## 3 Aleksey Nemov Russia
## 4 Alicia Coutts Australia
## 5 Missy Franklin United States
## 6 Ryan Lochte United States
## 7 Allison Schmitt United States
## 8 Ian Thorpe Australia
## 9 Dara Torres United States
## 10 Cindy Klassen Canada
## # ... with 6,960 more rows
```

```
athlete_sport
```

```
## # A tibble: 6,975 x 2
##   Athlete      Sport
##   <chr>      <chr>
## 1 Michael Phelps Swimming
## 2 Natalie Coughlin Swimming
## 3 Aleksey Nemov Gymnastics
## 4 Alicia Coutts Swimming
## 5 Missy Franklin Swimming
## 6 Ryan Lochte Swimming
## 7 Allison Schmitt Swimming
## 8 Ian Thorpe Swimming
```

```
## 9 Dara Torres      Swimming
## 10 Cindy Klassen   Speed Skating
## # ... with 6,965 more rows
```

For this example the key is the column called 'Athlete'

```
inner_join(athlete_country, athlete_sport, by='Athlete')
```

```
## # A tibble: 6,994 x 3
##   Athlete      Country      Sport
##   <chr>      <chr>      <chr>
## 1 Michael Phelps United States Swimming
## 2 Natalie Coughlin United States Swimming
## 3 Aleksey Nemov   Russia      Gymnastics
## 4 Alicia Coutts   Australia    Swimming
## 5 Missy Franklin  United States Swimming
## 6 Ryan Lochte     United States Swimming
## 7 Allison Schmitt United States Swimming
## 8 Ian Thorpe      Australia    Swimming
## 9 Dara Torres     United States Swimming
## 10 Cindy Klassen  Canada       Speed Skating
## # ... with 6,984 more rows
```

the structure to reproduce left and right join is the same that the example above.

5. Dealing with NA

Counting the na values

```
sapply(pigeon_tb, function(x) sum(is.na(x)))
```

```
##      Pos Breeder  Pigeon   Name   Color    Sex    Ent  Arrival
##      0      0      0      0      0      0      0      0
##   Speed  To.Win Eligible
##      0      0      0
```

This is weird especially when I new that in name there are too many rows in blank, then one of the levels of the variable must be ""

```
levels(pigeon_tb$Name)
```

```
## [1] ""           "\"the Duck\"" "Alice"        "BATTLE BORN 27"
## [5] "Bella"      "BLACK NIGTH 9" "Canned Heat"  "Charlie"
## [9] "Christie"   "Color Me Hot"  "Edward"       "Elle"
## [13] "Gage"       "Gypsy"        "Jack Frost"   "Kingston"
## [17] "Lil Dat"    "Perch Potato"  "Pop's Pick"   "Rogue Brew"
## [21] "SEMPER FI 11"
```

The level "" is defining as NA

```
levels(pigeon_tb$Name)[levels(pigeon_tb$Name)==""]<-NA
levels(pigeon_tb$Name)
```

```
## [1] "\"the Duck\"" "Alice"        "BATTLE BORN 27" "Bella"
## [5] "BLACK NIGTH 9" "Canned Heat"  "Charlie"        "Christie"
## [9] "Color Me Hot"  "Edward"       "Elle"          "Gage"
## [13] "Gypsy"        "Jack Frost"   "Kingston"      "Lil Dat"
## [17] "Perch Potato"  "Pop's Pick"   "Rogue Brew"    "SEMPER FI 11"
```

```
pigeon_tb
```

```
## # A tibble: 400 x 11
##       Pos Breeder Pigeon Name Color Sex Ent Arrival Speed To.Win
##   <int> <fct>   <fct> <fct> <fct> <fct> <fct> <fct>   <dbl> <fct>
## 1     1     1 Texas ~ 19633~ <NA> BCWF H    1   42:14.0  172. 0:00:~
## 2     2     2 Junior~ 0402~~ <NA> SIWF H    1   47:36.0  164. 0:05:~
## 3     3     3 Jerry ~ 0404~~ Perc~ BB   H    1   47:41.0  163. 0:05:~
## 4     4     4 Alias~~ 2013~~ <NA> BBSP H    1   47:43.0  163. 0:05:~
## 5     5     5 Greg G~ 5749~~ <NA> BC   H    1   47:44.0  163. 0:05:~
## 6     6     6 Dal-Te~ 0032~~ <NA> BC   H    1   47:51.0  163. 0:05:~
## 7     7     7 Greg G~ 5768~~ <NA> BBWF C    2   47:53.0  163. 0:05:~
## 8     8     8 N C Sy~ 1067~~ <NA> BC   H    1   47:57.0  163. 0:05:~
## 9     9     9 Baldwi~ 1194~~ <NA> BB   H    1   48:02.0  163. 0:05:~
## 10    10    10 Texas ~ 19632~ <NA> BC   H    2   48:03.0  163. 0:05:~
## # ... with 390 more rows, and 1 more variable: Eligible <fct>
```

References

Van der Loo, M. and De Jonge, E. (2013) An introduction to data cleaning with R. https://cran.r-project.org/doc/contrib/de_Jonge+van_der_Loo-Introduction_to_data_cleaning_with_R.pdf

Hadley Wickham, Romain François, Lionel Henry and Kirill Müller (2019). dplyr: A Grammar of Data Manipulation. R package version 0.8.3. <https://CRAN.R-project.org/package=dplyr>

Hadley Wickham and Lionel Henry (2019). tidyr: Easily Tidy Data with ‘spread()’ and ‘gather()’ Functions. R package version 0.8.3. <https://CRAN.R-project.org/package=tidyr>

Kirill Müller and Hadley Wickham (2019). tibble: Simple Data Frames. R package version 2.1.3. <https://CRAN.R-project.org/package=tibble>

Hadley Wickham and Jennifer Bryan (2019). readxl: Read Excel Files. R package version 1.3.1. <https://CRAN.R-project.org/package=readxl>

Hadley Wickham (2017). tidyverse: Easily Install and Load the ‘Tidyverse’. R package version 1.2.1. <https://CRAN.R-project.org/package=tidyverse>

Usefull resources

- Presentation data cleaning Jonge. https://www.r-project.ro/conference2017/presentations/uRos2017_data-cleaning-workshop.pdf