

What better way to analyze your activities data from Apple Health and Runkeeper into R and generating some visualizations and counters. After that I will wrapping it together into a Shiny App.

Want do I want to achieve for now?

* Number of activities, steps, kilometers ect.
* Heatmap of last X years number of activities colored by amount.

**Loading and have a look at the data**

Export Runkeeper data, the option is available after login > account settings > export data > download your data.

For export and convert your Apple Health data follow my previous post [Analysing your Apple Health Data with Splunk](http://www.networkx.nl/data-science/visualising-health-data/).

runkeeper <- read.csv("data/cardioActivities.csv", stringsAsFactors=FALSE)

steps <- read.csv("data/StepCount.csv", stringsAsFactors = FALSE)

dim(runkeeper)

## [1] 676 14

dim(steps)

## [1] 47790 9

head(runkeeper)

## Activity.Id Date Type

## 1 47f59049-9bd5-4b9a-a63c-d51fca7e0bf8 2018-12-31 10:22:33 Running

## 2 7099bd74-7685-453e-a962-62ee711dc18e 2018-12-27 20:24:09 Running

## 3 c2250f5b-8567-47ef-97d7-504b174809e1 2018-12-23 13:28:04 Running

## 4 787c8438-d279-49fb-8bab-d1f348935264 2018-12-18 21:07:42 Boxing / MMA

## 5 2c41d494-204b-4be3-9b4e-d7acac5a3187 2018-12-16 13:04:44 Running

## 6 86b6429a-13a8-4122-b200-c3aadef271f5 2018-12-04 20:24:27 Boxing / MMA

## Route.Name Distance..km. Duration Average.Pace Average.Speed..km.h.

## 1 8.12 50:49 6:15 9.59

## 2 15.99 1:27:36 5:29 10.95

## 3 8.31 50:04 6:01 9.96

## 4 0.00 1:20:00 NA

## 5 10.51 56:47 5:24 11.10

## 6 0.00 1:00:00 NA

## Calories.Burned Climb..m. Average.Heart.Rate..bpm. Friend.s.Tagged Notes

## 1 772.0 83 134

## 2 1485.0 57 147

## 3 788.0 88 134

## 4 1032.4 0 NA

## 5 979.0 41 145

## 6 774.3 0 NA

## GPX.File

## 1 2018-12-31-102233.gpx

## 2 2018-12-27-202409.gpx

## 3 2018-12-23-132804.gpx

## 4

## 5 2018-12-16-130444.gpx

## 6

head(steps)

## sourceName sourceVersion

## 1 John’s iPhone 10.1.1

## 2 John’s iPhone 10.1.1

## 3 John’s iPhone 10.1.1

## 4 John’s iPhone 10.1.1

## 5 John’s iPhone 10.1.1

## 6 John’s iPhone 10.1.1

## device

## 1 <<HKDevice: 0x2836d9310>, name:iPhone, manufacturer:Apple, model:iPhone, hardware:iPhone8,1, software:10.1.1>

## 2 <<HKDevice: 0x2836db070>, name:iPhone, manufacturer:Apple, model:iPhone, hardware:iPhone8,1, software:10.1.1>

## 3 <<HKDevice: 0x2836d9d10>, name:iPhone, manufacturer:Apple, model:iPhone, hardware:iPhone8,1, software:10.1.1>

## 4 <<HKDevice: 0x2836dad50>, name:iPhone, manufacturer:Apple, model:iPhone, hardware:iPhone8,1, software:10.1.1>

## 5 <<HKDevice: 0x2836da080>, name:iPhone, manufacturer:Apple, model:iPhone, hardware:iPhone8,1, software:10.1.1>

## 6 <<HKDevice: 0x2836dabc0>, name:iPhone, manufacturer:Apple, model:iPhone, hardware:iPhone8,1, software:10.1.1>

## type unit creationDate startDate

## 1 StepCount count 2016-11-24 09:13:55 +0100 2016-11-24 08:55:10 +0100

## 2 StepCount count 2016-11-24 09:13:55 +0100 2016-11-24 09:10:09 +0100

## 3 StepCount count 2016-11-24 09:13:55 +0100 2016-11-24 09:11:11 +0100

## 4 StepCount count 2016-11-24 09:13:55 +0100 2016-11-24 09:12:13 +0100

## 5 StepCount count 2016-11-24 09:49:04 +0100 2016-11-24 09:13:19 +0100

## 6 StepCount count 2016-11-24 09:49:04 +0100 2016-11-24 09:33:57 +0100

## endDate value

## 1 2016-11-24 08:55:33 +0100 26

## 2 2016-11-24 09:11:11 +0100 114

## 3 2016-11-24 09:12:13 +0100 105

## 4 2016-11-24 09:13:19 +0100 25

## 5 2016-11-24 09:22:42 +0100 108

## 6 2016-11-24 09:42:33 +0100 89

str(runkeeper)

## 'data.frame': 676 obs. of 14 variables:

## $ Activity.Id : chr "47f59049-9bd5-4b9a-a63c-d51fca7e0bf8" "7099bd74-7685-453e-a962-62ee711dc18e" "c2250f5b-8567-47ef-97d7-504b174809e1" "787c8438-d279-49fb-8bab-d1f348935264" ...

## $ Date : chr "2018-12-31 10:22:33" "2018-12-27 20:24:09" "2018-12-23 13:28:04" "2018-12-18 21:07:42" ...

## $ Type : chr "Running" "Running" "Running" "Boxing / MMA" ...

## $ Route.Name : chr "" "" "" "" ...

## $ Distance..km. : num 8.12 15.99 8.31 0 10.51 ...

## $ Duration : chr "50:49" "1:27:36" "50:04" "1:20:00" ...

## $ Average.Pace : chr "6:15" "5:29" "6:01" "" ...

## $ Average.Speed..km.h. : num 9.59 10.95 9.96 NA 11.1 ...

## $ Calories.Burned : num 772 1485 788 1032 979 ...

## $ Climb..m. : int 83 57 88 0 41 0 62 0 0 0 ...

## $ Average.Heart.Rate..bpm.: int 134 147 134 NA 145 NA 145 NA NA NA ...

## $ Friend.s.Tagged : chr "" "" "" "" ...

## $ Notes : chr "" "" "" "" ...

## $ GPX.File : chr "2018-12-31-102233.gpx" "2018-12-27-202409.gpx" "2018-12-23-132804.gpx" "" ...

str(steps)

## 'data.frame': 47790 obs. of 9 variables:

## $ sourceName : chr "John’s iPhone" "John’s iPhone" "John’s iPhone" "John’s iPhone" ...

## $ sourceVersion: chr "10.1.1" "10.1.1" "10.1.1" "10.1.1" ...

## $ device : chr "<<HKDevice: 0x2836d9310>, name:iPhone, manufacturer:Apple, model:iPhone, hardware:iPhone8,1, software:10.1.1>" "<<HKDevice: 0x2836db070>, name:iPhone, manufacturer:Apple, model:iPhone, hardware:iPhone8,1, software:10.1.1>" "<<HKDevice: 0x2836d9d10>, name:iPhone, manufacturer:Apple, model:iPhone, hardware:iPhone8,1, software:10.1.1>" "<<HKDevice: 0x2836dad50>, name:iPhone, manufacturer:Apple, model:iPhone, hardware:iPhone8,1, software:10.1.1>" ...

## $ type : chr "StepCount" "StepCount" "StepCount" "StepCount" ...

## $ unit : chr "count" "count" "count" "count" ...

## $ creationDate : chr "2016-11-24 09:13:55 +0100" "2016-11-24 09:13:55 +0100" "2016-11-24 09:13:55 +0100" "2016-11-24 09:13:55 +0100" ...

## $ startDate : chr "2016-11-24 08:55:10 +0100" "2016-11-24 09:10:09 +0100" "2016-11-24 09:11:11 +0100" "2016-11-24 09:12:13 +0100" ...

## $ endDate : chr "2016-11-24 08:55:33 +0100" "2016-11-24 09:11:11 +0100" "2016-11-24 09:12:13 +0100" "2016-11-24 09:13:19 +0100" ...

## $ value : int 26 114 105 25 108 89 329 284 89 18 ...

From the Runkeeper data we need Date and from Apple Health StepCount we need the endDate (thats when your step is done). Both has type chr. I could convert it as date, but I leave the data what it is and will do convert it when necessary.

**Create new variables**

First we load the lubridate package.

library(lubridate)

I will create some new variables and convert Date and endDate as Date so I can extract the year with the year function of the lubridate package.

runkeeper$year <- year(as.Date(runkeeper$Date, origin = '1900-01-01'))

steps$year <- year(as.Date(steps$endDate, origin = '1900-01-01'))

summary(runkeeper$year)

## Min. 1st Qu. Median Mean 3rd Qu. Max.

## 2010 2013 2016 2015 2017 2018

summary(steps$year)

## Min. 1st Qu. Median Mean 3rd Qu. Max.

## 2016 2017 2017 2017 2018 2019

As you can see in the above summary of both dataframes, Runkeeper has more years than steps. That doesn’t matter becouse we are now only looking for 2018 in the Shiny App and I will group it by year. In the app I create a year slider.

I will parse the period with hour, minuts ans seconds of the variable Duration. And I will calculate the duration in minutus. Some of the Duration variables has no leading zero for the hour and I cannot use it with the function hms.

Add a leading zero if not exists.

runkeeper$Duration[1:5]

## [1] "50:49" "1:27:36" "50:04" "1:20:00" "56:47"

runkeeper$Duration <- ifelse(nchar(runkeeper$Duration) < 6,

paste0("0:", runkeeper$Duration), runkeeper$Duration)

runkeeper$Duration[1:5]

## [1] "0:50:49" "1:27:36" "0:50:04" "1:20:00" "0:56:47"

Now I can parse the Duration with hms.

runkeeper$lub <- hms(runkeeper$Duration)

runkeeper$time\_minutes <- hour(runkeeper$lub)\*60 +

minute(runkeeper$lub) +

second(runkeeper$lub)/60

runkeeper$lub[1:5]

## [1] "50M 49S" "1H 27M 36S" "50M 4S" "1H 20M 0S" "56M 47S"

runkeeper$time\_minutes[1:5]

## [1] 50.81667 87.60000 50.06667 80.00000 56.78333

**Group the data by year**

First we load the package dplyr.

library(dplyr)

Group both dataframes by year and do some summarises like count and sum of kilometers, climb, calories and duration

grouped\_runkeeper\_year <- runkeeper %>%

group\_by(year) %>%

summarise(cnt = n(),

km = sum(Distance..km.),

climb = sum(Climb..m.),

calories = sum(Calories.Burned),

duration = sum(time\_minutes, na.rm = TRUE))

grouped\_steps\_year <- steps %>%

group\_by(year) %>%

summarise(cnt = sum(value))

grouped\_runkeeper\_year

## # A tibble: 9 x 6

## year cnt km climb calories duration

## <dbl> <int> <dbl> <int> <dbl> <dbl>

## 1 2010 3 10.8 0 901 71.8

## 2 2011 96 657. 1951 56333 3864.

## 3 2012 46 410. 1458 36876 2278.

## 4 2013 46 258. 488 22654 1456.

## 5 2014 24 175. 698 14382 992.

## 6 2015 117 1202. 5910 98518 7537.

## 7 2016 93 661. 3076 65275. 5397.

## 8 2017 128 408. 1569 89872. 7404.

## 9 2018 123 366. 1818 86792. 7062.

grouped\_steps\_year

## # A tibble: 4 x 2

## year cnt

## <dbl> <int>

## 1 2016 328759

## 2 2017 3360951

## 3 2018 4366289

## 4 2019 3692

**Some simple visualizations**

library(ggplot2)

library(RColorBrewer)

grouped\_runkeeper\_year %>%

ggplot(aes(x=year, y=cnt )) +

geom\_bar(stat = "identity", col="white", fill="#ee8300") +

geom\_text(aes(label=cnt), hjust=1.2, color="white", size=3.5) +

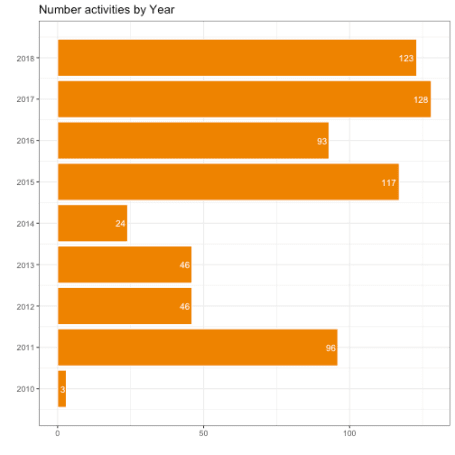
labs(x="", y="", title="Number activities by Year") +

scale\_x\_continuous(breaks=seq(min(grouped\_runkeeper\_year$year),

max(grouped\_runkeeper\_year$year),1)) +

coord\_flip() +

theme\_bw()



grouped\_steps\_year %>%

ggplot(aes(x=year, y=cnt )) +

geom\_bar(stat = "identity", col="white", fill="#ee8300") +

geom\_text(aes(label=cnt), hjust=1.2, color="white", size=3.5) +

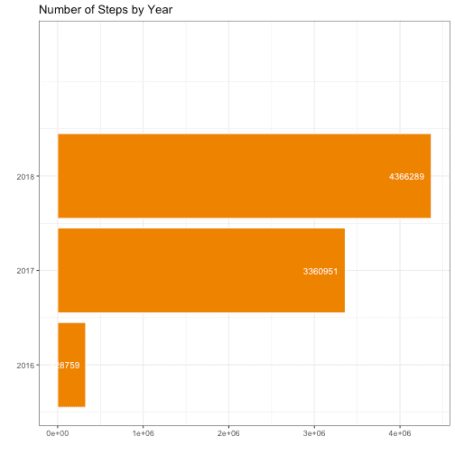
labs(x="", y="", title="Number of Steps by Year") +

scale\_x\_continuous(breaks=seq(min(grouped\_runkeeper\_year$year),

max(grouped\_runkeeper\_year$year),1)) +

coord\_flip() +

theme\_bw()



Add a heatmap of number of activities last 3 years from now.

calendar\_heatmap <- runkeeper %>%

select(Date,time\_minutes,year) %>%

filter(year >= year(now()) - 3) %>%

mutate(

week = week(as.Date(Date)),

wday = wday(as.Date(Date), week\_start = 1),

month = month(as.Date(Date), label = TRUE, abbr = TRUE),

day = weekdays(as.Date(Date))

)

cols <- rev(rev(brewer.pal(7,"Oranges"))[1:5])

calendar\_heatmap %>%

ggplot(aes(month, reorder(day, -wday), fill = time\_minutes)) +

geom\_tile(colour = "white") +

scale\_fill\_gradientn('Activity \nMinutes', colours = cols) +

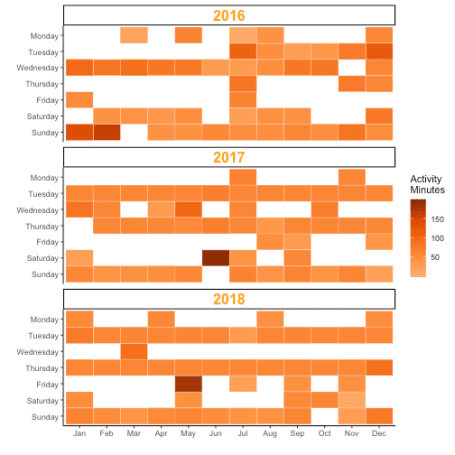
facet\_wrap(~ year, ncol = 1) +

theme\_classic() +

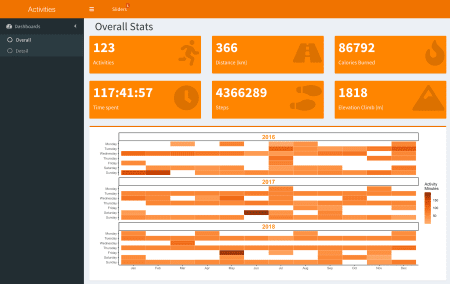
theme(strip.text.x = element\_text(size = 16, face = "bold", colour = "orange")) +

ylab("") +

xlab("")



**Screenshot Shiny App**



**SessionInfo**

sessionInfo()

## R version 3.5.1 (2018-07-02)

## Platform: x86\_64-apple-darwin15.6.0 (64-bit)

## Running under: macOS 10.14.2

##

## Matrix products: default

## BLAS: /System/Library/Frameworks/Accelerate.framework/Versions/A/Frameworks/vecLib.framework/Versions/A/libBLAS.dylib

## LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib

##

## locale:

## [1] en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8

##

## attached base packages:

## [1] stats graphics grDevices utils datasets methods base

##

## other attached packages:

## [1] RWordPress\_0.2-3 knitr\_1.20

## [3] bindrcpp\_0.2.2 RColorBrewer\_1.1-2

## [5] ggplot2\_3.0.0 shinydashboardPlus\_0.6.0

## [7] shinydashboard\_0.7.1 shiny\_1.2.0

## [9] dplyr\_0.7.8 lubridate\_1.7.4

##

## loaded via a namespace (and not attached):

## [1] Rcpp\_0.12.19 prettyunits\_1.0.2 ps\_1.2.1

## [4] utf8\_1.1.4 assertthat\_0.2.0 rprojroot\_1.3-2

## [7] digest\_0.6.17 mime\_0.5 R6\_2.2.2

## [10] plyr\_1.8.4 backports\_1.1.2 evaluate\_0.12

## [13] highr\_0.7 pillar\_1.3.0 rlang\_0.3.0.1

## [16] lazyeval\_0.2.1 curl\_3.2 rstudioapi\_0.8

## [19] callr\_3.0.0 rmarkdown\_1.10 desc\_1.2.0

## [22] labeling\_0.3 devtools\_2.0.1 stringr\_1.3.1

## [25] RCurl\_1.95-4.11 munsell\_0.5.0 compiler\_3.5.1

## [28] httpuv\_1.4.5.1 pkgconfig\_2.0.2 base64enc\_0.1-3

## [31] pkgbuild\_1.0.2 htmltools\_0.3.6 tidyselect\_0.2.5

## [34] tibble\_1.4.2 XML\_3.98-1.16 fansi\_0.4.0

## [37] crayon\_1.3.4 withr\_2.1.2 later\_0.7.5

## [40] bitops\_1.0-6 grid\_3.5.1 jsonlite\_1.5

## [43] xtable\_1.8-3 gtable\_0.2.0 magrittr\_1.5

## [46] scales\_1.0.0 cli\_1.0.1 stringi\_1.2.4

## [49] fs\_1.2.6 promises\_1.0.1 remotes\_2.0.2

## [52] testthat\_2.0.0 tools\_3.5.1 glue\_1.3.0

## [55] purrr\_0.2.5 hms\_0.4.2 processx\_3.2.0

## [58] pkgload\_1.0.2 yaml\_2.2.0 colorspace\_1.3-2

## [61] sessioninfo\_1.1.1 memoise\_1.1.0 bindr\_0.1.1

## [64] XMLRPC\_0.3-1 usethis\_1.4.0

Well wrap it up into a Shiny App which can be found on my GitHub.