When starting analyzing last.fm scrobbles with the last.week and last.year function I was always missing some plots or a pure data table. That is why I developed the package “analyzelastfm” as a simple R6 implementation. I wanted to have different album statistics, as e.g the #of plays per album, divided by the # of tracks on the album. This is now implemented. To get music listening statistic you would start with:

library(analyzelastfm)

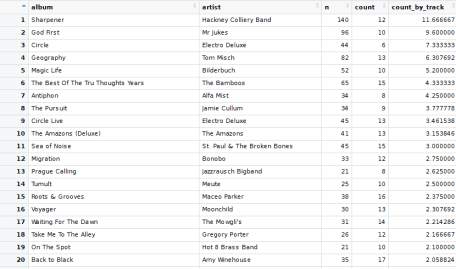
First it allows you to import your last year data by using the last.fm REST API. Therefore you need to have a last.fm API key. This can be derived by simply going to the [last.fm API website](https://www.last.fm/api/account/create). From there you will get the 32 character long key. After receiving the key I got my last.fm data by:

api\_key <- "PLEASE ENTER YOUR KEY HERE"  
data <- UserData$new("zappingseb",api\_key,2018)

The data object now contains the function albumstats that allows me to get the specific information I wanted. I’ll exclude the artist “Die drei ???” as it’s a kids detective story from Germany I listen to a lot and which’s albums are split into 1 minute tracks, that really screws up my statistic.

View(data$albumstats(  
 sort\_by="by\_album\_count", # album track plays / nr tracks on album  
 exclude\_artist="Die drei ???", # exclude my audio book favorite  
 exclude\_album=c(""), # exclude tracks without album name  
 min\_tracks=5) # have minimum 5 tracks on the album (NO EPs)  
)

The result looks like that:

Album statistics of 2018 of zappingseb

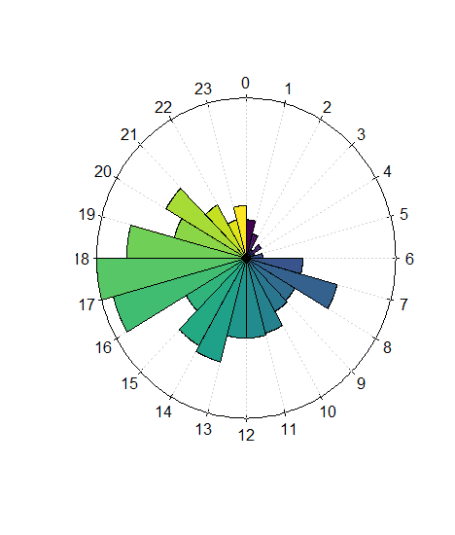
The statistic shows n the number of plays, count the number of tracks on the album and count\_by\_track=n/count .

The top 5 albums you can find here:

* [Hackney Collery Band — Sharpener](https://www.amazon.de/gp/product/B0762Y3T14/ref=as_li_tl?ie=UTF8&tag=zappingseb-21&camp=1638&creative=6742&linkCode=as2&creativeASIN=B0762Y3T14&linkId=c2e12707d54eee718e3b30b91a967fed)
* [Mr Jukes — God first](https://www.amazon.de/gp/product/B06XS8TTS4/ref=as_li_tl?ie=UTF8&tag=zappingseb-21&camp=1638&creative=6742&linkCode=as2&creativeASIN=B06XS8TTS4&linkId=f712ea96a435a93e069b778990d66645)
* [Electro Deluxe — Circle](https://www.amazon.de/gp/product/B01M24YCCH/ref=as_li_tl?ie=UTF8&tag=zappingseb-21&camp=1638&creative=6742&linkCode=as2&creativeASIN=B01M24YCCH&linkId=8f3115b23d7953db404d1710a6cb797d)
* [Tom Misch — Geography](https://www.amazon.de/gp/product/B07BS3SFDC/ref=as_li_tl?ie=UTF8&tag=zappingseb-21&camp=1638&creative=6742&linkCode=as2&creativeASIN=B07BS3SFDC&linkId=32d00c3201b0dc902a9c830a459ed266)
* [Bilderbuch — Magic Life](https://www.amazon.de/gp/product/B01M0G01UU/ref=as_li_tl?ie=UTF8&camp=1638&creative=6742&creativeASIN=B01M0G01UU&linkCode=as2&tag=zappingseb-21&linkId=abf7103142c4e6c2895766cd1c2d3dcc)

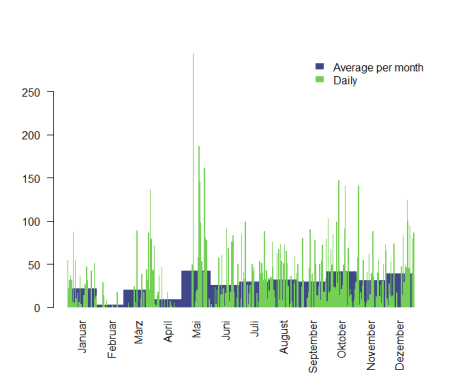
Additionally to such calculations, I was also interested in when I was listening to music. Therefore I added some plots to the package. The first one is the listening clock:

data$clock.plot()

Clock plot for my music history

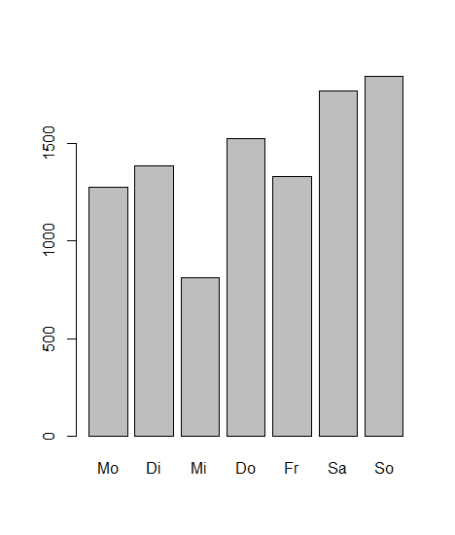
My most important statistic though, is which time of the year I spent listening to music. I’m most interested in some specific days and an average play/month, that can tell me in which mood I was in. Therefore I created the function daily\_month\_plot . It plots the average plays per month and the daily plays in one plot. The average plays per month are visualized behind the daily spikes. Here you can see that February was a really quite month for me.

data$daily\_month\_plot()



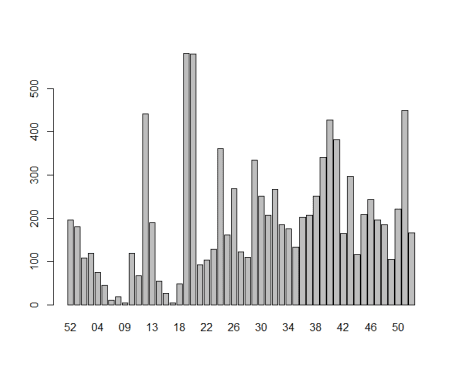
For some simple statistics, I also included an aggregation plotting function called barplots . It can plot aggregations per weekday, week, month or day. The data for this function is provided by the function bar\_data .

data$barplots("weekdays")



weekly (starting with week 52 of 2017):

data$barplots(“week”)



Where I can clearly see that I was not really listening to a lot of music during the beginning of the year and my listening activity increased dramatically by week 18.