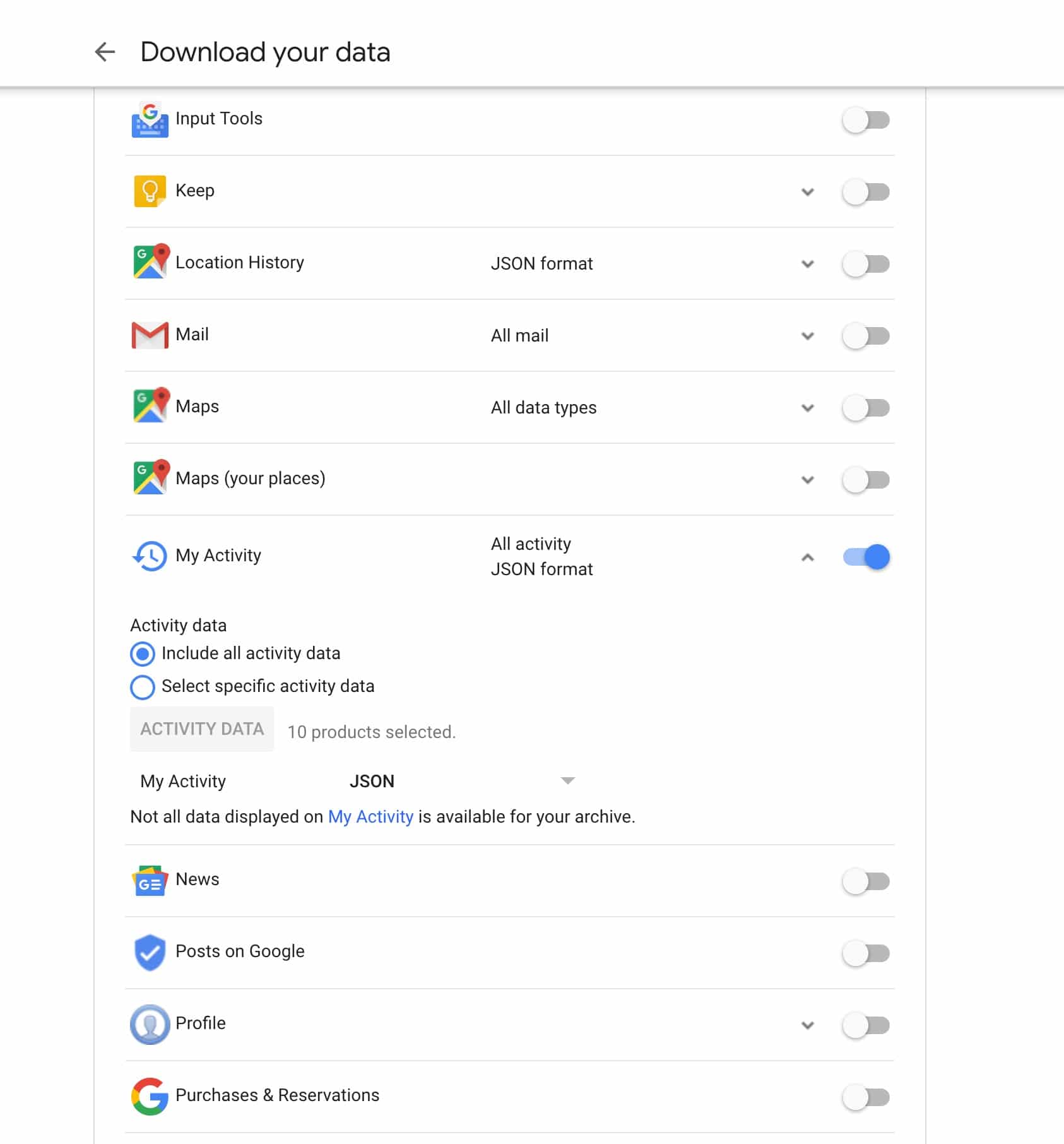
This is an attempt to see how the data that are collected from us, can also be used for the betterment of us – one’s self. When companies are so interested in collecting our personal data to show a push in Quarterly revenues, Why not use our own Data Science skills and get some useful insight that can help our life.

Embarking on that journey, I decided to start with my Android Mobile App usage using the data that Google lets us download. The reason I’m posting this article is for everyone else to introspect their usage and learn about it. So, for someone to replicate my results, I’ll explain in the below steps starting from how to download the data.

**How to download your Android Mobile usage data:**

* Go to your [Google Account](https://myaccount.google.com/?pli=1) (using the gmail id that you have used in your Android Phone) – Skip this if you are already logged in.
* Go to [Google Takeout](https://takeout.google.com/)
* Click **SELECT NONE** button and scroll down to see **My Activity**
* Select My Activity (enabling the grey/blue button) and click the down arrow to select JSON format and click Next (bottom-most button)
* In the next screen, you can select your preferred method and file format of download and click Create Archive.

[](https://datascienceplus.com/wp-content/uploads/2019/02/google_data.jpg)

Once your data is ready to be downloaded, You’d be notified to download it. The downloaded file would be a compressed-file (most like Zip – based on what you selected in the last screen). So unzip it and keep the JSON file ready for us to proceed further.

**Getting started with Analysis**

**Packages used**

We are going to use the following packages for our Analysis.

library(jsonlite)

library(tidyverse)

library(lubridate)

library(ggrepel)

library(viridis)

library(gganimate)

library(cowplot)

library(ggthemes)

If you have not got any of the above mentioned packages, All of them are available on CRAN. So, use install.packages() to install missing packages.

**Loading Data**

We’ve got a JSON input file and It’d better for us to do analysis with Dataframe (since it fits well with tidyverse). But this data processing is as easy as Pie with the help of jsonlite‘s omJSON()

function that takes a JSON file and outputs a flattened Dataframe.

me <- jsonlite::fromJSON("MyActivity.json")

With the above code, we are good to start with our Data Preprocessing.

**Data Preprocessing**

One of the columns that we would use in our Analysis, time

comes as a string with Data and Time in it. But for us to process time as time – it has to be in date-time format, so we’ll use the function parsedatetime() for converting string to date-time and withtz()

to change the time zone. As I live in India, I’ve converted it to Indian Standard Time. Please use your appropriate time zone for conversion.

# converting date-time in string to date-time format along with time-zone conversion

me$time\_ist <- with\_tz(parse\_date\_time(me$time),"Asia/Calcutta")

# remove incomplete years and irrelevant years too - Kept 2019 to see just January if required

me <- filter(me, year(time\_ist) %in% c(2017,2018,2019))

As you can see in the above code, We’ve also filtered our data to only include the years 2017, 2018 and 2019. This is simply to avoid partial data. Even though 2019 is also partial data, I’ve decided to include it in the main data to compare my apps across these three years. With that we’re good with the data preprocessing and let’s head to Analysis.

**Data Note**

A caveat that has to be called out here is that this Activity data includes everything from the app that you open and the apps that are shown up in the notification – so we’re proceeding further with an assumption that every notification is also part of our interaction (or at least in my case, Every time a notification popus up, I tend to check it).

**Sample/Head Data**

# Sample

tibble::tibble(head(me))

*# A tibble: 6 x 1*

*`head(me)`$header $title $titleUrl $time $products $details $time\_ist*

*1 OnePlus Launcher Used On… https://play… 2019-… 2019-02-12 12:34:01*

*2 صلاتك Salatuk (Pr… Used صل… https://play… 2019-… 2019-02-12 12:34:01*

*3 Google Chrome: Fa… Used Go… https://play… 2019-… 2019-02-12 12:19:23*

*4 Firefox Browser f… Used Fi… https://play… 2019-… 2019-02-12 12:18:38*

*5 Hangouts Used Ha… https://play… 2019-… 2019-02-12 12:18:15*

*6 Gmail Used Gm… https://play… 2019-… 2019-02-12 12:17:50*

**Top apps – each year**

The code in this section will make three different plots of top app usage for three different years and finally stitch all of them together.

# Top apps

me\_count <- me %>%

group\_by(year = year(time\_ist),header) %>%

count() %>%

arrange(desc(n)) %>%

ungroup() %>%

group\_by(year) %>%

top\_n(20,n) #%>% #View()

#mutate(header = fct\_reorder(header,n)) %>%

me\_count %>%

filter(year %in% "2017") %>%

ggplot(aes(fct\_reorder(header,n),n, label = n)) +

geom\_bar(aes(fill = n),stat = "identity") +

#scale\_y\_log10() +

coord\_flip() +

theme(axis.text.y = element\_text(angle = 0, hjust = 1,size = 8)) +

scale\_fill\_viridis() +

theme\_minimal() +

theme(legend.position="none") +

labs(

title = "Most used 20 Apps",

subtitle = "2017",

x = "App name"

) -> y1

me\_count %>%

filter(year %in% "2018") %>%

ggplot(aes(fct\_reorder(header,n),n, label = n)) +

geom\_bar(aes(fill = n),stat = "identity") +

#scale\_y\_log10() +

coord\_flip() +

theme(axis.text.y = element\_text(angle = 0, hjust = 1,size = 8)) +

scale\_fill\_viridis() +

theme\_minimal() +

theme(legend.position="none") +

labs(

subtitle = "2018",

x = "App name"

) -> y2

me\_count %>%

filter(year %in% "2019") %>%

ggplot(aes(fct\_reorder(header,n),n, label = n)) +

geom\_bar(aes(fill = n),stat = "identity") +

#scale\_y\_log10() +

coord\_flip() +

theme(axis.text.y = element\_text(angle = 0, hjust = 1,size = 8)) +

scale\_fill\_viridis() +

theme\_minimal() +

theme(legend.position="none") +

labs(

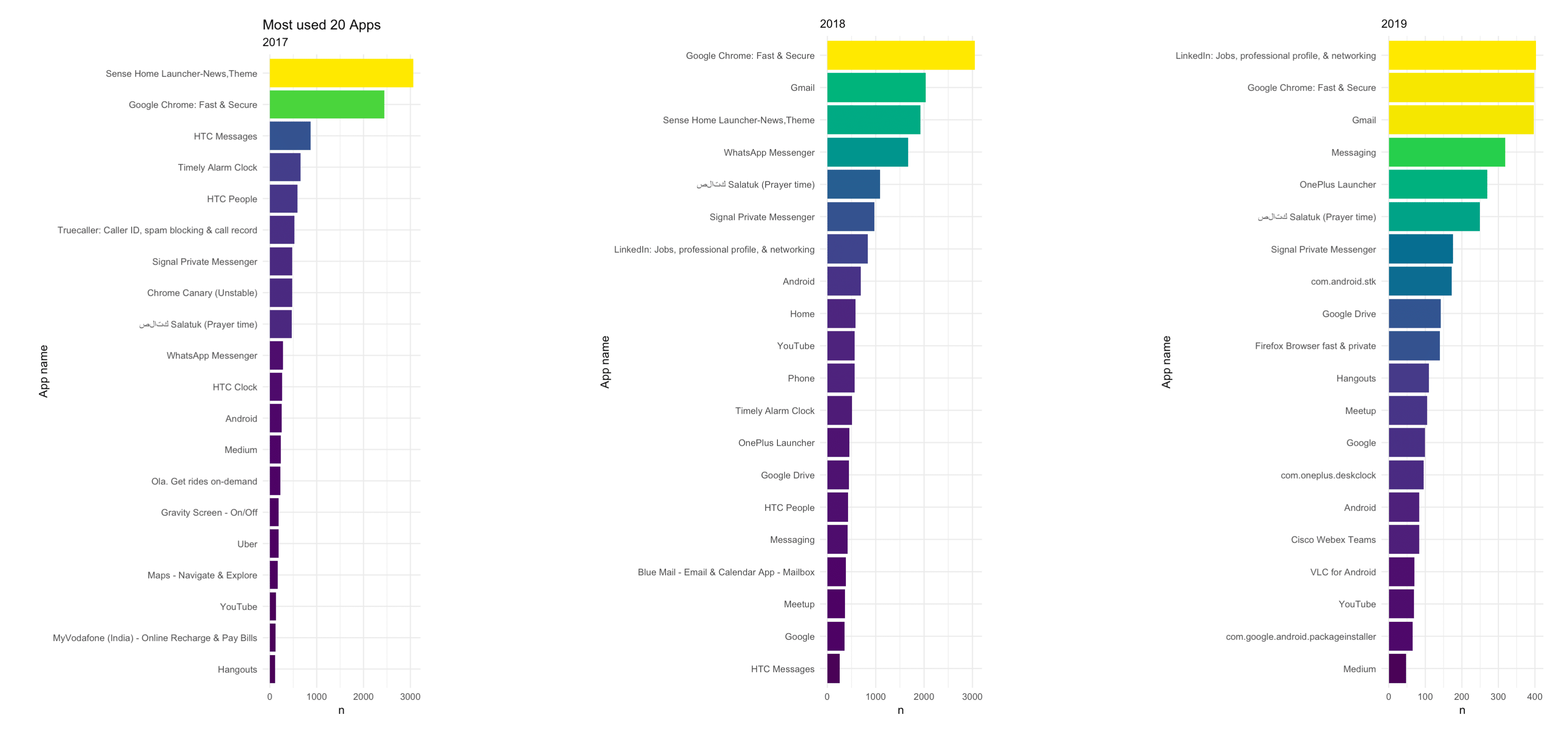
subtitle = "2019",

x = "App name"

) -> y3

cowplot::plot\_grid(y1,y2,y3, ncol = 3, scale = 0.7, vjust = 0, label\_size = 8)

Gives this plot:

[](https://datascienceplus.com/wp-content/uploads/2019/02/app_usage_by_year.png)

This plot clearly tells me how my app usage pattern has changed or evolved over time. It also denotes my change of handset from HTC One (that comes with Sense Launcher) to my recent Oneplus (that comes with Oneplus Launcher). You can also notice that I’ve moved on from Whatsapp to Signal messenger.

**Overall Daily usage Trend**

# Overall Daily usage trend

me %&#x3E;%

filter(!str\_detect(header,&#x22;com.&#x22;)) %&#x3E;%

filter(as.Date(time\_ist) &#x3E;= as.Date(&#x22;2017-01-01&#x22;)) %&#x3E;%

group\_by(Date = as.Date(time\_ist)) %&#x3E;%

count(n = n()) %&#x3E;%

ggplot(aes(Date,n, group = 1, color = &#x22;red&#x22;)) +

geom\_line(aes(alpha = 0.8),show.legend = FALSE) +

stat\_smooth() +

# Courtesy: https://stackoverflow.com/a/42929948

scale\_x\_date(date\_breaks = &#x22;1 month&#x22;, date\_labels = &#x22;%b %Y&#x22;) +

labs(

title = &#x22;Daily-wise usage&#x22;,

subtitle = &#x22;2+ years (including some 2019)&#x22;,

x = &#x22;Months&#x22;,

y = &#x22;# of Interactions&#x22;

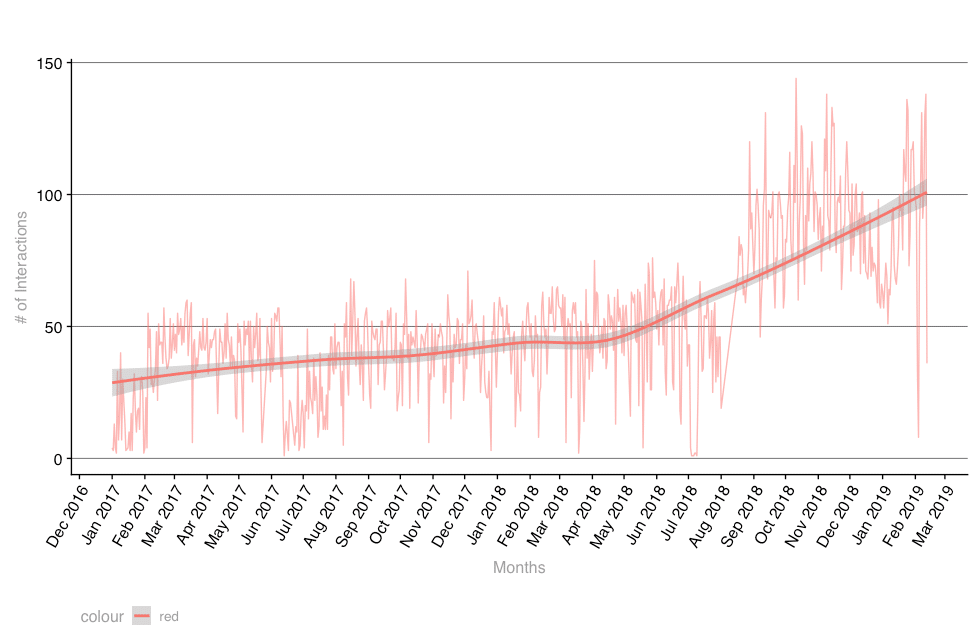
) +

theme(axis.text.x=element\_text(angle=60, hjust=1))+

theme(legend.position=&#x22;none&#x22;) +

ggthemes::theme\_hc(style = &#x22;darkunica&#x22;)

Gives this plot:

[](https://datascienceplus.com/wp-content/uploads/2019/02/daily-usage-trend.png)

This plot scared me the most. My phone usage has really spiked since I bought a new phone which doesn’t seem to be a good sign for my productivity.

**Weekday vs Weeknd**

This plot is to see if I’m really a phone addict even while at home with my family.

me %>%

filter(!str\_detect(header,"com.")) %>%

group\_by(Date = as.Date(time\_ist)) %>%

count(n = n()) %>%

mutate(year = as.factor(year(Date)),

weekday = weekdays(Date, abbr = TRUE)) %>%

mutate(what\_day = ifelse(weekday %in% c("Sat","Sun"),"Weekend","Weekday")) %>%

filter(year %in% c(2017,2018)) %>%

group\_by(year,what\_day) %>%

summarize(n = mean(n)) %>%

ggplot(aes(fct\_relevel(what\_day, c("Weekday","Weekend")),

n, group = year, color = year)) +

geom\_line() +

labs(

title = "Weekday vs Weekend usage",

subtitle = "For two years",

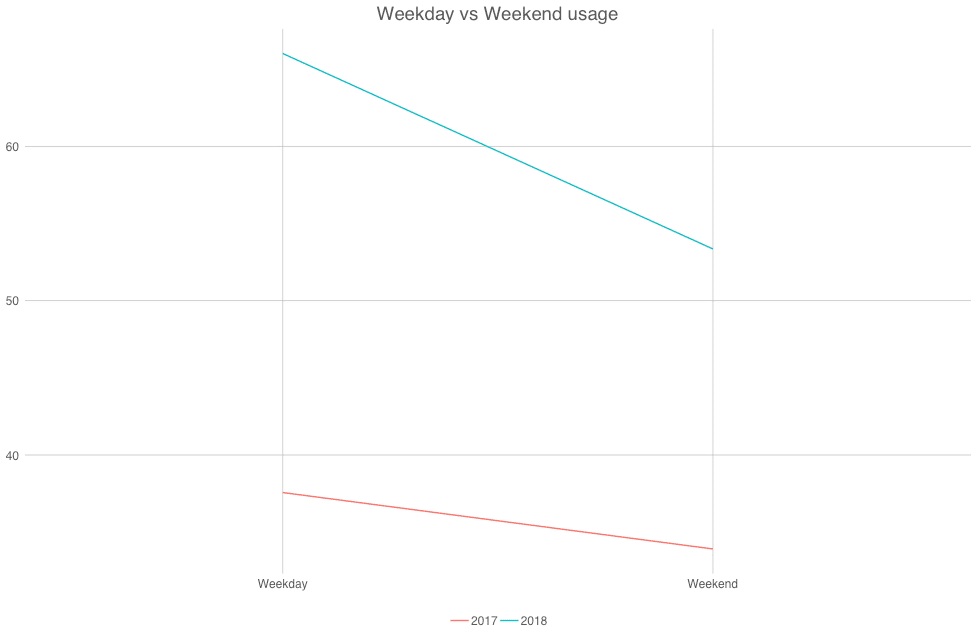
x = "Weekday / Weekend",

y = "# of Interactions"

) +

ggthemes::theme\_excel\_new()

Gives this plot:

[](https://datascienceplus.com/wp-content/uploads/2019/02/weekday_weekend-1.png)

Luckily, it turns out I’m not that level of an Addict I worried I would be.

**Messaging usage**

Over the years, I’ve used a variety of messaging apps from normal SMS to IMs.

# Messaging Usage

p <- me %>%

filter(str\_detect(tolower(header), regex("signal|message|whatsapp"))) %>%

mutate(ym = as.Date(paste0(format(as.Date(time\_ist),"%Y-%m"),"-01"))) %>%

group\_by(ym) %>%

count() %>%

#https://community.rstudio.com/t/tweenr-gganimate-with-line-plot/4027/10

ggplot(aes(ym,n, group = 1)) + geom\_line(color = "green") +

geom\_point() +

ggthemes::theme\_hc(style = "darkunica") +

theme(axis.text.x = element\_text(colour = "white",

angle = 60),

axis.text.y = element\_text(colour = "white")) +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%b %Y") +

labs(

title = "Messaging usage",

x = "Year-Month"

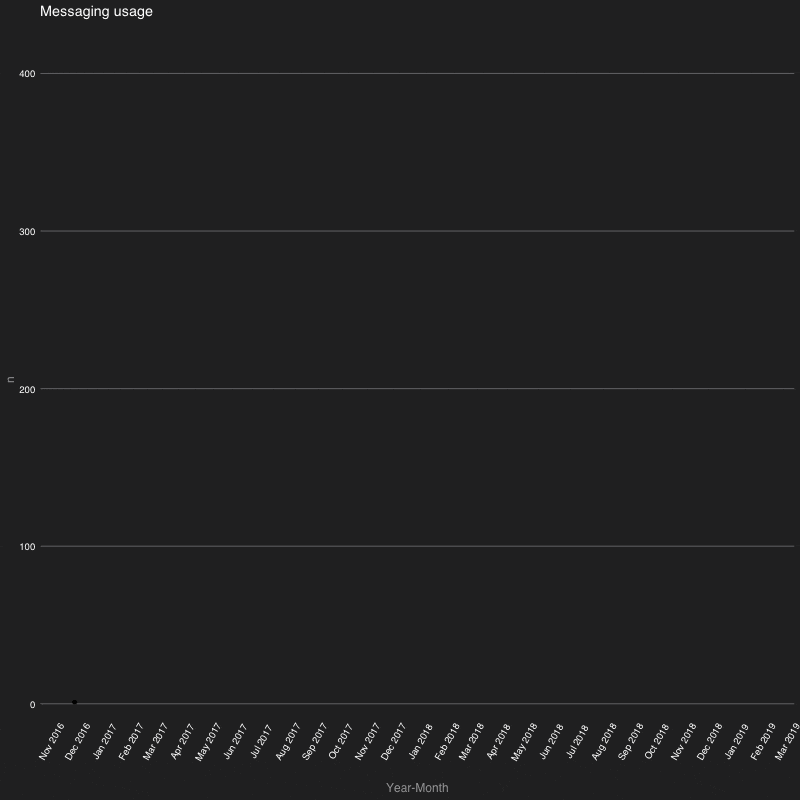
) +

transition\_reveal(ym) +

ease\_aes('cubic-in-out')

animate(p, nframes = 20, renderer = gifski\_renderer("msging.gif"), width = 800, height = 800)

Gives this animation:

[](https://datascienceplus.com/wp-content/uploads/2019/02/msging.gif)

This graph shows how this is one of the drivers of my overall phone usage. Similar spikes around similar period.

**Youtube usage**

# YouTube Usage

yt <- me %>%

filter(header %in% "YouTube") %>%

mutate(ym = as.Date(paste0(format(as.Date(time\_ist),"%Y-%m"),"-01"))) %>%

group\_by(ym) %>%

count() %>%

#https://community.rstudio.com/t/tweenr-gganimate-with-line-plot/4027/10

ggplot(aes(ym,n, group = 1)) + geom\_line(color = "red") +

geom\_point() +

ggthemes::theme\_hc(style = "darkunica") +

theme(axis.text.x = element\_text(colour = "white",

angle = 60),

axis.text.y = element\_text(colour = "white")) +

scale\_x\_date(date\_breaks = "1 month", date\_labels = "%b %Y") +

labs(

title = "YouTube usage",

x = "Year-Month"

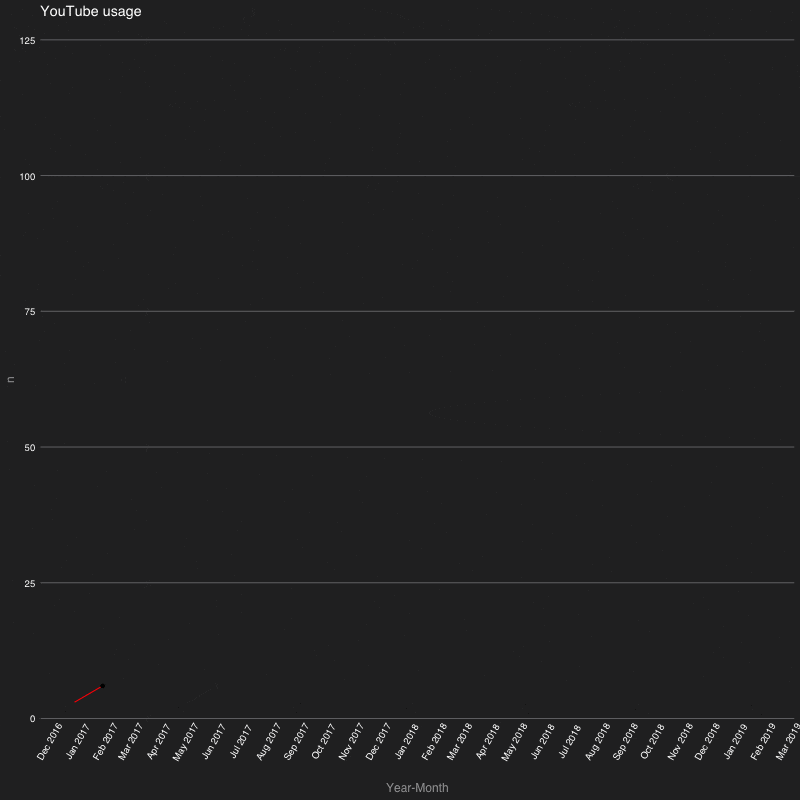
) +

transition\_reveal(ym) +

ease\_aes('quintic-in-out')

#anim\_save("yt.gif", yt , width = 600, height = 600)

animate(yt, nframes = 10, renderer = gifski\_renderer("yt2.gif"), width = 800, height = 800)

Gives this animation:  
[](https://datascienceplus.com/wp-content/uploads/2019/02/yt2.gif)

This is my Youtube usage where I primarily consume media content and this is also very much inline with my overall Phone usage which means it could be another potential driver. Possibly that my phone screen size increased, so I enjoy watching more videos 😑 which again isn’t something that I wanted it to be this way.

**Conclusion**

While I was under this constant belief that I’m one of those few Digital Minimalists, This analysis proves that I’m not so much of it and Yet I’ve got areas to work out in terms of reducing my phone usage and improve my lifestyle. Please note that this post is written in a cookbook-style than tutorial-style, this way you can be up and running with your Android activity analysis. If you have a doubt regarding the code (logic) please feel free to drop it in comments, I’d be happy to clarify them. Hope this post helps you in your Data-driven self-introspection – at least, the Android phone usage.

**References**

* If you are interested in learning about handling web data, Check out this [Datacamp Tutorial on Working with Web Data](https://www.datacamp.com/courses/foundations-of-functional-programming-with-purrr?tap_a=5644-dce66f&tap_s=210728-e54afe)
* The entire code-base (along with a few more sections and plots) is available on [my github](https://github.com/amrrs/data-driven-introspection)

Feel free to star/fork and use it!