What do CRAN download counts mean? [probably need a catchier title than this]

Yiwen Zhang

2021-06-09

Contents

Preface

[need to add more context here] This is written as part of the ETC5543 research project supervised by Drs Emi Tanaka and Hien Nguyen.

```
url <- "http://cran.rstudio.com/web/packages/packages.rds"
pkgs <- readRDS(url(url)) %>%
  as.data.frame() %>%
  rename(package = Package)
```

Chapter 1

Abstract

Data are closely related to social and economic activities in our daily life, so data mining and analysis will definitely benefit us. In many statistical and data analysis tools, we focus on R in this project, and aim to explore the evolution of R packages. By obtaining the daily and recent half year total download count of 17700 R packages (up to 2021-06-07) and R itself, we analyze their pattern characteristics, as well as the relationship between the release date, update times, number of commits on Github, name length and alphabetic order of the packages. Finally, we also check the changes of the top 15 downloaded packages on 1st April from 2013 to 2021, to have a glance of how the user preferences changes to some extent.

Chapter 2

Introduction

Data, which has penetrated into every industry and business field, has become an increasingly important production factor and consumption in our lives. However, data itself must be processed to uncover the information within it. For this reason many statistical tools were born, such as SPSS, Python, R and so on. In this project, we focus on the R language.

R is one of the most popular statistical languages and has been ranking competitively even among the general purpose programming languages (peaking 8th in **?**). The TIOBE Programming Community index aggregates several search engines to derive a metric of the popularity of a programming language. It is important to note, however, that programming languages ranked higher than R are mostly general purpose languages and therefore naturally has a larger user base whereas R is exclusively for data analysis and statistical computation alone.

In the process of using R, it often need the help of R packages to extend its functions. Due to that, R packages have become an indispensable part for R users. Although there are many sources to install R packages such as Bioconductor, Gitlab, GitHub or R-Forge, we only focus on CRAN (The Comprehensive R Archive Network), which is the official R packages repository. And there are 106 CRAN mirrors in 49 regions, but we focus only on the RStudio CRAN mirror, albeit the most popular. And as of 2021-06-09, there are 17,694 R packages on CRAN.

Based on that, researchers have started to study the popularity of R packages from CRAN. The most intuitive embodiment of popularity is the total number of downloads, although it does not directly reflect the number of users, because it includes the same user's repeated downloads, updates and test downloads caused by the server. But we still assume in this

project that, the download amount is a relatively reliable and simple measure of packages' popularity. Back to the previous studies, the R-package `adjustedcranlogs`(**?**) finds that there are spike in downloads due to automatic re-downloads and package update, and it also provides a method to remove these download logs. And package `packageRank`(**?**) provides a way to compute the rank percentile for packages and filter the invalid downloads including small and medium size logs. What's more, there are also some packages such as `pkgsearch`(**?**) and `Visualize.CRAN.Downloads`(**?**) providing some visualization methods to explore the package download trend or for convenient searching. Some of them are function extensions based on previous packages, and some of them propose their own new functions, but generally speaking, they are inclined to be more tool like to help users explore the characteristics of package download logs from as many dimensions as possible.

As for us, the purpose of this project is to explore the download logs of R packages on CRAN and analyze the relationship between some influential factors with it to help users and developers better understand the download amount pattern and also figure out what would determine a package's popularity to some extent. There are some terms involved in this project :

- R language : R is a language and environment for statistical computing and graphics which can be extended easily via packages and provide an Open Source route to participation in statistical methodology. It is available as Free Software under the term of the Free Software Foundation's GNU General Public License (**?**).
- R Studio : It is an integrated development environment (IDE) for R. It includes a console, syntax-highlighting editor that supports direct code execution, as well as tools for plotting, history, debugging and workspace management.(**?**)
- R Packages : They are collections of functions and data sets developed by the community. They increase the power of R by improving existing base R functionalities, or by adding new ones.
- CRAN : It is a network of ftp and web servers around the world that store identical, up-to-date, versions of code and documentation for R.(**?**)
- CRAN mirror : It is a website containing differently located servers, which aims to facilitate people from different regions and countries to access CRAN more smoothly and quickly. And each server is called a mirror.(**?**)
- Github API : It is the abbreviation of Application Programming Interface, which is a software intermediary that allows two applications to

talk to each other.

- CRAN task view : It aims to provide some guidance which packages on CRAN are relevant for tasks related to a certain topic (**?**).

Chapter 3

Methods

In this project, we extracted R packages names from CRAN and got summary daily download logs of R packages through web API maintained by r-hub(**?**). And we also collected release dates of R packages through CRAN. In addition, we also tried to scrap the number of commits in Github repository for R packages. However, due to the rate limit of Github API, we only captured part of the data in the end. Then we constructed eight exploratory data analysis sections:

- a) we explored the daily download of all the packages on CRAN and figured out the number of packages occupied by different downloaded groups;
- b) we analyzed the daily download trend of R itself and the found out the most popular version of R;
- c) we compared the top 15 downloaded packages on CRAN on 1st April. from year 2013 to 2021 to see how the user preference changes;
- d) we figured out the relationship between initial release date and the total download count for last half a year;
- e) we compared two closely related packages `fable` and `forecast` to see the difference between their MA (moving average);
- f) we studied how the number of commits of master (main) branch on Github repository influence the total download count for last half a year;
- g) we explored the relationship between the number of updates of packages and its download count;

- h) we analyzed the name length pattern for both CRAN task view packages and all the packages on CRAN;
- i) we explored how the alphabetical order of packages affected the download count and its statistic features (variance, median).

Chapter 4

Summary

In this project, we collected summary daily download logs of R packages through web API maintained by r-hub(**?**) and also used daily download data in CRAN for a time period from 2013-04-01 to 2021-04-01 to explore the daily download pattern of all packages both in general and of each year. In that case, we found that it is true that the cumulative number of downloads increases with time, and the variance also increases, which indicates that some packages with larger downloads grow rapidly. In addition, there is also strong weekly seasonality in the daily download plot. The download count will peak through weekdays and drop on weekend, for users may tend to work and study during weekdays while rest on weekends. What's more, through Lorenz curve, we also found that most of the cumulative downloads came from the top 10% downloaded packages, so we could also see that the distribution of downloads is quite unequal. Part of the reason is that these top 10% downloaded packages contain quite a lot popular and frequently used packages, such as `tidyverse` and `rlang`, which would be more probably to gain high downloads. In addition, there are other packages that often get high download volume, which can be divided into the following four categories:

- Packages maintained by R studio
- Packages created by authors from R core group
- Packages created by authors from R secondary group
- Packages created by R related authors
- Packages created by top 20 prolific maintainers (This is resourced at **?**)

However, the existence of these packages may make it difficult to reflect the popularity of other packages, we excluded these packages for the analysis of user preferences. And we found that the topic of newly added packages

on 1st Oct of each year come from quite different areas, while the packages remaining most stably popular during 2017 to 2019 is about JAVA dependency. Definitely, JAVA always ranked the top three among programming languages according to TIOBE Index(**?**), which shows that the number of users under JAVA related packages would be probably huge.

As for R itself, its download pattern is quite similar to that of total R packages on CRAN. And the most used OS for R users is windows OS. Also, the most popular version of R is 3.2.1.

After exploration of the characteristics of download pattern for R package and R itself, we then extracted the release dates of all packages and taskview packages from CRAN to compare the total download count of last half a year among packages with different release date or with different numbers of updates. And we found that for packages from the same topic, earlier release date usually would bring more download count, while packages with more times of updates would not always have higher downloads. So to sum up, packages released earlier and kept updated are more likely to have higher downloads.

In the next section, we initially tried to scrape the number of commits in Github repository of all packages on CRAN through Github API by R, to check whether more commits would result in more downloads or not. But there came a tricky problem on the rate limit of Github API. As documented in **?**, unauthenticated users could only be permitted to send 60 requests per hour. And only after get authentication, could the rate limit be expanded up to 5000 per hour. However, after trying several methods to get authentication, the rate limit was failed increase. So, we switched to make this done with python by setting random agent to avoid the API limit.Meanwhile, in order to display our initial research idea, we still had a look at the last 1% downloaded packages on this question with the original method, and that is also adapted as well. Therefore, we would expect that generally, if a package has more commits on Github repository, it would probably gain more download count.

The last two parts are about analysis for package name. We compared the average downloads among packages with different name length and different alphabetical orders. It is believed that over half of the packages tend to have shorter names probably for the sake of being easily remembered by users. And alphabetical order played little roles in promoting the download volume.

Chapter 5

Conclusion

In conclusion, we hold the belief that there are many factors that affect the download amount and popularity of the R packages on CRAN, such as the popularity of the creator, the application field of the package, the release date, whether to keep updated, and the length of the name. In addition, we also assume that the number of commits on Github repository will also probably affect the download amount of packages. However, due to technical limitations, the sample size we got is too small to make a strong conclusion on this point. We hope that in the future, we can get over this difficulty and add supplement. Anyway, we could generally believed that a relatively popular package should have earlier release date, shorter name and maybe more commits on Github repository if they have. There is also another point that could not be ignored is to keep updated.

Bibliography

Cran mirrors.

Latest news.

The most prolific package maintainers on cran.

Resources in the rest api.

(1991). The gnu operating system and the free software movement.

(2021). Open source and professional software for data science teams.

Csárdi, G. and Salmon, M. (2020). *pkgsearch: Search and Query CRAN R Packages*. R package version 3.0.3.

Morgan-Wall, T. (2017). *adjustedcranlogs: Remove Automated and Repeated Downloads from 'RStudio' 'CRAN' Download Logs*. R package version 0.2.0.

Peter Li (2021). *packageRank: Computation and Visualization of Package Download Counts and Percentiles*. R package version 0.4.2.

Ponce, M. (2021). *Visualize.CRAN.Downloads: Visualize Downloads from 'CRAN' Packages*. R package version 1.0.1.

R-Hub. r-hub/cranlogs.app.