Good morning everyone, I am Yiwen Zhang, my research project is about R and R packages on CRAN.

For most R users like us, maybe we would be quite care about

What would make an R package popular?

Before that, let me introduce some interesting things :

Have you ever known there are over 2 million R users? And in 2020, R ranks 8th among programming languages? Also, there are 17,686 R-packages on CRAN. And recently, the most trending R-packages are about web technologies.

So, how do we understand the popularity of R-packages? In this research, we assume the download count is a relatively reliable and simple reflection for this. So, we should first understand the CRAN download count and then explore the factors that may linked to this.

This is the data source. We get the download logs from CRAN mirror site and also summary date from cranlogs pkgs based on the web API maintained by r-hub. And commits from Github repo.

By looking at the daily download pattern for all the CRAN R-packages, we could observe that it rises with time with an increasing variance, actually, there is also a strong weekly seasonality, which is quite reasonable, for people would more likely to work during weekdays so the download would be higher, while when they rest on weekends, the download would jump down. Behind that, what do CRAN download count mean? When we have a closer look, we detected two unusual huge spikes in 2014 and 2018. In 2014, the spike comes from several highly-downloaded pkgs such as these two pkgs, while spike in 2018 is mostly caused by only tidyverse. And spike in 2014 comes from one IP while the one in 2018 comes from several consecutive Ips, it seems they are from same person or just server test issue. Not only that, the country of those download logs is also quite concentrated. 2014 is mainly from Indonesia and 2018 is from US.

Next, we found that only 10% R pkgs would share nearly 90% cumulative downloads. That is because these pkgs contain many frequently used pkgs like tidyverse. So the distribution of download count for R-pkgs is quite unequal.

Then let’s turn to the factors that would be linked to the download count. They are : XXX

But here I will only take two to explain.

About the initial release date, the object is CRAN task view pkgs. CRAN task view pkgs are pkgs classified according to their application areas. In this research we select three types of pkgs : Time series, Bayesian and Econometics. The reason for this is that the user base of different topics could be different, so comparing across different topics will be unfair. As a result, it is believed that earlier released pkgs of the same topic would gain more download count like displayed in this plot. All of the three types of pkgs’ download count increase with their release date, that is probably because earlier released pkgs are more likely to have a larger user base. For the choice of pkgs may be less for users at the very beginning.

And to further figure out whether earlier release would benefit the download volume, we compared two closely related pkgs : fable and forecast. It can be seen that forecast has a more stable trend than fable. But looking at the purple dashed line in fable, which marks the update of it. Soon after that day, fable get a significant increase while forecast drop down. So, that means earlier release may bring a higher download but not guarantee a faster increase.

The last part is about name analysis. Maybe we sometimes could notice that a later released movie would have a longer name distinguish itself to previous movies with similar themes. For example, the movie about Sherlock Holmes. The one born in 1995 add a suffix “the greatest detective” to distinguish itself from the one born in 1922. And we guess this is coincidence in R pkgs! Actually, we did find that for pkgs from same topic, the name length went up with the release date.

In conclusion, an earlier release, keeping update, shorter names and maybe more Github commits, better with a heated topic would make an R pkg popular!