A major new release of Homebrew has landed and now includes support for Linux as well as Windows! via the Windows Subsystem for Linux. There are overall stability and speed improvements baked in as well. The aforelinked notification has all the info you need to see the minutiae. Unless you’ve been super-lax in updating, brew update will get you the latest release.

There are extra formulae analytics endpoints and the homebrewanalytics R package has been updated to handle them. A change worth noting in the package is that all the API calls are memoised to avoid hammering the Homebrew servers (though the “API” is really just file endpoints and they aren’t big files but bandwidth is bandwidth). Use the facilities in the memoise package to invalidate the cache if you have long running scripts.

Use your favorite social coding site to install it (If I don’t maintain mirrors on your open social coding platform of choice just drop a note in the comments and I’ll start mirroring there as well):

library(hrbrthemes)

library(homebrewanalytics)

library(tidyverse)

f <- brew\_formulae()

mutate(f, n\_dependencies = lengths(build\_dependencies)) %>%

count(n\_dependencies) %>%

mutate(n\_dependencies = factor(n\_dependencies)) %>%

ggplot() +

geom\_col(aes(n\_dependencies, n), fill = ft\_cols$slate, width = 0.65) +

scale\_y\_comma("# formulae") +

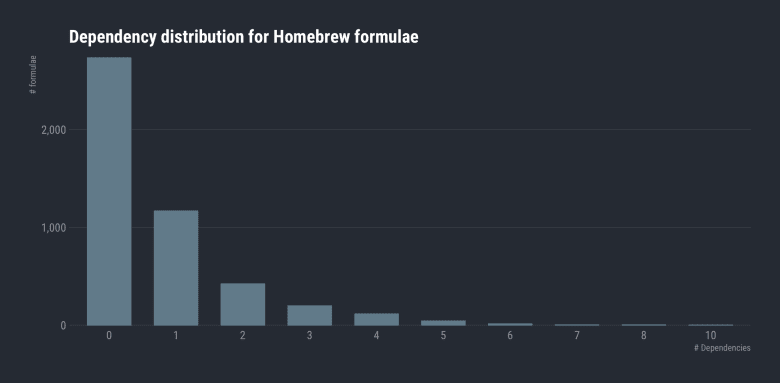
labs(

x = "# Dependencies",

title = "Dependency distribution for Homebrew formulae"

) +

theme\_ft\_rc(grid="Y")

[](https://rud.is/b/2019/02/02/homebrew-2-0-0-released-homebrewanalytics-package-updated/01-dep-dist/)

Given how long it sometimes takes to upgrade my various Homebrew installations I was surprised to see 0 be so prevalent, but one of the major changes in 2.0.0 is going to be more binary installs (unless you really need custom builds) so that is likely part of my experience, especially with the formulae I need to support cybersecurity and spatial operations.

We can also see which formuale are in the top 50%:

unlist(f$dependencies) %>%

table(dnn = "library") %>%

broom::tidy() %>%

arrange(desc(n)) %>%

mutate(pct = n/sum(n), cpct = cumsum(pct)) %>%

filter(cpct <= 0.5) %>%

mutate(pct = scales::percent(pct)) %>%

mutate(library = factor(library, levels = rev(library))) %>%

ggplot(aes(n, library)) +

geom\_segment(aes(xend=0, yend=library), color = ft\_cols$slate, size=3.5) +

geom\_text(

aes(x = (n+max(n)\*0.005), label = sprintf("%s (%s)", n, pct)),

hjust = 0, size = 3, family = font\_rc, color = ft\_cols$gray

) +

scale\_x\_comma(position = "top", limits=c(0, 500)) +

labs(

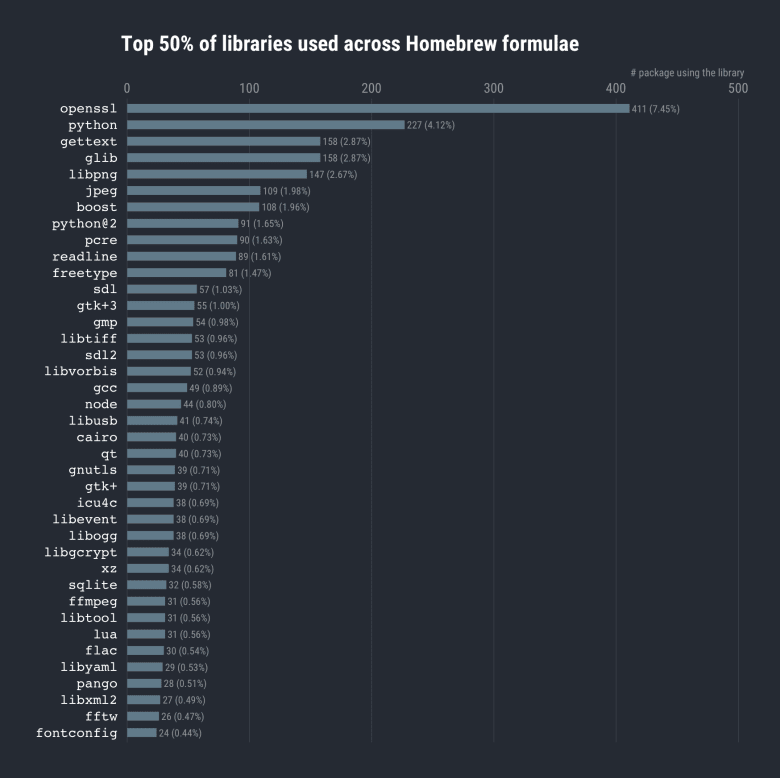
x = "# package using the library", y = NULL,

title = "Top 50% of libraries used across Homebrew formulae"

) +

theme\_ft\_rc(grid="X") +

theme(axis.text.y = element\_text(family = "mono"))

[](https://rud.is/b/2019/02/02/homebrew-2-0-0-released-homebrewanalytics-package-updated/02-50-libs-01/)

It seems openssl is pretty popular (not surprising but always good to see cybersecurity things at the top of good lists for a change)! macOS ships with an even more dreadful (*I know that’s hard to imagine*) default Python setup than usual so it being number 2 is not unexpected.

And, finally, we can also check on how frequently formulae are installed. Let’s look back on the last 90 days:

ggplot() +

geom\_density(

aes(x = installs$count, y = stat(count)),

color = ft\_cols$slate, fill = alpha(ft\_cols$slate, 1/2)

) +

scale\_x\_comma("# install events", trans = "log10") +

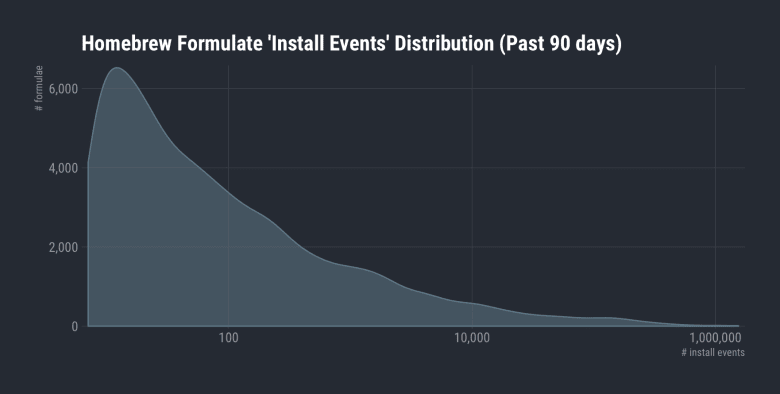
scale\_y\_comma("# formulae") +

labs(

title = "Homebrew Formulate 'Install Events' Distribution (Past 90 days)"

) +

theme\_ft\_rc(grid="XY")

[](https://rud.is/b/2019/02/02/homebrew-2-0-0-released-homebrewanalytics-package-updated/03-install-events/)

I’ll let you play with the package to find out who the heavy hitters are and explore more about the Homebrew ecosystem.