been using GNU+Linux distros for about 10 years now, and have settled for openSUSE as my main operating system around 3 years ago, perhaps even more. If you’re a gamer, you might have heard about SteamOS  
and how more and more games are available on GNU+Linux. I don’t really care about  
games, I play the occasional one (currently [Tangledeep](http://www.tangledeep.com/)) when I find  
the time, but still follow the news about gaming on GNU+Linux. Last week,  
Valve announced something quite big; it is now possible to run Windows games on GNU+Linux directly  
from Steam, using a modified version of [Wine](https://en.wikipedia.org/wiki/Wine_(software))  
they call Proton. The feature is still in Beta, and Valve announced that they guarantee around  
30 games to work already flawlessly. Of course, people have tried running a lot of other games, and,  
as was to be expected from Free Software and Open Source fans, GNU+Linux gamers created a Google Sheet  
that lists which games were tried and how they run. You can take a look at the sheet [here](https://docs.google.com/spreadsheets/d/1DcZZQ4HL_Ol969UbXJmFG8TzOHNnHoj8Q1f8DIFe8-8/htmlview?sle=true&pru=AAABZbqTTkc*IvT11ShwA2kjoe_4lPefiQ#gid=1003113831).

In this blog post, I will play around with this sheet. This blog post lists some {tidyverse} tricks  
I find useful and use often. Perhaps these tricks will be useful to you too! Let’s start by loading  
the needed packages:

library(tidyverse)

library(magrittr)

library(readxl)

Since I’m lazy and don’t want to type the whole name of the file I’ll be using some little regex:

steam <- read\_excel(Sys.glob("Steam\*"), sheet = "Main", skip = 2)

glimpse(steam)

## Observations: 8,570

## Variables: 9

## $ SteamDB "LINK", "LINK", "LINK", "LINK", "LINK", "LINK", "LIN...

## $ Game "64", "1849", "1982", "1982", "am Weapon: Revival", ...

## $ Submitted "5 days ago", "12 days ago", "11 days ago", "11 days...

## $ Status "Garbage", "Platinum", "Gold", "Platinum", "Platinum...

## $ Notes "Crashes with a debug log", "Plays OK.", "Gamepad su...

## $ Distro "Arch (4.18.5)", "Manjaro XFCE", "Gentoo AMD64 (Kern...

## $ Driver "Nvidia 396.54 / Intel xf86-video-intel (1:2.99.917+...

## $ Specs "Intel Core i7-7700HQ / Nvidia GTX 1050 (Mobile)", "...

## $ Proton "3.7 Beta", "3.7-4 Beta", "3.7-4 Beta", "Default", "...

Let’s count how many unique games are in the data:

steam %>%

count(Game)

## # A tibble: 3,855 x 2

## Game n

##

## 1 .hack//G.U. Last Recode 2

## 2 $1 Ride 1

## 3 0rbitalis 1

## 4 10 Second Ninja 4

## 5 100% Orange Juice 17

## 6 1000 Amps 3

## 7 12 Labours of Hercules VII: Fleecing the Fleece (Platinum Edition) 1

## 8 16bit trader 1

## 9 1849 1

## 10 1953 - KGB Unleased 1

## # ... with 3,845 more rows

That’s quite a lot of games! However, not everyone of them is playable:

steam %>%

count(Status)

## # A tibble: 8 x 2

## Status n

##

## 1 Borked 205

## 2 bronze 1

## 3 Bronze 423

## 4 Garbage 2705

## 5 Gold 969

## 6 Platinum 2596

## 7 Primary 1

## 8 Silver 1670

Around 2500 have the status “Platinum”, but some games might have more than one status:

steam %>%

filter(Game == "100% Orange Juice") %>%

count(Status)

## # A tibble: 5 x 2

## Status n

##

## 1 Bronze 5

## 2 Garbage 3

## 3 Gold 2

## 4 Platinum 6

## 5 Silver 1

More games run like *Garbage* than *Platinum*. But perhaps we can dig a little deeper and see if  
we find some patterns.

Let’s take a look at the GNU+Linux distros:

steam %>%

count(Distro)

## # A tibble: 2,085 x 2

## Distro n

##

## 1 ? 2

## 2 "\"Arch Linux\" (64 bit)" 1

## 3 "\"Linux Mint 18.3 Sylvia 64bit" 1

## 4 "\"Manjaro Stable 64-bit (Kernel 4.14.66)" 1

## 5 "\"Solus\" (64 bit)" 2

## 6 (K)ubuntu 18.04 64-bit (Kernel 4.15.0) 2

## 7 (L)Ubuntu 18.04.1 LTS 1

## 8 18.04.1 1

## 9 18.04.1 LTS 2

## 10 18.04.1 LTS (64 Bit) 1

## # ... with 2,075 more rows

Ok the distro column is pretty messy. Let’s try to bring some order to it:

steam %<>%

mutate(distribution = as\_factor(case\_when(

grepl("buntu|lementary|antergos|steam|mint|18.|pop|neon", Distro, ignore.case = TRUE) ~ "Ubuntu",

grepl("arch|manjaro", Distro, ignore.case = TRUE) ~ "Arch Linux",

grepl("gentoo", Distro, ignore.case = TRUE) ~ "Gentoo",

grepl("fedora", Distro, ignore.case = TRUE) ~ "Fedora",

grepl("suse", Distro, ignore.case = TRUE) ~ "openSUSE",

grepl("debian|sid|stretch|lmde", Distro, ignore.case = TRUE) ~ "Debian",

grepl("solus", Distro, ignore.case = TRUE) ~ "Solus",

grepl("slackware", Distro, ignore.case = TRUE) ~ "Slackware",

grepl("void", Distro, ignore.case = TRUE) ~ "Void Linux",

TRUE ~ "Other"

)))

The %<>% operator is shorthand for a <- a %>% f(). It passes a to f() and assigns the  
result back to a. Anyways, let’s take a look at the distribution column:

steam %>%

count(distribution)

## # A tibble: 10 x 2

## distribution n

##

## 1 Ubuntu 6632

## 2 Arch Linux 805

## 3 Solus 175

## 4 Debian 359

## 5 Fedora 355

## 6 Gentoo 42

## 7 Void Linux 38

## 8 Other 76

## 9 openSUSE 66

## 10 Slackware 22

I will group distributions that have less than 100 occurrences into a single category  
(meaning I will keep the 5 more common values):

steam %<>%

mutate(distribution = fct\_lump(distribution, n = 5, other\_level = "Other"))

steam %>%

count(distribution)

## # A tibble: 6 x 2

## distribution n

##

## 1 Ubuntu 6632

## 2 Arch Linux 805

## 3 Solus 175

## 4 Debian 359

## 5 Fedora 355

## 6 Other 244

Let’s do the same for the CPUs:

steam %<>%

mutate(CPU = as\_factor(case\_when(

grepl("intel|i\\d|xeon|core2|\\d{4}k|q\\d{4}|pentium", Specs, ignore.case = TRUE) ~ "Intel",

grepl("ryzen|threadripper|tr|amd|fx|r\\d|\\d{4}x|phenom", Specs, ignore.case = TRUE) ~ "AMD",

TRUE ~ NA\_character\_

)))

steam %>%

count(CPU)

## # A tibble: 3 x 2

## CPU n

##

## 1 Intel 5768

## 2 AMD 2319

## 3 483

And the same for the GPUs:

steam %<>%

mutate(GPU = as\_factor(case\_when(

grepl("nvidia|geforce|3\\d{2}|nouveau|gtx|gt\\s?\\d{1,}|9\\d0|1060|1070|1080", Specs, ignore.case = TRUE) ~ "Nvidia",

grepl("amd|radeon|ati|rx|vega|r9", Specs, ignore.case = TRUE) ~ "AMD",

grepl("intel|igpu|integrated|hd\\d{4}|hd\\sgraphics", Specs, ignore.case = TRUE) ~ "Intel",

TRUE ~ NA\_character\_

)))

steam %>%

count(GPU)

## # A tibble: 4 x 2

## GPU n

##

## 1 Nvidia 6086

## 2 AMD 1374

## 3 Intel 413

## 4 697

I will also add a rank for the Status column:

steam %<>%

mutate(rank\_status = case\_when(

Status == "Platinum" ~ 5,

Status == "Gold" ~ 4,

Status == "Silver" ~ 3,

Status == "Bronze" ~ 2,

Status == "Garbage" ~ 1

))

Now, what are the top 5 most frequent combinations of Status, distribution, CPU and GPU?

steam %>%

filter(!is.na(CPU), !is.na(GPU)) %>%

count(Status, distribution, CPU, GPU) %>%

mutate(total = sum(n)) %>%

mutate(freq = n / total) %>%

top\_n(5)

## Selecting by freq

## # A tibble: 5 x 7

## Status distribution CPU GPU n total freq

##

## 1 Garbage Ubuntu Intel Nvidia 1025 7443 0.138

## 2 Gold Ubuntu Intel Nvidia 361 7443 0.0485

## 3 Platinum Ubuntu Intel Nvidia 1046 7443 0.141

## 4 Platinum Ubuntu AMD Nvidia 338 7443 0.0454

## 5 Silver Ubuntu Intel Nvidia 650 7443 0.0873

Unsurprisingly, Ubuntu, or distributions using Ubuntu as a base, are the most popular ones. Nvidia  
is the most popular GPU, Intel for CPUs and in most cases, this combo of hardware and distribution  
is associated with positive ratings (even though there are almost as many “Garbage” ratings than  
“Platinum” ratings).

Now let’s compute some dumb averages of Statuses by distribution, CPU and GPU. Since I’m going  
to run the same computation three times, I’ll write a function to do that.

compute\_avg <- function(dataset, var){

var <- enquo(var)

dataset %>%

select(rank\_status, (!!var)) %>%

group\_by((!!var)) %>%

mutate(wt = n()) %>%

summarise(average\_rating = weighted.mean(rank\_status, (!!var), wt, na.rm = TRUE))

}

Let’s see now if we can rank distribution by Steam play rating:

compute\_avg(steam, distribution)

## # A tibble: 6 x 2

## distribution average\_rating

##

## 1 Ubuntu 3.03

## 2 Arch Linux 3.05

## 3 Solus 3.03

## 4 Debian 3.01

## 5 Fedora 3.07

## 6 Other 3.16

How about for hardware?

compute\_avg(steam, GPU)

## # A tibble: 4 x 2

## GPU average\_rating

##

## 1 Nvidia 3.07

## 2 AMD 2.90

## 3 Intel 3.01

## 4 NA

compute\_avg(steam, CPU)

## # A tibble: 3 x 2

## CPU average\_rating

##

## 1 Intel 3.03

## 2 AMD 3.06

## 3 NA

To wrap this up, what are the games with the most ratings? Perhaps this can give us a hint about which  
games GNU+Linux users prefer:

steam %>%

count(Game) %>%

top\_n(10)

## Selecting by n

## # A tibble: 10 x 2

## Game n

##

## 1 Age of Empires II: HD Edition 43

## 2 Borderlands 39

## 3 DiRT 3 Complete Edition 32

## 4 DOOM 62

## 5 Fallout: New Vegas 45

## 6 Grim Dawn 34

## 7 No Man's Sky 40

## 8 Path of Exile 35

## 9 Quake Champions 32

## 10 The Elder Scrolls V: Skyrim 46