Grouping and Chaining with dplyr

#### group\_by函數的使用

library(dplyr) ## 導入dplyr套件

cran <- tbl\_df(mydf)

rm(mydf)

by\_package <- group\_by(cran, package)

by\_package

summarize(by\_package, mean(size))

pack\_sum <- summarize(by\_package,

count = n(),

unique = n\_distinct(ip\_id), ## distinct 等價於length(unique(x))

countries = n\_distinct(country),

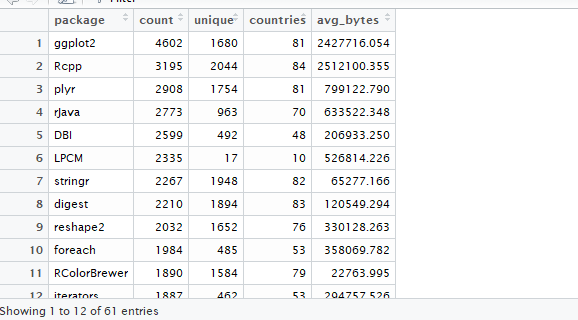
avg\_bytes = mean(size))

quantile(pack\_sum$count, probs = 0.99) ##0.99樣本分位數

top\_counts <- filter(pack\_sum, count > 679) ## 經常使用的包/下載的包 top1

top\_counts\_sorted <- arrange(top\_counts, desc(count))

View( top\_counts\_sorted)

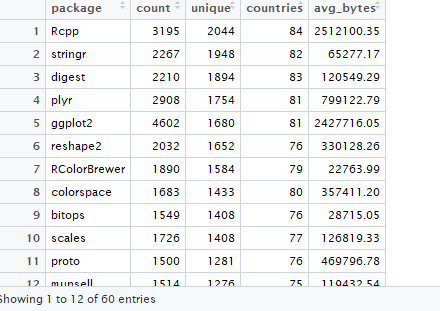


quantile(pack\_sum$unique, probs = 0.99) ## 查看unique的0.99分位數

top\_unique <- filter(pack\_sum,unique>465)

View(top\_unique)

top\_unique\_sorted <- arrange(top\_unique, desc(unique))



#### chaining

#### 沒有使用piping的程序

by\_package <- group\_by(cran, package)

pack\_sum <- summarize(by\_package,

count = n(),

unique = n\_distinct(ip\_id),

countries = n\_distinct(country),

avg\_bytes = mean(size))

# Here's the new bit, but using the same approach we've

# been using this whole time.

top\_countries <- filter(pack\_sum, countries > 60)

result1 <- arrange(top\_countries, desc(countries), avg\_bytes)

# Print the results to the console.

print(result1)

#### 使用piping的程序

result2 <-

arrange(

filter(

summarize(

group\_by(cran,

package

),

count = n(),

unique = n\_distinct(ip\_id),

countries = n\_distinct(country),

avg\_bytes = mean(size)

),

countries > 60

),

desc(countries),

avg\_bytes

)

print(result2)

####（建議）改進的程序

result3 <-

cran %>%

group\_by(package) %>%

summarize(count = n(),

unique = n\_distinct(ip\_id),

countries = n\_distinct(country),

avg\_bytes = mean(size)

) %>%

filter(countries > 60) %>%

arrange(desc(countries), avg\_bytes)

# Print result to console

print(result3)

cran %>%

select(ip\_id,country, package,size) %>%

print

cran %>%

select(ip\_id, country, package, size) %>%

mutate(size\_mb = size / 2^20)

cran %>%

select(ip\_id, country, package, size) %>%

mutate(size\_mb = size / 2^20) %>%

filter(size\_mb < 0.5)

cran %>%

select(ip\_id, country, package, size) %>%

mutate(size\_mb = size / 2^20) %>%

filter(size\_mb <= 0.5) %>%

arrange(desc(size\_mb))