Manipulating Data with dplyr

library(dplyr)

mydf <- read.csv(path2csv, stringAsFactor=FALSE)

cran <- tbl\_df(my\_df)

rm(“mydf”)

#### Select用法

select(cran, ip\_id, package, country) ## select表示選取特定的變量

select(cran, r\_arch:country)

select(cran, country:r\_arch) ## reverse

select(cran, -time) ## 刪除掉time這一列的數據，在原數據不刪除，只是select

-(5:20) ## 產生-5，-6，…，-20

select(cran, -(X:size)) ##刪除從X列到size列

#### filter用法

filter(cran, package == "swirl") ## 選擇package == “swirl”的全部數據

filter(cran, r\_version == "3.1.1", country == "US") ## 多個條件

filter(cran, r\_version <= "3.0.2", country == "IN") ##

filter(cran, country == "US" | country == "IN") ##這是或者的關係

filter(cran, size > 100500, r\_os == "linux-gnu")

#### is.na的用法

!is.na(c(3, 5, NA, 10))

is.na(c(3, 5, NA, 10))

#### 把他們組合一起

filter(cran, !is.na(r\_version))

#### arrange的用法

cran2 <- select(cran, size:ip\_id)

arrange(cran2, ip\_id) ## 將cran2按照ip\_id這一列進行排序，默認為升序asc

arrange(cran2,desc( ip\_id))

arrange(cran2, package, ip\_id) ##多個排序條件，先按照package排序，在按照ip\_id排序

arrange(cran2, country, desc(r\_version), ip\_id) ## order: country (ascending), r\_version (descending), and ip\_id (ascending).

#### mutate的用法：Add new variables

cran3 <- select(cran, ip\_id, package, size)

mutate(cran3, size\_mb = size / 2^20) ##給cran3添加新的列size\_mb

mutate(cran3, size\_mb = size / 2^20, size\_gb = size\_mb / 2^10) ## 添加兩個變量

#### summarize的用法

summarize(cran, avg\_bytes = mean(size))