Week 3

Martin Aragoneses

July 30, 2015

The `dplyr` package:

1: Manipulating Data with dplyr

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
##
## The following objects are masked from 'package:stats':
##
##     filter, lag
##
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# mydf <- read.csv(path2csv, stringsAsFactors = FALSE)
## load data into 'data frame tbl' or 'tbl_df'
# cran <- tbl_df(mydf)
# rm("mydf")
# ?tbl_df
#cran
## data manipulation tasks: select(), filter(), arrange(), mutate(), and summarize()

### select() ###    return only a subset of the columns/ specifies the columns you want to keep
# select(cran, ip_id, package, country)
# select(cran, r_arch:country) # left-to-right
# select(cran, country:r_arch) # right-to-left
## you can specify the columns we want to throw away
# select(cran, -time)
# select(cran, -(X:size))

### filter() ###    return only a subset of the rows/  specifies the rows you want to keep
# filter(cran, package == "swirl")
# filter(cran, r_version == "3.1.1", country == "US")   # BOTH CONDITIONS ARE TRUE (AND)
# filter(cran, r_version <= "3.0.2", country == "IN")
## EITHER ... OR ... (...|...)
# filter(cran, country == "US" | country == "IN")
# filter(cran, !is.na(r_version))    remove missing values (NA)

### arrange() ### sorts/reorders the rows according to the values of a particular variable
# arrange(cran2, ip_id)
# arrange(cran2, desc(ip_id))
```

```
# arrange(cran2, package, ip_id)
# arrange(cran2, country, desc(r_version), ip_id)

### mutate() ###  create a new variable based on the value of one or more variables already in a datase
# mutate(cran3, size_mb = size / 2^20) #MB
# mutate(cran3, size_mb = size / 2^20, size_gb = size_mb/2^10) #GB
# mutate(cran3, correct_size = size + 1000)

### summarize()
# summarize(cran, avg_bytes = mean(size))
```

2: Grouping and Chaining with dplyr

```
### group_by(): ###  Grouping data & summarize()
## group the data by package name
## any operation we apply to the grouped data will take place on a per package basis
# by_package <- group_by(cran, package)
# summarize(by_package, mean(size))  ## get a distinct average by group
#pack_sum <- summarize(by_package,
#                      count = n(),
#                      unique = n_distinct(ip_id),
#                      countries = n_distinct(country),
#                      avg_bytes = mean(size))
# quantile(pack_sum$count, probs = 0.99)
# top_counts <- filter(pack_sum, count > 679)
# View(top_counts)
# top_counts_sorted <- arrange(top_counts, desc(count))
# top_countries <- filter(pack_sum, countries > 60)
# result1 <- arrange(top_countries, desc(countries), avg_bytes)

## "Chaining" or "Piping"
# 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)

## The code on the right of %>% operates on the result obtained on the left

# cran %>%
#   select(ip_id, country, package, size) %>%
#   print

# cran %>%
#   select(ip_id, country, package, size) %>%
#   mutate(size_mb = size / 2^20) %>%
```

```
#   filter(size_mb <= 0.5) %>%
#   arrange(desc(size_mb)) %>%
#   print
```