Pipes

MSDS 597 Data Wrangling & Husbandry

February 03, 2020

The origin of pipes

Pipes are commonly used in Unix/Linux programming. The character “|” is used to pass the output of one function to another, such as

```
gunzip example.txt.gz | cut -f1,3 | head
```

That would uncompress a file, pick out the first and third column, and then look at the first few rows.

Pipes in R are implemented using the maggritr package

Much of the `magrittr` package is loaded as part of the `tidyverse` package. Instead of the unix notation, use `%>%`. The output is of the call on the left is put into the first argument of the function on the right.

To quote Hadley Wickham,

`x %>% f(y)` turns into `f(x, y)`, and `x %>% f(y) %>% g(z)` turns into `g(f(x, y), z)` and so on

Here's an example from R for Data Science

```
delays <- flights %>%
  group_by(dest) %>%
  summarise(
    count = n(),
    dist = mean(distance, na.rm = TRUE),
    delay = mean(arr_delay, na.rm = TRUE)
  ) %>%
  filter(count > 20, dest != "HNL")

delays
```

```
## # A tibble: 96 x 4
##    dest  count  dist delay
##    <chr> <int> <dbl> <dbl>
##  1 ABQ     254 1826   4.38
##  2 ACK     265  199   4.85
##  3 ALB     439  143  14.4
##  4 ATL   17215  757. 11.3
##  5 AUS    2439 1514   6.02
```

Without pipes, you might have chosen to use intermediate steps such as this

```r
temp1 <- group_by(flights, dest)
temp2 <- summarise(temp1,   count = n(),
    dist = mean(distance, na.rm = TRUE),
    delay = mean(arr_delay, na.rm = TRUE)
  )
delays <- filter(count > 20, dest != "HNL")
```

or even

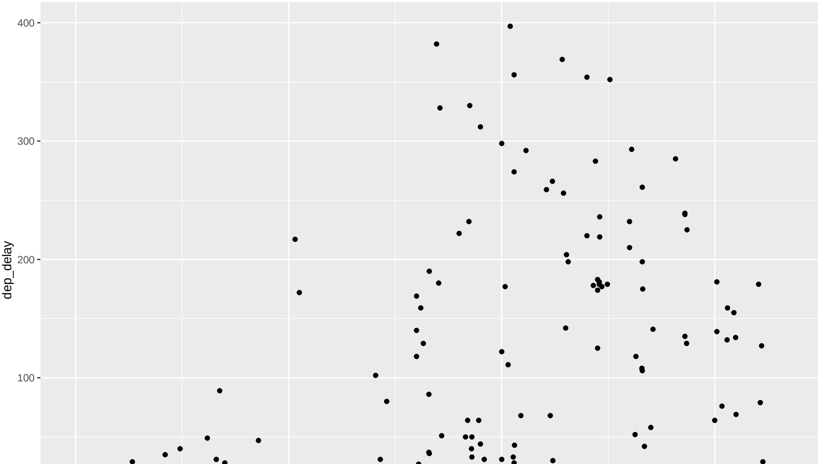
```
delays <- filter(summarise(group_by(flights, dest),   count
    dist = mean(distance, na.rm = TRUE),
    delay = mean(arr_delay, na.rm = TRUE)
  ), count > 20, dest != "HNL")
```

The most common style is to use a single function per line when using pipes. Put the %>% operator at the end of lines, not the beginning, or R will think you're finished with your expression earlier than you intended.

```r
delays <- flights %>%
  group_by(dest) %>%
  summarise(
    count = n(),
    dist = mean(distance, na.rm = TRUE),
    delay = mean(arr_delay, na.rm = TRUE)
  ) %>%
  filter(count > 20, dest != "HNL")
```

For ggplot we still use the + notation, but since data is the first argument of ggplot we can pipe into it at least

```r
flights %>%
  filter(origin == "EWR" & month == 9 & day == 12) %>%
  ggplot(mapping = aes(sched_dep_time, dep_delay)) +
  geom_point()
```



In class exercise

- Use pipes to start with the babynames data frame and ultimately list the most popular names, by sex, totalled over all years
- Now list the most popular names, by sex, totalled over all years since 2000

There is one feature of the `magrittr` package that I find useful that is not loaded by `tidyverse`, the %$% operator.

```
library(magrittr)

flights %$% cor(distance, arr_delay)
```

```
## [1] NA
```

```
flights %$%
  cor(distance, arr_delay, use = "pairwise.complete.obs") %
  round(digits = 2)
```

```
## [1] -0.06
```