WK5 925

Zichun Liu

9/25/2017

Why dplyr

1. Easier to read and write
2. More efficient

```
# compare efficiency
microbenchmark(
  filter(flights, month == 1, day == 1),
  subset(flights, subset = month == 1 & day == 1)
)
```

```
## Unit: milliseconds
##                                          expr       min        lq
##            filter(flights, month == 1, day == 1)  4.941193  7.797154
##  subset(flights, subset = month == 1 & day == 1) 10.477060 15.873613
##       mean   median        uq      max neval cld
##   8.896212  8.56473  9.741162 16.92796   100  a
##  19.091702 18.79468 21.096396 65.77061   100   b
```

%>% Operator

Although not required, the dplyr packages make use of the pipe operator `%>%` developed by Stefan Milton Bache in the R package `magrittr`. Although all the functions in dplyr can be used without the pipe operator, one of the great conveniences these packages provide is the ability to string multiple functions together by incorporating `%>%`.

This operator will forward a value, or the result of an expression, into the next function call/expression. For instance a function to filter data can be written as:

```
filter(data, variable == numeric_value)

data %>% filter(variable == numeric_value)
```

Both functions complete the same task and the benefit of using `%>%` is not evident; however, when you desire to perform multiple functions its advantage becomes obvious. For instance, if we want to filter some data, summarize it, and then order the summarized results we would write it out as:

```
#Nested Option:
arrange(summarize(filter(data, variable == numeric_value),Total = sum(variable)),desc(Total))

#or
#Multiple Object Option:
a = filter(data, variable == numeric_value)
b = summarise(a, Total = sum(variable))
c = arrange(b, desc(Total))

#or
#%>% Option:
data %>%
```

```
  filter(variable == "value") %>%
  summarise(Total = sum(variable)) %>%
  arrange(desc(Total))
```

As your function tasks get longer the %>% operator becomes more efficient and makes your code more legible. In addition, although not covered in this tutorial, the %>% operator allows you to flow from data manipulation tasks straight into vizualization functions (via ggplot and ggvis) and also into many analytic functions.

dplyr functions

```
select() # select columns # Reduce dataframe size to only desired variables for current task

filter() # select rows # Reduce rows/observations with matching conditions

group_by() # Group data by categorical variables

summarise() #Perform summary statistics on variables

arrange() # Order variable values

XXX_join() # Join two datasets together

mutate() # Creates new variables
```

```
#select
flights %>%
  select(month, day, carrier, distance)
```

```
## # A tibble: 336,776 × 4
##    month   day carrier distance
##    <int> <int>   <chr>    <dbl>
## 1      1     1      UA     1400
## 2      1     1      UA     1416
## 3      1     1      AA     1089
## 4      1     1      B6     1576
## 5      1     1      DL      762
## 6      1     1      UA      719
## 7      1     1      B6     1065
## 8      1     1      EV      229
## 9      1     1      B6      944
## 10     1     1      AA      733
## # ... with 336,766 more rows
```

```
# filter
flights %>%
  select(month, day, carrier, distance) %>%
  filter(month == 1, day == 1)
```

```
## # A tibble: 842 × 4
##    month   day carrier distance
##    <int> <int>   <chr>    <dbl>
## 1      1     1      UA     1400
## 2      1     1      UA     1416
## 3      1     1      AA     1089
```

```
## 4      1     1      B6     1576
## 5      1     1      DL      762
## 6      1     1      UA      719
## 7      1     1      B6     1065
## 8      1     1      EV      229
## 9      1     1      B6      944
## 10     1     1      AA      733
## # ... with 832 more rows
```

```
# summarise and group_by
flights %>%
  select(month, day, carrier, distance) %>%
  filter(month == 1, day == 1) %>%
  group_by(carrier) %>%
  summarise(meandistance = mean(distance))
```

```
## # A tibble: 14 × 2
##    carrier meandistance
##      <chr>        <dbl>
## 1       9E     520.3571
## 2       AA    1337.7128
## 3       AS    2402.0000
## 4       B6    1106.2025
## 5       DL    1222.0357
## 6       EV     491.4569
## 7       F9    1620.0000
## 8       FL     686.6000
## 9       HA    4983.0000
## 10      MQ     577.0000
## 11      UA    1496.4909
## 12      US     833.1562
## 13      VX    2502.3333
## 14      WN     895.7037
```

```
#arrange
flights %>%
  select(month, day, carrier, distance) %>%
  filter(month == 1, day == 1) %>%
  group_by(carrier) %>%
  summarise(meandistance = mean(distance)) %>%
  arrange(desc(meandistance))
```

```
## # A tibble: 14 × 2
##    carrier meandistance
##      <chr>        <dbl>
## 1       HA    4983.0000
## 2       VX    2502.3333
## 3       AS    2402.0000
## 4       F9    1620.0000
## 5       UA    1496.4909
## 6       AA    1337.7128
## 7       DL    1222.0357
## 8       B6    1106.2025
## 9       WN     895.7037
## 10      US     833.1562
## 11      FL     686.6000
```

```
## 12      MQ     577.0000
## 13      9E     520.3571
## 14      EV     491.4569
```

```
# mutate()
flights %>%
  select(month, day, carrier, distance) %>%
  filter(month == 1, day == 1) %>%
  group_by(carrier) %>%
  summarise(meandistance = mean(distance), N = n()) %>%
  arrange(desc(meandistance)) %>%
  mutate(totaldistance = meandistance * N)
```

```
## # A tibble: 14 × 4
##    carrier meandistance     N totaldistance
##      <chr>        <dbl> <int>         <dbl>
## 1       HA    4983.0000     1          4983
## 2       VX    2502.3333    12         30028
## 3       AS    2402.0000     2          4804
## 4       F9    1620.0000     2          3240
## 5       UA    1496.4909   165        246921
## 6       AA    1337.7128    94        125745
## 7       DL    1222.0357   112        136868
## 8       B6    1106.2025   163        180311
## 9       WN     895.7037    27         24184
## 10      US     833.1562    32         26661
## 11      FL     686.6000    10          6866
## 12      MQ     577.0000    78         45006
## 13      9E     520.3571    28         14570
## 14      EV     491.4569   116         57009
```

```
(flight_1 = flights %>%
  select(month, day, carrier, distance) %>%
  filter(month == 1, day == 1) %>%
  group_by(carrier) %>%
  summarise(meandistance = mean(distance), N = n()) %>%
  arrange(desc(meandistance)) %>%
  mutate(totaldistance = meandistance * N))
```

```
## # A tibble: 14 × 4
##    carrier meandistance     N totaldistance
##      <chr>        <dbl> <int>         <dbl>
## 1       HA    4983.0000     1          4983
## 2       VX    2502.3333    12         30028
## 3       AS    2402.0000     2          4804
## 4       F9    1620.0000     2          3240
## 5       UA    1496.4909   165        246921
## 6       AA    1337.7128    94        125745
## 7       DL    1222.0357   112        136868
## 8       B6    1106.2025   163        180311
## 9       WN     895.7037    27         24184
## 10      US     833.1562    32         26661
## 11      FL     686.6000    10          6866
## 12      MQ     577.0000    78         45006
## 13      9E     520.3571    28         14570
## 14      EV     491.4569   116         57009
```

```r
(flight_2 = flights %>%
  select(month, day, carrier, air_time) %>%
  filter(month == 1, day == 1) %>%
  group_by(carrier) %>%
  summarise(meantime = mean(air_time), N1 = n()) %>%
  arrange(meantime) %>% na.omit())
```

```
## # A tibble: 8 × 3
##   carrier meantime    N1
##     <chr>    <dbl> <int>
## 1      FL 120.5000    10
## 2      US 139.6250    32
## 3      WN 162.8148    27
## 4      DL 184.7768   112
## 5      F9 249.5000     2
## 6      AS 337.0000     2
## 7      VX 353.6667    12
## 8      HA 659.0000     1
```

```r
inner_join(flight_1, flight_2, by = "carrier")
```

```
## # A tibble: 8 × 6
##   carrier meandistance     N totaldistance meantime    N1
##     <chr>        <dbl> <int>         <dbl>    <dbl> <int>
## 1      HA    4983.0000     1          4983 659.0000     1
## 2      VX    2502.3333    12         30028 353.6667    12
## 3      AS    2402.0000     2          4804 337.0000     2
## 4      F9    1620.0000     2          3240 249.5000     2
## 5      DL    1222.0357   112        136868 184.7768   112
## 6      WN     895.7037    27         24184 162.8148    27
## 7      US     833.1562    32         26661 139.6250    32
## 8      FL     686.6000    10          6866 120.5000    10
```

```r
left_join(flight_1, flight_2, by = "carrier")
```

```
## # A tibble: 14 × 6
##    carrier meandistance     N totaldistance meantime    N1
##      <chr>        <dbl> <int>         <dbl>    <dbl> <int>
## 1       HA    4983.0000     1          4983 659.0000     1
## 2       VX    2502.3333    12         30028 353.6667    12
## 3       AS    2402.0000     2          4804 337.0000     2
## 4       F9    1620.0000     2          3240 249.5000     2
## 5       UA    1496.4909   165        246921       NA    NA
## 6       AA    1337.7128    94        125745       NA    NA
## 7       DL    1222.0357   112        136868 184.7768   112
## 8       B6    1106.2025   163        180311       NA    NA
## 9       WN     895.7037    27         24184 162.8148    27
## 10      US     833.1562    32         26661 139.6250    32
## 11      FL     686.6000    10          6866 120.5000    10
## 12      MQ     577.0000    78         45006       NA    NA
## 13      9E     520.3571    28         14570       NA    NA
## 14      EV     491.4569   116         57009       NA    NA
```

```r
full_join(flight_1, flight_2, by = "carrier")
```

```
## # A tibble: 14 × 6
```

```
##    carrier meandistance     N totaldistance meantime    N1
##      <chr>        <dbl> <int>         <dbl>    <dbl> <int>
## 1       HA    4983.0000     1          4983 659.0000     1
## 2       VX    2502.3333    12         30028 353.6667    12
## 3       AS    2402.0000     2          4804 337.0000     2
## 4       F9    1620.0000     2          3240 249.5000     2
## 5       UA    1496.4909   165        246921       NA    NA
## 6       AA    1337.7128    94        125745       NA    NA
## 7       DL    1222.0357   112        136868 184.7768   112
## 8       B6    1106.2025   163        180311       NA    NA
## 9       WN     895.7037    27         24184 162.8148    27
## 10      US     833.1562    32         26661 139.6250    32
## 11      FL     686.6000    10          6866 120.5000    10
## 12      MQ     577.0000    78         45006       NA    NA
## 13      9E     520.3571    28         14570       NA    NA
## 14      EV     491.4569   116         57009       NA    NA
```

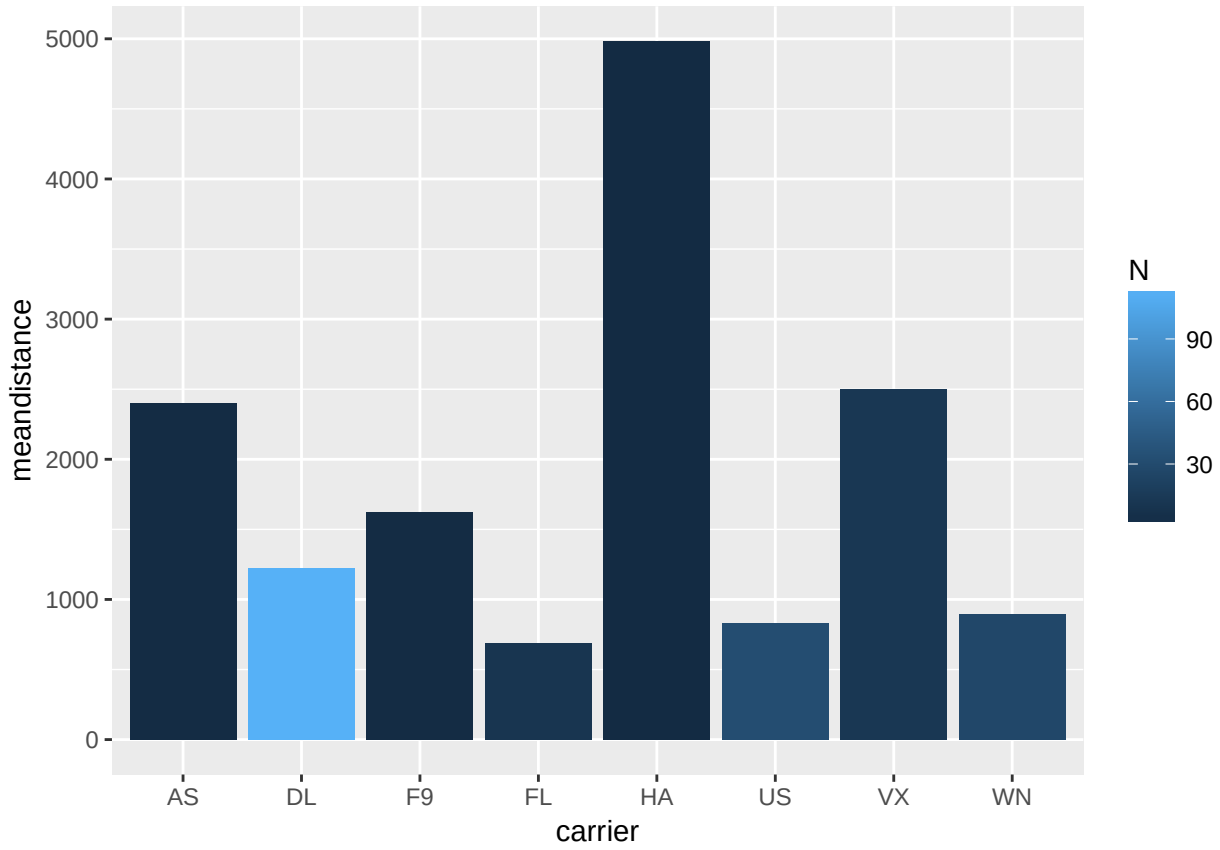
```
semi_join(flight_1, flight_2, by = "carrier")
```

```
## # A tibble: 8 × 4
##   carrier meandistance     N totaldistance
##     <chr>        <dbl> <int>         <dbl>
## 1      FL     686.6000    10          6866
## 2      US     833.1562    32         26661
## 3      WN     895.7037    27         24184
## 4      DL    1222.0357   112        136868
## 5      F9    1620.0000     2          3240
## 6      AS    2402.0000     2          4804
## 7      VX    2502.3333    12         30028
## 8      HA    4983.0000     1          4983
```

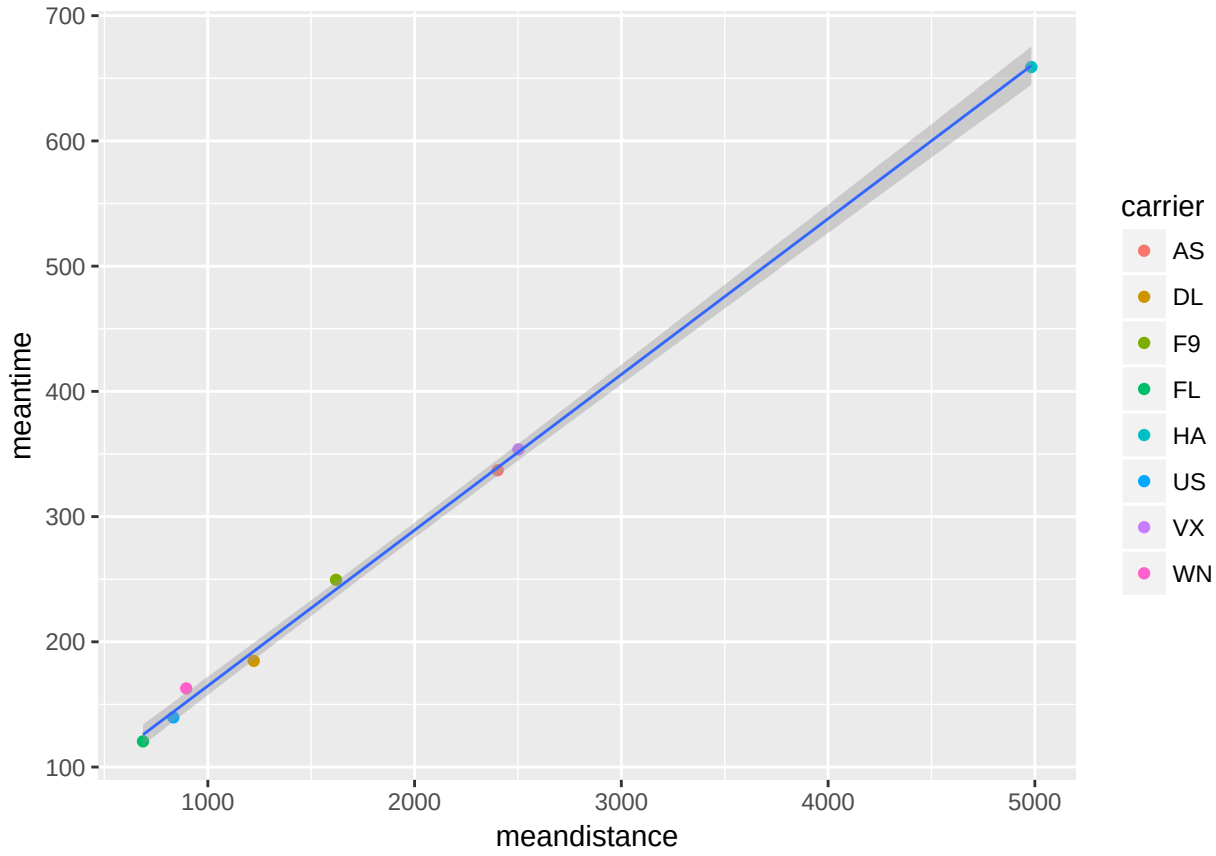
```
anti_join(flight_1, flight_2, by = "carrier")
```

```
## # A tibble: 6 × 4
##   carrier meandistance     N totaldistance
##     <chr>        <dbl> <int>         <dbl>
## 1      EV     491.4569   116         57009
## 2      9E     520.3571    28         14570
## 3      MQ     577.0000    78         45006
## 4      B6    1106.2025   163        180311
## 5      AA    1337.7128    94        125745
## 6      UA    1496.4909   165        246921
```

```
library(ggplot2)
inner_join(flight_1, flight_2, by = "carrier") %>%
  ggplot() +
  geom_bar(aes(x = carrier, y = meandistance, fill = N), stat = "identity")
```




```
inner_join(flight_1, flight_2, by = "carrier") %>%
  ggplot() +
  geom_point(aes(x = meandistance, y = meantime, col = carrier)) +
  geom_smooth(aes(x = meandistance, y = meantime), method = "lm", lwd = 0.5)
```


700
600
500
400
300
200
100
meantime
1000
2000
3000
4000
5000
meandistance
carrier
AS
DL
F9
FL
HA
US
VX
WN

End