Short introduction to dplyr

Ekaterina Edelstein

13 Juli 2017

What is dplyr and why is it useful?

- R-Package written by Hadley Wickham
- focussed on working with data frames
- It provides “verbs” that corresponds to the tasks for data manipulations such as filtering for rows, selecting columns, re-ordering rows, adding new columns and summarizing data
- in comparison to base functions in R (such as apply(), lapply(), sapply()) functions in dplyr are easier to work with: cleaner and simpler code
- is faster than some traditional functions
- It uses efficient backends, so you spend less time waiting for the computer.

```r
require(dplyr)
require(nycflights13)
```

```
flights
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            600        -6      812
##  6  2013     1     1      554            558        -4      740
##  7  2013     1     1      555            600        -5      913
##  8  2013     1     1      557            600        -3      709
##  9  2013     1     1      557            600        -3      838
## 10  2013     1     1      558            600        -2      753
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

Note that `flights` is a tibble, a modern reimagining of the data frame, usefull for large datasets.

```r
str(flights)
```

```
## Classes 'tbl_df', 'tbl' and 'data.frame':    336776 obs. of  19 variables:
##  $ year          : int  2013 2013 2013 2013 2013 2013 2013 2013 2013 2013 ...
##  $ month         : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ day           : int  1 1 1 1 1 1 1 1 1 1 ...
##  $ dep_time      : int  517 533 542 544 554 554 555 557 557 558 ...
##  $ sched_dep_time: int  515 529 540 545 600 558 600 600 600 600 ...
##  $ dep_delay     : num  2 4 2 -1 -6 -4 -5 -3 -3 -2 ...
##  $ arr_time      : int  830 850 923 1004 812 740 913 709 838 753 ...
##  $ sched_arr_time: int  819 830 850 1022 837 728 854 723 846 745 ...
##  $ arr_delay     : num  11 20 33 -18 -25 12 19 -14 -8 8 ...
##  $ carrier       : chr  "UA" "UA" "AA" "B6" ...
##  $ flight        : int  1545 1714 1141 725 461 1696 507 5708 79 301 ...
##  $ tailnum       : chr  "N14228" "N24211" "N619AA" "N804JB" ...
##  $ origin        : chr  "EWR" "LGA" "JFK" "JFK" ...
##  $ dest          : chr  "IAH" "IAH" "MIA" "BQN" ...
##  $ air_time      : num  227 227 160 183 116 150 158 53 140 138 ...
##  $ distance      : num  1400 1416 1089 1576 762 ...
##  $ hour          : num  5 5 5 5 6 5 6 6 6 6 ...
##  $ minute        : num  15 29 40 45 0 58 0 0 0 0 ...
##  $ time_hour     : POSIXct, format: "2013-01-01 05:00:00" "2013-01-01 05:00:00" ...
```

Verbs for data manipulation

- `filter()` for selecting rows with filter criteria
- `arrange()` for re-ordering rows
- `select()` for selecting columns based on their name
- `mutate()` for defining a new columns, that are functions of existing columns
- `summarise()` for summarising values (e.g. groups)
- `group_by()` allows group operations
- `sample_n()` and `sample_frac()` for taking random samples

filter()

```
flights %>%
  filter(dep_delay == 0, month == 1, day == 1)
```

```
## # A tibble: 59 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      559            559         0      702
##  2  2013     1     1      600            600         0      851
##  3  2013     1     1      600            600         0      837
##  4  2013     1     1      607            607         0      858
##  5  2013     1     1      615            615         0     1039
##  6  2013     1     1      615            615         0      833
##  7  2013     1     1      635            635         0     1028
##  8  2013     1     1      655            655         0     1021
##  9  2013     1     1      739            739         0     1104
## 10  2013     1     1      745            745         0     1135
## # ... with 49 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

arrange()

```r
flights %>%
  arrange(dep_delay)
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013    12     7     2040           2123       -43       40
##  2  2013     2     3     2022           2055       -33     2240
##  3  2013    11    10     1408           1440       -32     1549
##  4  2013     1    11     1900           1930       -30     2233
##  5  2013     1    29     1703           1730       -27     1947
##  6  2013     8     9      729            755       -26     1002
##  7  2013    10    23     1907           1932       -25     2143
##  8  2013     3    30     2030           2055       -25     2213
##  9  2013     3     2     1431           1455       -24     1601
## 10  2013     5     5      934            958       -24     1225
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```
flights %>%
  arrange(desc(dep_delay))
```

```
## # A tibble: 336,776 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     9      641            900      1301     1242
##  2  2013     6    15     1432           1935      1137     1607
##  3  2013     1    10     1121           1635      1126     1239
##  4  2013     9    20     1139           1845      1014     1457
##  5  2013     7    22      845           1600      1005     1044
##  6  2013     4    10     1100           1900       960     1342
##  7  2013     3    17     2321            810       911      135
##  8  2013     6    27      959           1900       899     1236
##  9  2013     7    22     2257            759       898      121
## 10  2013    12     5      756           1700       896     1058
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

select()

```
flights %>%
  select(year, month, day)
```

```
## # A tibble: 336,776 x 3
##     year month   day
##    <int> <int> <int>
##  1  2013     1     1
##  2  2013     1     1
##  3  2013     1     1
##  4  2013     1     1
##  5  2013     1     1
##  6  2013     1     1
##  7  2013     1     1
##  8  2013     1     1
##  9  2013     1     1
## 10  2013     1     1
## # ... with 336,766 more rows
```

```
flights %>%
  select(year:day)
```

```
## # A tibble: 336,776 x 3
##     year month   day
##    <int> <int> <int>
##  1  2013     1     1
##  2  2013     1     1
##  3  2013     1     1
##  4  2013     1     1
##  5  2013     1     1
##  6  2013     1     1
##  7  2013     1     1
##  8  2013     1     1
##  9  2013     1     1
## 10  2013     1     1
## # ... with 336,766 more rows
```

```
flights %>%
  select(-(year:day))
```

```
## # A tibble: 336,776 x 16
##    dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
##       <int>          <int>     <dbl>    <int>          <int>     <dbl>
##  1      517            515         2      830            819        11
##  2      533            529         4      850            830        20
##  3      542            540         2      923            850        33
##  4      544            545        -1     1004           1022       -18
##  5      554            600        -6      812            837       -25
##  6      554            558        -4      740            728        12
##  7      555            600        -5      913            854        19
##  8      557            600        -3      709            723       -14
##  9      557            600        -3      838            846        -8
## 10      558            600        -2      753            745         8
## # ... with 336,766 more rows, and 10 more variables: carrier <chr>,
## #   flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>,
## #   distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

```r
flights %>%
  select(-(year:day), 6:8)
```

```
## # A tibble: 336,776 x 16
##    dep_time sched_dep_time dep_delay arr_time sched_arr_time arr_delay
##       <int>          <int>     <dbl>    <int>          <int>     <dbl>
##  1      517            515         2      830            819        11
##  2      533            529         4      850            830        20
##  3      542            540         2      923            850        33
##  4      544            545        -1     1004           1022       -18
##  5      554            600        -6      812            837       -25
##  6      554            558        -4      740            728        12
##  7      555            600        -5      913            854        19
##  8      557            600        -3      709            723       -14
##  9      557            600        -3      838            846        -8
## 10      558            600        -2      753            745         8
## # ... with 336,766 more rows, and 10 more variables: carrier <chr>,
## #   flight <int>, tailnum <chr>, origin <chr>, dest <chr>, air_time <dbl>,
## #   distance <dbl>, hour <dbl>, minute <dbl>, time_hour <dttm>
```

```r
flights %>%
  select(6:8, -(year:day))
```

```
## # A tibble: 336,776 x 3
##    dep_delay arr_time sched_arr_time
##        <dbl>    <int>          <int>
##  1         2      830            819
##  2         4      850            830
##  3         2      923            850
##  4        -1     1004           1022
##  5        -6      812            837
##  6        -4      740            728
##  7        -5      913            854
##  8        -3      709            723
##  9        -3      838            846
## 10        -2      753            745
## # ... with 336,766
```

```r
flights %>%
  select(flight, tailnum, contains("dep"))
```

```
## # A tibble: 336,776 x 5
##    flight tailnum dep_time sched_dep_time dep_delay
##     <int>   <chr>    <int>          <int>     <dbl>
##  1   1545  N14228      517            515         2
##  2   1714  N24211      533            529         4
##  3   1141  N619AA      542            540         2
##  4    725  N804JB      544            545        -1
##  5    461  N668DN      554            600        -6
##  6   1696  N39463      554            558        -4
##  7    507  N516JB      555            600        -5
##  8   5708  N829AS      557            600        -3
##  9     79  N593JB      557            600        -3
## 10    301  N3ALAA      558            600        -2
## # ... with 336,766 more rows
```

Select helpers: `c()`, `starts_with()`, `ends_with()`, `matches()`, `one_of()` etc.

mutate()

```r
flights %>%
  mutate(speedmph = distance/air_time * 60) %>%
  select(flight, origin, dest, distance, air_time, speedmph) %>%
  arrange(desc(speedmph))
```

```
## # A tibble: 336,776 x 6
##    flight origin  dest distance air_time speedmph
##     <int>  <chr> <chr>    <dbl>    <dbl>    <dbl>
##  1   1499    LGA   ATL      762       65 703.3846
##  2   4667    EWR   MSP     1008       93 650.3226
##  3   4292    EWR   GSP      594       55 648.0000
##  4   3805    EWR   BNA      748       70 641.1429
##  5   1902    LGA   PBI     1035      105 591.4286
##  6    315    JFK   SJU     1598      170 564.0000
##  7    707    JFK   SJU     1598      172 557.4419
##  8    936    JFK   STT     1623      175 556.4571
##  9    347    JFK   SJU     1598      173 554.2197
## 10   1503    JFK   SJU     1598      173 554.2197
## # ... with 336,766 more rows
```

summarise()

```
flights %>%
  summarise(avgDelay = mean(dep_delay, na.rm = TRUE))
```

```
## # A tibble: 1 x 1
##   avgDelay
##      <dbl>
## 1 12.63907
```

group_by()

```r
flights %>%
  group_by(year, month) %>%
  summarise(avgDelay = mean(dep_delay, na.rm = TRUE),
            numberOfFlights = n()) %>%
  ungroup()
```

```
## # A tibble: 12 x 4
##     year month  avgDelay numberOfFlights
##    <int> <int>     <dbl>           <int>
##  1  2013     1 10.036665           27004
##  2  2013     2 10.816843           24951
##  3  2013     3 13.227076           28834
##  4  2013     4 13.938038           28330
##  5  2013     5 12.986859           28796
##  6  2013     6 20.846332           28243
##  7  2013     7 21.727787           29425
##  8  2013     8 12.611040           29327
##  9  2013     9  6.722476           27574
## 10  2013    10  6.243988           28889
## 11  2013    11  5.435362           27268
## 12  2013    12 16.576688           28135
```

usefull functions: `n()`, `n_distinct(x)`, `first(x)`, `last(x)`, `nth(x, n)`

sample_n()

```r
flights %>%
  sample_n(10)
```

```
## # A tibble: 10 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013    11    19     1457           1459        -2     1631
##  2  2013     2    14     1459           1459         0     1614
##  3  2013     7    28     1641           1610        31       NA
##  4  2013    12    11     1250           1300       -10     1353
##  5  2013     7    23      611            605         6      713
##  6  2013     7     4     1528           1535        -7     1653
##  7  2013     8    23     1540           1425        75     1715
##  8  2013     3    27     1718           1720        -2     2010
##  9  2013     2    18     1821           1825        -4     2042
## 10  2013     3     5     1642           1610        32     1744
## # ... with 12 more variables: sched_arr_time <int>, arr_delay <dbl>,
## #   carrier <chr>, flight <int>, tailnum <chr>, origin <chr>, dest <chr>,
## #   air_time <dbl>, distance <dbl>, hour <dbl>, minute <dbl>,
## #   time_hour <dttm>
```

sample_frac()

```
flights %>%
  sample_frac(1/10)
```

```
## # A tibble: 33,678 x 19
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     8    21     1722           1725        -3     1908
##  2  2013    10     8     1726           1720         6     2015
##  3  2013    10     4     2059           2045        14     2228
##  4  2013    12    13     1922           1930        -8     2034
##  5  2013     7     9     1850           1734        76     2207
##  6  2013    10    22     1446           1435        11     1632
##  7  2013     3    11      627            630        -3      956
##  8  2013     4    10      629            634        -5      840
##  9  2013    10     8     1352           1355        -3     1621
## 10  2013     5     7     1125           1131        -6     1441
## # ... with 33,668 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

Grouped operations

```r
flights %>%
  group_by(year, month, day) %>%
  arrange(sched_dep_time, .by_group = TRUE)
```

```
## # A tibble: 336,776 x 19
## # Groups:   year, month, day [365]
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1      517            515         2      830
##  2  2013     1     1      533            529         4      850
##  3  2013     1     1      542            540         2      923
##  4  2013     1     1      544            545        -1     1004
##  5  2013     1     1      554            558        -4      740
##  6  2013     1     1      559            559         0      702
##  7  2013     1     1      554            600        -6      812
##  8  2013     1     1      555            600        -5      913
##  9  2013     1     1      557            600        -3      709
## 10  2013     1     1      557            600        -3      838
## # ... with 336,766 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

```r
flights %>%
  group_by(month) %>%
  mutate(avgDepDelay = mean(dep_delay, na.rm = TRUE)) %>%
  select(year: dep_delay, avgDepDelay)
```

```
## # A tibble: 336,776 x 7
## # Groups:   month [12]
##     year month   day dep_time sched_dep_time dep_delay avgDepDelay
##    <int> <int> <int>    <int>          <int>     <dbl>       <dbl>
##  1  2013     1     1      517            515         2    10.03667
##  2  2013     1     1      533            529         4    10.03667
##  3  2013     1     1      542            540         2    10.03667
##  4  2013     1     1      544            545        -1    10.03667
##  5  2013     1     1      554            600        -6    10.03667
##  6  2013     1     1      554            558        -4    10.03667
##  7  2013     1     1      555            600        -5    10.03667
##  8  2013     1     1      557            600        -3    10.03667
##  9  2013     1     1      557            600        -3    10.03667
## 10  2013     1     1      558            600        -2    10.03667
## # ... with 336,766 more rows
```

```
flights %>%
  group_by(year, month, day) %>%
  sample_n(10)
```

```
## # A tibble: 3,650 x 19
## # Groups:   year, month, day [365]
##     year month   day dep_time sched_dep_time dep_delay arr_time
##    <int> <int> <int>    <int>          <int>     <dbl>    <int>
##  1  2013     1     1     1604           1510        54     1817
##  2  2013     1     1      856            900        -4     1226
##  3  2013     1     1     2221           2000       141     2331
##  4  2013     1     1     1306           1240        26     1622
##  5  2013     1     1     2240           2245        -5     2340
##  6  2013     1     1      743            730        13     1107
##  7  2013     1     1     1112           1100        12     1440
##  8  2013     1     1     1356           1350         6     1612
##  9  2013     1     1      646            645         1      910
## 10  2013     1     1     1657           1650         7     1921
## # ... with 3,640 more rows, and 12 more variables: sched_arr_time <int>,
## #   arr_delay <dbl>, carrier <chr>, flight <int>, tailnum <chr>,
## #   origin <chr>, dest <chr>, air_time <dbl>, distance <dbl>, hour <dbl>,
## #   minute <dbl>, time_hour <dttm>
```

Speed

```
system.time(flights %>%
  group_by(month) %>%
  summarise(avgDelay = mean(dep_delay, na.rm = TRUE),
            minDelay = min(dep_delay, na.rm = TRUE),
            maxDelay = max(dep_delay, na.rm = TRUE)))
```

```
##    user  system elapsed
##    0.05    0.00    0.04
```

```
system.time(aggregate(dep_delay ~ month, data = flights,
          FUN = function(x) c(avgDelay = mean(x, na.rm = TRUE),
                              minDelay = min(x, na.rm = TRUE),
                              maxDelay = max(x, na.rm = TRUE))))
```

```
##    user  system elapsed
##    0.37    0.01    0.40
```

- “(. . .) both packages (dplyr, data.table) to be comparable in “split apply combine” style analysis, except when there are very large numbers of groups (>100K) at which point data.table becomes substantially faster.”