Post01: What dplyr Provides for Data Analysts

Haibin Lim

10/31/2017

** Post01: "What dplyr Provides for Data Analysts" **

What is dplyr? Why do we need it?

- dplyr is a package in R that contains functions associated with manipulating tables and data frames. Dplyr provides alternate ways other than [,] and dollar signs($) to select rows and columns in order to make a new data frame.
- This post will contain detailed description of each usage of each functions in dplyr and show why the dplyr is more useful that other functions and notations with manipulating date frames.
- dplyr contains functions such as mutate(), arrange(), select(), filter(), group_by(), and etc. This post has examples as well as explanations on how each functions are used and some graphs that can be made from these functions using ggplot (details of ggplot will not be mentioned in this post).

Examples with graphs (what can be graphed, or visualized?)

- In this post, data of star wars character will be used; the data was called from our stat133 github data folder. Only 10 of the characters will be analyzed.

- let’s first down the data table to be used as an example.

```r
library(readr)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.2
```

```
##
## 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
```

```r
library(ggplot2)
dat <- read_csv('/Users/haibinlim/stat133/stat133-hws-fall17/post01/data/starwars.csv')
```

```
## Parsed with column specification:
## cols(
##   name = col_character(),
##   gender = col_character(),
##   height = col_double(),
##   weight = col_double(),
##   eyecolor = col_character(),
##   haircolor = col_character(),
##   skincolor = col_character(),
##   homeland = col_character(),
##   born = col_character(),
##   died = col_character(),
##   jedi = col_character(),
##   species = col_character(),
##   weapon = col_character()
## )
```

```
dat
```

```
## # A tibble: 20 x 13
##                name gender height weight eyecolor haircolor skincolor
##               <chr>  <chr>  <dbl>  <dbl>    <chr>     <chr>     <chr>
##  1 Anakin Skywalker   male   1.88   84.0     blue     blond      fair
##  2    Padme Amidala female   1.65   45.0    brown     brown     light
##  3   Luke Skywalker   male   1.72   77.0     blue     blond      fair
##  4   Leia Skywalker female   1.50   49.0    brown     brown     light
##  5     Qui-Gon Jinn   male   1.93   88.5     blue     brown     light
##  6   Obi-Wan Kenobi   male   1.82   77.0 bluegray    auburn      fair
##  7         Han Solo   male   1.80   80.0    brown     brown     light
##  8  Sheev Palpatine   male   1.73   75.0     blue       red      pale
##  9            R2-D2   male   0.96   32.0     <NA>      <NA>      <NA>
## 10            C-3PO   male   1.67   75.0     <NA>      <NA>      <NA>
## 11             Yoda   male   0.66   17.0    brown     brown     green
## 12       Darth Maul   male   1.75   80.0   yellow      none       red
## 13            Dooku   male   1.93   86.0    brown     brown     light
## 14        Chewbacca   male   2.28  112.0     blue     brown      <NA>
## 15            Jabba   male   3.90     NA   yellow      none tan-green
## 16 Lando Calrissian   male   1.78   79.0    brown     blank      dark
## 17        Boba Fett   male   1.83   78.0    brown     black     brown
## 18       Jango Fett   male   1.83   79.0    brown     black     brown
## 19         Grievous   male   2.16  159.0     gold     black    orange
## 20     Chief Chirpa   male   1.00   50.0    black      gray     brown
## # ... with 6 more variables: homeland <chr>, born <chr>, died <chr>,
## #   jedi <chr>, species <chr>, weapon <chr>
```

* functions with dplyr include…

- ** select() - allows you to select specific columns **

```
eyec_dat <- select(dat, name, eyecolor)
eyec_dat
```

```
## # A tibble: 20 x 2
##                name eyecolor
##               <chr>    <chr>
##  1 Anakin Skywalker     blue
##  2    Padme Amidala    brown
##  3   Luke Skywalker     blue
##  4   Leia Skywalker    brown
##  5     Qui-Gon Jinn     blue
##  6   Obi-Wan Kenobi bluegray
##  7         Han Solo    brown
##  8  Sheev Palpatine     blue
##  9            R2-D2     <NA>
## 10            C-3PO     <NA>
## 11             Yoda    brown
## 12       Darth Maul   yellow
## 13            Dooku    brown
## 14        Chewbacca     blue
## 15            Jabba   yellow
## 16 Lando Calrissian    brown
## 17        Boba Fett    brown
## 18       Jango Fett    brown
## 19         Grievous     gold
## 20     Chief Chirpa    black
```

- to select all columns except a certian column use “-” (its subtractor operator)

```
except_name <- head(select(dat, -name))
except_name
```

```
## # A tibble: 6 x 12
##   gender height weight eyecolor haircolor skincolor   homeland    born
##    <chr>  <dbl>  <dbl>    <chr>     <chr>     <chr>      <chr>   <chr>
## 1   male   1.88   84.0     blue     blond      fair   Tatooine 41.9BBY
## 2 female   1.65   45.0    brown     brown     light      Naboo   46BBY
## 3   male   1.72   77.0     blue     blond      fair   Tatooine   19BBY
## 4 female   1.50   49.0    brown     brown     light   Alderaan   19BBY
## 5   male   1.93   88.5     blue     brown     light unk_planet   92BBY
## 6   male   1.82   77.0 bluegray    auburn      fair    Stewjon   57BBY
## # ... with 4 more variables: died <chr>, jedi <chr>, species <chr>,
## #   weapon <chr>
```

- selecting range of columns by name, we use “:” (in this case the data frame produces all data until skincolor)

```
skcolor <- select(dat, name:skincolor)
skcolor
```

```
## # A tibble: 20 x 7
##                 name gender height weight eyecolor haircolor skincolor
##                <chr>  <chr>  <dbl>  <dbl>    <chr>     <chr>     <chr>
##  1 Anakin Skywalker   male   1.88   84.0     blue     blond      fair
##  2    Padme Amidala female   1.65   45.0    brown     brown     light
##  3   Luke Skywalker   male   1.72   77.0     blue     blond      fair
##  4   Leia Skywalker female   1.50   49.0    brown     brown     light
##  5     Qui-Gon Jinn   male   1.93   88.5     blue     brown     light
##  6   Obi-Wan Kenobi   male   1.82   77.0 bluegray    auburn      fair
##  7         Han Solo   male   1.80   80.0    brown     brown     light
##  8  Sheev Palpatine   male   1.73   75.0     blue       red      pale
##  9            R2-D2   male   0.96   32.0     <NA>      <NA>      <NA>
## 10            C-3PO   male   1.67   75.0     <NA>      <NA>      <NA>
## 11             Yoda   male   0.66   17.0    brown     brown     green
## 12       Darth Maul   male   1.75   80.0   yellow      none       red
## 13            Dooku   male   1.93   86.0    brown     brown     light
## 14        Chewbacca   male   2.28  112.0     blue     brown      <NA>
## 15            Jabba   male   3.90     NA   yellow      none tan-green
## 16 Lando Calrissian   male   1.78   79.0    brown     blank      dark
## 17        Boba Fett   male   1.83   78.0    brown     black     brown
## 18       Jango Fett   male   1.83   79.0    brown     black     brown
## 19         Grievous   male   2.16  159.0     gold     black    orange
## 20     Chief Chirpa   male   1.00   50.0    black      gray     brown
```

- In case of bigger data frame with more columns, you can select a column with a letter that starts with a character string such as “g”, or “h”, using “starts_with”.

```
select(dat, starts_with("g"))
```

```
## # A tibble: 20 x 1
##    gender
##     <chr>
##  1   male
##  2 female
##  3   male
##  4 female
##  5   male
##  6   male
##  7   male
##  8   male
##  9   male
## 10   male
## 11   male
## 12   male
## 13   male
## 14   male
## 15   male
## 16   male
## 17   male
## 18   male
## 19   male
## 20   male
```

instead of “starts_with” you can use “ends_with”, “contains”, “matches”,“one_of”

- ** filter functions extract certain columns and rows bigger than or smaller than the called values**

```
filter(dat, weight >= 50)
```

```
## # A tibble: 15 x 13
##                 name gender height weight eyecolor haircolor skincolor
##                <chr>  <chr>  <dbl>  <dbl>    <chr>     <chr>     <chr>
##  1 Anakin Skywalker   male   1.88   84.0     blue     blond      fair
##  2   Luke Skywalker   male   1.72   77.0     blue     blond      fair
##  3     Qui-Gon Jinn   male   1.93   88.5     blue     brown     light
##  4   Obi-Wan Kenobi   male   1.82   77.0 bluegray    auburn      fair
##  5         Han Solo   male   1.80   80.0    brown     brown     light
##  6  Sheev Palpatine   male   1.73   75.0     blue       red      pale
##  7            C-3PO   male   1.67   75.0     <NA>      <NA>      <NA>
##  8       Darth Maul   male   1.75   80.0   yellow      none       red
##  9            Dooku   male   1.93   86.0    brown     brown     light
## 10        Chewbacca   male   2.28  112.0     blue     brown      <NA>
## 11 Lando Calrissian   male   1.78   79.0    brown     blank      dark
## 12        Boba Fett   male   1.83   78.0    brown     black     brown
## 13       Jango Fett   male   1.83   79.0    brown     black     brown
## 14         Grievous   male   2.16  159.0     gold     black    orange
## 15     Chief Chirpa   male   1.00   50.0    black      gray     brown
## # ... with 6 more variables: homeland <chr>, born <chr>, died <chr>,
## #   jedi <chr>, species <chr>, weapon <chr>
```

- you can do multiple extractions such as weight greater than or equal to 50 and height less than or equal to 1.8

```r
new_wh <- filter(dat, weight >=50, height <= 1.8)
new_wh
```

```
## # A tibble: 7 x 13
##                name gender height weight eyecolor haircolor skincolor
##               <chr>  <chr>  <dbl>  <dbl>    <chr>     <chr>     <chr>
## 1   Luke Skywalker   male   1.72     77     blue     blond      fair
## 2         Han Solo   male   1.80     80    brown     brown     light
## 3  Sheev Palpatine   male   1.73     75     blue       red      pale
## 4            C-3PO   male   1.67     75     <NA>      <NA>      <NA>
## 5       Darth Maul   male   1.75     80   yellow      none       red
## 6 Lando Calrissian   male   1.78     79    brown     blank      dark
## 7     Chief Chirpa   male   1.00     50    black      gray     brown
## # ... with 6 more variables: homeland <chr>, born <chr>, died <chr>,
## #   jedi <chr>, species <chr>, weapon <chr>
```

- filter functions can also organize non-numerical columns. I can extract eye colors blue and black.

```r
filter(dat, eyecolor %in% c("blue", "black"))
```

```
## # A tibble: 6 x 13
##                name gender height weight eyecolor haircolor skincolor
##               <chr>  <chr>  <dbl>  <dbl>    <chr>     <chr>     <chr>
## 1 Anakin Skywalker   male   1.88   84.0     blue     blond      fair
## 2   Luke Skywalker   male   1.72   77.0     blue     blond      fair
## 3     Qui-Gon Jinn   male   1.93   88.5     blue     brown     light
## 4  Sheev Palpatine   male   1.73   75.0     blue       red      pale
## 5        Chewbacca   male   2.28  112.0     blue     brown      <NA>
## 6     Chief Chirpa   male   1.00   50.0    black      gray     brown
## # ... with 6 more variables: homeland <chr>, born <chr>, died <chr>,
## #   jedi <chr>, species <chr>, weapon <chr>
```

- ** pipe operator (%<%), you can use pipe operator instead of nesting **

```r
dat %>%
  select(name, eyecolor)
```

```
## # A tibble: 20 x 2
##                 name eyecolor
##                <chr>    <chr>
##  1 Anakin Skywalker     blue
##  2    Padme Amidala    brown
##  3   Luke Skywalker     blue
##  4   Leia Skywalker    brown
##  5     Qui-Gon Jinn     blue
##  6   Obi-Wan Kenobi bluegray
##  7         Han Solo    brown
##  8  Sheev Palpatine     blue
##  9            R2-D2     <NA>
## 10            C-3PO     <NA>
## 11             Yoda    brown
## 12       Darth Maul   yellow
## 13            Dooku    brown
## 14        Chewbacca     blue
## 15            Jabba   yellow
## 16 Lando Calrissian    brown
## 17        Boba Fett    brown
## 18       Jango Fett    brown
## 19         Grievous     gold
## 20     Chief Chirpa    black
```

- this is the same thing as using

```r
justeye <- select(dat, name, eyecolor)
justeye
```

```
## # A tibble: 20 x 2
##                 name eyecolor
##                <chr>    <chr>
##  1 Anakin Skywalker     blue
##  2    Padme Amidala    brown
##  3   Luke Skywalker     blue
##  4   Leia Skywalker    brown
##  5     Qui-Gon Jinn     blue
##  6   Obi-Wan Kenobi bluegray
##  7          Han Solo    brown
##  8  Sheev Palpatine     blue
##  9             R2-D2     <NA>
## 10             C-3PO     <NA>
## 11              Yoda    brown
## 12        Darth Maul   yellow
## 13             Dooku    brown
## 14         Chewbacca     blue
## 15             Jabba   yellow
## 16 Lando Calrissian    brown
## 17         Boba Fett    brown
## 18        Jango Fett    brown
## 19          Grievous     gold
## 20     Chief Chirpa    black
```

- *Right now, it seems like there isn't a reason we should use pipe operator but pipe operator comes in more handy once many functions need to be combined*

- ** arrange: re-orders rows by specific column (taxonomically) **

```
dat %>%
  arrange(haircolor)
```

```
## # A tibble: 20 x 13
##                 name gender height weight eyecolor haircolor skincolor
##                <chr>  <chr>  <dbl>  <dbl>    <chr>     <chr>     <chr>
##  1   Obi-Wan Kenobi   male   1.82   77.0 bluegray    auburn      fair
##  2        Boba Fett   male   1.83   78.0    brown     black     brown
##  3       Jango Fett   male   1.83   79.0    brown     black     brown
##  4         Grievous   male   2.16  159.0     gold     black    orange
##  5 Lando Calrissian   male   1.78   79.0    brown     blank      dark
##  6 Anakin Skywalker   male   1.88   84.0     blue     blond      fair
##  7   Luke Skywalker   male   1.72   77.0     blue     blond      fair
##  8    Padme Amidala female   1.65   45.0    brown     brown     light
##  9   Leia Skywalker female   1.50   49.0    brown     brown     light
## 10     Qui-Gon Jinn   male   1.93   88.5     blue     brown     light
## 11          Han Solo   male   1.80   80.0    brown     brown     light
## 12              Yoda   male   0.66   17.0    brown     brown     green
## 13             Dooku   male   1.93   86.0    brown     brown     light
## 14         Chewbacca   male   2.28  112.0     blue     brown      <NA>
## 15     Chief Chirpa   male   1.00   50.0    black      gray     brown
## 16       Darth Maul   male   1.75   80.0   yellow      none       red
## 17            Jabba   male   3.90     NA   yellow      none tan-green
## 18  Sheev Palpatine   male   1.73   75.0     blue       red      pale
## 19            R2-D2   male   0.96   32.0     <NA>      <NA>      <NA>
## 20            C-3PO   male   1.67   75.0     <NA>      <NA>      <NA>
## # ... with 6 more variables: homeland <chr>, born <chr>, died <chr>,
## #   jedi <chr>, species <chr>, weapon <chr>
```

- With arrange functions, you can arrange the values in descending order (desc()), or increasing order (inc())
- in this case, I can select name, weight, eyecolor, haircolor and arrange the weight (numerical data)

```
dat %>%
  select(name, weight, eyecolor, skincolor) %>%
  arrange(skincolor, desc(weight))
```

```
## # A tibble: 20 x 4
##                name weight eyecolor skincolor
##               <chr>  <dbl>    <chr>     <chr>
##  1       Jango Fett   79.0    brown     brown
##  2        Boba Fett   78.0    brown     brown
##  3     Chief Chirpa   50.0    black     brown
##  4 Lando Calrissian   79.0    brown      dark
##  5 Anakin Skywalker   84.0     blue      fair
##  6   Luke Skywalker   77.0     blue      fair
##  7   Obi-Wan Kenobi   77.0 bluegray      fair
##  8             Yoda   17.0    brown     green
##  9     Qui-Gon Jinn   88.5     blue     light
## 10            Dooku   86.0    brown     light
## 11         Han Solo   80.0    brown     light
## 12   Leia Skywalker   49.0    brown     light
## 13    Padme Amidala   45.0    brown     light
## 14         Grievous  159.0     gold    orange
## 15  Sheev Palpatine   75.0     blue      pale
## 16       Darth Maul   80.0   yellow       red
## 17            Jabba     NA   yellow tan-green
## 18        Chewbacca  112.0     blue      <NA>
## 19            C-3PO   75.0     <NA>      <NA>
## 20            R2-D2   32.0     <NA>      <NA>
```

- ** mutate function creates new columns **
- We can add numerical proportions of weight and height

```
dat %>%
  mutate(weight_height_prop = weight/height)
```

```
## # A tibble: 20 x 14
##                name gender height weight eyecolor haircolor skincolor
##               <chr>  <chr>  <dbl>  <dbl>    <chr>     <chr>     <chr>
##  1 Anakin Skywalker   male   1.88   84.0     blue     blond      fair
##  2    Padme Amidala female   1.65   45.0    brown     brown     light
##  3   Luke Skywalker   male   1.72   77.0     blue     blond      fair
##  4   Leia Skywalker female   1.50   49.0    brown     brown     light
##  5     Qui-Gon Jinn   male   1.93   88.5     blue     brown     light
##  6   Obi-Wan Kenobi   male   1.82   77.0 bluegray    auburn      fair
##  7         Han Solo   male   1.80   80.0    brown     brown     light
##  8  Sheev Palpatine   male   1.73   75.0     blue       red      pale
##  9            R2-D2   male   0.96   32.0     <NA>      <NA>      <NA>
## 10            C-3PO   male   1.67   75.0     <NA>      <NA>      <NA>
## 11             Yoda   male   0.66   17.0    brown     brown     green
## 12       Darth Maul   male   1.75   80.0   yellow      none       red
## 13            Dooku   male   1.93   86.0    brown     brown     light
## 14        Chewbacca   male   2.28  112.0     blue     brown      <NA>
## 15            Jabba   male   3.90     NA   yellow      none tan-green
## 16 Lando Calrissian   male   1.78   79.0    brown     blank      dark
## 17        Boba Fett   male   1.83   78.0    brown     black     brown
## 18       Jango Fett   male   1.83   79.0    brown     black     brown
## 19         Grievous   male   2.16  159.0     gold     black    orange
## 20     Chief Chirpa   male   1.00   50.0    black      gray     brown
## # ... with 7 more variables: homeland <chr>, born <chr>, died <chr>,
## #   jedi <chr>, species <chr>, weapon <chr>, weight_height_prop <dbl>
```

- there are more functions besides mean(): sd(), max(), min(), median(), sum().
- more functions used within summary: n() (calculates length of vector), first() (returns first value in vector), last() (reutnrs last value of vector), n_distinct() (number of distinct values). These are some functions that were not used often in class but they come in handy when you have to

- ** summarise function creates summary statistics such as the mean, max, etc. **
- there is a mean() function that calculates the average of the given column.

```
dat %>%
  summarise(avg_height = mean(height))
```

```
## # A tibble: 1 x 1
##   avg_height
##        <dbl>
## 1      1.789
```

- more examples with more functions

```
dat %>%
  summarise(max_height = max(height), min_height = min(height), total = n())
```

```
## # A tibble: 1 x 3
##   max_height min_height total
##        <dbl>      <dbl> <int>
## 1        3.9       0.66    20
```

- ** group_by function relates back to the concept of "split-apply-combine" **
- group_by allows you to select certain columns and split the data

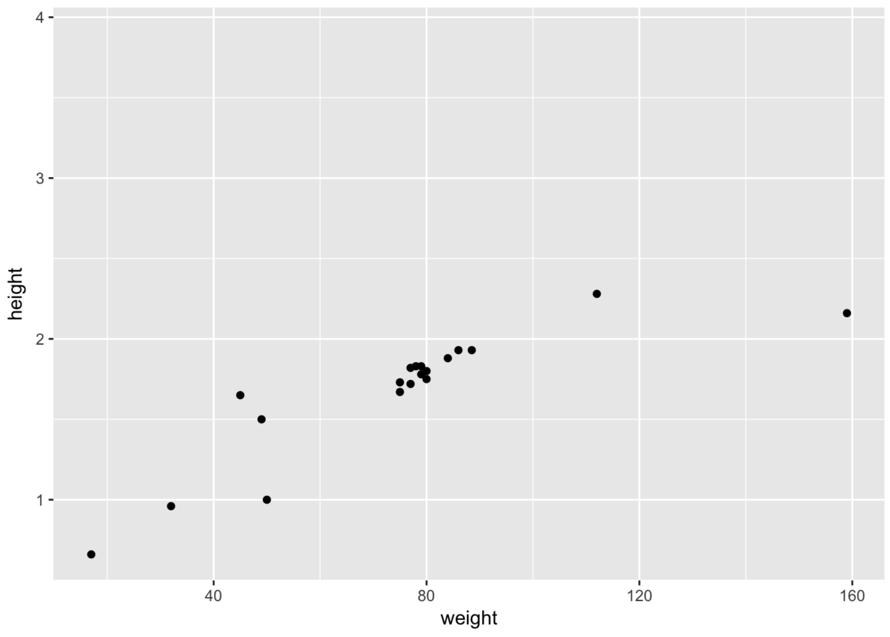
```
dat %>%
  group_by(homeland) %>%
  summarise(avg_height = mean(height),
            max_height = max(height),
            total = n()
            )
```

```
## # A tibble: 14 x 4
##       homeland avg_height max_height total
##          <chr>      <dbl>      <dbl> <int>
##  1    Alderaan   1.500000       1.50     1
##  2 ConcordDawn   1.830000       1.83     1
##  3    Corellia   1.800000       1.80     1
##  4    Dathomir   1.750000       1.75     1
##  5       Endor   1.000000       1.00     1
##  6       Kalee   2.160000       2.16     1
##  7      Kamino   1.830000       1.83     1
##  8    Kashyyyk   2.280000       2.28     1
##  9       Naboo   1.446667       1.73     3
## 10     Serenno   1.930000       1.93     1
## 11     Socorro   1.780000       1.78     1
## 12     Stewjon   1.820000       1.82     1
## 13    Tatooine   2.292500       3.90     4
## 14  unk_planet   1.295000       1.93     2
```

- Dplyr does not involve alot of graphing. But dplyr functions can help when it comes to using ggplot with analyzed data.
- this is a simple usage of ggplot with weight and height as the basis of the axis.

```
ggplot(data = dat, aes(x = weight, y = height)) + geom_point()
```

```
## Warning: Removed 1 rows containing missing values (geom_point).
```



Reference page

1. for basic explanations regarding simple usage of functions in dplyr click
2. if reading through the explanations weren't enough, the "Hands-on dplyr" provides video tutorials
3. Clean example of piping
4. Selecting rows with specific values click
5. Another dplyr example of usage of functions with shorter data frame click here
6. Formatting this blog assignment reference to Markdown Basics
7. Data table taken from our class github