A Grammar for Data Manipulation

Azarudheen

February 19, 2019

Brainstorming

Airquality is a dataset which gives the daily air quality measurements in New York from May to September 1973.

The data frame contains 154 observations with 6 vairables which are Ozone, Solar, Wind, Temp, Month, Day.

You have an assignment for finding the average wind speed with Temperature haveing more than 75 degrees in the first 5 days of June and July.

How will you get it done. . . .. ?

Introduction

dplyr is a package for data manipulation, written and maintained by Hadley Wickham.

It provides some great, easy-to-use functions that are very handy when performing exploratory data analysis and manipulation.

How to get dplyr

```r
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.4.4

##
## 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
```

Lets Start . . .

```
data(airquality)
```

- tbl_df creates a local data frame which will simply wrap the data set and print neatly

```
aq = tbl_df(airquality)
aq
```

```
## # A tibble: 153 x 6
##    Ozone Solar.R  Wind  Temp Month   Day
##    <int>   <int> <dbl> <int> <int> <int>
##  1    41     190   7.4    67     5     1
##  2    36     118   8      72     5     2
##  3    12     149  12.6    74     5     3
##  4    18     313  11.5    62     5     4
##  5    NA      NA  14.3    56     5     5
##  6    28      NA  14.9    66     5     6
##  7    23     299   8.6    65     5     7
```

Filter

- Base R approach to filtering forces you to repeat the data frame's name.
- The filter function will return all the rows that satisfy a following condition.
- Suppose if we need to return all the rows and column were the Temp is larger than 70. How will you do ?

```r
#dplyr approach
filter(aq, Temp>70)
```

```
## # A tibble: 120 x 6
##    Ozone Solar.R  Wind  Temp Month   Day
##    <int>   <int> <dbl> <int> <int> <int>
##  1    36     118   8      72     5     2
##  2    12     149  12.6    74     5     3
##  3     7      NA   6.9    74     5    11
##  4    11     320  16.6    73     5    22
##  5    45     252  14.9    81     5    29
##  6   115     223   5.7    79     5    30
##  7    37     279   7.4    76     5    31
##  8    NA     286   8.6    78     6     1
##  9    NA     287   9.7    74     6     2
## 10    NA     186   9.2    84     6     4
## # ... with 110 more rows
```

- Suppose we need the airquality measures for May were the Temp is lesser than 70.
- How can you do?

```r
filter(aq, Temp<70, Month == 5)
```

```
## # A tibble: 24 x 6
##    Ozone Solar.R  Wind  Temp Month   Day
##    <int>   <int> <dbl> <int> <int> <int>
##  1    41     190   7.4    67     5     1
##  2    18     313  11.5    62     5     4
##  3    NA      NA  14.3    56     5     5
##  4    28      NA  14.9    66     5     6
##  5    23     299   8.6    65     5     7
##  6    19      99  13.8    59     5     8
##  7     8      19  20.1    61     5     9
##  8    NA     194   8.6    69     5    10
##  9    16     256   9.7    69     5    12
## 10    11     290   9.2    66     5    13
## # ... with 14 more rows
```

```r
filter(aq, Temp<70 & Month == 5)
```

```
## # A tibble: 24 x 6
```

Try this. . .

- Filter the first 5 days airquality measures of every month which has the Wind greater than 8.

```r
filter(aq, Wind>8 & Day <=5)
```

```
## # A tibble: 12 x 6
##    Ozone Solar.R  Wind  Temp Month   Day
##    <int>   <int> <dbl> <int> <int> <int>
##  1    12     149  12.6    74     5     3
##  2    18     313  11.5    62     5     4
##  3    NA      NA  14.3    56     5     5
##  4    NA     286   8.6    78     6     1
##  5    NA     287   9.7    74     6     2
##  6    NA     242  16.1    67     6     3
##  7    NA     186   9.2    84     6     4
##  8    NA     220   8.6    85     6     5
##  9    49     248   9.2    85     7     2
## 10    32     236   9.2    81     7     3
## 11    NA     101  10.9    84     7     4
## 12     9      24  13.8    81     8     2
```

Select

- Select the Ozone, Wind and Temp columns from the dataframe.
- Try to get this with Base R comments.

```
head(select(aq, Ozone, Solar.R, Wind, Temp))
```

```
## # A tibble: 6 x 4
##   Ozone Solar.R  Wind  Temp
##   <int>   <int> <dbl> <int>
## 1    41     190   7.4    67
## 2    36     118   8      72
## 3    12     149  12.6    74
## 4    18     313  11.5    62
## 5    NA      NA  14.3    56
## 6    28      NA  14.9    66
```

```
head(select(aq, Ozone:Temp))
```

```
## # A tibble: 6 x 4
##   Ozone Solar.R  Wind  Temp
##   <int>   <int> <dbl> <int>
## 1    41     190   7.4    67
## 2    36     118   8      72
## 3    12     149  12.6    74
```

```r
# To select all the columns except the column Ozone
select(aq, -Ozone)
```

```
## # A tibble: 153 x 5
##    Solar.R  Wind  Temp Month   Day
##      <int> <dbl> <int> <int> <int>
##  1     190   7.4    67     5     1
##  2     118   8      72     5     2
##  3     149  12.6    74     5     3
##  4     313  11.5    62     5     4
##  5      NA  14.3    56     5     5
##  6      NA  14.9    66     5     6
##  7     299   8.6    65     5     7
##  8      99  13.8    59     5     8
##  9      19  20.1    61     5     9
## 10     194   8.6    69     5    10
## # ... with 143 more rows
```

Chaining or Pipelining (%>%)

- Select the Wind, Solar.R, and Temp for May and June Month

```r
aq %>%
  select(Wind, Solar.R, Temp, Month)%>%
  filter(Month <= 6)
```

```
## # A tibble: 61 x 4
##     Wind Solar.R  Temp Month
##    <dbl>   <int> <int> <int>
##  1   7.4     190    67     5
##  2   8       118    72     5
##  3  12.6     149    74     5
##  4  11.5     313    62     5
##  5  14.3      NA    56     5
##  6  14.9      NA    66     5
##  7   8.6     299    65     5
##  8  13.8      99    59     5
##  9  20.1      19    61     5
## 10   8.6     194    69     5
## # ... with 51 more rows
```

Try this. . .

- mtcars is data frame which comprises fuel consumption and 10 aspects of automobile design and performance of 32 automobiles.
- Select the Horsepower, Weights, Rear axle ratio and Miles/gallon for 6 cylinders automatic automobile.

```r
mtcars%>%
  select(hp, wt, drat, mpg, cyl, am)%>%
  filter(cyl == 6 & am == 0)
```

```
##    hp    wt drat  mpg cyl am
## 1 110 3.215 3.08 21.4   6  0
## 2 105 3.460 2.76 18.1   6  0
## 3 123 3.440 3.92 19.2   6  0
## 4 123 3.440 3.92 17.8   6  0
```

```
# If we need to select all the columns which starts with th
head(select(tbl_df(mtcars), starts_with("c")))
```

```
## # A tibble: 6 x 2
##     cyl  carb
##   <dbl> <dbl>
## 1     6     4
## 2     6     4
## 3     4     1
## 4     6     1
## 5     8     2
## 6     6     1
```

```
head(select(tbl_df(mtcars), ends_with("p")))
```

```
## # A tibble: 6 x 2
##    disp    hp
##   <dbl> <dbl>
## 1   160   110
## 2   160   110
```

```r
# If we need to select all the colums which starts with le
select(tbl_df(mtcars), starts_with("c"), ends_with("p"))
```

```
## # A tibble: 32 x 4
##       cyl  carb  disp    hp
##     <dbl> <dbl> <dbl> <dbl>
##  1      6     4  160    110
##  2      6     4  160    110
##  3      4     1  108     93
##  4      6     1  258    110
##  5      8     2  360    175
##  6      6     1  225    105
##  7      8     4  360    245
##  8      4     2  147.    62
##  9      4     2  141.    95
## 10      6     4  168.   123
## # ... with 22 more rows
```

Mutate

- Add a new column to the airquality datafram that displays the Temp in Celsius.

```r
mutate(aq, TempC = (Temp-32)*5/9)
```

```
## # A tibble: 153 x 7
##     Ozone Solar.R  Wind  Temp Month   Day TempC
##     <int>   <int> <dbl> <int> <int> <int> <dbl>
##  1    41     190   7.4    67     5     1  19.4
##  2    36     118   8      72     5     2  22.2
##  3    12     149  12.6    74     5     3  23.3
##  4    18     313  11.5    62     5     4  16.7
##  5    NA      NA  14.3    56     5     5  13.3
##  6    28      NA  14.9    66     5     6  18.9
##  7    23     299   8.6    65     5     7  18.3
##  8    19      99  13.8    59     5     8  15
##  9     8      19  20.1    61     5     9  16.1
## 10    NA     194   8.6    69     5    10  20.6
## # ... with 143 more rows
```

Try this

- Convert the weights of each automobile from lbs to kg and display that in a column WtKg

```r
tbl_df(mtcars)%>%
  mutate(WtKg = wt * 0.45359237)%>%
  select(wt,WtKg)
```

```
## # A tibble: 32 x 2
##       wt  WtKg
##    <dbl> <dbl>
##  1  2.62  1.19
##  2  2.88  1.30
##  3  2.32  1.05
##  4  3.22  1.46
##  5  3.44  1.56
##  6  3.46  1.57
##  7  3.57  1.62
##  8  3.19  1.45
##  9  3.15  1.43
## 10  3.44  1.56
## # ... with 22 more rows
```

Left and Right Join

- Create two data frame with same ID and join the second dataset to first dataset.

```r
# Creating dataframe using tribble
df1 <- tribble(
  ~ID, ~y,
  "A", 5,
  "B", 6,
  "C", 8,
  "D", 9,
  "E", 10)
df2 = tribble(
  ~ID, ~x,
  "A", 11,
  "B", 15,
  "C", 20,
  "E", 22)
```

```r
# Joining df2 to df1 in the left
left_join(df1, df2, by = 'ID')
```

```
## # A tibble: 5 x 3
##   ID        y     x
##   <chr> <dbl> <dbl>
## 1 A         5    11
## 2 B         6    15
## 3 C         8    20
## 4 D         9    NA
## 5 E        10    22
```

```r
# Joinging df2 to df1 in the right
right_join(df1, df2, by = 'ID')
```

```
## # A tibble: 4 x 3
##   ID        y     x
##   <chr> <dbl> <dbl>
## 1 A         5    11
## 2 B         6    15
```

Inner Join

- When we are 100% sure that the two datasets won't match, we can consider to return only rows existing in both dataset.
- This is possible when we need a clean dataset or when we don't want to impute missing values with the mean or median.

```r
inner_join(df1, df2, by = 'ID')
```

```
## # A tibble: 4 x 3
##   ID          y     x
##   <chr> <dbl> <dbl>
## 1 A         5    11
## 2 B         6    15
## 3 C         8    20
## 4 E        10    22
```

```r
# Full join function keeps all the observations and replace
full_join(df1, df2, by = 'ID')
```

```
## # A tibble: 5 x 3
##   ID          y     x
##   <chr> <dbl> <dbl>
## 1 A         5    11
## 2 B         6    15
## 3 C         8    20
## 4 D         9    NA
```

Summarise

- Calculate the mean temperature for all the months in the airquality dataset.

```r
summarise(airquality,
          mean(Temp, na.rm = TRUE),
          sd(Temp, na.rm = TRUE))
```

```
##   mean(Temp, na.rm = TRUE) sd(Temp, na.rm = TRUE)
## 1                 77.88235                9.46527
```

Try this. . .

- Calculate the mean temperature for each months separately and display it.

```r
summarise(group_by(airquality, Month),
          mean(Temp, na.rm = TRUE))
```

```
## # A tibble: 5 x 2
##   Month `mean(Temp, na.rm = TRUE)`
##   <int>                      <dbl>
## 1     5                       65.5
## 2     6                       79.1
## 3     7                       83.9
## 4     8                       84.0
## 5     9                       76.9
```

Try this. . .

- Calculate the average wind speed in last 10 days of every month separately and display it.

```r
airquality %>%
  filter(Day>=20 & Day <=30)%>%
  group_by(Month)%>%
  summarise(mean(Wind, na.rm = TRUE))
```

```
## # A tibble: 5 x 2
##   Month `mean(Wind, na.rm = TRUE)`
##   <int>                      <dbl>
## 1     5                      11.8
## 2     6                       8.17
## 3     7                       9
## 4     8                       8.89
## 5     9                      11.0
```

generating Random Samples

```
strata = read.csv("D:/Christ_University/Data_Sets/Start.csv
strata1 = filter(strata, Age>=16, Age <=30)
strata2 = filter(strata, Age>=31 & Age <=40)
strata3 = filter(strata, Age>=41 & Age <=60)
S1_size = round((nrow(strata1)/nrow(strata))*60)
```

```r
sample_n(strata1, S1_size)
```

```
##    Age Sex PS Time
## 1   25    M  3  360
## 2   24    M  3  600
## 3   24    M  3  360
## 4   26    M  6  270
## 5   27    M  4  240
## 6   24    M  9   10
## 7   24    M  4   10
## 8   25    M  5   20
## 9   22    M  4   30
## 10  19    M  3   30
## 11  19    M  5   10
## 12  25    M  4   10
## 13  24    M  5  420
## 14  27    M  3 1140
## 15  20    M  6   10
## 16  23    M  5   10
## 17  17    M  3  720
```

Try this. . .

- Iris is a dataset which is contains the measurements of petal length, petal width, sepal length and sepal width for three iris species. Do the following,
- 1. Find the mean petal length and sepal length for the species veriscolor.
- 2. Find the mean sepal width whose petal width is more than 1 for each species separately and display it.
- 3. Stratify the Iris dataset based on species and hence select total of 50 random samples from the stratum and hence find correlation between petal length and sepal length for each species separately and display it.