Data wrangling with dplyr

CMSACamp 2020

June 02, 2020

Data frame

Our data are usually presented as a csv file and after loading a csv file into the R studio, we will achieve a data frame. A data frame can be considered a special case of matrix where each column usually represent a variable and all variables have exactly the same length.

```
nba <- read.csv("http://www.stat.cmu.edu/cmsac/sure/materials/data/intro_r/nba_2020_player_stats.csv")
nba_tibble <- dplyr::as_tibble(nba)
# Alt-Shift-K: keyboard shortcuts
# Alt-Dash: insert assignment arrow
```

We can use the functions head() and tail() to view a sample of the data frame. Examples are viewing the first six rows of the data frame "nba" and the last three rows.

```
head(nba)
```

```
##                    player position age team games games_started minutes_played
## 1             Steven Adams        C  26  OKC    58            58           1564
## 2              Bam Adebayo       PF  22  MIA    65            65           2235
## 3        LaMarcus Aldridge        C  34  SAS    53            53           1754
## 4 Nickeil Alexander-Walker       SG  21  NOP    41             0            501
## 5            Grayson Allen       SG  24  MEM    30             0            498
## 6            Jarrett Allen        C  21  BRK    64            58           1647
##   field_goals field_goal_attempts three_pointers three_point_attempts
## 1         262                443              1                    3
## 2         408                719              1                   13
## 3         391                793             61                  157
## 4          77                227             40                  117
## 5          79                176             33                   91
## 6         267                413              0                    5
##   two_pointers two_point_attempts free_throws free_throw_attempts
## 1          261                440         108                183
## 2          407                706         236                342
## 3          330                636         158                191
## 4           37                110          17                 28
## 5           46                 85          30                 35
## 6          267                408         147                237
##   offensive_rebounds defensive_rebounds assists steals blocks turnovers
## 1                196                347     141     50     65        86
## 2                165                518     333     78     85       185
## 3                103                289     129     36     87        74
## 4                  8                 72      74     11      7        40
## 5                  5                 61      43      6      1        23
## 6                195                410      85     37     85        72
```

```
##   personal_fouls
## 1            111
## 2            164
## 3            128
## 4             46
## 5             36
## 6            144
```

```
tail(nba, 3)
```

```
##          player position age team games games_started minutes_played
## 568 Cody Zeller        C  27  CHO    58            39           1341
## 569  Ante Žižić        C  23  CLE    22             0            221
## 570 Ivica Zubac        C  22  LAC    64            62           1156
##     field_goals field_goal_attempts three_pointers three_point_attempts
## 568         251                 479             18                   75
## 569          41                  72              0                    0
## 570         202                 336              0                    2
##     two_pointers two_point_attempts free_throws free_throw_attempts
## 568          233                 404         122                 179
## 569           41                  72          14                  19
## 570          202                 334         107                 141
##     offensive_rebounds defensive_rebounds assists steals blocks turnovers
## 568                160                251      88     40     25        75
## 569                 18                 48       6      7      5        10
## 570                166                295      69     13     60        53
##     personal_fouls
## 568            140
## 569             27
## 570            146
```

To view the dimentions of the data frame, use the function dim()

```
dim(nba)
```

```
## [1] 570  22
```

Tow view the summary statistics for all variables and the data structure types, use summary() and str().

```
summary(nba)
```

```
##     player            position              age            team
##  Length:570         Length:570         Min.   :19.00   Length:570
##  Class :character   Class :character   1st Qu.:22.00   Class :character
##  Mode  :character   Mode  :character   Median :25.00   Mode  :character
##                                        Mean   :25.61
##                                        3rd Qu.:28.00
##                                        Max.   :43.00
##      games       games_started   minutes_played    field_goals
##  Min.   : 1.00   Min.   : 0.00   Min.   :   1.0   Min.   :  0.0
##  1st Qu.:14.00   1st Qu.: 0.00   1st Qu.: 206.0   1st Qu.: 25.0
##  Median :40.00   Median : 4.00   Median : 715.5   Median : 92.5
##  Mean   :35.97   Mean   :17.04   Mean   : 823.4   Mean   :139.1
##  3rd Qu.:56.00   3rd Qu.:33.00   3rd Qu.:1396.8   3rd Qu.:216.0
##  Max.   :66.00   Max.   :65.00   Max.   :2243.0   Max.   :623.0
##  field_goal_attempts three_pointers   three_point_attempts  two_pointers
##  Min.   :   0.0      Min.   :  0.00   Min.   :  0.00       Min.   :  0.00
```

```
##  1st Qu.:  54.5      1st Qu.:  2.25   1st Qu.:  9.25       1st Qu.: 16.00
##  Median : 210.0      Median : 22.00   Median : 69.00       Median : 58.00
##  Mean   : 302.6      Mean   : 41.33   Mean   :115.63       Mean   : 97.78
##  3rd Qu.: 479.0      3rd Qu.: 65.00   3rd Qu.:186.50       3rd Qu.:153.00
##  Max.   :1386.0      Max.   :271.00   Max.   :769.00       Max.   :540.00
##  two_point_attempts  free_throws     free_throw_attempts offensive_rebounds
##  Min.   :  0.00     Min.   :  0.00   Min.   :  0.00      Min.   :  0.00
##  1st Qu.: 30.25     1st Qu.:  7.00   1st Qu.: 10.25      1st Qu.:  7.00
##  Median :114.50     Median : 32.00   Median : 44.00      Median : 21.00
##  Mean   :186.94     Mean   : 60.15   Mean   : 78.02      Mean   : 34.47
##  3rd Qu.:278.75     3rd Qu.: 76.00   3rd Qu.:102.75      3rd Qu.: 48.75
##  Max.   :998.00     Max.   :619.00   Max.   :719.00      Max.   :244.00
##  defensive_rebounds    assists          steals           blocks
##  Min.   :  0.0      Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 25.0      1st Qu.: 12.00   1st Qu.:  5.00   1st Qu.:  3.00
##  Median : 88.0      Median : 47.50   Median : 20.00   Median :  9.00
##  Mean   :118.4      Mean   : 82.95   Mean   : 26.09   Mean   : 16.85
##  3rd Qu.:182.5      3rd Qu.:109.00   3rd Qu.: 40.00   3rd Qu.: 22.00
##  Max.   :653.0      Max.   :636.00   Max.   :115.00   Max.   :187.00
##    turnovers      personal_fouls
##  Min.   :  0.00   Min.   :  0.00
##  1st Qu.:  9.00   1st Qu.: 20.00
##  Median : 33.50   Median : 64.00
##  Mean   : 47.29   Mean   : 70.13
##  3rd Qu.: 70.00   3rd Qu.:113.75
##  Max.   :289.00   Max.   :246.00
```

```
str(nba)
```

```
## 'data.frame':    570 obs. of  22 variables:
##  $ player              : chr  "Steven Adams" "Bam Adebayo" "LaMarcus Aldridge" "Nickeil Alexander-Wa
##  $ position            : chr  "C" "PF" "C" "SG" ...
##  $ age                 : int  26 22 34 21 24 21 27 29 26 26 ...
##  $ team                : chr  "OKC" "MIA" "SAS" "NOP" ...
##  $ games               : int  58 65 53 41 30 64 10 18 3 59 ...
##  $ games_started       : int  58 65 53 0 0 58 0 2 0 20 ...
##  $ minutes_played      : int  1564 2235 1754 501 498 1647 117 380 17 1140 ...
##  $ field_goals         : int  262 408 391 77 79 267 19 25 1 138 ...
##  $ field_goal_attempts : int  443 719 793 227 176 413 44 86 6 280 ...
##  $ three_pointers      : int  1 1 61 40 33 0 5 9 0 16 ...
##  $ three_point_attempts: int  3 13 157 117 91 5 16 36 3 62 ...
##  $ two_pointers        : int  261 407 330 37 46 267 14 16 1 122 ...
##  $ two_point_attempts  : int  440 706 636 110 85 408 28 50 3 218 ...
##  $ free_throws         : int  108 236 158 17 30 147 7 19 1 43 ...
##  $ free_throw_attempts : int  183 342 191 28 35 237 11 29 2 66 ...
##  $ offensive_rebounds  : int  196 165 103 8 5 195 2 24 0 54 ...
##  $ defensive_rebounds  : int  347 518 289 72 61 410 7 63 2 203 ...
##  $ assists             : int  141 333 129 74 43 85 21 21 0 132 ...
##  $ steals              : int  50 78 36 11 6 37 5 18 0 46 ...
##  $ blocks              : int  65 85 87 7 1 85 2 8 1 31 ...
##  $ turnovers           : int  86 185 74 40 23 72 8 17 0 55 ...
##  $ personal_fouls      : int  111 164 128 46 36 144 7 27 2 85 ...
```

Data manipulation with package "dplyr"

An easier way to manipulate the data frame is through the package "dplyr", which is included in the package "tidyverse". The operation we can do includes: selecting specific columns, filtering for rows, re-ordering rows, adding new columns and summarizing data. "Split-apply-combine" concept can also be achieved by "dplyr". Before we use the package, remember to call it:

```
library(tidyverse)
```

Selecting columns using select()

The function select() can be use to select certain column with the column names. The first parameter is the name of the data frame. For example, if I only need to view the column "player" and "games":

```
nba_pg <- select(nba, player, games)
head(nba_pg)
```

```
##                      player games
## 1             Steven Adams    58
## 2              Bam Adebayo    65
## 3        LaMarcus Aldridge    53
## 4 Nickeil Alexander-Walker    41
## 5            Grayson Allen    30
## 6            Jarrett Allen    64
```

To select all the columns except a specific column, use the - (subtraction) operator. For example,

```
head(select(nba, -player))
```

```
##   position age team games games_started minutes_played field_goals
## 1        C  26  OKC    58            58           1564         262
## 2       PF  22  MIA    65            65           2235         408
## 3        C  34  SAS    53            53           1754         391
## 4       SG  21  NOP    41             0            501          77
## 5       SG  24  MEM    30             0            498          79
## 6        C  21  BRK    64            58           1647         267
##   field_goal_attempts three_pointers three_point_attempts two_pointers
## 1                 443              1                     3          261
## 2                 719              1                    13          407
## 3                 793             61                   157          330
## 4                 227             40                   117           37
## 5                 176             33                    91           46
## 6                 413              0                     5          267
##   two_point_attempts free_throws free_throw_attempts offensive_rebounds
## 1                440         108                183                196
## 2                706         236                342                165
## 3                636         158                191                103
## 4                110          17                 28                  8
## 5                 85          30                 35                  5
## 6                408         147                237                195
##   defensive_rebounds assists steals blocks turnovers personal_fouls
## 1                347     141     50     65        86            111
## 2                518     333     78     85       185            164
## 3                289     129     36     87        74            128
## 4                 72      74     11      7        40             46
## 5                 61      43      6      1        23             36
## 6                410      85     37     85        72            144
```

To select a range of columns by name, use the : (colon) operator. For example,

```
head(select(nba, player:games))
```

```
##                     player position age team games
## 1             Steven Adams        C  26  OKC    58
## 2              Bam Adebayo       PF  22  MIA    65
## 3        LaMarcus Aldridge        C  34  SAS    53
## 4 Nickeil Alexander-Walker       SG  21  NOP    41
## 5            Grayson Allen       SG  24  MEM    30
## 6            Jarrett Allen        C  21  BRK    64
```

To select all columns that start with certain character strings, use the function starts_with(). Ohter matching options are:

1. ends_with() = Select columns that end with a character string
2. contains() = Select columns that contain a character string
3. matches() = Select columns that match a regular expression
4. one_of() = Select columns names that are from a group of names

```
head(select(nba, starts_with("three")))
```

```
##   three_pointers three_point_attempts
## 1              1                    3
## 2              1                   13
## 3             61                  157
## 4             40                  117
## 5             33                   91
## 6              0                    5
```

```
head(select(nba, contains("throw")))
```

```
##   free_throws free_throw_attempts
## 1         108                183
## 2         236                342
## 3         158                191
## 4          17                 28
## 5          30                 35
## 6         147                237
```

Selecting rows using filter()

We can also select the rows that satisfied certain criteria. For example, if I want to select the rows that the number of assists is larger than 500.

```
filter(nba, assists > 500)
```

```
##         player position age team games games_started minutes_played field_goals
## 1 LeBron James       PG  35  LAL    60            60           2094         586
## 2  Ricky Rubio       PG  29  PHO    57            57           1802         252
## 3   Trae Young       PG  21  ATL    60            60           2120         546
##   field_goal_attempts three_pointers three_point_attempts two_pointers
## 1                1176            133                  381          453
## 2                 611             66                  188          186
## 3                1249            205                  568          341
##   two_point_attempts free_throws free_throw_attempts offensive_rebounds
## 1                795         239                343                 59
```

```
## 2                423         174                 204                  42
## 3                681         481                 559                  32
##   defensive_rebounds assists steals blocks turnovers personal_fouls
## 1                414     636     74     30       239            106
## 2                223     507     88      9       155            143
## 3                223     560     65      8       289            104
```

We can also filter mutiple criteria.

```
filter(nba, age > 30, team %in% c("HOU", "GSW"))
```

```
##               player position age team games games_started minutes_played
## 1      Ryan Anderson       PF  31  HOU     2             0             14
## 2    DeMarre Carroll       SF  33  HOU     6             0             96
## 3     Tyson Chandler        C  37  HOU    26             5            219
## 4      Stephen Curry       PG  31  GSW     5             5            139
## 5        Eric Gordon       SG  31  HOU    34            13            972
## 6         Jeff Green       PF  33  HOU    10             0            201
## 7       Jeremy Pargo       PG  33  GSW     3             0             44
## 8    Thabo Sefolosha       PF  35  HOU    41             0            436
## 9        P.J. Tucker       PF  34  HOU    64            64           2203
## 10 Russell Westbrook       PG  31  HOU    53            53           1905
##    field_goals field_goal_attempts three_pointers three_point_attempts
## 1            2                  7              1                    5
## 2           12                 25              4                   14
## 3           14                 18              0                    0
## 4           33                 82             12                   49
## 5          162                438             92                  288
## 6           41                 66             14                   34
## 7           11                 22              3                    7
## 8           37                 91             15                   54
## 9          162                374             95                  257
## 10         568               1199             51                  201
##    two_pointers two_point_attempts free_throws free_throw_attempts
## 1             1                  2          0                   0
## 2             8                 11         12                  16
## 3            14                 18          6                  13
## 4            21                 33         26                  26
## 5            70                150         78                 103
## 6            27                 32          8                  10
## 7             8                 15          0                   1
## 8            22                 37          3                   8
## 9            67                117         37                  44
## 10          517                998        269                 346
##    offensive_rebounds defensive_rebounds assists steals blocks turnovers
## 1                   0                 7       2      1      0         1
## 2                   4                12       7      2      2         6
## 3                  25                41       6      6      8         8
## 4                   4                22      33      5      2        16
## 5                   8                57      52     22     12        36
## 6                   9                23      12      9      5         9
## 7                   1                 2       8      1      0         4
## 8                  21                72      25     24     12        15
## 9                 102               339     101     70     31        67
## 10                 94               332     370     88     17       236
```

```
##    personal_fouls
## 1               1
## 2               8
## 3              32
## 4              11
## 5              68
## 6              19
## 7               8
## 8              43
## 9             209
## 10            187
```

Pipe operator: %>%

Before we go any futher, let's introduce the pipe operator: `%>%`. This operator allows you to pipe the output from one function to the input of another function. For example, the following two statements will generate the same output:

```
head(select(nba, player, games))
```

```
##                     player games
## 1             Steven Adams    58
## 2              Bam Adebayo    65
## 3        LaMarcus Aldridge    53
## 4 Nickeil Alexander-Walker    41
## 5            Grayson Allen    30
## 6            Jarrett Allen    64
```

```
nba %>% select(player, games) %>% head
```

```
##                     player games
## 1             Steven Adams    58
## 2              Bam Adebayo    65
## 3        LaMarcus Aldridge    53
## 4 Nickeil Alexander-Walker    41
## 5            Grayson Allen    30
## 6            Jarrett Allen    64
```

The reason why we introduce the pipe operator is to easily combine multiple functions without using too many parentheses or name the variables in each step.

Arrange or re-order rows using arrange()

To arrange the data frame by a specific order we need to use the function arrange(). The default is by increasing order and a negaive operator will provide the decreasing order.

```
head(arrange(nba, personal_fouls))
```

```
##                 player position age team games games_started minutes_played
## 1 Kostas Antetokounmpo       PF  22  LAL     3             0              5
## 2         Troy Daniels       SG  28  DEN     1             0              1
## 3          Jacob Evans       SG  22  MIN     2             0              4
## 4             Kyle Guy       SG  22  SAC     2             0              4
## 5         Jared Harper       PG  22  PHO     3             0              8
## 6  Talen Horton-Tucker       SG  19  LAL     2             0              5
##   field_goals field_goal_attempts three_pointers three_point_attempts
## 1           0                   0              0                    0
```

```
## 2           0                   1              0                    1
## 3           0                   1              0                    1
## 4           1                   2              0                    1
## 5           1                   4              0                    2
## 6           0                   1              0                    1
##   two_pointers two_point_attempts free_throws free_throw_attempts
## 1            0                   0          0                    0
## 2            0                   0          0                    0
## 3            0                   0          0                    0
## 4            1                   1          0                    0
## 5            1                   2          0                    0
## 6            0                   0          0                    0
##   offensive_rebounds defensive_rebounds assists steals blocks turnovers
## 1                  0                  1       1      0      0         0
## 2                  0                  0       0      0      0         0
## 3                  0                  0       0      0      0         0
## 4                  0                  0       0      0      0         0
## 5                  0                  0       0      0      0         0
## 6                  0                  0       2      1      0         0
##   personal_fouls
## 1              0
## 2              0
## 3              0
## 4              0
## 5              0
## 6              0
```

```
head(arrange(nba, -personal_fouls))
```

```
##              player position age team games games_started minutes_played
## 1     Dillon Brooks       SG  24  MEM    65            65           1851
## 2 Jaren Jackson Jr.        C  20  MEM    54            54           1512
## 3       P.J. Tucker       PF  34  HOU    64            64           2203
## 4      James Harden       SG  30  HOU    61            61           2241
## 5      Daniel Theis        C  27  BOS    58            57           1382
## 6     Dwight Howard        C  34  LAL    62             2           1193
##   field_goals field_goal_attempts three_pointers three_point_attempts
## 1         372                925            132                  358
## 2         328                701            135                  340
## 3         162                374             95                  257
## 4         603               1386            271                  769
## 5         216                382             26                   81
## 6         188                257              3                    5
##   two_pointers two_point_attempts free_throws free_throw_attempts
## 1          240                567         144                178
## 2          193                361         123                166
## 3           67                117          37                 44
## 4          332                617         619                719
## 5          190                301          81                106
## 6          185                252          89                180
##   offensive_rebounds defensive_rebounds assists steals blocks turnovers
## 1                 61                152     128     57     25       108
## 2                 53                201      76     36     87        94
## 3                102                339     101     70     31        67
## 4                 64                324     450    106     53       273
```

```
## 5               130               250      95     35     75        48
## 6               156               301      42     27     76        73
##   personal_fouls
## 1            246
## 2            219
## 3            209
## 4            206
## 5            201
## 6            200
```

It can also combine with the select() and filter(), with the pipe operator.

```
nba %>% select(player, team, age, games) %>% arrange(-age) %>% filter(games > 50) %>% head
```

```
##              player team age games
## 1      Vince Carter  ATL  43    60
## 2      LeBron James  LAL  35    60
## 3       J.J. Redick  NOP  35    54
## 4 LaMarcus Aldridge  SAS  34    53
## 5        Taj Gibson  NYK  34    62
## 6     Dwight Howard  LAL  34    62
```

Create new columns using mutate()

Sometimes the data does not provide the variable that we are interested in directly and in that case, we can manipulate the current variables and add new variables into the data frame. For example, if I am interested in number of fouls per game,

```
nba %>% mutate(foul_per_game = personal_fouls / games) %>% head
```

```
##                     player position age team games games_started minutes_played
## 1             Steven Adams        C  26  OKC    58            58           1564
## 2              Bam Adebayo       PF  22  MIA    65            65           2235
## 3        LaMarcus Aldridge        C  34  SAS    53            53           1754
## 4 Nickeil Alexander-Walker       SG  21  NOP    41             0            501
## 5            Grayson Allen       SG  24  MEM    30             0            498
## 6            Jarrett Allen        C  21  BRK    64            58           1647
##   field_goals field_goal_attempts three_pointers three_point_attempts
## 1         262                443              1                    3
## 2         408                719              1                   13
## 3         391                793             61                  157
## 4          77                227             40                  117
## 5          79                176             33                   91
## 6         267                413              0                    5
##   two_pointers two_point_attempts free_throws free_throw_attempts
## 1          261                440         108                183
## 2          407                706         236                342
## 3          330                636         158                191
## 4           37                110          17                 28
## 5           46                 85          30                 35
## 6          267                408         147                237
##   offensive_rebounds defensive_rebounds assists steals blocks turnovers
## 1                196                347     141     50     65        86
## 2                165                518     333     78     85       185
## 3                103                289     129     36     87        74
## 4                  8                 72      74     11      7        40
```

```
## 5                  5                 61      43      6      1        23
## 6                195                410      85     37     85        72
##   personal_fouls foul_per_game
## 1            111      1.913793
## 2            164      2.523077
## 3            128      2.415094
## 4             46      1.121951
## 5             36      1.200000
## 6            144      2.250000
```

Create summaries of the data frame using summarise()

TO create summary statistics for a given column in the data frame, we can use summarise() function. For example, if I need to extract the mean, min, max number of assists,

```
nba %>% summarise(avg_assist = mean(assists),
                  min_assist = min(assists),
                  max_assist = max(assists))
```

```
##   avg_assist min_assist max_assist
## 1   82.95088          0        636
```

The advantage of summarise is more obvious if we combine it with the group_by() , the group operators. Since players at the different position tend to have very different statistics, we can group the players by their poisition first and then generate summary statistics. For example,

```
nba %>% group_by(position) %>% summarise(avg_assist = mean(assists),
                                         min_assist = min(assists),
                                         max_assist = max(assists))
```

```
## # A tibble: 5 x 4
##   position avg_assist min_assist max_assist
##   <chr>         <dbl>      <int>      <int>
## 1 C              62.4          0        446
## 2 PF             56.5          0        333
## 3 PG            165.           0        636
## 4 SF             60.8          0        340
## 5 SG             77.5          0        450
```

Exercise

1. Read the MLB data frame and show the first 2 row and the last 4 row.

```
#R code here
mlb <- dplyr::as_tibble(read.csv("http://www.stat.cmu.edu/cmsac/sure/materials/data/intro_r/mlb_teams_da
print(head(mlb, 2))
```

```
## # A tibble: 2 x 22
##    year league team_id team_name win_world_series final_rank games_played  wins
##   <int> <chr>  <chr>   <chr>     <chr>                 <int>        <int> <int>
## 1  1871 <NA>   BS1     Boston R~ <NA>                      3           31    20
## 2  1871 <NA>   CH1     Chicago ~ <NA>                      2           28    19
## # ... with 14 more variables: losses <int>, runs_scored <int>, hits <int>,
## #   at_bats <int>, walks <int>, strikeouts <int>, homeruns <int>,
## #   hit_by_pitch <int>, sacrifice_flies <int>, runs_allowed <int>,
## #   hits_allowed <int>, walks_allowed <int>, strikeouts_against <int>,
## #   homeruns_allowed <int>
```

```r
print(tail(mlb, 4))
```

```
## # A tibble: 4 x 22
##    year league team_id team_name win_world_series final_rank games_played  wins
##   <int> <chr>  <chr>   <chr>     <chr>                 <int>        <int> <int>
## 1  2018 AL     TBA     Tampa Ba~ N                         3          162    90
## 2  2018 AL     TEX     Texas Ra~ N                         5          162    67
## 3  2018 AL     TOR     Toronto ~ N                         4          162    73
## 4  2018 NL     WAS     Washingt~ N                         2          162    82
## # ... with 14 more variables: losses <int>, runs_scored <int>, hits <int>,
## #   at_bats <int>, walks <int>, strikeouts <int>, homeruns <int>,
## #   hit_by_pitch <int>, sacrifice_flies <int>, runs_allowed <int>,
## #   hits_allowed <int>, walks_allowed <int>, strikeouts_against <int>,
## #   homeruns_allowed <int>
```

```r
# location of the dataset: http://www.stat.cmu.edu/cmsac/sure/materials/data/intro_r/mlb_teams_data.csv
```

2. Find out the dimension of MLB data frame. Discuss the difference between summary() and str()

```r
dim(mlb)
```

```
## [1] 2895   22
```

`summary()` shows the summary statistics for each column in the data frame while `str()` shows the data type of each column and its first few observations.

3. Select all the columns related to the game results in MLB data frame.

```r
mlb %>% select(c(wins, losses))
```

```
## # A tibble: 2,895 x 2
##     wins losses
##    <int>  <int>
##  1    20     10
##  2    19      9
##  3    10     19
##  4     7     12
##  5    16     17
##  6    21      7
##  7     4     21
##  8    13     15
##  9    15     15
## 10    35     19
## # ... with 2,885 more rows
```

4. Select the columns end with the string "allowed" and show the head.

```r
mlb %>% select(ends_with("allowed")) %>% head()
```

```
## # A tibble: 6 x 4
##   runs_allowed hits_allowed walks_allowed homeruns_allowed
##          <int>        <int>         <int>            <int>
## 1          303          367            42                2
## 2          241          308            28                6
## 3          341          346            53               13
## 4          243          261            21                5
## 5          313          373            42                7
## 6          266          329            53                3
```

5.(option) Compare following statements and discuss the difference. Hint: read the help document of "regex".

```
head(select(mlb, contains("runs")))
head(select(mlb, matches("runs")))

head(select(mlb, contains("runs.*")))
head(select(mlb, matches("runs.*")))
```

`contains()` returns the column names that literally contain the parameter string. `matches()` is able to take wildcards (.*). That's why `contains()` returns nothing when passed a string with a wildcard while `matches()` returned 6 rows.

6. Select the rows that wins more than 100 times in year 2018.

```
mlb %>% filter(year == 2018, wins > 100)
```

```
## # A tibble: 2 x 22
##    year league team_id team_name win_world_series final_rank games_played  wins
##   <int> <chr>  <chr>   <chr>     <chr>                 <int>        <int> <int>
## 1  2018 AL     BOS     Boston R~ Y                         1          162   108
## 2  2018 AL     HOU     Houston ~ N                         1          162   103
## # ... with 14 more variables: losses <int>, runs_scored <int>, hits <int>,
## #   at_bats <int>, walks <int>, strikeouts <int>, homeruns <int>,
## #   hit_by_pitch <int>, sacrifice_flies <int>, runs_allowed <int>,
## #   hits_allowed <int>, walks_allowed <int>, strikeouts_against <int>,
## #   homeruns_allowed <int>
```

7. Use the pipe operator to select the teams that win world series after(include) year 2010. Only present the year, the team name and the number of wins.

```
mlb %>% filter(year >= 2010, win_world_series == "Y") %>% select(c(year, team_name, wins))
```

```
## # A tibble: 9 x 3
##    year team_name             wins
##   <int> <chr>                <int>
## 1  2010 San Francisco Giants    92
## 2  2011 St. Louis Cardinals     90
## 3  2012 San Francisco Giants    94
## 4  2013 Boston Red Sox          97
## 5  2014 San Francisco Giants    88
## 6  2015 Kansas City Royals      95
## 7  2016 Chicago Cubs           103
## 8  2017 Houston Astros         101
## 9  2018 Boston Red Sox         108
```

8. Use the pipe operator to select the teams that win world series after(include) year 2010. Only present the year, the team name and the number of wins. Re-arrange the data frame with the number of wins in a decreasing order.

```
mlb %>% filter(year >= 2010, win_world_series == "Y") %>% select(c(year, team_name, wins)) %>% arrange(
```

```
## # A tibble: 9 x 3
##    year team_name             wins
##   <int> <chr>                <int>
## 1  2018 Boston Red Sox         108
## 2  2016 Chicago Cubs           103
## 3  2017 Houston Astros         101
## 4  2013 Boston Red Sox          97
```

```
## 5  2015 Kansas City Royals      95
## 6  2012 San Francisco Giants    94
## 7  2010 San Francisco Giants    92
## 8  2011 St. Louis Cardinals     90
## 9  2014 San Francisco Giants    88
```

9. Use the pipe operator to add two new columns in the MLB data frame which represents the percentage of wins over the numebr of games_played and the number of homeruns per games_played.

```
mlb %>% mutate(p_wins = wins/games_played,
               p_hr = homeruns/games_played)
```

```
## # A tibble: 2,895 x 24
##     year league team_id team_name win_world_series final_rank games_played  wins
##    <int> <chr>  <chr>   <chr>     <chr>                 <int>        <int> <int>
##  1  1871 <NA>   BS1     Boston R~ <NA>                      3           31    20
##  2  1871 <NA>   CH1     Chicago ~ <NA>                      2           28    19
##  3  1871 <NA>   CL1     Clevelan~ <NA>                      8           29    10
##  4  1871 <NA>   FW1     Fort Way~ <NA>                      7           19     7
##  5  1871 <NA>   NY2     New York~ <NA>                      5           33    16
##  6  1871 <NA>   PH1     Philadel~ <NA>                      1           28    21
##  7  1871 <NA>   RC1     Rockford~ <NA>                      9           25     4
##  8  1871 <NA>   TRO     Troy Hay~ <NA>                      6           29    13
##  9  1871 <NA>   WS3     Washingt~ <NA>                      4           32    15
## 10  1872 <NA>   BL1     Baltimor~ <NA>                      2           58    35
## # ... with 2,885 more rows, and 16 more variables: losses <int>,
## #   runs_scored <int>, hits <int>, at_bats <int>, walks <int>,
## #   strikeouts <int>, homeruns <int>, hit_by_pitch <int>,
## #   sacrifice_flies <int>, runs_allowed <int>, hits_allowed <int>,
## #   walks_allowed <int>, strikeouts_against <int>, homeruns_allowed <int>,
## #   p_wins <dbl>, p_hr <dbl>
```

10. Use the pipe operator and group operator to generate summary statistics (mean, min, max) for the the final rank, for each team.

```
mlb %>% group_by(team_name) %>%
  summarise(mean_final_rank = mean(final_rank),
            min_final_rank = min(final_rank),
            max_final_rank = max(final_rank)) %>%
  ungroup()
```

```
## # A tibble: 139 x 4
##    team_name             mean_final_rank min_final_rank max_final_rank
##    <chr>                           <dbl>          <int>          <int>
##  1 Altoona Mountain City           10                10             10
##  2 Anaheim Angels                   2.5               1              4
##  3 Arizona Diamondbacks             2.90              1              5
##  4 Atlanta Braves                   3.13              1              7
##  5 Baltimore Canaries               4.33              2              8
##  6 Baltimore Marylands              9                 9              9
##  7 Baltimore Monumentals            4                 4              4
##  8 Baltimore Orioles                3.92              1             12
##  9 Baltimore Terrapins              5.5               3              8
## 10 Boston Americans                 3.71              1              8
## # ... with 129 more rows
```

11. Explore the missing values. In MLB data frame, which columns contain mising value "NA"? Is there

any pattern for the missing value?

```
mlb %>% filter(anyNA(c(league, win_world_series, hit_by_pitch, sacrifice_flies))) %>%
  filter(is.na(hit_by_pitch), !is.na(sacrifice_flies))
```

```
## # A tibble: 0 x 22
## # ... with 22 variables: year <int>, league <chr>, team_id <chr>,
## #   team_name <chr>, win_world_series <chr>, final_rank <int>,
## #   games_played <int>, wins <int>, losses <int>, runs_scored <int>,
## #   hits <int>, at_bats <int>, walks <int>, strikeouts <int>, homeruns <int>,
## #   hit_by_pitch <int>, sacrifice_flies <int>, runs_allowed <int>,
## #   hits_allowed <int>, walks_allowed <int>, strikeouts_against <int>,
## #   homeruns_allowed <int>
```

If there is no win_world_series, then league must also be blank. Likewise, if hit_by_pitch is blank, then so is sacrifice_flies.