Data Wrangling

June 7, 2023

Read and preview data

Our data are usually presented as a csv file and after loading a csv file into R studio, we will have a "data frame". A data frame can be considered a special case of matrix where each column represents a measurement or variable of interest for each observation which correspond to the rows of the dataset. After loading the `tidyverse` suite of packages, we use the `read_csv()` function to load the NBA stats dataset (SPORTS) or the heart disease dataset (HEALTH) from the other day:

```
library(tidyverse)

# SPORTS
nba_stats <- read_csv("./files/nba_stats.csv")

# HEALTH
heart_disease <- read_csv("./files/heart_disease.csv")
```

By default, `read_csv()` reads in the dataset as a `tbl` (aka `tibble`) object instead of a `data.frame` object. You can read about the differences here, but it's not that meaningful for purposes.

We can use the functions `head()` and `tail()` to view a sample of the data. Use the `head()` function to view the first 6 rows, then use the `tail()` function to view the last 3 rows:

```
# INSERT CODE HERE

head(nba_stats, n = 6)
```

```
## # A tibble: 6 x 22
##   player     position   age team  games games_started minutes_played field_goals
##   <chr>      <chr>    <dbl> <chr> <dbl>         <dbl>          <dbl>       <dbl>
## 1 Precious ~ C           22 TOR      73            28           1725         265
## 2 Steven Ad~ C           28 MEM      76            75           1999         210
## 3 Bam Adeba~ C           24 MIA      56            56           1825         406
## 4 Santi Ald~ PF          21 MEM      32             0            360          53
## 5 LaMarcus ~ C           36 BRK      47            12           1050         252
## 6 Nickeil A~ SG          23 NOP      50            19           1317         237
## # i 14 more variables: field_goal_attempts <dbl>, three_pointers <dbl>,
## #   three_point_attempts <dbl>, two_pointers <dbl>, two_point_attempts <dbl>,
## #   free_throws <dbl>, free_throw_attempts <dbl>, offensive_rebounds <dbl>,
## #   defensive_rebounds <dbl>, assists <dbl>, steals <dbl>, blocks <dbl>,
## #   turnovers <dbl>, personal_fouls <dbl>
```

```
tail(nba_stats, n = 3)
```

```
## # A tibble: 3 x 22
##   player     position   age team  games games_started minutes_played field_goals
##   <chr>      <chr>    <dbl> <chr> <dbl>         <dbl>          <dbl>       <dbl>
## 1 Omer Yurt~ C           23 MIA      56            12            706         130
## 2 Cody Zell~ C           29 POR      27             0            355          51
```

```
## 3 Ivica Zubac C           24 LAC      76            76           1852         310
## # i 14 more variables: field_goal_attempts <dbl>, three_pointers <dbl>,
## #   three_point_attempts <dbl>, two_pointers <dbl>, two_point_attempts <dbl>,
## #   free_throws <dbl>, free_throw_attempts <dbl>, offensive_rebounds <dbl>,
## #   defensive_rebounds <dbl>, assists <dbl>, steals <dbl>, blocks <dbl>,
## #   turnovers <dbl>, personal_fouls <dbl>
```

```
head(heart_disease, n = 6)
```

```
## # A tibble: 6 x 10
##     Cost   Age Gender Interventions Drugs ERVisit Complications Comorbidities
##    <dbl> <dbl> <chr>          <dbl> <dbl>   <dbl>         <dbl>         <dbl>
## 1   179.    63 Female             2     1       4             0             3
## 2   319     59 Female             2     0       6             0             0
## 3  9311.    62 Female            17     0       2             0             5
## 4   281.    60 Male               9     0       7             0             2
## 5 18727.    55 Female             5     2       7             0             0
## 6   453.    66 Female             1     0       3             0             4
## # i 2 more variables: Duration <dbl>, id <dbl>
```

```
tail(heart_disease, n = 3)
```

```
## # A tibble: 3 x 10
##    Cost   Age Gender Interventions Drugs ERVisit Complications Comorbidities
##   <dbl> <dbl> <chr>          <dbl> <dbl>   <dbl>         <dbl>         <dbl>
## 1 2678.    68 Female             3     2       6             0            10
## 2 1282.    58 Female             7     2       2             0             7
## 3  586     56 Female             4     2       6             0             3
## # i 2 more variables: Duration <dbl>, id <dbl>
```

View the dimensions of the data with `dim()`:

```
# INSERT CODE HERE
dim(nba_stats)
```

```
## [1] 715  22
```

Quickly view summary statistics for all variables with the `summary()` function:

```
# Uncomment the following code by deleting the # at the front:
summary(nba_stats)
```

```
##     player            position              age            team
##  Length:715         Length:715         Min.   :19.00   Length:715
##  Class :character   Class :character   1st Qu.:23.00   Class :character
##  Mode  :character   Mode  :character   Median :25.00   Mode  :character
##                                        Mean   :25.94
##                                        3rd Qu.:29.00
##                                        Max.   :41.00
##      games       games_started  minutes_played    field_goals
##  Min.   : 1.00   Min.   : 0.0   Min.   :   1.0   Min.   :  0.0
##  1st Qu.:11.00   1st Qu.: 0.0   1st Qu.: 115.5   1st Qu.: 13.0
##  Median :36.00   Median : 4.0   Median : 566.0   Median : 82.0
##  Mean   :36.42   Mean   :17.2   Mean   : 830.4   Mean   :139.8
##  3rd Qu.:62.00   3rd Qu.:26.5   3rd Qu.:1432.0   3rd Qu.:219.0
##  Max.   :82.00   Max.   :82.0   Max.   :2854.0   Max.   :774.0
##  field_goal_attempts three_pointers   three_point_attempts  two_pointers
##  Min.   :   0.0      Min.   :  0.00   Min.   :  0          Min.   :  0.00
```

```
##  1st Qu.:  33.5      1st Qu.:  1.00   1st Qu.:  6          1st Qu.:  7.00
##  Median : 183.0      Median : 17.00   Median : 53          Median : 52.00
##  Mean   : 303.1      Mean   : 42.79   Mean   :121          Mean   : 96.97
##  3rd Qu.: 474.5      3rd Qu.: 63.00   3rd Qu.:189          3rd Qu.:142.00
##  Max.   :1564.0      Max.   :285.00   Max.   :750          Max.   :724.00
##  two_point_attempts  free_throws     free_throw_attempts offensive_rebounds
##  Min.   :   0.0     Min.   :  0.00   Min.   :  0.00      Min.   :  0.00
##  1st Qu.:  16.5     1st Qu.:  4.00   1st Qu.:  6.00      1st Qu.:  4.00
##  Median : 101.0     Median : 28.00   Median : 37.00      Median : 20.00
##  Mean   : 182.1     Mean   : 58.26   Mean   : 75.22      Mean   : 35.56
##  3rd Qu.: 265.0     3rd Qu.: 79.00   3rd Qu.:103.00      3rd Qu.: 46.00
##  Max.   :1393.0     Max.   :654.00   Max.   :803.00      Max.   :349.00
##  defensive_rebounds    assists          steals           blocks
##  Min.   :  0.0      Min.   :  0.00   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 14.0      1st Qu.:  6.50   1st Qu.:  3.00   1st Qu.:  1.00
##  Median : 73.0      Median : 41.00   Median : 18.00   Median :  8.00
##  Mean   :117.4      Mean   : 84.81   Mean   : 26.25   Mean   : 16.22
##  3rd Qu.:178.0      3rd Qu.:114.00   3rd Qu.: 42.00   3rd Qu.: 22.00
##  Max.   :813.0      Max.   :737.00   Max.   :138.00   Max.   :177.00
##    turnovers      personal_fouls
##  Min.   :  0.00   Min.   :  0.00
##  1st Qu.:  5.00   1st Qu.: 11.00
##  Median : 28.00   Median : 52.00
##  Mean   : 44.92   Mean   : 67.56
##  3rd Qu.: 62.50   3rd Qu.:110.50
##  Max.   :303.00   Max.   :286.00
```

```
# summary(heart_disease)
```

View the data structure types with `str()`:

```
str(nba_stats)
```

```
## spc_tbl_ [715 x 22] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ player              : chr [1:715] "Precious Achiuwa" "Steven Adams" "Bam Adebayo" "Santi Aldama"
##  $ position            : chr [1:715] "C" "C" "C" "PF" ...
##  $ age                 : num [1:715] 22 28 24 21 36 23 23 26 23 23 ...
##  $ team                : chr [1:715] "TOR" "MEM" "MIA" "MEM" ...
##  $ games               : num [1:715] 73 76 56 32 47 50 15 66 56 54 ...
##  $ games_started       : num [1:715] 28 75 56 0 12 19 2 61 56 1 ...
##  $ minutes_played      : num [1:715] 1725 1999 1825 360 1050 ...
##  $ field_goals         : num [1:715] 265 210 406 53 252 237 16 255 369 131 ...
##  $ field_goal_attempts : num [1:715] 603 384 729 132 458 632 48 569 545 294 ...
##  $ three_pointers      : num [1:715] 56 0 0 6 14 95 10 159 1 32 ...
##  $ three_point_attempts: num [1:715] 156 1 6 48 46 305 33 389 10 110 ...
##  $ two_pointers        : num [1:715] 209 210 406 47 238 142 6 96 368 99 ...
##  $ two_point_attempts  : num [1:715] 447 383 723 84 412 327 15 180 535 184 ...
##  $ free_throws         : num [1:715] 78 108 256 20 89 70 11 64 165 36 ...
##  $ free_throw_attempts : num [1:715] 131 199 340 32 102 97 12 74 233 53 ...
##  $ offensive_rebounds  : num [1:715] 146 349 137 33 73 36 1 32 192 25 ...
##  $ defensive_rebounds  : num [1:715] 327 411 427 54 185 128 22 190 410 75 ...
##  $ assists             : num [1:715] 82 256 190 21 42 139 17 100 92 152 ...
##  $ steals              : num [1:715] 37 65 80 6 14 41 5 46 44 71 ...
##  $ blocks              : num [1:715] 41 60 44 10 47 19 4 18 75 7 ...
##  $ turnovers           : num [1:715] 84 115 148 16 44 85 8 43 94 40 ...
```

```
##  $ personal_fouls      : num [1:715] 151 153 171 36 78 88 15 96 97 73 ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   player = col_character(),
##   ..   position = col_character(),
##   ..   age = col_double(),
##   ..   team = col_character(),
##   ..   games = col_double(),
##   ..   games_started = col_double(),
##   ..   minutes_played = col_double(),
##   ..   field_goals = col_double(),
##   ..   field_goal_attempts = col_double(),
##   ..   three_pointers = col_double(),
##   ..   three_point_attempts = col_double(),
##   ..   two_pointers = col_double(),
##   ..   two_point_attempts = col_double(),
##   ..   free_throws = col_double(),
##   ..   free_throw_attempts = col_double(),
##   ..   offensive_rebounds = col_double(),
##   ..   defensive_rebounds = col_double(),
##   ..   assists = col_double(),
##   ..   steals = col_double(),
##   ..   blocks = col_double(),
##   ..   turnovers = col_double(),
##   ..   personal_fouls = col_double()
##   .. )
##  - attr(*, "problems")=<externalptr>
```

```
# str(heart_disease)
```

What's the difference between the output from the two functions?

The function `summary()` gives the key descriptive statistics for the numeric (double) variables in the `nba_data` tibble whereas the function `dim()` gives the type of variable that each column vector is in the `nba_data` tibble (e.g. numeric/double, character, etc.) as well as the first few entries in each column vector in the `nba_data` tibble.

Data manipulation with `dplyr`

An easier way to manipulate the data frame is through the `dplyr` package, which is in the `tidyverse` suite of packages. The operations we can do include: selecting specific columns, filtering for rows, re-ordering rows, adding new columns and summarizing data. The "split-apply-combine" concept can be achieved by `dplyr`.

Selecting columns with `select()`

The function `select()` can be use to select certain column with the column names.

(SPORTS) First create a new table called `nba_stats_pg` that only contains the `player` and `games` columns:

```
# INSERT CODE HERE
nba_stats_pg <- nba_stats %>%
  select(player, games)
```

To select all the columns except a specific column, use the `-` (subtraction) operator. For example, view the output from uncommenting the following line of code:

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

```
## # A tibble: 6 x 21
##   position   age team  games games_started minutes_played field_goals
##   <chr>    <dbl> <chr> <dbl>         <dbl>          <dbl>       <dbl>
## 1 C           22 TOR      73            28           1725         265
## 2 C           28 MEM      76            75           1999         210
## 3 C           24 MIA      56            56           1825         406
## 4 PF          21 MEM      32             0            360          53
## 5 C           36 BRK      47            12           1050         252
## 6 SG          23 NOP      50            19           1317         237
## # i 14 more variables: field_goal_attempts <dbl>, three_pointers <dbl>,
## #   three_point_attempts <dbl>, two_pointers <dbl>, two_point_attempts <dbl>,
## #   free_throws <dbl>, free_throw_attempts <dbl>, offensive_rebounds <dbl>,
## #   defensive_rebounds <dbl>, assists <dbl>, steals <dbl>, blocks <dbl>,
## #   turnovers <dbl>, personal_fouls <dbl>
```

To select a range of columns by name (that are in consecutive order), use the `:` (colon) operator. For example, view the output from uncommenting the following line of code:

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

```
## # A tibble: 6 x 5
##   player                   position   age team  games
##   <chr>                    <chr>    <dbl> <chr> <dbl>
## 1 Precious Achiuwa         C           22 TOR      73
## 2 Steven Adams             C           28 MEM      76
## 3 Bam Adebayo              C           24 MIA      56
## 4 Santi Aldama             PF          21 MEM      32
## 5 LaMarcus Aldridge        C           36 BRK      47
## 6 Nickeil Alexander-Walker SG          23 NOP      50
```

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

```
# Uncomment the following lines of code
head(select(nba_stats, starts_with("three")))
```

```
## # A tibble: 6 x 2
##   three_pointers three_point_attempts
##            <dbl>                <dbl>
## 1             56                  156
## 2              0                    1
## 3              0                    6
## 4              6                   48
## 5             14                   46
## 6             95                  305
```

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

```
## # A tibble: 6 x 2
```

```
##   free_throws free_throw_attempts
##         <dbl>               <dbl>
## 1          78                 131
## 2         108                 199
## 3         256                 340
## 4          20                  32
## 5          89                 102
## 6          70                  97
```

(HEALTH) First create a new table called `heart_disease_ad` that only contains the `Age` and `Drugs` columns:

```
# INSERT CODE HERE
```

To select all the columns except a specific column, use the `-` (subtraction) operator. For example, view the output from uncommenting the following line of code:

```
# head(select(heart_disease, -Interventions))
```

To select a range of columns by name (that are in consecutive order), use the `:` (colon) operator. For example, view the output from uncommenting the following line of code:

```
#head(select(heart_disease, Drugs:Duration))
```

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

```
# Uncomment the following lines of code
#head(select(heart_disease, starts_with("Com")))
#head(select(heart_disease, contains("er")))
```

Selecting rows using filter()

(SPORTS) We can also select the rows/observations that satisfy certain criteria. Try selecting the rows with more than 500 assists:

```
# INSERT CODE HERE
nba_stats %>%
  filter(assists > 500)
```

```
## # A tibble: 8 x 22
##   player     position   age team  games games_started minutes_played field_goals
##   <chr>      <chr>    <dbl> <chr> <dbl>         <dbl>          <dbl>       <dbl>
## 1 LaMelo Ba~ PG          20 CHO      75            75           2422         538
## 2 Luka Donč~ PG          22 DAL      65            65           2301         641
## 3 Darius Ga~ PG          22 CLE      68            68           2430         542
## 4 Nikola Jo~ C           26 DEN      74            74           2476         764
## 5 Dejounte ~ PG          25 SAS      68            68           2366         573
## 6 Chris Paul PG          36 PHO      65            65           2139         363
## 7 Russell W~ PG          33 LAL      78            78           2678         548
## 8 Trae Young PG          23 ATL      76            76           2652         711
## # i 14 more variables: field_goal_attempts <dbl>, three_pointers <dbl>,
```

```
## #   three_point_attempts <dbl>, two_pointers <dbl>, two_point_attempts <dbl>,
## #   free_throws <dbl>, free_throw_attempts <dbl>, offensive_rebounds <dbl>,
## #   defensive_rebounds <dbl>, assists <dbl>, steals <dbl>, blocks <dbl>,
## #   turnovers <dbl>, personal_fouls <dbl>
```

We can also filter on mutiple criteria. Select rows with `age` above 30 and the team is either "HOU" or "GSW":

```
# INSERT CODE HERE
nba_stats %>%
  filter(age > 30, team %in% c("HOU", "GSW"))
```

```
## # A tibble: 7 x 22
##   player     position   age team  games games_started minutes_played field_goals
##   <chr>      <chr>    <dbl> <chr> <dbl>         <dbl>          <dbl>       <dbl>
## 1 D.J. Augu~ PG          34 HOU      34             2            510          55
## 2 Nemanja B~ C           33 GSW      71             0           1142         160
## 3 Stephen C~ PG          33 GSW      64            64           2211         535
## 4 Eric Gord~ SG          33 HOU      57            46           1669         268
## 5 Draymond ~ PF          31 GSW      46            44           1329         135
## 6 Andre Igu~ SF          38 GSW      31             0            603          46
## 7 Klay Thom~ SG          31 GSW      32            32            941         246
## # i 14 more variables: field_goal_attempts <dbl>, three_pointers <dbl>,
## #   three_point_attempts <dbl>, two_pointers <dbl>, two_point_attempts <dbl>,
## #   free_throws <dbl>, free_throw_attempts <dbl>, offensive_rebounds <dbl>,
## #   defensive_rebounds <dbl>, assists <dbl>, steals <dbl>, blocks <dbl>,
## #   turnovers <dbl>, personal_fouls <dbl>
```

(HEALTH) We can also select the rows/observations that satisfy certain criteria. Try selecting the rows with more than 500 assists:

```
# INSERT CODE HERE
```

We can also filter on mutiple criteria. Select rows with `Age` above 60 and the `gender` is 'Male':

```
# INSERT CODE HERE
```

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 negative operator will provide the decreasing order.

(SPORTS) First arrange the `nba_stats` table by `personal_fouls` in ascending order:

```
# INSERT CODE HERE
nba_stats %>%
  arrange(personal_fouls) %>%
  select(personal_fouls, everything())
```

```
## # A tibble: 715 x 22
##    personal_fouls player position   age team  games games_started minutes_played
##             <dbl> <chr>  <chr>    <dbl> <chr> <dbl>         <dbl>          <dbl>
##  1              0 Joel ~ SG          21 WAS       7             0             20
##  2              0 Chaun~ SF          23 LAL       2             0             21
##  3              0 Ahmad~ PG          25 IND       1             0              1
##  4              0 Zylan~ SF          26 UTA       1             0              5
##  5              0 Sam D~ PF          27 TOR       1             0              1
##  6              0 Damye~ SG          27 NYK       2             0             21
##  7              0 Jeff ~ PG          24 MIL       1             0              3
```

```
##  8             0 Jaime~ C           24 WAS       1             0             3
##  9             0 Rob E~ SG          25 OKC       2             0            11
## 10             0 James~ SF          31 BRK       2             0            14
## # i 705 more rows
## # i 14 more variables: field_goals <dbl>, field_goal_attempts <dbl>,
## #   three_pointers <dbl>, three_point_attempts <dbl>, two_pointers <dbl>,
## #   two_point_attempts <dbl>, free_throws <dbl>, free_throw_attempts <dbl>,
## #   offensive_rebounds <dbl>, defensive_rebounds <dbl>, assists <dbl>,
## #   steals <dbl>, blocks <dbl>, turnovers <dbl>
```

Next by descending order:

```
# INSERT CODE HERE
nba_stats %>%
  arrange(desc(personal_fouls)) %>%
  select(personal_fouls, everything())
```

```
## # A tibble: 715 x 22
##    personal_fouls player position   age team  games games_started minutes_played
##             <dbl> <chr>  <chr>    <dbl> <chr> <dbl>         <dbl>          <dbl>
##  1            286 Jae'S~ SF          26 HOU      78            77           2056
##  2            272 Jaren~ PF          22 MEM      78            78           2126
##  3            267 Karl-~ C           26 MIN      74            74           2476
##  4            247 Jonas~ C           29 NOP      74            74           2240
##  5            238 Herbe~ PF          23 NOP      78            69           2335
##  6            237 LaMel~ PG          20 CHO      75            75           2422
##  7            235 Russe~ PG          33 LAL      78            78           2678
##  8            225 Pasca~ PF          27 TOR      68            68           2578
##  9            224 Jaden~ PF          21 MIN      70            31           1803
## 10            224 Mason~ C           31 CHO      73            73           1793
## # i 705 more rows
## # i 14 more variables: field_goals <dbl>, field_goal_attempts <dbl>,
## #   three_pointers <dbl>, three_point_attempts <dbl>, two_pointers <dbl>,
## #   two_point_attempts <dbl>, free_throws <dbl>, free_throw_attempts <dbl>,
## #   offensive_rebounds <dbl>, defensive_rebounds <dbl>, assists <dbl>,
## #   steals <dbl>, blocks <dbl>, turnovers <dbl>
```

Try combining a pipeline of `select()`, `filter()`, and `arrange()` steps together with the `%>%` operator by:

1. Selecting the `player`, `team`, `age`, and `games` columns,
2. Filter to select only rows with `games` above 50,
3. Sort by `age` in descending order

```
# INSERT CODE HERE
nba_stats %>%
  select(player, team, age, games) %>%
  filter(games > 50) %>%
  arrange(desc(age))
```

```
## # A tibble: 254 x 4
##    player          team    age games
##    <chr>           <chr> <dbl> <dbl>
##  1 Carmelo Anthony LAL      37    69
##  2 LeBron James    LAL      37    56
##  3 Taj Gibson      NYK      36    52
##  4 Dwight Howard   LAL      36    60
##  5 Chris Paul      PHO      36    65
```

```
##  6 P.J. Tucker     MIA      36    71
##  7 Rudy Gay        UTA      35    55
##  8 Jeff Green      DEN      35    75
##  9 George Hill     MIL      35    54
## 10 Al Horford      BOS      35    69
## # i 244 more rows
```

(HEALTH) First arrange the `heart_disease` table by `Duration` in ascending order:

```
# INSERT CODE HERE
```

Next by descending order:

```
# INSERT CODE HERE
```

Try combining a pipeline of `select()`, `filter()`, and `arrange()` steps together with the `%>%` operator by:

1. Selecting the `Age`, `Cost`, `ERVisit`, and `Duration` columns,
2. Filter to select only rows with `Age` above 60,
3. Sort by `Duration` in descending order

```
# INSERT CODE HERE
```

Create new columns using `mutate()`

Sometimes the data does not include the variable that we are interested in and we need to manipulate the current variables to add new variables into the data frame.

(SPORTS) Create a new column `fouls_per_game` by taking the `personal_fouls` and dividing by `games` (reassign this output to the `nba_stats` table following the commented code chunk so this column is added to the table):

```
nba_stats <- nba_stats %>%
  mutate(fouls_per_game = personal_fouls/games)
```

(HEALTH) Create a new column `cost_per_day` by taking the `Cost` and dividing by `Duration` (reassign this output to the `heart_disease` table following the commented code chunk so this column is added to the table):

```
# heart_disease <- heart_disease %>%
#   mutate(INSERT CODE HERE)
```

Create summaries of the data with `summarize()`

To create summary statistics for a given column in the data frame, we can use `summarize()` function.

(SPORTS) Compute the `mean`, `min`, and `max` number of `assists`:

```
nba_stats %>%
  summarize(mean_assists = mean(assists), min_assists = min(assists), max_assists = max(assists))
```

```
## # A tibble: 1 x 3
##   mean_assists min_assists max_assists
##          <dbl>       <dbl>       <dbl>
## 1         84.8           0         737
```

The advantage of `summarize` 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, first `group_by()` position and then compute the same summary statistics:

```
nba_stats %>%
  group_by(position) %>%
    summarize(mean_assists = mean(assists), min_assists = min(assists), max_assists = max(assists))
```

```
## # A tibble: 5 x 4
##   position mean_assists min_assists max_assists
##   <chr>           <dbl>       <dbl>       <dbl>
## 1 C                64.2           0         584
## 2 PF               67.2           0         388
## 3 PG              150.            0         737
## 4 SF               64.2           0         358
## 5 SG               77.8           0         379
```

(HEALTH) Compute the `mean`, `min`, and `max` number of `Cost`:

```
# INSERT CODE HERE
```

The advantage of `summarize` is more obvious if we combine it with the `group_by()`, the group operators. Try to `group_by()` the `Gender` column first and then compute the same summary statistics:

```
# INSERT CODE HERE
```