Homework 2 - data wrangling with the tidyverse

Runxiang Liu

14/2/2023

Contents

Homework 2 - data wrangling with the `tidyverse`

Which NBA player scored the most points in 1991?

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.2     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
##            Player
## 1 Michael Jordan*
```

Which player had the best free throw percentage from the year 2000 to the most recent year in the data?

```
##       Player
## 1 Drew Barry
```

Rename the variable `pos` to `position`.

```
data %>%
  rename(Position = Pos) %>% head(5)
```

```
##   Year          Player Position Age  Tm  G GS MP PER   TS. X3PAr   FTr ORB.
## 1 1950 Curly Armstrong      G-F  31 FTW 63 NA NA  NA 0.368    NA 0.467   NA
## 2 1950    Cliff Barker       SG  29 INO 49 NA NA  NA 0.435    NA 0.387   NA
## 3 1950   Leo Barnhorst       SF  25 CHS 67 NA NA  NA 0.394    NA 0.259   NA
## 4 1950      Ed Bartels        F  24 TOT 15 NA NA  NA 0.312    NA 0.395   NA
## 5 1950      Ed Bartels        F  24 DNN 13 NA NA  NA 0.308    NA 0.378   NA
##   DRB. TRB. AST. STL. BLK. TOV. USG. blanl  OWS  DWS   WS WS.48 blank2 OBPM
## 1   NA   NA   NA   NA   NA   NA   NA    NA -0.1  3.6  3.5    NA     NA   NA
## 2   NA   NA   NA   NA   NA   NA   NA    NA  1.6  0.6  2.2    NA     NA   NA
## 3   NA   NA   NA   NA   NA   NA   NA    NA  0.9  2.8  3.6    NA     NA   NA
## 4   NA   NA   NA   NA   NA   NA   NA    NA -0.5 -0.1 -0.6    NA     NA   NA
## 5   NA   NA   NA   NA   NA   NA   NA    NA -0.5 -0.1 -0.6    NA     NA   NA
##   DBPM BPM VORP  FG FGA   FG. X3P X3PA X3P. X2P X2PA  X2P.  eFG.  FT FTA   FT.
## 1   NA  NA   NA 144 516 0.279  NA   NA   NA 144  516 0.279 0.279 170 241 0.705
## 2   NA  NA   NA 102 274 0.372  NA   NA   NA 102  274 0.372 0.372  75 106 0.708
## 3   NA  NA   NA 174 499 0.349  NA   NA   NA 174  499 0.349 0.349  90 129 0.698
## 4   NA  NA   NA  22  86 0.256  NA   NA   NA  22   86 0.256 0.256  19  34 0.559
## 5   NA  NA   NA  21  82 0.256  NA   NA   NA  21   82 0.256 0.256  17  31 0.548
##   ORB DRB TRB AST STL BLK TOV  PF PTS
## 1  NA  NA  NA 176  NA  NA  NA 217 458
## 2  NA  NA  NA 109  NA  NA  NA  99 279
## 3  NA  NA  NA 140  NA  NA  NA 192 438
## 4  NA  NA  NA  20  NA  NA  NA  29  63
## 5  NA  NA  NA  20  NA  NA  NA  27  59
```

Use this variable to create two variables that are called `first_position` and `second_position`. Hint: separate by splitting the position variable in two.

```
data$first_position <- str_split(data$position, "-") %>% unlist %>% .[[1]]
data$second_position <- str_split(data$position, "-") %>% unlist %>% .[[2]]
```

Create two new datasets.

a new dataset from the original dataset that includes all data except the age variable (be sure to give this dataset a new name).

```
data %>%
  select(-Age) -> data1
head(data1, 5)
```

```
##   Year          Player Pos  Tm  G GS MP PER   TS. X3PAr   FTr ORB. DRB. TRB.
## 1 1950 Curly Armstrong G-F FTW 63 NA NA  NA 0.368    NA 0.467   NA   NA   NA
## 2 1950    Cliff Barker  SG INO 49 NA NA  NA 0.435    NA 0.387   NA   NA   NA
```

```
## 3 1950   Leo Barnhorst  SF CHS 67 NA NA  NA 0.394    NA 0.259   NA   NA   NA
## 4 1950      Ed Bartels   F TOT 15 NA NA  NA 0.312    NA 0.395   NA   NA   NA
## 5 1950      Ed Bartels   F DNN 13 NA NA  NA 0.308    NA 0.378   NA   NA   NA
##   AST. STL. BLK. TOV. USG. blanl  OWS  DWS   WS WS.48 blank2 OBPM DBPM BPM VORP
## 1   NA   NA   NA   NA   NA    NA -0.1  3.6  3.5    NA      NA   NA   NA  NA   NA
## 2   NA   NA   NA   NA   NA    NA  1.6  0.6  2.2    NA      NA   NA   NA  NA   NA
## 3   NA   NA   NA   NA   NA    NA  0.9  2.8  3.6    NA      NA   NA   NA  NA   NA
## 4   NA   NA   NA   NA   NA    NA -0.5 -0.1 -0.6    NA      NA   NA   NA  NA   NA
## 5   NA   NA   NA   NA   NA    NA -0.5 -0.1 -0.6    NA      NA   NA   NA  NA   NA
##    FG FGA   FG. X3P X3PA X3P. X2P X2PA  X2P.  eFG.  FT FTA   FT. ORB DRB TRB
## 1 144 516 0.279  NA   NA   NA 144  516 0.279 0.279 170 241 0.705  NA  NA  NA
## 2 102 274 0.372  NA   NA   NA 102  274 0.372 0.372  75 106 0.708  NA  NA  NA
## 3 174 499 0.349  NA   NA   NA 174  499 0.349 0.349  90 129 0.698  NA  NA  NA
## 4  22  86 0.256  NA   NA   NA  22   86 0.256 0.256  19  34 0.559  NA  NA  NA
## 5  21  82 0.256  NA   NA   NA  21   82 0.256 0.256  17  31 0.548  NA  NA  NA
##   AST STL BLK TOV  PF PTS
## 1 176  NA  NA  NA 217 458
## 2 109  NA  NA  NA  99 279
## 3 140  NA  NA  NA 192 438
## 4  20  NA  NA  NA  29  63
## 5  20  NA  NA  NA  27  59
```

a new dataset from the original dataset that includes the year, the player name, and age.

```
data %>%
  select(Year, Player, Age) -> data2
head(data2, 5)
```

```
##   Year          Player Age
## 1 1950 Curly Armstrong  31
## 2 1950    Cliff Barker  29
## 3 1950   Leo Barnhorst  25
## 4 1950      Ed Bartels  24
## 5 1950      Ed Bartels  24
```

add a new column to both datasets called mergeid that includes a sequence of numbers beginning with a 1 in the first row of the data and ending with the total number of rows in the last row of the data

```
data1$mergeid <- seq(1,nrow(data1),1)

data2$mergeid <- seq(1,nrow(data2),1)
```

Join the two datasets from question (6) together to recreate the original dataset plus the new merge id.

```
new_data <- merge(data1, data2, by = "mergeid")
head(new_data, 5)
```

```
##   mergeid Year.x        Player.x Pos  Tm  G GS MP PER   TS. X3PAr   FTr ORB.
## 1       1   1950 Curly Armstrong G-F FTW 63 NA NA  NA 0.368    NA 0.467   NA
## 2       2   1950    Cliff Barker  SG INO 49 NA NA  NA 0.435    NA 0.387   NA
## 3       3   1950   Leo Barnhorst  SF CHS 67 NA NA  NA 0.394    NA 0.259   NA
```

```
## 4       4   1950      Ed Bartels   F TOT 15 NA NA  NA 0.312    NA 0.395   NA
## 5       5   1950      Ed Bartels   F DNN 13 NA NA  NA 0.308    NA 0.378   NA
##   DRB. TRB. AST. STL. BLK. TOV. USG. blanl  OWS  DWS   WS WS.48 blank2 OBPM
## 1   NA   NA   NA   NA   NA   NA   NA    NA -0.1  3.6  3.5    NA      NA   NA
## 2   NA   NA   NA   NA   NA   NA   NA    NA  1.6  0.6  2.2    NA      NA   NA
## 3   NA   NA   NA   NA   NA   NA   NA    NA  0.9  2.8  3.6    NA      NA   NA
## 4   NA   NA   NA   NA   NA   NA   NA    NA -0.5 -0.1 -0.6    NA      NA   NA
## 5   NA   NA   NA   NA   NA   NA   NA    NA -0.5 -0.1 -0.6    NA      NA   NA
##   DBPM BPM VORP  FG FGA   FG. X3P X3PA X3P. X2P X2PA  X2P.  eFG.  FT FTA   FT.
## 1   NA  NA   NA 144 516 0.279  NA   NA   NA 144  516 0.279 0.279 170 241 0.705
## 2   NA  NA   NA 102 274 0.372  NA   NA   NA 102  274 0.372 0.372  75 106 0.708
## 3   NA  NA   NA 174 499 0.349  NA   NA   NA 174  499 0.349 0.349  90 129 0.698
## 4   NA  NA   NA  22  86 0.256  NA   NA   NA  22   86 0.256 0.256  19  34 0.559
## 5   NA  NA   NA  21  82 0.256  NA   NA   NA  21   82 0.256 0.256  17  31 0.548
##   ORB DRB TRB AST STL BLK TOV  PF PTS Year.y        Player.y Age
## 1  NA  NA  NA 176  NA  NA  NA 217 458   1950 Curly Armstrong  31
## 2  NA  NA  NA 109  NA  NA  NA  99 279   1950    Cliff Barker  29
## 3  NA  NA  NA 140  NA  NA  NA 192 438   1950   Leo Barnhorst  25
## 4  NA  NA  NA  20  NA  NA  NA  29  63   1950      Ed Bartels  24
## 5  NA  NA  NA  20  NA  NA  NA  27  59   1950      Ed Bartels  24
```

Subset the original dataset to 1995. Group the data by year and team name and then summarize the average number of points per team. Arrange from most to least points.

```
data %>% filter(Year==1995) %>% group_by(Year, Tm) %>% summarize(avg_pts = mean(PTS)) %>% arrange(desc(a
```

```
## `summarise()` has grouped output by 'Year'. You can override using the `.groups` argument.
```

```
head(data3, 5)
```

```
## # A tibble: 5 x 3
## # Groups:   Year [1]
##    Year Tm    avg_pts
##   <int> <chr>   <dbl>
## 1  1995 SEA      647.
## 2  1995 ORL      606.
## 3  1995 PHO      605.
## 4  1995 DAL      604.
## 5  1995 MIL      582.
```

Reshape the data in the previous question into a wide format using the tidyr package. Create a wide dataset that keeps year in a single column, but spreads team names to multiple individual columns with each column delineating points per team in 1995.

```
data3 %>% spread(Tm, avg_pts) -> data4
head(data4, 5)
```

```
## # A tibble: 1 x 29
## # Groups:   Year [1]
##    Year   ATL   BOS   CHH   CHI   CLE   DAL   DEN   DET   GSW   HOU   IND   LAC
##   <int> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1  1995  440.  496.  516.  520.  494.  604.  489.  503.  482.  499.  542.  495.
```

```
## # ... with 16 more variables: LAL <dbl>, MIA <dbl>, MIL <dbl>, MIN <dbl>,
## #   NJN <dbl>, NYK <dbl>, ORL <dbl>, PHI <dbl>, PHO <dbl>, POR <dbl>,
## #   SAC <dbl>, SAS <dbl>, SEA <dbl>, TOT <dbl>, UTA <dbl>, WSB <dbl>
```

Now return the data to a long (tidy) format by moving teams back into a single column and points in a single column.

```
data4 %>% gather(Tm, avg_pts, -Year) -> data5
head(data5, 5)
```

```
## # A tibble: 5 x 3
## # Groups:   Year [1]
##    Year Tm    avg_pts
##   <int> <chr>   <dbl>
## 1  1995 ATL      440.
## 2  1995 BOS      496.
## 3  1995 CHH      516.
## 4  1995 CHI      520.
## 5  1995 CLE      494.
```