Laboratory Exercise Week 4

Ryan Estes Section 006

9/13/17

*Directions*:

* Write your R code inside the code chunks after each question.
* Write your answer comments after the # sign.
* To generate the word document output, click the button Knit and wait for the word document to appear.
* RStudio will prompt you (only once) to install the knitr package.
* Submit your completed laboratory exercise using Blackboard's Turnitin feature. Your Turnitin upload link is found on your Blackboard Course shell under the Laboratory folder.

For this exercise, you will need to use the package mosaic to find numerical and graphical summaries.

library(mosaic) # load the package mosaic to use its functions

## Loading required package: dplyr

##   
## 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

## Loading required package: lattice

## Loading required package: ggformula

## Loading required package: ggplot2

##   
## New to ggformula? Try the tutorials:   
## learnr::run\_tutorial("introduction", package = "ggformula")  
## learnr::run\_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

## Loading required package: Matrix

##   
## The 'mosaic' package masks several functions from core packages in order to add   
## additional features. The original behavior of these functions should not be affected by this.  
##   
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.

##   
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':  
##   
## mean

## The following objects are masked from 'package:dplyr':  
##   
## count, do, tally

## The following objects are masked from 'package:stats':  
##   
## binom.test, cor, cor.test, cov, fivenum, IQR, median,  
## prop.test, quantile, sd, t.test, var

## The following objects are masked from 'package:base':  
##   
## max, mean, min, prod, range, sample, sum

library(dplyr) # load the package dplyr to use its functions  
library(gapminder) # load the package gapminder for question 1

1. Using the gapminder data in the lesson, do the following:
   1. use filter to select all countries with the following arguments:
      1. life expectancy larger than 60 years.
      2. United Kingdom and Vietnam and years greater than 1990.
   2. use arrange and slice to select the countries with the top 15 GDP per capital gdpPercap. Use the pipe %>% operator to string multiple functions.
   3. use mutate to create a new variable called gdpPercap\_lifeExp which is the quotient of gdpPercap and lifeExp and display the output.
   4. use summarise to find the average or mean value of the variable gdpPercap\_lifeExp created in part (iii).
   5. use group\_by to group the countries by continent; and summarise to compute the average life expectancy lifeExp within each continent. Use the pipe %>% operator to string multiple functions.

### Code chunk

# load the necessary packages  
library(mosaic)  
library(dplyr)  
library(gapminder)  
  
# i)  
# a)  
filter(gapminder, lifeExp > 60)

## # A tibble: 877 x 6  
## country continent year lifeExp pop gdpPercap  
## <fctr> <fctr> <int> <dbl> <int> <dbl>  
## 1 Albania Europe 1962 64.820 1728137 2312.889  
## 2 Albania Europe 1967 66.220 1984060 2760.197  
## 3 Albania Europe 1972 67.690 2263554 3313.422  
## 4 Albania Europe 1977 68.930 2509048 3533.004  
## 5 Albania Europe 1982 70.420 2780097 3630.881  
## 6 Albania Europe 1987 72.000 3075321 3738.933  
## 7 Albania Europe 1992 71.581 3326498 2497.438  
## 8 Albania Europe 1997 72.950 3428038 3193.055  
## 9 Albania Europe 2002 75.651 3508512 4604.212  
## 10 Albania Europe 2007 76.423 3600523 5937.030  
## # ... with 867 more rows

# b)  
filter(gapminder, country %in% c("United States", "Vietnam"), year > 1990)

## # A tibble: 8 x 6  
## country continent year lifeExp pop gdpPercap  
## <fctr> <fctr> <int> <dbl> <int> <dbl>  
## 1 United States Americas 1992 76.090 256894189 32003.9322  
## 2 United States Americas 1997 76.810 272911760 35767.4330  
## 3 United States Americas 2002 77.310 287675526 39097.0995  
## 4 United States Americas 2007 78.242 301139947 42951.6531  
## 5 Vietnam Asia 1992 67.662 69940728 989.0231  
## 6 Vietnam Asia 1997 70.672 76048996 1385.8968  
## 7 Vietnam Asia 2002 73.017 80908147 1764.4567  
## 8 Vietnam Asia 2007 74.249 85262356 2441.5764

# ii)  
gapminder %>%  
 arrange(desc(gdpPercap)) %>%  
 slice(1:15)

## # A tibble: 15 x 6  
## country continent year lifeExp pop gdpPercap  
## <fctr> <fctr> <int> <dbl> <int> <dbl>  
## 1 Kuwait Asia 1957 58.033 212846 113523.13  
## 2 Kuwait Asia 1972 67.712 841934 109347.87  
## 3 Kuwait Asia 1952 55.565 160000 108382.35  
## 4 Kuwait Asia 1962 60.470 358266 95458.11  
## 5 Kuwait Asia 1967 64.624 575003 80894.88  
## 6 Kuwait Asia 1977 69.343 1140357 59265.48  
## 7 Norway Europe 2007 80.196 4627926 49357.19  
## 8 Kuwait Asia 2007 77.588 2505559 47306.99  
## 9 Singapore Asia 2007 79.972 4553009 47143.18  
## 10 Norway Europe 2002 79.050 4535591 44683.98  
## 11 United States Americas 2007 78.242 301139947 42951.65  
## 12 Norway Europe 1997 78.320 4405672 41283.16  
## 13 Ireland Europe 2007 78.885 4109086 40676.00  
## 14 Kuwait Asia 1997 76.156 1765345 40300.62  
## 15 Hong Kong, China Asia 2007 82.208 6980412 39724.98

# iii)  
gapminder %>%  
 mutate(gdpPercap\_lifeExp = gdpPercap / lifeExp) %>%  
 select(country, year, gdpPercap\_lifeExp)

## # A tibble: 1,704 x 3  
## country year gdpPercap\_lifeExp  
## <fctr> <int> <dbl>  
## 1 Afghanistan 1952 27.06313  
## 2 Afghanistan 1957 27.06228  
## 3 Afghanistan 1962 26.66190  
## 4 Afghanistan 1967 24.57957  
## 5 Afghanistan 1972 20.50491  
## 6 Afghanistan 1977 20.45146  
## 7 Afghanistan 1982 24.53986  
## 8 Afghanistan 1987 20.88080  
## 9 Afghanistan 1992 15.58145  
## 10 Afghanistan 1997 15.21302  
## # ... with 1,694 more rows

# iv)  
gapminder %>%  
 mutate(gdpPercap\_lifeExp = gdpPercap / lifeExp) %>%  
 summarise(avgGDP\_Expect = mean(gdpPercap\_lifeExp))

## # A tibble: 1 x 1  
## avgGDP\_Expect  
## <dbl>  
## 1 106.08

# v)  
gapminder %>%  
 group\_by(continent) %>%  
 summarise(mean(lifeExp))

## # A tibble: 5 x 2  
## continent `mean(lifeExp)`  
## <fctr> <dbl>  
## 1 Africa 48.86533  
## 2 Americas 64.65874  
## 3 Asia 60.06490  
## 4 Europe 71.90369  
## 5 Oceania 74.32621

# last R code line

1. The data set MLB-TeamBatting-S16.csv contains MLB Team Batting Data for selected variables. Load the data set from the given url using the code below. This data set was obtained from [Baseball Reference](https://www.baseball-reference.com/leagues/MLB/2016-standard-batting.shtml).
   * Tm - Team
   * Lg - League: American League (AL), National League (NL)
   * BatAge - Battersâ average age
   * RPG - Runs Scored Per Game
   * G - Games Played or Pitched
   * AB - At Bats
   * R - Runs Scored/Allowed
   * H - Hits/Hits Allowed
   * HR - Home Runs Hit/Allowed
   * RBI - Runs Batted In
   * SO - Strikeouts
   * BA - Hits/At Bats
   * SH - Sacrifice Hits (Sacrifice Bunts)
   * SF - Sacrifice Flies

* Using the mlb16.data data, do the following:
  1. use filter to select teams with the following arguments:  
     a) Cardinals team STL.  
     b) teams with Hits H more than 1400 last 2016 season.  
     c) team league Lg is National League NL.
  2. use arrange to select teams in decreasing number of home runs HR.
  3. use group\_by to group the teams per league lg; and arrange to display the teams in decreasing number of RBI within each league. Use the pipe %>% operator to string multiple functions.
  4. use group\_by to group the teams per league; and summarise to compute the average RBI within each league. Use the pipe %>% operator to string multiple functions.

### Code chunk

# load the data set  
mlb16.data <- read.csv("https://raw.githubusercontent.com/jpailden/rstatlab/master/data/MLB-TeamBatting-S16.csv")  
str(mlb16.data) # check structure

## 'data.frame': 30 obs. of 14 variables:  
## $ Tm : Factor w/ 30 levels "ARI","ATL","BAL",..: 1 2 3 4 5 6 7 8 9 10 ...  
## $ Lg : Factor w/ 2 levels "AL","NL": 2 2 1 1 2 1 2 1 2 1 ...  
## $ BatAge: num 26.7 28.9 28.4 28.5 27.4 28.3 27.8 28.9 27.8 29.8 ...  
## $ RPG : num 4.64 4.03 4.59 5.42 4.99 4.23 4.42 4.83 5.22 4.66 ...  
## $ G : int 162 161 162 162 162 162 162 161 162 161 ...  
## $ AB : int 5665 5514 5524 5670 5503 5550 5487 5484 5614 5526 ...  
## $ R : int 752 649 744 878 808 686 716 777 845 750 ...  
## $ H : int 1479 1404 1413 1598 1409 1428 1403 1435 1544 1476 ...  
## $ HR : int 190 122 253 208 199 168 164 185 204 211 ...  
## $ RBI : int 709 615 710 836 767 656 678 733 805 719 ...  
## $ SO : int 1427 1240 1324 1160 1339 1285 1284 1246 1330 1303 ...  
## $ BA : num 0.261 0.255 0.256 0.282 0.256 0.257 0.256 0.262 0.275 0.267 ...  
## $ SH : int 43 64 17 8 42 29 58 31 54 17 ...  
## $ SF : int 38 52 36 40 37 44 44 60 34 38 ...

head(mlb16.data) # show first six rows

## Tm Lg BatAge RPG G AB R H HR RBI SO BA SH SF  
## 1 ARI NL 26.7 4.64 162 5665 752 1479 190 709 1427 0.261 43 38  
## 2 ATL NL 28.9 4.03 161 5514 649 1404 122 615 1240 0.255 64 52  
## 3 BAL AL 28.4 4.59 162 5524 744 1413 253 710 1324 0.256 17 36  
## 4 BOS AL 28.5 5.42 162 5670 878 1598 208 836 1160 0.282 8 40  
## 5 CHC NL 27.4 4.99 162 5503 808 1409 199 767 1339 0.256 42 37  
## 6 CHW AL 28.3 4.23 162 5550 686 1428 168 656 1285 0.257 29 44

# i)  
# a)  
filter(mlb16.data, Tm == "STL")

## Tm Lg BatAge RPG G AB R H HR RBI SO BA SH SF  
## 1 STL NL 28.5 4.81 162 5548 779 1415 225 745 1318 0.255 37 41

# b)  
filter(mlb16.data, H > 1400)

## Tm Lg BatAge RPG G AB R H HR RBI SO BA SH SF  
## 1 ARI NL 26.7 4.64 162 5665 752 1479 190 709 1427 0.261 43 38  
## 2 ATL NL 28.9 4.03 161 5514 649 1404 122 615 1240 0.255 64 52  
## 3 BAL AL 28.4 4.59 162 5524 744 1413 253 710 1324 0.256 17 36  
## 4 BOS AL 28.5 5.42 162 5670 878 1598 208 836 1160 0.282 8 40  
## 5 CHC NL 27.4 4.99 162 5503 808 1409 199 767 1339 0.256 42 37  
## 6 CHW AL 28.3 4.23 162 5550 686 1428 168 656 1285 0.257 29 44  
## 7 CIN NL 27.8 4.42 162 5487 716 1403 164 678 1284 0.256 58 44  
## 8 CLE AL 28.9 4.83 161 5484 777 1435 185 733 1246 0.262 31 60  
## 9 COL NL 27.8 5.22 162 5614 845 1544 204 805 1330 0.275 54 34  
## 10 DET AL 29.8 4.66 161 5526 750 1476 211 719 1303 0.267 17 38  
## 11 KCR AL 28.6 4.17 162 5552 675 1450 147 640 1224 0.261 38 34  
## 12 LAA AL 28.5 4.43 162 5431 717 1410 156 686 991 0.260 36 49  
## 13 MIA NL 28.3 4.07 161 5547 655 1460 128 626 1213 0.263 46 38  
## 14 MIN AL 27.0 4.46 162 5618 722 1409 200 690 1426 0.251 27 43  
## 15 PIT NL 28.9 4.50 162 5542 729 1426 153 696 1334 0.257 41 36  
## 16 SEA AL 30.4 4.74 162 5583 768 1446 223 735 1288 0.259 24 41  
## 17 SFG NL 29.2 4.41 162 5565 715 1437 130 675 1107 0.258 42 46  
## 18 STL NL 28.5 4.81 162 5548 779 1415 225 745 1318 0.255 37 41  
## 19 TEX AL 28.4 4.72 162 5525 765 1446 215 746 1220 0.262 18 40  
## 20 WSN NL 28.8 4.71 162 5490 763 1403 203 735 1252 0.256 48 63

# c)  
filter(mlb16.data, Lg == "NL")

## Tm Lg BatAge RPG G AB R H HR RBI SO BA SH SF  
## 1 ARI NL 26.7 4.64 162 5665 752 1479 190 709 1427 0.261 43 38  
## 2 ATL NL 28.9 4.03 161 5514 649 1404 122 615 1240 0.255 64 52  
## 3 CHC NL 27.4 4.99 162 5503 808 1409 199 767 1339 0.256 42 37  
## 4 CIN NL 27.8 4.42 162 5487 716 1403 164 678 1284 0.256 58 44  
## 5 COL NL 27.8 5.22 162 5614 845 1544 204 805 1330 0.275 54 34  
## 6 LAD NL 28.9 4.48 162 5518 725 1376 189 680 1321 0.249 30 32  
## 7 MIA NL 28.3 4.07 161 5547 655 1460 128 626 1213 0.263 46 38  
## 8 MIL NL 27.5 4.14 162 5330 671 1299 194 641 1543 0.244 53 39  
## 9 NYM NL 29.5 4.14 162 5459 671 1342 218 649 1302 0.246 35 41  
## 10 PHI NL 26.9 3.77 162 5434 610 1305 161 574 1376 0.240 46 30  
## 11 PIT NL 28.9 4.50 162 5542 729 1426 153 696 1334 0.257 41 36  
## 12 SDP NL 28.1 4.23 162 5419 686 1275 177 654 1500 0.235 36 36  
## 13 SFG NL 29.2 4.41 162 5565 715 1437 130 675 1107 0.258 42 46  
## 14 STL NL 28.5 4.81 162 5548 779 1415 225 745 1318 0.255 37 41  
## 15 WSN NL 28.8 4.71 162 5490 763 1403 203 735 1252 0.256 48 63

# ii)  
mlb16.data %>%  
 arrange(desc(HR))

## Tm Lg BatAge RPG G AB R H HR RBI SO BA SH SF  
## 1 BAL AL 28.4 4.59 162 5524 744 1413 253 710 1324 0.256 17 36  
## 2 STL NL 28.5 4.81 162 5548 779 1415 225 745 1318 0.255 37 41  
## 3 SEA AL 30.4 4.74 162 5583 768 1446 223 735 1288 0.259 24 41  
## 4 TOR AL 30.0 4.69 162 5479 759 1358 221 728 1362 0.248 26 40  
## 5 NYM NL 29.5 4.14 162 5459 671 1342 218 649 1302 0.246 35 41  
## 6 TBR AL 27.7 4.15 162 5481 672 1333 216 647 1482 0.243 18 28  
## 7 TEX AL 28.4 4.72 162 5525 765 1446 215 746 1220 0.262 18 40  
## 8 DET AL 29.8 4.66 161 5526 750 1476 211 719 1303 0.267 17 38  
## 9 BOS AL 28.5 5.42 162 5670 878 1598 208 836 1160 0.282 8 40  
## 10 COL NL 27.8 5.22 162 5614 845 1544 204 805 1330 0.275 54 34  
## 11 WSN NL 28.8 4.71 162 5490 763 1403 203 735 1252 0.256 48 63  
## 12 MIN AL 27.0 4.46 162 5618 722 1409 200 690 1426 0.251 27 43  
## 13 CHC NL 27.4 4.99 162 5503 808 1409 199 767 1339 0.256 42 37  
## 14 HOU AL 26.6 4.47 162 5545 724 1367 198 689 1452 0.247 27 31  
## 15 MIL NL 27.5 4.14 162 5330 671 1299 194 641 1543 0.244 53 39  
## 16 ARI NL 26.7 4.64 162 5665 752 1479 190 709 1427 0.261 43 38  
## 17 LAD NL 28.9 4.48 162 5518 725 1376 189 680 1321 0.249 30 32  
## 18 CLE AL 28.9 4.83 161 5484 777 1435 185 733 1246 0.262 31 60  
## 19 NYY AL 29.9 4.20 162 5458 680 1378 183 647 1188 0.252 21 49  
## 20 SDP NL 28.1 4.23 162 5419 686 1275 177 654 1500 0.235 36 36  
## 21 OAK AL 28.7 4.03 162 5500 653 1352 169 634 1145 0.246 13 34  
## 22 CHW AL 28.3 4.23 162 5550 686 1428 168 656 1285 0.257 29 44  
## 23 CIN NL 27.8 4.42 162 5487 716 1403 164 678 1284 0.256 58 44  
## 24 PHI NL 26.9 3.77 162 5434 610 1305 161 574 1376 0.240 46 30  
## 25 LAA AL 28.5 4.43 162 5431 717 1410 156 686 991 0.260 36 49  
## 26 PIT NL 28.9 4.50 162 5542 729 1426 153 696 1334 0.257 41 36  
## 27 KCR AL 28.6 4.17 162 5552 675 1450 147 640 1224 0.261 38 34  
## 28 SFG NL 29.2 4.41 162 5565 715 1437 130 675 1107 0.258 42 46  
## 29 MIA NL 28.3 4.07 161 5547 655 1460 128 626 1213 0.263 46 38  
## 30 ATL NL 28.9 4.03 161 5514 649 1404 122 615 1240 0.255 64 52

# iii)  
mlb16.data %>%  
 group\_by(Lg) %>%  
 arrange(desc(RBI))

## # A tibble: 30 x 14  
## # Groups: Lg [2]  
## Tm Lg BatAge RPG G AB R H HR RBI SO  
## <fctr> <fctr> <dbl> <dbl> <int> <int> <int> <int> <int> <int> <int>  
## 1 BOS AL 28.5 5.42 162 5670 878 1598 208 836 1160  
## 2 COL NL 27.8 5.22 162 5614 845 1544 204 805 1330  
## 3 CHC NL 27.4 4.99 162 5503 808 1409 199 767 1339  
## 4 TEX AL 28.4 4.72 162 5525 765 1446 215 746 1220  
## 5 STL NL 28.5 4.81 162 5548 779 1415 225 745 1318  
## 6 SEA AL 30.4 4.74 162 5583 768 1446 223 735 1288  
## 7 WSN NL 28.8 4.71 162 5490 763 1403 203 735 1252  
## 8 CLE AL 28.9 4.83 161 5484 777 1435 185 733 1246  
## 9 TOR AL 30.0 4.69 162 5479 759 1358 221 728 1362  
## 10 DET AL 29.8 4.66 161 5526 750 1476 211 719 1303  
## # ... with 20 more rows, and 3 more variables: BA <dbl>, SH <int>,  
## # SF <int>

# iv)  
mlb16.data %>%  
 group\_by(Lg) %>%  
 summarise(mean(RBI))

## # A tibble: 2 x 2  
## Lg `mean(RBI)`  
## <fctr> <dbl>  
## 1 AL 699.7333  
## 2 NL 683.2667

# last R code line