**Practice with the R package: dplyr**

We are going to explore Sean Lahman’s historical baseball database, which contains complete seasonal records for all players on all Major League Baseball teams going back to 1871. These data are made available in R via the Lahman package.

1. Download the package Lahman. Then load it into R/retrieve it from the library so you can access the data set called Teams.
2. There are 2835 rows and 48 columns in the data set, so don’t try to view the whole thing!!!!! Instead take a quick look at just the first 10 rows. Also determine what type of object Teams is.

dataframe

Ben worked for the New York Mets from 2004 to 2012. We are going to take a look at how well the team performed during those years. We’ll start by asking: How many wins and losses did the Mets have during each of these years?

1. First we need to look at the column names and determine which ones hold the information we’re looking for. In R print a list of column names, then look at the document explaining the variables in the data set.

[1] "yearID" "lgID" "teamID" "franchID" "divID"

[6] "Rank" "G" "Ghome" "W" "L"

[11] "DivWin" "WCWin" "LgWin" "WSWin" "R"

[16] "AB" "H" "X2B" "X3B" "HR"

[21] "BB" "SO" "SB" "CS" "HBP"

[26] "SF" "RA" "ER" "ERA" "CG"

[31] "SHO" "SV" "IPouts" "HA" "HRA"

[36] "BBA" "SOA" "E" "DP" "FP"

[41] "name" "park" "attendance" "BPF" "PPF"

[46] "teamIDBR" "teamIDlahman45" "teamIDretro"

1. What are the names for the columns that show the year, team name, number of wins, and number of losses?

yearID, teamID, W, L

1. Use the filter() and select() commands to quickly create a subset containing only this data for the years that Ben worked for the Mets.

Note: NYN is the abbreviation used for the team. It stands for New York National League Club.

1. Do this using the filter and select commands separately.See guide below.

ben.subset<-filter(Teams, teamID=="NYN",yearID>=2004,yearID<=2012)

ben.subset<-select(ben.subset,yearID,teamID,W,L)

1. Do this in one line, either nesting the commands or piping the commands together.

ben.subset1<-Teams %>%

filter(teamID=="NYN",yearID>=2004,yearID<=2012) %>%

select(yearID,teamID,W,L)

Guide for part (a): Filter the rows of the Teams data frame so that you only have the rows that correspond to the New York Mets. There are 54 of those, since the Mets joined the National League in 1962.

Next, filter these data so as to include only those seasons in which Ben worked for the team—those with yearID between 2004 and 2012.

Finally, select only those columns that were relevant to our question.

We’ve answered the simple question of how the Mets performed during the time that Ben was there, but since we are data scientists, we are interested in deeper questions. For example, some of these seasons were subpar—the Mets had more losses than wins. Did the team just get unlucky in those seasons? Or did they actually play as badly as their record indicates?

In order to answer this question, we need a model for the expected number of wins per season (or expected win percentage). It turns out that one of the most widely used contributions to the field of baseball analytics (courtesy of Bill James) is exactly that!

The simplest version of this model is: 🡨 expected win percentage

1. What does RA and R stand for? Look at the document explaining the variables in the data set.

RA stands for opponents run scored (runs allowed)

R stands for runs scored

1. Create a new subset of data containing everything the previous one did, plus RA and R.

ben.subset2<-Teams %>%

filter(teamID=="NYN",yearID>=2004,yearID<=2012) %>%

select(yearID,teamID,W,L,R,RA)

1. Compute the *actual* win percentage, call it WPct. This will be the number of wins divided by the total number of games: W/(W+L). Use the mutate command to add WPct to your data frame.

ben.subset3<-Teams %>%

filter(teamID=="NYN",yearID>=2004,yearID<=2012) %>%

select(yearID,teamID,W,L,R,RA) %>%

mutate(WPct = W/(W+L))

1. Now compute the *expected* win percentage (formula given above), call it E\_WPct, and add it to your data frame.

ben.subset4<-Teams %>%

filter(teamID=="NYN",yearID>=2004,yearID<=2012) %>%

select(yearID,teamID,W,L,R,RA) %>%

mutate(WPct = W/(W+L),E\_WPct = 1/(1+(RA/R)^2))

1. In how many seasons did the Mets perform as expected or better? Which seasons were these? Use filter() to print just these rows of data.

filter(ben.subset4, WPct>=E\_WPct)

yearID teamID W L R RA WPct E\_WPct

1 2006 NYN 97 65 834 731 0.5987654 0.5655308

2 2007 NYN 88 74 804 750 0.5432099 0.5347071

3 2012 NYN 74 88 650 709 0.4567901 0.4566674

1. Ok, so which seasons were worst? We can simply sort the rows of the data frame using arrange(). But first, we need to define what is meant by “worst.”

* 1. Define worst as having the lowest *actual* win percentage.

arrange(ben.subset4, WPct)

yearID teamID W L R RA WPct E\_WPct

1 2009 NYN 70 92 671 757 0.4320988 0.4399936

2 2004 NYN 71 91 684 731 0.4382716 0.4668211

3 2012 NYN 74 88 650 709 0.4567901 0.4566674

4 2011 NYN 77 85 718 742 0.4753086 0.4835661

5 2010 NYN 79 83 656 652 0.4876543 0.5030581

6 2005 NYN 83 79 722 648 0.5123457 0.5538575

7 2007 NYN 88 74 804 750 0.5432099 0.5347071

8 2008 NYN 89 73 799 715 0.5493827 0.5553119

9 2006 NYN 97 65 834 731 0.5987654 0.5655308

* 1. Now let’s sort by how much worse they did than expected, i.e. looking at the difference

between actual win percent and expected win percent. (Negative values would indicate that they did worse than expected, while positive values would indicate that they did better than expected.)

arrange(ben.subset4, WPct-E\_WPct)

yearID teamID W L R RA WPct E\_WPct

1 2005 NYN 83 79 722 648 0.5123457 0.5538575

2 2004 NYN 71 91 684 731 0.4382716 0.4668211

3 2010 NYN 79 83 656 652 0.4876543 0.5030581

4 2011 NYN 77 85 718 742 0.4753086 0.4835661

5 2009 NYN 70 92 671 757 0.4320988 0.4399936

6 2008 NYN 89 73 799 715 0.5493827 0.5553119

7 2012 NYN 74 88 650 709 0.4567901 0.4566674

8 2007 NYN 88 74 804 750 0.5432099 0.5347071

9 2006 NYN 97 65 834 731 0.5987654 0.5655308

1. If we want we can add this difference to the data frame, too! Add it and call it diff.

ben.subset5<-Teams %>%

filter(teamID=="NYN",yearID>=2004,yearID<=2012) %>%

select(yearID,teamID,W,L,R,RA) %>%

mutate(WPct = W/(W+L),E\_WPct = 1/(1+(RA/R)^2),diff=WPct-E\_WPct)

1. Let’s also add what diff equates to in terms of the number of games. Determine this by multiplying diff by total number of games. Call this new variable Game\_diff.

ben.subset6<-Teams %>%

filter(teamID=="NYN",yearID>=2004,yearID<=2012) %>%

select(yearID,teamID,W,L,R,RA) %>%

mutate(WPct = W/(W+L),E\_WPct = 1/(1+(RA/R)^2),diff=WPct-E\_WPct,Game\_diff=(W+L)\*diff)

1. Now that we have two diffs, should be more specific with the first one we created. Rename the

diff column pct\_diff using the rename() command.

ben.subset7<-Teams %>%

filter(teamID=="NYN",yearID>=2004,yearID<=2012) %>%

select(yearID,teamID,W,L,R,RA) %>%

mutate(WPct = W/(W+L),E\_WPct = 1/(1+(RA/R)^2),diff=WPct-E\_WPct,

Game\_diff=(W+L)\*diff) %>%

rename(pct\_diff=diff)

You can see that 2006 was the Mets’ most fortunate year—since they won five more games than our model predicts—but 2005 was the least fortunate—since they won almost seven games fewer than our model predicts.

This type of analysis helps us understand how the Mets performed in individual seasons, but we know that any randomness that occurs in individual years is likely to average out over time. So while it is clear that the Mets performed well in some seasons and poorly in others, what can we say about their overall performance?

1. Let’s use summarize() to find the Met’s average *actual* win percentage.

summarize(ben.subset7, avg\_WPct=mean(WPct))

avg\_WPct

1 0.4993141

1. In addition to average actual win percentage, also use summarize to count the number or seasons we are looking at, and the total number of wins and losses during these seasons.

summarize(ben.subset7, seasons=n(),totalWins=sum(W),totalLosses=sum(L),avg\_WPct=mean(WPct))

seasons totalWins totalLosses avg\_WPct

1 9 728 730 0.4993141

Usually, when we are summarizing a data frame, like we did above, it is more interesting to consider different groups. In this case, we can discretize these years into three chunks: one for each of the three general managers under whom Ben worked. Jim Duquette was the Mets’ general manager in 2004, Omar Minaya from 2005 to 2010, and Sandy Alderson from 2011 to 2012.

1. Add a column to your data frame that specifies the manager for each year using the given code:

Name of your data frame %>% mutate(gm = ifelse(yearID==2004,"Duquette",

ifelse(yearID>2010, "Alderson","Minaya")))

ben.subset8<-Teams %>%

filter(teamID=="NYN",yearID>=2004,yearID<=2012) %>%

select(yearID,teamID,W,L,R,RA) %>%

mutate(WPct = W/(W+L),E\_WPct = 1/(1+(RA/R)^2),diff=WPct-E\_WPct,

Game\_diff=(W+L)\*diff) %>%

rename(pct\_diff=diff) %>%

mutate(gm = ifelse(yearID==2004,"Duquette",ifelse(yearID>2010, "Alderson","Minaya")))

1. Look at the details for creating the general manager (gm) variable:

gm=ifelse(yearID==2004,"Duquette", ifelse(yearID>2010, "Alderson","Minaya")))

Write a few statements explaining what this code has done.

Its two if else statements with one nested in the other. First, if the year is 2004, then the manager is Duquette. Then the else of the first statement triggers the second if else. So now if its not 2004, then we move on to if the year is greater than 2010 which would make the manager for those years Alderson. Now the second else happens which basically means all other cases (years not 2004 and less than 2010) are Minaya.

1. Now compute the same summaries as in #16, but use group\_by() to compare based on who the general manager was at the time.

group\_by(ben.subset8,gm) %>%

summarize(seasons=n(),totalWins=sum(W),totalLosses=sum(L),avg\_WPct=mean(WPct))

gm seasons totalWins totalLosses avg\_WPct

<chr> <int> <int> <int> <dbl>

1 Alderson 2 151 173 0.466

2 Duquette 1 71 91 0.438

3 Minaya 6 506 466 0.521

1. Our summaries don’t all make sense now that we’re making comparisons. This is because the general managers each worked for a different number of seasons! List at least 3 summaries that would make sense to compare across general managers.

Avg win percent, avg expected win pct, avg wins, avg losses, avg game diff

1. Compute the summaries that you listed in #20. Also include the number of seasons that each general manager worked.

group\_by(ben.subset8,gm) %>%

summarize(seasons=n(),avg\_Wins=mean(W),avg\_Losses=mean(L),avg\_WPct=mean(WPct),

avg\_E\_WPct=mean(E\_WPct),avg\_Game\_diff=mean(Game\_diff))

gm seasons avg\_Wins avg\_Losses avg\_WPct avg\_E\_WPct avg\_Game\_diff

<chr> <int> <dbl> <dbl> <dbl> <dbl> <dbl>

1 Alderson 2 75.5 86.5 0.466 0.470 -0.659

2 Duquette 1 71 91 0.438 0.467 -4.63

3 Minaya 6 84.3 77.7 0.521 0.525 -0.783

1. So who was the best general manager? Write a few statements based on your summaries to back up your claim.

To me, Minaya was the better gm because he averaged more wins and less losses over a larger span of time. His average win percentage and average estimated win percentage was also higher than the other three gms. Minaya’s avg game diff is lower than Alderson, but with such a small sample size for Alderson and that being his only statistic better than Minaya, its hard to make a case for him. Also their value in that category is pretty close while other stats are not. Because of this Minaya was the best of the three gms.

**Challenge!**

Complete all of your work in one long command. The full power of the chaining/piping operator is revealed when we do all the analysis at once. We will still retain the step-by-step logic.

1. You will not do *everything* above*,* but do the following in one long command:
   1. Specify you want the columns: yearID, teamID, W, L, R, RA – but only for the Mets during the years that Ben worked with them.
   2. Add columns for: WPct, E\_WPct, gm, and a new variable: *expected* number of games the Mets should win (E\_W).
   3. Summarize the data you have by computing the number of seasons per gm, and the average for the following variables per gm: W, E\_W, WPct
   4. Arrange the summarized results so that the highest average WPct is in the top row.

ben.challenge<-Teams %>%

filter(teamID=="NYN",yearID>=2004,yearID<=2012) %>%

select(yearID,teamID,W,L,R,RA) %>%

mutate(WPct = W/(W+L),E\_WPct = 1/(1+(RA/R)^2),E\_W=(E\_WPct\*(W+L))) %>%

mutate(gm = ifelse(yearID==2004,"Duquette",ifelse(yearID>2010, "Alderson","Minaya")))%>%

group\_by(gm) %>%

summarize(seasons=n(),avg\_Wins=mean(W),avg\_E\_W=mean(E\_W),avg\_WPct=mean(WPct))%>%

arrange(-avg\_WPct)

gm seasons avg\_Wins avg\_E\_W avg\_WPct

<chr> <int> <dbl> <dbl> <dbl>

1 Minaya 6 84.3 85.1 0.521

2 Alderson 2 75.5 76.2 0.466

3 Duquette 1 71 75.6 0.438