Lab Tidyverse

Abhi Thanvi, Paul Holaway

June 22nd, 2022

Contents

Lab Tidyverse

Welcome

Just like learning a new spoken language, you will not learn the language without practice. Labs are an important part of this course. Collaboration on labs is **extremely encouraged**. If you find yourself stuck for more than a few minutes, ask a neighbor or course staff for help. When you are giving help to your neighbor, explain the **idea and approach** to the problem without sharing the answer itself so they can figure it out on their own. This will be better for them and for you. For them because it will stick more and they will have a better understanding of the concept. For you because if you can explain it to other students, that means you understand it better too.

Importing Datasets

RStudio allows you to import data sets in a very methodical and simple way.

1. **CSV File**, is a file type that has values separated by commas.

- Each value after a comma in a CSV file maps to a column in the data frame.
- Each row in a CSV file maps to a row in the data frame.
- In our class, we will be mostly using CSV files for our data sets.

2. **Excel Spreadsheets**, is something familiar with and basically represents a data set on a pretty software.

- This is similar to a CSV file in terms that Excel also uses rows and columns to organize the data set.
- We will not be using much of Excel in this course, but it never hurts to be a little familiar with it.

3. **Data Set vs Data Frame?**, this question has a very technical answer, but let's boil it down to something simple.

- Data set is any set of data and it could structured (i.e. rows/column in Excel, JSON, CSV, etc.) or unstructured (i.e. regular text files, emails, etc.). Unstructured data is something we won't deal with, since it is quite tedious and complicated to process.
- Data frame is a structure in which a structured data set could be easily shaped into. Think of it like an actual frame of connect4. The data are like the coins, and the frame is structured in rows and columns that you can put yours coins into.
- Usually, we need to import our data sets and store it into a data frame to analyze and manipulate data for our use-case.

Importing Datasets in RStudio

Importing a data set in RStudio is very simple. In the top right window (a.k.a the Local Environment Window), you can see the drop down titled `Import Dataset`. When you click it, you will see bunch of options of the locations you would like to import your data from. You would be familiar with Excel as a name, but we will be using `From Text (base)...` option as we are importing a CSV file. This should be the first option on the drop down.

Once you choose it, it should open a window to ask you to choose a file. Now all you need to do is find the file we downloaded named `MLB_Batters.csv`.

RStudio will pop up a window asking for Name and bunch of other information. All we need to do is:

1. Change the name to whatever you like.
2. Make Heading -> 'Yes' (it will be a 'No' by default)
3. Check the box that says `String as Factors`

Now you should see that the data set is imported as a data frame on RStudio. It should remind you of an Excel spreadsheet in terms of its table structure. On the top right window, you can the data set being imported with its count of observations. (row count) and count of variables (column count).

If you need any help finding this file you downloaded, please feel free ask for help from your group members or the instructors. Remember we are trying to learn, collaborate and have fun :)

Problem 1

Part 1: Importing Tidyverse

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.7     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
```

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

Part 2: Reading the CSV

In the next cell, copy and past the `RStudio` import code for the `MLB_Batters` data set.

```
MLB_Batters_csv <- read.csv("~/Desktop/DPI 2022/MLB_Batters.csv", stringsAsFactors=TRUE)
```

There may or may not be an import glitch with this data set. If you end up with the first variable being `ï..Player`, copy and paste the line of code below and rerun the cell above.
`MLB_Batters = MLB_Batters %>% rename(Player = "ï..Player")`
In this cell, make sure that you can read the Data frame by printing it. Print out the first ten observations for the sanity of your instructors when they grade this.

```
head(MLB_Batters_csv, 10)
```

```
##             Player Team Pos Age   G  AB   R   H X2B X3B HR RBI SB CS BB  SO SH
## 1  Whit Merrifield   KC  2B  31 162 681 105 206  41  10 16  74 20 10 45 126  0
## 2    Marcus Semien  OAK  SS  29 162 657 123 187  43   7 33  92 10  8 87 102  0
## 3    Rafael Devers  BOS  3B  23 156 647 129 201  54   4 32 115  8  8 48 119  1
## 4  Jonathan Villar  BAL  2B  28 162 642 111 176  33   5 24  73 40  9 61 176  2
## 5     Ozzie Albies  ATL  2B  23 160 640 102 189  43   8 24  86 15  4 54 112  0
## 6  Eduardo Escobar  ARI  2B  31 158 636  94 171  29  10 35 118  5  1 50 130  0
## 7   Starlin Castro  MIA  2B  30 162 636  68 172  31   4 22  86  2  2 28 111  0
## 8       Jose Abreu  CWS  1B  33 159 634  85 180  38   1 33 123  2  2 36 152  0
## 9    Jorge Polanco  MIN  SS  26 153 631 107 186  40   7 22  79  4  3 60 116  2
## 10    Ronald Acuna  ATL  OF  22 156 626 127 175  22   2 41 101 37  9 76 188  0
##    SF HBP   AVG   OBP   SLG   OPS
## 1   4   5 0.302 0.348 0.463 0.811
## 2   1   2 0.285 0.369 0.522 0.891
## 3   2   4 0.311 0.361 0.555 0.916
## 4   4   4 0.274 0.339 0.453 0.792
## 5   4   4 0.295 0.352 0.500 0.852
## 6  10   3 0.269 0.320 0.511 0.831
## 7   9   3 0.270 0.300 0.436 0.736
## 8  10  13 0.284 0.330 0.503 0.833
## 9   7   4 0.295 0.356 0.485 0.841
## 10  1   9 0.280 0.365 0.518 0.883
```

Things to look out for!

As a good data scientist, we always want to have a sense of what the data set or our data frame contains. Before we analyze, its a good practice to have a broad idea of the shape or value our data contains.

Reflection Question

Question: How many rows and columns does our data set have? Type out the answer in the format `m` rows and `n` columns.

```
  171 rows and 23 columns
```

Think About...

What the column names represent and how it might us help answer some questions? Discuss this with your group and write at least three sentences about what your group discussed.

```
  The column names represent different statistics about the batters (for example, their names, age, bat
```

Problem 2

Part 1: Summary

Now that we have imported the data set and have a very general idea of what it looks like. Let's get a more elaborate summary of our data. In the below cell, find the summary of our data set.

```
summary(MLB_Batters_csv)
```

```
##                Player         Team      Pos          Age              G         
##  Aaron Altherr   :  3   SF     : 32   1B: 78   Min.   :21.00   Min.   :  1.00  
##  Corban Joseph   :  3   SEA    : 27   2B:120   1st Qu.:26.00   1st Qu.: 25.00  
##  Keon Broxton    :  3   CLE    : 26   3B: 55   Median :28.00   Median : 69.00  
##  Martin Maldonado:  3   LAA    : 26   C :123   Mean   :28.59   Mean   : 72.27  
##  Travis d'Arnaud :  3   TB     : 26   DH: 12   3rd Qu.:31.00   3rd Qu.:120.00  
##  Tyler Austin    :  3   MIA    : 25   OF:246   Max.   :46.00   Max.   :162.00
##  (Other)         :675   (Other):531   SS: 59
##        AB              R                 H               X2B
##  Min.   :  1.0   Min.   :  0.00   Min.   :  0.00   Min.   : 0.00
##  1st Qu.: 56.0   1st Qu.:  6.00   1st Qu.: 12.00   1st Qu.: 2.00
##  Median :191.0   Median : 25.00   Median : 44.00   Median : 9.00
##  Mean   :233.3   Mean   : 33.43   Mean   : 59.66   Mean   :12.17
##  3rd Qu.:393.0   3rd Qu.: 54.00   3rd Qu.: 99.00   3rd Qu.:20.00
##  Max.   :681.0   Max.   :135.00   Max.   :206.00   Max.   :54.00
##
##       X3B              HR              RBI            SB
##  Min.   : 0.00   Min.   : 0.000   Min.   :  0   Min.   : 0.00
##  1st Qu.: 0.00   1st Qu.: 1.000   1st Qu.:  5   1st Qu.: 0.00
##  Median : 0.00   Median : 6.000   Median : 23   Median : 1.00
##  Mean   : 1.12   Mean   : 9.717   Mean   : 32   Mean   : 3.26
##  3rd Qu.: 2.00   3rd Qu.:15.000   3rd Qu.: 53   3rd Qu.: 4.00
##  Max.   :10.00   Max.   :53.000   Max.   :126   Max.   :46.00
##
##        CS               BB               SO               SH
##  Min.   : 0.000   Min.   :  0.00   Min.   :  0.00   Min.   : 0.0000
##  1st Qu.: 0.000   1st Qu.:  4.00   1st Qu.: 18.00   1st Qu.: 0.0000
##  Median : 0.000   Median : 16.00   Median : 49.00   Median : 0.0000
##  Mean   : 1.189   Mean   : 22.66   Mean   : 58.39   Mean   : 0.4935
##  3rd Qu.: 2.000   3rd Qu.: 35.00   3rd Qu.: 93.00   3rd Qu.: 1.0000
##  Max.   :10.000   Max.   :119.00   Max.   :189.00   Max.   :11.0000
##
##        SF              HBP              AVG              OBP
##  Min.   : 0.000   Min.   : 0.000   Min.   :0.0000   Min.   :0.0000
##  1st Qu.: 0.000   1st Qu.: 0.000   1st Qu.:0.2000   1st Qu.:0.2690
##  Median : 1.000   Median : 2.000   Median :0.2390   Median :0.3120
##  Mean   : 1.642   Mean   : 2.843   Mean   :0.2243   Mean   :0.2931
##  3rd Qu.: 3.000   3rd Qu.: 4.000   3rd Qu.:0.2690   3rd Qu.:0.3420
##  Max.   :12.000   Max.   :27.000   Max.   :0.5000   Max.   :0.6250
##
##       SLG              OPS
##  Min.   :0.0000   Min.   :0.0000
##  1st Qu.:0.3120   1st Qu.:0.5950
##  Median :0.4020   Median :0.7150
##  Mean   :0.3796   Mean   :0.6727
##  3rd Qu.:0.4680   3rd Qu.:0.8040
##  Max.   :0.8330   Max.   :1.2620
##
```

Whoa!!! Now that's a lot of information for Abhi to figure out! He is not good with numbers, can you help him out? Are there any ways we can filter some categories out?

Part 2: Helping Abhi Out...

Abhi usually likes looking at how old people are, don't ask why! Is there anyway we could filter out the Players and Ages from the data set and just show that? (Hint: The answer is yes, we can.)

```
head(MLB_Batters_csv %>% select(c("Player", "Age")), 10)
```

```
##             Player Age
## 1  Whit Merrifield  31
## 2    Marcus Semien  29
## 3    Rafael Devers  23
## 4  Jonathan Villar  28
## 5     Ozzie Albies  23
## 6  Eduardo Escobar  31
## 7    Starlin Castro  30
## 8        Jose Abreu  33
## 9    Jorge Polanco  26
## 10    Ronald Acuna  22
```

Problem 3

Part 1: Abhi has to say something…

"Hey, this is Abhi! Thank you ladies (and gentlemen) for helping me look at the ages of these players. Can you actually just give me 25 people, but also with their Batting Average (I think it's called AVG in the data set).

```
head(MLB_Batters_csv %>% select(c("Player","AVG")), 25)
```

```
##              Player   AVG
## 1   Whit Merrifield 0.302
## 2     Marcus Semien 0.285
## 3     Rafael Devers 0.311
## 4   Jonathan Villar 0.274
## 5      Ozzie Albies 0.295
## 6   Eduardo Escobar 0.269
## 7    Starlin Castro 0.270
## 8        Jose Abreu 0.284
## 9     Jorge Polanco 0.295
## 10     Ronald Acuna 0.280
## 11      Eric Hosmer 0.265
## 12     Amed Rosario 0.287
## 13  Xander Bogaerts 0.309
## 14  Cesar Hernandez 0.279
## 15      DJ LeMahieu 0.327
## 16     Trey Mancini 0.291
## 17     Elvis Andrus 0.275
## 18 Francisco Lindor 0.284
## 19 Paul Goldschmidt 0.260
## 20  Freddie Freeman 0.295
## 21     Mookie Betts 0.295
## 22      Pete Alonso 0.260
## 23   David Fletcher 0.290
## 24     Kevin Pillar 0.264
## 25      Jorge Soler 0.265
```

Part 2: Old is Gold!

We want to recognize some of the experienced players and we just want to see the players above 30 years old. Filter out the people from your last section (Part 3: Abhi has to say something…), we want to see the people above 30 years old. We do not wish to see people that have just celebrated their 30th birthday! Print out the first 25 players.

Hints:

1. This should have more than 50 people in it.
2. You can call this new data: temp_old

```
temp_old = MLB_Batters_csv %>% filter(Age > 30)
head(temp_old, 25)
```

```
##              Player Team Pos Age   G  AB   R   H X2B X3B HR RBI SB CS  BB  SO
## 1   Whit Merrifield   KC  2B  31 162 681 105 206  41  10 16  74 20 10  45 126
## 2   Eduardo Escobar  ARI  2B  31 158 636  94 171  29  10 35 118  5  1  50 130
## 3        Jose Abreu  CWS  1B  33 159 634  85 180  38   1 33 123  2  2  36 152
## 4       DJ LeMahieu  NYY  2B  31 145 602 109 197  33   2 26 102  5  2  46  90
## 5      Elvis Andrus  TEX  SS  31 147 600  81 165  27   4 12  72 31  8  34  96
## 6  Paul Goldschmidt  STL  1B  32 161 597  97 155  25   1 34  97  3  1  78 166
## 7      Kevin Pillar   SF  OF  31 156 595  82 157  37   3 21  87 14  5  18  86
## 8  Charlie Blackmon  COL  OF  33 140 580 112 182  42   7 32  86  2  5  40 104
## 9     J.D. Martinez  BOS  OF  32 146 575  98 175  33   2 36 105  2  0  72 138
## 10 Michael Brantley  HOU  OF  32 148 575  88 179  40   2 22  90  3  2  51  66
## 11   Carlos Santana  CLE  1B  33 158 573 110 161  30   1 34  93  4  0 108 108
## 12       Tommy Pham   TB  OF  32 145 567  77 155  33   2 21  68 25  4  81 123
## 13       Adam Eaton  WAS  OF  31 151 566 103 158  25   7 15  49 15  3  65 106
## 14     Yuli Gurriel  HOU  1B  35 144 564  85 168  40   2 31 104  5  3  37  65
## 15    Shin-Soo Choo  TEX  OF  37 151 563  93 149  31   2 24  61 15  1  78 165
## 16     Lorenzo Cain  MIL  OF  33 148 562  75 146  30   0 11  48 18  8  50 106
## 17      Alex Gordon   KC  OF  36 150 556  77 148  31   1 13  76  5  3  51 100
## 18     Kole Calhoun  LAA  OF  32 152 552  92 128  29   1 33  74  4  1  70 162
## 19   Josh Donaldson  ATL  3B  34 155 549  96 142  33   0 37  94  4  2 100 155
## 20   Starling Marte  PIT  OF  31 132 539  97 159  31   6 23  82 25  6  25  94
## 21     Brandon Belt   SF  1B  31 156 526  76 123  32   3 17  57  4  3  83 127
## 22       Joey Votto  CIN  1B  36 142 525  79 137  32   1 15  47  5  0  76 123
## 23   Mike Moustakas  MIL  2B  31 143 523  80 133  30   1 35  87  3  0  53  98
## 24  Yasmani Grandal  MIL   C  31 153 513  79 126  26   2 28  77  5  1 109 139
## 25     Josh Reddick  HOU  OF  33 141 501  57 138  19   3 14  56  5  2  36  66
##    SH SF HBP   AVG   OBP   SLG   OPS
## 1   0  4   5 0.302 0.348 0.463 0.811
## 2   0 10   3 0.269 0.320 0.511 0.831
## 3   0 10  13 0.284 0.330 0.503 0.833
## 4   1  4   2 0.327 0.375 0.518 0.893
## 5   0 10   4 0.275 0.313 0.393 0.706
## 6   0  3   2 0.260 0.346 0.476 0.822
## 7   0  6   9 0.264 0.293 0.442 0.735
## 8   0  5   9 0.314 0.364 0.576 0.940
## 9   0  5   4 0.304 0.383 0.557 0.940
## 10  0  4   7 0.311 0.372 0.503 0.875
## 11  0  2   3 0.281 0.397 0.515 0.912
## 12  0  1   5 0.273 0.369 0.450 0.819
## 13  9  3  13 0.279 0.365 0.428 0.793
## 14  0  6   5 0.298 0.343 0.541 0.884
## 15  0  1  18 0.265 0.371 0.455 0.826
## 16  0  4   6 0.260 0.325 0.372 0.697
## 17  1  6  19 0.266 0.345 0.396 0.741
## 18  0  2   7 0.232 0.325 0.467 0.792
## 19  0  2   8 0.259 0.379 0.521 0.900
## 20  2  4  16 0.295 0.342 0.503 0.845
## 21  0  4   3 0.234 0.339 0.403 0.742
## 22  0  3   4 0.261 0.357 0.411 0.768
## 23  0  2   6 0.254 0.329 0.516 0.845
## 24  0  5   5 0.246 0.380 0.468 0.848
## 25  1  9   0 0.275 0.319 0.409 0.728
```

Part 3: James the Popular!

We think the most common first name for male in United States is James! You can Google it and let us know! Can you filter the people who are named James from the people you selected being over 30 year old? Again, we know the answer is yes! ;)

Hint: You could make a new variable and do all the filters again or use an already existing variable. We recommend the latter.

```
temp_old %>% filter(Player == "James")
```

```
##  [1] Player Team   Pos    Age    G      AB     R      H      X2B    X3B
## [11] HR     RBI    SB     CS     BB     SO     SH     SF     HBP    AVG
## [21] OBP    SLG    OPS
## <0 rows> (or 0-length row.names)
```

```
# there is no player who's older than 30 and named James
```

Reflection

You would have probably noticed that you did not find James! That is okay, he is not lost…hopefully! The reason is that when you search for "James", our data set contains columns with First and Last Name. Therefore, when you search for only "James" it is not found.

This is very common to happen, our types of data does not match sometimes and we need to find workarounds for it. Sometimes we want to find the age of the people, but they are in string format. Sometimes we want to find if the person is a Male or Female, but the data sets has many variations like M, F, male, female, Boy, Girl, etc. As a Data Scientist, you should prepare to face these challenges that can definitely be tackled.

BONUS:

Can you find a solution to Problem 3 Part 2 (James issue), you are welcome to Google it and collaborate with others. Feel free to ask instructors, but they most likely will ask you to Google it. You are always allowed to view outside sources to to grow as a Data Scientist, even in real world! We are a community helping each other out to find solutions and grow!

Type your solution into the R code chunk to the James problem if you figure it out! :)

```
MLB_Batters_csv %>% filter(substr(Player, 1, 5) == "James")
```

```
##         Player Team Pos Age   G  AB  R   H X2B X3B HR RBI SB CS BB  SO SH SF
## 1 James McCann  CWS   C  29 118 439 62 120  26   1 18  60  4  1 30 137  1  0
##   HBP   AVG   OBP  SLG   OPS
## 1   6 0.273 0.328 0.46 0.788
```

```
# since there's no players named James who are older than 30, we'll extract this info from the original
```

Submission

Once you have finished your lab…

1. Go to the top left and click `File` and `Save`.
2. Click on the `Knit` button to convert this file to a PDF.
3. Submit **BOTH** the `.Rmd` file and `.pdf` file to Blackboard by 11:59 PM tonight.