tidyr

How to create tidy data

Serena Gibbons

I. Introduction

This project will demonstrate tidyr, a package designed for tidying data. Tidy data is defined as data where each variable is in a column, each observation is a row, and each value is a cell. Tidyr is a part of the tidyverse, a collection of R packages used for data science. Tidyr contains functions including gather(), spread(), separate(), and unite(), among others which help to create tidy data, a standard way of storing data. Most datasets however, are not ready to be analyzed, and must be cleaned and manipulated first.

Tidyr is a helpful package because it provides a simple way to manipulate raw data into tidy data, which enables the user to spend more time working on data analysis; this becomes increasingly important with large datasets. A New York Times Article from 2014 stated that data scientists spend from "50 percent to 80 percent of their time mired in this more mundane labor of collecting and preparing unruly digital data, before it can be explored[.]" Since cleaning data is one of the most time consuming and essential parts of the data science process, the use of tidyr can help to simplify this task.

II. Tidy Data

Data tidying is aimed at structuring data in a way to better enable analysis. The principles of tidy data were designed to provide a standard for organizing data within a dataset. Most real world datasets start off as messy datasets and all are messy in different ways, however tidyr makes it easy to tidy data with few tools according to these standards.

Standards of tidy data include: 1. Each variable is a column 2. Each observation a row 3. Each value is a cell and thus there is one type of observational unit per table

III. The Data

The data which will be used in this project is from the dataset file MLB2008.csv, which contains individual offensive statistics from the 2008 Major League Baseball season.

```
mlb<-read.csv("http://www.exploredata.net/ftp/MLB2008.csv")
names(mlb)
```

```
##   [1] "PLAYER"       "Record_ID."   "SALARY"       "ROOKIE"       "POS"          "G"
##   [7] "PA"           "AB"           "H"            "X1B"          "X2B"          "X3B"
##  [13] "HR"           "TB"           "BB"           "UBB"          "IBB"          "HBP"
##  [19] "SF"           "SH"           "ROE"          "RBI"          "LEADOFF_PA"   "DP"
##  [25] "TP"           "WP"           "PB"           "END_GAME"     "SO"           "BK"
##  [31] "INTERFERENCE" "FC"           "TOB"          "OUT"          "SIT_DP"       "GIDP"
##  [37] "PITCHES"      "BALLS"        "STRIKES"      "FB"           "GB"           "LD"
##  [43] "POP"          "Batted.Balls" "PA_P"         "PA_C"         "PA_1B"        "PA_2B"
##  [49] "PA_3B"        "PA_SS"        "PA_LF"        "PA_CF"        "PA_RF"        "PA_DH"
```

```
##  [55] "PA_PH"        "PA_PR"        "G_P"          "G_C"          "G_1B"         "G_2B"
##  [61] "G_3B"         "G_SS"         "G_LF"         "G_CF"         "G_RF"         "G_DH"
##  [67] "G_PH"         "G_PR"         "AVG"          "OBP"          "SLG"          "D_ISO"
##  [73] "TBP"          "BBr"          "UBBr"         "IBBR"         "SO.BB"        "ABR"
##  [79] "HITR"         "B1R"          "B2R"          "B3R"          "HRr"          "HBPR"
##  [85] "SFR"          "SHR"          "ROEr"         "SOr"          "OUTR"         "NSOR"
##  [91] "RBIR"         "LEADOFFR"     "END_GAMER"    "DP."          "FB."          "GB."
##  [97] "LD."          "POP."         "SB"           "CS"           "PICKOFF"      "R"
## [103] "SB."          "RUNR"         "D_BPF"        "PA."          "D_MLVr"       "D_PMLVr"
## [109] "D_RPMLVr"     "D_VORPr"      "D_MLV"        "D_PMLV"       "D_RPMLV"      "D_VORP"
## [115] "D_NETDP"      "D_EqA"        "D_EqR"        "D_RAR"        "D_RAP"        "D_RARP"
## [121] "D_OUTS_EQ"    "PA_ROB"       "R1"           "R2"           "R3"           "R1_BI"
## [127] "R2_BI"        "R3_BI"        "ROB"          "OBI"          "R1BI."        "R2BI."
## [133] "R3BI."        "OBI."
```

```
mlb2<-mlb[ ,1:56]
# the data mlb is truncated in mlb2 to the first 56 columns for ease of demonstration
```

Dataset variables include: . AB (At-bats): Official plate appearances where the batter doesn't walk, get hit by a pitch, hit a recognized sacrifice or is interfered with by the catcher . BB: Hitters: Base on balls (walks); Pitchers: Base on balls (walks) allowed . G: games played . H: hits or hits allowed . PA: plate appearances . PA_P: plate appearances at the position of pitcher . PA_C: plate appearances at the position of catcher . PA_1B: plate appearances at the position of first base . PA_2B: plate appearances at the position of second base . PA_3B: plate appearances at the position of third base . POS: primary position played (position where the most PA were accumulated). Position numbers: 1=pitcher, 2=catcher, 3=first base, 4=second base, 5=third base, 6=shortstop, 7=left field, 8=center field, 9=right field, 10=designated hitter, 11=pinch hitter, 12=pinch runner. The label "0" can apply to a pinch-hitter (or -runner) when the team bats around, coming to that lineup spot for the second time in an inning.

A complete list of variables can be obtained from the glossary of the Baseball Prospectus, where the data was obtained: https://legacy.baseballprospectus.com/glossary/index.php?context=all&category=true

IV. Using tidyr

The package tidyr can be installed into R on its own through install.packages("tidyr"), or alternatively can be obtained by installing the whole tidyverse through install.packages("tidyverse").

```
# install.packages("tidyr")
library(tidyr)
```

V. Spreading and Gathering data

The two main functions of tidyr are spread() and gather(), which can be used to reorganize the values of a table or columns into a new layout.

The gather function takes multiple columns, and gathers them into key-value pairs. It moves column names into a key column, gathering column values into a single value column and making "wide" data longer. It should be used when columns are present that are not variables, which can be collapsed into key-value pairs. The usage of the gather function is as follows: gather(data, key, value, .), where data is a data frame, key is the name of a new key column, value is the name of a new value column, and "." is an optional argument to specify the columns to gather.

```
mlb_gather <- gather(mlb2, plateAppearances, paAmount, c(45:56))
head(mlb_gather)
```

```
##            PLAYER Record_ID.  SALARY ROOKIE POS   G  PA  AB   H X1B X2B X3B HR  TB BB UBB IBB
## 1      Gregg Zaun          1 3750000      0   2  85 288 245  58  40  12   0  6  88 38  37   1
## 2    Henry Blanco          2 3175000      0   2  54 128 120  35  29   3   0  3  47  6   5   1
## 3     Moises Alou          7 7500000      0   7  15  54  49  17  15   2   0  0  19  2   2   0
## 4 Corey Patterson          9 3000000      0   8 123 392 366  75  46  17   2 10 126 16  16   0
## 5     Rod Barajas         10  700000      0   2 100 377 349  87  53  23   0 11 143 17  17   0
## 6    Rich Aurilia         12 4500000      0   3 134 440 407 115  83  21   1 10 168 30  26   4
##   HBP SF SH ROE RBI LEADOFF_PA DP TP WP PB END_GAME SO BK INTERFERENCE FC TOB OUT SIT_DP GIDP
## 1   1  1  3   4  30         63  7  0  1  2        7 38  0            0  0  97 187     51    6
## 2   0  0  2   0  12         27  4  0  0  0        4 22  0            0  0  41  87     33    4
## 3   2  1  0   2   9         12  1  0  0  0        0  4  0            0  0  21  31     12    1
## 4   1  4  5   7  34        120  5  0  2  1        8 57  0            0  3  92 290     74    3
## 5   7  4  0   2  49         70 10  0  3  2        2 61  1            0  0 111 264     82    9
## 6   1  2  0   4  52         91 12  0  1  0        7 56  0            0  2 146 288     78   11
##   PITCHES BALLS STRIKES  FB  GB LD POP Batted.Balls plateAppearances paAmount
## 1    1154   462     692  55  94 34  28          211             PA_P        0
## 2     447   149     298  33  43 17   8          101             PA_P        0
## 3     152    58      94  11  20 11   4           46             PA_P        0
## 4    1299   414     885  85 160 43  30          318             PA_P        0
## 5    1458   465     993 101 109 47  35          292             PA_P        0
## 6    1562   559    1003 108 142 69  34          353             PA_P        0
```

As shown in this example, the various different plate appearances columns (such as PA__1B, PA__2B, PA__3B, etc.) were gathered into a key column of plateApperances which then contained all the column names as variables, and created a value column paAmount.

The spread function takes a key column and a value column and spreads them into multiple columns, making "long" data wider. It moves the key column values into the column names, therefore spreading the values of the value column into new columns. The usage of the spread function is as follows: spread(data, key, value), where data is a data frame, key is the name of a column contain keys, and value is the name of a column containing values.

```
mlb_spread <- spread(mlb_gather, plateAppearances, paAmount)
head(mlb_spread)
```

```
##            PLAYER Record_ID.  SALARY ROOKIE POS   G  PA  AB   H X1B X2B X3B HR  TB BB UBB IBB
## 1      Gregg Zaun          1 3750000      0   2  85 288 245  58  40  12   0  6  88 38  37   1
## 2    Henry Blanco          2 3175000      0   2  54 128 120  35  29   3   0  3  47  6   5   1
## 3     Moises Alou          7 7500000      0   7  15  54  49  17  15   2   0  0  19  2   2   0
## 4 Corey Patterson          9 3000000      0   8 123 392 366  75  46  17   2 10 126 16  16   0
## 5     Rod Barajas         10  700000      0   2 100 377 349  87  53  23   0 11 143 17  17   0
## 6    Rich Aurilia         12 4500000      0   3 134 440 407 115  83  21   1 10 168 30  26   4
##   HBP SF SH ROE RBI LEADOFF_PA DP TP WP PB END_GAME SO BK INTERFERENCE FC TOB OUT SIT_DP GIDP
## 1   1  1  3   4  30         63  7  0  1  2        7 38  0            0  0  97 187     51    6
## 2   0  0  2   0  12         27  4  0  0  0        4 22  0            0  0  41  87     33    4
## 3   2  1  0   2   9         12  1  0  0  0        0  4  0            0  0  21  31     12    1
## 4   1  4  5   7  34        120  5  0  2  1        8 57  0            0  3  92 290     74    3
## 5   7  4  0   2  49         70 10  0  3  2        2 61  1            0  0 111 264     82    9
## 6   1  2  0   4  52         91 12  0  1  0        7 56  0            0  2 146 288     78   11
##   PITCHES BALLS STRIKES  FB  GB LD POP Batted.Balls PA_1B PA_2B PA_3B PA_C PA_CF PA_DH PA_LF
```

```
## 1    1154   462     692  55  94 34  28          211     0     0     0  273     0     8     0
## 2     447   149     298  33  43 17   8          101     1     0     0  109     0     0     0
## 3     152    58      94  11  20 11   4           46     0     0     0    0     0     4    49
## 4    1299   414     885  85 160 43  30          318     0     0     0    0   370     0     0
## 5    1458   465     993 101 109 47  35          292    14     0     0  356     0     5     0
## 6    1562   559    1003 108 142 69  34          353   213     1   195    0     0     8     0
##   PA_P PA_PH PA_PR PA_RF PA_SS
## 1    0     7     0     0     0
## 2    0    18     0     0     0
## 3    0     1     0     0     0
## 4    0    22     0     0     0
## 5    0     2     0     0     0
## 6    0    23     0     0     0
```

Using the spread function as shown here, the "long" data created by the gather function in mlb_gather, was spread back into its original layout.

VI. Separating and Uniting Data

The separate() function separates one column into multiple by separating each cell in a column to make several columns. The usage of the separate function is as follows: separate(data, col, into), where col is the name of the column to separate, and into is the character vector of new column names. It is important to note that while all columns previously have not needed quotations around the names, when using separate() the new columns must be designated by a character vector.

```
mlb_sep <- separate(mlb2, PLAYER, c("firstName", "lastName"), sep = " ")
```

```
## Warning: Expected 2 pieces. Additional pieces discarded in 2 rows [127, 159].
```

```
# Note: a warning message is shown due to 2 players have more than 2 names listed.
head(mlb_sep)
```

```
##   firstName  lastName Record_ID.  SALARY ROOKIE POS   G  PA  AB   H X1B X2B X3B HR  TB BB UBB
## 1     Gregg      Zaun          1 3750000      0   2  85 288 245  58  40  12   0  6  88 38  37
## 2     Henry    Blanco          2 3175000      0   2  54 128 120  35  29   3   0  3  47  6   5
## 3    Moises      Alou          7 7500000      0   7  15  54  49  17  15   2   0  0  19  2   2
## 4     Corey Patterson          9 3000000      0   8 123 392 366  75  46  17   2 10 126 16  16
## 5       Rod   Barajas         10  700000      0   2 100 377 349  87  53  23   0 11 143 17  17
## 6      Rich   Aurilia         12 4500000      0   3 134 440 407 115  83  21   1 10 168 30  26
##   IBB HBP SF SH ROE RBI LEADOFF_PA DP TP WP PB END_GAME SO BK INTERFERENCE FC TOB OUT SIT_DP
## 1   1   1  1  3   4  30         63  7  0  1  2        7 38  0            0  0  97 187     51
## 2   1   0  0  2   0  12         27  4  0  0  0        4 22  0            0  0  41  87     33
## 3   0   2  1  0   2   9         12  1  0  0  0        0  4  0            0  0  21  31     12
## 4   0   1  4  5   7  34        120  5  0  2  1        8 57  0            0  3  92 290     74
## 5   0   7  4  0   2  49         70 10  0  3  2        2 61  1            0  0 111 264     82
## 6   4   1  2  0   4  52         91 12  0  1  0        7 56  0            0  2 146 288     78
##   GIDP PITCHES BALLS STRIKES  FB  GB LD POP Batted.Balls PA_P PA_C PA_1B PA_2B PA_3B PA_SS
## 1    6    1154   462     692  55  94 34  28          211    0  273     0     0     0     0
## 2    4     447   149     298  33  43 17   8          101    0  109     1     0     0     0
## 3    1     152    58      94  11  20 11   4           46    0    0     0     0     0     0
## 4    3    1299   414     885  85 160 43  30          318    0    0     0     0     0     0
## 5    9    1458   465     993 101 109 47  35          292    0  356    14     0     0     0
```

```
## 6   11    1562   559    1003 108 142 69  34          353    0    0   213     1   195     0
##   PA_LF PA_CF PA_RF PA_DH PA_PH PA_PR
## 1     0     0     0     8     7     0
## 2     0     0     0     0    18     0
## 3    49     0     0     4     1     0
## 4     0   370     0     0    22     0
## 5     0     0     0     5     2     0
## 6     0     0     0     8    23     0
```

By default, separate() assumes that the column will be separated on a non-numeric value if the sep argument is not included. Here the space character was specified as the separator in order to separate by the player's first and last name.

The unite() function is the complement of the separate function and can unite multiple columns into one. The usage of the unite function is as follows: unite(data, col, .) where col is the name of the new column to be created and "." is to designate the columns to unite.

```
mlb_unite <- unite(mlb_sep, player, c(firstName, lastName))
head(mlb_unite)
```

```
##            player Record_ID.  SALARY ROOKIE POS   G  PA  AB   H X1B X2B X3B HR  TB BB UBB IBB
## 1      Gregg_Zaun          1 3750000      0   2  85 288 245  58  40  12   0  6  88 38  37   1
## 2    Henry_Blanco          2 3175000      0   2  54 128 120  35  29   3   0  3  47  6   5   1
## 3     Moises_Alou          7 7500000      0   7  15  54  49  17  15   2   0  0  19  2   2   0
## 4 Corey_Patterson          9 3000000      0   8 123 392 366  75  46  17   2 10 126 16  16   0
## 5     Rod_Barajas         10  700000      0   2 100 377 349  87  53  23   0 11 143 17  17   0
## 6    Rich_Aurilia         12 4500000      0   3 134 440 407 115  83  21   1 10 168 30  26   4
##   HBP SF SH ROE RBI LEADOFF_PA DP TP WP PB END_GAME SO BK INTERFERENCE FC TOB OUT SIT_DP GIDP
## 1   1  1  3   4  30         63  7  0  1  2        7 38  0            0  0  97 187     51    6
## 2   0  0  2   0  12         27  4  0  0  0        4 22  0            0  0  41  87     33    4
## 3   2  1  0   2   9         12  1  0  0  0        0  4  0            0  0  21  31     12    1
## 4   1  4  5   7  34        120  5  0  2  1        8 57  0            0  3  92 290     74    3
## 5   7  4  0   2  49         70 10  0  3  2        2 61  1            0  0 111 264     82    9
## 6   1  2  0   4  52         91 12  0  1  0        7 56  0            0  2 146 288     78   11
##   PITCHES BALLS STRIKES  FB  GB LD POP Batted.Balls PA_P PA_C PA_1B PA_2B PA_3B PA_SS PA_LF
## 1    1154   462     692  55  94 34  28          211    0  273     0     0     0     0     0
## 2     447   149     298  33  43 17   8          101    0  109     1     0     0     0     0
## 3     152    58      94  11  20 11   4           46    0    0     0     0     0     0    49
## 4    1299   414     885  85 160 43  30          318    0    0     0     0     0     0     0
## 5    1458   465     993 101 109 47  35          292    0  356    14     0     0     0     0
## 6    1562   559    1003 108 142 69  34          353    0    0   213     1   195     0     0
##   PA_CF PA_RF PA_DH PA_PH PA_PR
## 1     0     0     8     7     0
## 2     0     0     0    18     0
## 3     0     0     4     1     0
## 4   370     0     0    22     0
## 5     0     0     5     2     0
## 6     0     0     8    23     0
```

By default, the new column values will be united by an underscore as seen here. The sep argument can also be used to select how the values will be united, such as by a space or by a dash.

```
mlb_unite2 <- unite(mlb_sep, player, c(firstName, lastName), sep = " ")
head(mlb_unite2)
```

```
##            player Record_ID.  SALARY ROOKIE POS   G  PA  AB   H X1B X2B X3B HR  TB BB UBB IBB
## 1      Gregg Zaun          1 3750000      0   2  85 288 245  58  40  12   0  6  88 38  37   1
## 2    Henry Blanco          2 3175000      0   2  54 128 120  35  29   3   0  3  47  6   5   1
## 3     Moises Alou          7 7500000      0   7  15  54  49  17  15   2   0  0  19  2   2   0
## 4 Corey Patterson          9 3000000      0   8 123 392 366  75  46  17   2 10 126 16  16   0
## 5     Rod Barajas         10  700000      0   2 100 377 349  87  53  23   0 11 143 17  17   0
## 6    Rich Aurilia         12 4500000      0   3 134 440 407 115  83  21   1 10 168 30  26   4
##   HBP SF SH ROE RBI LEADOFF_PA DP TP WP PB END_GAME SO BK INTERFERENCE FC TOB OUT SIT_DP GIDP
## 1   1  1  3   4  30         63  7  0  1  2        7 38  0            0  0  97 187     51    6
## 2   0  0  2   0  12         27  4  0  0  0        4 22  0            0  0  41  87     33    4
## 3   2  1  0   2   9         12  1  0  0  0        0  4  0            0  0  21  31     12    1
## 4   1  4  5   7  34        120  5  0  2  1        8 57  0            0  3  92 290     74    3
## 5   7  4  0   2  49         70 10  0  3  2        2 61  1            0  0 111 264     82    9
## 6   1  2  0   4  52         91 12  0  1  0        7 56  0            0  2 146 288     78   11
##   PITCHES BALLS STRIKES  FB  GB LD POP Batted.Balls PA_P PA_C PA_1B PA_2B PA_3B PA_SS PA_LF
## 1    1154   462     692  55  94 34  28          211    0  273     0     0     0     0     0
## 2     447   149     298  33  43 17   8          101    0  109     1     0     0     0     0
## 3     152    58      94  11  20 11   4           46    0    0     0     0     0     0    49
## 4    1299   414     885  85 160 43  30          318    0    0     0     0     0     0     0
## 5    1458   465     993 101 109 47  35          292    0  356    14     0     0     0     0
## 6    1562   559    1003 108 142 69  34          353    0    0   213     1   195     0     0
##   PA_CF PA_RF PA_DH PA_PH PA_PR
## 1     0     0     8     7     0
## 2     0     0     0    18     0
## 3     0     0     4     1     0
## 4   370     0     0    22     0
## 5     0     0     5     2     0
## 6     0     0     8    23     0
```

```
mlb_unite3 <- unite(mlb_sep, player, c(firstName, lastName), sep = " ")
head(mlb_unite3)
```

```
##            player Record_ID.  SALARY ROOKIE POS   G  PA  AB   H X1B X2B X3B HR  TB BB UBB IBB
## 1      Gregg Zaun          1 3750000      0   2  85 288 245  58  40  12   0  6  88 38  37   1
## 2    Henry Blanco          2 3175000      0   2  54 128 120  35  29   3   0  3  47  6   5   1
## 3     Moises Alou          7 7500000      0   7  15  54  49  17  15   2   0  0  19  2   2   0
## 4 Corey Patterson          9 3000000      0   8 123 392 366  75  46  17   2 10 126 16  16   0
## 5     Rod Barajas         10  700000      0   2 100 377 349  87  53  23   0 11 143 17  17   0
## 6    Rich Aurilia         12 4500000      0   3 134 440 407 115  83  21   1 10 168 30  26   4
##   HBP SF SH ROE RBI LEADOFF_PA DP TP WP PB END_GAME SO BK INTERFERENCE FC TOB OUT SIT_DP GIDP
## 1   1  1  3   4  30         63  7  0  1  2        7 38  0            0  0  97 187     51    6
## 2   0  0  2   0  12         27  4  0  0  0        4 22  0            0  0  41  87     33    4
## 3   2  1  0   2   9         12  1  0  0  0        0  4  0            0  0  21  31     12    1
## 4   1  4  5   7  34        120  5  0  2  1        8 57  0            0  3  92 290     74    3
## 5   7  4  0   2  49         70 10  0  3  2        2 61  1            0  0 111 264     82    9
## 6   1  2  0   4  52         91 12  0  1  0        7 56  0            0  2 146 288     78   11
##   PITCHES BALLS STRIKES  FB  GB LD POP Batted.Balls PA_P PA_C PA_1B PA_2B PA_3B PA_SS PA_LF
## 1    1154   462     692  55  94 34  28          211    0  273     0     0     0     0     0
## 2     447   149     298  33  43 17   8          101    0  109     1     0     0     0     0
## 3     152    58      94  11  20 11   4           46    0    0     0     0     0     0    49
```

```
## 4    1299   414     885  85 160 43  30          318    0    0     0     0     0     0     0
## 5    1458   465     993 101 109 47  35          292    0  356    14     0     0     0     0
## 6    1562   559    1003 108 142 69  34          353    0    0   213     1   195     0     0
##   PA_CF PA_RF PA_DH PA_PH PA_PR
## 1     0     0     8     7     0
## 2     0     0     0    18     0
## 3     0     0     4     1     0
## 4   370     0     0    22     0
## 5     0     0     5     2     0
## 6     0     0     8    23     0
```

References:

Easily Tidy Data with 'spread()' and 'gather()' Functions. (n.d.). Retrieved from https://tidyr.tidyverse.org/

Lohr, S. (2017, December 20). For Big-Data Scientists, 'Janitor Work' Is Key Hurdle to Insights. Retrieved from https://www.nytimes.com/2014/08/18/technology/for-big-data-scientists-hurdle-to-insights-is-janitor-work.html

Minicy Catom Software Engineering Ltd. (n.d.). Baseball Data Set. Retrieved from MINE: Maximal Information-based Nonparametric Exploration http://www.exploredata.net/Downloads/Baseball-Data-Set

Wickham, H. (2014). Tidy Data. Journal of Statistical Software, 59(10). doi:10.18637/jss.v059.i10. Retrieved from http://vita.had.co.nz/papers/tidy-data.html.