Project1

Joyce Aldrich

2022-09-24

R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

Loading the txt file into R

```
data <- read.table("/Users/joycealdrich/Documents/SPS Data Science/Data 607/Project_1/tournamentinfo.tx
                   sep="+")
```

removing the row of "—-"

```
library(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
```

```
data <- data %>%
         filter(V1 != "-----------------------------------------------------------------------------------------
```

Create row indicator

```
row_odd <- seq_len(nrow(data)) %% 2
row_odd
```

```
##   [1] 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
##  [38] 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
##  [75] 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1
## [112] 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
```

#subset odd rows

```
data_row_odd <- data[row_odd == 1, ]
data_row_odd
```

```
##  [1] " Pair | Player Name                    |Total|Round|Round|Round|Round|Round|Round|Round| "
##  [2] "    1 | GARY HUA                       |6.0  |W  39|W  21|W  18|W  14|W   7|D  12|D   4|"
##  [3] "    2 | DAKSHESH DARURI                |6.0  |W  63|W  58|L   4|W  17|W  16|W  20|W   7|"
##  [4] "    3 | ADITYA BAJAJ                   |6.0  |L   8|W  61|W  25|W  21|W  11|W  13|W  12|"
##  [5] "    4 | PATRICK H SCHILLING            |5.5  |W  23|D  28|W   2|W  26|D   5|W  19|D   1|"
##  [6] "    5 | HANSHI ZUO                     |5.5  |W  45|W  37|D  12|D  13|D   4|W  14|W  17|"
##  [7] "    6 | HANSEN SONG                    |5.0  |W  34|D  29|L  11|W  35|D  10|W  27|W  21|"
##  [8] "    7 | GARY DEE SWATHELL              |5.0  |W  57|W  46|W  13|W  11|L   1|W   9|L   2|"
##  [9] "    8 | EZEKIEL HOUGHTON               |5.0  |W   3|W  32|L  14|L   9|W  47|W  28|W  19|"
## [10] "    9 | STEFANO LEE                    |5.0  |W  25|L  18|W  59|W   8|W  26|L   7|W  20|"
## [11] "   10 | ANVIT RAO                      |5.0  |D  16|L  19|W  55|W  31|D   6|W  25|W  18|"
## [12] "   11 | CAMERON WILLIAM MC LEMAN       |4.5  |D  38|W  56|W   6|L   7|L   3|W  34|W  26|"
## [13] "   12 | KENNETH J TACK                 |4.5  |W  42|W  33|D   5|W  38|H    |D   1|L   3|"
## [14] "   13 | TORRANCE HENRY JR              |4.5  |W  36|W  27|L   7|D   5|W  33|L   3|W  32|"
## [15] "   14 | BRADLEY SHAW                   |4.5  |W  54|W  44|W   8|L   1|D  27|L   5|W  31|"
## [16] "   15 | ZACHARY JAMES HOUGHTON         |4.5  |D  19|L  16|W  30|L  22|W  54|W  33|W  38|"
## [17] "   16 | MIKE NIKITIN                   |4.0  |D  10|W  15|H    |W  39|L   2|W  36|U    |"
## [18] "   17 | RONALD GRZEGORCZYK             |4.0  |W  48|W  41|L  26|L   2|W  23|W  22|L   5|"
## [19] "   18 | DAVID SUNDEEN                  |4.0  |W  47|W   9|L   1|W  32|L  19|W  38|L  10|"
## [20] "   19 | DIPANKAR ROY                   |4.0  |D  15|W  10|W  52|D  28|W  18|L   4|L   8|"
## [21] "   20 | JASON ZHENG                    |4.0  |L  40|W  49|W  23|W  41|W  28|L   2|L   9|"
## [22] "   21 | DINH DANG BUI                  |4.0  |W  43|L   1|W  47|L   3|W  40|W  39|L   6|"
## [23] "   22 | EUGENE L MCCLURE               |4.0  |W  64|D  52|L  28|W  15|H    |L  17|W  40|"
## [24] "   23 | ALAN BUI                       |4.0  |L   4|W  43|L  20|W  58|L  17|W  37|W  46|"
## [25] "   24 | MICHAEL R ALDRICH              |4.0  |L  28|L  47|W  43|L  25|W  60|W  44|W  39|"
## [26] "   25 | LOREN SCHWIEBERT               |3.5  |L   9|W  53|L   3|W  24|D  34|L  10|W  47|"
## [27] "   26 | MAX ZHU                        |3.5  |W  49|W  40|W  17|L   4|L   9|D  32|L  11|"
## [28] "   27 | GAURAV GIDWANI                 |3.5  |W  51|L  13|W  46|W  37|D  14|L   6|U    |"
## [29] "   28 | SOFIA ADINA STANESCU-BELLU      |3.5  |W  24|D   4|W  22|D  19|L  20|L   8|D  36|"
## [30] "   29 | CHIEDOZIE OKORIE               |3.5  |W  50|D   6|L  38|L  34|W  52|W  48|U    |"
## [31] "   30 | GEORGE AVERY JONES             |3.5  |L  52|D  64|L  15|W  55|L  31|W  61|W  50|"
## [32] "   31 | RISHI SHETTY                   |3.5  |L  58|D  55|W  64|L  10|W  30|W  50|L  14|"
## [33] "   32 | JOSHUA PHILIP MATHEWS          |3.5  |W  61|L   8|W  44|L  18|W  51|D  26|L  13|"
## [34] "   33 | JADE GE                        |3.5  |W  60|L  12|W  50|D  36|L  13|L  15|W  51|"
## [35] "   34 | MICHAEL JEFFERY THOMAS         |3.5  |L   6|W  60|L  37|W  29|D  25|L  11|W  52|"
## [36] "   35 | JOSHUA DAVID LEE               |3.5  |L  46|L  38|W  56|L   6|W  57|D  52|W  48|"
## [37] "   36 | SIDDHARTH JHA                  |3.5  |L  13|W  57|W  51|D  33|H    |L  16|D  28|"
## [38] "   37 | AMIYATOSH PWNANANDAM           |3.5  |B    |L   5|W  34|L  27|H    |L  23|W  61|"
## [39] "   38 | BRIAN LIU                      |3.0  |D  11|W  35|W  29|L  12|H    |L  18|L  15|"
## [40] "   39 | JOEL R HENDON                  |3.0  |L   1|W  54|W  40|L  16|W  44|L  21|L  24|"
## [41] "   40 | FOREST ZHANG                   |3.0  |W  20|L  26|L  39|W  59|L  21|W  56|L  22|"
## [42] "   41 | KYLE WILLIAM MURPHY            |3.0  |W  59|L  17|W  58|L  20|X    |U    |U    |"
## [43] "   42 | JARED GE                       |3.0  |L  12|L  50|L  57|D  60|D  61|W  64|W  56|"
## [44] "   43 | ROBERT GLEN VASEY              |3.0  |L  21|L  23|L  24|W  63|W  59|L  46|W  55|"
## [45] "   44 | JUSTIN D SCHILLING             |3.0  |B    |L  14|L  32|W  53|L  39|L  24|W  59|"
## [46] "   45 | DEREK YAN                      |3.0  |L   5|L  51|D  60|L  56|W  63|D  55|W  58|"
## [47] "   46 | JACOB ALEXANDER LAVALLEY       |3.0  |W  35|L   7|L  27|L  50|W  64|W  43|L  23|"
## [48] "   47 | ERIC WRIGHT                    |2.5  |L  18|W  24|L  21|W  61|L   8|D  51|L  25|"
## [49] "   48 | DANIEL KHAIN                   |2.5  |L  17|W  63|H    |D  52|H    |L  29|L  35|"
## [50] "   49 | MICHAEL J MARTIN               |2.5  |L  26|L  20|D  63|D  64|W  58|H    |U    |"
## [51] "   50 | SHIVAM JHA                     |2.5  |L  29|W  42|L  33|W  46|H    |L  31|L  30|"
```

```
## [52] "   51 | TEJAS AYYAGARI                  |2.5  |L  27|W  45|L  36|W  57|L  32|D  47|L  33|"
## [53] "   52 | ETHAN GUO                       |2.5  |W  30|D  22|L  19|D  48|L  29|D  35|L  34|"
## [54] "   53 | JOSE C YBARRA                   |2.0  |H    |L  25|H    |L  44|U    |W  57|U    |"
## [55] "   54 | LARRY HODGE                     |2.0  |L  14|L  39|L  61|B    |L  15|L  59|W  64|"
## [56] "   55 | ALEX KONG                       |2.0  |L  62|D  31|L  10|L  30|B    |D  45|L  43|"
## [57] "   56 | MARISA RICCI                    |2.0  |H    |L  11|L  35|W  45|H    |L  40|L  42|"
## [58] "   57 | MICHAEL LU                      |2.0  |L   7|L  36|W  42|L  51|L  35|L  53|B    |"
## [59] "   58 | VIRAJ MOHILE                    |2.0  |W  31|L   2|L  41|L  23|L  49|B    |L  45|"
## [60] "   59 | SEAN M MC CORMICK               |2.0  |L  41|B    |L   9|L  40|L  43|W  54|L  44|"
## [61] "   60 | JULIA SHEN                      |1.5  |L  33|L  34|D  45|D  42|L  24|H    |U    |"
## [62] "   61 | JEZZEL FARKAS                   |1.5  |L  32|L   3|W  54|L  47|D  42|L  30|L  37|"
## [63] "   62 | ASHWIN BALAJI                   |1.0  |W  55|U    |U    |U    |U    |U    |U    |"
## [64] "   63 | THOMAS JOSEPH HOSMER            |1.0  |L   2|L  48|D  49|L  43|L  45|H    |U    |"
## [65] "   64 | BEN LI                          |1.0  |L  22|D  30|L  31|D  49|L  46|L  42|L  54|"
```

#subset even rows

```
data_row_even <- data[row_odd == 0, ]
data_row_even
```

```
##  [1] " Num  | USCF ID / Rtg (Pre->Post)       | Pts |  1  |  2  |  3  |  4  |  5  |  6  |  7  | "
##  [2] "   ON | 15445895 / R: 1794   ->1817     |N:2  |W    |B    |W    |B    |W    |B    |W    |"
##  [3] "   MI | 14598900 / R: 1553   ->1663     |N:2  |B    |W    |B    |W    |B    |W    |B    |"
##  [4] "   MI | 14959604 / R: 1384   ->1640     |N:2  |W    |B    |W    |B    |W    |B    |W    |"
##  [5] "   MI | 12616049 / R: 1716   ->1744     |N:2  |W    |B    |W    |B    |W    |B    |B    |"
##  [6] "   MI | 14601533 / R: 1655   ->1690     |N:2  |B    |W    |B    |W    |B    |W    |B    |"
##  [7] "   OH | 15055204 / R: 1686   ->1687     |N:3  |W    |B    |W    |B    |B    |W    |B    |"
##  [8] "   MI | 11146376 / R: 1649   ->1673     |N:3  |W    |B    |W    |B    |B    |W    |W    |"
##  [9] "   MI | 15142253 / R: 1641P17->1657P24  |N:3  |B    |W    |B    |W    |B    |W    |W    |"
## [10] "   ON | 14954524 / R: 1411   ->1564     |N:2  |W    |B    |W    |B    |W    |B    |B    |"
## [11] "   MI | 14150362 / R: 1365   ->1544     |N:3  |W    |W    |B    |B    |W    |B    |W    |"
## [12] "   MI | 12581589 / R: 1712   ->1696     |N:3  |B    |W    |B    |W    |B    |W    |B    |"
## [13] "   MI | 12681257 / R: 1663   ->1670     |N:3  |W    |B    |W    |B    |     |W    |B    |"
## [14] "   MI | 15082995 / R: 1666   ->1662     |N:3  |B    |W    |B    |B    |W    |W    |B    |"
## [15] "   MI | 10131499 / R: 1610   ->1618     |N:3  |W    |B    |W    |W    |B    |B    |W    |"
## [16] "   MI | 15619130 / R: 1220P13->1416P20  |N:3  |B    |B    |W    |W    |B    |B    |W    |"
## [17] "   MI | 10295068 / R: 1604   ->1613     |N:3  |B    |W    |     |B    |W    |B    |     |"
## [18] "   MI | 10297702 / R: 1629   ->1610     |N:3  |W    |B    |W    |B    |W    |B    |W    |"
## [19] "   MI | 11342094 / R: 1600   ->1600     |N:3  |B    |W    |B    |W    |B    |W    |B    |"
## [20] "   MI | 14862333 / R: 1564   ->1570     |N:3  |W    |B    |W    |B    |W    |W    |B    |"
## [21] "   MI | 14529060 / R: 1595   ->1569     |N:4  |W    |B    |W    |B    |W    |B    |W    |"
## [22] "   ON | 15495066 / R: 1563P22->1562     |N:3  |B    |W    |B    |W    |W    |B    |W    |"
## [23] "   MI | 12405534 / R: 1555   ->1529     |N:4  |W    |B    |W    |B    |     |W    |B    |"
## [24] "   ON | 15030142 / R: 1363   ->1371     |     |B    |W    |B    |W    |B    |W    |B    |"
## [25] "   MI | 13469010 / R: 1229   ->1300     |N:4  |B    |W    |B    |B    |W    |W    |B    |"
## [26] "   MI | 12486656 / R: 1745   ->1681     |N:4  |B    |W    |B    |W    |B    |W    |B    |"
## [27] "   ON | 15131520 / R: 1579   ->1564     |N:4  |B    |W    |B    |W    |B    |W    |W    |"
## [28] "   MI | 14476567 / R: 1552   ->1539     |N:4  |W    |B    |W    |B    |W    |B    |     |"
## [29] "   MI | 14882954 / R: 1507   ->1513     |N:3  |W    |W    |B    |W    |B    |B    |W    |"
## [30] "   MI | 15323285 / R: 1602P6 ->1508P12  |N:4  |B    |W    |B    |W    |W    |B    |     |"
## [31] "   ON | 12577178 / R: 1522   ->1444     |     |W    |B    |B    |W    |W    |B    |B    |"
## [32] "   MI | 15131618 / R: 1494   ->1444     |     |B    |W    |B    |W    |B    |W    |B    |"
## [33] "   ON | 14073750 / R: 1441   ->1433     |N:4  |W    |B    |W    |B    |W    |B    |W    |"
## [34] "   MI | 14691842 / R: 1449   ->1421     |     |B    |W    |B    |W    |B    |W    |B    |"
## [35] "   MI | 15051807 / R: 1399   ->1400     |     |B    |W    |B    |B    |W    |B    |W    |"
```

```
## [36] "   MI | 14601397 / R: 1438   ->1392     |     |W    |W    |B    |W    |B    |B    |W    |"
## [37] "   MI | 14773163 / R: 1355   ->1367     |N:4  |W    |B    |W    |B    |     |W    |B    |"
## [38] "   MI | 15489571 / R:  980P12->1077P17  |     |     |B    |W    |W    |     |B    |W    |"
## [39] "   MI | 15108523 / R: 1423   ->1439     |N:4  |W    |B    |W    |W    |     |B    |B    |"
## [40] "   MI | 12923035 / R: 1436P23->1413     |N:4  |B    |W    |B    |W    |B    |W    |W    |"
## [41] "   MI | 14892710 / R: 1348   ->1346     |     |B    |B    |W    |W    |B    |W    |W    |"
## [42] "   MI | 15761443 / R: 1403P5 ->1341P9   |     |B    |W    |B    |W    |     |     |     |"
## [43] "   MI | 14462326 / R: 1332   ->1256     |     |B    |W    |B    |B    |W    |W    |B    |"
## [44] "   MI | 14101068 / R: 1283   ->1244     |     |W    |B    |W    |W    |B    |B    |W    |"
## [45] "   MI | 15323504 / R: 1199   ->1199     |     |     |W    |B    |B    |W    |B    |W    |"
## [46] "   MI | 15372807 / R: 1242   ->1191     |     |W    |B    |W    |B    |W    |B    |W    |"
## [47] "   MI | 15490981 / R:  377P3 ->1076P10  |     |B    |W    |B    |W    |B    |W    |W    |"
## [48] "   MI | 12533115 / R: 1362   ->1341     |     |W    |B    |W    |B    |W    |B    |W    |"
## [49] "   MI | 14369165 / R: 1382   ->1335     |     |B    |W    |     |B    |     |W    |B    |"
## [50] "   MI | 12531685 / R: 1291P12->1259P17  |     |W    |W    |B    |W    |B    |     |     |"
## [51] "   MI | 14773178 / R: 1056   ->1111     |     |W    |B    |W    |B    |     |B    |W    |"
## [52] "   MI | 15205474 / R: 1011   ->1097     |     |B    |W    |B    |W    |B    |W    |W    |"
## [53] "   MI | 14918803 / R:  935   ->1092     |N:4  |B    |W    |B    |W    |B    |W    |B    |"
## [54] "   MI | 12578849 / R: 1393   ->1359     |     |     |B    |     |W    |     |W    |     |"
## [55] "   MI | 12836773 / R: 1270   ->1200     |     |B    |B    |W    |     |W    |B    |W    |"
## [56] "   MI | 15412571 / R: 1186   ->1163     |     |W    |B    |W    |B    |     |W    |B    |"
## [57] "   MI | 14679887 / R: 1153   ->1140     |     |     |B    |W    |W    |     |B    |W    |"
## [58] "   MI | 15113330 / R: 1092   ->1079     |     |B    |W    |W    |B    |W    |B    |     |"
## [59] "   MI | 14700365 / R:  917   -> 941     |     |W    |B    |W    |B    |W    |     |B    |"
## [60] "   MI | 12841036 / R:  853   -> 878     |     |W    |     |B    |B    |W    |W    |B    |"
## [61] "   MI | 14579262 / R:  967   -> 984     |     |W    |B    |B    |W    |B    |     |     |"
## [62] "   ON | 15771592 / R:  955P11-> 979P18  |     |B    |W    |B    |W    |B    |W    |B    |"
## [63] "   MI | 15219542 / R: 1530   ->1535     |     |B    |     |     |     |     |     |     |"
## [64] "   MI | 15057092 / R: 1175   ->1125     |     |W    |B    |W    |B    |B    |     |     |"
## [65] "   MI | 15006561 / R: 1163   ->1112     |     |B    |W    |W    |B    |W    |B    |B    |"
```

#creating a new data_frame to store tournament information

```
Info <- data_frame(Info_1= data_row_odd,
                   Info_2 = data_row_even)
```

```
## Warning: `data_frame()` was deprecated in tibble 1.1.0.
## Please use `tibble()` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was generated.
```

add a new variable by concating 2 varible into 1

```
library(stringr)

Info_2<- mutate ( Info, Tournamentinfo =str_c(Info$Info_1, " ", Info$Info_2))
```

#removing the header from original txt

```
Info_2<- Info_2 [-c(1), ]
```

##checking each str length

```
str_length(Info_2$Info_1)
```

```
##  [1] 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89
```

```
## [26] 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89
## [51] 89 89 89 89 89 89 89 89 89 89 89 89 89
```

```
str_length(Info_2$Info_2)
```

```
##  [1] 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89
## [26] 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89 89
## [51] 89 89 89 89 89 89 89 89 89 89 89 89 89
```

store a sample value which to find a location in the string

```
text<-c("    1 | GARY HUA                        |6.0  |W  39|W  21|W  18|W  14|W   7|D  12|D   4|    O
```

Finding the location in the string

```
str_locate(text, pattern = "    1 ")
```

```
##      start end
## [1,]     1   6
```

```
str_locate(text, pattern = "GARY HUA                         ")
```

```
##      start end
## [1,]     9  40
```

```
str_locate(text, pattern = "6.0  ")
```

```
##      start end
## [1,]    42  46
```

```
str_locate(text, pattern = "39")
```

```
##      start end
## [1,]    51  52
```

```
str_locate(text, pattern = "21")
```

```
##      start end
## [1,]    57  58
```

```
str_locate(text, pattern = "18")
```

```
##      start end
## [1,]    63  64
```

```
str_locate(text, pattern = "14")
```

```
##      start end
## [1,]    69  70
```

```
str_locate(text, pattern = "  7")
```

```
##      start end
## [1,]    74  76
```

```
str_locate(text, pattern = "12")
```

```
##      start end
## [1,]    81  82
```

```
str_locate(text, pattern = "  4")
```

```
##      start end
## [1,]    86  88
```

```
str_locate(text, pattern = "ON")
```

```
##      start end
## [1,]    94  95
```

```
str_locate(text, pattern = "1794")
```

```
##      start end
## [1,]   113 116
```

store values based on their location

```
Number <- substr(Info_2$Tournamentinfo,1,6)
Name <- substr(Info_2$Tournamentinfo,9,40)
Total_Points<- substr(Info_2$Tournamentinfo,42,46)
R1 <- substr(Info_2$Tournamentinfo,51,52)
R2 <- substr(Info_2$Tournamentinfo,57,58)
R3 <- substr(Info_2$Tournamentinfo,63,64)
R4 <- substr(Info_2$Tournamentinfo,69,70)
R5 <- substr(Info_2$Tournamentinfo,75,76)
R6 <- substr(Info_2$Tournamentinfo,81,82)
R7 <- substr(Info_2$Tournamentinfo,87,88)
state <- substr(Info_2$Tournamentinfo,94,95)
Pre_rating <- substr(Info_2$Tournamentinfo,113,116)
```

#creating a data.frame to store tournament info

```
Tournament_Info <- data.frame(Play_ID = Number,
                              Name = Name,
                              State=state,
                              Total_Points =Total_Points,
                              Pre_R=Pre_rating,
                              R1=R1,
                              R2=R2,
                              R3=R3,
                              R4=R4,
                              R5=R5,
                              R6=R6,
                              R7=R7
                              )
```

Creating Reference table to store players' ID and pre_rating

```
Ref_Table <- data.frame(PlayID=as.numeric(Number),
                        Pre_R= as.numeric(Pre_rating))
```

##creating the round table to store each round's opponent ID

```
R_Table <- data.frame(Play_ID=as.numeric(Number),
                          R1_ID=as.numeric(R1),
```

```
                          R2_ID=as.numeric(R2),
                          R3_ID=as.numeric(R3),
                          R4_ID=as.numeric(R4),
                          R5_ID=as.numeric(R5),
                          R6_ID=as.numeric(R6),
                          R7_ID=as.numeric(R7)
)
```

#insert R1's component's pre_rating

```
library(dplyr)

Temp1<-left_join(R_Table,Ref_Table,by =c("R1_ID"="PlayID"))

colnames(Temp1)[9] <- "Pre_R_R1"
```

#insert R2's component's pre_rating

```
library(dplyr)

Temp2<-left_join(Temp1,Ref_Table,by =c("R2_ID"="PlayID"))

colnames(Temp2)[10] <- "Pre_R_R2"
```

#insert R3's component's pre_rating

```
library(dplyr)

Temp3<-left_join(Temp2,Ref_Table,by =c("R3_ID"="PlayID"))

colnames(Temp3)[11] <- "Pre_R_R3"
```

#insert R4's component's pre_rating

```
library(dplyr)

Temp4<-left_join(Temp3,Ref_Table,by =c("R4_ID"="PlayID"))

colnames(Temp4)[12] <- "Pre_R_R4"
```

#insert R5's component's pre_rating

```
library(dplyr)

Temp5<-left_join(Temp4,Ref_Table,by =c("R5_ID"="PlayID"))

colnames(Temp5)[13] <- "Pre_R_R5"
```

#insert R6's component's pre_rating

```
library(dplyr)

Temp6<-left_join(Temp5,Ref_Table,by =c("R6_ID"="PlayID"))

colnames(Temp6)[14] <- "Pre_R_R6"
```

#insert R7's component's pre_rating

```
library(dplyr)

Temp7<-left_join(Temp6,Ref_Table,by =c("R7_ID"="PlayID"))

colnames(Temp7)[15] <- "Pre_R_R7"
```

creating a new data.frame for computing the pre_average rating

```
Pre_Rating_Cal <- subset(Temp7, select= -c(2:8))
```

counting NA in the each row

```
Pre_Rating_Cal$CountNa <- rowSums(is.na(Pre_Rating_Cal))
```

replacing NA into 0

```
Pre_Rating_Cal[is.na(Pre_Rating_Cal)] =0
```

adding new column for Average_Pre_Rating_Opponents (=Sum_Pre_Rating/#ofRoundPlay

```
Pre_Rating_Cal_Final <-mutate(Pre_Rating_Cal,Average_Pre_Rating_Opponents = (Pre_R_R1+Pre_R_R2+Pre_R_R3
           )
```

creating a final data.frame to store the project1 values

```
Project1 <- data_frame(Player_Name = Tournament_Info$Name,
                       Player_State = Tournament_Info$State,
                       Total_Points = Tournament_Info$Total_Points,
                       Player_Pre_Rating = Tournament_Info$Pre_R,
                       Average_Pre_Rating_Opponents = round(Pre_Rating_Cal_Final$Average_Pre_Rating_Opp
```

##export to csv file

```
write.csv(Project1, file='/Users/joycealdrich/Documents/SPS Data Science/Data 607/Project_1/DATA607_Pro
```

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.