Wrangling Chess Tournament Data

Alec McCabe

9/12/2021

Setup

Load libraries

```
library(tidyverse)
```

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

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.4     v dplyr   1.0.7
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   2.0.1     v forcats 0.5.1
```

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

```
library(stringr)
library("RMySQL")
```

```
## Loading required package: DBI
```

Establish connection to SQL

After processing and formatting the input data as per assignment instructions, we will be saving the output in SQL. I've created two tables:

- players: a table of players
- scores: a table of player scores from tournaments played

```
mydb = dbConnect(MySQL(), user = 'root', dbname='data_607', host='localhost')
```

The below code will close all open SQL connections if run

```
lapply(dbListConnections(dbDriver(drv = "MySQL")), dbDisconnect)
```

Convert [input data] –> [desired assignment format]

This assignment's input data is a "|" delimited .txt file containing information about players in a chess tournament. This project aims to wrangle the input data into a tabular format, with the following columns:

- Player's Name
- Player's State
- Total Number of Points
- Player's Pre-Rating
- Average Pre Chess Rating of Opponents

Import data and inspect

```
data = read.table("https://raw.githubusercontent.com/man-of-moose/masters_607/main/projects/project_1/c
                  sep="|",
                  fill=TRUE
                  )
```

```
data %>%
  select(V1:V11)
```

```
##                                                                                          V1
## 1   -----------------------------------------------------------------------------------------
## 2                                                                                      Pair
## 3                                                                                      Num
## 4   -----------------------------------------------------------------------------------------
## 5                                                                                          1
## 6                                                                                         ON
## 7   -----------------------------------------------------------------------------------------
## 8                                                                                          2
## 9                                                                                         MI
## 10  -----------------------------------------------------------------------------------------
## 11                                                                                         3
## 12                                                                                        MI
## 13  -----------------------------------------------------------------------------------------
## 14                                                                                         4
## 15                                                                                        MI
## 16  -----------------------------------------------------------------------------------------
## 17                                                                                         5
## 18                                                                                        MI
## 19  -----------------------------------------------------------------------------------------
## 20                                                                                         6
## 21                                                                                        OH
## 22  -----------------------------------------------------------------------------------------
```

```
## 23                                                                                         7
## 24                                                                                        MI
## 25  -----------------------------------------------------------------------------------------
## 26                                                                                         8
## 27                                                                                        MI
## 28  -----------------------------------------------------------------------------------------
## 29                                                                                         9
## 30                                                                                        ON
## 31  -----------------------------------------------------------------------------------------
## 32                                                                                        10
## 33                                                                                        MI
## 34  -----------------------------------------------------------------------------------------
## 35                                                                                        11
## 36                                                                                        MI
## 37  -----------------------------------------------------------------------------------------
## 38                                                                                        12
## 39                                                                                        MI
## 40  -----------------------------------------------------------------------------------------
## 41                                                                                        13
## 42                                                                                        MI
## 43  -----------------------------------------------------------------------------------------
## 44                                                                                        14
## 45                                                                                        MI
## 46  -----------------------------------------------------------------------------------------
## 47                                                                                        15
## 48                                                                                        MI
## 49  -----------------------------------------------------------------------------------------
## 50                                                                                        16
## 51                                                                                        MI
## 52  -----------------------------------------------------------------------------------------
## 53                                                                                        17
## 54                                                                                        MI
## 55  -----------------------------------------------------------------------------------------
## 56                                                                                        18
## 57                                                                                        MI
## 58  -----------------------------------------------------------------------------------------
## 59                                                                                        19
## 60                                                                                        MI
## 61  -----------------------------------------------------------------------------------------
## 62                                                                                        20
## 63                                                                                        MI
## 64  -----------------------------------------------------------------------------------------
## 65                                                                                        21
## 66                                                                                        ON
## 67  -----------------------------------------------------------------------------------------
## 68                                                                                        22
## 69                                                                                        MI
## 70  -----------------------------------------------------------------------------------------
## 71                                                                                        23
## 72                                                                                        ON
## 73  -----------------------------------------------------------------------------------------
## 74                                                                                        24
## 75                                                                                        MI
## 76  -----------------------------------------------------------------------------------------
```

```
## 77                                                                                      25
## 78                                                                                      MI
## 79  ----------------------------------------------------------------------------------------
## 80                                                                                      26
## 81                                                                                      ON
## 82  ----------------------------------------------------------------------------------------
## 83                                                                                      27
## 84                                                                                      MI
## 85  ----------------------------------------------------------------------------------------
## 86                                                                                      28
## 87                                                                                      MI
## 88  ----------------------------------------------------------------------------------------
## 89                                                                                      29
## 90                                                                                      MI
## 91  ----------------------------------------------------------------------------------------
## 92                                                                                      30
## 93                                                                                      ON
## 94  ----------------------------------------------------------------------------------------
## 95                                                                                      31
## 96                                                                                      MI
## 97  ----------------------------------------------------------------------------------------
## 98                                                                                      32
## 99                                                                                      ON
## 100 ----------------------------------------------------------------------------------------
## 101                                                                                     33
## 102                                                                                     MI
## 103 ----------------------------------------------------------------------------------------
## 104                                                                                     34
## 105                                                                                     MI
## 106 ----------------------------------------------------------------------------------------
## 107                                                                                     35
## 108                                                                                     MI
## 109 ----------------------------------------------------------------------------------------
## 110                                                                                     36
## 111                                                                                     MI
## 112 ----------------------------------------------------------------------------------------
## 113                                                                                     37
## 114                                                                                     MI
## 115 ----------------------------------------------------------------------------------------
## 116                                                                                     38
## 117                                                                                     MI
## 118 ----------------------------------------------------------------------------------------
## 119                                                                                     39
## 120                                                                                     MI
## 121 ----------------------------------------------------------------------------------------
## 122                                                                                     40
## 123                                                                                     MI
## 124 ----------------------------------------------------------------------------------------
## 125                                                                                     41
## 126                                                                                     MI
## 127 ----------------------------------------------------------------------------------------
## 128                                                                                     42
## 129                                                                                     MI
## 130 ----------------------------------------------------------------------------------------
```

```
## 131                                                                                       43
## 132                                                                                       MI
## 133 ----------------------------------------------------------------------------------------
## 134                                                                                       44
## 135                                                                                       MI
## 136 ----------------------------------------------------------------------------------------
## 137                                                                                       45
## 138                                                                                       MI
## 139 ----------------------------------------------------------------------------------------
## 140                                                                                       46
## 141                                                                                       MI
## 142 ----------------------------------------------------------------------------------------
## 143                                                                                       47
## 144                                                                                       MI
## 145 ----------------------------------------------------------------------------------------
## 146                                                                                       48
## 147                                                                                       MI
## 148 ----------------------------------------------------------------------------------------
## 149                                                                                       49
## 150                                                                                       MI
## 151 ----------------------------------------------------------------------------------------
## 152                                                                                       50
## 153                                                                                       MI
## 154 ----------------------------------------------------------------------------------------
## 155                                                                                       51
## 156                                                                                       MI
## 157 ----------------------------------------------------------------------------------------
## 158                                                                                       52
## 159                                                                                       MI
## 160 ----------------------------------------------------------------------------------------
## 161                                                                                       53
## 162                                                                                       MI
## 163 ----------------------------------------------------------------------------------------
## 164                                                                                       54
## 165                                                                                       MI
## 166 ----------------------------------------------------------------------------------------
## 167                                                                                       55
## 168                                                                                       MI
## 169 ----------------------------------------------------------------------------------------
## 170                                                                                       56
## 171                                                                                       MI
## 172 ----------------------------------------------------------------------------------------
## 173                                                                                       57
## 174                                                                                       MI
## 175 ----------------------------------------------------------------------------------------
## 176                                                                                       58
## 177                                                                                       MI
## 178 ----------------------------------------------------------------------------------------
## 179                                                                                       59
## 180                                                                                       MI
## 181 ----------------------------------------------------------------------------------------
## 182                                                                                       60
## 183                                                                                       MI
## 184 ----------------------------------------------------------------------------------------
```

```
## 185                                                                                         61
## 186                                                                                         ON
## 187 -------------------------------------------------------------------------------------------
## 188                                                                                         62
## 189                                                                                         MI
## 190 -------------------------------------------------------------------------------------------
## 191                                                                                         63
## 192                                                                                         MI
## 193 -------------------------------------------------------------------------------------------
## 194                                                                                         64
## 195                                                                                         MI
## 196 -------------------------------------------------------------------------------------------
##                               V2    V3    V4    V5    V6    V7    V8    V9
## 1
## 2    Player Name                     Total Round Round Round Round Round Round
## 3    USCF ID / Rtg (Pre->Post)        Pts    1     2     3     4     5     6
## 4
## 5    GARY HUA                        6.0   W  39 W  21 W  18 W  14 W   7 D  12
## 6    15445895 / R: 1794   ->1817     N:2   W     B     W     B     W     B
## 7
## 8    DAKSHESH DARURI                 6.0   W  63 W  58 L   4 W  17 W  16 W  20
## 9    14598900 / R: 1553   ->1663     N:2   B     W     B     W     B     W
## 10
## 11   ADITYA BAJAJ                    6.0   L   8 W  61 W  25 W  21 W  11 W  13
## 12   14959604 / R: 1384   ->1640     N:2   W     B     W     B     W     B
## 13
## 14   PATRICK H SCHILLING             5.5   W  23 D  28 W   2 W  26 D   5 W  19
## 15   12616049 / R: 1716   ->1744     N:2   W     B     W     B     W     B
## 16
## 17   HANSHI ZUO                      5.5   W  45 W  37 D  12 D  13 D   4 W  14
## 18   14601533 / R: 1655   ->1690     N:2   B     W     B     W     B     W
## 19
## 20   HANSEN SONG                     5.0   W  34 D  29 L  11 W  35 D  10 W  27
## 21   15055204 / R: 1686   ->1687     N:3   W     B     W     B     B     W
## 22
## 23   GARY DEE SWATHELL               5.0   W  57 W  46 W  13 W  11 L   1 W   9
## 24   11146376 / R: 1649   ->1673     N:3   W     B     W     B     B     W
## 25
## 26   EZEKIEL HOUGHTON                5.0   W   3 W  32 L  14 L   9 W  47 W  28
## 27   15142253 / R: 1641P17->1657P24  N:3   B     W     B     W     B     W
## 28
## 29   STEFANO LEE                     5.0   W  25 L  18 W  59 W   8 W  26 L   7
## 30   14954524 / R: 1411   ->1564     N:2   W     B     W     B     W     B
## 31
## 32   ANVIT RAO                       5.0   D  16 L  19 W  55 W  31 D   6 W  25
## 33   14150362 / R: 1365   ->1544     N:3   W     W     B     B     W     B
## 34
## 35   CAMERON WILLIAM MC LEMAN        4.5   D  38 W  56 W   6 L   7 L   3 W  34
## 36   12581589 / R: 1712   ->1696     N:3   B     W     B     W     B     W
## 37
## 38   KENNETH J TACK                  4.5   W  42 W  33 D   5 W  38 H     D   1
## 39   12681257 / R: 1663   ->1670     N:3   W     B     W     B           W
## 40
## 41   TORRANCE HENRY JR               4.5   W  36 W  27 L   7 D   5 W  33 L   3
```

```
## 42   15082995 / R: 1666   ->1662      N:3   B     W     B     B     W     W
## 43
## 44   BRADLEY SHAW                      4.5   W  54 W  44 W   8 L   1 D  27 L   5
## 45   10131499 / R: 1610   ->1618      N:3   W     B     W     W     B     B
## 46
## 47   ZACHARY JAMES HOUGHTON            4.5   D  19 L  16 W  30 L  22 W  54 W  33
## 48   15619130 / R: 1220P13->1416P20   N:3   B     B     W     W     B     B
## 49
## 50   MIKE NIKITIN                      4.0   D  10 W  15 H     W  39 L   2 W  36
## 51   10295068 / R: 1604   ->1613      N:3   B     W           B     W     B
## 52
## 53   RONALD GRZEGORCZYK                4.0   W  48 W  41 L  26 L   2 W  23 W  22
## 54   10297702 / R: 1629   ->1610      N:3   W     B     W     B     W     B
## 55
## 56   DAVID SUNDEEN                     4.0   W  47 W   9 L   1 W  32 L  19 W  38
## 57   11342094 / R: 1600   ->1600      N:3   B     W     B     W     B     W
## 58
## 59   DIPANKAR ROY                      4.0   D  15 W  10 W  52 D  28 W  18 L   4
## 60   14862333 / R: 1564   ->1570      N:3   W     B     W     B     W     W
## 61
## 62   JASON ZHENG                       4.0   L  40 W  49 W  23 W  41 W  28 L   2
## 63   14529060 / R: 1595   ->1569      N:4   W     B     W     B     W     B
## 64
## 65   DINH DANG BUI                     4.0   W  43 L   1 W  47 L   3 W  40 W  39
## 66   15495066 / R: 1563P22->1562      N:3   B     W     B     W     W     B
## 67
## 68   EUGENE L MCCLURE                  4.0   W  64 D  52 L  28 W  15 H     L  17
## 69   12405534 / R: 1555   ->1529      N:4   W     B     W     B           W
## 70
## 71   ALAN BUI                          4.0   L   4 W  43 L  20 W  58 L  17 W  37
## 72   15030142 / R: 1363   ->1371            B     W     B     W     B     W
## 73
## 74   MICHAEL R ALDRICH                 4.0   L  28 L  47 W  43 L  25 W  60 W  44
## 75   13469010 / R: 1229   ->1300      N:4   B     W     B     B     W     W
## 76
## 77   LOREN SCHWIEBERT                  3.5   L   9 W  53 L   3 W  24 D  34 L  10
## 78   12486656 / R: 1745   ->1681      N:4   B     W     B     W     B     W
## 79
## 80   MAX ZHU                           3.5   W  49 W  40 W  17 L   4 L   9 D  32
## 81   15131520 / R: 1579   ->1564      N:4   B     W     B     W     B     W
## 82
## 83   GAURAV GIDWANI                    3.5   W  51 L  13 W  46 W  37 D  14 L   6
## 84   14476567 / R: 1552   ->1539      N:4   W     B     W     B     W     B
## 85
## 86   SOFIA ADINA STANESCU-BELLU        3.5   W  24 D   4 W  22 D  19 L  20 L   8
## 87   14882954 / R: 1507   ->1513      N:3   W     W     B     W     B     B
## 88
## 89   CHIEDOZIE OKORIE                  3.5   W  50 D   6 L  38 L  34 W  52 W  48
## 90   15323285 / R: 1602P6 ->1508P12   N:4   B     W     B     W     W     B
## 91
## 92   GEORGE AVERY JONES                3.5   L  52 D  64 L  15 W  55 L  31 W  61
## 93   12577178 / R: 1522   ->1444            W     B     B     W     W     B
## 94
## 95   RISHI SHETTY                      3.5   L  58 D  55 W  64 L  10 W  30 W  50
```

```
## 96   15131618 / R: 1494   ->1444            B     W     B     W     B     W
## 97
## 98   JOSHUA PHILIP MATHEWS            3.5   W  61 L   8 W  44 L  18 W  51 D  26
## 99   14073750 / R: 1441   ->1433      N:4   W     B     W     B     W     B
## 100
## 101  JADE GE                          3.5   W  60 L  12 W  50 D  36 L  13 L  15
## 102  14691842 / R: 1449   ->1421            B     W     B     W     B     W
## 103
## 104  MICHAEL JEFFERY THOMAS           3.5   L   6 W  60 L  37 W  29 D  25 L  11
## 105  15051807 / R: 1399   ->1400            B     W     B     B     W     B
## 106
## 107  JOSHUA DAVID LEE                 3.5   L  46 L  38 W  56 L   6 W  57 D  52
## 108  14601397 / R: 1438   ->1392            W     W     B     W     B     B
## 109
## 110  SIDDHARTH JHA                    3.5   L  13 W  57 W  51 D  33 H     L  16
## 111  14773163 / R: 1355   ->1367      N:4   W     B     W     B           W
## 112
## 113  AMIYATOSH PWNANANDAM             3.5   B     L   5 W  34 L  27 H     L  23
## 114  15489571 / R:  980P12->1077P17                B     W     W           B
## 115
## 116  BRIAN LIU                        3.0   D  11 W  35 W  29 L  12 H     L  18
## 117  15108523 / R: 1423   ->1439      N:4   W     B     W     W           B
## 118
## 119  JOEL R HENDON                    3.0   L   1 W  54 W  40 L  16 W  44 L  21
## 120  12923035 / R: 1436P23->1413      N:4   B     W     B     W     B     W
## 121
## 122  FOREST ZHANG                     3.0   W  20 L  26 L  39 W  59 L  21 W  56
## 123  14892710 / R: 1348   ->1346            B     B     W     W     B     W
## 124
## 125  KYLE WILLIAM MURPHY              3.0   W  59 L  17 W  58 L  20 X     U
## 126  15761443 / R: 1403P5 ->1341P9          B     W     B     W
## 127
## 128  JARED GE                         3.0   L  12 L  50 L  57 D  60 D  61 W  64
## 129  14462326 / R: 1332   ->1256            B     W     B     B     W     W
## 130
## 131  ROBERT GLEN VASEY                3.0   L  21 L  23 L  24 W  63 W  59 L  46
## 132  14101068 / R: 1283   ->1244            W     B     W     W     B     B
## 133
## 134  JUSTIN D SCHILLING               3.0   B     L  14 L  32 W  53 L  39 L  24
## 135  15323504 / R: 1199   ->1199                  W     B     B     W     B
## 136
## 137  DEREK YAN                        3.0   L   5 L  51 D  60 L  56 W  63 D  55
## 138  15372807 / R: 1242   ->1191            W     B     W     B     W     B
## 139
## 140  JACOB ALEXANDER LAVALLEY         3.0   W  35 L   7 L  27 L  50 W  64 W  43
## 141  15490981 / R:  377P3 ->1076P10         B     W     B     W     B     W
## 142
## 143  ERIC WRIGHT                      2.5   L  18 W  24 L  21 W  61 L   8 D  51
## 144  12533115 / R: 1362   ->1341            W     B     W     B     W     B
## 145
## 146  DANIEL KHAIN                     2.5   L  17 W  63 H     D  52 H     L  29
## 147  14369165 / R: 1382   ->1335            B     W           B           W
## 148
## 149  MICHAEL J MARTIN                 2.5   L  26 L  20 D  63 D  64 W  58 H
```

```
## 150  12531685 / R: 1291P12->1259P17        W     W     B     W     B
## 151
## 152  SHIVAM JHA                        2.5   L  29 W  42 L  33 W  46 H     L  31
## 153  14773178 / R: 1056   ->1111            W     B     W     B           B
## 154
## 155  TEJAS AYYAGARI                    2.5   L  27 W  45 L  36 W  57 L  32 D  47
## 156  15205474 / R: 1011   ->1097            B     W     B     W     B     W
## 157
## 158  ETHAN GUO                         2.5   W  30 D  22 L  19 D  48 L  29 D  35
## 159  14918803 / R:  935   ->1092      N:4   B     W     B     W     B     W
## 160
## 161  JOSE C YBARRA                     2.0   H     L  25 H     L  44 U     W  57
## 162  12578849 / R: 1393   ->1359                  B           W           W
## 163
## 164  LARRY HODGE                       2.0   L  14 L  39 L  61 B     L  15 L  59
## 165  12836773 / R: 1270   ->1200            B     B     W           W     B
## 166
## 167  ALEX KONG                         2.0   L  62 D  31 L  10 L  30 B     D  45
## 168  15412571 / R: 1186   ->1163            W     B     W     B           W
## 169
## 170  MARISA RICCI                      2.0   H     L  11 L  35 W  45 H     L  40
## 171  14679887 / R: 1153   ->1140                  B     W     W           B
## 172
## 173  MICHAEL LU                        2.0   L   7 L  36 W  42 L  51 L  35 L  53
## 174  15113330 / R: 1092   ->1079            B     W     W     B     W     B
## 175
## 176  VIRAJ MOHILE                      2.0   W  31 L   2 L  41 L  23 L  49 B
## 177  14700365 / R:  917   -> 941            W     B     W     B     W
## 178
## 179  SEAN M MC CORMICK                 2.0   L  41 B     L   9 L  40 L  43 W  54
## 180  12841036 / R:  853   -> 878            W           B     B     W     W
## 181
## 182  JULIA SHEN                        1.5   L  33 L  34 D  45 D  42 L  24 H
## 183  14579262 / R:  967   -> 984            W     B     B     W     B
## 184
## 185  JEZZEL FARKAS                     1.5   L  32 L   3 W  54 L  47 D  42 L  30
## 186  15771592 / R:  955P11-> 979P18         B     W     B     W     B     W
## 187
## 188  ASHWIN BALAJI                     1.0   W  55 U     U     U     U     U
## 189  15219542 / R: 1530   ->1535            B
## 190
## 191  THOMAS JOSEPH HOSMER              1.0   L   2 L  48 D  49 L  43 L  45 H
## 192  15057092 / R: 1175   ->1125            W     B     W     B     B
## 193
## 194  BEN LI                            1.0   L  22 D  30 L  31 D  49 L  46 L  42
## 195  15006561 / R: 1163   ->1112            B     W     W     B     W     B
## 196
##       V10 V11
## 1          NA
## 2   Round  NA
## 3     7    NA
## 4          NA
## 5   D   4  NA
## 6   W      NA
```

```
## 7           NA
## 8   W    7  NA
## 9   B       NA
## 10          NA
## 11  W   12  NA
## 12  W       NA
## 13          NA
## 14  D    1  NA
## 15  B       NA
## 16          NA
## 17  W   17  NA
## 18  B       NA
## 19          NA
## 20  W   21  NA
## 21  B       NA
## 22          NA
## 23  L    2  NA
## 24  W       NA
## 25          NA
## 26  W   19  NA
## 27  W       NA
## 28          NA
## 29  W   20  NA
## 30  B       NA
## 31          NA
## 32  W   18  NA
## 33  W       NA
## 34          NA
## 35  W   26  NA
## 36  B       NA
## 37          NA
## 38  L    3  NA
## 39  B       NA
## 40          NA
## 41  W   32  NA
## 42  B       NA
## 43          NA
## 44  W   31  NA
## 45  W       NA
## 46          NA
## 47  W   38  NA
## 48  W       NA
## 49          NA
## 50  U       NA
## 51          NA
## 52          NA
## 53  L    5  NA
## 54  W       NA
## 55          NA
## 56  L   10  NA
## 57  B       NA
## 58          NA
## 59  L    8  NA
## 60  B       NA
```

```
## 61          NA
## 62  L    9  NA
## 63  W       NA
## 64          NA
## 65  L    6  NA
## 66  W       NA
## 67          NA
## 68  W   40  NA
## 69  B       NA
## 70          NA
## 71  W   46  NA
## 72  B       NA
## 73          NA
## 74  W   39  NA
## 75  B       NA
## 76          NA
## 77  W   47  NA
## 78  B       NA
## 79          NA
## 80  L   11  NA
## 81  W       NA
## 82          NA
## 83  U       NA
## 84          NA
## 85          NA
## 86  D   36  NA
## 87  W       NA
## 88          NA
## 89  U       NA
## 90          NA
## 91          NA
## 92  W   50  NA
## 93  B       NA
## 94          NA
## 95  L   14  NA
## 96  B       NA
## 97          NA
## 98  L   13  NA
## 99  W       NA
## 100         NA
## 101 W   51  NA
## 102 B       NA
## 103         NA
## 104 W   52  NA
## 105 W       NA
## 106         NA
## 107 W   48  NA
## 108 W       NA
## 109         NA
## 110 D   28  NA
## 111 B       NA
## 112         NA
## 113 W   61  NA
## 114 W       NA
```

```
## 115         NA
## 116 L  15   NA
## 117 B       NA
## 118         NA
## 119 L  24   NA
## 120 W       NA
## 121         NA
## 122 L  22   NA
## 123 W       NA
## 124         NA
## 125 U       NA
## 126         NA
## 127         NA
## 128 W  56   NA
## 129 B       NA
## 130         NA
## 131 W  55   NA
## 132 W       NA
## 133         NA
## 134 W  59   NA
## 135 W       NA
## 136         NA
## 137 W  58   NA
## 138 W       NA
## 139         NA
## 140 L  23   NA
## 141 W       NA
## 142         NA
## 143 L  25   NA
## 144 W       NA
## 145         NA
## 146 L  35   NA
## 147 B       NA
## 148         NA
## 149 U       NA
## 150         NA
## 151         NA
## 152 L  30   NA
## 153 W       NA
## 154         NA
## 155 L  33   NA
## 156 W       NA
## 157         NA
## 158 L  34   NA
## 159 B       NA
## 160         NA
## 161 U       NA
## 162         NA
## 163         NA
## 164 W  64   NA
## 165 W       NA
## 166         NA
## 167 L  43   NA
## 168 B       NA
```

```
## 169          NA
## 170 L   42   NA
## 171 W        NA
## 172          NA
## 173 B        NA
## 174          NA
## 175          NA
## 176 L   45   NA
## 177 B        NA
## 178          NA
## 179 L   44   NA
## 180 B        NA
## 181          NA
## 182 U        NA
## 183          NA
## 184          NA
## 185 L   37   NA
## 186 B        NA
## 187          NA
## 188 U        NA
## 189          NA
## 190          NA
## 191 U        NA
## 192          NA
## 193          NA
## 194 L   54   NA
## 195 B        NA
## 196          NA
```

Change column names

Rename columns to make analysis meaningful

```
column_names <- c("id","name","total","round_1","round_2","round_3","round_4",
                  "round_5","round_6","round_7","delete")

colnames(data) <- column_names
```

Remove "———-" rows

Following data import, there are some rows that serve no purpose for our assignment. Additionally, an unexpected column of NAs was created which we will want to delete. We will use dplyr::filer and dplyr::select to achieve this.

```
data<- data %>%
        filter(str_detect(id, "[a-zA-z\\d]")) %>%
        select(-delete)
```

Looking at our data now

Due to the structure of the input data, and the steps we've taken so far, our current dataframe is structured in an interesting way. Rows with a numeric "id" value contain information about a player's name, their total score, and the rounds they played. While rows with a character "id" contain information about a player's state and pre_score.

```
head(data)
```

```
##       id                              name total round_1 round_2 round_3
## 1  Pair   Player Name                       Total   Round   Round   Round
## 2  Num    USCF ID / Rtg (Pre->Post)          Pts      1       2       3
## 3     1   GARY HUA                          6.0     W  39   W  21   W  18
## 4    ON   15445895 / R: 1794   ->1817       N:2     W       B       W
## 5     2   DAKSHESH DARURI                   6.0     W  63   W  58   L   4
## 6    MI   14598900 / R: 1553   ->1663       N:2     B       W       B
##   round_4 round_5 round_6 round_7
## 1   Round   Round   Round   Round
## 2     4       5       6       7
## 3   W  14   W   7   D  12   D   4
## 4   B       W       B       W
## 5   W  17   W  16   W  20   W   7
## 6   W       B       W       B
```

Get state_data character vector

We need to capture the state data in a character vector, to later be used to represent the state column of our output. We can do this by filtering for rows that contain 2 capital letters, and saving the "id" column into a variable called state_data

```
state_data <- data$id
state_data <- state_data[grepl("[A-Z]{2}",state_data)]
state_data <- str_trim(state_data, side = c("both"))
```

Get pre_score data

We can extract the pre_score data in the same way we did state_data. Unlike state_data, the pre_score data will require more advanced regex to properly extract.

```
pre_score_data <- data$name

data$id <- data$id %>%
  str_trim(side=c("both"))

pre_score_data <- data %>%
  filter(str_detect(id,"[A-Z]{2}")) %>%
  .$name

pre_score_data <- pre_score_data %>%
              str_extract("R:[^\\d]*\\d*") %>%
              str_replace("R:","") %>%
              str_trim(side=c("both"))
```

Remove rows with non-numeric values

Now that we've extracted the state data and pre_score data, we can remove rows with non-numeric values.

```
data <- data %>%
        filter(str_detect(id,"\\d"))
```

Add state and pre_score columns

And now we can take state_data and pre_score_data and add them as new columns to the recently filtered dataframe.

```
data$state <- state_data
data$pre_score <- as.integer(pre_score_data)
```

Rearrange columns for clarity

Using dply::select and everything() we can easily re-arrange our column values for easier reading.

```
data <- data %>%
          select(name, state, pre_score, total, everything())
```

```
head(data)
```

```
##                                 name state pre_score total id round_1 round_2
## 1  GARY HUA                            ON      1794 6.0    1   W  39   W  21
## 2  DAKSHESH DARURI                     MI      1553 6.0    2   W  63   W  58
## 3  ADITYA BAJAJ                        MI      1384 6.0    3   L   8   W  61
## 4  PATRICK H SCHILLING                 MI      1716 5.5    4   W  23   D  28
## 5  HANSHI ZUO                          MI      1655 5.5    5   W  45   W  37
## 6  HANSEN SONG                         OH      1686 5.0    6   W  34   D  29
##   round_3 round_4 round_5 round_6 round_7
## 1   W  18   W  14   W   7   D  12   D   4
## 2   L   4   W  17   W  16   W  20   W   7
## 3   W  25   W  21   W  11   W  13   W  12
## 4   W   2   W  26   D   5   W  19   D   1
## 5   D  12   D  13   D   4   W  14   W  17
## 6   L  11   W  35   D  10   W  27   W  21
```

Convert total into double

The 'total' value was parsed as a character. since we will be applying math to this later, we need to convert to a double.

```
data$total <- as.double(data$total)
```

Create new column, "oppo_ids"

This new column will include vectors containing the opponent ids for each respective player. As an example, Gary Hua's value here would be:

c(39,21,18,14,7,12,4)

This is achieved by first concatenating each of the "round_" columns. Following this, we use stringr to parse out and collect the opponent ids.

```
data <- data %>% mutate(oppo_ids = str_c(round_1,round_2,round_3,round_4,round_5,round_6,round_7))

data$oppo_ids <- data$oppo_ids %>%
  str_replace_all("[A-Z]","") %>%
  str_trim(side=c("both")) %>%
  str_replace_all("\\s{2,}","|")

data$oppo_ids <- data$oppo_ids %>% str_split("\\|")
```

Create function to calculate average opponent score

This function will use the previously created "oppo_id" column values as input, in order to filter for and average the correct opponent pre_scores.

```
get_avg_oppo_score <- function(id_data) {
  temp_df <- data %>%
              filter(id %in% id_data) %>%
              summarise(pre_score_mean = mean(pre_score, na.rm=TRUE))

  return(temp_df[,1])
}
```

Test it out on the first example

```
get_avg_oppo_score(c(39,21,18,14,7,12,4))
```

```
## [1] 1605.286
```

Apply function to entire dataframe

```
data$avg_oppo_score <- lapply(data$oppo_ids,FUN=get_avg_oppo_score)
```

Calculate total number of games played

We will need this later on for extra credit. Here we are counting how many games each player participated in.

```
data$number_of_games <- as.integer(
  lapply(
    lapply(data$oppo_ids,FUN=lengths),
    FUN=sum
    )
  )
```

Select only interesting columns

There are a few columns we don't need anymore such as all of the "round_" columns, the "oppo_ids" column, and others. We can use dplyr::select to select only what's interesting.

```
final_data <- data %>%
                select(name,
                       state,
                       total,
                       number_of_games,
                       pre_score,
                       avg_oppo_score
                       )
```

Round avg_oppo_score (Average Opponent Score) and inspect

Based on the description of this project, we will be rounding the values of avg_oppo_score with the round() function.

```
final_data$avg_oppo_score <- as.integer(
  lapply(final_data$avg_oppo_score,FUN=round)
  )
```

```
final_data
```

```
##                                  name state total number_of_games pre_score
## 1   GARY HUA                                ON   6.0               7      1794
## 2   DAKSHESH DARURI                         MI   6.0               7      1553
## 3   ADITYA BAJAJ                            MI   6.0               7      1384
## 4   PATRICK H SCHILLING                     MI   5.5               7      1716
## 5   HANSHI ZUO                              MI   5.5               7      1655
## 6   HANSEN SONG                             OH   5.0               7      1686
## 7   GARY DEE SWATHELL                       MI   5.0               7      1649
## 8   EZEKIEL HOUGHTON                        MI   5.0               7      1641
## 9   STEFANO LEE                             ON   5.0               7      1411
## 10  ANVIT RAO                               MI   5.0               7      1365
## 11  CAMERON WILLIAM MC LEMAN                MI   4.5               7      1712
## 12  KENNETH J TACK                          MI   4.5               6      1663
## 13  TORRANCE HENRY JR                       MI   4.5               7      1666
## 14  BRADLEY SHAW                            MI   4.5               7      1610
## 15  ZACHARY JAMES HOUGHTON                  MI   4.5               7      1220
## 16  MIKE NIKITIN                            MI   4.0               5      1604
## 17  RONALD GRZEGORCZYK                      MI   4.0               7      1629
## 18  DAVID SUNDEEN                           MI   4.0               7      1600
## 19  DIPANKAR ROY                            MI   4.0               7      1564
## 20  JASON ZHENG                             MI   4.0               7      1595
## 21  DINH DANG BUI                           ON   4.0               7      1563
## 22  EUGENE L MCCLURE                        MI   4.0               6      1555
## 23  ALAN BUI                                ON   4.0               7      1363
## 24  MICHAEL R ALDRICH                       MI   4.0               7      1229
## 25  LOREN SCHWIEBERT                        MI   3.5               7      1745
## 26  MAX ZHU                                 ON   3.5               7      1579
```

```
## 27  GAURAV GIDWANI                     MI   3.5               6        1552
## 28  SOFIA ADINA STANESCU-BELLU         MI   3.5               7        1507
## 29  CHIEDOZIE OKORIE                   MI   3.5               6        1602
## 30  GEORGE AVERY JONES                 ON   3.5               7        1522
## 31  RISHI SHETTY                       MI   3.5               7        1494
## 32  JOSHUA PHILIP MATHEWS              ON   3.5               7        1441
## 33  JADE GE                            MI   3.5               7        1449
## 34  MICHAEL JEFFERY THOMAS             MI   3.5               7        1399
## 35  JOSHUA DAVID LEE                   MI   3.5               7        1438
## 36  SIDDHARTH JHA                      MI   3.5               6        1355
## 37  AMIYATOSH PWNANANDAM               MI   3.5               5         980
## 38  BRIAN LIU                          MI   3.0               6        1423
## 39  JOEL R HENDON                      MI   3.0               7        1436
## 40  FOREST ZHANG                       MI   3.0               7        1348
## 41  KYLE WILLIAM MURPHY                MI   3.0               4        1403
## 42  JARED GE                           MI   3.0               7        1332
## 43  ROBERT GLEN VASEY                  MI   3.0               7        1283
## 44  JUSTIN D SCHILLING                 MI   3.0               6        1199
## 45  DEREK YAN                          MI   3.0               7        1242
## 46  JACOB ALEXANDER LAVALLEY           MI   3.0               7         377
## 47  ERIC WRIGHT                        MI   2.5               7        1362
## 48  DANIEL KHAIN                       MI   2.5               5        1382
## 49  MICHAEL J MARTIN                   MI   2.5               5        1291
## 50  SHIVAM JHA                         MI   2.5               6        1056
## 51  TEJAS AYYAGARI                     MI   2.5               7        1011
## 52  ETHAN GUO                          MI   2.5               7         935
## 53  JOSE C YBARRA                      MI   2.0               3        1393
## 54  LARRY HODGE                        MI   2.0               6        1270
## 55  ALEX KONG                          MI   2.0               6        1186
## 56  MARISA RICCI                       MI   2.0               5        1153
## 57  MICHAEL LU                         MI   2.0               6        1092
## 58  VIRAJ MOHILE                       MI   2.0               6         917
## 59  SEAN M MC CORMICK                  MI   2.0               6         853
## 60  JULIA SHEN                         MI   1.5               5         967
## 61  JEZZEL FARKAS                      ON   1.5               7         955
## 62  ASHWIN BALAJI                      MI   1.0               1        1530
## 63  THOMAS JOSEPH HOSMER               MI   1.0               5        1175
## 64  BEN LI                             MI   1.0               7        1163
##    avg_oppo_score
## 1            1605
## 2            1469
## 3            1564
## 4            1574
## 5            1501
## 6            1519
## 7            1372
## 8            1468
## 9            1523
## 10           1554
## 11           1468
## 12           1506
## 13           1498
## 14           1515
## 15           1484
```

```
## 16                 1386
## 17                 1499
## 18                 1480
## 19                 1426
## 20                 1411
## 21                 1470
## 22                 1300
## 23                 1214
## 24                 1357
## 25                 1363
## 26                 1507
## 27                 1222
## 28                 1522
## 29                 1314
## 30                 1144
## 31                 1260
## 32                 1379
## 33                 1277
## 34                 1375
## 35                 1150
## 36                 1388
## 37                 1385
## 38                 1539
## 39                 1430
## 40                 1391
## 41                 1248
## 42                 1150
## 43                 1107
## 44                 1327
## 45                 1152
## 46                 1358
## 47                 1392
## 48                 1356
## 49                 1286
## 50                 1296
## 51                 1356
## 52                 1495
## 53                 1345
## 54                 1206
## 55                 1406
## 56                 1414
## 57                 1363
## 58                 1391
## 59                 1319
## 60                 1330
## 61                 1327
## 62                 1186
## 63                 1350
## 64                 1263
```

Trim names

While you can't tell from the above tibble, many of the player names actually have surrounding white spaces. We can remoe with stringr::str_trim

```
final_data$name <- str_trim(final_data$name, side=c("both"))
```

Calculate expected score for each player

In chess a player's "total score" for a game is determined by whether or not the player wins (+1), loses (+0), or draws (+0.5)

The "expected score" for a player in one game can be represented as a modified probability that they will win, based on their pre_score relative to their opponent's pre_score.

The following function can perform the required calculation:

```
1/(10^(({oppo_pre_score}-{pre_score})/400)+1)
```

Because have already computed averages for our opponent_pre_scores, we can modify the above equation as such:

```
1/(10^(({oppo_Pre_score}-{pre_score})/400)+1) * {number_of_games}
```

The function used above was identified from the following sources:

- http://www.uschess.org/index.php/Players-Ratings/Do-NOT-edit-CLOSE-immediately.html
- https://chess.stackexchange.com/questions/18209/how-do-you-calculate-your-tournament-performance-rating

```
final_data <- final_data %>%
                mutate(
  expected_total = 1/(10^((avg_oppo_score-pre_score)/400)+1) * number_of_games
  )
```

EXTRA CREDIT: which player scored the most points relative to their expected score?

Answer is Aditya Bajaj, who performed very well throughout this tournament. In fact, Adtiya's total was more than 4.16 points above his expected total. He won 6 out of 7 games, despite the fact that, on average, he was rated nearly 200 points below each of his opponents.

```
final_data %>%
  mutate(score_differential = total - expected_total) %>%
  arrange(desc(score_differential)) %>%
  .[1,c('name','pre_score','avg_oppo_score','total','expected_total','score_differential')]
```

```
##           name pre_score avg_oppo_score total expected_total score_differential
## 1 ADITYA BAJAJ      1384           1564     6       1.833237           4.166763
```

Generate a .CSV file and load values into SQL

Generate a .CSV file

```
write.table(final_data, sep=",", file = "/Users/alecmccabe/Desktop/Masters Program/DATA 607/masters_607/
```

Create function to assign ids to players

The reason why I chose to include two tables in my SQL database was to allow for continued use of this script. When new tournaments happen, new players may participate.

This function will work by looking at the total list of tournament participants, and cross-reference that list against the existing SQL table data_607.players

This ensures that if a participant has already been counted in previous tournaments, they will be assigned the same player_id.

Alternatively, if there is a new participant, this function will ensure that their generated player_id does not match any existing ones.

```
assign_player_ids <- function(insert_data, mydb) {
  names <- insert_data$name
  players_string <- str_c('"',str_trim(names,side=c("both")),'"',collapse=",")
  insert_data$id <- NA
  query <- str_interp("SELECT player_name, id FROM data_607.players WHERE player_name in (${players_stri

  select_data <- dbGetQuery(mydb, query)

  for (row in 1:nrow(select_data)) {
    select_name <- select_data[row,"player_name"]
    select_id <- select_data[row,"id"]

    insert_data <- within(insert_data, id[name == select_name] <- select_id)
  }

  for (row in 1:nrow(insert_data)){
    if (is.na(insert_data[row,]$id)) {
      if (sum(!is.na(insert_data$id))>0) {
        max_id <- max(insert_data$id, na.rm=TRUE) +1
      } else {
        max_id <- 1
      }
      insert_data[row,]$id <- max_id
    }
  }

  return(insert_data)
}
```

Running id assignment

Because data_607.players is currently empty, each of the participants in this tournament will be provided with incremental ids, starting with 1 and ending at 64.

```
final_data <-assign_player_ids(final_data,mydb)
```

Create insert function to load into data_607.players

This function will load any new players, and their associated player_ids and state information into the data_607.players table.

```
insert_players <- function(data, mydb){

  names <- final_data$name
  players_string <- str_c('"',str_trim(names,side=c("both")),'"',collapse=",")
  query <- str_interp("SELECT player_name, id FROM data_607.players WHERE player_name in (${players_str

  select_data <- dbGetQuery(mydb, query)

  for (row in 1:nrow(final_data)){
    id <- as.integer(final_data[row, "id"])
    name <- str_trim(final_data[row, "name"], side=c("both"))
    state <- str_trim(final_data[row, "state"], side=c("both"))
    total <- final_data[row, "total"]
    pre_score <- final_data[row, "pre_score"]
    avg_opponent_score <- final_data[row, "avg_oppo_score"]

    insert_query <- str_interp('insert into data_607.players VALUES (${id},"${name}", "${state}")')

    if (name %in% select_data$player_name) {
      next
    } else {
      print(name)
      dbGetQuery(mydb, insert_query)
    }

  }
}
```

Create a function to insert into the data_607.scores table

This function will load a player's performance data and metrics into the data_607.scores table. This function takes 'tournament_id' variable as input in addition to data and db_connection.

```
insert_scores <- function(data, tournament_id, mydb){

  for (row in 1:nrow(final_data)){
    id <- as.integer(final_data[row, "id"])
    name <- str_trim(final_data[row, "name"], side=c("both"))
    total <- final_data[row, "total"]
    pre_score <- final_data[row, "pre_score"]
    avg_opponent_score <- final_data[row, "avg_oppo_score"]
    number_of_games <- final_data[row, "number_of_games"]
    expected_total <- final_data[row, "expected_total"]
```

```
    insert_query <- str_interp('insert into scores VALUES (DEFAULT,${tournament_id},${id},${number_of_ga

    dbGetQuery(mydb, insert_query)
  }
}
```

Insert into SQL tables The insert_players function prints to the console each player's name that is added to the SQL data ("new players"). As we see below, everyone is added.

```
insert_players(final_data, mydb)
```

```
insert_scores(final_data,1, mydb)
```

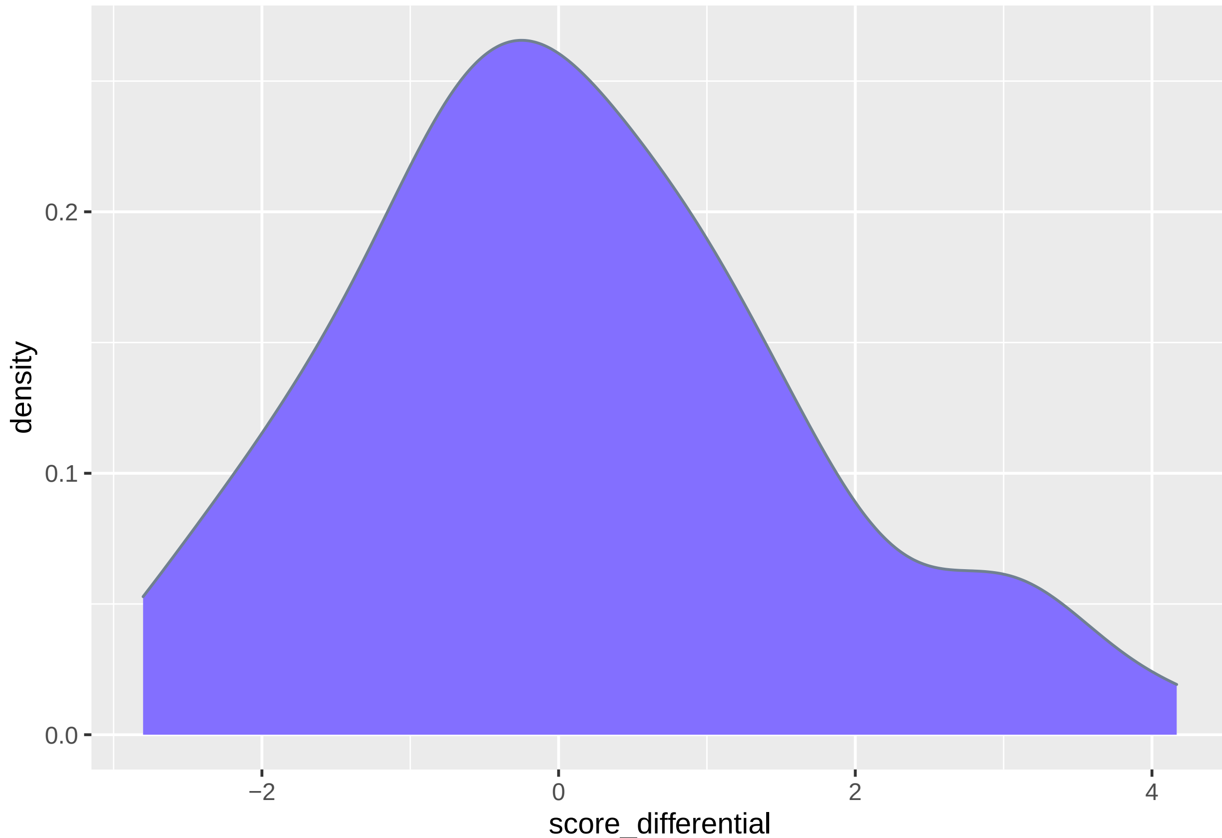
Visualize the distribution of 'score_differential' for players

'score_differential' will be defined as the difference between a player's expected total, and their actual total points.

```
final_data <- final_data %>%
  mutate(score_differential = total - expected_total)
```

The density plot below suggests that the score_difference distribution is nearly gaussian, with a mean centered around zero and only a slight right skew.

```
final_data %>%
  ggplot(aes(x=score_differential)) +
  geom_density(
    fill = "slateblue1",
    color = "slategrey"
    ) +
  theme_grey()
```

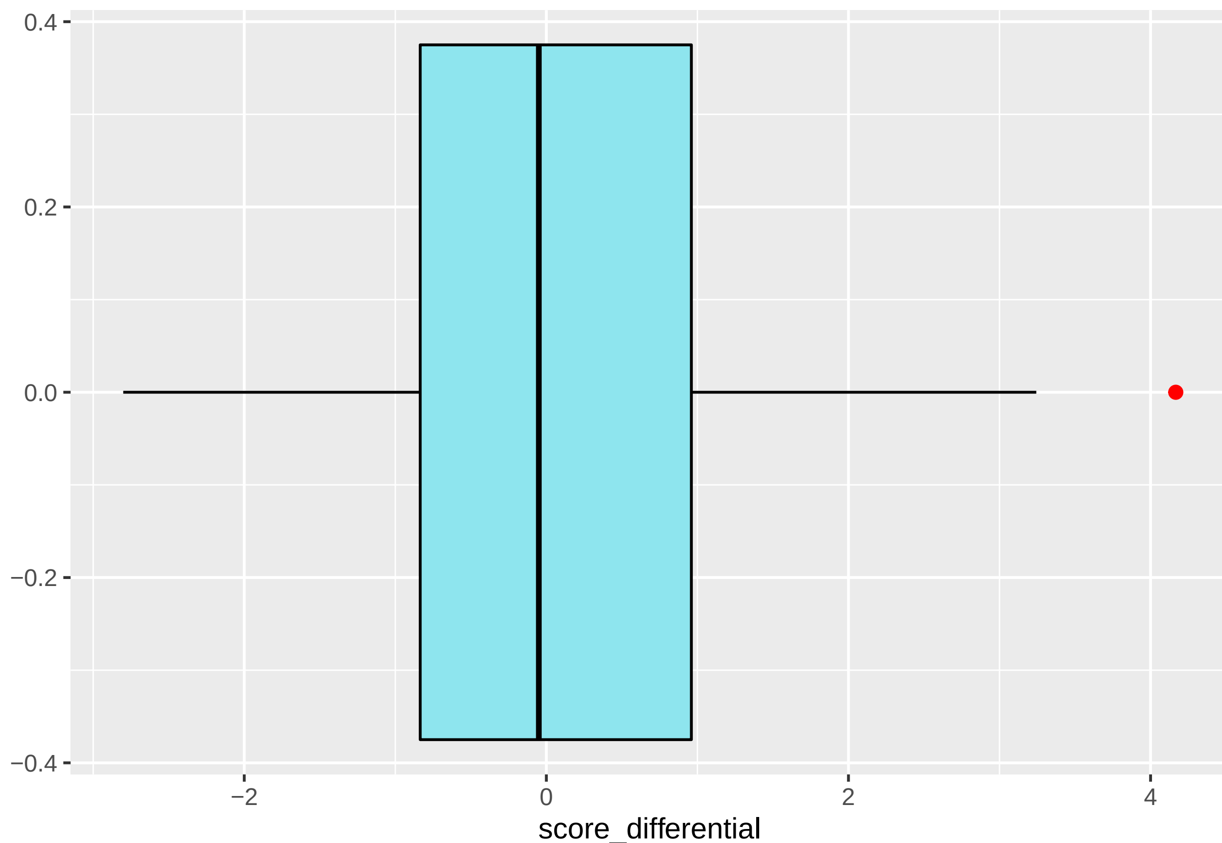


The below boxplot also shows that score_difference distribution is centered around zero, and seemingly normally distributed. Roughly 50% of the population's score_difference is between -1 and 1. Tail values stretch from -2.5 (performed much worse than expected) all the way to 2.5 (performed much better than expected).

There is also an outlier identified, with a score_differential of 4.166. As we discussed earlier, Aditya performed much better than expected.

Based on the data, one could make that claim that Aditya's performance was not a fluke, but rather a result of an "inaccurate" initial pre_score going into the tournament.

```
final_data %>%
  ggplot(aes(x=score_differential)) +
  geom_boxplot(
    color = "black",
    fill="cadetblue2",
    outlier.size=2,
    outlier.colour="red"
  ) +
  theme_grey()
```



The concept of an “innacurate” pre_score is interesting. Let’s see if there is any correlation between the absolute score_differential and a player’s pre_score.

```
final_data %>%
  mutate(
    score_differential = abs(score_differential)
  ) %>%
  select(pre_score, score_differential) %>%
  cor() %>%
  .[1,2]
```

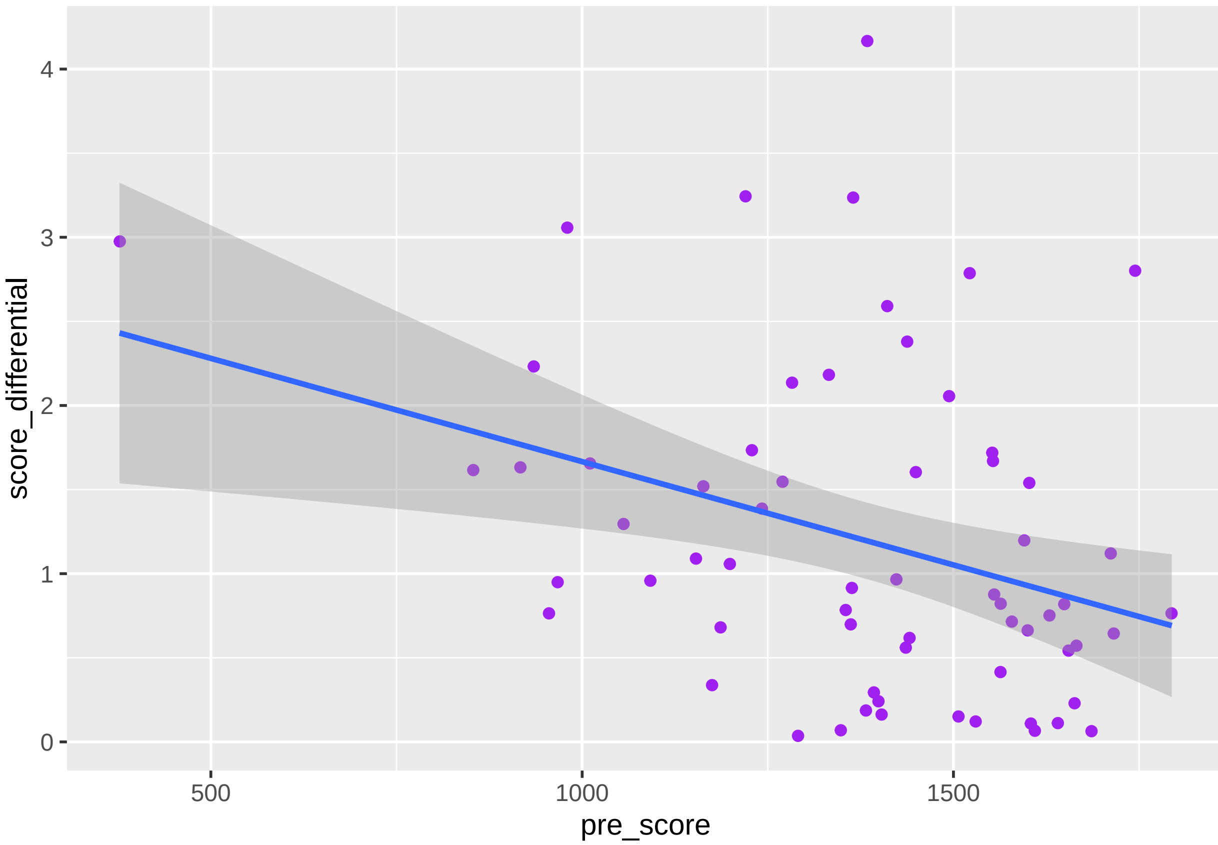
```
## [1] -0.3396838
```

There is a small but not insignificant correlation. The below graph illustrates the correlation between pre_score and score_differential. As pre_score goes up, the absolute score_differential goes down. This implies that players with higher pre_scores can place more faith in their expected_totals than players with low pre_scores.

And intuitively, this makes sense. Many players have low pre_scores simply because they don’t have as many tournament games played. Even Magnus Carlson was once lowly rated (though I doubt that lasted long).

```
final_data %>%
  mutate(
    score_differential = abs(score_differential)
  ) %>%
```

```
ggplot() +
geom_point(aes(x=pre_score, y=score_differential),
           position="jitter",
           color = "purple") +
geom_smooth(aes(x=pre_score, y=score_differential),method='lm', formula=y~x)+
theme_grey()
```



Test functions "assign_player_ids" and "insert_players"

Here we will make sure that the above functions work as expected when presented with new player data.

As an example, imagine that a second tournament includes all of the members of this tournament, plus one new member: Johnny Apple.

If our functions work as expected, then the assign_player_id function will provide Johnny with the id 65, and the insert_players function will only insert Johhny (since the previous players are already contained in the SQL table)

```
final_data <- add_row(final_data,
        name="Johnny Apple",
        state="OH",
        total=5,
        number_of_games = 7,
        pre_score=1111,
```

```r
        avg_oppo_score=1245,
        expected_total = 4,
        id=NA)
```

```r
final_data <-assign_player_ids(final_data,mydb)
```

```r
final_data[final_data$name=="Johnny Apple",c('name','id')]
```

```
##            name id
## 65 Johnny Apple 65
```

```r
insert_players(final_data, mydb)
```