NBA Player Analysis - Player Point Production for 2022-2023 NBA Season

Abhinav Yerramreddy

2024-12-11

```r
# Read in dataset from Kaggle
nba <- read.csv("2023_nba_player_stats.csv", header = TRUE, sep = ",")

# Clean data
clean_data<- nba[1:534,] #last few rows had NAs in position

# Remove redundant variables
clean_data <- clean_data[, !colnames(clean_data) %in% c("TOV","FP", "DD2", "TD3","X...")]

tail(clean_data)
```

```
##               PName POS Team Age GP W L  Min PTS FGM FGA   FG. X3PM X3PA  X3P.
## 529  Jacob Gilyard   G  MEM  24  1 0 1 40.8   3   1   3  33.3    1    3  33.3
## 530 Jordan Schakel   F  WAS  24  2 0 2  6.2   3   1   2  50.0    1    1 100.0
## 531   Trevor Keels   G  NYK  19  3 3 0  8.0   3   1   4  25.0    1    4  25.0
## 532   Chima Moneke   F  SAC  27  2 1 1  7.9   2   1   2  50.0    0    0   0.0
## 533    Chris Silva   F  DAL  26  1 1 0  3.0   2   1   1 100.0    0    0   0.0
## 534  Stanley Umude   G  DET  24  1 0 1  2.1   2   0   1   0.0    0    1   0.0
##     FTM FTA FT. OREB DREB REB AST STL BLK PF
## 529   0   0   0    0    4   4   7   3   0  3
## 530   0   0   0    0    0   0   1   1   0  1
## 531   0   0   0    0    2   2   0   0   0  0
## 532   0   2   0    2    0   2   1   0   0  2
## 533   0   0   0    0    0   0   0   0   0  2
## 534   2   2 100    0    0   0   0   1   1  0
```

```r
head(clean_data)
```

```
##                       PName POS Team Age GP  W  L    Min  PTS FGM  FGA  FG. X3PM
## 1              Jayson Tatum  SF  BOS  25 74 52 22 2732.2 2225 727 1559 46.6  240
## 2               Joel Embiid   C  PHI  29 66 43 23 2284.1 2183 728 1328 54.8   66
## 3               Luka Doncic  PG  DAL  24 66 33 33 2390.5 2138 719 1449 49.6  185
## 4 Shai Gilgeous-Alexander  PG  OKC  24 68 33 35 2416.0 2135 704 1381 51.0   58
## 5     Giannis Antetokounmpo  PF  MIL  28 63 47 16 2023.6 1959 707 1278 55.3   47
## 6           Anthony Edwards  SG  MIN  21 79 40 39 2841.5 1946 707 1541 45.9  213
##   X3PA X3P. FTM FTA  FT. OREB DREB REB AST STL BLK  PF
## 1  686 35.0 531 622 85.4   78  571 649 342  78  51 160
## 2  200 33.0 661 771 85.7  113  557 670 274  66 112 205
## 3  541 34.2 515 694 74.2   54  515 569 529  90  33 166
```

```
## 4  168 34.5 669 739 90.5   59  270 329 371 112  65 192
## 5  171 27.5 498 772 64.5  137  605 742 359  52  51 197
## 6  578 36.9 319 422 75.6   47  411 458 350 125  58 186
```

```r
clean_data$PTS <- as.numeric(clean_data$PTS)

clean_data$Min <- as.numeric(clean_data$Min)

# Filter out players who played fewer than 10 games
clean_data <- clean_data[clean_data$GP >= 10, ]

# Create Response Variable PPM (Points Per Minute)
clean_data$PPM <- ifelse(clean_data$Min == 0 | is.na(clean_data$Min), NA, clean_data$PTS / clean_data$Mi

which(is.na(clean_data$PPM))
```

```
## integer(0)
```

```r
mydata <- clean_data[, !colnames(clean_data) %in% c("PName","POS","Team","GP"
                                                    , "Min", "PTS", "FGM", "FGA"
                                                    , "X3PM", "X3PA", "FTM", "FTA", "REB")]

nba <- clean_data

colnames(mydata)
```

```
##  [1] "Age"  "W"    "L"    "FG."  "X3P." "FT."  "OREB" "DREB" "AST"  "STL"
## [11] "BLK"  "PF"   "PPM"
```

```r
mydata
```

```
##     Age  W  L  FG.  X3P.  FT. OREB DREB AST STL BLK  PF        PPM
## 1    25 52 22 46.6  35.0  85.4   78  571 342  78  51 160 0.81436205
## 2    29 43 23 54.8  33.0  85.7  113  557 274  66 112 205 0.95573749
## 3    24 33 33 49.6  34.2  74.2   54  515 529  90  33 166 0.89437356
## 4    24 33 35 51.0  34.5  90.5   59  270 371 112  65 192 0.88369205
## 5    28 47 16 55.3  27.5  64.5  137  605 359  52  51 197 0.96807669
## 6    21 40 39 45.9  36.9  75.6   47  411 350 125  58 186 0.68484955
## 7    28 44 33 45.9  34.3  75.7  141  626 316  49  21 233 0.70726628
## 8    26 44 24 48.4  38.6  86.7   63  226 301  99  27 168 0.79016609
## 9    24 38 35 42.9  33.5  88.6   56  161 741  80   9 104 0.75333569
## 10   28 38 39 48.5  37.5  84.8   42  303 327  69  18 159 0.69113769
## 11   32 27 31 46.3  37.1  91.4   44  233 425  50  18 109 0.88578752
## 12   25 44 29 51.2  32.4  78.0   40  266 447  83  23 172 0.74983574
## 13   33 37 37 50.4  32.4  87.2   34  309 377  83  36 186 0.67720764
## 14   26 46 21 49.1  33.5  76.5   78  381 232  75  26 172 0.74181879
## 15   29 35 36 48.0  32.4  77.4  131  425 415  65  36 228 0.64856712
## 16   25 32 34 49.9  39.2  87.5  130  440 123  42  38 137 0.74411441
## 17   28 48 21 63.2  38.3  82.2  167  650 678  87  47 174 0.72750753
## 18   21 20 56 41.6  33.8  78.6   43  241 281  59  18 131 0.64676043
## 19   23 44 38 43.0  33.6  87.0   32  193 369  63  21 214 0.68142061
## 20   26 42 41 46.8  38.2  89.5   79  285 273  91  61 159 0.56391739
```

```
## 21    35 30 26 49.3  42.7  91.5   39   302 352  52  20 117 0.84895941
## 22    26 40 28 49.1  41.6  82.9   40   201 421  61  15 152 0.68650944
## 23    31 32 28 49.4  37.9  90.5   59   245 331  66  45 165 0.72435955
## 24    23 40 21 46.6  30.7  74.8   61   296 493  66  16 100 0.81917569
## 25    38 30 25 50.0  32.1  76.8   65   392 375  50  32  88 0.81375710
## 26    31 38 37 43.7  38.9  76.9   55   273 429  70  38 153 0.59187679
## 27    25 40 35 54.0   8.3  80.6  184   504 240  88  61 208 0.58852964
## 28    26 38 36 46.4  34.4  83.2   53   336 448 112  19 106 0.56248608
## 29    26 47 32 61.5  37.3  74.2  251   722 573  65  39 279 0.55198128
## 30    33 38 31 43.6  41.2  87.9   39   247 163  49  29 130 0.66216157
## 31    27 27 38 49.8  38.5  85.1  114   432 174  58 100 196 0.71007313
## 32    23 40 29 46.2  41.0  86.3   28   157 538  85   9 148 0.60885910
## 33    21 33 47 48.5  36.1  84.2   70   259 283  77  17 183 0.56924905
## 34    26 34 19 49.4  35.1  85.5   46   194 293  51  18 159 0.80146017
## 35    33 35 29 53.9  35.0  85.0  141   234 340 117  21  80 0.68568756
## 36    30 31 25 56.3  25.7  78.4  195   507 148  59 114 146 0.76187976
## 37    32 40 42 52.0  34.9  83.5  159   744 265  60  57 179 0.52687154
## 38    20 31 41 42.7  29.8  73.8   84   413 269  60  39 160 0.59143104
## 39    22 40 33 43.4  31.0  74.0   60   308 201  31  15 179 0.57806504
## 40    23 17 46 45.2  32.9  74.9   56   262 183  46  11 125 0.67125479
## 41    30 41 38 43.8  36.9  81.2   28   242 515  67  24 187 0.50233002
## 42    34 34 13 56.0  40.4  91.9   17   296 235  34  67  99 0.81708338
## 43    27 29 35 44.8  33.3  73.0   55   408 239  36  29 145 0.60610121
## 44    23 37 30 43.9  37.8  93.4   27   333 280  51  16 101 0.57677486
## 45    30 35 45 45.8  42.5  82.2   67   333 225  92  26 156 0.54152061
## 46    29 34 35 39.3  34.2  89.8   30   250 495 123  38 193 0.52662722
## 47    33 32 24 45.7  37.1  87.1   45   297 288  83  20 159 0.68698747
## 48    29 19 44 41.5  32.7  80.9   51   208 319  74  16 119 0.59840605
## 49    23 27 35 44.7  37.7  89.4   18   143 253  42  14 142 0.60164924
## 50    20 33 45 43.4  32.3  82.8   87   231 116  48  13 162 0.58585313
## 51    26 43 22 45.4  39.8  83.3   48   209 400  66  16 103 0.60841849
## 52    29 28 35 47.5  40.1  81.3   53   230 150  51  53 152 0.57448230
## 53    32 50 17 47.9  38.4  85.9   79   262 495  79  25 116 0.59106529
## 54    21 48 31 55.4  21.6  67.4  187   524 224  60 119 218 0.47029794
## 55    34 15 44 48.8  41.1  88.4   36   187 152  34   8 113 0.67251308
## 56    30 30 31 44.4  33.8  81.6   72   174 270  33  13 120 0.63923955
## 57    27 39 32 46.9  39.6  82.9   35   180 437  70  29 140 0.54815329
## 58    20 34 42 48.2  32.5  73.1  148   451 469  57  31 143 0.53225193
## 59    24 38 20 47.9  40.8  88.3   43   248 254  56  22 150 0.67716535
## 60    35 57 21 53.1  37.4  78.4  157   363  99  37 193 203 0.52223393
## 61    31 33 19 51.2  41.6  87.1   57   281 204  72  28  84 0.70901288
## 62    30 48 34 47.3  37.4  84.7   87   279 128  57  10 103 0.46205860
## 63    22 39 21 48.1  43.4  84.5   23   153 212  49   8 132 0.60422661
## 64    33 39 19 44.1  38.5  86.7   40   314 618  71  31 112 0.56955504
## 65    23 47 34 44.8  37.0  81.9   58   279 279  80  14 166 0.51576298
## 66    21 16 58 41.6  34.3  74.7   73   215 387  61  17 242 0.52250141
## 67    24 36 31 58.9  29.2  76.0  172   495 115  37  53 190 0.59106766
## 68    21 38 39 45.6  28.1  77.2  179   333 371  83  61 170 0.44031969
## 69    23 41 22 50.6  35.5  78.8  108   318  60  65 189 227 0.65622378
## 70    29 24 26 50.6  36.5  84.2   41   155 271  45  33 107 0.69340666
## 71    23 28 28 49.0  40.0  87.1   33   172 585  91  25  69 0.61607095
## 72    34 36 37 43.6  31.1  65.6   89   334 551  76  33 162 0.54520651
## 73    24 31 35 43.3  36.9  83.9   30   143 106 104  14 102 0.54202077
## 74    22 39 40 48.4  40.6  90.5   59   222 112  89  43 155 0.46895425
```

```
## 75   24 24 49 44.4  34.8  73.0   73  285 175  66  79 193 0.48075307
## 76   24 44 31 48.5  40.2  72.5   42  209 221  80  24 181 0.51752315
## 77   22 18 41 44.2  36.6  78.4   76  238 338  82  17 156 0.55832798
## 78   25 33 34 47.6  38.7  83.8   95  237 131 128  50 200 0.47114055
## 79   30 40 39 54.7  34.9  82.6  225  579 140  20  52 245 0.56650747
## 80   27 33 34 51.5  37.6  77.2   87  404 121  30  72 169 0.64081914
## 81   27 29 33 54.8  37.3  78.3   88  378  89  36 140 215 0.60979619
## 82   25 23 22 48.4  39.0  88.2   23  223 262  32  19 116 0.72296990
## 83   27 45 23 56.4  34.7  60.8  164  282 203  54  51 129 0.53963311
## 84   20 18 57 55.3  33.3  71.5  242  436 291  70  70 258 0.51089510
## 85   30 47 27 50.1  38.9  87.6   67  357 186  68  40 149 0.44542058
## 86   24 41 21 48.7  41.4  80.0   64  277  65  37  29 119 0.60070082
## 87   24 24 53 42.2  36.1  86.1   98  264 118  70  13 123 0.49889604
## 88   22 36 39 52.1  35.6  81.2   84  253 248 103  35 191 0.46405344
## 89   27 46 27 39.6  32.6  77.9   45  199 189  65  16 240 0.47283566
## 90   25 47 30 44.8  35.9  88.9   32  171 298  49  20 125 0.60551419
## 91   22 22 60 56.9  31.5  68.0  126  326 123  41  30 147 0.45327633
## 92   25 35 32 46.1  35.0  82.6   46  238  94  35  17 199 0.48409861
## 93   26 45 36 39.5  35.7  76.9   35  248 124  65   9 107 0.48970427
## 94   29 31 29 47.9  39.7  81.2   24  151 109  50  18 125 0.65024568
## 95   31 37 34 40.1  38.5  77.0   21  230 129  49  12 120 0.47346901
## 96   28 57 25 46.2  38.1  87.5   52  241 321  54  76 177 0.43849437
## 97   24 36 38 38.8  33.0  77.2  128  216 154  75  23 243 0.44578419
## 98   19 21 58 40.8  30.7  78.6  122  447 101  43  74 227 0.41216078
## 99   30 46 21 48.4  44.4  87.0   42  238 248  45  18 109 0.57346026
## 100  28 47 23 49.6  37.0  76.8  154  518 105  29  16 115 0.54278487
## 101  22 47 33 45.3  41.1  76.5   89  282  98  61  42 162 0.40979133
## 102  27 16 32 43.1  31.9  76.0   65  184  54  68  18 141 0.62810985
## 103  25 44 24 64.4  10.0  73.3  221  445 113  54  84 153 0.43646683
## 104  24 41 35 70.5   0.0  54.1  184  518 144  65 189 212 0.42321751
## 105  22 41 38 51.7  39.8  73.6   92  215 148  74  76 272 0.39437203
## 106  30 34 36 65.9   0.0  64.4  231  583  87  56  95 210 0.43704910
## 107  25 37 34 50.8  29.2  80.3   77  385  85  42  73 220 0.43717130
## 108  26 53 27 48.3  35.8  75.8   63  265 268  87  51 190 0.40491336
## 109  27 42 38 41.0  35.0  87.6   49  209 171  42  12 166 0.40625000
## 110  28 45 29 43.1  39.2  72.2   53  229 287  72  24 170 0.40092969
## 111  27 26 46 62.9   0.0  59.2  236  417 197  66  86 195 0.47059441
## 112  23 20 48 45.9  28.5  86.0   56  189 448  89   9  98 0.44109492
## 113  24 26 31 52.5  35.6  73.8  120  377 132  30  33 159 0.51376147
## 114  33 28 51 68.0   0.0  63.6  229  471 245  46  47 217 0.41745823
## 115  34 25 44 44.6  37.1  82.1   16  118 183  43  25  78 0.43551259
## 116  32 28 40 49.9  39.4  85.3   82  338 252  60  37 232 0.43623815
## 117  21 13 23 41.1  37.6  84.3   42  189 304  46  11 118 0.66098754
## 118  24 29 31 51.4  31.0  78.3   69  165 341  87  26 131 0.47078013
## 119  24 35 29 52.9  39.8  86.4   33  160 215  33  19 108 0.45193142
## 120  21 40 42 46.4  41.5  85.7   78  249 100  72  70 147 0.35861891
## 121  29 37 29 41.5  32.9  85.7   21  144 298  50  10 145 0.41794652
## 122  30 51 25 46.2  42.3  82.4   35  173 183 112  35 145 0.34520410
## 123  26 51 29 43.8  37.1  80.6   28  172 417  83   6  32 0.42384243
## 124  24 31 43 49.7  42.4  85.2   33  174  87  32   9  97 0.39172598
## 125  37 33 26 44.0  37.5  83.1   27  224 524  91  22 126 0.43365456
## 126  26 43 33 63.4   0.0  69.7  236  520  77  29  98 219 0.37706278
## 127  22 41 30 46.8  38.6  79.6   49  180 150  47  26 177 0.37672686
## 128  35 34 33 42.8  38.5  83.4   33  149 450  73  14 139 0.39347172
```

```
## 129   22 41 39 63.8  30.8  78.1  217  359  81  56 107 251 0.42789138
## 130   19 32 48 47.2  36.0  71.4   63  173  94  38  24 136 0.44360733
## 131   21 16 60 37.7  28.0  82.1   28  193 470 104  28 219 0.36497028
## 132   22 28 32 45.4  36.4  89.4   47  241 235  37  31 158 0.50328651
## 133   23 32 36 53.7  34.6  67.7   73  260  77  42  53 176 0.62435047
## 134   24 50 27 42.5  39.0  79.3   69  243 197 126  41 191 0.36282445
## 135   28 35 30 65.3   0.0  60.3  258  459  58  45  79 139 0.45026299
## 136   21 22 60 44.8  34.3  75.2  195  301  88  96  47 189 0.43005885
## 137   30 27 27 44.7  40.6  83.1   23  142 150  43  18  86 0.50112687
## 138   25 31 49 45.5  33.2  84.1   63  283 130  83  34 209 0.39468869
## 139   22 17 12 60.8  36.8  71.4   58  144 133  32  16  65 0.78862044
## 140   27 53 19 44.0  39.9  90.5   61  176 163  62  14 117 0.38030526
## 141   28 42 34 52.9  37.2  75.0  142  451 290  88  21 195 0.30277099
## 142   23 30 43 42.7  36.6  83.8   62  215  98  55  34 235 0.40629817
## 143   33 20 30 47.5  32.5  81.1   33  182 206  42  12  70 0.46483607
## 144   25 20 43 51.8  37.4  76.1  116  286 180  37  49 199 0.50742746
## 145   33 35 30 42.6  36.5  78.2   29  229 114  42  21 136 0.39888225
## 146   23 37 37 44.3  37.2  87.1   16  196 204  54   6 122 0.41324702
## 147   26 44 37 51.9  38.9  78.0   77  199 184  42  22 149 0.38083538
## 148   30 39 37 49.3  32.8  76.2  157  264  29  47  64 143 0.46734493
## 149   23 32 43 44.1  35.0  79.0   35  164 341  70  13 179 0.34193393
## 150   25 38 25 48.6  31.9  73.9   53  230  58  19  26  72 0.48226467
## 151   24 32 46 73.2   0.0  67.9  163  270  84  33  99 187 0.43947139
## 152   23 34 31 44.8  28.9  77.8   36  134 132  42   5 101 0.56243509
## 153   22 12 26 43.9  38.7  78.0    8  140 136  43  17  57 0.59672354
## 154   29 43 18 41.5  33.6  74.6   46  145 382  93  23 172 0.35926002
## 155   25 40 37 43.3  30.3  78.9   48  231 189  54  14 113 0.38950893
## 156   22 30 35 41.9  28.6  75.0   41  169 246  41  29 108 0.53152604
## 157   22 33 43 43.7  29.7  73.9   74  413 211  65  29 209 0.34498119
## 158   22 47 30 47.0  35.3  75.0   85  286  97  45  48 143 0.41386692
## 159   28 24 28 51.9  36.1  66.1  114  357 149  43  44 185 0.49877821
## 160   33 36 32 41.1  33.3  91.0   25  119 233  47   6 114 0.41825095
## 161   23 36 37 44.1  40.9  82.0   27  148  88  48   8 105 0.49523400
## 162   23 40 40 49.3  31.2  80.5   45  175 206  62  26 182 0.32751716
## 163   24 24 24 50.6  39.3  81.9   37   68 138  27   6 112 0.59814879
## 164   27 38 33 46.4  35.6  80.5   88  256 117  71  31 145 0.32872888
## 165   21 34 40 72.0  33.3  51.6  231  389  69  26 173 169 0.39866134
## 166   26 42 30 43.5  39.7  81.7   80  245 252  97  10 133 0.35789696
## 167   19 15 51 44.0  30.2  82.9   28  153 127  33   7 115 0.43430563
## 168   29 42 34 38.6  38.9  72.5   55  329 283  65  49 239 0.27851569
## 169   28 48 29 45.1  37.2  69.4   25  151 115  35  11 122 0.43208680
## 170   20 38 29 52.5  37.0  65.2   70  161 125  41  31 157 0.47762478
## 171   22 36 20 39.9  37.1  84.3   20  115 172  39  15  96 0.60709351
## 172   31 19 45 45.7  41.3  75.7   23  115  91  13   6 118 0.49764115
## 173   29 24 41 43.1  36.5  85.2   41  134 174  69  20 117 0.42756276
## 174   24 27 29 44.8  36.5  85.8   14   95  64  30  17  71 0.50424056
## 175   31 12 39 43.6  41.4  81.4   19  139 114  38   8  98 0.58089808
## 176   27 57 24 42.3  42.1  81.6   35  167 197  66  29 158 0.35931118
## 177   27 25 17 47.0  40.4  84.2   34  151  78  50  14  83 0.54175696
## 178   31 31 44 54.3  44.1  85.3   45  189 397  81  10 105 0.42583857
## 179   24 31 35 46.9  33.5  76.4   97  172 162 103  42 208 0.33273520
## 180   27 18 49 50.8  39.4  79.3   65  182 103  46  20  63 0.44540823
## 181   29 36 33 50.9  41.0  73.5   66  302 335  78  63 146 0.33067566
## 182   23 31 37 47.6  28.3  75.9  129  265  66  21  60 153 0.50203316
```

```
## 183   26 37 31 40.2  33.4  87.2   27  118 167  62   5 154 0.36447376
## 184   24 55 24 45.4  39.5  77.0   87  276 131  41  31 192 0.31347809
## 185   19 34 38 46.5  39.0  89.4   37  116  73  42  12  87 0.45597260
## 186   29 51 27 44.2  40.1  86.7   19  166  77  31  14 145 0.42135203
## 187   26 52 24 47.9  37.8  85.3   39  152 240  25  13 123 0.40640551
## 188   27 28 34 48.0  38.2  83.1   26  184 326  43  13  75 0.37497789
## 189   23 26 44 54.6  26.5  75.9   69  336  69  30  85  92 0.42115052
## 190   28 19 18 47.3  39.6  61.1   61  125  85  45  28 107 0.53179871
## 191   24 43 35 54.8  32.2  69.1  170  412 184  83  21 190 0.32930787
## 192   33 41 32 52.7  30.5  71.3   66  459 500  74  61 229 0.26854109
## 193   36 44 19 47.6  44.6  71.4   73  317 189  30  61 121 0.32059956
## 194   37 30 25 40.4  34.5  85.9   43  182 281  57  21 143 0.35803691
## 195   19 15 41 45.3  24.6  69.8   95  200 142  43  23 138 0.42121150
## 196   19 16 51 64.8   0.0  61.1  229  366  75  44  59 182 0.36644512
## 197   21 30 27 44.1  38.3  86.8    8   87  78  20   7  55 0.63739975
## 198   30 40 34 44.2  44.5  90.4   39  186  99  30   7 118 0.40098254
## 199   27 15 14 49.5  36.6  87.4   48  188 140  20  17 111 0.62904911
## 200   26 27 30 50.0  31.3  84.1   69  190  84  37  13 136 0.54078414
## 201   32 44 35 45.6  39.5  71.1  145  281 117  50  62 179 0.30187904
## 202   25 33 26 62.3  44.1  73.8   86  253  31  16  30 102 0.53536755
## 203   27 44 38 63.0   0.0  60.6  274  486 207  52  50 224 0.29525950
## 204   21 12 38 44.2  32.7  73.8  115  288  70  20  34 133 0.40096174
## 205   31 40 34 45.7  42.6  64.3   23  141 101  34  13 140 0.36799371
## 206   27 43 31 45.8  36.3  81.5   68  232  66  27  31  97 0.45060936
## 207   32 37 41 40.9  38.0  70.3   32  251 108  54  13 166 0.23725946
## 208   32 37 24 46.3  40.5  92.7   13   87  99  36   7  96 0.46336830
## 209   26 32 24 65.6  16.7  71.9   92  218  72  34  37 131 0.51279699
## 210   29 31 35 39.1  33.7  76.3  114  202 100  55  36 144 0.27429311
## 211   25 30 31 41.1  33.6  81.3   28  113 186  67  10 125 0.41984733
## 212   22 27 33 53.7  40.2  72.3   55  127 103  42   5 156 0.35472973
## 213   26 37 22 49.2  49.4  94.9   15  143  91  32   5  82 0.41517755
## 214   24 36 26 43.6  40.8  79.1   51  178  88  24  14  74 0.45619056
## 215   23 40 36 43.0  38.1  87.2   62  246  95  59  16 120 0.29089149
## 216   28 31 42 37.4  35.2  74.8   19  103 196  45  16  79 0.40271967
## 217   25 21 44 62.9 100.0  74.9  147  269  38  11  71 154 0.43688922
## 218   24 38 34 39.1  33.5  72.4  109  142 105  56  34 114 0.39167777
## 219   21 20 33 41.9  32.7  72.3   55  106 155  67  27 120 0.42057950
## 220   24 12 44 57.3  23.8  58.8   57  137  55  53  16 124 0.52414901
## 221   32 27 36 43.0  36.8  79.4   22  132  95  35  10  97 0.39233371
## 222   26 31 47 64.1  38.9  66.4  132  286 103  40 102 169 0.32708215
## 223   22 31 36 39.3  31.5  76.4   27  128 120  40  11  99 0.43687680
## 224   25 55 25 45.5  41.8  70.6   35  169  71  29  21  99 0.39692999
## 225   34 35 27 38.9  33.4  87.9   59  340 116  16  12  97 0.41064946
## 226   23 30 25 48.5  26.9  70.2  100  228  50  31  30 102 0.44541868
## 227   31 37 39 73.2   0.0  66.7  149  165  69  48  26 209 0.34707456
## 228   24  7 35 52.9  28.8  75.0   94  175  36  19  30  81 0.50883392
## 229   31 25  8 43.6  31.5  90.2   28  112 163  23   5  68 0.62062937
## 230   25 37 30 44.6  34.4  80.9   29  158  66  23  12  68 0.47310804
## 231   27 30 40 42.4  35.4  85.4   15   94  90  28   7  66 0.51484326
## 232   28 27 21 41.5  36.8  76.6   12   94 214  33   8  88 0.50913973
## 233   29 31 23 46.7  38.1  84.4   15  117  84  28  15 123 0.41341719
## 234   22 48 28 49.4  36.3  75.7   52  135  87  56  30 158 0.29650248
## 235   24 37 33 51.2  39.3  83.1   68  143  80  39  15 110 0.36919994
## 236   34 44 34 42.0  39.1  70.8   62  237 121  54  44 149 0.27959757
```

```
## 237   25 19 35 41.2  21.6  73.6   27  140 261  75  25 122 0.34105627
## 238   23 29 30 42.7  35.5  81.2   43   78  67  16  15  99 0.38620576
## 239   22 44 26 42.9  30.7  63.1   50  143  59  29  19 107 0.37028208
## 240   30 43 18 39.2  33.9  65.9   50  233  77  37  11  69 0.32300548
## 241   22 26 36 56.0  39.1  72.0   42  196 130  32  12 118 0.38845633
## 242   27 35 34 52.8  25.0  75.2  119  161  68  16  30 126 0.46601942
## 243   21 24 39 56.3  14.3  65.1  113  169  51  31  93 154 0.43653459
## 244   24 47 33 45.4  32.0  80.6   16   89 185  44  15 112 0.31098825
## 245   22 11 34 55.8  20.0  70.1   73  194  30   7  27 108 0.51891144
## 246   30 23 19 39.7  33.0  74.7   16  112 146  57  12  99 0.40611433
## 247   30 34 31 61.0  33.3  69.6  106  248 119  56  37 133 0.31938007
## 248   36 33 23 48.8  28.8  74.4   38  106  69  18  17  99 0.40051324
## 249   25 35 24 67.1   0.0  48.4  267  288  53  56 109 160 0.27379154
## 250   25 38 26 42.3  36.6  79.1   20  119  67  43  12 104 0.51248514
## 251   24 40 21 61.4  26.7  55.1  121  186  96  58  29  97 0.36352851
## 252   31 12 50 42.7  38.9  79.2   21   85 217  34   9  88 0.34788985
## 253   28 26 27 51.7  37.3  43.6   97  164 105  45  14 112 0.35240464
## 254   20 20 28 43.8  40.4  82.3   11   78  65  17   7  64 0.59288538
## 255   25 36 26 47.1  36.7  76.7   56  159  96  28  20 134 0.34728033
## 256   23 12 51 45.8  34.9  76.7   34  143  33  24  12  95 0.43606178
## 257   34 34 33 40.0  33.5  72.3   44  203 194  63  41 187 0.22846133
## 258   24 22 40 44.8  32.2  76.8   57  148 168  58  26  95 0.36819251
## 259   24 47 35 53.5  21.6  67.6  209  327 102  53  64 213 0.24935512
## 260   29 32 35 60.6   0.0  53.6  138  306  33  45  27 117 0.46990932
## 261   24 27 38 38.6  31.0  71.7   62  156  44  33  13  79 0.36742984
## 262   28 22 26 45.0  43.1  90.0   21   77  59  45  12  93 0.33519081
## 263   21 35 35 49.1  28.8  62.8   51  231  83  38  34 113 0.37922427
## 264   23 14 26 44.6  31.3  85.3   18   70  57  39  14  72 0.39106710
## 265   21 18 25 63.7   0.0  69.1   93  211  18  28  45  94 0.46739130
## 266   32 15 19 40.7  30.3  68.1   14   78 178  27  12  59 0.51146151
## 267   31 32 31 44.9  39.1  76.9   37  166  50  15  24  98 0.40524812
## 268   32 27 29 37.9  36.7  78.7   15  122 112  28  13  56 0.38167170
## 269   27 33 20 46.1  35.6  91.7   13   66  68  31  10  83 0.51344743
## 270   29 33 34 45.5  36.4  80.8   42  154 193  98  46 159 0.23740003
## 271   21 16 48 46.5  23.6  75.0   20   51  73  33  15  59 0.47316205
## 272   26 37 31 59.6  27.8  61.9  116  170  36  26  35 144 0.36335222
## 273   31 26 24 47.4  34.5  86.7   58  122 194  92  17  59 0.30216181
## 274   32 30 27 54.0  37.8  77.6   72  133  51  25  23 105 0.45939500
## 275   36 27 31 42.1  35.9  68.9   16   65 151  13   4  65 0.41954023
## 276   27 45 27 43.1  33.9  77.8   64  246 100  77  25 112 0.24367385
## 277   24 29 30 44.4  38.4  66.7   15   86 108  32  21  88 0.41067994
## 278   25 19 27 36.9  31.6  84.7   19   96  64  25   9  65 0.40347749
## 279   29 26 31 45.8  39.1  82.9   53  154  74  20   7  99 0.44913151
## 280   29 28 14 59.7   0.0  36.4  214  271  97  36  46  98 0.31853878
## 281   22 49 27 49.5  35.4  62.5   48  133  59  41  17 101 0.30480061
## 282   23 22 32 44.2  36.7  71.7   27  114  50  20  20  66 0.36779019
## 283   24 10 42 41.7  36.5  82.1   28  115  41  28  28 103 0.28864603
## 284   26 19 21 37.7  37.7  88.2   11   56  93  26   5  61 0.42880159
## 285   25 39 23 44.8  31.6  70.2   48  159  33  27  19  79 0.29272646
## 286   27 25 27 36.9  33.0  79.5   31   55  42  15   8  62 0.42831026
## 287   24 25 24 48.5  38.7  66.1   58  167  46  13  43 105 0.42035398
## 288   28 30 28 49.1  44.4  72.3   30  111  48  25  17  80 0.34705756
## 289   26 41 25 58.9  23.7  74.0   47  151  40  22  21  81 0.46589260
## 290   29 35 22 59.8   0.0  69.3   67   92  33  17  25  75 0.46982822
```

```
## 291   26 29 35 50.0   33.8   73.8    60    93  34   29   37   86 0.35706291
## 292   35 32 14 43.5   40.9   85.7    13   115 150   33    6   74 0.30378534
## 293   29 26 16 48.9   32.8   80.0    26    95  40   22   12   65 0.45725936
## 294   25 22 23 48.8   38.9   92.7     7    71 135   23    5   64 0.37886219
## 295   20 37 26 47.6   36.3   69.8    28    77  51   18    7   54 0.36959980
## 296   22  9 38 43.0   35.3   75.0    18    69  54   27    9   51 0.40288154
## 297   23 14 29 46.7   26.2   69.6    57    57  50   24   13   67 0.35177070
## 298   26 38 22 47.6   37.4   92.7    24   104  80   22   14   84 0.33408324
## 299   21 14 30 40.2   29.5   85.1    14   109 152   37   12   61 0.34555464
## 300   26 38 33 43.5   36.5   67.9    40   100  53   83   32  113 0.24408530
## 301   36 29 27 37.8   25.2   85.7    33   130  57   19   18   68 0.35924473
## 302   22 16 27 44.4   33.3   83.3    71   109  43   24   11   71 0.35885461
## 303   26 24 18 56.6    0.0   43.9    40   223 256   54   24  139 0.26334842
## 304   29  8 14 53.7   47.2   77.4     9    91 124   25   10   58 0.51101322
## 305   20 23 26 43.6   40.7   70.4    44   197  76   28   12  111 0.31502953
## 306   32 20 28 44.5   39.7   75.0    31   135  57   38   33   86 0.36989468
## 307   23 44 25 59.3   16.7   74.5   112   150  27   45   51  123 0.38003178
## 308   27  7 24 48.0   28.3   72.5    41    77  84   21    7  106 0.41771218
## 309   25 24 11 74.7    0.0   61.0   104   188  50   22   48   68 0.33879781
## 310   26 10 30 44.3   28.1   68.0    43    68  28   28    3   91 0.43080738
## 311   22 34 19 56.1   26.2   64.5    65    73  18   17   23  105 0.38059907
## 312   21 16 18 43.4   36.5   62.9    45   114  64   20   43   58 0.39355483
## 313   25 32 16 41.2   36.4   75.0    25    63  64   14    1   37 0.42003734
## 314   29 38 18 47.2   33.6   75.0     9    85  44   14    9   69 0.33209418
## 315   29 19 23 37.1   32.8   90.6     8    61  46   14    0   77 0.38812455
## 316   22 34 18 39.5   33.1   73.0    38    77  35   21    7   70 0.37924152
## 317   37 48 27 42.7   39.3   82.6    95   200  60   39   15  180 0.13853445
## 318   30 33 28 57.8    0.0   35.7    89   171  56   18   88  114 0.30088091
## 319   23 23 20 42.2   30.3   68.4    70   115  62   14   12   52 0.39866606
## 320   20 22 47 39.9   31.7   70.2    36   116  36   22   23   66 0.27043765
## 321   28 23 15 52.7   27.3   77.9    58   122  33   17   13   64 0.57242735
## 322   27 45 24 66.5   23.1   82.1    86   111  53   11   46   81 0.32458649
## 323   26 16 38 36.3   35.0   90.6    11    62  26   24    5   62 0.37447181
## 324   30 25 27 43.5   35.0   76.9     5    78  75   28    4   80 0.25091016
## 325   28 33 27 37.0   32.9   83.9    45    94  53   38   22  117 0.27428696
## 326   26 32 31 41.8   40.9   84.4    25    49  89   37   12   79 0.29220395
## 327   20 16 30 39.6   32.5   75.0    20    73  54   15    6   48 0.31257942
## 328   24 29 24 36.0   35.3   75.8    15    61  70   21    4   65 0.35610886
## 329   34 18 22 41.1   36.6   83.3     8    36  56   15    3   32 0.43416873
## 330   23 17 14 36.7   36.0   66.7     2    34  63    6    4   23 0.61960386
## 331   34 23 31 54.5   17.6   69.2    71    95  75   54    5   88 0.30169705
## 332   21  3  9 41.5   27.9   83.7    12    62  72   10    7   34 0.59720140
## 333   26 28 26 43.1   33.9   46.4    57   131  43   36   17   82 0.24344956
## 334   22 22 25 55.1   10.3   69.9    39    93  34   20   19   67 0.38190461
## 335   36 27 19 47.0   35.8   76.4    15    70 110   26    6   55 0.27841114
## 336   21 22 26 54.4   54.5   63.6    99   101  37   15   32   89 0.41085973
## 337   20 33 26 41.8   31.4   65.0    36   152 134   43   11   99 0.21782938
## 338   21 15 42 34.2   28.6   66.7    18    80 132   31    7   59 0.24715661
## 339   21 21 54 48.6   40.7   61.7   120   187  64   44   29  135 0.23191095
## 340   22 19 24 45.4   37.7   54.3    34   104  38   22   34   44 0.36351834
## 341   22 36 28 35.8   29.9   66.7    13    37  72   37    8   60 0.29202841
## 342   25 16 16 44.5   42.4   66.7    15    38  53   15    4   23 0.47446035
## 343   31 18 19 45.6   34.8   71.1    30   105  52   12   30   85 0.23661002
## 344   20 21 35 41.9   28.6   75.6    48    83  34   10   13   69 0.34883721
```

```
## 345  25 20 21 39.3  35.8  80.0   16   59  28  14  11  42 0.40316503
## 346  21 21 16 42.9  25.8  77.3   16   63  35  14   6  58 0.37446505
## 347  30 20 25 43.1  39.0  86.7   10   43  21  11   8  53 0.41886946
## 348  26 30 14 41.2  35.4  65.2   20  130  37  28  21  89 0.23104531
## 349  34 18 28 37.7  32.2  62.5    5   50  41  19  19  68 0.29032715
## 350  22  9 26 64.4  37.5  59.5   75  118  48  16  33  74 0.39180941
## 351  34 23 16 76.5 100.0  45.8   52  150  34  12  23  71 0.33964840
## 352  33 23 15 51.0  30.8  67.3   34  100  26   9  19  75 0.46691694
## 353  27 15 14 40.9  31.1  71.4   49   96  98  30  11  89 0.25264243
## 354  30 21 38 53.8  13.8  73.1   37   62  34   7  12  70 0.31219981
## 355  20 20 31 36.3  33.8  83.3   34   65  30  12   7  63 0.23762620
## 356  19 20 19 42.0  26.5  65.2   19   86  46  14   7  42 0.33937049
## 357  22  9 25 35.8  30.3  66.7    8   64  41  14   5  49 0.37051406
## 358  21 13 27 37.6  34.6  65.9   11   33  59  19   6  43 0.45389049
## 359  35 20 22 64.0  40.0  58.5   30   77  11   4  26  54 0.52468265
## 360  20  9 28 32.1  38.5  59.1   12   70  98  25   5  67 0.27499626
## 361  31  4 28 40.8  42.3  81.8   20   54  28  16   1  39 0.34930330
## 362  22 29 29 48.0   8.3  62.7   84   87  31  21  57 128 0.22698927
## 363  24 15 13 54.3  35.9  78.8   33   30  18   3   2  43 0.74363188
## 364  36 37 15 36.3  31.5  85.7   34   79  34  20  13  71 0.21818625
## 365  24 31 21 77.6   0.0  75.0   99  143  25  15  28  96 0.21910389
## 366  26  5 25 53.3  45.0  66.7   22   73  67  16   6  48 0.37060703
## 367  25 26 24 38.6  27.5  77.8   17   36  53   7   5  50 0.27506797
## 368  25  9 33 43.2  30.0  48.8   21   87  38  18   7  53 0.36317568
## 369  32 23 10 34.3  25.6  80.0    9   61 115  26   6  51 0.30212014
## 370  34 27 14 48.5  34.8  65.6   47  108  61  14   9  75 0.29882229
## 371  37 21 28 52.0  33.3  71.4   32   61  34  15  12  84 0.34970858
## 372  23 13 25 55.2  23.1  54.8   61   76  39  16  26  64 0.36558669
## 373  30 15 12 33.7  30.7  85.7    4   45  36  16   3  46 0.35971223
## 374  29 13 19 56.5  37.1  66.7   27   61  23  24  21  52 0.34353529
## 375  21  7 10 45.4  40.2  82.4   10   50  10   6   5  26 0.51546392
## 376  21 10 18 38.6  24.3  51.9   10   55  28  10   8  48 0.38570752
## 377  27 22 21 42.1  30.8  83.3   15   47 148  32   4  28 0.23728814
## 378  22 13 33 55.8  21.1  73.1   37   86  15  17  30  65 0.28566230
## 379  30 31 17 51.8  28.6  91.2   10   36  79  17   4  49 0.31076801
## 380  26 14 23 45.1  32.4  91.3   23   39  31   8   5  61 0.34345225
## 381  30 19 33 44.8  18.6  79.2   28   96  65  16  10  80 0.22139694
## 382  23 17 19 63.5   0.0  45.8   57   82   4   4  14  39 0.52703162
## 383  23 27  8 42.4  41.9 100.0    6   39  22   6   0  31 0.44650623
## 384  24 13 26 52.7   0.0  70.0   54   83  31   6  34  74 0.37768452
## 385  34 12 15 38.4  30.2  91.7    8   32  46   7   5  18 0.44983723
## 386  23 15 10 40.5  36.7  73.7   13   32  67  19   1  32 0.40428380
## 387  21  6 25 36.3  23.8  55.6    2   43  47  15   2  29 0.33510515
## 388  21  9 13 41.8  41.4  76.3    7   23  20   9   0  21 0.57591623
## 389  23 17 14 42.3  29.4  70.8   11   34  36  13  13  34 0.50987433
## 390  27 12 29 64.0  58.8  75.9   42   80  15   8  21  55 0.29744449
## 391  25 12 22 43.9  37.2  75.6   28   42  16  17   5  53 0.30853392
## 392  25 24 13 43.2  39.1  72.2    6   36  22  10   3  18 0.39603960
## 393  25 19 19 56.8  36.4  68.4   22   34  13  10   6  41 0.47815617
## 394  24 20 27 36.4  25.4  66.7    7   55  58  16   7  65 0.22218565
## 395  33 17 18 40.9  39.4 100.0   13   58  31   9   5  48 0.37172627
## 396  29 22 20 61.8  62.5  78.9   29   51   9   3  11  50 0.37399310
## 397  24  8 19 35.2  30.2  76.2   26   70  37  19  11  40 0.24710425
## 398  20 19 22 41.5  41.9  87.5    7   68  21   9   7  33 0.24980484
```

```
## 399  23 17 19 81.9  0.0  35.0  31  88  11  6 15 34 0.34867503
## 400  32 13  5 47.9 43.6  83.3  17  52  27 12  6 27 0.36502797
## 401  27 20 22 42.1 25.4  56.3  27  96  25 16  6 43 0.19885901
## 402  26 15 19 37.0 38.1  83.3   4  22  16  5  1 27 0.39329465
## 403  20 15 16 39.4 38.1  66.7   1  39  11  6  4 14 0.54030115
## 404  33  3 28 38.5 28.0  76.9  27  80  53  4 14 57 0.33789444
## 405  20 15  9 52.7 35.3  59.3  19  32  20  5  2 28 0.68181818
## 406  22 18 13 52.8  0.0  71.0  48  78  25 12 12 53 0.27287697
## 407  22 11 14 45.5 44.1  86.4   5  28  23 10  2 24 0.48551959
## 408  24 14 13 46.9 32.1  68.4   9  38  58  8  5 24 0.34029851
## 409  35 25 12 64.0 50.0  77.8  23  28  19  2  8 43 0.37520938
## 410  19  5 23 53.5  0.0  72.0  46  56  24 10 19 56 0.26987242
## 411  34 25 18 39.7 16.7  50.0   5  49 100  8  7 33 0.27142498
## 412  21 22 13 38.9 33.8  77.8  19  44  21 12  4 49 0.25343954
## 413  34  7 24 68.3  0.0  74.1  20  40   9  5  2 10 0.59614261
## 414  30 13  9 59.4 50.0  85.7  24  45  30 22  5 43 0.27475517
## 415  22 17 19 38.7 30.9  73.3  23  38  23 14  6 40 0.25920166
## 416  30  6  9 62.7  0.0  68.6  25  39  10  3  4 33 0.45244691
## 417  24 10  8 56.1 35.0  77.3  15  37  12 11  1 21 0.40833333
## 418  27 19 21 35.6 35.5  84.6  10  53  46  6  1 34 0.29411765
## 420  29  9 16 36.1 30.4 100.0   4  11  17  7  2 22 0.33246171
## 421  27 29 14 33.3 37.0  66.7  10  49  23 15  4 39 0.25985401
## 422  30 13 23 49.2 39.4  87.5   7  45  32  6  4 22 0.37888463
## 423  23 13 18 52.0 26.3  54.5  27  20  10  3  6 26 0.33788914
## 424  22  6 12 33.7 23.8  50.0   8  31   8  4  3 13 0.37451571
## 425  20 19 19 44.4 25.9  66.7  12  25  22 11  5 24 0.39738195
## 426  23 10 13 46.2  8.3  70.6  11  18  21 10  1 31 0.37280702
## 427  19  7  8 40.6 22.9  94.7   9  22  10  7  2 19 0.40097800
## 428  20 22 14 42.2 13.3  46.2   9  29  58 12  5 15 0.27708703
## 429  20 13 10 49.2 42.9  55.0   8  29  11  2 11 14 0.40300913
## 432  33 11  5 48.1 33.3  61.5  15  29   4  2  7 23 0.35598706
## 433  35  6  5 44.9 43.2 100.0   3  11   6  6  3 16 0.43979813
## 434  25 17  8 43.9 31.3  66.7   6  17  31  9  3 13 0.23579436
## 435  21 10 12 51.1 50.0  62.5  31  19   4  7  8 15 0.30179028
## 436  26 15  7 48.0 33.3 100.0   8  22   8  4  9 26 0.32404775
## 437  22  5 10 50.0 40.0  25.0   6  15   2  2  0 11 0.51708218
## 439  25  8  3 41.5 40.0  55.6  15  29   6 14  4  5 0.44354839
## 441  24  9 14 41.7  0.0  66.7   5  32  38  7  2 17 0.18772563
## 442  36  7 11 44.9 20.0  50.0   4  26  15  7  6 32 0.31403941
## 444  30 25 12 43.5  0.0  50.0  15  29  15  3  3 22 0.24733269
## 445  36 16  9 40.0 42.3  75.0   5  12  13  9  2 15 0.30902349
## 448  32 19 13 34.0 33.3  57.1   1  12  41  7  0 20 0.22086466
## 449  20  7  8 50.0 33.3 100.0   5  20   5  4  6 10 0.47817048
## 450  24 19 16 42.1 33.3  87.5   7  27  13  7  7 39 0.18548387
## 451  23  2 12 38.6 25.0 100.0  11  26   4  4  1 23 0.22738757
## 452  20  4 15 32.1 22.9  75.0   3  14  11  7  9 22 0.20352782
## 453  25  7  8 40.0 26.3 100.0   2  22  22 10  4 23 0.27777778
## 454  24  5  5 39.5  8.3  69.2  10  28   6  6  2 14 0.32140248
## 457  29 15 11 53.3  0.0  68.8  24  36  12  4 11 28 0.26674938
## 458  30  9 11 59.4 50.0  80.0   8  17   7  5  5 24 0.26576020
## 460  20 13  3 48.6 40.0  75.0   6   8   8  7  3 15 0.47032475
## 461  21 16 13 41.9 11.8  80.0   4  14  18  8  5 13 0.27504912
## 462  23  4 10 43.9 27.8   0.0   2  20  27  2  0  7 0.33117932
## 464  27  2  8 60.7  0.0  40.0  16  22   6  6  4 13 0.30206677
```

```
## 467  22 15 14 40.5  23.8  50.0    6   20  17   5   1  17 0.21765417
## 469  29  4 13 36.1  50.0  66.7   10   36  10  16  10  32 0.17976374
## 470  23  8  4 42.9  37.5 100.0    4   16   3   4   3   8 0.36817102
## 471  22  7  8 30.0  14.3 100.0    4   11   4   6   2  12 0.28680688
## 472  21 10  6 39.3  14.3  41.7    2   17   3   4   5  13 0.30075188
## 476  29  9 11 24.3  34.8   0.0    0   15  23   5   0  17 0.15081206
## 478  27 17  6 45.8  25.0 100.0   18   30   7   2   8  35 0.14568765
## 481  23  7  9 43.8  50.0  57.1    4    4   6   5   1  10 0.33850494
## 483  20  5  7 35.0  33.3 100.0    3    9   6   1   1  10 0.37037037
## 485  26 11  5 30.3  12.5   0.0    1   13  10   6   1   7 0.27742749
## 486  25 13  4 42.1  33.3  66.7    6   11   4   3   2   9 0.31818182
## 488  24 10  6 39.1  14.3 100.0    5    7   2   1   0   2 0.32061069
## 490  28 15  8 25.9  25.0  50.0    2   15   9   4   4   7 0.18621974
## 493  20 11  1 42.1  28.6  50.0    2    7  11   2   2   5 0.28919330
## 494  25  5  7 50.0  60.0 100.0    1   10   1   1   1   2 0.56886228
## 499  23  2  8 18.5  16.0 100.0    2    4   4   2   0   6 0.17758047
## 518  22  8  6 15.0   0.0   0.0    4    3   7   3   1  10 0.08759124
## 521  22  8  3 25.0   0.0 100.0    3    8   3   2   2   5 0.21052632
```

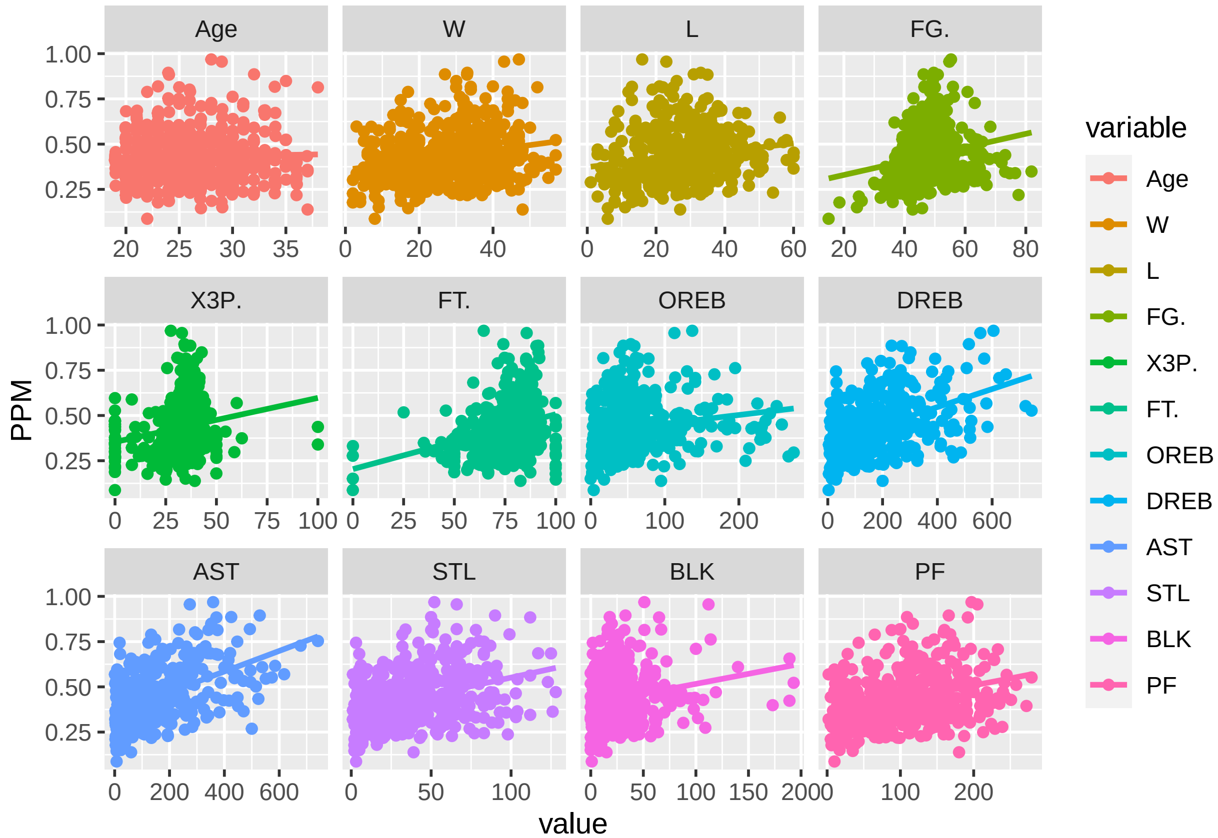
```r
#numerical_data <- clean_data[, !colnames(clean_data) %in% c("PName", "POS", "Team")]
model <- lm(PPM ~ Age + GP + W + L +
              Min + PTS + FGM + FGA + FG. + X3PM + X3PA + X3P.
            + FTM + FTA + FT. + OREB + DREB + REB + AST +
              STL + BLK + PF,data = clean_data)


summary(model)
```

```
##
## Call:
## lm(formula = PPM ~ Age + GP + W + L + Min + PTS + FGM + FGA +
##     FG. + X3PM + X3PA + X3P. + FTM + FTA + FT. + OREB + DREB +
##     REB + AST + STL + BLK + PF, data = clean_data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.232255 -0.030445  0.002664  0.027428  0.279031
##
## Coefficients: (3 not defined because of singularities)
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.200e-01  3.253e-02   3.689 0.000252 ***
## Age         -8.234e-04  6.715e-04  -1.226 0.220773
## GP           2.585e-04  3.871e-04   0.668 0.504568
## W           -2.783e-04  4.166e-04  -0.668 0.504579
## L                   NA         NA      NA       NA
## Min         -2.337e-04  1.770e-05 -13.204  < 2e-16 ***
## PTS         -1.780e-04  2.444e-04  -0.728 0.466688
## FGM          8.565e-04  5.268e-04   1.626 0.104685
## FGA          3.027e-04  1.090e-04   2.778 0.005700 **
## FG.          3.758e-03  5.030e-04   7.470 4.21e-13 ***
## X3PM         1.070e-03  5.432e-04   1.970 0.049441 *
## X3PA        -1.522e-04  1.984e-04  -0.768 0.443168
## X3P.         1.037e-03  2.656e-04   3.906 0.000108 ***
## FTM                 NA         NA      NA       NA
```

```
## FTA          4.623e-04  2.058e-04   2.246 0.025175 *
## FT.          1.023e-03  2.381e-04   4.297 2.12e-05 ***
## OREB         1.873e-07  1.209e-04   0.002 0.998765
## DREB         1.690e-05  5.712e-05   0.296 0.767432
## REB                 NA         NA      NA       NA
## AST         -9.529e-06  4.018e-05  -0.237 0.812654
## STL          1.149e-06  2.022e-04   0.006 0.995470
## BLK         -3.849e-05  1.589e-04  -0.242 0.808659
## PF           1.413e-04  1.104e-04   1.281 0.200989
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05852 on 450 degrees of freedom
## Multiple R-squared:  0.8453, Adjusted R-squared:  0.8387
## F-statistic: 129.4 on 19 and 450 DF,  p-value: < 2.2e-16
```

```r
library(ggplot2)
library(reshape2)

reshaped <- melt(mydata,id.vars="PPM")

ggplot(reshaped) +
  geom_jitter(aes(value, PPM, colour = variable)) +
  geom_smooth(aes(value, PPM, colour = variable), method = "lm", se = FALSE) +
  facet_wrap(~variable, scales = "free_x")
```

```
## `geom_smooth()` using formula = 'y ~ x'
```




```r
Rvalues <- c()  # Initialize an empty vector to store R-squared values

for (i in colnames(mydata)) {
  if (i != "PPM")
  {
    model <- lm(as.formula(paste("PPM ~", i)), data = mydata)
    Rvalues <- c(Rvalues, summary(model)$r.squared)
  }
}

cols_new <- c("Age","W","L","FG.","X3P.","FT.","OREB","DREB","AST","STL","BLK","PF")

result <- cbind(cols_new, Rvalues)

result_df <- as.data.frame(result)
colnames(result_df) <- c("Variable", "R_Squared")

result_df
```

```
##    Variable            R_Squared
## 1       Age 0.000749702068109721
## 2         W    0.056403578267626
## 3         L   0.0294548997406518
## 4       FG.    0.047622369618905
## 5      X3P.   0.0405741971809625
```

```
## 6       FT.   0.0816683370017414
## 7      OREB   0.0305522519088502
## 8      DREB    0.213186294659351
## 9       AST    0.262151918677083
## 10      STL    0.130221557082557
## 11      BLK   0.0428525323589375
## 12       PF    0.109345001867128
```

```
lm_initial <- lm(PPM ~ Age + W + L + FG.+ X3P.
            + FT. + OREB + DREB + AST +
              STL + BLK + PF,data = mydata)


summary(lm_initial)
```

```
##
## Call:
## lm(formula = PPM ~ Age + W + L + FG. + X3P. + FT. + OREB + DREB +
##     AST + STL + BLK + PF, data = mydata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43210 -0.07219  0.00254  0.06152  0.32475
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.151e-01  5.285e-02   2.179 0.029859 *
## Age         -4.095e-03  1.211e-03  -3.381 0.000785 ***
## W           -1.974e-03  5.792e-04  -3.409 0.000711 ***
## L           -1.408e-03  6.216e-04  -2.265 0.023954 *
## FG.          4.769e-03  7.543e-04   6.322 6.16e-10 ***
## X3P.         1.454e-03  4.698e-04   3.095 0.002092 **
## FT.          1.890e-03  4.060e-04   4.654 4.27e-06 ***
## OREB        -9.725e-04  2.035e-04  -4.780 2.37e-06 ***
## DREB         6.764e-04  8.505e-05   7.953 1.45e-14 ***
## AST          3.906e-04  6.488e-05   6.021 3.56e-09 ***
## STL         -1.100e-04  3.416e-04  -0.322 0.747628
## BLK          1.873e-04  2.865e-04   0.654 0.513685
## PF          -1.250e-04  1.903e-04  -0.657 0.511727
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1083 on 457 degrees of freedom
## Multiple R-squared:  0.4616, Adjusted R-squared:  0.4474
## F-statistic: 32.65 on 12 and 457 DF,  p-value: < 2.2e-16
```

```
lm_1 <- lm(PPM ~Age + W + L + FG.+ X3P.
            + FT. + OREB +DREB+ AST + BLK + PF,data = mydata)


summary(lm_1)
```

```
##
```

```
## Call:
## lm(formula = PPM ~ Age + W + L + FG. + X3P. + FT. + OREB + DREB +
##     AST + BLK + PF, data = mydata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43025 -0.07180  0.00270  0.06247  0.32551
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.146e-01  5.276e-02   2.171 0.030435 *
## Age         -4.074e-03  1.208e-03  -3.372 0.000811 ***
## W           -2.016e-03  5.641e-04  -3.573 0.000390 ***
## L           -1.444e-03  6.111e-04  -2.363 0.018558 *
## FG.          4.796e-03  7.490e-04   6.403 3.77e-10 ***
## X3P.         1.451e-03  4.693e-04   3.093 0.002103 **
## FT.          1.884e-03  4.053e-04   4.649 4.37e-06 ***
## OREB        -9.764e-04  2.029e-04  -4.812 2.03e-06 ***
## DREB         6.796e-04  8.437e-05   8.056 6.89e-15 ***
## AST          3.790e-04  5.394e-05   7.027 7.70e-12 ***
## BLK          1.813e-04  2.857e-04   0.635 0.526015
## PF          -1.393e-04  1.848e-04  -0.754 0.451392
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1082 on 458 degrees of freedom
## Multiple R-squared:  0.4615, Adjusted R-squared:  0.4485
## F-statistic: 35.68 on 11 and 458 DF,  p-value: < 2.2e-16
```

```r
lm_temp <- lm(PPM ~Age + W + L + FG.+ X3P.
            + FT. + OREB +DREB+ AST + BLK,data = mydata)


summary(lm_temp)
```

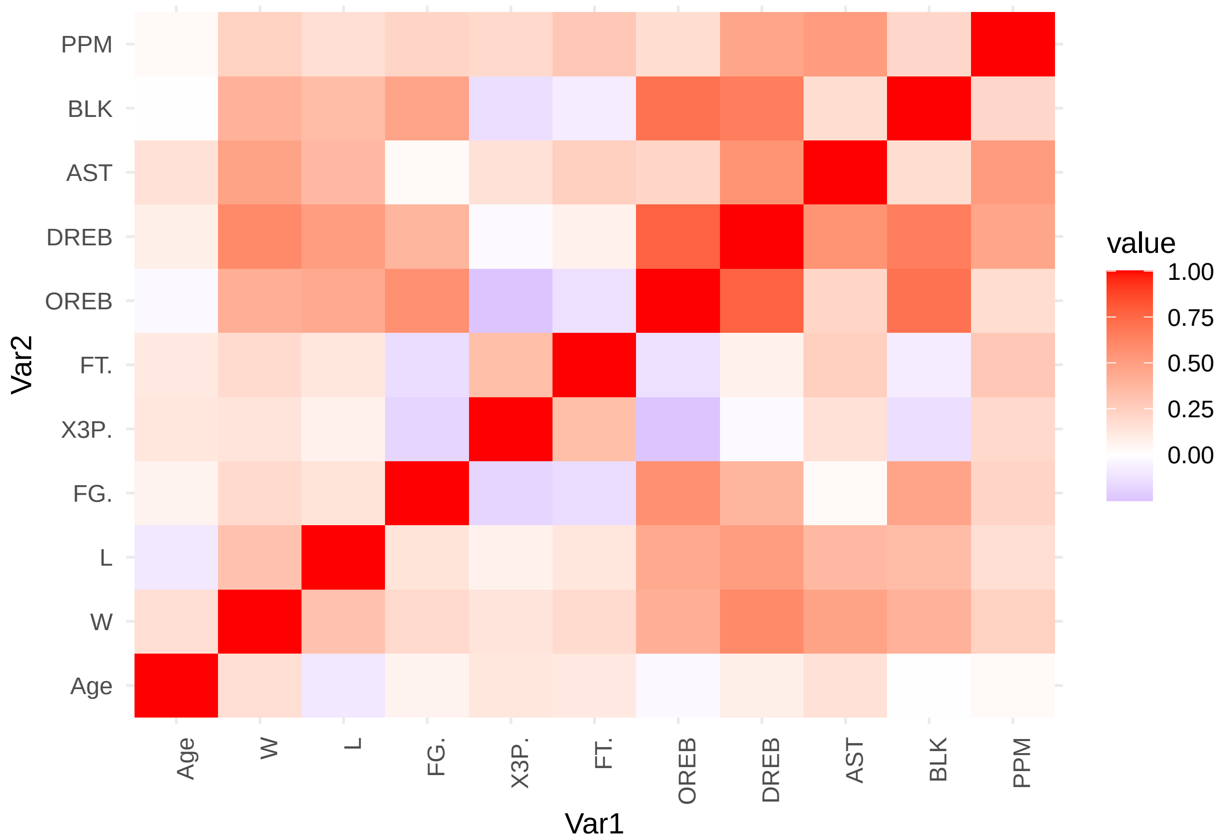
```
##
## Call:
## lm(formula = PPM ~ Age + W + L + FG. + X3P. + FT. + OREB + DREB +
##     AST + BLK, data = mydata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.43534 -0.07247  0.00263  0.06347  0.32400
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.162e-01  5.269e-02   2.205 0.027959 *
## Age         -4.052e-03  1.207e-03  -3.356 0.000856 ***
## W           -2.177e-03  5.216e-04  -4.175 3.57e-05 ***
## L           -1.670e-03  5.323e-04  -3.136 0.001820 **
## FG.          4.833e-03  7.470e-04   6.471 2.51e-10 ***
## X3P.         1.435e-03  4.686e-04   3.063 0.002322 **
## FT.          1.868e-03  4.045e-04   4.618 5.04e-06 ***
## OREB        -9.998e-04  2.004e-04  -4.988 8.65e-07 ***
```

```
## DREB         6.658e-04  8.232e-05   8.089 5.42e-15 ***
## AST          3.705e-04  5.273e-05   7.028 7.64e-12 ***
## BLK          1.235e-04  2.750e-04   0.449 0.653716
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1082 on 459 degrees of freedom
## Multiple R-squared:  0.4608, Adjusted R-squared:  0.449
## F-statistic: 39.22 on 10 and 459 DF,  p-value: < 2.2e-16
```

```r
anova(lm_temp,lm_initial)
```

```
## Analysis of Variance Table
##
## Model 1: PPM ~ Age + W + L + FG. + X3P. + FT. + OREB + DREB + AST + BLK
## Model 2: PPM ~ Age + W + L + FG. + X3P. + FT. + OREB + DREB + AST + STL +
##     BLK + PF
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    459 5.3702
## 2    457 5.3623  2 0.0078695 0.3353 0.7153
```

```r
library(ggplot2)
library(reshape2)

library(MASS)

#install.packages("car")  # If not installed already

final_data <- mydata[, !colnames(mydata) %in% c("STL","PF")]

# Reshape the correlation matrix into a long format
corr_melted <- melt(cor(final_data))

# Create a heatmap
ggplot(corr_melted, aes(Var1, Var2, fill = value)) +
  geom_tile() +
  scale_fill_gradient2(midpoint = 0, low = "blue", high = "red", mid = "white") +
  theme_minimal() +
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```




```r
library(car)
```

```
## Warning: package 'car' was built under R version 4.3.3
```

```
## Loading required package: carData
```

```r
vif(lm_temp)
```

```
##      Age        W        L      FG.     X3P.      FT.     OREB     DREB
## 1.107785 1.734695 1.493464 1.573205 1.280944 1.234448 4.458350 4.922991
##      AST      BLK
## 1.918789 2.235989
```

```r
lm_final <- lm(PPM ~Age+W+L+ FG.+ X3P.
            + FT. + OREB +DREB+ AST + PF,data = mydata)
vif(lm_final)
```

```
##      Age        W        L      FG.     X3P.      FT.     OREB     DREB
## 1.108250 2.027394 1.943167 1.540963 1.282716 1.235277 4.425903 4.998717
##      AST       PF
## 1.919888 4.792338
```

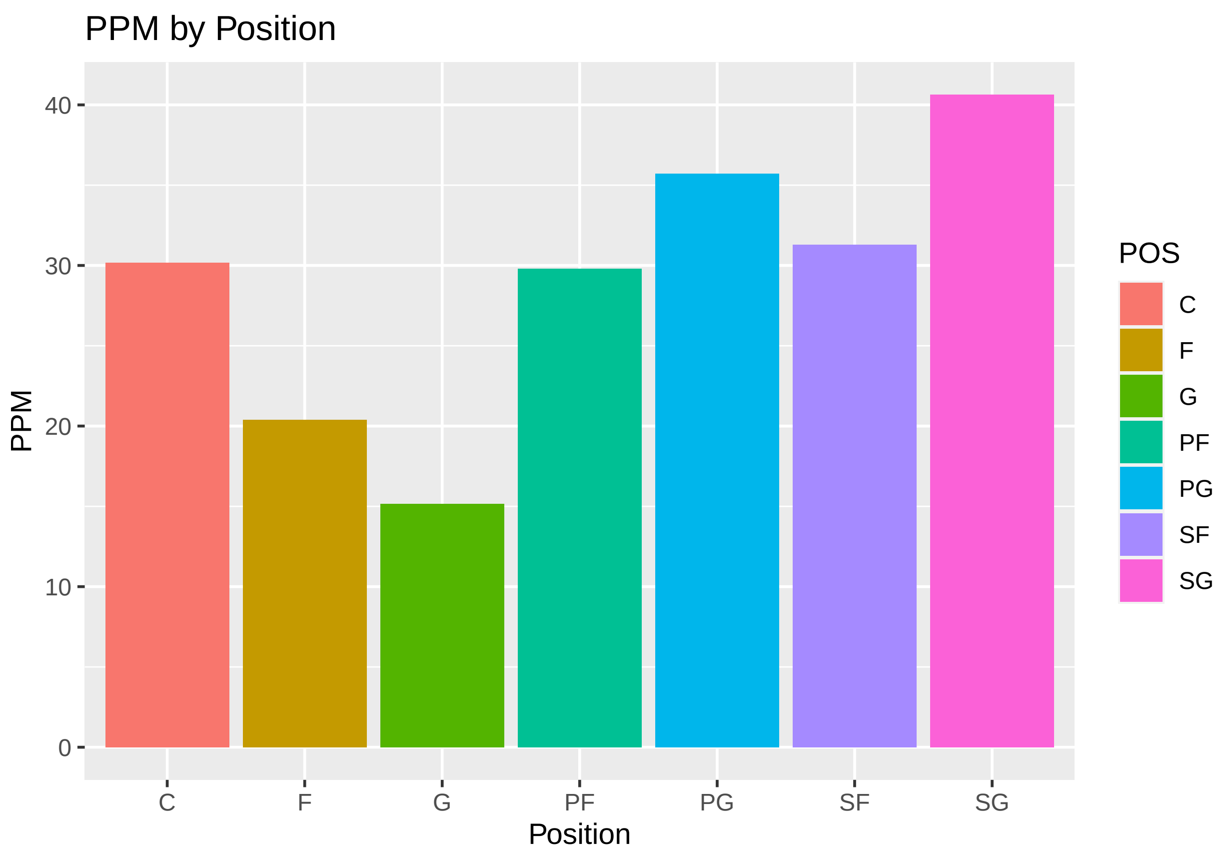
```r
summary(lm_final)
```

```
##
## Call:
## lm(formula = PPM ~ Age + W + L + FG. + X3P. + FT. + OREB + DREB +
##     AST + PF, data = mydata)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.42813 -0.06986  0.00274  0.06121  0.32333
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.128e-01  5.266e-02   2.142 0.032705 *
## Age         -4.083e-03  1.207e-03  -3.382 0.000782 ***
## W           -2.018e-03  5.638e-04  -3.579 0.000381 ***
## L           -1.486e-03  6.071e-04  -2.447 0.014770 *
## FG.          4.871e-03  7.391e-04   6.590 1.21e-10 ***
## X3P.         1.443e-03  4.688e-04   3.079 0.002204 **
## FT.          1.872e-03  4.046e-04   4.628 4.81e-06 ***
## OREB        -9.539e-04  1.997e-04  -4.778 2.39e-06 ***
## DREB         6.893e-04  8.293e-05   8.311 1.08e-15 ***
## AST          3.719e-04  5.273e-05   7.053 6.48e-12 ***
## PF          -1.078e-04  1.779e-04  -0.606 0.544820
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1081 on 459 degrees of freedom
## Multiple R-squared:  0.461,  Adjusted R-squared:  0.4492
## F-statistic: 39.25 on 10 and 459 DF,  p-value: < 2.2e-16
```

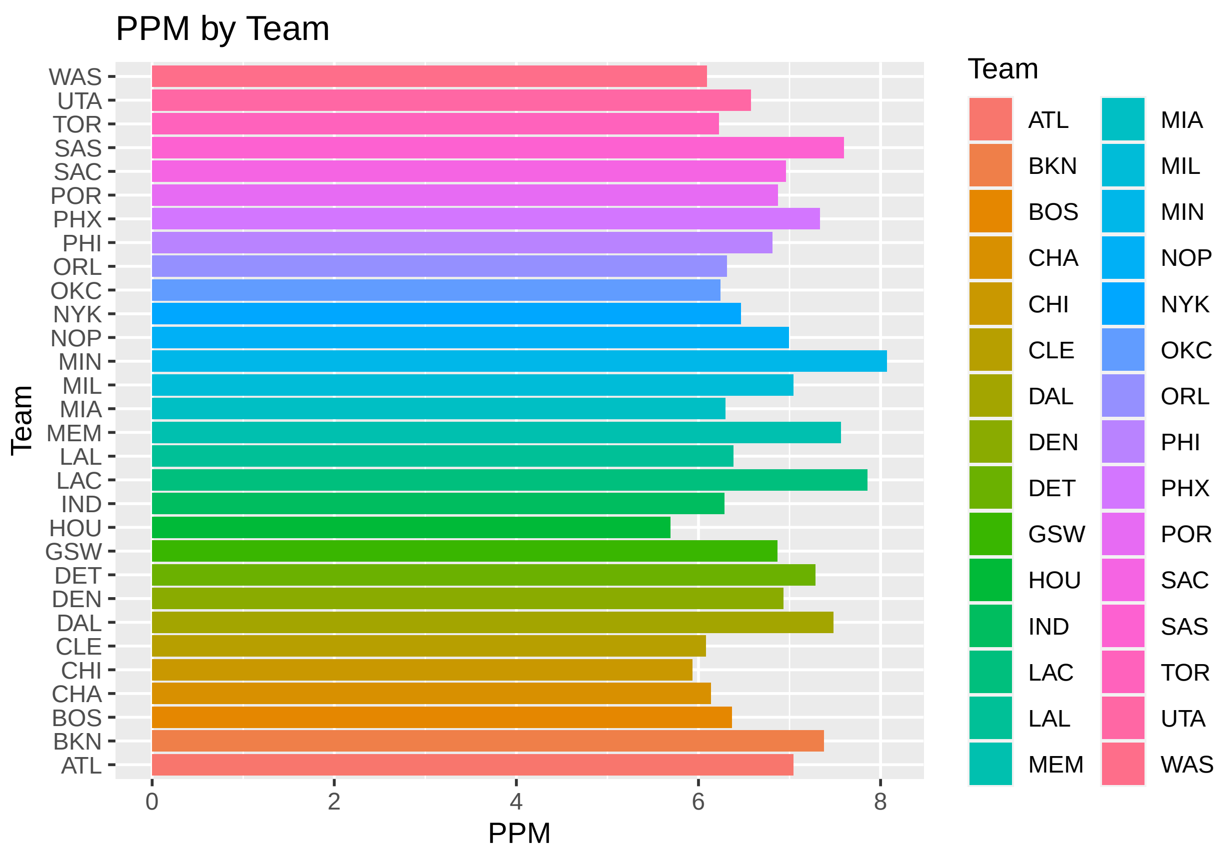
```r
anova(lm_final,lm_initial)
```

```
## Analysis of Variance Table
##
## Model 1: PPM ~ Age + W + L + FG. + X3P. + FT. + OREB + DREB + AST + PF
## Model 2: PPM ~ Age + W + L + FG. + X3P. + FT. + OREB + DREB + AST + STL +
##     BLK + PF
##   Res.Df    RSS Df Sum of Sq      F Pr(>F)
## 1    459 5.3682
## 2    457 5.3623  2 0.0059322 0.2528 0.7767
```

```r
# Plot PPM by Position
ggplot(data = clean_data, aes(x = POS, y = PPM, fill = POS)) +
  geom_bar(stat = "identity") +
  labs(title = "PPM by Position", x = "Position", y = "PPM")
```




```r
# Plot PPM by Team
ggplot(data = clean_data, aes(x = PPM, y = Team, fill = Team)) +
  geom_bar(stat = "identity") +
  labs(title = "PPM by Team", x = "PPM", y = "Team")
```

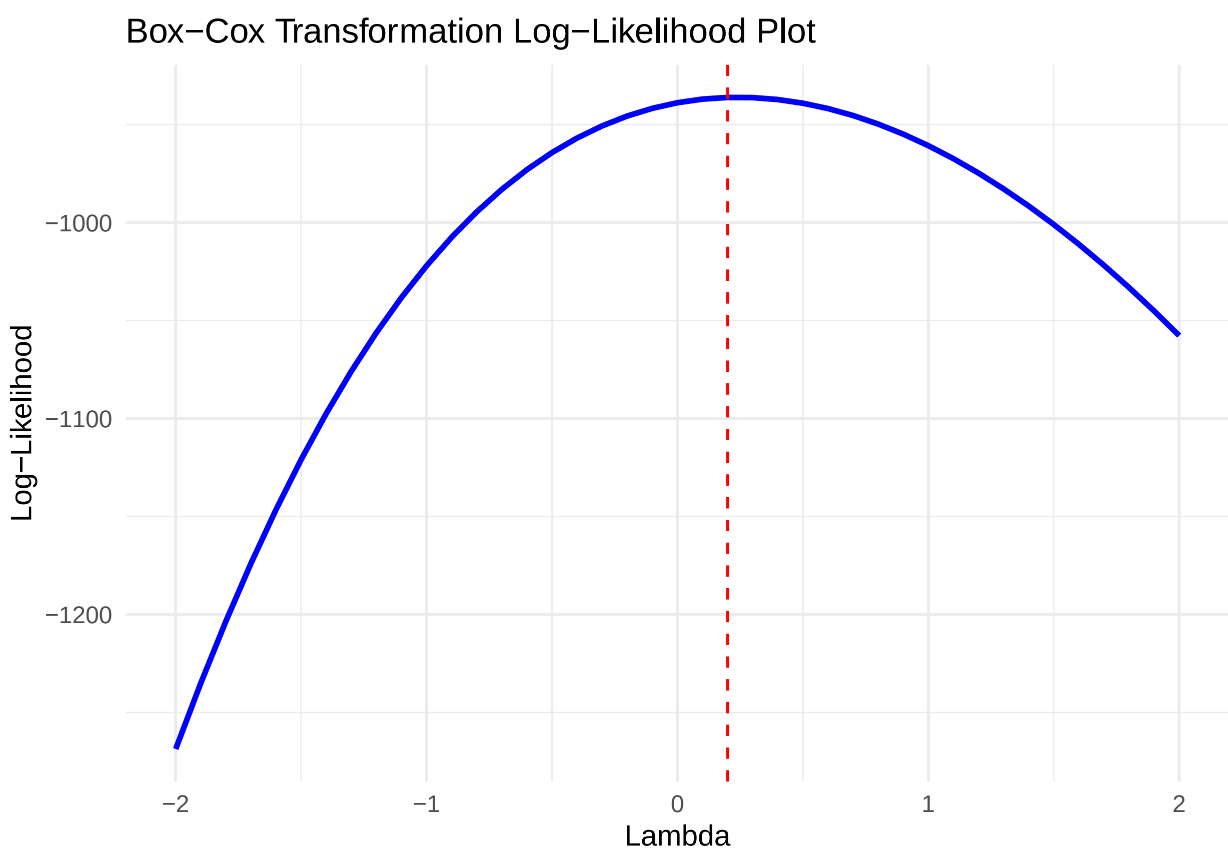



```r
# Load necessary libraries
library(MASS)   # For Box-Cox transformation
library(ggplot2)  # For visualization
library(dplyr)  # For data manipulation
```

```
##
## Attaching package: 'dplyr'

## The following object is masked from 'package:car':
##
##     recode

## The following object is masked from 'package:MASS':
##
##     select

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
# Apply Box-Cox transformation to determine the optimal lambda
boxcox_result <- boxcox(PPM ~ 1, data = mydata, lambda = seq(-2, 2, 0.1), plotit = FALSE)

# Extract lambda values and their corresponding log-likelihoods
lambda_values <- boxcox_result$x
log_likelihoods <- boxcox_result$y

# Plot the Box-Cox transformation log-likelihood against lambda values
boxcox_plot <- data.frame(lambda = lambda_values, log_likelihood = log_likelihoods)
ggplot(boxcox_plot, aes(x = lambda, y = log_likelihood)) +
  geom_line(color = "blue", size = 1) +
  geom_vline(xintercept = lambda_values[which.max(log_likelihoods)], linetype = "dashed", color = "red"
  labs(
    title = "Box-Cox Transformation Log-Likelihood Plot",
    x = "Lambda",
    y = "Log-Likelihood"
  ) +
  theme_minimal()
```

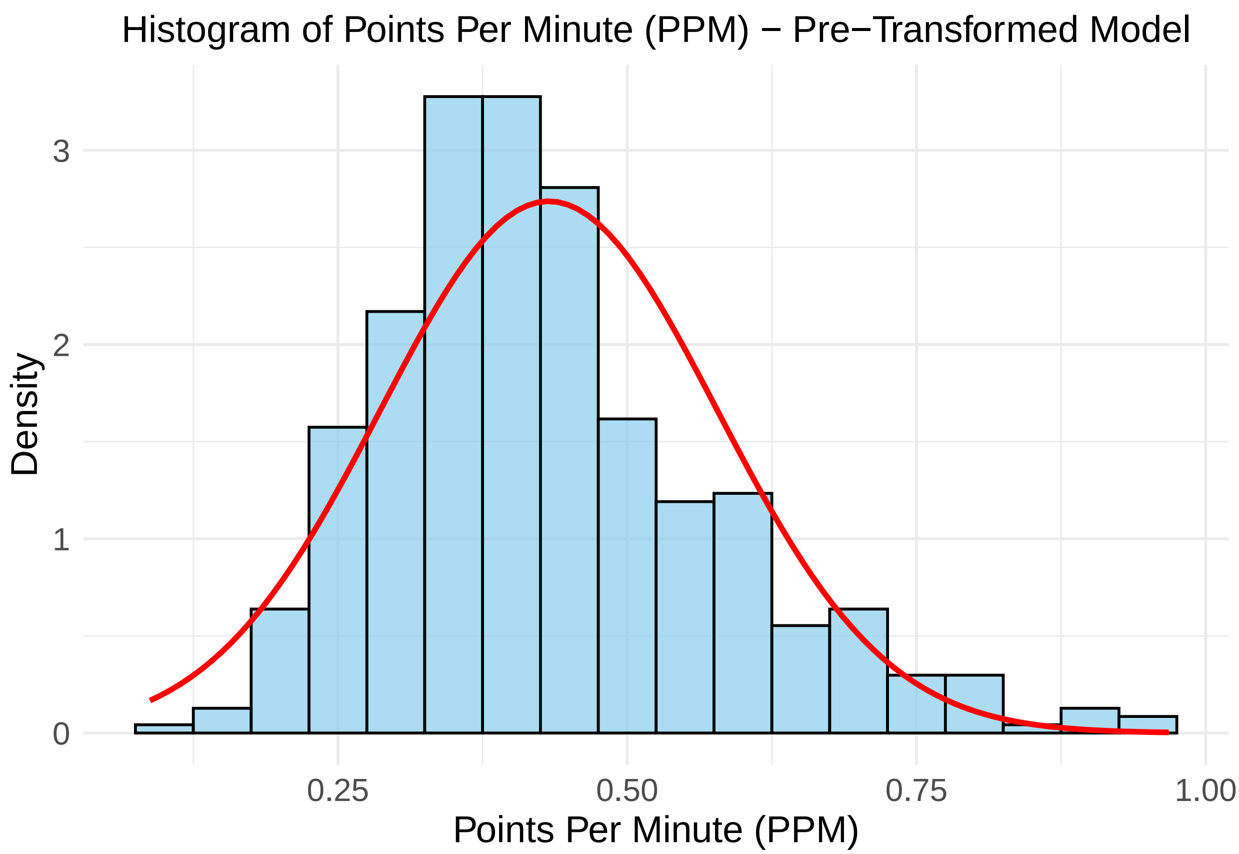
```
## Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
## i Please use `linewidth` instead.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```



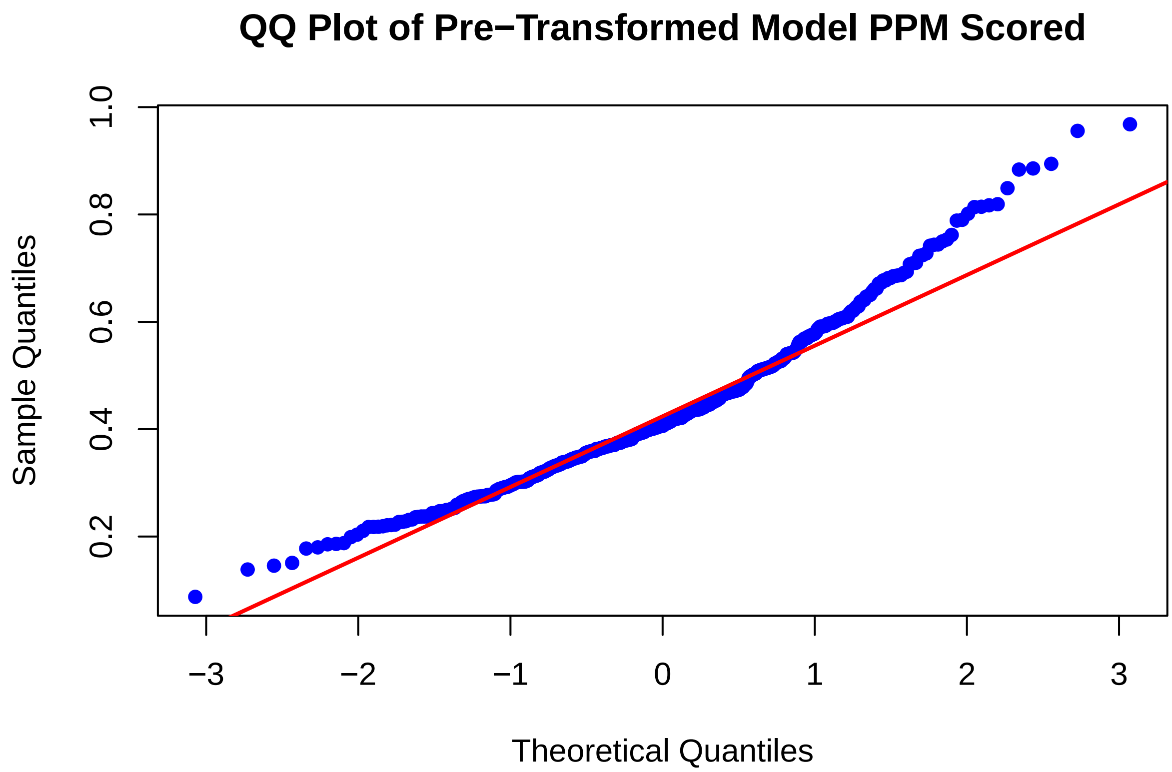
```
# Display the optimal lambda
lambda_optimal <- lambda_values[which.max(log_likelihoods)]
lambda_optimal
```

```
## [1] 0.2
```

```
# Plot histogram of PPM with density curve
library(ggplot2)

original_ppm_plot <- ggplot(mydata, aes(x = PPM)) +
  # Histogram with density scaling
  geom_histogram(aes(y = after_stat(density)), binwidth = 0.05, fill = "skyblue", color = "black", alpha
  # Overlay normal density curve
  stat_function(
    fun = dnorm,
    args = list(mean = mean(mydata$PPM, na.rm = TRUE), sd = sd(mydata$PPM, na.rm = TRUE)),
    color = "red", size = 1
  ) +
  # Titles and labels
  labs(
    title = "Histogram of Points Per Minute (PPM) - Pre-Transformed Model",
    x = "Points Per Minute (PPM)",
    y = "Density"
  ) +
  # Styling
  theme_minimal() +
  theme(
    axis.text = element_text(size = 12),
    axis.title = element_text(size = 14),
    plot.title = element_text(size = 14, hjust = 0.5)
  )

# Print the plot
print(original_ppm_plot)
```




```r
# Create a QQ plot for the Pre-Transformed Model for 'PPM' variable
qqnorm(mydata$PPM,
       main = "QQ Plot of Pre-Transformed Model PPM Scored",
       xlab = "Theoretical Quantiles",
       ylab = "Sample Quantiles",
       col = "blue",
       pch = 16)
qqline(mydata$PPM, col = "red", lwd = 2)
```

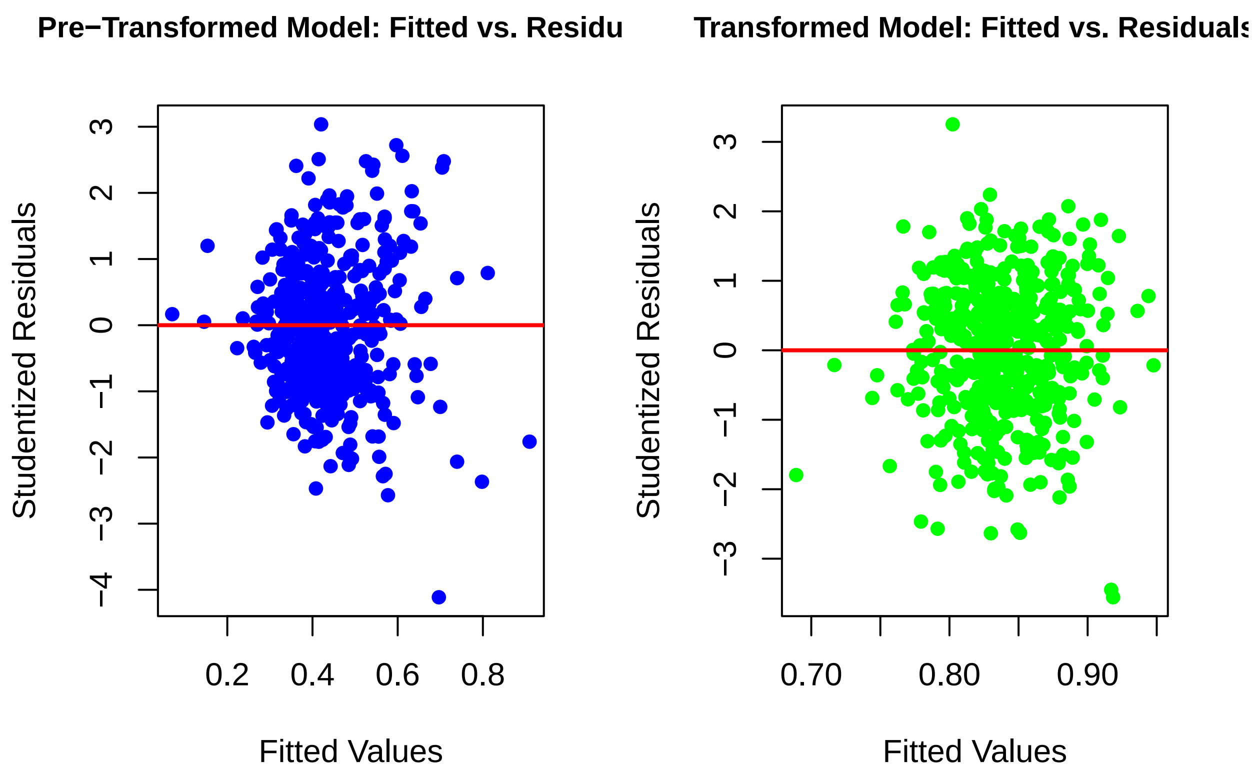



```r
# Load necessary libraries
library(lmtest)
```

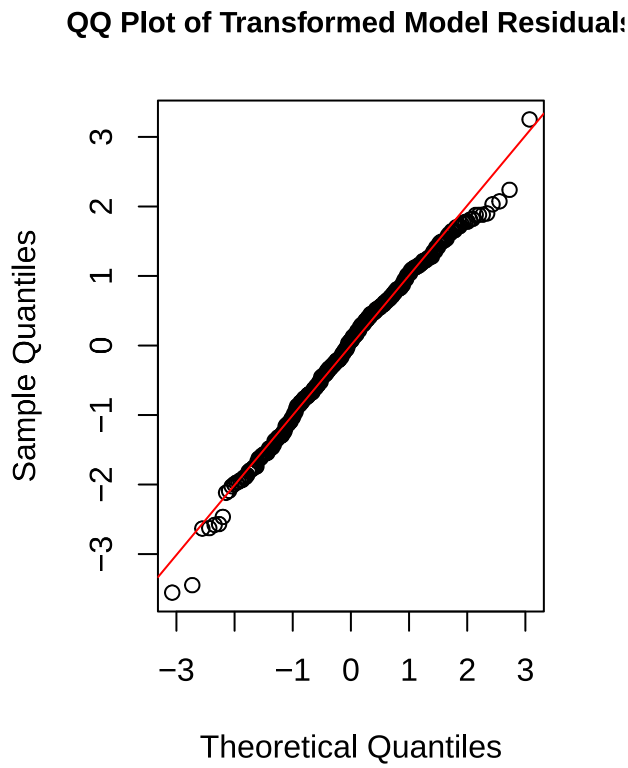
```
## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

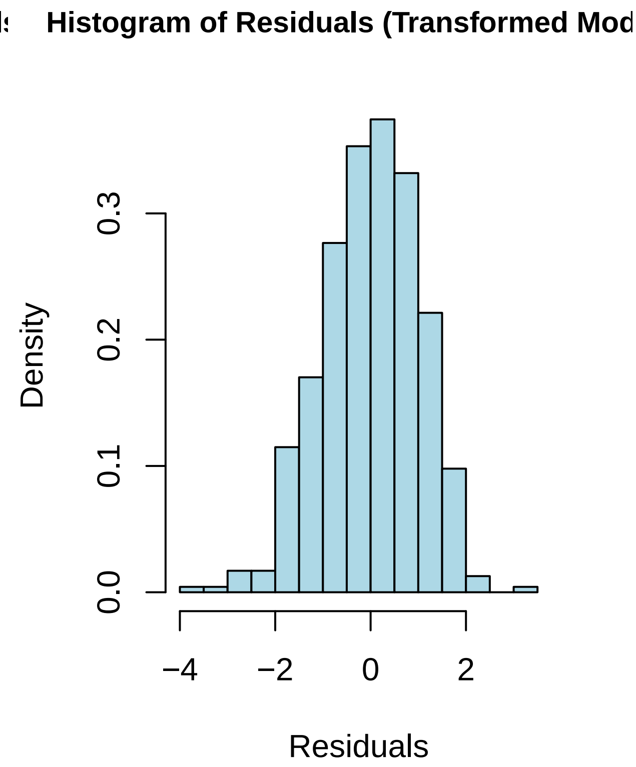
```r
library(MASS)
library(ggplot2)

# Apply Box-Cox Transformation to Dependent Variable (PPM) by Optimal Lambda Value
mydata$PPPM_transformed <- mydata$PPM^(0.2)

# Transformed Model with Log Transformations
# Adding 1 to avoid log(0)
# Log Transformations
mydata$FG._log <- (log(mydata$FG. + 1)^2)
mydata$W_log <- (log(mydata$W + 1)^2)
mydata$L_log <- (log(mydata$L + 1)^2)
```

```r
mydata$Age_log <- (log(mydata$Age + 1)^2)
mydata$OREB_log <- (log(mydata$OREB + 1)^2)
mydata$DREB_log <- (log(mydata$DREB + 1)^2)
mydata$AST_log <- (log(mydata$AST + 1)^2)

# Fit the Transformed Model
model_transformed <- lm(
  PPPM_transformed ~  FG._log + W_log + L_log + Age_log + (FG._log)^2 + OREB_log + DREB_log + AST_log,
  data = mydata
)

# Plot Residual Analysis - Fitted vs. Studentized Residuals

# Set up side-by-side plotting
par(mfrow = c(1, 2))  # 1 row, 2 columns

# Plot for lm_final (pre-transformed)
plot(
  fitted(lm_final), rstudent(lm_final),
  main = "Pre-Transformed Model: Fitted vs. Residuals",
  xlab = "Fitted Values", ylab = "Studentized Residuals",
  col = "blue", pch = 16,
  xlim = c(min(fitted(lm_final)), max(fitted(lm_final))), cex.main = 0.9
)
abline(h = 0, col = "red", lwd = 2)

# Plot for Transformed Model
plot(
  fitted(model_transformed), rstudent(model_transformed),
  main = "Transformed Model: Fitted vs. Residuals",
  xlab = "Fitted Values", ylab = "Studentized Residuals",
  col = "green", pch = 16,
  xlim = c(min(fitted(model_transformed)), max(fitted(model_transformed))), cex.main = 0.9
)
abline(h = 0, col = "red", lwd = 2)
```




```r
# Extract residuals from the transformed model
residuals_transformed <- rstudent(model_transformed)

# QQ Plot of residuals
qqnorm(residuals_transformed, main = "QQ Plot of Transformed Model Residuals", cex.main = 0.9)
qqline(residuals_transformed, col = "red")

# Histogram with density scaling
hist(residuals_transformed,
     main = "Histogram of Residuals (Transformed Model)",
     xlab = "Residuals",
     col = "lightblue",
     breaks = 20,
     freq = FALSE,
     cex.main = 0.9)
```

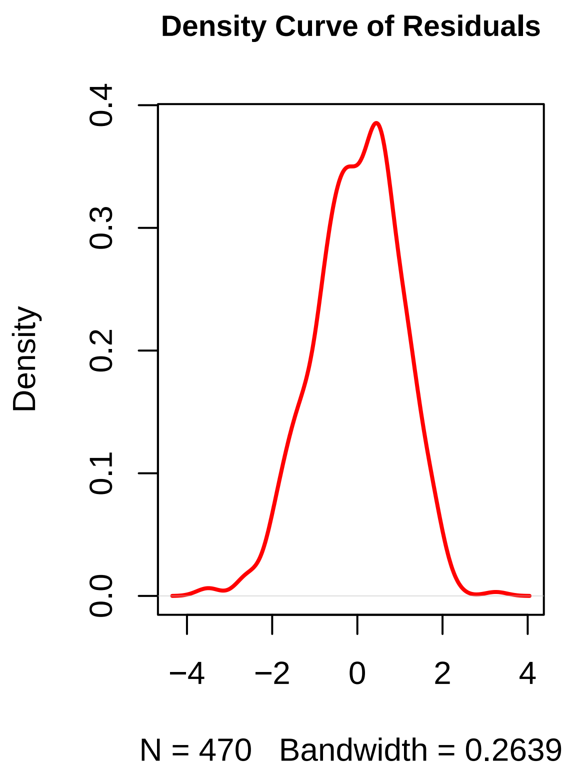




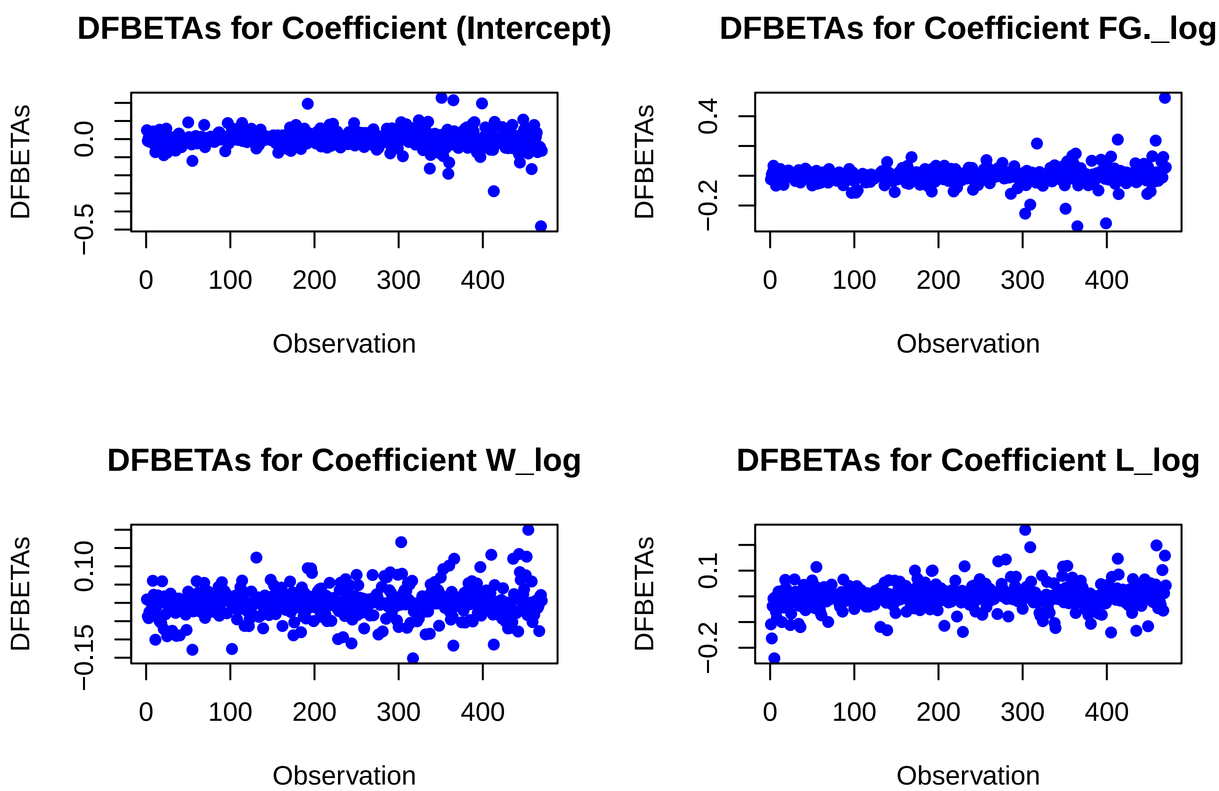

```r
# Density of Histogram Curve
plot(density(residuals_transformed), main = "Density Curve of Residuals", col = "red", lwd = 2, cex.mai

# Summary of transformed model
summary(model_transformed)
```

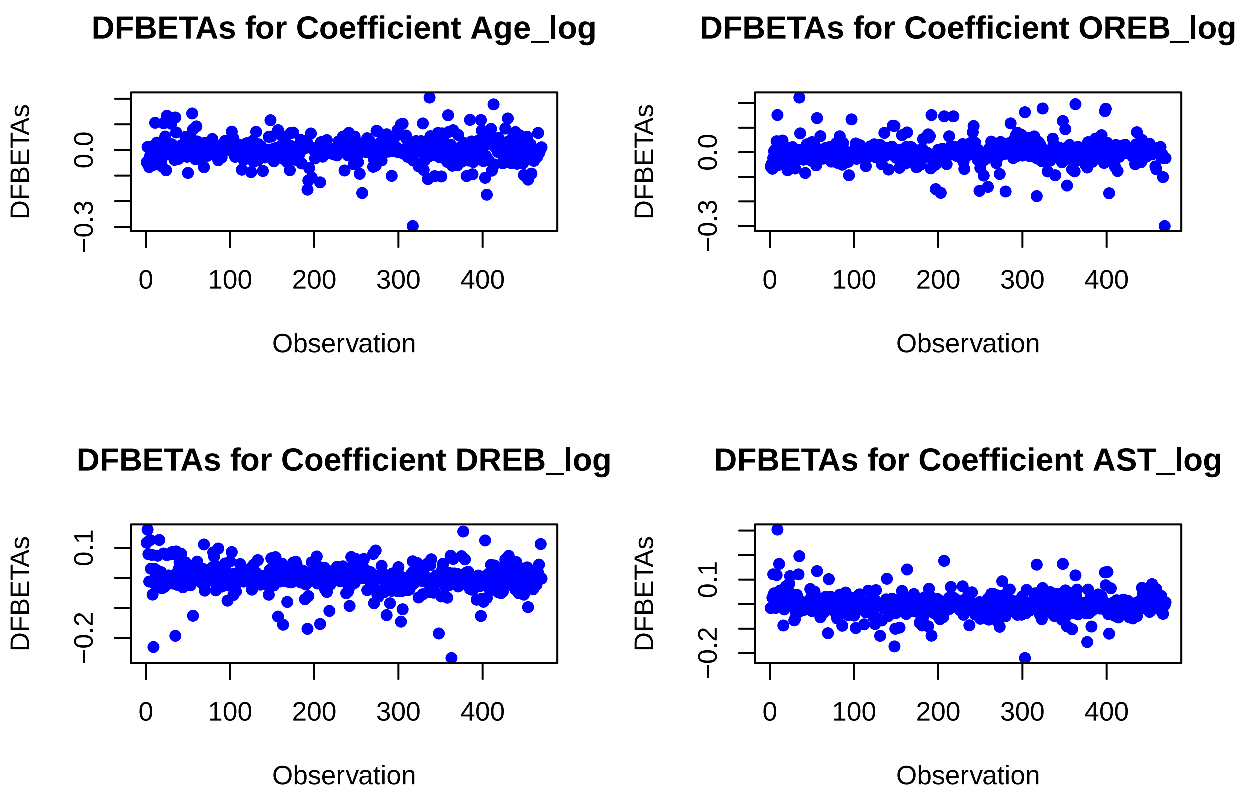
```
##
## Call:
## lm(formula = PPPM_transformed ~ FG._log + W_log + L_log + Age_log +
##     (FG._log)^2 + OREB_log + DREB_log + AST_log, data = mydata)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.152630 -0.029421  0.003113  0.029697  0.140146
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  0.6695377  0.0342100  19.571  < 2e-16 ***
## FG._log      0.0141850  0.0019510   7.271 1.54e-12 ***
## W_log       -0.0035532  0.0008855  -4.013 7.00e-05 ***
## L_log       -0.0035756  0.0009660  -3.701 0.000240 ***
## Age_log     -0.0069513  0.0020579  -3.378 0.000792 ***
## OREB_log    -0.0038357  0.0006499  -5.902 6.94e-09 ***
## DREB_log     0.0046880  0.0006773   6.921 1.50e-11 ***
## AST_log      0.0024269  0.0003765   6.446 2.89e-10 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0439 on 462 degrees of freedom
## Multiple R-squared:  0.4073, Adjusted R-squared:  0.3983
## F-statistic: 45.36 on 7 and 462 DF,  p-value: < 2.2e-16
```




```r
dfbetas_values <- dfbetas(model_transformed)

# Leverage vs. Residuals Plot
influence_plot <- influence.measures(model_transformed)
influence_plot_df <- influence_plot$infmat

# Plot
par(mfrow=c(2,2))

# DFBETAs plots
for(i in 1:ncol(dfbetas_values)) {
  plot(dfbetas_values[, i],
       main = paste("DFBETAs for Coefficient",
                    colnames(dfbetas_values)[i]),
       xlab = "Observation",
       ylab = "DFBETAs",
       pch = 16, col = "blue")
  abline(h = c(-2, 2), col = "red", lwd = 2)
}
```


DFBETAs for Coefficient (Intercept)
DFBETAs
0.0
−0.5
0
100
200
300
400
Observation
DFBETAs for Coefficient FG._log
DFBETAs
0.4
−0.2
0
100
200
300
400
Observation
DFBETAs for Coefficient W_log
DFBETAs
0.10
−0.15
0
100
200
300
400
Observation
DFBETAs for Coefficient L_log
DFBETAs
0.1
−0.2
0
100
200
300
400
Observation




```r
cat("\nDFBETAs (Significant Influence on Coefficients):\n")
```

```
##
## DFBETAs (Significant Influence on Coefficients):
```

```r
influential_dfbetas <- which(abs(dfbetas_values) > 2)
print(influential_dfbetas)
```

```
## integer(0)
```

```r
# Residuals from transformed model
r_student_residuals <- rstudent(model_transformed)

# Standardized residuals
standardized_residuals <- rstandard(model_transformed)

# Studentized residuals
studentized_residuals <- studres(model_transformed)

# Plot
par(mfrow=c(2,2))  # Plot in a 2x2 layout

# R-Student Residuals
plot(r_student_residuals,
```

```r
     main = "R-Student Residuals",
     xlab = "Observation",
     ylab = "R-Student Residuals",
     pch = 16, col = "blue")
abline(h = c(-3, 3), col = "red", lwd = 2)

# Standardized Residuals
plot(standardized_residuals,
     main = "Standardized Residuals",
     xlab = "Observation",
     ylab = "Standardized Residuals",
     pch = 16, col = "blue")
abline(h = c(-3, 3), col = "red", lwd = 2)

# Studentized Residuals
plot(studentized_residuals,
     main = "Studentized Residuals",
     xlab = "Observation",
     ylab = "Studentized Residuals",
     pch = 16, col = "blue")
abline(h = c(-3, 3), col = "red", lwd = 2)

# Identify observations with large residuals
outliers_r_student <- which(abs(r_student_residuals) > 3)
outliers_standardized <- which(abs(standardized_residuals) > 3)
outliers_studentized <- which(abs(studentized_residuals) > 3)

# Print the outlier points
cat("Outliers based on R-Student Residuals:\n")
```

```
## Outliers based on R-Student Residuals:
```

```r
print(outliers_r_student)
```

```
## 192 303 363
## 192 303 363
```

```r
cat("\nOutliers based on Standardized Residuals:\n")
```

```
##
## Outliers based on Standardized Residuals:
```

```r
print(outliers_standardized)
```

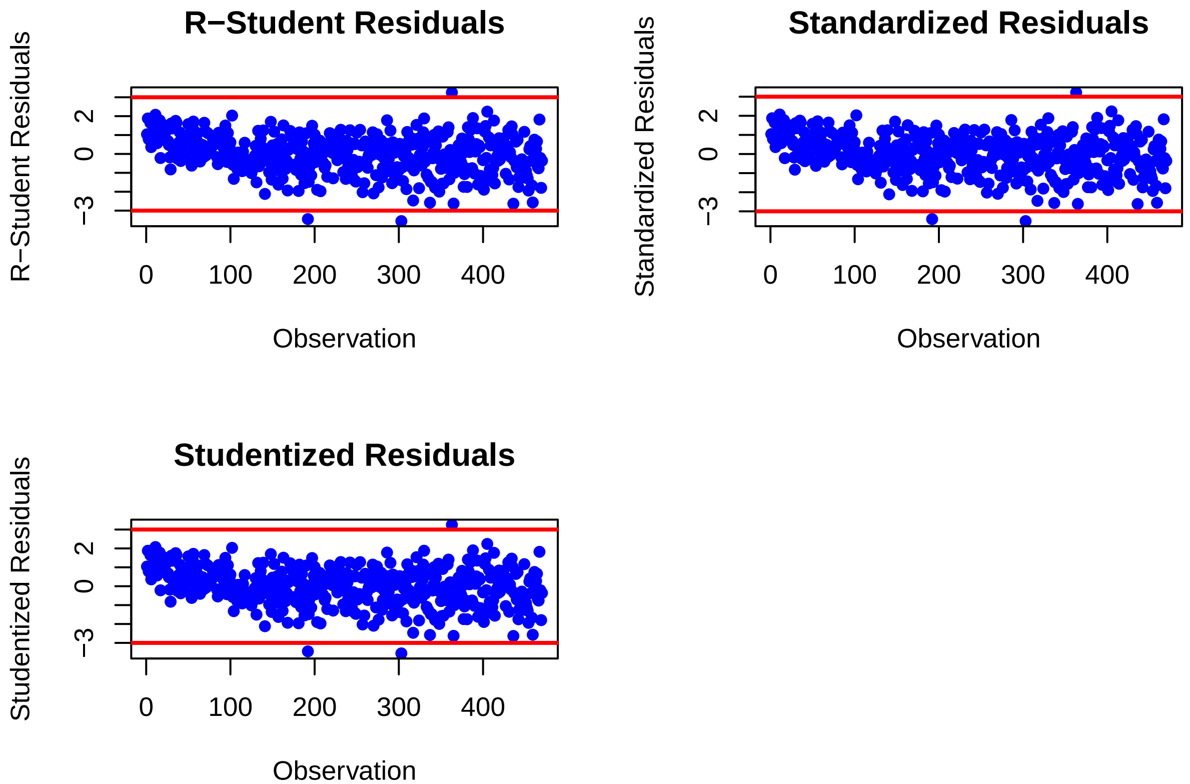
```
## 192 303 363
## 192 303 363
```

```r
cat("\nOutliers based on Studentized Residuals:\n")
```

```
##
## Outliers based on Studentized Residuals:
```

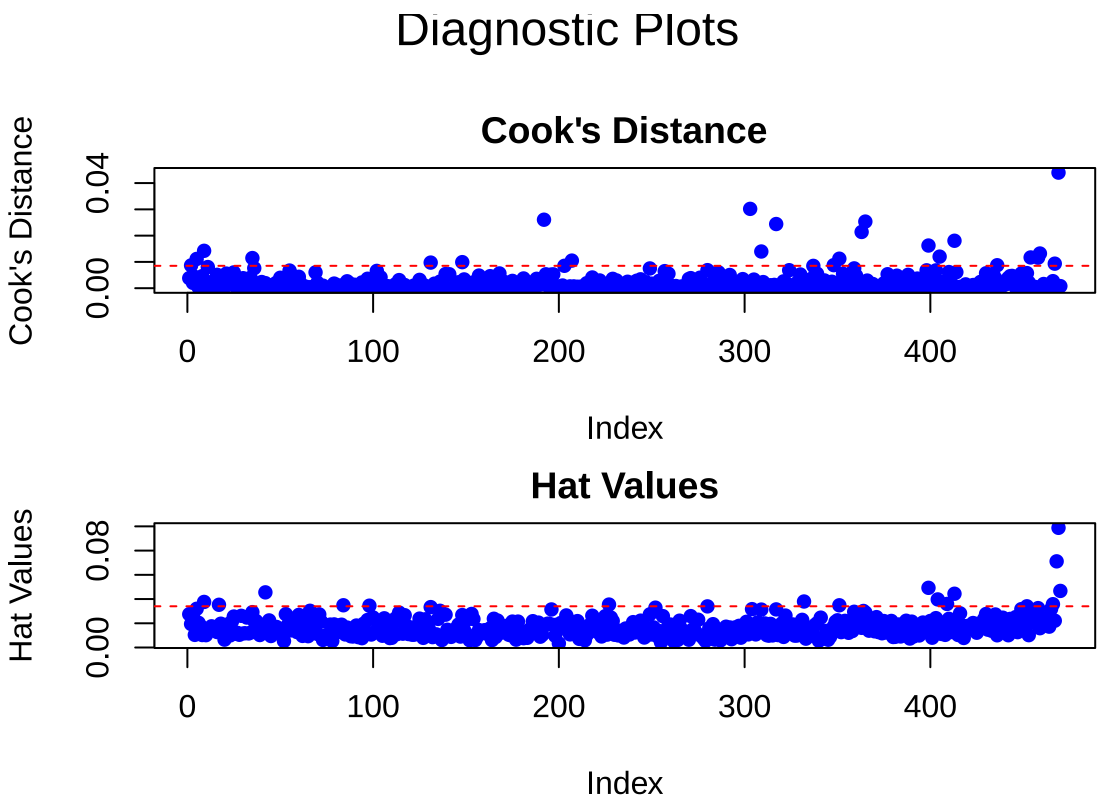
```r
print(outliers_studentized)
```

```
## 192 303 363
## 192 303 363
```

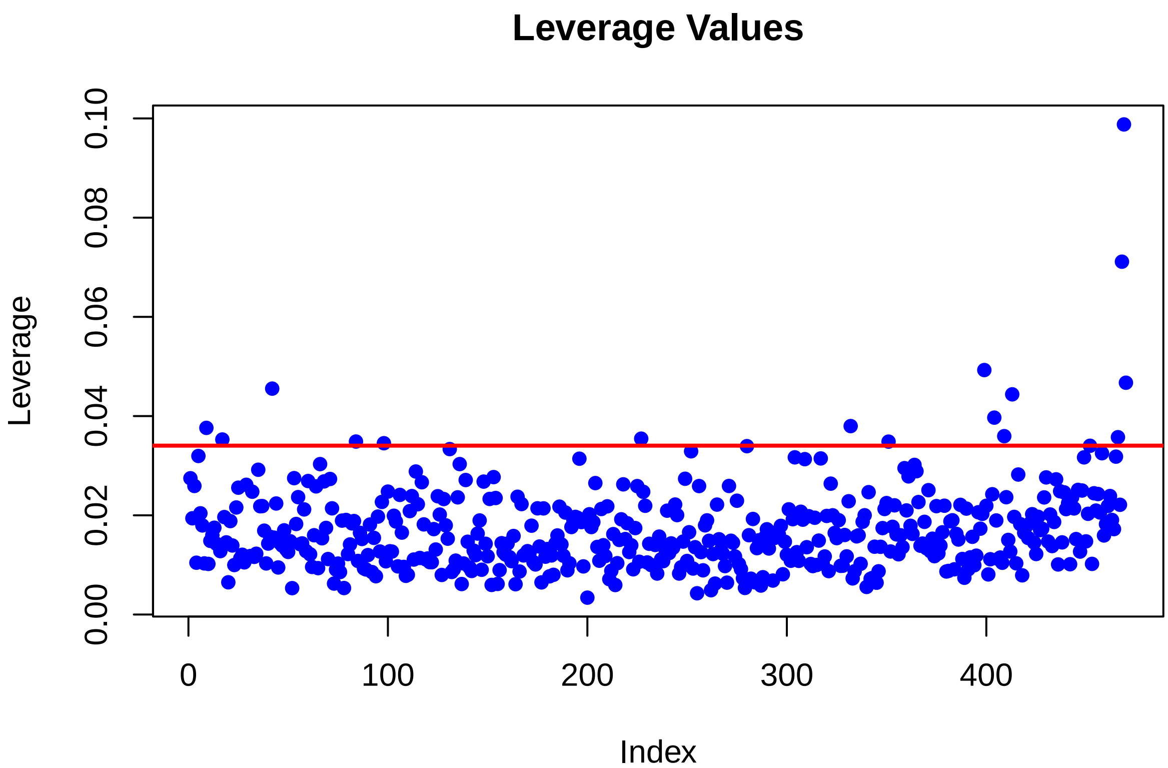



```r
cooks_distance <- cooks.distance(model_transformed)
hat_values <- lm.influence(model_transformed)$hat

par(mfrow = c(2, 1), mar = c(4, 4, 2, 1), oma = c(2, 2, 2, 1))

# Plot Cook's Distance
plot(cooks_distance,
     main = "Cook's Distance",
     xlab = "Index",
     ylab = "Cook's Distance",
     pch = 16,
     col = "blue")
abline(h = 4 / length(cooks_distance), col = "red", lty = 2)

# Plot Hat Values
plot(hat_values,
     main = "Hat Values",
     xlab = "Index",
     ylab = "Hat Values",
     pch = 16,
```

```r
      col = "blue")
abline(h = 2 * mean(hat_values), col = "red", lty = 2)

mtext("Diagnostic Plots", outer = TRUE, line = 1, cex = 1.5)
```




```r
# Compute leverage (hat values)
leverage <- hatvalues(model_transformed)

# Define the threshold for high leverage points
leverage_threshold <- 2 * mean(leverage)

# Plot
plot(leverage, main = "Leverage Values",
     xlab = "Index",
     ylab = "Leverage",
     pch = 16, col = "blue")

abline(h = leverage_threshold, col = "red", lwd = 2)
```


Leverage Values
Leverage
0.00
0.02
0.04
0.06
0.08
0.10
0
100
200
300
400
Index