Report

Zachary Katz

2023-04-08

Read in csv files:

```r
# Player, Year, Salary, Team
adjusted_nba_salaries <- read.csv("./data/adjusted_nba_salaries.csv")
# Team, Year, Total Salary
total_salary_by_team <- read.csv("./data/total_salary_by_team.csv")
# Team, Year, Wins
nba_team_wins <- read.csv("./data/nba_team_wins.csv")
# See Report advanced stats description
advanced_stats <- read.csv("./data/advanced_stats.csv")
# See Report advanced stats description
advanced_stats_2021 <- read.csv("./data/2021_advanced.csv")
# See Report shooting stats description
shooting_stats <- read.csv("./data/shooting_stats.csv")
# See Report shooting stats description
opponent_shooting_stats <- read.csv("./data/opponent_shooting_stats.csv")
# See Report total stats description
total_stats <- read.csv("./data/total_stats.csv")
# See Report total stats description
opponent_total_stats <- read.csv("./data/opponent_total_stats.csv")


# View(nba_salaries)
# View(nba_team_wins)
# View(advanced_stats)
# View(shooting_stats)
# View(opponent_shooting_stats)
# View(total_stats)
# View(opponent_total_stats)
```

Basic Summary

```r
print("==================================================")
```

```
## [1] "=================================================="
```

```r
print("NBA Salaries CSV")
```

```
## [1] "NBA Salaries CSV"
```

```r
# sapply(nba_salaries, summary)
describe(adjusted_nba_salaries)
```

```
##                vars    n     mean        sd  median   trimmed       mad
## rank*             1 9467   279.14    191.40     253    262.64    204.60
```

```
## name*              2 9467     927.55     532.60     930     927.74     686.44
## position*          3 9467       5.60       2.92       6       5.75       4.45
## team*              4 9467      33.73      12.66      34      33.85      17.79
## salary             5 9445 4488069.73 5368354.23 2351839 3387946.19 2397125.50
## year               6 9445    2011.44       5.62    2012    2011.70       5.93
## adjusted_salary    7 9445 5186067.03 5999312.34 2817823 3977314.86 2901495.11
##                     min      max    range  skew kurtosis       se
## rank*              1.00      700      699  0.59    -0.51     1.97
## name*              1.00     1857     1856 -0.01    -1.19     5.47
## position*          1.00       10        9 -0.51    -1.26     0.03
## team*              1.00       54       53 -0.03    -1.29     0.13
## salary          2692.00 40231758 40229066  2.21     5.94 55238.28
## year            2000.00     2020       20 -0.31    -0.91     0.06
## adjusted_salary 3194.36 40231758 40228564  1.96     4.20 61730.60
```

```r
print("===================================================")
```

```
## [1] "==================================================="
```

```r
print("NBA Total Salaries CSV")
```

```
## [1] "NBA Total Salaries CSV"
```

```r
# sapply(nba_salaries, summary)
describe(total_salary_by_team)
```

```
##              vars   n        mean          sd   median     trimmed          mad
## year*           1 659       12.09        5.93       12       12.11         7.41
## team*           2 659       23.00       12.67       23       22.99        17.79
## total_salary    3 657 74554647.10 33198770.74 75881848 75163178.07 27461435.94
##                   min       max     range  skew kurtosis         se
## year*             1.0        22        21 -0.03    -1.14       0.23
## team*             1.0        44        43  0.04    -1.30       0.49
## total_salary 702807.5 174907550 174204742 -0.09     0.12 1295207.40
```

```r
print("===================================================")
```

```
## [1] "==================================================="
```

```r
print("NBA Team Wins CSV")
```

```
## [1] "NBA Team Wins CSV"
```

```r
# sapply(nba_team_wins, summary)
describe(nba_team_wins)
```

```
##       vars   n    mean    sd median trimmed   mad  min  max range  skew
## year     1 600 2009.50  5.77 2009.5 2009.50  7.41 2000 2019    19  0.00
## team*    2 600   15.50  8.66   15.5   15.50 11.12    1   30    29  0.00
## wins     3 596   40.31 12.38   41.0   40.49 13.34    7   73    66 -0.13
##       kurtosis   se
## year     -1.21 0.24
## team*    -1.21 0.35
## wins     -0.68 0.51
```

```r
print("===================================================")
```

```
## [1] "==================================================="
```

```r
print("NBA Advanced Stats CSV")
```

```
## [1] "NBA Advanced Stats CSV"
```

```r
# sapply(advanced_stats, summary)
describe(advanced_stats)
```

```
##             vars   n      mean       sd    median  trimmed       mad       min
## Rk             1 596     15.40     8.61     15.00     15.40     10.38      1.00
## team*          2 616     16.44     9.43     16.00     16.38     11.86      1.00
## Age            3 616     26.72     1.69     26.60     26.65      1.63     22.70
## W              4 596     40.31    12.38     41.00     40.49     13.34      7.00
## L              5 596     40.31    12.28     40.00     40.02     13.34      9.00
## PW             6 616     40.37    11.84     41.00     40.50     13.34      7.00
## PL             7 616     40.25    11.76     40.00     40.02     13.34     15.00
## MOV            8 616     -0.01     4.43      0.00      0.07      4.56    -13.91
## SOS            9 616      0.00     0.39      0.00      0.00      0.45     -0.95
## SRS           10 616     -0.01     4.33      0.00      0.06      4.51    -13.96
## ORtg          11 616    106.56     3.81    106.30    106.58      3.85     92.20
## DRtg          12 616    106.56     3.66    106.60    106.63      3.85     94.10
## NRtg          13 596      0.00     4.79      0.20      0.08      5.11    -15.20
## Pace          14 616     93.33     3.73     92.65     93.07      3.63     86.20
## FTr           15 616      0.29     0.04      0.29      0.29      0.04      0.19
## X3PAr         16 616      0.25     0.07      0.23      0.24      0.07      0.10
## TS.           17 616      0.54     0.02      0.54      0.54      0.02      0.47
## eFG.          18 616      0.50     0.02      0.50      0.50      0.03      0.43
## TOV.          19 616     13.46     1.03     13.40     13.45      1.04     10.70
## ORB.          20 616     26.09     3.09     26.30     26.10      3.19     18.00
## FT.FGA        21 616      0.22     0.03      0.22      0.22      0.03      0.14
## eFG..1        22 616      0.50     0.02      0.50      0.50      0.02      0.43
## TOV..1        23 616     13.46     1.09     13.50     13.45      1.04     10.50
## DRB.          24 616     73.90     2.65     73.70     73.84      2.82     67.70
## FT.FGA.1      25 616      0.22     0.03      0.22      0.22      0.03      0.16
## Arena*        26 616     37.56    23.07     39.00     37.86     31.13      1.00
## Attend.       27 616 700729.75 93878.16 704974.00 703209.21 105988.11 460719.00
## Attend..G     28 616  17449.24  2071.13  17439.50  17479.75   2309.15  11286.00
## year          29 616   2009.55     5.75   2010.00   2009.56      7.41   2000.00
##                   max     range  skew kurtosis      se
## Rk              30.00     29.00  0.00    -1.21    0.35
## team*           33.00     32.00  0.06    -1.22    0.38
## Age             32.00      9.30  0.36    -0.25    0.07
## W               73.00     66.00 -0.13    -0.68    0.51
## L               72.00     63.00  0.19    -0.64    0.50
## PW              67.00     60.00 -0.10    -0.65    0.48
## PL              66.00     51.00  0.15    -0.66    0.47
## MOV             11.63     25.54 -0.14    -0.38    0.18
## SOS              0.96      1.91  0.00    -0.71    0.02
## SRS             11.35     25.31 -0.14    -0.38    0.17
## ORtg           116.70     24.50 -0.09    -0.02    0.15
## DRtg           117.60     23.50 -0.16    -0.13    0.15
## NRtg            11.60     26.80 -0.14    -0.49    0.20
## Pace           105.10     18.90  0.63    -0.10    0.15
## FTr              0.42      0.22  0.29    -0.14    0.00
## X3PAr            0.52      0.42  0.64     0.12    0.00
```

```
## TS.            0.60      0.13  0.15    -0.02    0.00
## eFG.           0.57      0.14  0.22    -0.17    0.00
## TOV.          17.10      6.40  0.12     0.13    0.04
## ORB.          35.30     17.30 -0.04    -0.37    0.12
## FT.FGA         0.30      0.16  0.28    -0.15    0.00
## eFG..1         0.56      0.13  0.07    -0.27    0.00
## TOV..1        16.50      6.00  0.07    -0.06    0.04
## DRB.          81.60     13.90  0.22    -0.37    0.11
## FT.FGA.1       0.34      0.18  0.52     0.52    0.00
## Arena*        73.00     72.00 -0.12    -1.38    0.93
## Attend.   913176.00 452457.00 -0.18    -0.45 3782.46
## Attend..G  22273.00  10987.00 -0.13    -0.40   83.45
## year        2019.00     19.00 -0.01    -1.20    0.23
```

```r
print("====================================================")
```

```
## [1] "===================================================="
```

```r
print("NBA Shooting Stats CSV")
```

```
## [1] "NBA Shooting Stats CSV"
```

```r
# sapply(shooting_stats, summary)
describe(shooting_stats)
```

```
##           vars   n     mean      sd   median  trimmed    mad      min      max
## Rk           1 596    15.40    8.61    15.00    15.40  10.38     1.00    30.00
## team*        2 616    18.63   10.48    18.00    18.59  13.34     1.00    37.00
## G            3 616    80.62    4.25    82.00    81.98   0.00    64.00    82.00
## MP           4 616 19494.23 1028.35 19805.00 19807.58  74.13 15560.00 20080.00
## FG.          5 616     0.45    0.02     0.45     0.45   0.01     0.41     0.50
## Dist.        6 616    12.55    0.91    12.60    12.55   0.89    10.00    15.40
## X2P          7 616     0.75    0.07     0.77     0.76   0.07     0.50     0.90
## X0.3         8 616     0.29    0.03     0.29     0.29   0.03     0.20     0.42
## X3.10        9 616     0.15    0.03     0.15     0.15   0.03     0.08     0.24
## X10.16      10 616     0.12    0.03     0.11     0.11   0.03     0.04     0.26
## X16.3P      11 616     0.19    0.06     0.20     0.20   0.05     0.04     0.34
## X3P         12 616     0.25    0.07     0.23     0.24   0.07     0.10     0.50
## X2P.1       13 616     0.49    0.02     0.48     0.48   0.02     0.43     0.57
## X0.3.1      14 616     0.62    0.03     0.62     0.62   0.04     0.54     0.72
## X3.10.1     15 616     0.39    0.03     0.39     0.39   0.03     0.29     0.48
## X10.16.1    16 616     0.39    0.03     0.40     0.39   0.03     0.32     0.48
## X16.3P.1    17 616     0.40    0.02     0.40     0.40   0.02     0.32     0.47
## X3P.1       18 616     0.35    0.02     0.36     0.35   0.02     0.28     0.42
## X2P.2       19 616     0.52    0.05     0.52     0.52   0.05     0.37     0.71
## X3P.2       20 616     0.85    0.04     0.85     0.85   0.04     0.67     0.96
## X.FGA       21 616     0.05    0.01     0.05     0.05   0.01     0.01     0.10
## Md.         22 616   298.72   83.78   295.00   295.44  86.73    75.00   642.00
## X.FGA.1     23 586     0.24    0.03     0.24     0.24   0.04     0.15     0.33
## Md..1       24 616   852.41  241.21   879.00   881.05 166.79     0.00  1294.00
## X.3PA       25 616     0.25    0.05     0.24     0.25   0.05     0.11     0.45
## X3P.        26 616     0.38    0.03     0.39     0.38   0.03     0.25     0.48
## Att.        27 585    13.70    5.71    13.00    13.41   5.93     2.00    36.00
## Md..2       28 555     0.34    0.59     0.00     0.22   0.00     0.00     3.00
## year        29 616  2009.55    5.75  2010.00  2009.56   7.41  2000.00  2019.00
##              range  skew kurtosis    se
```

```
## Rk         29.00  0.00    -1.21  0.35
## team*      36.00  0.08    -1.19  0.42
## G          18.00 -2.90     6.80  0.17
## MP       4520.00 -2.87     6.69 41.43
## FG.         0.10  0.18     0.44  0.00
## Dist.       5.40 -0.01    -0.16  0.04
## X2P         0.40 -0.56    -0.15  0.00
## X0.3        0.21  0.33     0.17  0.00
## X3.10       0.16  0.27     0.01  0.00
## X10.16      0.22  0.97     1.70  0.00
## X16.3P      0.30 -0.38    -0.23  0.00
## X3P         0.40  0.56    -0.15  0.00
## X2P.1       0.14  0.48     0.05  0.00
## X0.3.1      0.19  0.09    -0.44  0.00
## X3.10.1     0.19 -0.17    -0.06  0.00
## X10.16.1    0.17  0.00     0.04  0.00
## X16.3P.1    0.15  0.08     0.46  0.00
## X3P.1       0.14 -0.11     0.63  0.00
## X2P.2       0.34  0.38     0.59  0.00
## X3P.2       0.29 -0.48     0.69  0.00
## X.FGA       0.09  0.44     0.20  0.00
## Md.       567.00  0.38     0.17  3.38
## X.FGA.1     0.18  0.03    -0.50  0.00
## Md..1    1294.00 -1.92     5.01  9.72
## X.3PA       0.34  0.56     0.53  0.00
## X3P.        0.23 -0.21     0.77  0.00
## Att.       34.00  0.58     0.33  0.24
## Md..2       3.00  1.88     3.64  0.02
## year       19.00 -0.01    -1.20  0.23
```

```r
print("==================================================")
```

```
## [1] "=================================================="
```

```r
print("NBA Opponent Shooting Stats CSV")
```

```
## [1] "NBA Opponent Shooting Stats CSV"
```

```r
# sapply(opponent_shooting_stats, summary)
describe(opponent_shooting_stats)
```

```
##          vars   n     mean      sd   median  trimmed   mad      min      max
## Rk          1 596    15.40    8.61    15.00    15.40 10.38     1.00    30.00
## team*       2 616    18.63   10.48    18.00    18.59 13.34     1.00    37.00
## G           3 616    80.62    4.25    82.00    81.98  0.00    64.00    82.00
## MP          4 616 19494.24 1028.30 19805.00 19807.57 74.13 15560.00 20080.00
## FG.         5 616     0.45    0.02     0.45     0.45  0.02     0.41     0.50
## Dist.       6 616    12.55    0.72    12.50    12.52  0.74    10.40    15.40
## X2P         7 616     0.75    0.06     0.77     0.76  0.05     0.54     0.87
## X0.3        8 616     0.29    0.03     0.29     0.29  0.03     0.20     0.42
## X3.10       9 616     0.15    0.03     0.15     0.15  0.03     0.09     0.22
## X10.16     10 616     0.12    0.02     0.11     0.11  0.02     0.04     0.20
## X16.3P     11 616     0.19    0.05     0.21     0.20  0.04     0.05     0.29
## X3P        12 616     0.25    0.06     0.23     0.24  0.05     0.13     0.46
## X2P.1      13 616     0.49    0.02     0.48     0.49  0.02     0.42     0.56
## X0.3.1     14 616     0.62    0.03     0.62     0.62  0.04     0.53     0.71
```

```
## X3.10.1     15 616     0.39    0.03     0.40     0.39   0.03     0.31     0.49
## X10.16.1    16 616     0.40    0.02     0.40     0.40   0.02     0.32     0.47
## X16.3P.1    17 616     0.40    0.02     0.40     0.40   0.02     0.34     0.47
## X3P.1       18 616     0.36    0.02     0.36     0.36   0.01     0.30     0.41
## X2P.2       19 616     0.53    0.04     0.52     0.53   0.04     0.37     0.66
## X3P.2       20 616     0.85    0.03     0.85     0.85   0.03     0.74     0.93
## X.FGA       21 616     0.05    0.01     0.05     0.05   0.01     0.02     0.10
## Md.         22 616   298.72   58.65   299.00   296.61  57.82   104.00   642.00
## X.FGA.1     23 586     0.24    0.03     0.24     0.24   0.03     0.17     0.32
## Md..1       24 586   896.03  127.46   893.00   892.85 126.02   604.00  1276.00
## X.3PA       25 616     0.25    0.03     0.25     0.25   0.03     0.13     0.38
## X3P.        26 616     0.39    0.03     0.39     0.39   0.03     0.30     0.47
## year        27 616  2009.55    5.75  2010.00  2009.56   7.41  2000.00  2019.00
## Att.        28  62    15.38    5.95    14.00    14.82   5.19     6.00    31.00
## Md..2       29  62     0.30    0.58     0.00     0.17   0.00     0.00     2.00
##             range  skew kurtosis    se
## Rk          29.00  0.00    -1.21  0.35
## team*       36.00  0.08    -1.19  0.42
## G           18.00 -2.90     6.80  0.17
## MP        4520.00 -2.87     6.69 41.43
## FG.          0.09 -0.01    -0.18  0.00
## Dist.        5.00  0.51     0.67  0.03
## X2P          0.33 -0.86    -0.06  0.00
## X0.3         0.21  0.13     0.46  0.00
## X3.10        0.13  0.11    -0.54  0.00
## X10.16       0.16  0.76     0.67  0.00
## X16.3P       0.23 -0.87    -0.20  0.00
## X3P          0.33  0.86    -0.06  0.00
## X2P.1        0.14  0.34    -0.01  0.00
## X0.3.1       0.18  0.09    -0.47  0.00
## X3.10.1      0.18 -0.06     0.24  0.00
## X10.16.1     0.15  0.13     0.10  0.00
## X16.3P.1     0.13  0.10     0.41  0.00
## X3P.1        0.11 -0.03     0.64  0.00
## X2P.2        0.28  0.02     0.32  0.00
## X3P.2        0.19 -0.24    -0.21  0.00
## X.FGA        0.08  0.56     2.20  0.00
## Md.        538.00  0.62     2.19  2.36
## X.FGA.1      0.15  0.22    -0.28  0.00
## Md..1      672.00  0.23    -0.18  5.27
## X.3PA        0.25 -0.09     0.22  0.00
## X3P.         0.17 -0.10     0.27  0.00
## year        19.00 -0.01    -1.20  0.23
## Att.        25.00  0.84     0.31  0.76
## Md..2        2.00  1.79     2.12  0.07
```

```r
print("==================================================")
```

```
## [1] "=================================================="
```

```r
print("NBA Total Stats CSV")
```

```
## [1] "NBA Total Stats CSV"
```

```r
# sapply(total_stats, summary)
describe(total_stats)
```

```
##       vars   n     mean      sd   median  trimmed    mad      min      max
## Rk       1 596    15.40    8.61    15.00    15.40  10.38     1.00    30.00
## team*    2 616    18.63   10.48    18.00    18.59  13.34     1.00    37.00
## G        3 616    80.62    4.25    82.00    81.98   0.00    64.00    82.00
## MP       4 616 19494.23 1028.30 19805.00 19807.56  74.13 15560.00 20080.00
## FG       5 616  3014.41  225.03  3022.00  3027.12 176.43  2193.00  3612.00
## FGA      6 616  6653.03  422.27  6684.00  6691.91 321.72  5086.00  7706.00
## FG.      7 616     0.45    0.02     0.45     0.45   0.01     0.41     0.50
## X3P      8 616   584.52  198.90   545.00   571.89 197.93   214.00  1323.00
## X3PA     9 616  1639.90  534.60  1518.00  1602.83 521.13   641.00  3721.00
## X3P.    10 616     0.35    0.02     0.35     0.35   0.02     0.28     0.42
## X2P     11 616  2429.89  209.45  2467.00  2449.97 166.79  1605.00  2891.00
## X2PA    12 616  5013.13  510.16  5103.50  5056.98 448.49  3251.00  6211.00
## X2P.    13 616     0.49    0.02     0.48     0.48   0.02     0.43     0.57
## FT      14 616  1466.51  181.29  1467.50  1469.45 167.53   889.00  1977.00
## FTA     15 616  1935.44  241.52  1939.50  1938.33 229.06  1198.00  2587.00
## FT.     16 616     0.76    0.03     0.76     0.76   0.03     0.66     0.83
## ORB     17 616   893.91  117.98   897.50   894.71 116.38   509.00  1345.00
## DRB     18 616  2534.57  204.62  2527.00  2540.78 177.17  1879.00  3316.00
## TRB     19 616  3428.48  230.77  3450.00  3449.36 174.95  2560.00  4078.00
## AST     20 616  1770.89  183.08  1768.00  1769.71 156.41  1224.00  2491.00
## STL     21 616   612.25   74.41   609.00   611.31  72.65   390.00   844.00
## BLK     22 616   395.10   67.08   391.00   393.52  68.20   195.00   624.00
## TOV     23 616  1166.32  108.14  1172.00  1168.13  97.11   738.00  1514.00
## PF      24 616  1694.11  171.14  1703.50  1703.06 159.38  1109.00  2189.00
## PTS     25 616  8079.86  628.31  8085.50  8112.31 484.07  5739.00  9686.00
## year    26 616  2009.57    5.77  2010.00  2009.58   7.41  2000.00  2020.00
##         range  skew kurtosis    se
## Rk      29.00  0.00    -1.21  0.35
## team*   36.00  0.08    -1.19  0.42
## G       18.00 -2.90     6.80  0.17
## MP    4520.00 -2.87     6.69 41.43
## FG    1419.00 -0.70     1.54  9.07
## FGA   2620.00 -1.20     2.59 17.01
## FG.      0.10  0.17     0.40  0.00
## X3P   1109.00  0.60    -0.08  8.01
## X3PA  3080.00  0.68     0.08 21.54
## X3P.     0.14 -0.09     0.58  0.00
## X2P   1286.00 -0.95     1.14  8.44
## X2PA  2960.00 -0.84     0.71 20.55
## X2P.     0.14  0.51     0.15  0.00
## FT    1088.00 -0.18     0.27  7.30
## FTA   1389.00 -0.15     0.08  9.73
## FT.      0.17 -0.37     0.26  0.00
## ORB    836.00  0.03     0.16  4.75
## DRB   1437.00 -0.32     0.95  8.24
## TRB   1518.00 -1.08     2.19  9.30
## AST   1267.00  0.12     1.00  7.38
## STL    454.00  0.11     0.04  3.00
## BLK    429.00  0.21     0.08  2.70
## TOV    776.00 -0.15     0.36  4.36
## PF    1080.00 -0.53     0.80  6.90
## PTS   3947.00 -0.70     1.61 25.32
## year    20.00 -0.01    -1.20  0.23
```

```r
print("==================================================")
```

```
## [1] "=================================================="
```

```r
print("NBA Opponent Total Stats CSV")
```

```
## [1] "NBA Opponent Total Stats CSV"
```

```r
# sapply(opponent_total_stats, summary)
describe(opponent_total_stats)
```

```
##       vars   n     mean      sd   median  trimmed    mad      min      max
## Rk       1 596    15.40    8.61    15.00    15.40  10.38     1.00    30.00
## team*    2 616    18.63   10.48    18.00    18.59  13.34     1.00    37.00
## G        3 616    80.62    4.25    82.00    81.98   0.00    64.00    82.00
## MP       4 616 19494.23 1028.30 19805.00 19807.56  74.13 15560.00 20080.00
## FG       5 616  3014.41  231.22  3031.00  3027.36 195.70  2207.00  3560.00
## FGA      6 616  6653.03  434.00  6687.00  6688.86 320.98  5098.00  7669.00
## FG.      7 616     0.45    0.02     0.45     0.45   0.02     0.41     0.50
## X3P      8 616   584.52  171.53   538.50   568.29 154.93   273.00  1073.00
## X3PA     9 616  1639.90  468.94  1510.50  1592.74 422.54   839.00  2976.00
## X3P.    10 616     0.36    0.01     0.36     0.36   0.01     0.30     0.41
## X2P     11 616  2429.89  197.39  2455.50  2447.63 157.90  1754.00  2899.00
## X2PA    12 616  5013.12  463.71  5085.00  5059.85 375.10  3307.00  6055.00
## X2P.    13 616     0.49    0.02     0.48     0.49   0.02     0.42     0.56
## FT      14 616  1488.88  585.62  1470.00  1468.30 174.95   901.00 15309.00
## FTA     15 616  1932.16  249.21  1942.00  1937.66 236.47     0.76  2653.00
## FT.     16 616     2.29   37.96     0.76     0.76   0.01     0.72   943.00
## ORB     17 616   896.55  121.98   896.50   895.21  98.59   628.00  2569.00
## DRB     18 616  2536.10  209.46  2527.00  2540.40 177.91  1889.00  3512.00
## TRB     19 616  3425.72  244.99  3452.00  3446.53 187.55  1808.00  4018.00
## AST     20 616  1768.98  178.84  1778.50  1776.83 171.98   635.00  2202.00
## STL     21 616   611.68   65.41   613.00   612.17  59.30   288.00   810.00
## BLK     22 616   396.77   69.06   397.50   394.74  58.56   245.00  1312.00
## TOV     23 616  1166.84  112.76  1169.00  1168.80 106.75   745.00  1633.00
## PF      24 616  1705.09  317.16  1699.00  1704.26 140.85  1070.00  8400.00
## PTS     25 616  8069.49  668.24  8101.00  8114.43 489.26  2013.00  9788.00
## year    26 615  2009.55    5.76  2010.00  2009.55   7.41  2000.00  2019.00
##          range  skew kurtosis    se
## Rk       29.00  0.00    -1.21  0.35
## team*    36.00  0.08    -1.19  0.42
## G        18.00 -2.90     6.80  0.17
## MP     4520.00 -2.87     6.69 41.43
## FG     1353.00 -0.69     1.12  9.32
## FGA    2571.00 -1.10     2.31 17.49
## FG.       0.09 -0.09    -0.22  0.00
## X3P     800.00  0.76    -0.36  6.91
## X3PA   2137.00  0.80    -0.35 18.89
## X3P.      0.11 -0.05     0.61  0.00
## X2P    1145.00 -0.89     1.10  7.95
## X2PA   2748.00 -0.93     1.02 18.68
## X2P.      0.14  0.28    -0.04  0.00
## FT    14408.00 21.32   500.52 23.60
## FTA    2652.24 -0.85     5.44 10.04
## FT.     942.28 24.70   609.01  1.53
```

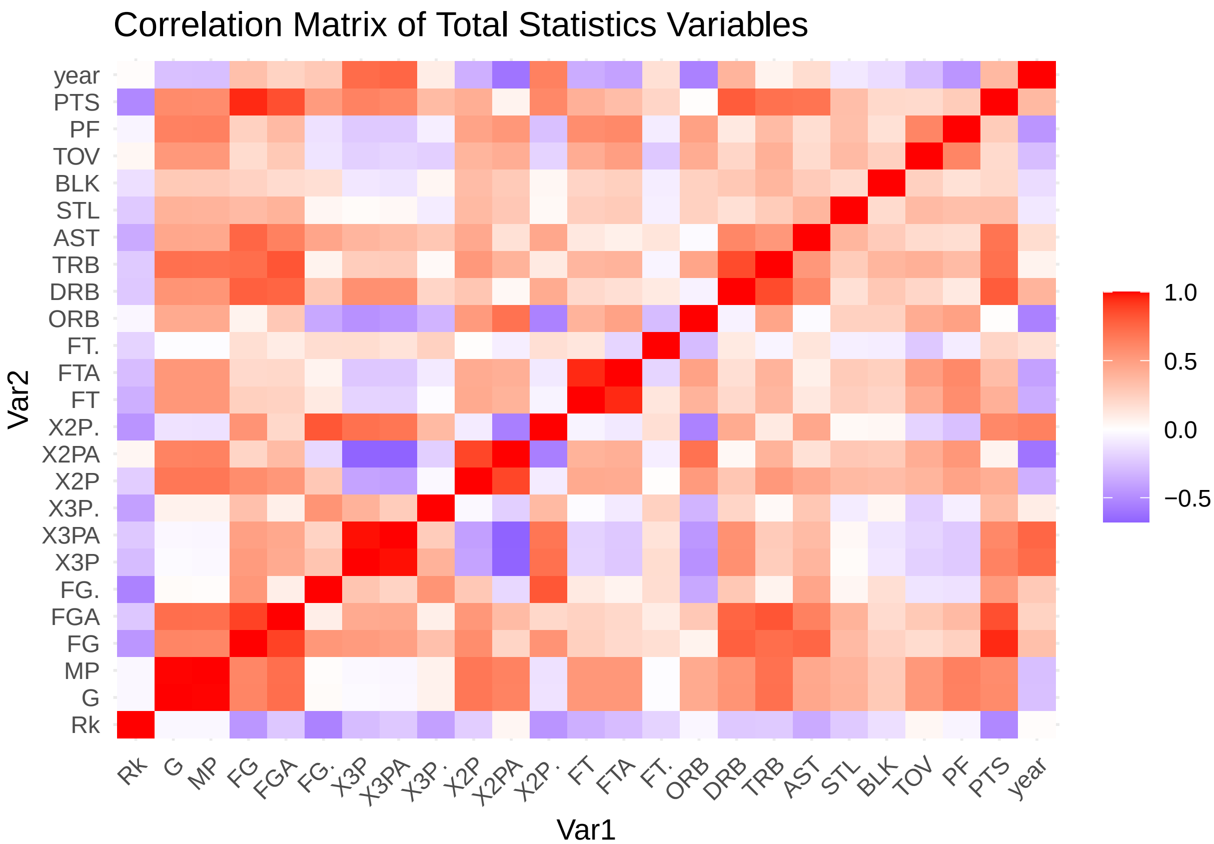
```
## ORB    1941.00  4.10    55.83  4.91
## DRB    1623.00 -0.16     1.19  8.44
## TRB    2210.00 -1.25     3.96  9.87
## AST    1567.00 -0.75     2.46  7.21
## STL     522.00 -0.17     0.95  2.64
## BLK    1067.00  3.80    48.57  2.78
## TOV     888.00 -0.10     0.61  4.54
## PF     7330.00 15.15   319.61 12.78
## PTS    7775.00 -1.78    11.37 26.92
## year     19.00 -0.01    -1.21  0.23
```

```r
print("==================================================")
```

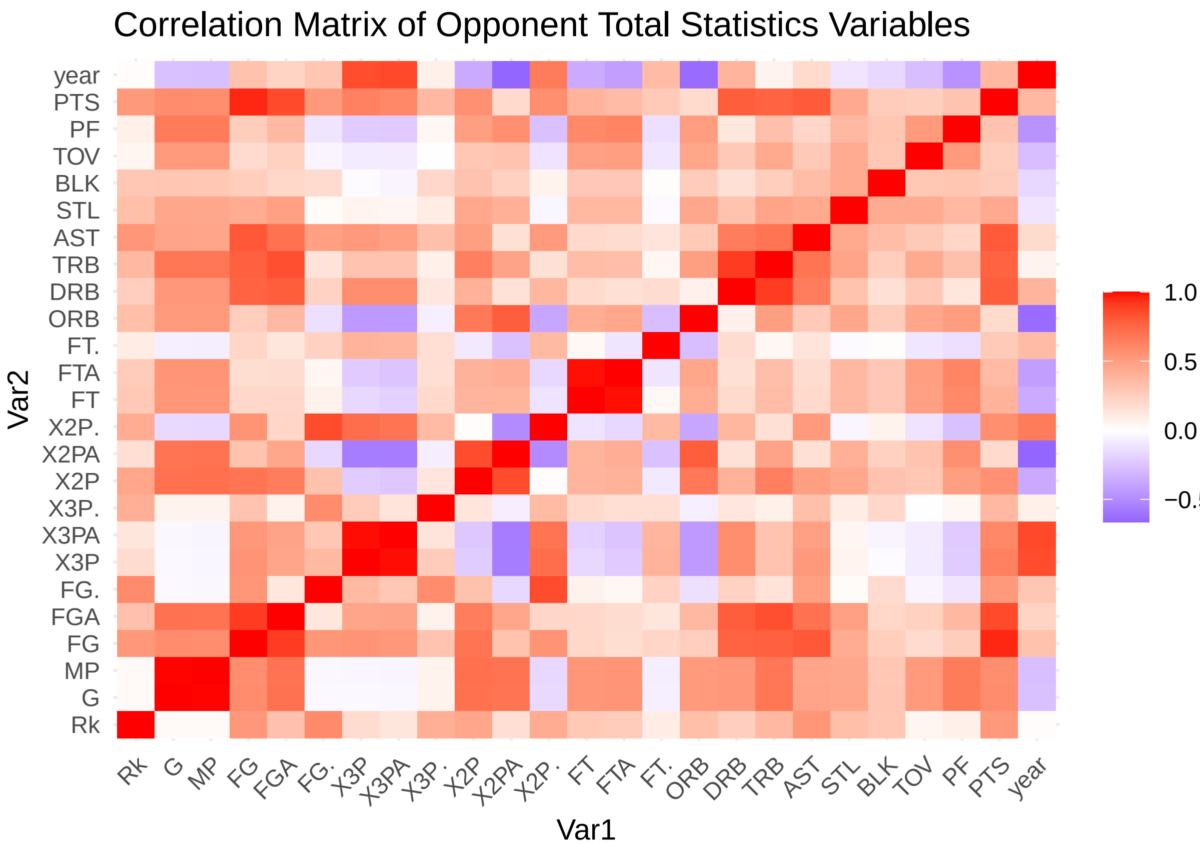
```
## [1] "=================================================="
```

Basic Visualization (Correlation Matrices)

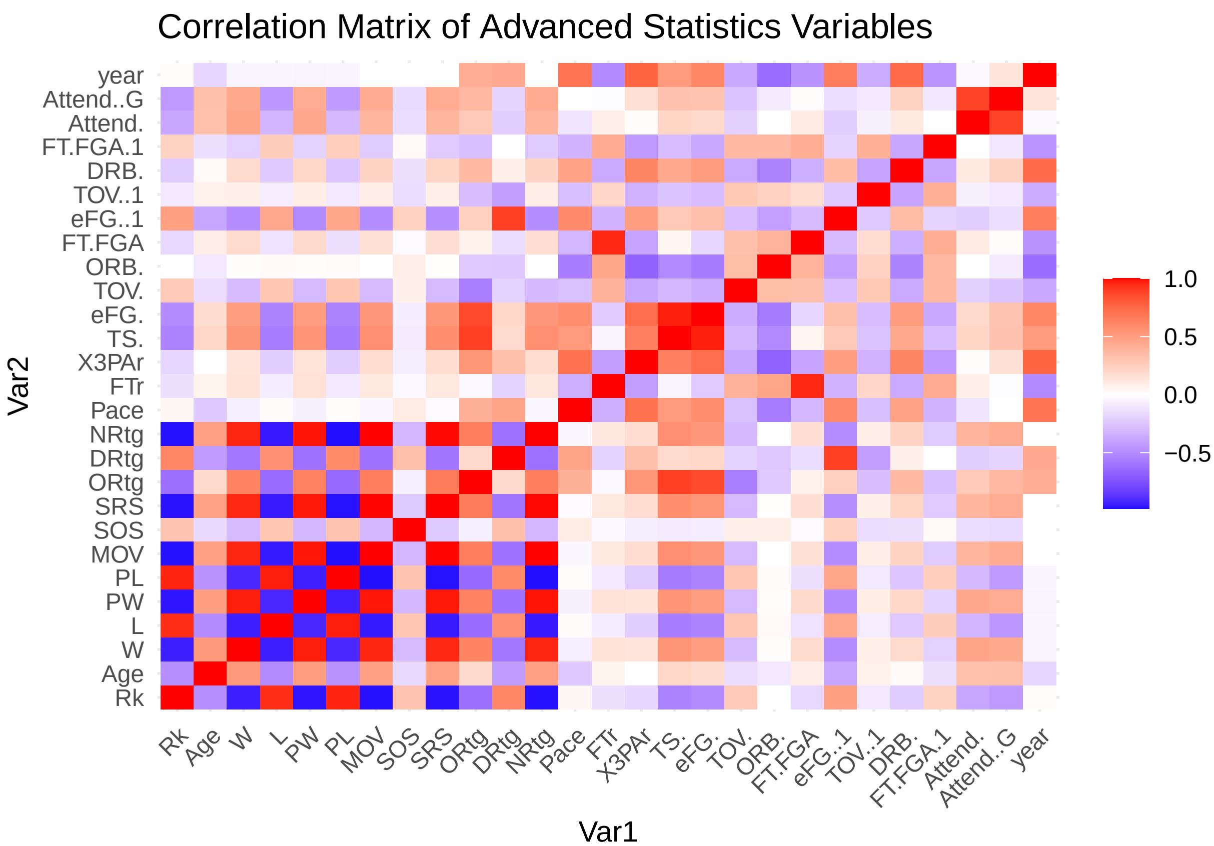
```r
# =========================================================
#               Correlation Matrices
# =========================================================
create_correlation_matrix <- function(data, name) {
    # Create correlation matrix
    numeric_data <- data[, sapply(data, is.numeric)]

    # Remove rows with missing values
    numeric_data <- numeric_data[complete.cases(numeric_data),]

    cor_mat <- cor(numeric_data)

    # Melt correlation matrix into long format for ggplot2
    cor_melt <- melt(cor_mat)

    # Create ggplot2 plot of correlation matrix
    ggplot(cor_melt, aes(x = Var1, y = Var2, fill = value)) +
      geom_tile() +
      scale_fill_gradient2(low = "blue", high = "red", mid = "white", midpoint = 0) +
      theme_minimal() +
      theme(axis.text.x = element_text(angle = 45, hjust = 1),
            legend.title = element_blank()) +
      labs(title = name)
}

create_correlation_matrix(total_stats, "Correlation Matrix of Total Statistics Variables")
```



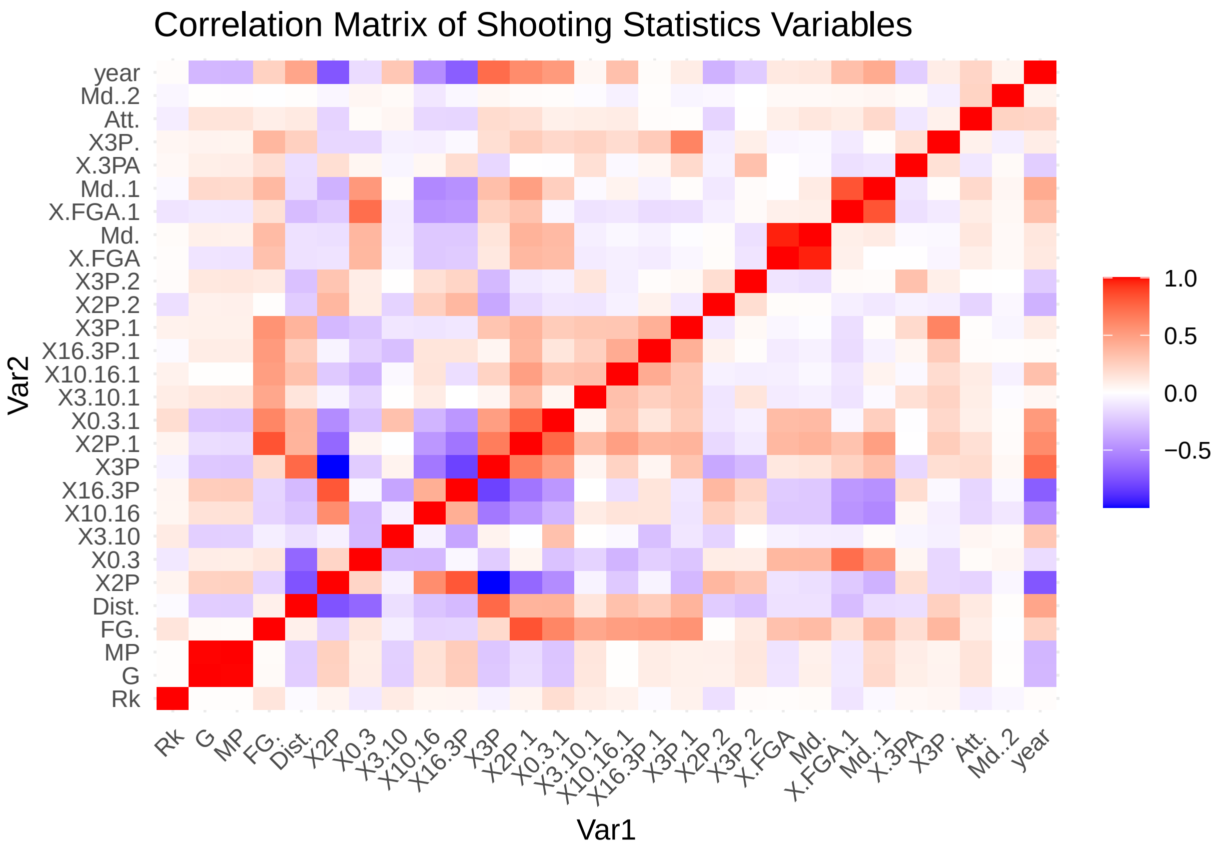

```
create_correlation_matrix(opponent_total_stats, "Correlation Matrix of Opponent Total Statistics Variabl
```



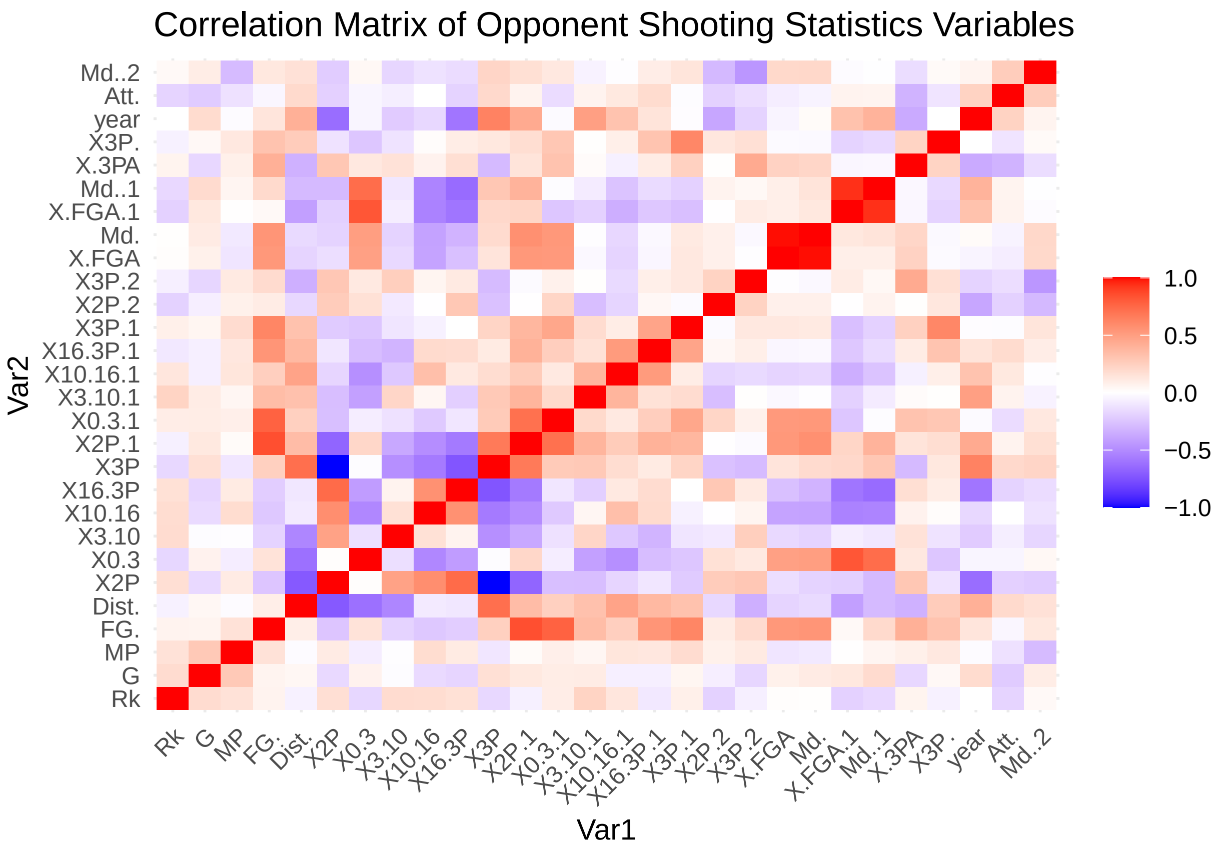

```
create_correlation_matrix(advanced_stats, "Correlation Matrix of Advanced Statistics Variables")
```




```r
create_correlation_matrix(shooting_stats, "Correlation Matrix of Shooting Statistics Variables")
```

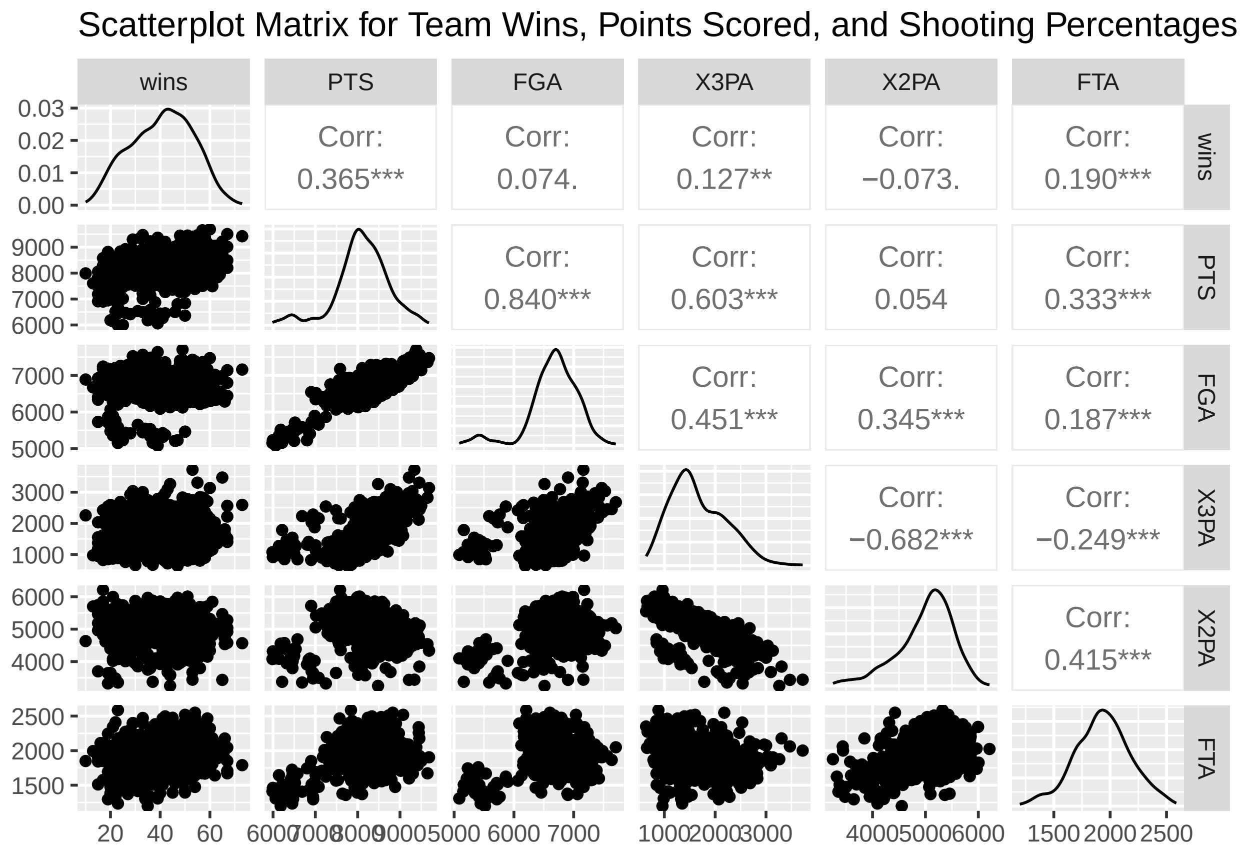



```
create_correlation_matrix(opponent_shooting_stats, "Correlation Matrix of Opponent Shooting Statistics V
```

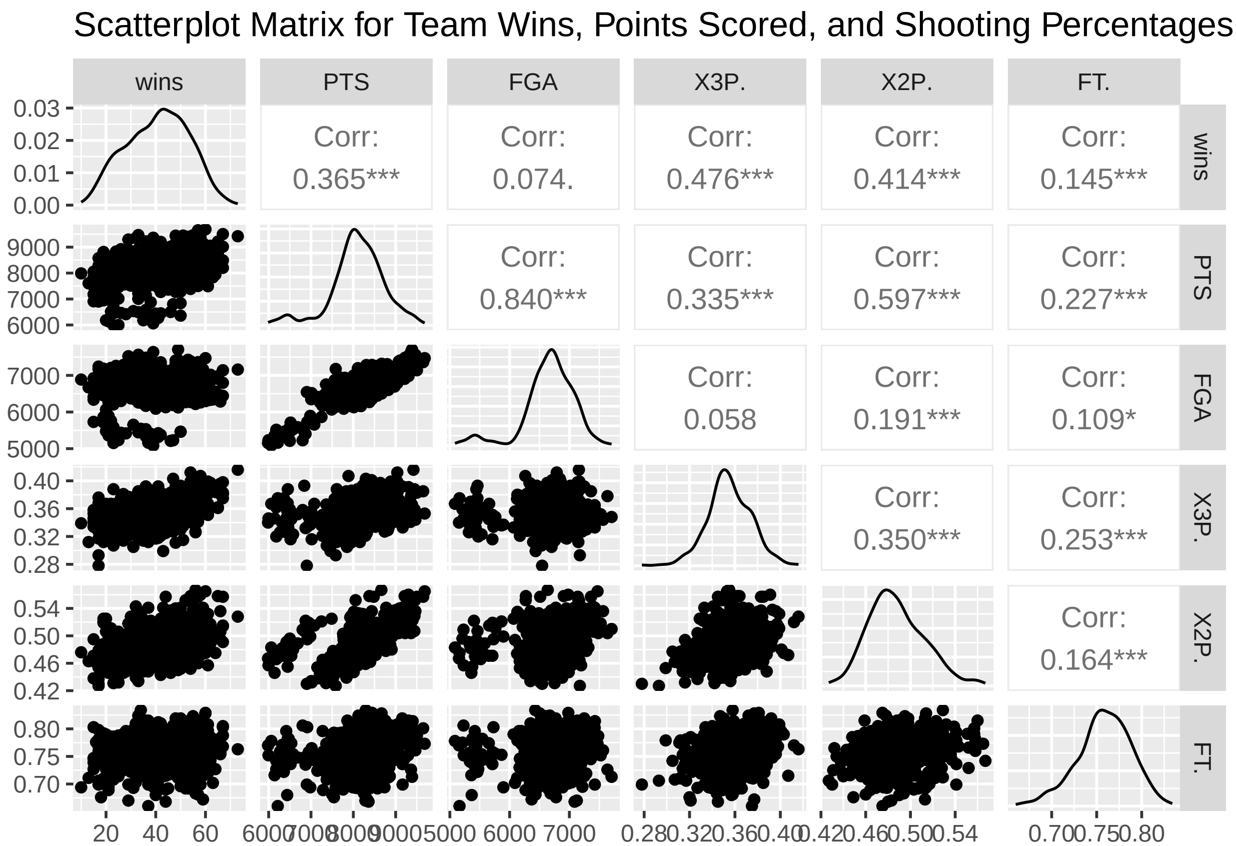


Basic Visualization (Scatterplot Matrices)

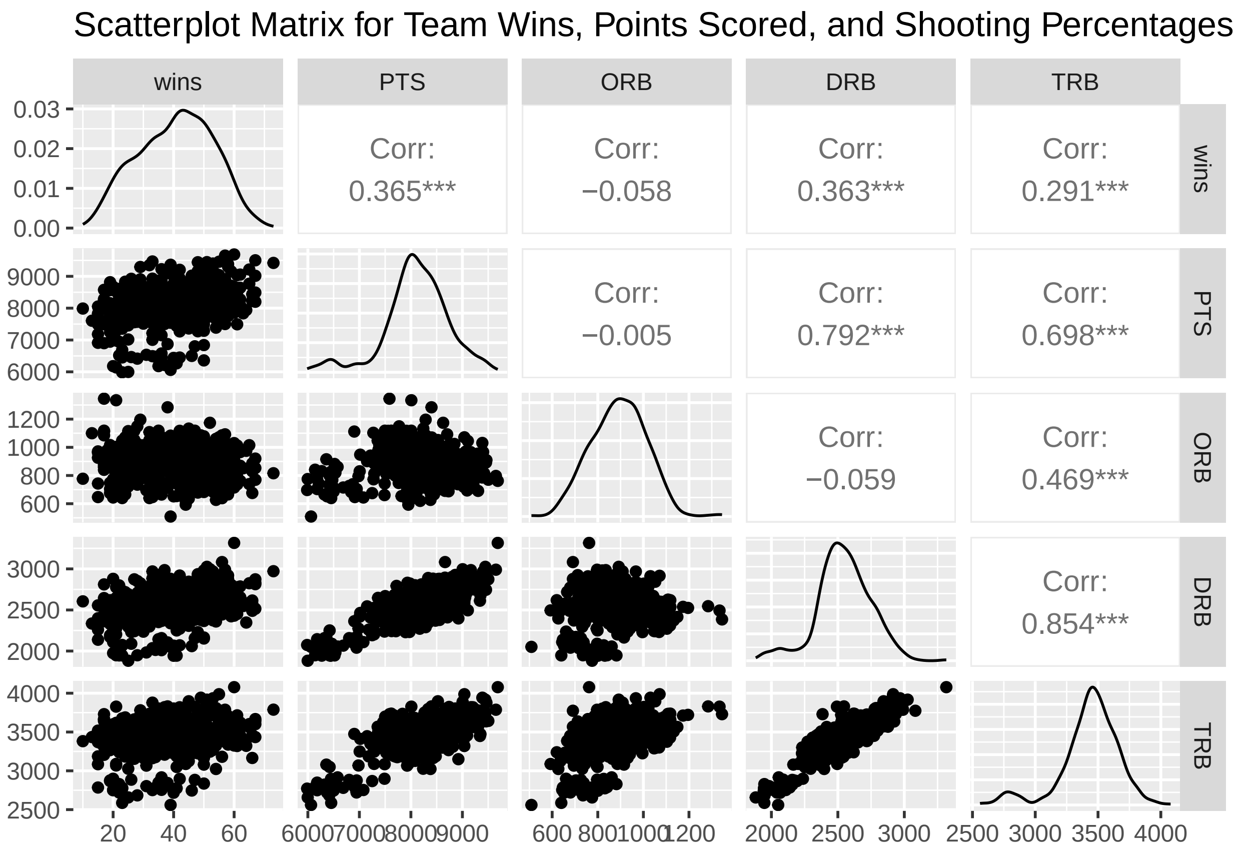
```
# ==========================================================
#                ScatterPlot Matrices
# ==========================================================
create_stat_to_wins <- function(stat) {
    stat$year <- as.integer(stat$year)
    merged_data <- left_join(stat, nba_team_wins, by = c("year", "team"))
    merged_data <- na.omit(merged_data)
    return(merged_data)
}

total_stats_wins <- create_stat_to_wins(total_stats)

# Relationship between wins and field goal attempts
ggpairs(total_stats_wins[,c("wins", "PTS", "FGA", "X3PA", "X2PA", "FTA")], title = "Scatterplot Matrix :
```



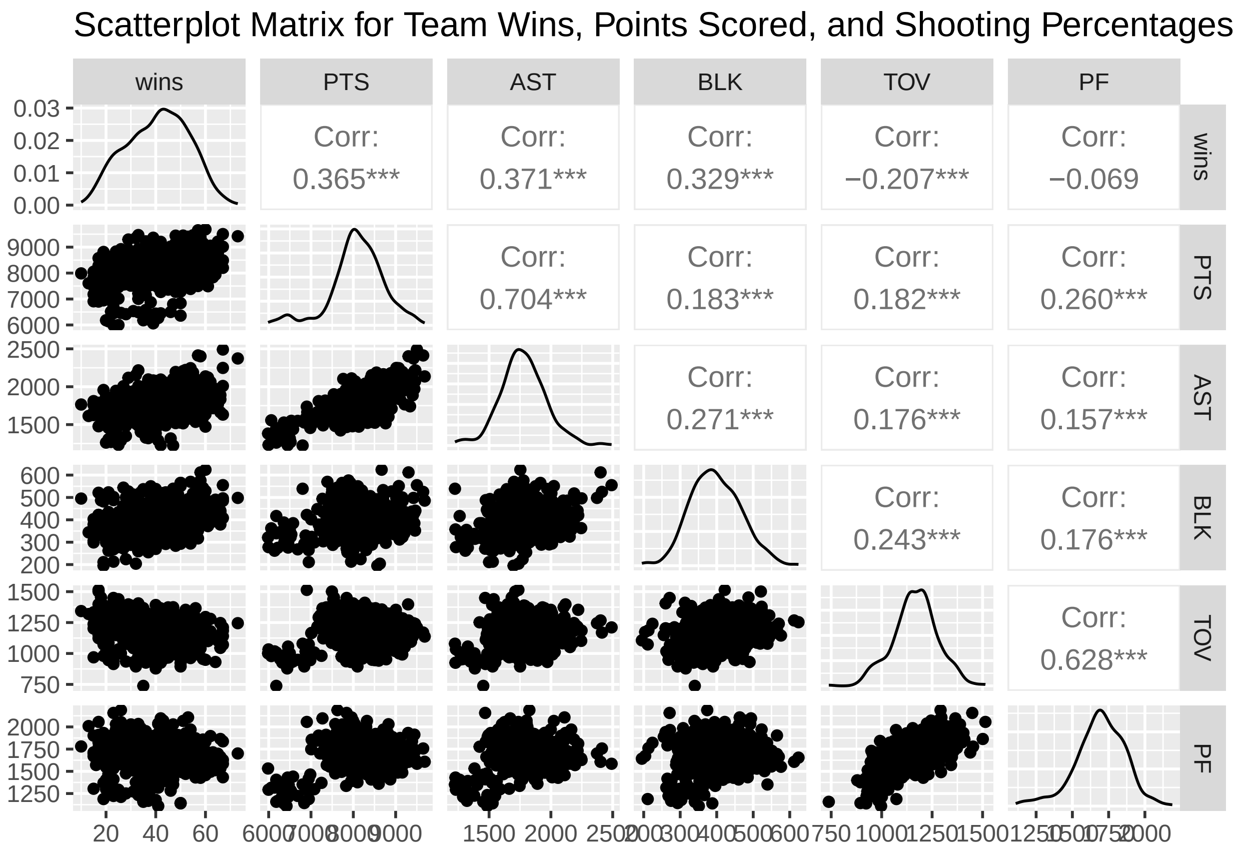

```r
# Relationship between wins and field goal percentage
ggpairs(total_stats_wins[,c("wins", "PTS", "FGA", "X3P.", "X2P.", "FT.")], title = "Scatterplot Matrix 
```




```r
# Relationship between wins and rebounds (*total rebounds*)
ggpairs(total_stats_wins[,c("wins", "PTS", "ORB", "DRB", "TRB")], title = "Scatterplot Matrix for Team W
```




```r
# Relationship between wins and assists, blocks, turnovers, personal fouls
ggpairs(total_stats_wins[,c("wins", "PTS", "AST", "BLK", "TOV", "PF")], title = "Scatterplot Matrix for
```

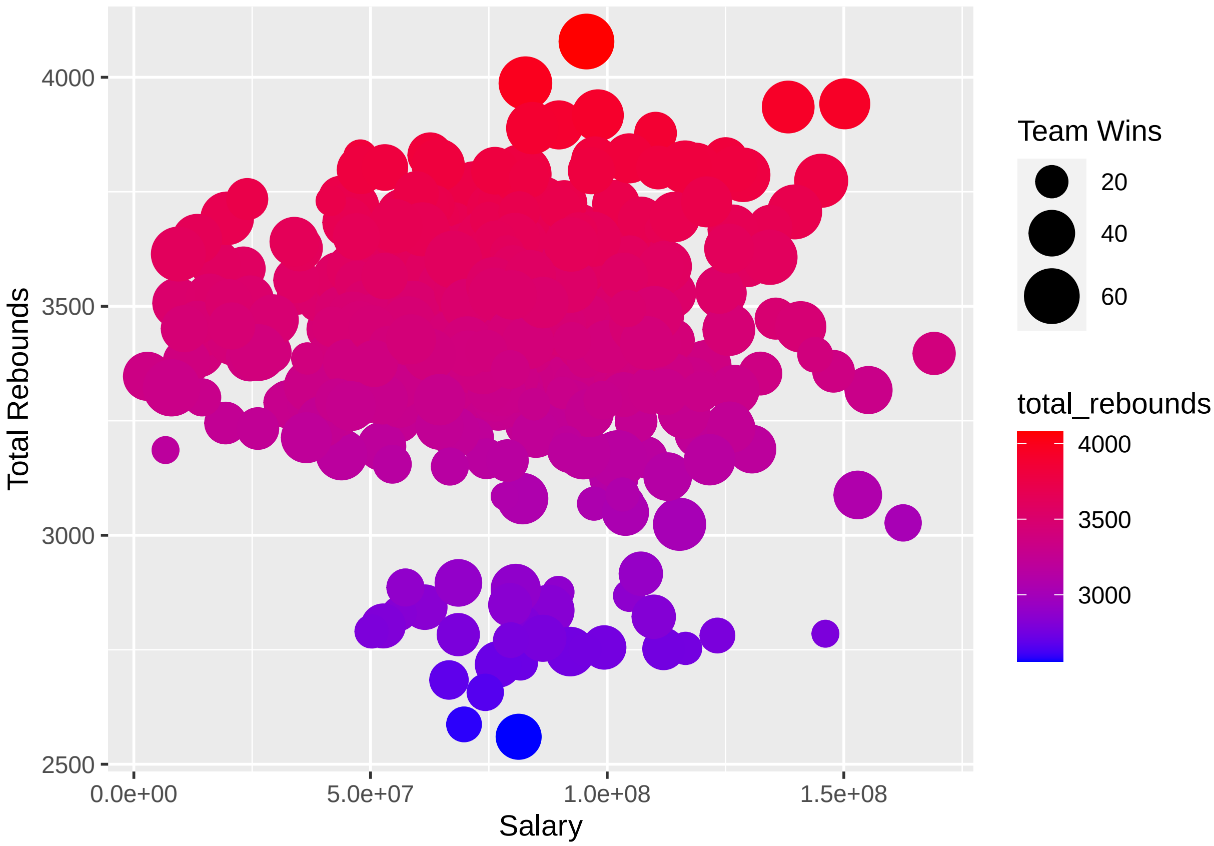


Basic Visualization (Boxplots)

```r
# ============================================================
#                Bubble Charts
# ============================================================
nba_total_salary_to_wins <- create_stat_to_wins(total_salary_by_team)
```

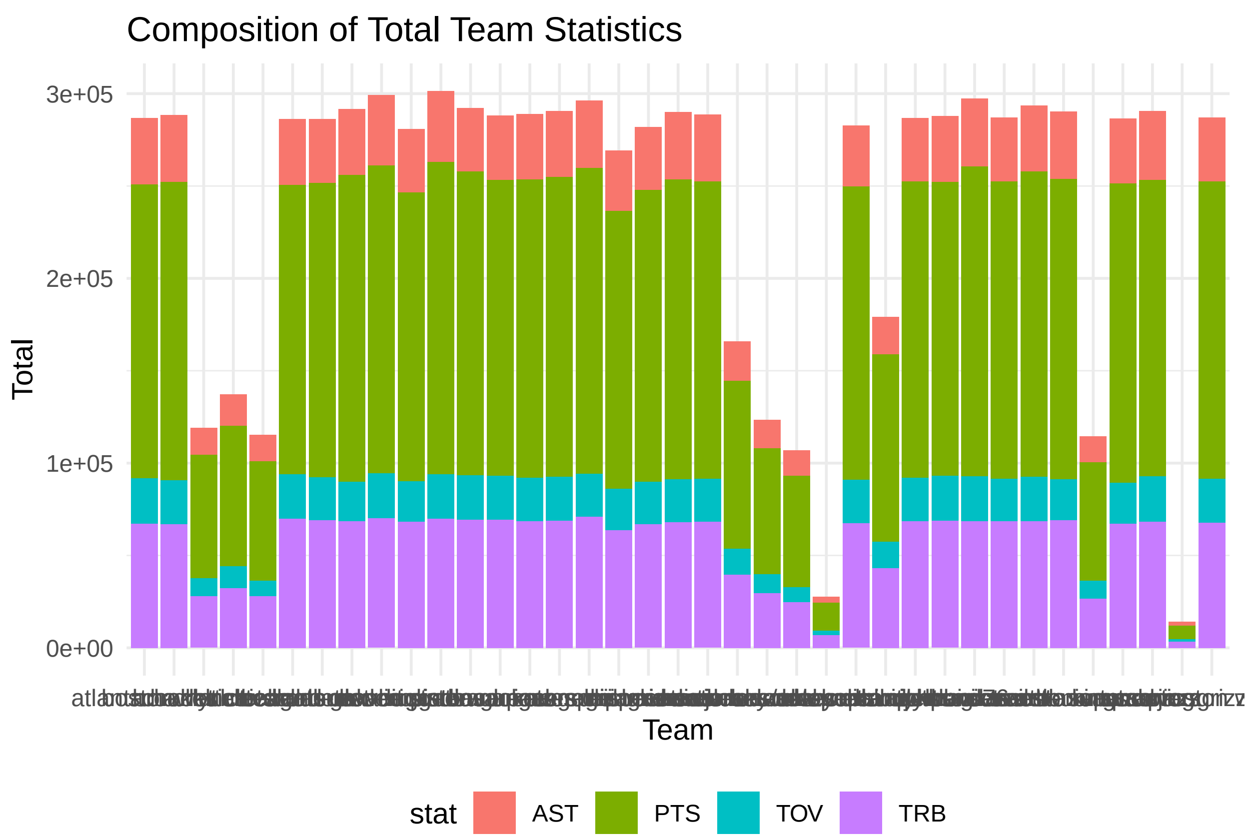
```
## Warning in create_stat_to_wins(total_salary_by_team): NAs introduced by coercion
```

```r
salary_wins_advanced <- left_join(advanced_stats, nba_total_salary_to_wins, by = c("year", "team"))
salary_wins_advanced <- na.omit(salary_wins_advanced)

salary_wins_advanced_total <- left_join(total_stats, nba_total_salary_to_wins, by = c("year", "team"))
salary_wins_advanced_total <- na.omit(salary_wins_advanced_total)

bubble_data <- data.frame(
  team_wins = salary_wins_advanced_total$wins,
  salary = salary_wins_advanced_total$total_salary,
  total_rebounds = salary_wins_advanced_total$TRB
)
ggplot(bubble_data, aes(x = salary, y = total_rebounds, size = team_wins, color = total_rebounds)) +
  geom_point() +
  scale_size_continuous(range = c(2, 10)) +
  labs(x = "Salary", y = "Total Rebounds", size = "Team Wins") + scale_color_gradient(low = "blue", hig
```



Basic Visualization (Stacked Bar Plot)

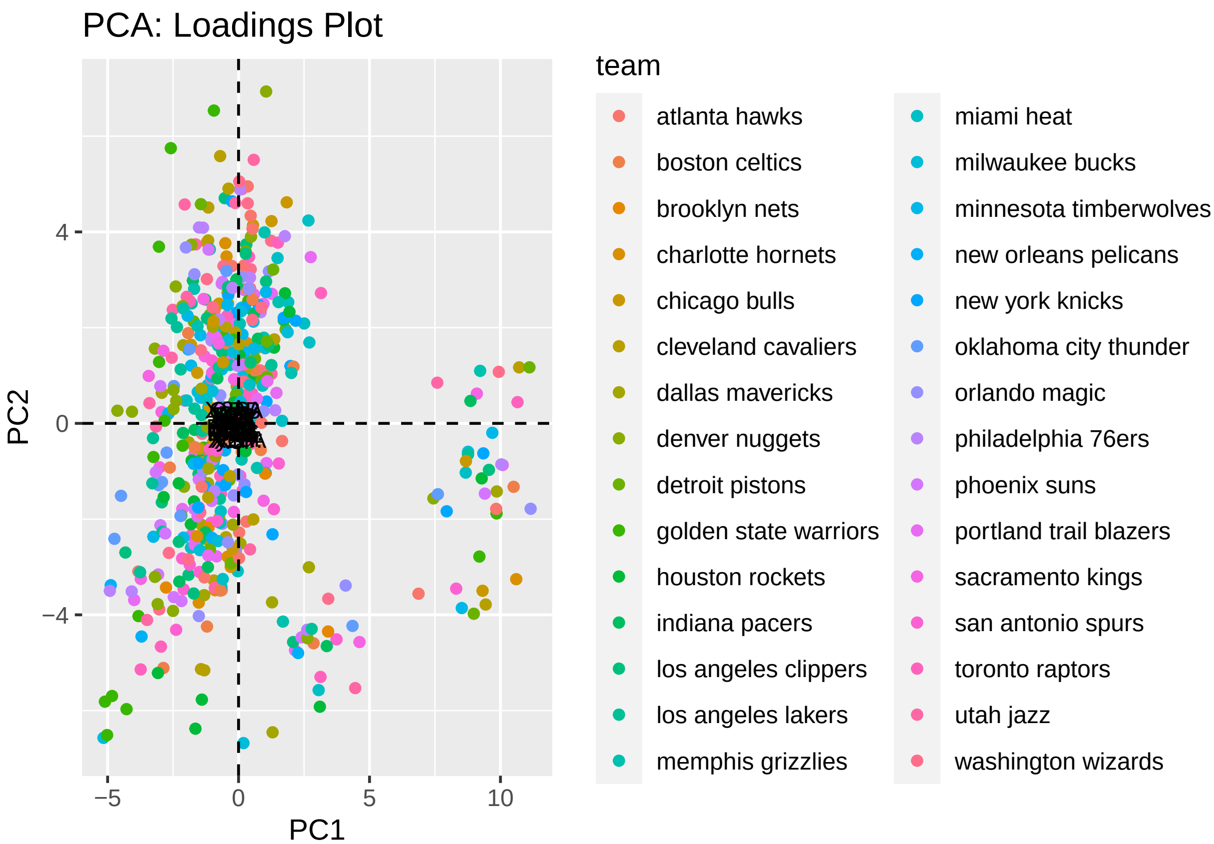
```
# ==========================================================
#               Stacked Bar Plot
# ==========================================================
# Creating a subset of total_stats dataframe for selected variables
team_stats <- total_stats[, c("team", "PTS", "AST", "TRB", "TOV")]

# Calculating the total of each variable by team
team_stats_summary <- aggregate(. ~ team, data = team_stats, FUN = sum)

# Converting data from wide to long format
team_stats_long <- gather(team_stats_summary, key = "stat", value = "total", -team)

# Creating the stacked bar chart
ggplot(team_stats_long, aes(x = team, y = total, fill = stat)) +
  geom_bar(stat = "identity") +
  ggtitle("Composition of Total Team Statistics") +
  xlab("Team") +
  ylab("Total") +
  theme_minimal() +
  theme(legend.position = "bottom")
```



Basic Analysis (Principal Component Analysis)

```
nba_total_to_wins <- create_stat_to_wins(total_stats)

# Removing non-numeric columns
numeric_data <- nba_total_to_wins[, sapply(nba_total_to_wins, is.numeric)]

# Removing rows with missing values
numeric_data <- na.omit(numeric_data)

scaled_data <- scale(numeric_data)
pca <- prcomp(scaled_data, center = TRUE, scale. = TRUE)

# Extracting the first two principal components
pca_data <- data.frame(pca$x[,1:2])

# Extracting the loadings
loadings <- data.frame(pca$rotation[,1:2])

# Performing PCA on numeric data
pca <- prcomp(numeric_data, center = TRUE, scale. = TRUE)

# Extract the first two principal components and create a data frame
pca_data <- data.frame(pca$x[, 1:2], team = nba_total_to_wins$team)

# Extracting loadings and creating a data frame
loadings <- data.frame(variable = colnames(numeric_data),
                       PC1 = pca$rotation[, 1],
```

```
                        PC2 = pca$rotation[, 2])

# Plotting the loadings on the first two principal components
ggplot(pca_data, aes(x = PC1, y = PC2)) +
  geom_point(aes(color = team)) +
  geom_hline(yintercept = 0, linetype = "dashed") +
  geom_vline(xintercept = 0, linetype = "dashed") +
  geom_text(data = loadings, aes(x = PC1, y = PC2, label = variable), size = 3) +
  ggtitle("PCA: Loadings Plot")
```



Generating Model using LASSO selection on Advanced NBA Statistics

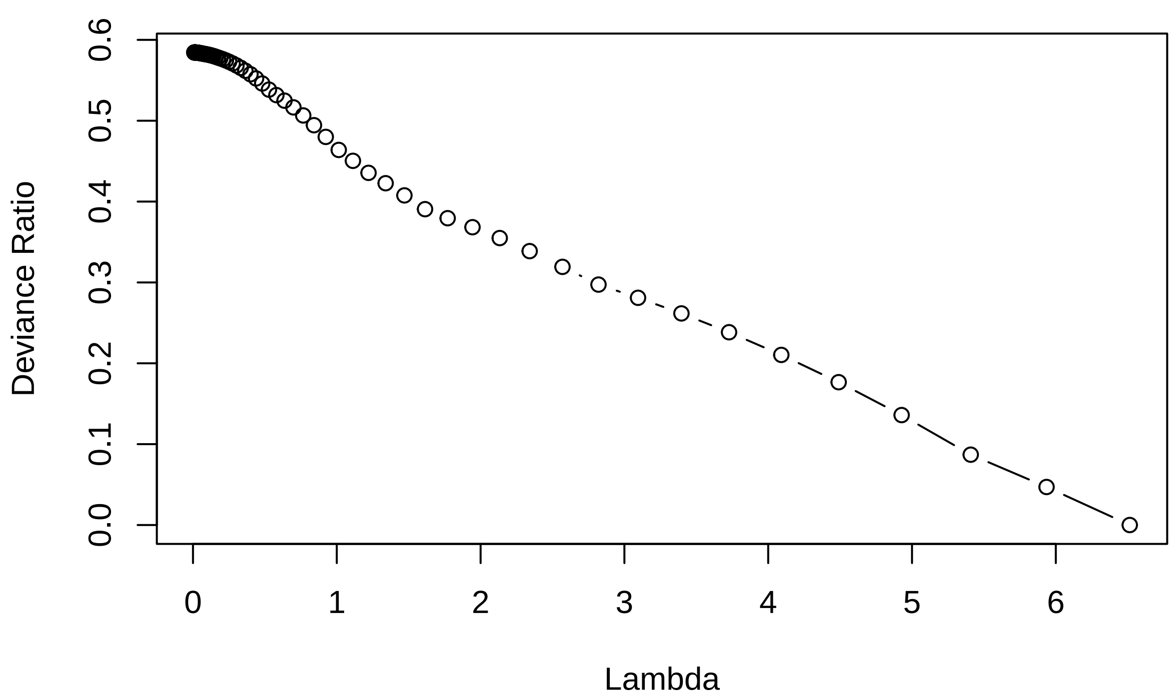
```
add_wins_vector <- function(data) {
    data_to_wins <- left_join(data, nba_team_wins, by = c("year", "team"))
    data_to_wins <- na.omit(data_to_wins)
    return(data_to_wins)
}

generate_advanced_stats_model <- function(data, exclude, selection_type, title) {

    data_to_wins <- add_wins_vector(data)

    rsquared_vec <- c()
    rsquared_adj_vec <- c()

    cols <- ncol(data_to_wins)
    column_iterator <- seq(from = 1, to = cols, by = 1)
```

```
    col_names_to_avoid <- exclude
    predictor_variables_df <- data_to_wins[!names(data_to_wins) %in% col_names_to_avoid]

    print("Number of Regressors:")
    print(dim(predictor_variables_df)[2])

    print("LASSO Model for prediction of NBA team success based on NBA Advanced Stats")
    print("=====================================================")
    print("Predictor Variables for LASSO model:")
    print(dim(predictor_variables_df))
    print(colnames(predictor_variables_df))

    wins_col_vector <- data_to_wins$wins

    print("=====================================================")
    print("Response Variable for LASSO Model:")
    print(length(wins_col_vector))

    stopifnot(!any(is.na(predictor_variables_df)), !any(is.na(wins_col_vector)))
    model <- glmnet(x = data.matrix(predictor_variables_df), y = data.matrix(wins_col_vector), alpha = a
    num_cols <- dim(model$beta)[2]

    print("=====================================================")
    print("Lambda Values tested by LASSO model:")
    print("------------------------------------------------------")
    print(model$lambda[num_cols])
    print("")
    print("Beta Values generated by LASSO model (best lambda):")
    print("------------------------------------------------------")
    print(model$beta[,num_cols])
    print("")
    print("LASSO model R-squared:")
    print("------------------------------------------------------")
    print(model$dev.ratio[num_cols])
    print("=====================================================")

    # Creating a plot of the lambda values tested by the Lasso model
    plot(model$lambda, model$dev.ratio, type = "b", xlab = "Lambda", ylab = "Deviance Ratio", main = tit

    # Create a sample matrix
    beta_vals <- model$beta[,num_cols]
    barplot(beta_vals, col="blue", main=title, las=2)
    par(cex.axis=0.8, cex.main=1.0)

    return (model)
}

aparam <- c("Rk", "team", "W","L", "PW", "PL", "MOV", "Arena", "wins", "SRS", "ORtg", "DRtg", "NRtg", "
aparam_min <- c("Rk", "team", "W","L", "PW", "PL", "MOV", "Arena", "wins", "ORtg", "NRtg", "TS.", "eFG.

advanced_stats_lasso_model <- generate_advanced_stats_model(advanced_stats, aparam, 1, "Lasso Model: Lan
```

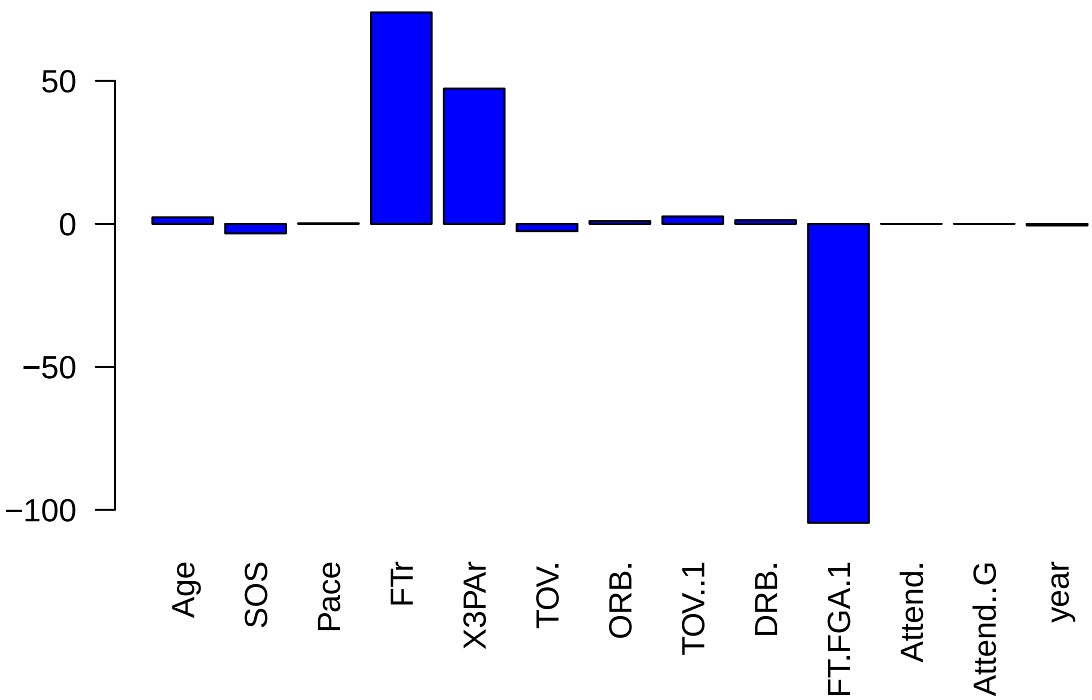
```
## [1] "Number of Regressors:"
## [1] 13
```

```
## [1] "LASSO Model for prediction of NBA team success based on NBA Advanced Stats"
## [1] "=================================================="
## [1] "Predictor Variables for LASSO model:"
## [1] 593  13
##  [1] "Age"       "SOS"       "Pace"      "FTr"       "X3PAr"     "TOV."
##  [7] "ORB."      "TOV..1"    "DRB."      "FT.FGA.1"  "Attend."   "Attend..G"
## [13] "year"
## [1] "=================================================="
## [1] "Response Variable for LASSO Model:"
## [1] 593
## [1] "=================================================="
## [1] "Lambda Values tested by LASSO model:"
## [1] "---------------------------------------------------"
## [1] 0.008814152
## [1] ""
## [1] "Beta Values generated by LASSO model (best lambda):"
## [1] "---------------------------------------------------"
##           Age           SOS          Pace           FTr         X3PAr
##  2.236430e+00 -3.372984e+00  1.430079e-01  7.391317e+01  4.727701e+01
##          TOV.          ORB.        TOV..1          DRB.      FT.FGA.1
## -2.600797e+00  9.467194e-01  2.559539e+00  1.284589e+00 -1.045326e+02
##       Attend.     Attend..G          year
##  4.014742e-05 -3.116696e-04 -5.938927e-01
## [1] ""
## [1] "LASSO model R-squared:"
## [1] "---------------------------------------------------"
## [1] 0.5844173
## [1] "=================================================="
```

Lasso Model: Lambda vs. Deviance Ratio (AParam)



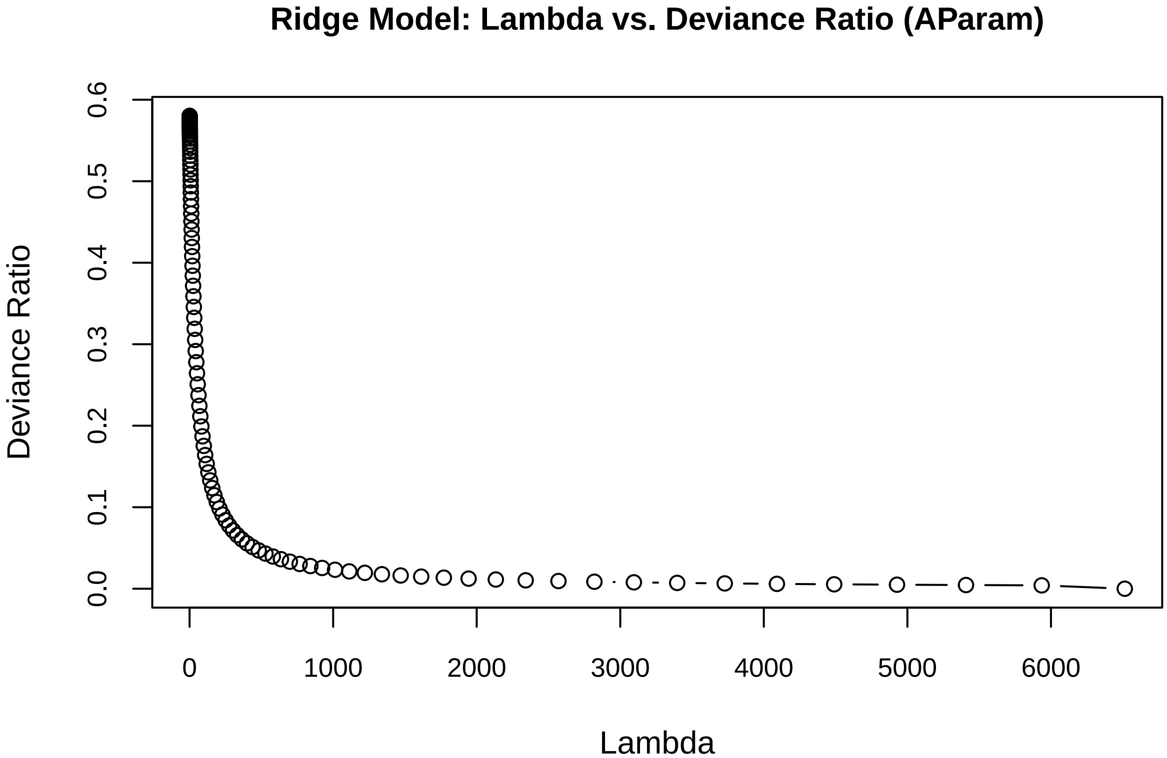
Lasso Model: Lambda vs. Deviance Ratio (AParam)

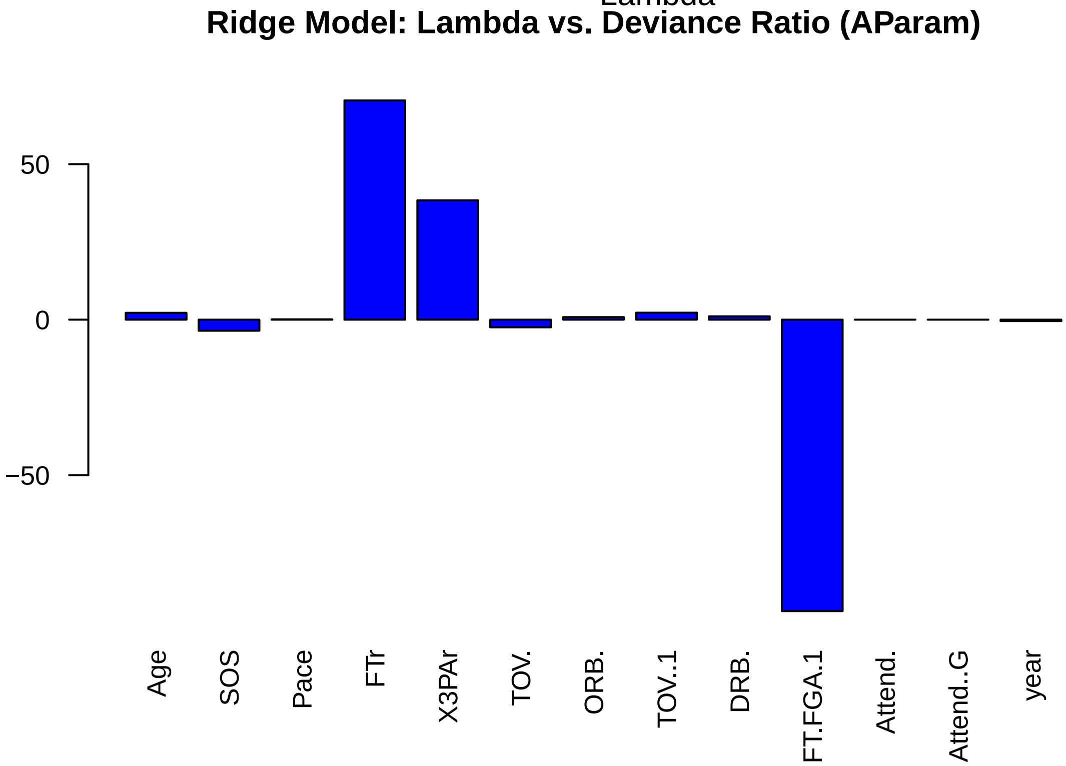



```r
advanced_stats_ridge_model <- generate_advanced_stats_model(advanced_stats, aparam, 0, "Ridge Model: La
```

```
## [1] "Number of Regressors:"
## [1] 13
## [1] "LASSO Model for prediction of NBA team success based on NBA Advanced Stats"
## [1] "================================================="
## [1] "Predictor Variables for LASSO model:"
## [1] 593  13
##  [1] "Age"        "SOS"        "Pace"       "FTr"        "X3PAr"      "TOV."
##  [7] "ORB."       "TOV..1"     "DRB."       "FT.FGA.1"   "Attend."    "Attend..G"
## [13] "year"
## [1] "================================================="
## [1] "Response Variable for LASSO Model:"
## [1] 593
## [1] "================================================="
## [1] "Lambda Values tested by LASSO model:"
## [1] "--------------------------------------------------"
## [1] 0.6514294
## [1] ""
## [1] "Beta Values generated by LASSO model (best lambda):"
## [1] "--------------------------------------------------"
##           Age           SOS          Pace           FTr         X3PAr
##  2.204324e+00 -3.558693e+00  1.125987e-01  7.050862e+01  3.839643e+01
##          TOV.          ORB.        TOV..1          DRB.      FT.FGA.1
## -2.465219e+00  8.113684e-01  2.245312e+00  1.074091e+00 -9.373863e+01
##       Attend.     Attend..G          year
##  3.271845e-05  4.496010e-05 -4.656873e-01
## [1] ""
## [1] "LASSO model R-squared:"
```

```
## [1] "---------------------------------------------------"
## [1] 0.5802607
## [1] "=================================================="
```

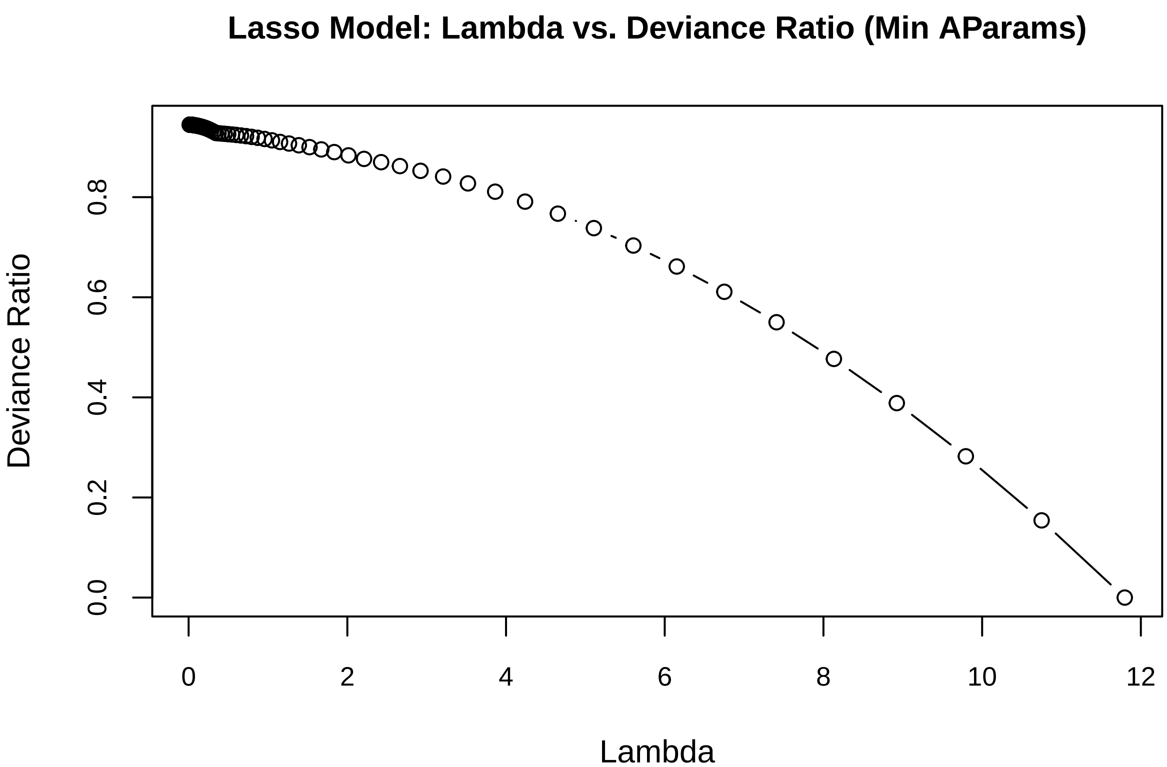


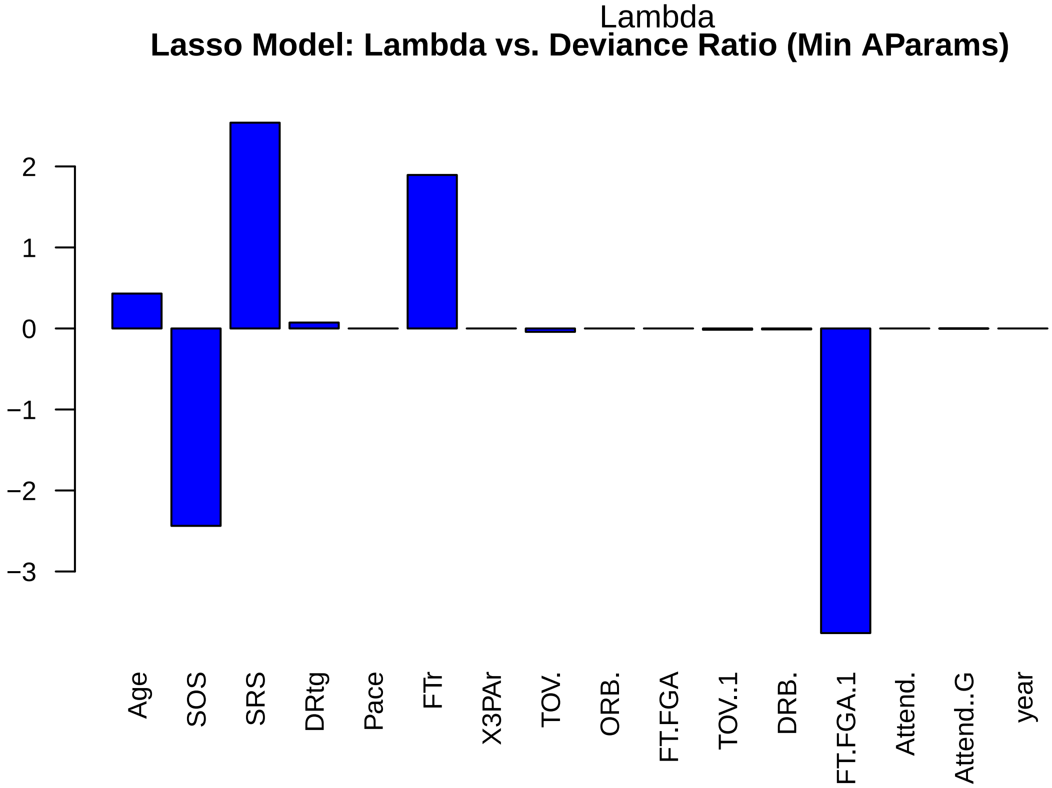



```
advanced_stats_lasso_model_min <- generate_advanced_stats_model(advanced_stats, aparam_min, 1, "Lasso Mo
```

```
## [1] "Number of Regressors:"
```

```
## [1] 16
## [1] "LASSO Model for prediction of NBA team success based on NBA Advanced Stats"
## [1] "=================================================="
## [1] "Predictor Variables for LASSO model:"
## [1] 593  16
##  [1] "Age"       "SOS"       "SRS"       "DRtg"      "Pace"      "FTr"
##  [7] "X3PAr"     "TOV."      "ORB."      "FT.FGA"    "TOV..1"    "DRB."
## [13] "FT.FGA.1"  "Attend."   "Attend..G" "year"
## [1] "=================================================="
## [1] "Response Variable for LASSO Model:"
## [1] 593
## [1] "=================================================="
## [1] "Lambda Values tested by LASSO model:"
## [1] "--------------------------------------------------"
## [1] 0.01325257
## [1] ""
## [1] "Beta Values generated by LASSO model (best lambda):"
## [1] "--------------------------------------------------"
##           Age           SOS           SRS          DRtg          Pace
##  4.309248e-01 -2.436652e+00  2.540225e+00  7.306266e-02  0.000000e+00
##           FTr         X3PAr          TOV.          ORB.        FT.FGA
##  1.894945e+00  0.000000e+00 -4.182925e-02  0.000000e+00  0.000000e+00
##        TOV..1          DRB.      FT.FGA.1       Attend.     Attend..G
## -1.538552e-02 -1.198701e-02 -3.760136e+00  4.281245e-05 -1.536244e-03
##          year
##  1.445727e-04
## [1] ""
## [1] "LASSO model R-squared:"
## [1] "--------------------------------------------------"
## [1] 0.9448032
## [1] "=================================================="
```

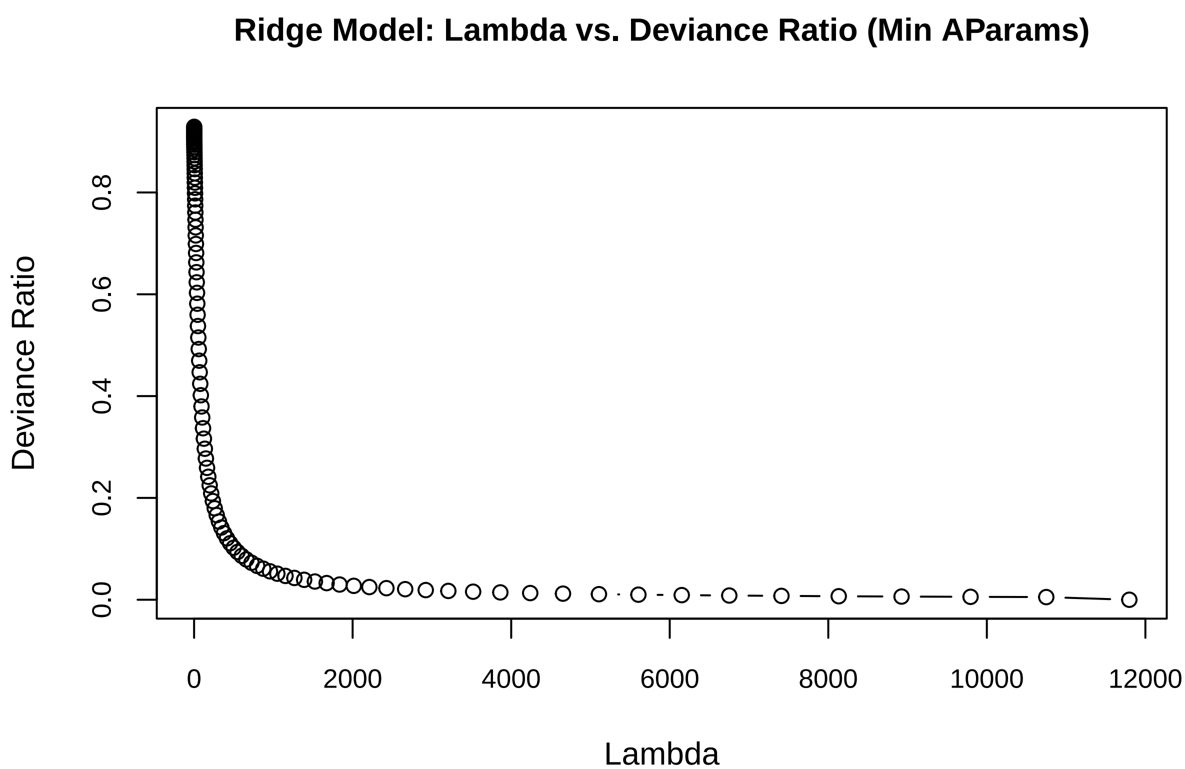




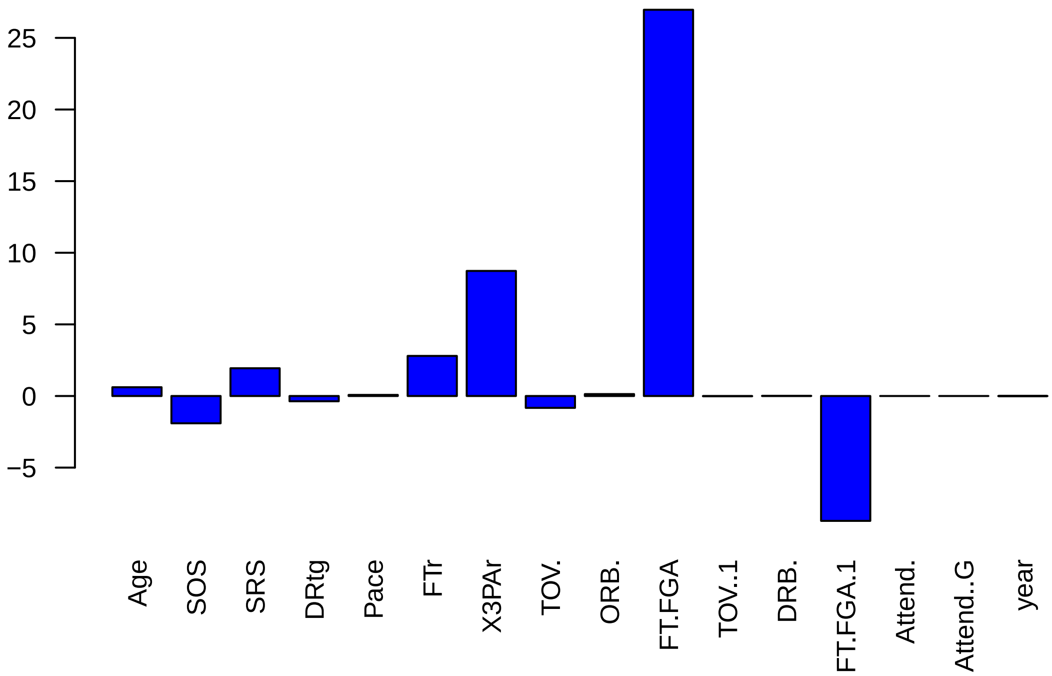

```
advanced_stats_ridge_model_min <- generate_advanced_stats_model(advanced_stats, aparam_min, 0, "Ridge M
```

```
## [1] "Number of Regressors:"
## [1] 16
## [1] "LASSO Model for prediction of NBA team success based on NBA Advanced Stats"
## [1] "=================================================="
```

```
## [1] "Predictor Variables for LASSO model:"
## [1] 593  16
##  [1] "Age"       "SOS"       "SRS"       "DRtg"      "Pace"      "FTr"
##  [7] "X3PAr"     "TOV."      "ORB."      "FT.FGA"    "TOV..1"    "DRB."
## [13] "FT.FGA.1"  "Attend."   "Attend..G" "year"
## [1] "===================================================="
## [1] "Response Variable for LASSO Model:"
## [1] 593
## [1] "===================================================="
## [1] "Lambda Values tested by LASSO model:"
## [1] "----------------------------------------------------"
## [1] 1.179764
## [1] ""
## [1] "Beta Values generated by LASSO model (best lambda):"
## [1] "----------------------------------------------------"
##           Age           SOS           SRS          DRtg          Pace
##  0.6124474652 -1.9051218388  1.9381152161 -0.3670043958  0.0762525827
##           FTr         X3PAr          TOV.          ORB.        FT.FGA
##  2.7978909351  8.7317968571 -0.8285605510  0.1341963825 26.9617968607
##        TOV..1          DRB.      FT.FGA.1       Attend.     Attend..G
## -0.0169751014  0.0107244807 -8.7177218195  0.0000259183 -0.0005954183
##          year
## -0.0033747656
## [1] ""
## [1] "LASSO model R-squared:"
## [1] "----------------------------------------------------"
## [1] 0.9288919
## [1] "===================================================="
```



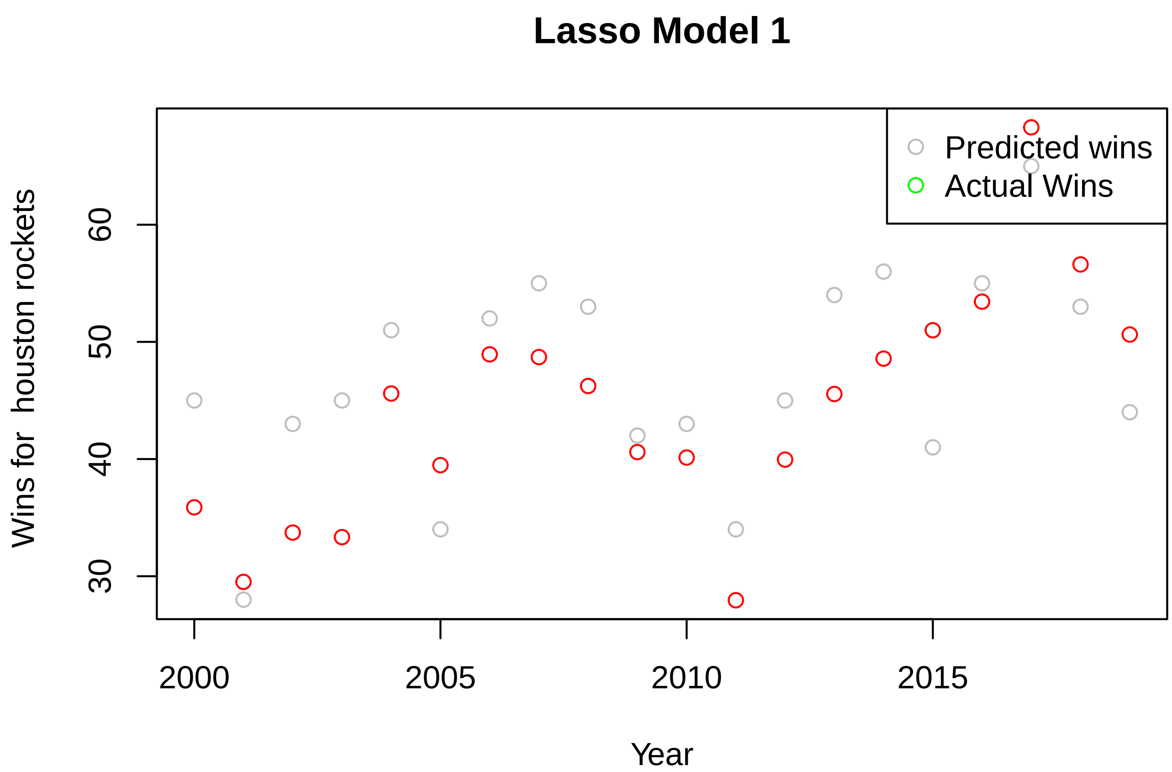
Ridge Model: Lambda vs. Deviance Ratio (Min AParams)



Model Evaluation

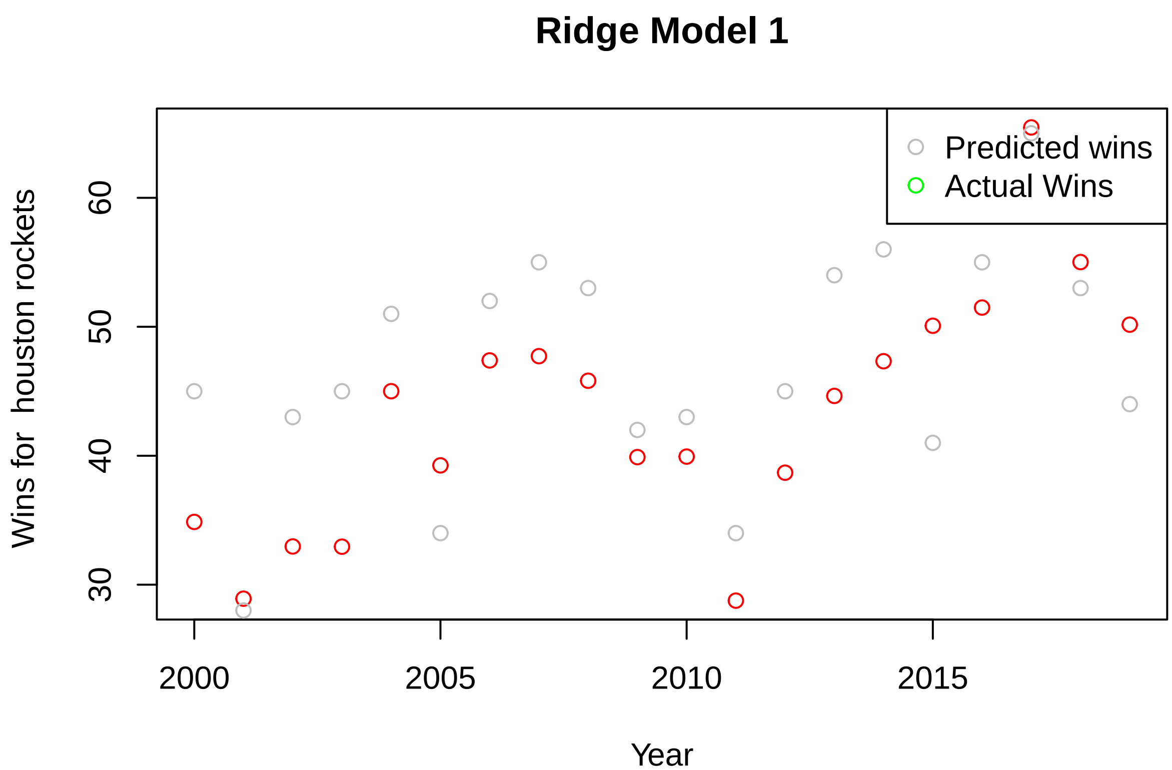
```
predict_team_success <- function(team, model, exclude, title) {
    advanced_stats_team <- advanced_stats[advanced_stats$team == team,]

    advanced_stats_team_wins <- add_wins_vector(advanced_stats_team)
    advanced_stats_team_wins_predictors <- advanced_stats_team_wins[!names(advanced_stats_team_wins) %i

    num_cols <- dim(model$beta)[2]
    best_lambda_value <- model$lambda[num_cols]

    print("Number of Regressors:")
    print(dim(advanced_stats_team_wins_predictors)[2])
    print(names(advanced_stats_team_wins_predictors))

    results <- predict(model, newx=data.matrix(advanced_stats_team_wins_predictors), s=best_lambda_valu

    plot(advanced_stats_team_wins$year, results, col = "red", xlab = "Year", ylab = paste("Wins for ", t
    points(advanced_stats_team_wins$year, advanced_stats_team_wins$wins, col = "grey")
    legend("topright", legend = c("Predicted wins", "Actual Wins"), col = c("grey", "green"), pch=1)

    return(results)
}

results_lasso <- predict_team_success("houston rockets", advanced_stats_lasso_model, aparam, "Lasso Mode
```

```
## [1] "Number of Regressors:"
## [1] 13
##  [1] "Age"       "SOS"       "Pace"      "FTr"       "X3PAr"     "TOV."
##  [7] "ORB."      "TOV..1"    "DRB."      "FT.FGA.1"  "Attend."   "Attend..G"
## [13] "year"
```



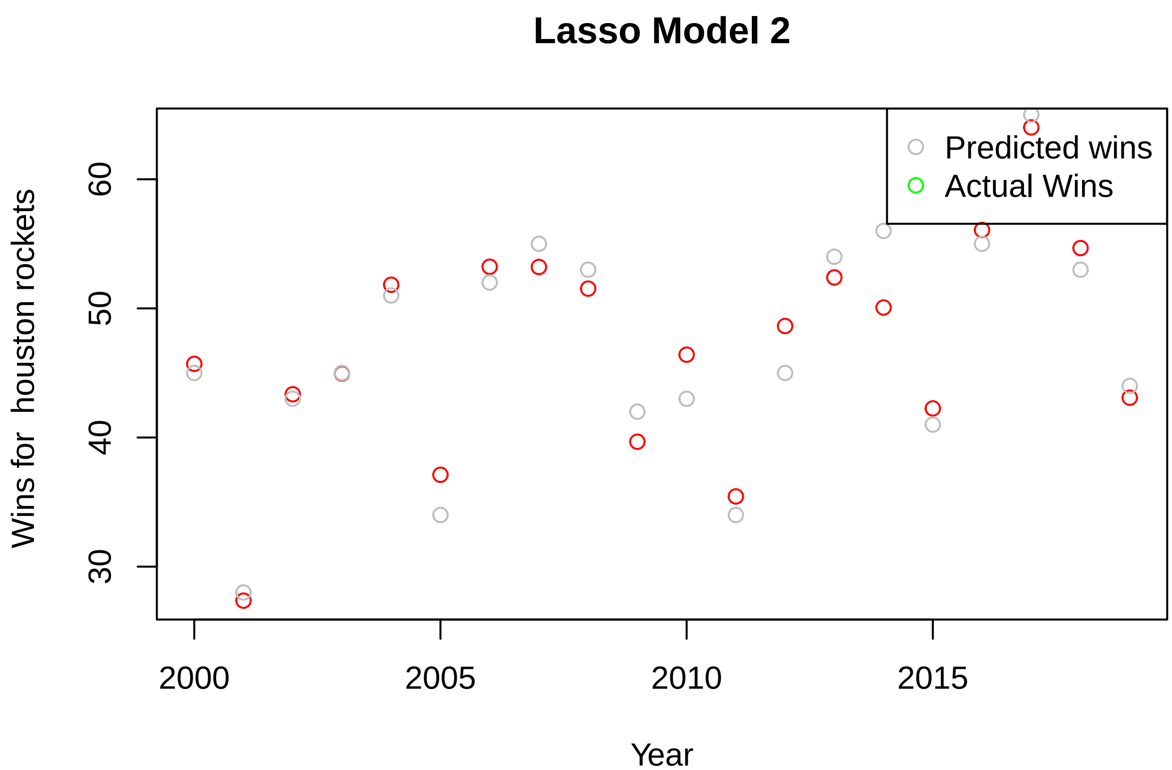

```r
results_ridge <- predict_team_success("houston rockets", advanced_stats_ridge_model, aparam, "Ridge Mode
```

```
## [1] "Number of Regressors:"
## [1] 13
##  [1] "Age"       "SOS"       "Pace"      "FTr"       "X3PAr"     "TOV."
##  [7] "ORB."      "TOV..1"    "DRB."      "FT.FGA.1"  "Attend."   "Attend..G"
## [13] "year"
```



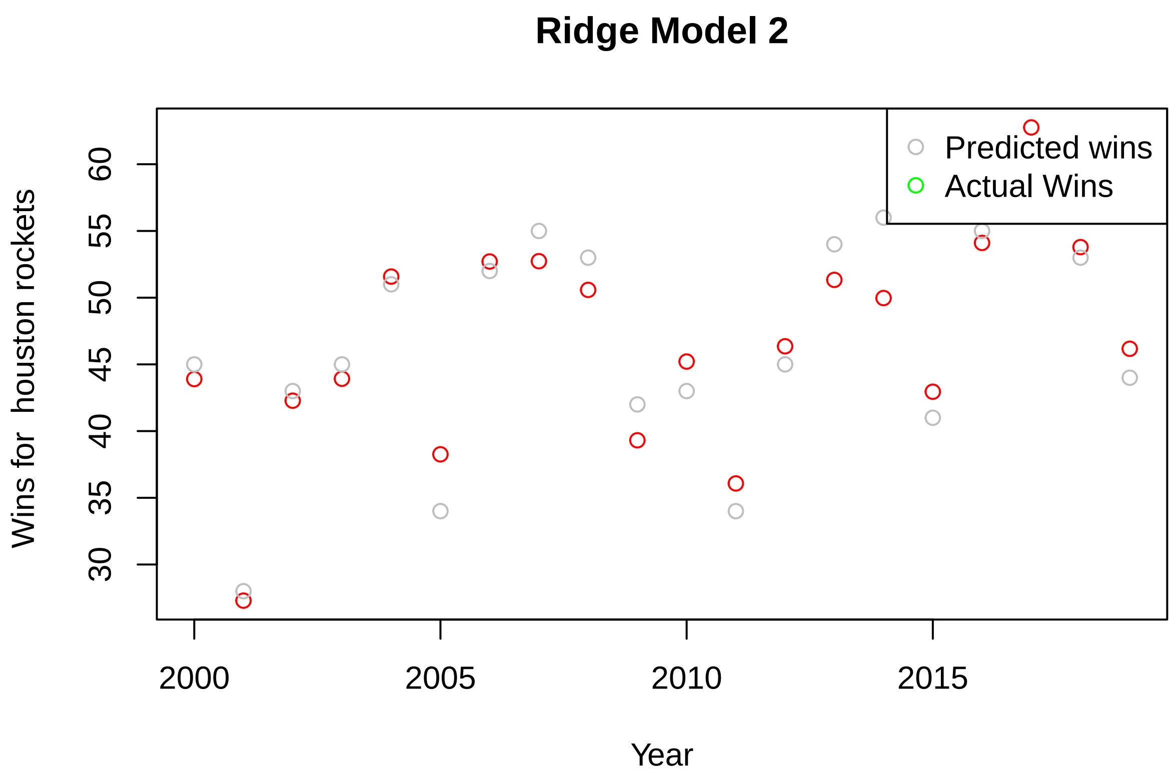

```
results_lasso_min <- predict_team_success("houston rockets", advanced_stats_lasso_model_min, aparam_min
```

```
## [1] "Number of Regressors:"
## [1] 16
##  [1] "Age"       "SOS"       "SRS"       "DRtg"      "Pace"      "FTr"
##  [7] "X3PAr"     "TOV."      "ORB."      "FT.FGA"    "TOV..1"    "DRB."
## [13] "FT.FGA.1"  "Attend."   "Attend..G" "year"
```




```r
results_ridge_min <- predict_team_success("houston rockets", advanced_stats_ridge_model_min, aparam_min
```

```
## [1] "Number of Regressors:"
## [1] 16
##  [1] "Age"        "SOS"        "SRS"        "DRtg"       "Pace"       "FTr"
##  [7] "X3PAr"      "TOV."       "ORB."       "FT.FGA"     "TOV..1"     "DRB."
## [13] "FT.FGA.1"   "Attend."    "Attend..G"  "year"
```

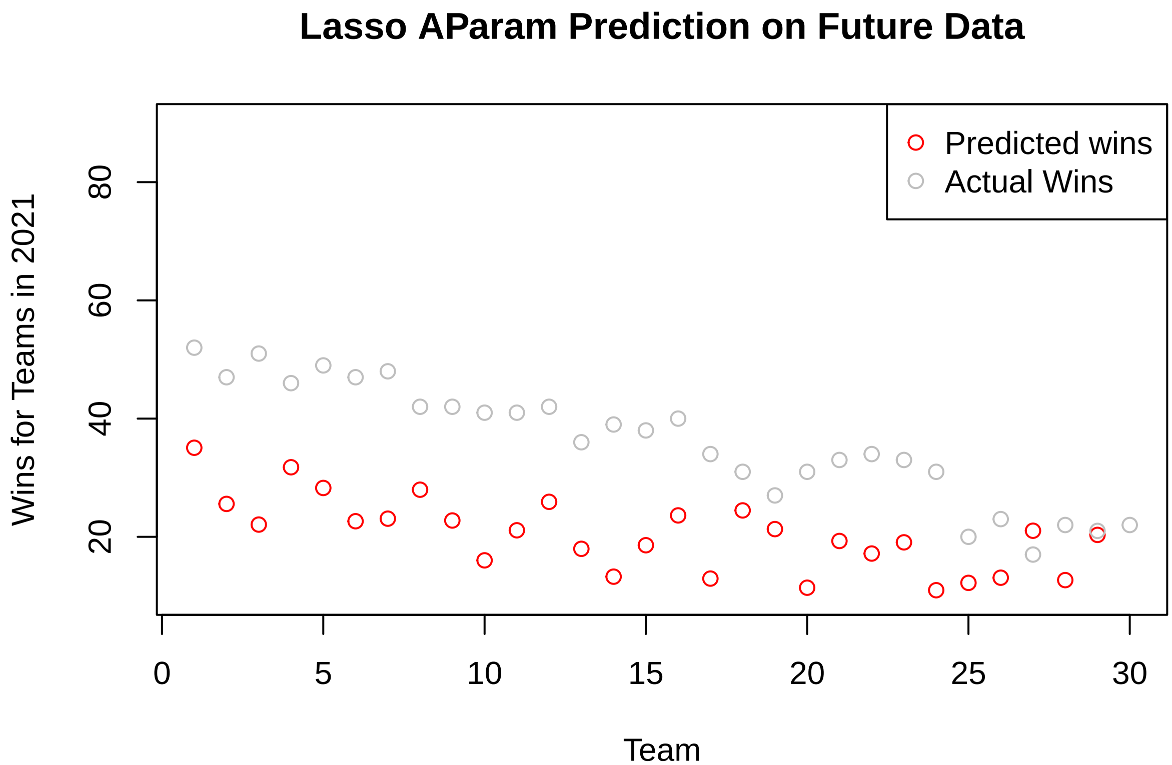


Applying our model to future data and conducting a residual analysis

```
predict_on_future_data <- function(future_data, model, exclude, title) {
      future_data_predictors <- future_data[!names(future_data) %in% exclude]

      year_str <- future_data$year[1]

      num_cols <- dim(model$beta)[2]
      best_lambda_value <- model$lambda[num_cols]

      print("Number of Regressors:")
      print(dim(future_data_predictors)[2])
      print(names(future_data_predictors))

      new_year_string <- paste("Wins for Teams in", year_str)

      results <- predict(model, newx=data.matrix(future_data_predictors), s=best_lambda_value, type="re

      team_inds <- seq(from=1, to=length(future_data$team), by=1)
      plot(team_inds, results, col = "red", xlab = "Team", ylab = new_year_string, ylim=c(10, 90), main=

      points(team_inds, future_data$W, col = "grey")
      legend("topright", legend = c("Predicted wins", "Actual Wins"), col = c("red", "grey"), pch=1)

      # Compute the residuals
      residuals <- results - future_data$W

      # Return both the predicted values and residuals
      return(list(predictions = results, residuals = residuals))
}
```

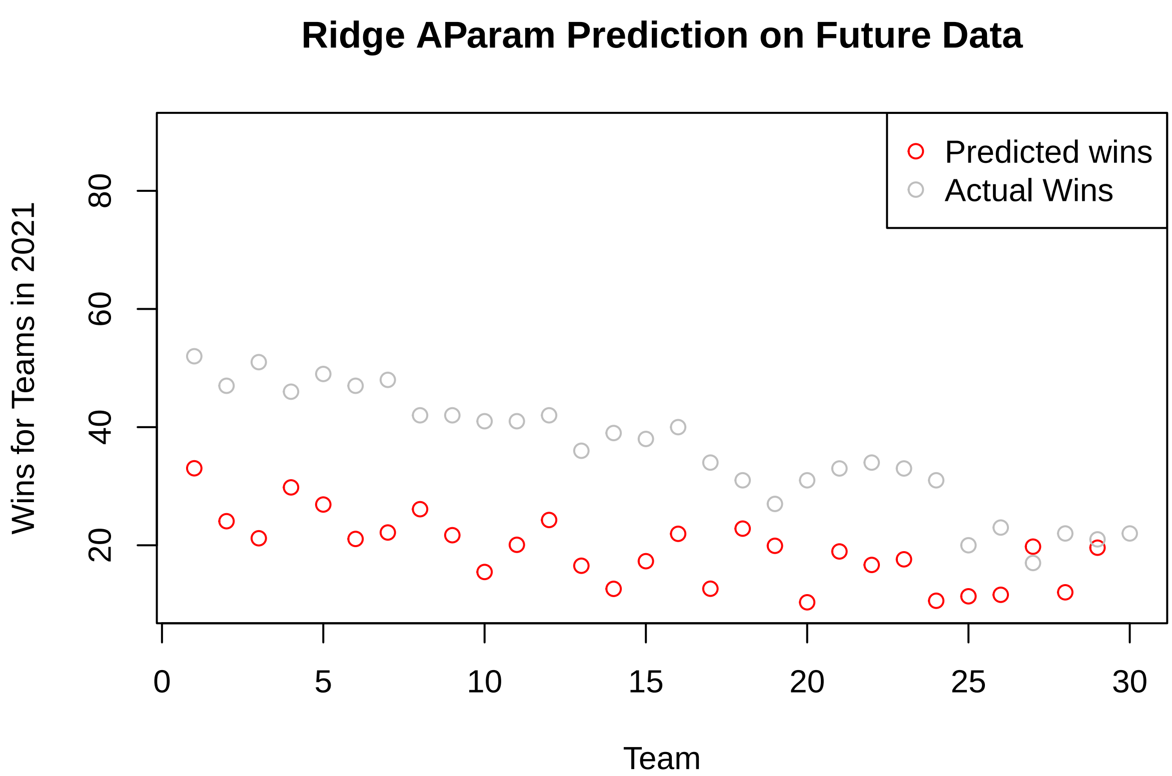
```
residuals_future_lasso <- predict_on_future_data(advanced_stats_2021, advanced_stats_lasso_model, aparam
```

```
## [1] "Number of Regressors:"
## [1] 13
##  [1] "Age"      "SOS"      "Pace"     "FTr"      "X3PAr"    "TOV."
##  [7] "ORB."     "TOV..1"   "DRB."     "FT.FGA.1" "Attend."  "Attend..G"
## [13] "year"
```



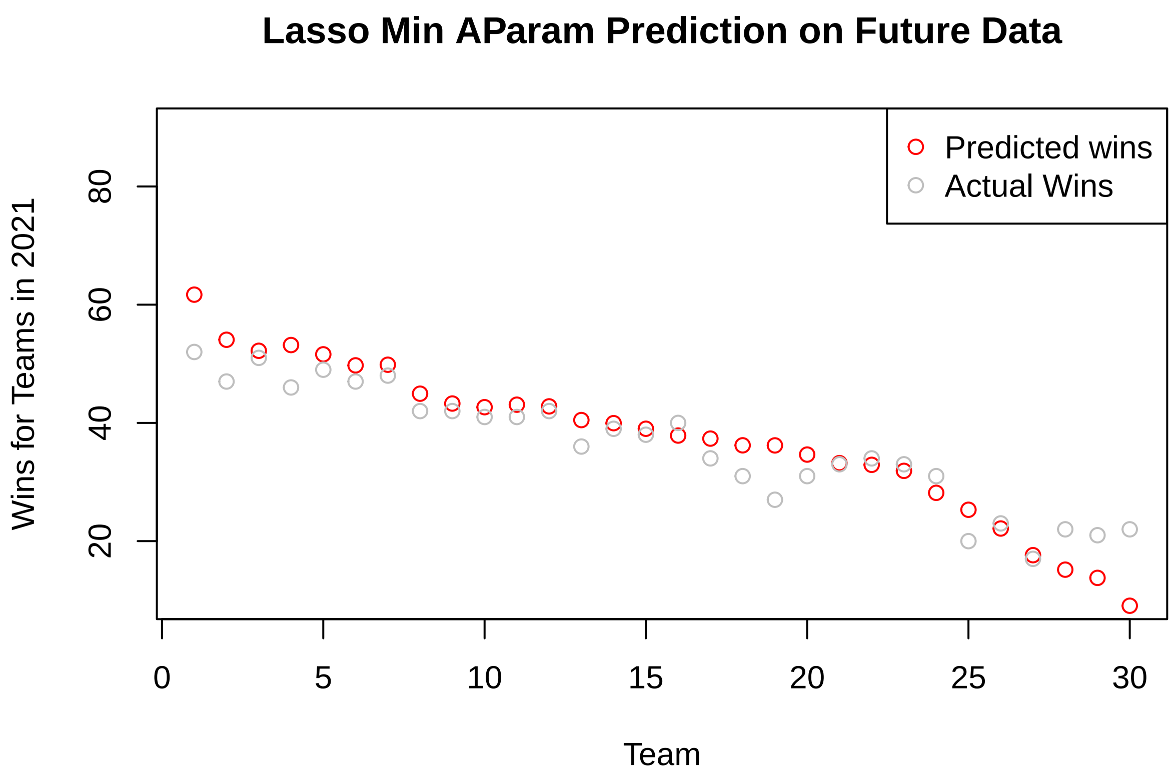

```
residuals_future_ridge <- predict_on_future_data(advanced_stats_2021, advanced_stats_ridge_model, aparam
```

```
## [1] "Number of Regressors:"
## [1] 13
##  [1] "Age"      "SOS"      "Pace"     "FTr"      "X3PAr"    "TOV."
##  [7] "ORB."     "TOV..1"   "DRB."     "FT.FGA.1" "Attend."  "Attend..G"
## [13] "year"
```



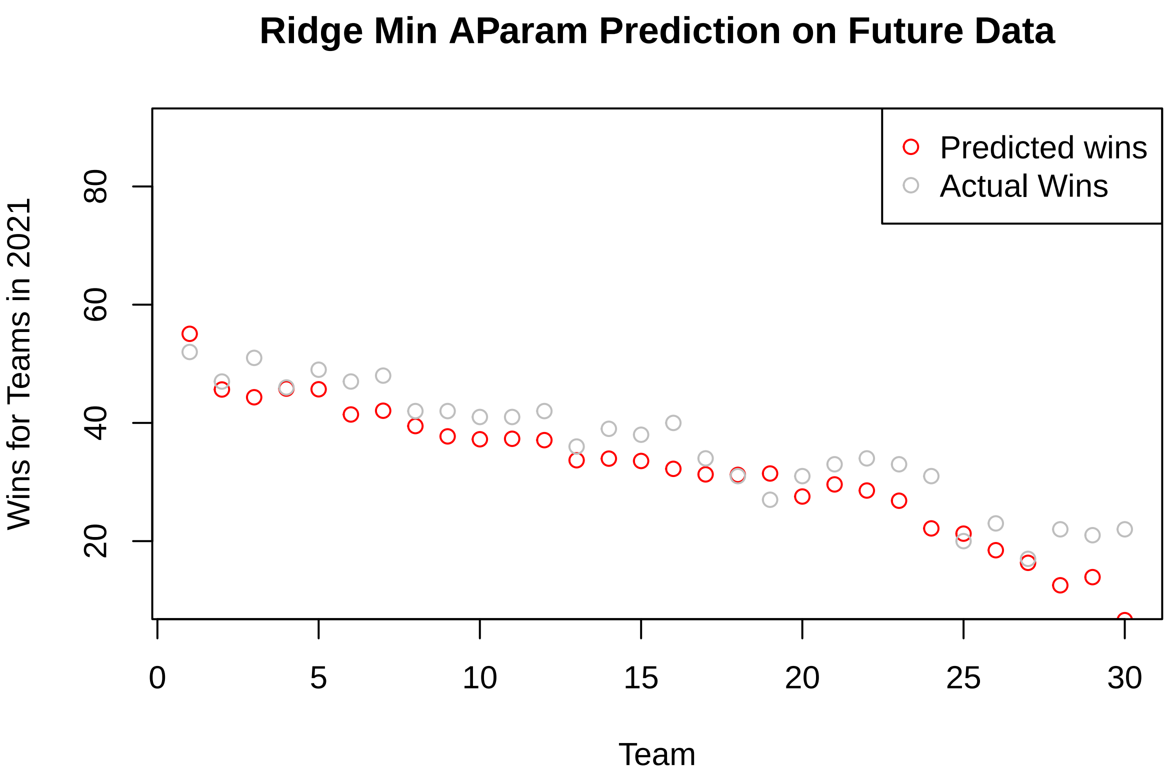

```
residuals_future_lasso_min <- predict_on_future_data(advanced_stats_2021, advanced_stats_lasso_model_mi
```

```
## [1] "Number of Regressors:"
## [1] 16
##  [1] "Age"       "SOS"       "SRS"       "DRtg"      "Pace"      "FTr"
##  [7] "X3PAr"     "TOV."      "ORB."      "FT.FGA"    "TOV..1"    "DRB."
## [13] "FT.FGA.1"  "Attend."   "Attend..G" "year"
```




```
residuals_future_ridge_min <- predict_on_future_data(advanced_stats_2021, advanced_stats_ridge_model_mi
```

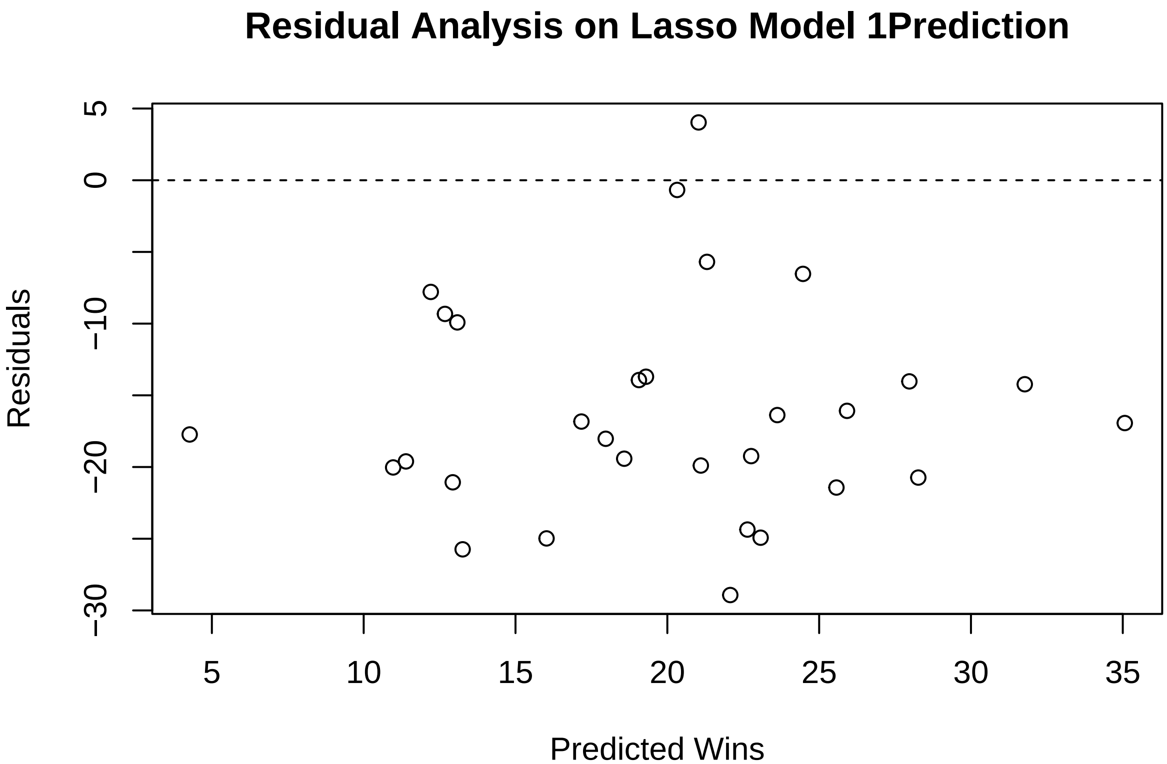
```
## [1] "Number of Regressors:"
## [1] 16
##  [1] "Age"        "SOS"        "SRS"        "DRtg"       "Pace"       "FTr"
##  [7] "X3PAr"      "TOV."       "ORB."       "FT.FGA"     "TOV..1"     "DRB."
## [13] "FT.FGA.1"   "Attend."    "Attend..G"  "year"
```



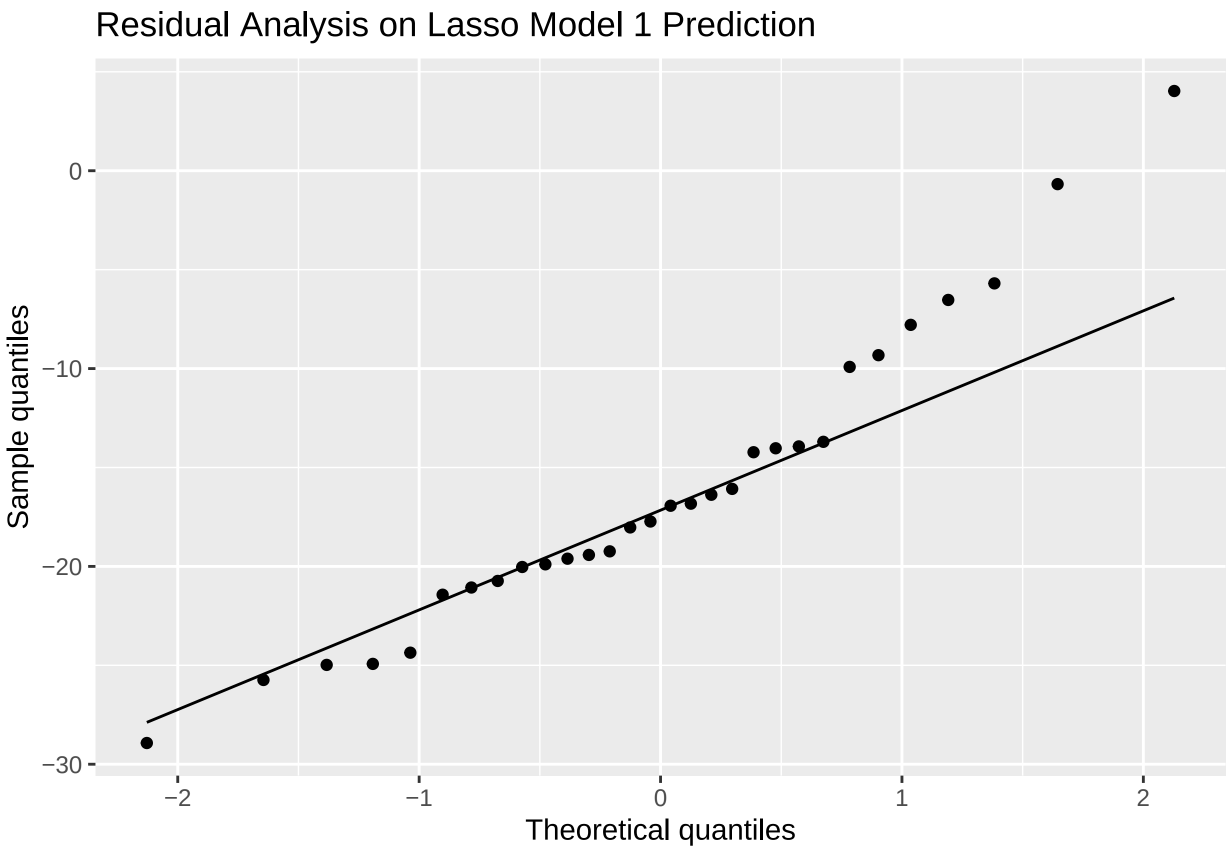
Conducting additional residual analysis

```
plot_residuals <- function(residuals, predicted_values, wins, residual_title) {
    # Step 1: Plotting the residuals
    # Plotting the predicted values against the residuals
    plot(predicted_values, residuals, xlab="Predicted Wins", ylab="Residuals", main=residual_title)
    # Plotting the actual wins as points to compare with predicted wins
    points(predicted_values, wins, col = "grey")
    # Adding a horizontal line at y=0 to indicate the zero residual line
    abline(h=0, lty=2)
}

check_residual_normality <- function(residuals, predicted_values, title) {
    # Step 2: Check for normality of residuals
    # Creating a histogram of the residuals
    ggplot(data = data.frame(residuals = residuals), aes(x = residuals)) +
      geom_histogram() +
      labs(x = "Residuals", y = "Frequency") +
      ggtitle(title)

    # Creating a QQ plot of the residuals
    ggplot(data = data.frame(residuals = residuals), aes(sample = residuals)) +
      stat_qq() +
      stat_qq_line() +
      labs(x = "Theoretical quantiles", y = "Sample quantiles") +
      ggtitle(title)
}

check_residual_hetero <- function(residuals, predicted_values, title) {
    # Step 3: Check for heteroscedasticity
    # Plotting the residuals against the predicted values
```

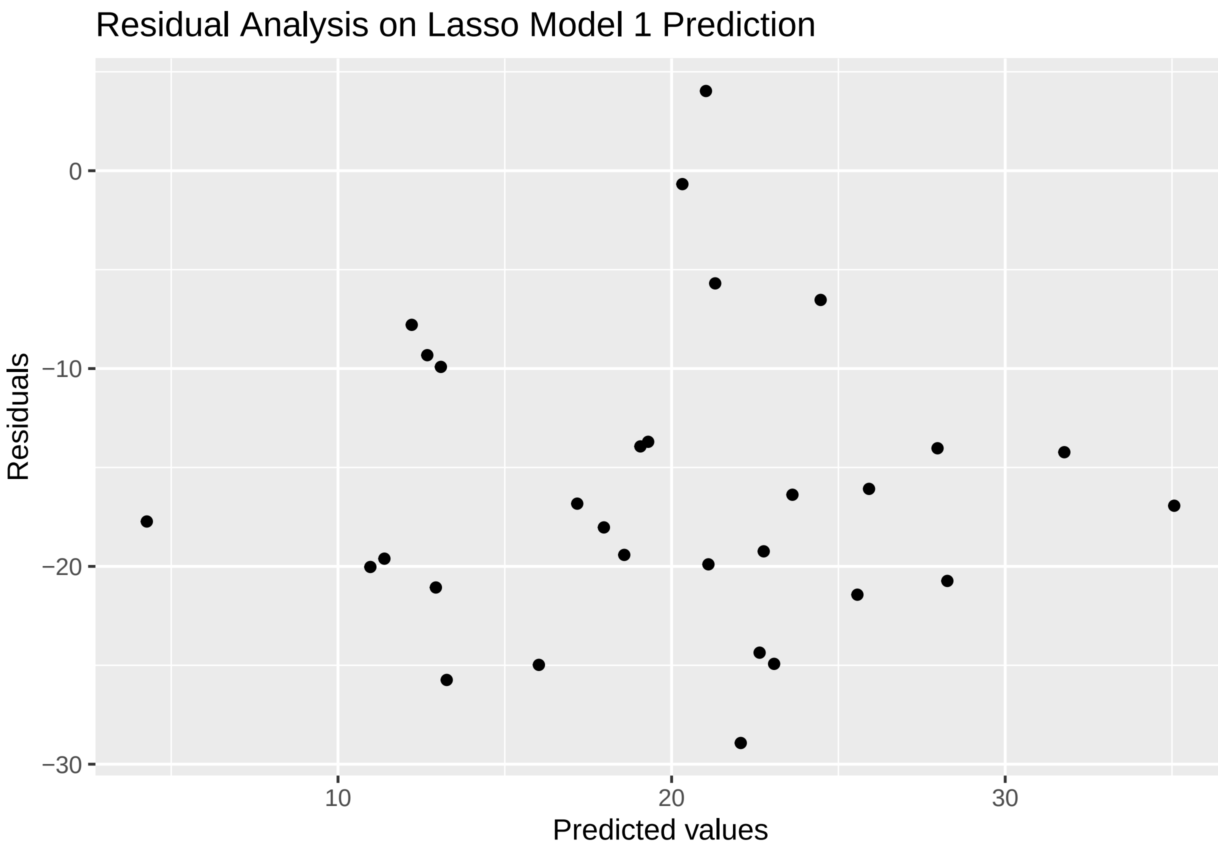
```
    ggplot(data = data.frame(residuals = residuals, predicted_values = predicted_values),
           aes(x = predicted_values, y = residuals)) +
      geom_point() +
      labs(x = "Predicted values", y = "Residuals") +
      ggtitle(title)
}

plot_residuals(residuals_future_lasso$residuals, residuals_future_lasso$predictions, advanced_stats_202
```



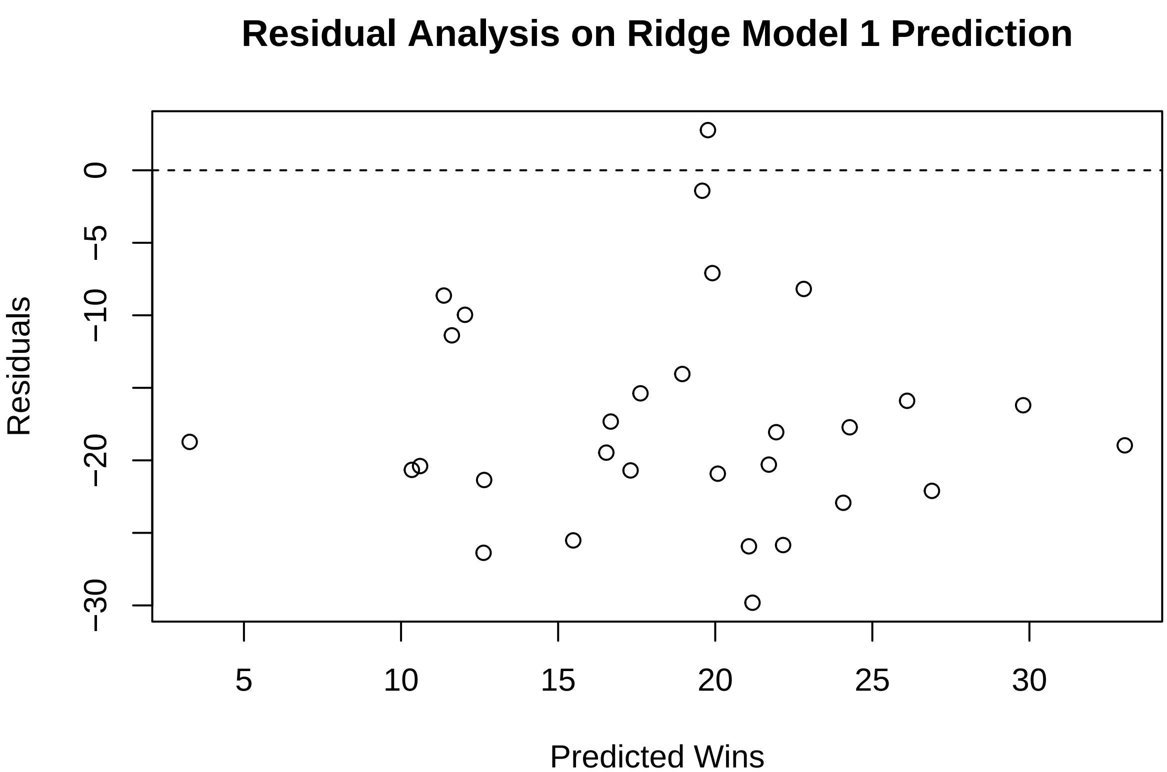

```
check_residual_normality(residuals_future_lasso$residuals, residuals_future_lasso$predictions, "Residua
```



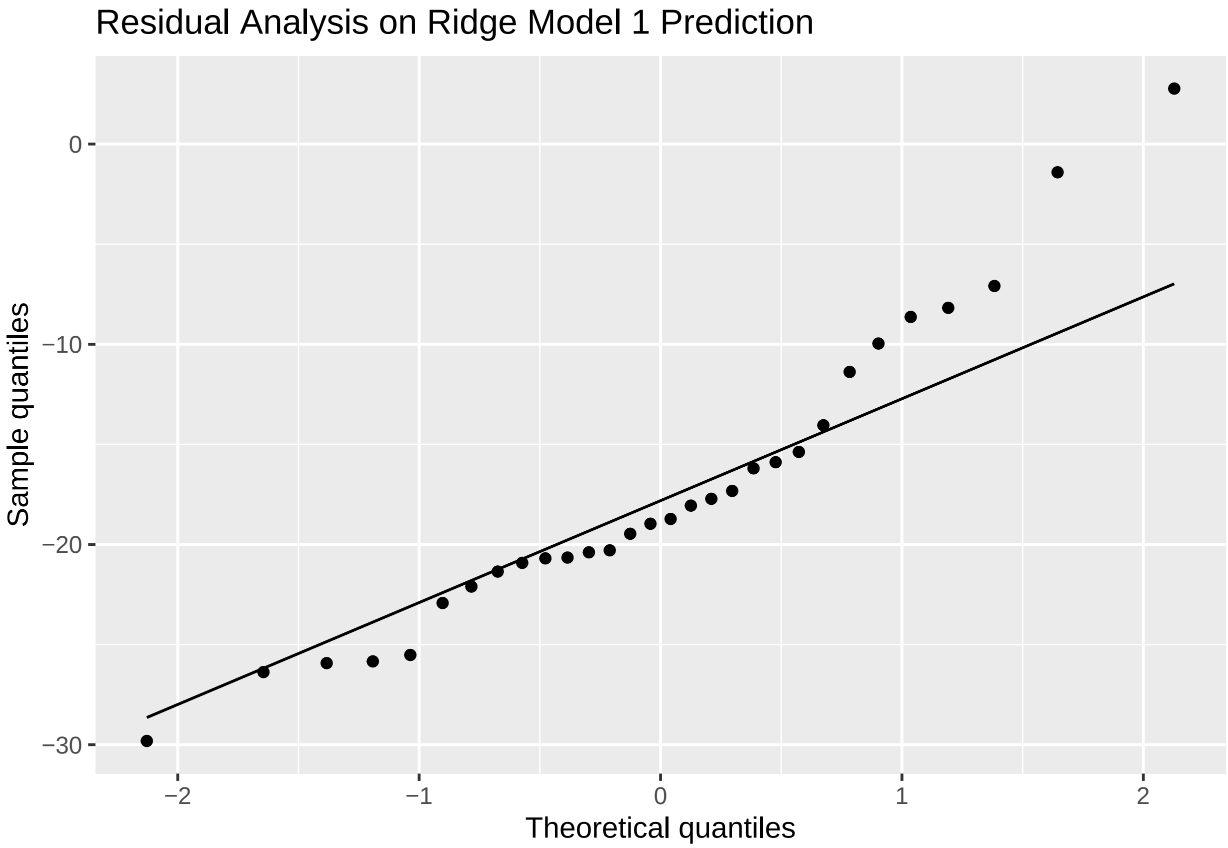

```
check_residual_hetero(residuals_future_lasso$residuals, residuals_future_lasso$predictions, "Residual An
```



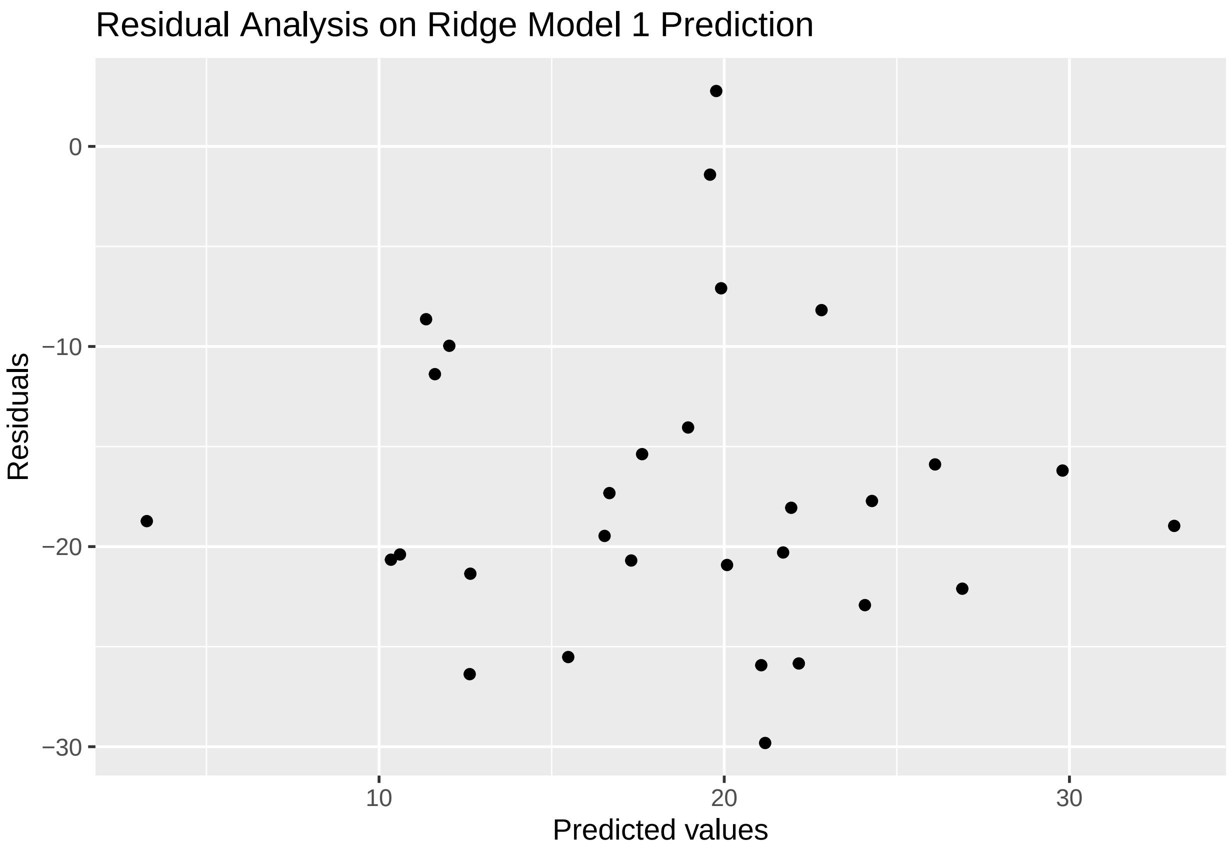

```r
plot_residuals(residuals_future_ridge$residuals, residuals_future_ridge$predictions, advanced_stats_202
```



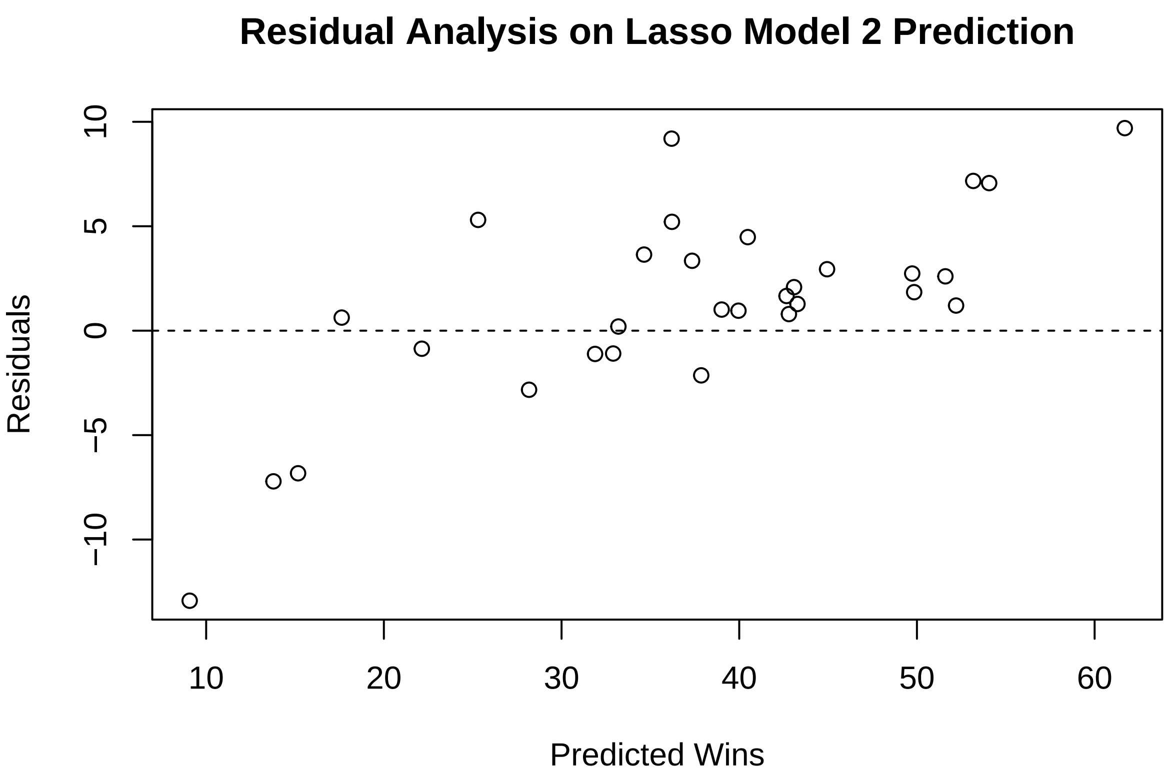
```
check_residual_normality(residuals_future_ridge$residuals, residuals_future_ridge$predictions, "Residual
```



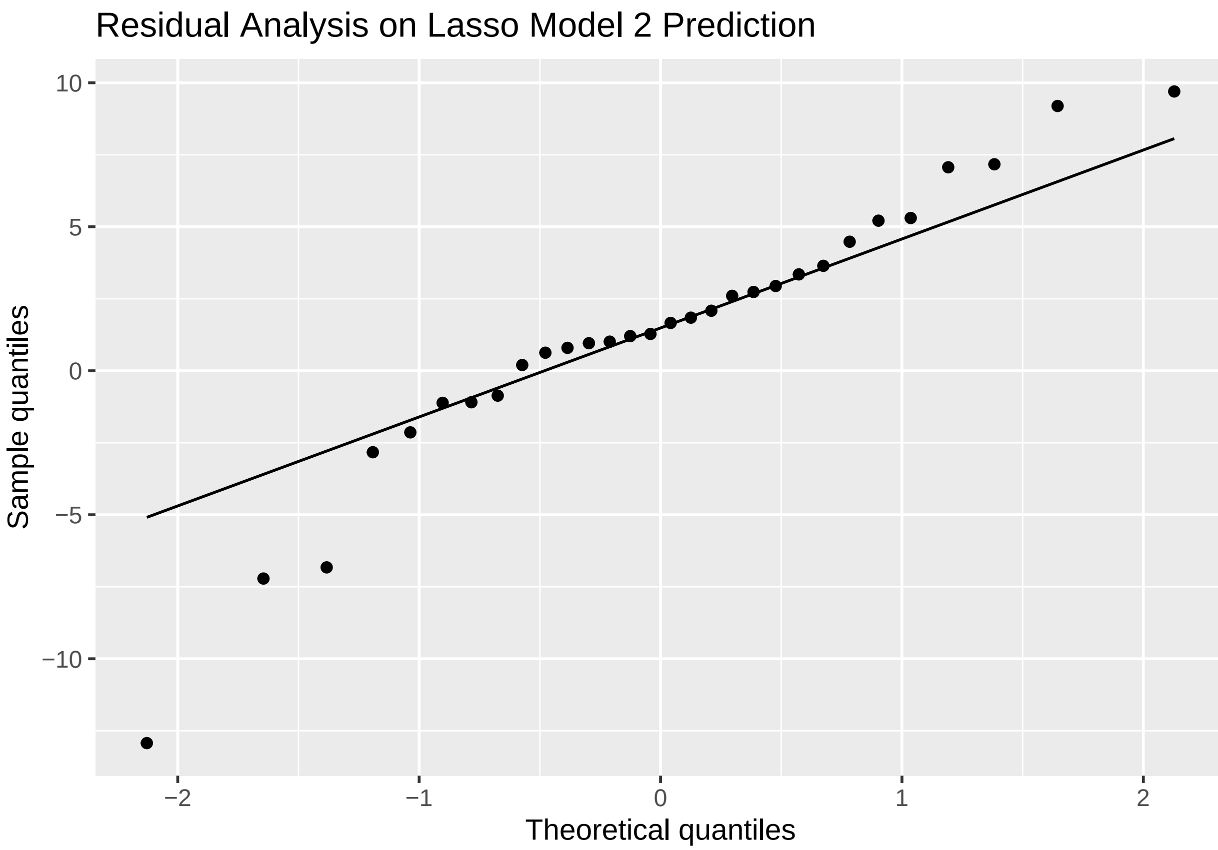

```
check_residual_hetero(residuals_future_ridge$residuals, residuals_future_ridge$predictions, "Residual An
```



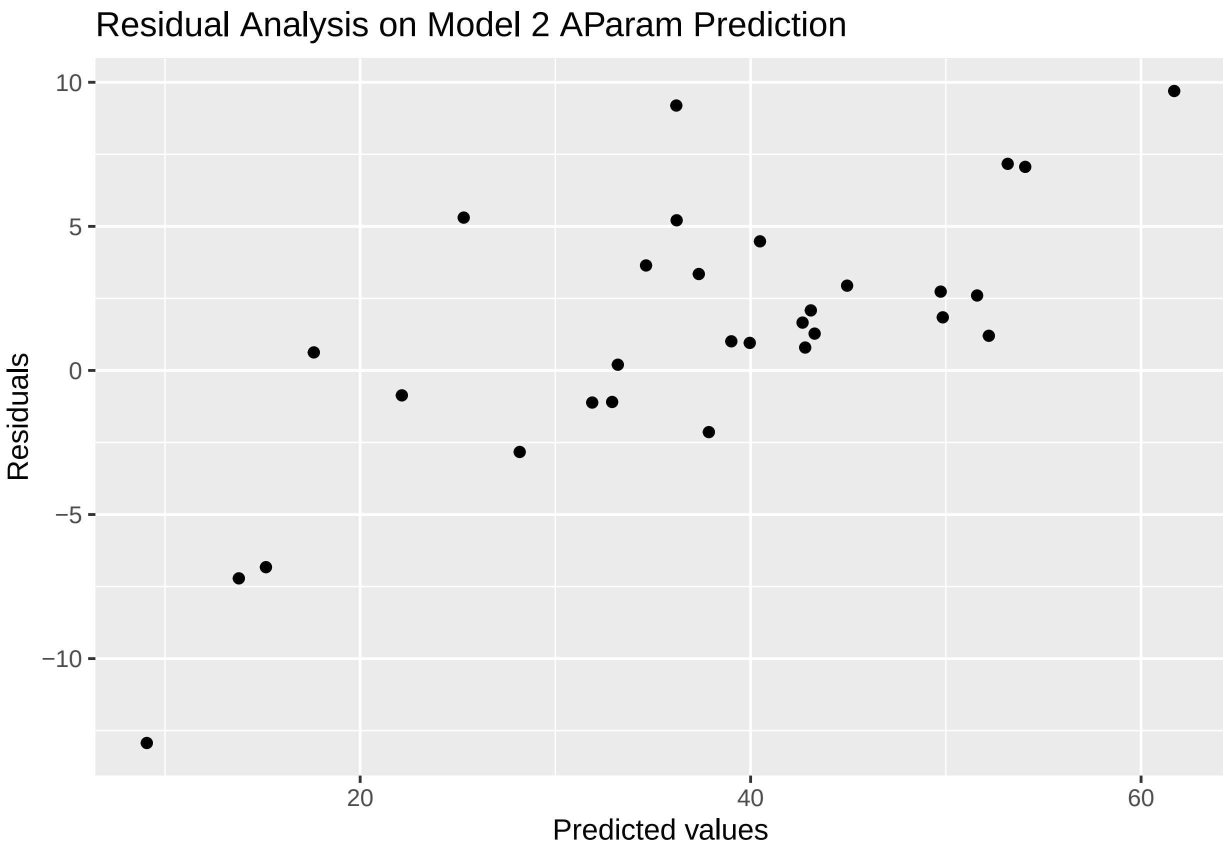

```
plot_residuals(residuals_future_lasso_min$residuals, residuals_future_lasso_min$predictions, advanced_st
```



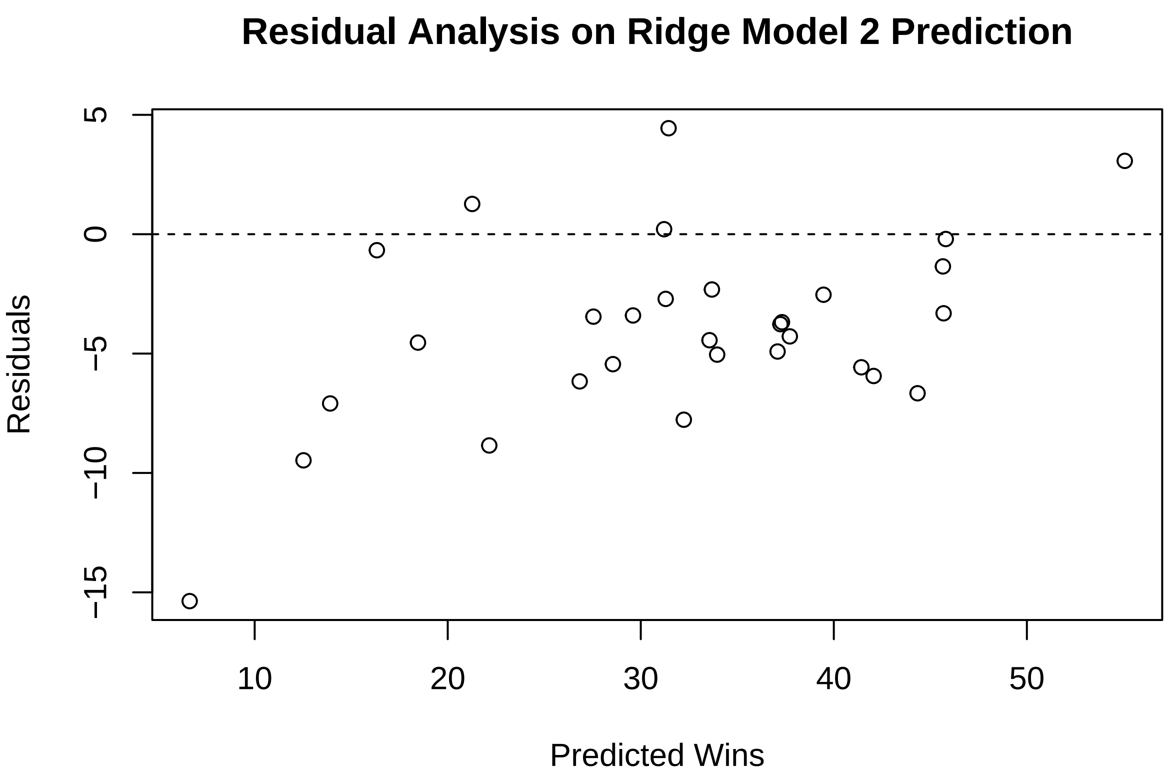
```
check_residual_normality(residuals_future_lasso_min$residuals, residuals_future_lasso_min$predictions,
```



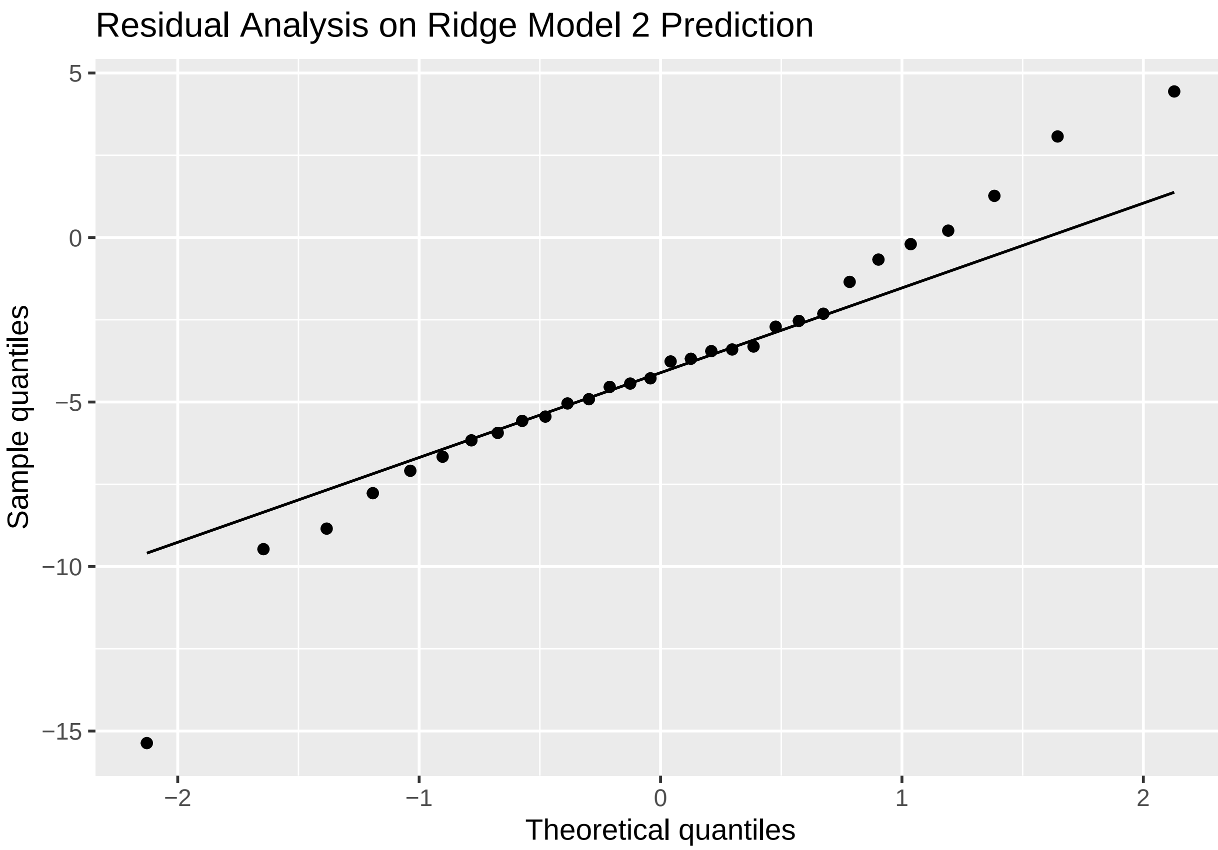

```
check_residual_hetero(residuals_future_lasso_min$residuals, residuals_future_lasso_min$predictions, "Re
```



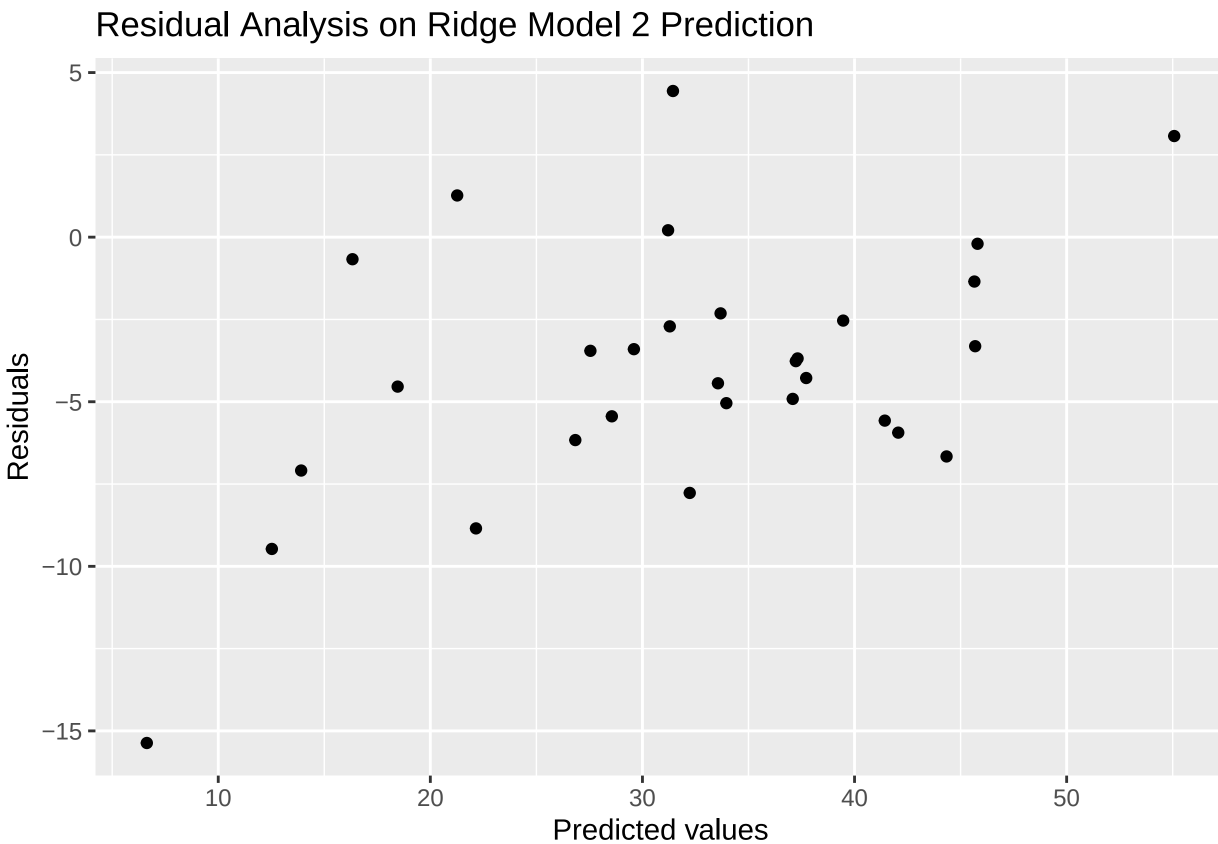

```
plot_residuals(residuals_future_ridge_min$residuals, residuals_future_ridge_min$predictions, advanced_st
```



```r
check_residual_normality(residuals_future_ridge_min$residuals, residuals_future_ridge_min$predictions,
```




```r
check_residual_hetero(residuals_future_ridge_min$residuals, residuals_future_ridge_min$predictions, "Re
```


Residual Analysis on Ridge Model 2 Prediction
5
0
−5
−10
−15
Residuals
10
20
30
40
50
Predicted values