(a) Data Gathering and Integration

```
library(dplyr)
```

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

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

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

```
Stats <- read.csv("C:/Users/charan/OneDrive/Desktop/dump/Seasons_Stats.csv", header=FALSE)
salary_1718 <- read.csv("C:/Users/charan/OneDrive/Desktop/reg-project/NBA_season1718_salary.csv", header=FALSE)
```

```
# rm(per_game_stats)
```

Here I am using two datasets, one is the stats dataset and other is the Salary dataset.

Stats:

This dataset contains detailed performance metrics for NBA players from the year 1950 to 2017. It includes various statistics, such as points, assists, rebounds, and more, which are crucial for building features.

Salary_1718:

This includes the salaries of NBA players for the 2017-2018 season. This serves as the target variable for the regression model, providing the actual salaries that we aim to predict.

Filtering and Merging

Since we are trying to predict the salaries of the players for the 17-18 Season on basis of their performance in 16-17 season, We do not require all data present in Stats dataset,hence we will filter it out in accordance to our objective.

First Let's rename the columns:

```
colnames(Stats) <- c("Sl.no","Year",    "Player",   "Pos",  "Age",  "Tm",   "G",    "GS",   "MP",   "PER",  "TS%"
,  "3PAr", "FTr",  "ORB%", "DRB%", "TRB%", "AST%", "STL%", "BLK%", "TOV%", "USG%", "blank1",   "OWS",  "DWS",  "W
S",   "WS/48",    "blank2",   "OBPM", "DBPM", "BPM",  "VORP", "FG",   "FGA",  "FG%",  "3P",   "3PA",  "3P%",  "2P
",   "2PA",  "2P%",  "eFG%", "FT",   "FTA",  "FT%",  "ORB",  "DRB",  "TRB",  "AST",  "STL",  "BLK",  "TOV",  "PF"
,   "PTS")

Stats <- Stats[-1, ]
head(Stats)
```

```
##   Sl.no Year          Player Pos Age  Tm  G GS MP PER   TS% 3PAr   FTr ORB%
## 2     0 1950 Curly Armstrong G-F  31 FTW 63           0.368      0.467
## 3     1 1950    Cliff Barker  SG  29 INO 49           0.435      0.387
## 4     2 1950   Leo Barnhorst  SF  25 CHS 67           0.394      0.259
## 5     3 1950      Ed Bartels   F  24 TOT 15           0.312      0.395
## 6     4 1950      Ed Bartels   F  24 DNN 13           0.308      0.378
## 7     5 1950      Ed Bartels   F  24 NYK  2           0.376       0.75
##   DRB% TRB% AST% STL% BLK% TOV% USG% blank1  OWS  DWS   WS WS/48 blank2 OBPM
## 2                                          -0.1  3.6  3.5
## 3                                           1.6  0.6  2.2
## 4                                           0.9  2.8  3.6
## 5                                          -0.5 -0.1 -0.6
## 6                                          -0.5 -0.1 -0.6
## 7                                             0    0    0
##   DBPM BPM VORP  FG FGA   FG% 3P 3PA 3P%  2P 2PA   2P%  eFG%  FT FTA   FT% ORB
## 2               144 516 0.279            144 516 0.279 0.279 170 241 0.705
## 3               102 274 0.372            102 274 0.372 0.372  75 106 0.708
## 4               174 499 0.349            174 499 0.349 0.349  90 129 0.698
## 5                22  86 0.256             22  86 0.256 0.256  19  34 0.559
## 6                21  82 0.256             21  82 0.256 0.256  17  31 0.548
## 7                 1   4  0.25              1   4  0.25  0.25   2   3 0.667
##   DRB TRB AST STL BLK TOV  PF PTS
## 2         176             217 458
## 3         109              99 279
## 4         140             192 438
## 5          20              29  63
## 6          20              27  59
## 7           0               2   4
```

Let's convert the columns into their best suited forms:

```
library(dplyr)
Stats <- Stats %>%
  mutate(across(everything(), ~type.convert(as.character(.), as.is = TRUE)))
Stats <- Stats %>%
  mutate(
    Player = as.character(Player),
    Pos = as.factor(Pos),
    Tm = as.factor(Tm),
    G = as.integer(G),
    GS = as.integer(GS),
    MP = as.integer(MP),
    FG = as.integer(FG),
    FGA = as.integer(FGA),
    `3P` = as.numeric(`3P`),
    `3PA` = as.numeric(`3PA`),
    `2P` = as.integer(`2P`),
    `2PA` = as.integer(`2PA`),
    FT = as.numeric(FT),
    FTA = as.numeric(FTA),
    ORB = as.integer(ORB),
    DRB = as.integer(DRB),
    TRB = as.integer(TRB),
    AST = as.integer(AST),
    STL = as.integer(STL),
    BLK = as.integer(BLK),
    TOV = as.integer(TOV),
    PF = as.integer(PF),
    PTS = as.integer(PTS)
  )
str(Stats)
```

```
## 'data.frame':    24691 obs. of  53 variables:
##  $ Sl.no : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ Year  : int  1950 1950 1950 1950 1950 1950 1950 1950 1950 1950 ...
##  $ Player: chr  "Curly Armstrong" "Cliff Barker" "Leo Barnhorst" "Ed Bartels" ...
##  $ Pos   : Factor w/ 24 levels "","C","C-F","C-PF",..: 10 21 17 6 6 6 9 10 7 7 ...
##  $ Age   : int  31 29 25 24 24 24 22 23 28 28 ...
##  $ Tm    : Factor w/ 70 levels "","AND","ATL",..: 23 27 15 63 22 43 27 64 63 23 ...
##  $ G     : int  63 49 67 15 13 2 60 3 65 36 ...
##  $ GS    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ MP    : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ PER   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ TS%   : num  0.368 0.435 0.394 0.312 0.308 0.376 0.422 0.275 0.346 0.362 ...
##  $ 3PAr  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ FTr   : num  0.467 0.387 0.259 0.395 0.378 0.75 0.301 0.313 0.395 0.48 ...
##  $ ORB%  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ DRB%  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ TRB%  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ AST%  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ STL%  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ BLK%  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ TOV%  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ USG%  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ blank1: logi  NA NA NA NA NA NA ...
##  $ OWS   : num  -0.1 1.6 0.9 -0.5 -0.5 0 3.6 -0.1 -2.2 -0.7 ...
##  $ DWS   : num  3.6 0.6 2.8 -0.1 -0.1 0 1.2 0 5 2.2 ...
##  $ WS    : num  3.5 2.2 3.6 -0.6 -0.6 0 4.8 -0.1 2.8 1.5 ...
##  $ WS/48 : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ blank2: logi  NA NA NA NA NA NA ...
##  $ OBPM  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ DBPM  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ BPM   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ VORP  : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ FG    : int  144 102 174 22 21 1 340 5 226 125 ...
##  $ FGA   : int  516 274 499 86 82 4 936 16 813 435 ...
##  $ FG%   : num  0.279 0.372 0.349 0.256 0.256 0.25 0.363 0.313 0.278 0.287 ...
##  $ 3P    : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ 3PA   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ 3P%   : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ 2P    : int  144 102 174 22 21 1 340 5 226 125 ...
##  $ 2PA   : int  516 274 499 86 82 4 936 16 813 435 ...
##  $ 2P%   : num  0.279 0.372 0.349 0.256 0.256 0.25 0.363 0.313 0.278 0.287 ...
##  $ eFG%  : num  0.279 0.372 0.349 0.256 0.256 0.25 0.363 0.313 0.278 0.287 ...
##  $ FT    : num  170 75 90 19 17 2 215 0 209 132 ...
##  $ FTA   : num  241 106 129 34 31 3 282 5 321 209 ...
##  $ FT%   : num  0.705 0.708 0.698 0.559 0.548 0.667 0.762 0 0.651 0.632 ...
##  $ ORB   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ DRB   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ TRB   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ AST   : int  176 109 140 20 20 0 233 2 163 75 ...
##  $ STL   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ BLK   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ TOV   : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ PF    : int  217 99 192 29 27 2 132 6 273 140 ...
##  $ PTS   : int  458 279 438 63 59 4 895 10 661 382 ...
```

Let's rename and remove unwanted columns in Salary dataset:

```
colnames(salary_1718) <- c("Rank","Player","Team", "Salary17_18")
salary_1718 <- salary_1718[-1, ]
salary_1718 <- salary_1718[ ,-1]
head(salary_1718)
```

```
##           Player Team Salary17_18
## 2  Stephen Curry  GSW    34682550
## 3   LeBron James  CLE    33285709
## 4   Paul Millsap  DEN    31269231
## 5 Gordon Hayward  BOS    29727900
## 6  Blake Griffin  DET    29512900
## 7     Kyle Lowry  TOR    28703704
```

Converting salary column into numeric:

```
salary_1718$Salary17_18 <- as.numeric(salary_1718$Salary17_18)
```

Now let's filter out the stats data and fetch stats of the players for 16-17 season:

```
stats17 <- Stats %>%
  filter(Year >= 2017) %>%
  select(Year:G, MP, PER, FG:PTS) %>%
  distinct(Player, .keep_all = TRUE)
head(stats17)
```

```
##   Year        Player Pos Age  Tm  G   MP  PER  FG FGA   FG% 3P 3PA   3P%  2P
## 1 2017  Alex Abrines  SG  23 OKC 68 1055 10.1 134 341 0.393 94 247 0.381  40
## 2 2017    Quincy Acy  PF  26 TOT 38  558 11.8  70 170 0.412 37  90 0.411  33
## 3 2017  Steven Adams   C  23 OKC 80 2389 16.5 374 655 0.571  0   1 0.000 374
## 4 2017 Arron Afflalo  SG  31 SAC 61 1580  9.0 185 420 0.440 62 151 0.411 123
## 5 2017 Alexis Ajinca   C  28 NOP 39  584 12.9  89 178 0.500  0   4 0.000  89
## 6 2017  Cole Aldrich   C  28 MIN 62  531 12.7  45  86 0.523  0   0    NA  45
##   2PA   2P%  eFG%  FT FTA   FT% ORB DRB TRB AST STL BLK TOV  PF PTS
## 1  94 0.426 0.531  44  49 0.898  18  68  86  40  37   8  33 114 406
## 2  80 0.413 0.521  45  60 0.750  20  95 115  18  14  15  21  67 222
## 3 654 0.572 0.571 157 257 0.611 282 333 615  86  88  78 146 195 905
## 4 269 0.457 0.514  83  93 0.892   9 116 125  78  21   7  42 104 515
## 5 174 0.511 0.500  29  40 0.725  46 131 177  12  20  22  31  77 207
## 6  86 0.523 0.523  15  22 0.682  51 107 158  25  25  23  17  85 105
```

Merging The two data sets

```
stats_and_salary <- merge(stats17, salary_1718, by.x = "Player", by.y = "Player")
summary(stats_and_salary)
```

```
##     Player               Year           Pos          Age              Tm
##  Length:442         Min.   :2017   SG     :97   Min.   :19.00   TOT    : 53
##  Class :character   1st Qu.:2017   PG     :92   1st Qu.:23.00   PHO    : 17
##  Mode  :character   Median :2017   C      :89   Median :26.00   BOS    : 16
##                     Mean   :2017   SF     :85   Mean   :26.19   CHI    : 16
##                     3rd Qu.:2017   PF     :78   3rd Qu.:29.00   MIL    : 16
##                     Max.   :2017   PF-C   : 1   Max.   :40.00   UTA    : 16
##                                    (Other): 0                   (Other):308
##        G               MP             PER              FG
##  Min.   : 2.00   Min.   :   9.0   Min.   :-2.10   Min.   :  1.0
##  1st Qu.:42.25   1st Qu.: 625.2   1st Qu.:10.20   1st Qu.: 78.5
##  Median :66.00   Median :1369.5   Median :13.00   Median :185.0
##  Mean   :57.91   Mean   :1352.3   Mean   :13.64   Mean   :219.5
##  3rd Qu.:76.00   3rd Qu.:2022.5   3rd Qu.:16.30   3rd Qu.:318.5
##  Max.   :82.00   Max.   :3048.0   Max.   :30.80   Max.   :824.0
##
##       FGA              FG%               3P              3PA
##  Min.   :   3.0   Min.   :0.1000   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 181.5   1st Qu.:0.4052   1st Qu.:  4.00   1st Qu.: 18.25
##  Median : 416.5   Median :0.4440   Median : 40.50   Median :117.50
##  Mean   : 480.7   Mean   :0.4507   Mean   : 54.98   Mean   :153.21
##  3rd Qu.: 684.0   3rd Qu.:0.4870   3rd Qu.: 90.00   3rd Qu.:252.75
##  Max.   :1941.0   Max.   :0.7500   Max.   :324.00   Max.   :789.00
##
##       3P%               2P             2PA              2P%
##  Min.   :0.0000   Min.   :  0.0   Min.   :   2.0   Min.   :0.000
##  1st Qu.:0.2845   1st Qu.: 55.0   1st Qu.: 110.2   1st Qu.:0.454
##  Median :0.3415   Median :126.5   Median : 254.5   Median :0.491
##  Mean   :0.3127   Mean   :164.5   Mean   : 327.5   Mean   :0.492
##  3rd Qu.:0.3780   3rd Qu.:227.8   3rd Qu.: 454.8   3rd Qu.:0.536
##  Max.   :1.0000   Max.   :730.0   Max.   :1421.0   Max.   :1.000
##  NA's   :28
##       eFG%              FT              FTA              FT%
##  Min.   :0.1000   Min.   :  0.00   Min.   :  0.00   Min.   :0.0000
##  1st Qu.:0.4750   1st Qu.: 29.25   1st Qu.: 39.25   1st Qu.:0.6880
##  Median :0.5070   Median : 67.00   Median : 91.00   Median :0.7700
##  Mean   :0.5051   Mean   :100.50   Mean   :129.71   Mean   :0.7472
##  3rd Qu.:0.5370   3rd Qu.:131.75   3rd Qu.:170.25   3rd Qu.:0.8367
##  Max.   :0.8750   Max.   :746.00   Max.   :881.00   Max.   :1.0000
##                                                     NA's   :4
##       ORB              DRB              TRB              AST
##  Min.   :  0.00   Min.   :  0.00   Min.   :   0.00   Min.   :  0.0
##  1st Qu.: 16.00   1st Qu.: 78.25   1st Qu.:  99.25   1st Qu.: 37.0
##  Median : 35.00   Median :164.00   Median : 205.50   Median : 82.0
##  Mean   : 55.46   Mean   :186.32   Mean   : 241.77   Mean   :127.3
##  3rd Qu.: 75.75   3rd Qu.:252.75   3rd Qu.: 328.25   3rd Qu.:165.5
##  Max.   :345.00   Max.   :817.00   Max.   :1116.00   Max.   :906.0
##
##       STL              BLK              TOV              PF
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.0
##  1st Qu.: 18.25   1st Qu.:  7.00   1st Qu.: 31.25   1st Qu.: 61.0
##  Median : 39.00   Median : 17.00   Median : 61.00   Median :114.0
##  Mean   : 43.28   Mean   : 26.06   Mean   : 75.18   Mean   :110.7
##  3rd Qu.: 60.00   3rd Qu.: 33.75   3rd Qu.:106.00   3rd Qu.:153.8
##  Max.   :157.00   Max.   :214.00   Max.   :464.00   Max.   :278.0
##
##       PTS             Team            Salary17_18
##  Min.   :   3.0   Length:442         Min.   :   17224
##  1st Qu.: 209.2   Class :character   1st Qu.: 1518316
##  Median : 497.0   Mode  :character   Median : 3519283
##  Mean   : 594.5                      Mean   : 7106258
##  3rd Qu.: 841.2                      3rd Qu.:10812701
##  Max.   :2558.0                      Max.   :34682550
##
```

From from the summary, after merging we can see that there are few Na values to be handled.

(c) Cleaning

```
head(stats_and_salary)
```

```
##            Player Year Pos Age  Tm  G   MP  PER  FG FGA   FG% 3P 3PA   3P%  2P
## 1    A.J. Hammons 2017   C  24 DAL 22  163  8.4  17  42 0.405  5  10 0.500  12
## 2    Aaron Brooks 2017  PG  32 IND 65  894  9.5 121 300 0.403 48 128 0.375  73
## 3    Aaron Gordon 2017  SF  21 ORL 80 2298 14.4 393 865 0.454 77 267 0.288 316
## 4 Al-Farouq Aminu 2017  SF  26 POR 61 1773 11.3 183 466 0.393 70 212 0.330 113
## 5      Al Horford 2017   C  30 BOS 68 2193 17.7 379 801 0.473 86 242 0.355 293
## 6    Al Jefferson 2017   C  32 IND 66  931 18.9 235 471 0.499  0   1 0.000 235
##   2PA   2P%  eFG%  FT FTA   FT% ORB DRB TRB AST STL BLK TOV  PF  PTS Team
## 1  32 0.375 0.464   9  20 0.450   8  28  36   4   1  13  10  21   48  MIA
## 2 172 0.424 0.483  32  40 0.800  18  51  69 125  25   9  66  93  322  MIN
## 3 598 0.528 0.499 156 217 0.719 116 289 405 150  64  40  89 172 1019  ORL
## 4 254 0.445 0.468  96 136 0.706  77 374 451  99  60  44  94 102  532  POR
## 5 559 0.524 0.527 108 135 0.800  95 369 464 337  52  87 116 138  952  BOS
## 6 470 0.500 0.499  65  85 0.765  75 203 278  57  19  16  33 125  535  IND
##   Salary17_18
## 1     1312611
## 2     2116955
## 3     5504420
## 4     7319035
## 5    27734405
## 6     9769821
```

Here we can remove the Team column from the stats_and_salary data as it is not required for analysis. This column indicates the team to which the player belonged in 17-18 season, Which is not required for us.

```
library(dplyr)
stats_and_salary <- stats_and_salary %>% select(-Team)
```

Now let's check the summary and also check if there are any Na values in our merged dataset.

```
summary(stats_and_salary)
```

```
##     Player               Year           Pos          Age              Tm
##  Length:442         Min.   :2017   SG     :97   Min.   :19.00   TOT    : 53
##  Class :character   1st Qu.:2017   PG     :92   1st Qu.:23.00   PHO    : 17
##  Mode  :character   Median :2017   C      :89   Median :26.00   BOS    : 16
##                     Mean   :2017   SF     :85   Mean   :26.19   CHI    : 16
##                     3rd Qu.:2017   PF     :78   3rd Qu.:29.00   MIL    : 16
##                     Max.   :2017   PF-C   : 1   Max.   :40.00   UTA    : 16
##                                    (Other): 0                   (Other):308
##        G               MP             PER              FG
##  Min.   : 2.00   Min.   :   9.0   Min.   :-2.10   Min.   :  1.0
##  1st Qu.:42.25   1st Qu.: 625.2   1st Qu.:10.20   1st Qu.: 78.5
##  Median :66.00   Median :1369.5   Median :13.00   Median :185.0
##  Mean   :57.91   Mean   :1352.3   Mean   :13.64   Mean   :219.5
##  3rd Qu.:76.00   3rd Qu.:2022.5   3rd Qu.:16.30   3rd Qu.:318.5
##  Max.   :82.00   Max.   :3048.0   Max.   :30.80   Max.   :824.0
##
##       FGA              FG%               3P              3PA
##  Min.   :   3.0   Min.   :0.1000   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 181.5   1st Qu.:0.4052   1st Qu.:  4.00   1st Qu.: 18.25
##  Median : 416.5   Median :0.4440   Median : 40.50   Median :117.50
##  Mean   : 480.7   Mean   :0.4507   Mean   : 54.98   Mean   :153.21
##  3rd Qu.: 684.0   3rd Qu.:0.4870   3rd Qu.: 90.00   3rd Qu.:252.75
##  Max.   :1941.0   Max.   :0.7500   Max.   :324.00   Max.   :789.00
##
##       3P%               2P             2PA              2P%
##  Min.   :0.0000   Min.   :  0.0   Min.   :   2.0   Min.   :0.000
##  1st Qu.:0.2845   1st Qu.: 55.0   1st Qu.: 110.2   1st Qu.:0.454
##  Median :0.3415   Median :126.5   Median : 254.5   Median :0.491
##  Mean   :0.3127   Mean   :164.5   Mean   : 327.5   Mean   :0.492
##  3rd Qu.:0.3780   3rd Qu.:227.8   3rd Qu.: 454.8   3rd Qu.:0.536
##  Max.   :1.0000   Max.   :730.0   Max.   :1421.0   Max.   :1.000
##  NA's   :28
##       eFG%              FT              FTA              FT%
##  Min.   :0.1000   Min.   :  0.00   Min.   :  0.00   Min.   :0.0000
##  1st Qu.:0.4750   1st Qu.: 29.25   1st Qu.: 39.25   1st Qu.:0.6880
##  Median :0.5070   Median : 67.00   Median : 91.00   Median :0.7700
##  Mean   :0.5051   Mean   :100.50   Mean   :129.71   Mean   :0.7472
##  3rd Qu.:0.5370   3rd Qu.:131.75   3rd Qu.:170.25   3rd Qu.:0.8367
##  Max.   :0.8750   Max.   :746.00   Max.   :881.00   Max.   :1.0000
##                                                     NA's   :4
##       ORB              DRB              TRB              AST
##  Min.   :  0.00   Min.   :  0.00   Min.   :   0.00   Min.   :  0.0
##  1st Qu.: 16.00   1st Qu.: 78.25   1st Qu.:  99.25   1st Qu.: 37.0
##  Median : 35.00   Median :164.00   Median : 205.50   Median : 82.0
##  Mean   : 55.46   Mean   :186.32   Mean   : 241.77   Mean   :127.3
##  3rd Qu.: 75.75   3rd Qu.:252.75   3rd Qu.: 328.25   3rd Qu.:165.5
##  Max.   :345.00   Max.   :817.00   Max.   :1116.00   Max.   :906.0
##
##       STL              BLK              TOV              PF
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.0
##  1st Qu.: 18.25   1st Qu.:  7.00   1st Qu.: 31.25   1st Qu.: 61.0
##  Median : 39.00   Median : 17.00   Median : 61.00   Median :114.0
##  Mean   : 43.28   Mean   : 26.06   Mean   : 75.18   Mean   :110.7
##  3rd Qu.: 60.00   3rd Qu.: 33.75   3rd Qu.:106.00   3rd Qu.:153.8
##  Max.   :157.00   Max.   :214.00   Max.   :464.00   Max.   :278.0
##
##       PTS          Salary17_18
##  Min.   :   3.0   Min.   :   17224
##  1st Qu.: 209.2   1st Qu.: 1518316
##  Median : 497.0   Median : 3519283
##  Mean   : 594.5   Mean   : 7106258
##  3rd Qu.: 841.2   3rd Qu.:10812701
##  Max.   :2558.0   Max.   :34682550
##
```

```
sum(is.na(stats_and_salary))
```

```
## [1] 32
```

our Merged data has 32 Na values in total to be handled. We can replace these Na values by the means of their columns.I chose this approach because, since it is a sports data containing stats of each player, we cannot omit an entire player's data because of just few missing values.

```
numeric_cols <- sapply(stats_and_salary, is.numeric)
stats_and_salary[, numeric_cols] <-
  lapply(stats_and_salary[, numeric_cols],
         function(col) ifelse(is.na(col), mean(col, na.rm = TRUE), col))
sum(is.na(stats_and_salary))
```

```
## [1] 0
```

```
summary(stats_and_salary)
```

```
##     Player               Year           Pos          Age              Tm
##  Length:442         Min.   :2017   SG     :97   Min.   :19.00   TOT    : 53
##  Class :character   1st Qu.:2017   PG     :92   1st Qu.:23.00   PHO    : 17
##  Mode  :character   Median :2017   C      :89   Median :26.00   BOS    : 16
##                     Mean   :2017   SF     :85   Mean   :26.19   CHI    : 16
##                     3rd Qu.:2017   PF     :78   3rd Qu.:29.00   MIL    : 16
##                     Max.   :2017   PF-C   : 1   Max.   :40.00   UTA    : 16
##                                    (Other): 0                   (Other):308
##        G               MP             PER              FG
##  Min.   : 2.00   Min.   :   9.0   Min.   :-2.10   Min.   :  1.0
##  1st Qu.:42.25   1st Qu.: 625.2   1st Qu.:10.20   1st Qu.: 78.5
##  Median :66.00   Median :1369.5   Median :13.00   Median :185.0
##  Mean   :57.91   Mean   :1352.3   Mean   :13.64   Mean   :219.5
##  3rd Qu.:76.00   3rd Qu.:2022.5   3rd Qu.:16.30   3rd Qu.:318.5
##  Max.   :82.00   Max.   :3048.0   Max.   :30.80   Max.   :824.0
##
##       FGA              FG%               3P              3PA
##  Min.   :   3.0   Min.   :0.1000   Min.   :  0.00   Min.   :  0.00
##  1st Qu.: 181.5   1st Qu.:0.4052   1st Qu.:  4.00   1st Qu.: 18.25
##  Median : 416.5   Median :0.4440   Median : 40.50   Median :117.50
##  Mean   : 480.7   Mean   :0.4507   Mean   : 54.98   Mean   :153.21
##  3rd Qu.: 684.0   3rd Qu.:0.4870   3rd Qu.: 90.00   3rd Qu.:252.75
##  Max.   :1941.0   Max.   :0.7500   Max.   :324.00   Max.   :789.00
##
##       3P%               2P             2PA              2P%
##  Min.   :0.0000   Min.   :  0.0   Min.   :   2.0   Min.   :0.000
##  1st Qu.:0.2900   1st Qu.: 55.0   1st Qu.: 110.2   1st Qu.:0.454
##  Median :0.3360   Median :126.5   Median : 254.5   Median :0.491
##  Mean   :0.3127   Mean   :164.5   Mean   : 327.5   Mean   :0.492
##  3rd Qu.:0.3750   3rd Qu.:227.8   3rd Qu.: 454.8   3rd Qu.:0.536
##  Max.   :1.0000   Max.   :730.0   Max.   :1421.0   Max.   :1.000
##
##       eFG%              FT              FTA              FT%
##  Min.   :0.1000   Min.   :  0.00   Min.   :  0.00   Min.   :0.0000
##  1st Qu.:0.4750   1st Qu.: 29.25   1st Qu.: 39.25   1st Qu.:0.6880
##  Median :0.5070   Median : 67.00   Median : 91.00   Median :0.7695
##  Mean   :0.5051   Mean   :100.50   Mean   :129.71   Mean   :0.7472
##  3rd Qu.:0.5370   3rd Qu.:131.75   3rd Qu.:170.25   3rd Qu.:0.8353
##  Max.   :0.8750   Max.   :746.00   Max.   :881.00   Max.   :1.0000
##
##       ORB              DRB              TRB              AST
##  Min.   :  0.00   Min.   :  0.00   Min.   :   0.00   Min.   :  0.0
##  1st Qu.: 16.00   1st Qu.: 78.25   1st Qu.:  99.25   1st Qu.: 37.0
##  Median : 35.00   Median :164.00   Median : 205.50   Median : 82.0
##  Mean   : 55.46   Mean   :186.32   Mean   : 241.77   Mean   :127.3
##  3rd Qu.: 75.75   3rd Qu.:252.75   3rd Qu.: 328.25   3rd Qu.:165.5
##  Max.   :345.00   Max.   :817.00   Max.   :1116.00   Max.   :906.0
##
##       STL              BLK              TOV              PF
##  Min.   :  0.00   Min.   :  0.00   Min.   :  0.00   Min.   :  0.0
##  1st Qu.: 18.25   1st Qu.:  7.00   1st Qu.: 31.25   1st Qu.: 61.0
##  Median : 39.00   Median : 17.00   Median : 61.00   Median :114.0
##  Mean   : 43.28   Mean   : 26.06   Mean   : 75.18   Mean   :110.7
##  3rd Qu.: 60.00   3rd Qu.: 33.75   3rd Qu.:106.00   3rd Qu.:153.8
##  Max.   :157.00   Max.   :214.00   Max.   :464.00   Max.   :278.0
##
##       PTS          Salary17_18
##  Min.   :   3.0   Min.   :   17224
##  1st Qu.: 209.2   1st Qu.: 1518316
##  Median : 497.0   Median : 3519283
##  Mean   : 594.5   Mean   : 7106258
##  3rd Qu.: 841.2   3rd Qu.:10812701
##  Max.   :2558.0   Max.   :34682550
##
```

b.Data Exploration

Per-Game stats:

The original dataset includes whole-season statistics (e.g., total points, total assists) for each player. However, these cumulative totals can be misleading when comparing players who played different numbers of games.

Per-game statistics provide a more accurate representation of a player's average performance, making comparisons fairer.

Some players may not play the entire season due to injuries, resting, or other factors. As a result, their cumulative season totals could appear lower, even if they have strong performances when they are on the court.

By calculating per-game statistics (e.g., PPG - Points Per Game, MPG - Minutes Per Game), we normalize the data to reflect a player's contribution in each game they play, rather than across an entire season.

Creation of Per-Game Statistics

The dataset originally contained cumulative season totals (e.g., total points, total minutes played)

MPG (Minutes Per Game): MP/G

PPG (Points Per Game): PTS/G

APG (Assists Per Game): AST/G

RPG (Rebounds Per Game): TRB/G

TOPG (Turnovers Per Game): TOV/G

BPG (Blocks Per Game): BLK/G

SPG (Steals Per Game): STL/G

```
stats_and_salary <- stats_and_salary %>%
  mutate(MPG = MP / G, PPG = PTS / G, APG = AST / G,
         RPG = TRB / G, TOPG = TOV / G, BPG = BLK / G,
         SPG = STL / G)
head(stats_and_salary)
```

```
##            Player Year Pos Age  Tm  G   MP  PER  FG FGA   FG% 3P 3PA   3P%  2P
## 1    A.J. Hammons 2017   C  24 DAL 22  163  8.4  17  42 0.405  5  10 0.500  12
## 2    Aaron Brooks 2017  PG  32 IND 65  894  9.5 121 300 0.403 48 128 0.375  73
## 3    Aaron Gordon 2017  SF  21 ORL 80 2298 14.4 393 865 0.454 77 267 0.288 316
## 4 Al-Farouq Aminu 2017  SF  26 POR 61 1773 11.3 183 466 0.393 70 212 0.330 113
## 5      Al Horford 2017   C  30 BOS 68 2193 17.7 379 801 0.473 86 242 0.355 293
## 6    Al Jefferson 2017   C  32 IND 66  931 18.9 235 471 0.499  0   1 0.000 235
##   2PA   2P%  eFG%  FT FTA   FT% ORB DRB TRB AST STL BLK TOV  PF  PTS
## 1  32 0.375 0.464   9  20 0.450   8  28  36   4   1  13  10  21   48
## 2 172 0.424 0.483  32  40 0.800  18  51  69 125  25   9  66  93  322
## 3 598 0.528 0.499 156 217 0.719 116 289 405 150  64  40  89 172 1019
## 4 254 0.445 0.468  96 136 0.706  77 374 451  99  60  44  94 102  532
## 5 559 0.524 0.527 108 135 0.800  95 369 464 337  52  87 116 138  952
## 6 470 0.500 0.499  65  85 0.765  75 203 278  57  19  16  33 125  535
##   Salary17_18       MPG       PPG       APG      RPG      TOPG       BPG
## 1     1312611  7.409091  2.181818 0.1818182 1.636364 0.4545455 0.5909091
## 2     2116955 13.753846  4.953846 1.9230769 1.061538 1.0153846 0.1384615
## 3     5504420 28.725000 12.737500 1.8750000 5.062500 1.1125000 0.5000000
## 4     7319035 29.065574  8.721311 1.6229508 7.393443 1.5409836 0.7213115
## 5    27734405 32.250000 14.000000 4.9558824 6.823529 1.7058824 1.2794118
## 6     9769821 14.106061  8.106061 0.8636364 4.212121 0.5000000 0.2424242
##          SPG
## 1 0.04545455
## 2 0.38461538
## 3 0.80000000
## 4 0.98360656
## 5 0.76470588
## 6 0.28787879
```

Now we can only keep our per-game stats for analysis by removing the cumulative stats of the player for entire season.

```
per_game_stats <- stats_and_salary %>% select(Player, Tm, MPG, PPG, APG, RPG, TOPG, BPG, SPG,  Salary17_18)
head(per_game_stats)
```

```
##            Player  Tm       MPG       PPG       APG      RPG      TOPG
## 1    A.J. Hammons DAL  7.409091  2.181818 0.1818182 1.636364 0.4545455
## 2    Aaron Brooks IND 13.753846  4.953846 1.9230769 1.061538 1.0153846
## 3    Aaron Gordon ORL 28.725000 12.737500 1.8750000 5.062500 1.1125000
## 4 Al-Farouq Aminu POR 29.065574  8.721311 1.6229508 7.393443 1.5409836
## 5      Al Horford BOS 32.250000 14.000000 4.9558824 6.823529 1.7058824
## 6    Al Jefferson IND 14.106061  8.106061 0.8636364 4.212121 0.5000000
##         BPG        SPG Salary17_18
## 1 0.5909091 0.04545455     1312611
## 2 0.1384615 0.38461538     2116955
## 3 0.5000000 0.80000000     5504420
## 4 0.7213115 0.98360656     7319035
## 5 1.2794118 0.76470588    27734405
## 6 0.2424242 0.28787879     9769821
```

correlation-1

```
library(corrplot)
```

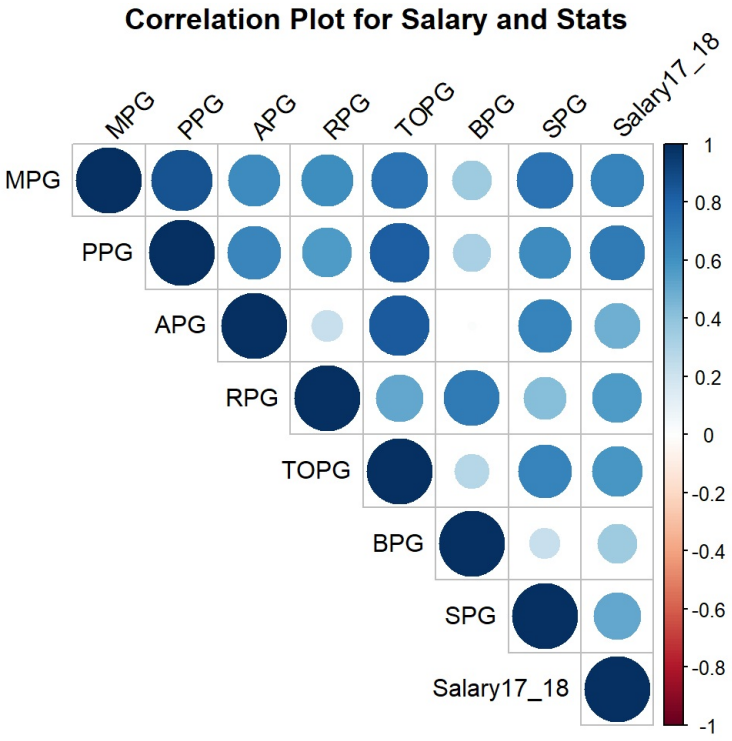
```
## corrplot 0.95 loaded
```

```
library(dplyr)

numeric_columns <- per_game_stats %>%
  select(MPG, PPG, APG, RPG, TOPG, BPG, SPG, Salary17_18)

cor_matrix <- cor(numeric_columns, use = "complete.obs")

corrplot(cor_matrix,
         method = "circle",
         type = "upper",
         tl.col = "black",
         tl.srt = 45,
         title = "Correlation Plot for Salary and Stats",
         mar = c(0, 0, 1, 0))
```



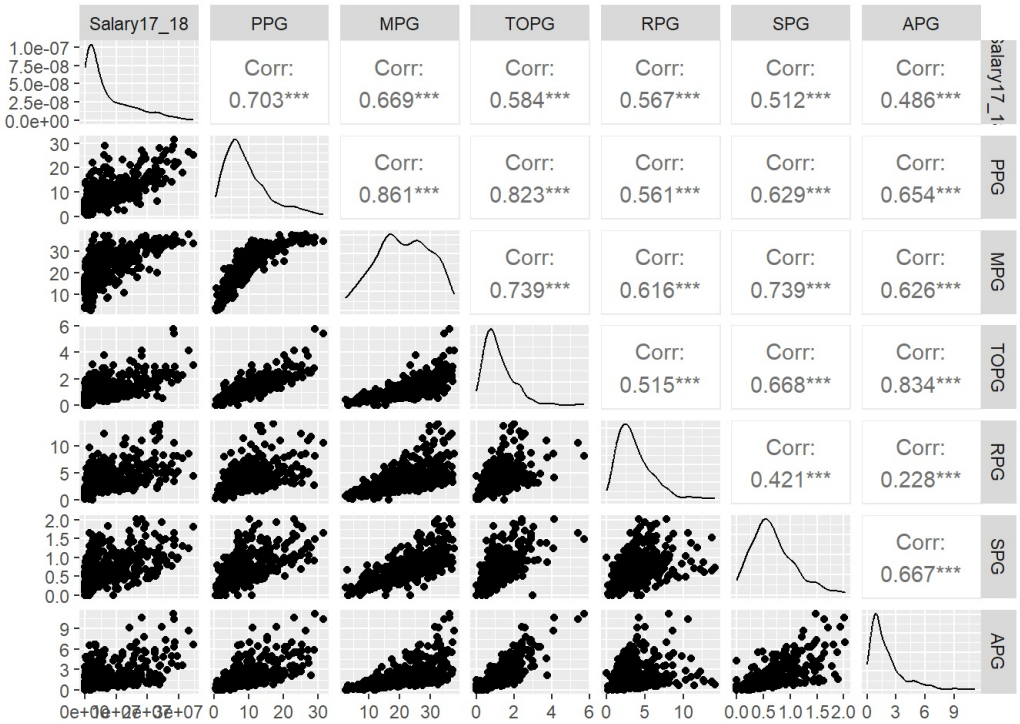
Correlation-2

```
library(dplyr)
library(GGally)
```

```
## Loading required package: ggplot2
```

```
## Registered S3 method overwritten by 'GGally':
##   method from
##   +.gg   ggplot2
```

```
stats_salary_cor <-
  per_game_stats %>%
  select(Salary17_18, PPG, MPG, TOPG, RPG, SPG, APG)
ggpairs(stats_salary_cor)
```



By observing these two plots:

The Strength of correlation between the variables is ranked as follows: PPG>MPG>TOPG>RPG>PER>SPG>APG

The number of turnovers (TOPG) made by players shows a positive correlation with their salary. We interpreted this as: “The more turnovers a player makes, the more involved they are in ball movements during the game.“

This could suggest that players who make turnovers might be more important to their team, possibly due to their aggressive playstyle.

We realize that this interpretation may not be fully accurate. For more precise analysis, we should consider additional data, such as how long each player holds the ball during games.

Let’s visualize a scatter plot between PPG and salary as they are highly correlated.

```
library(dplyr)
per_game_stats <- per_game_stats %>%
  rename(Team = Tm)
```

```
library(plotly)
```

```
##
## Attaching package: 'plotly'
```

```
## The following object is masked from 'package:ggplot2':
##
##     last_plot
```

```
## The following object is masked from 'package:stats':
##
##     filter
```

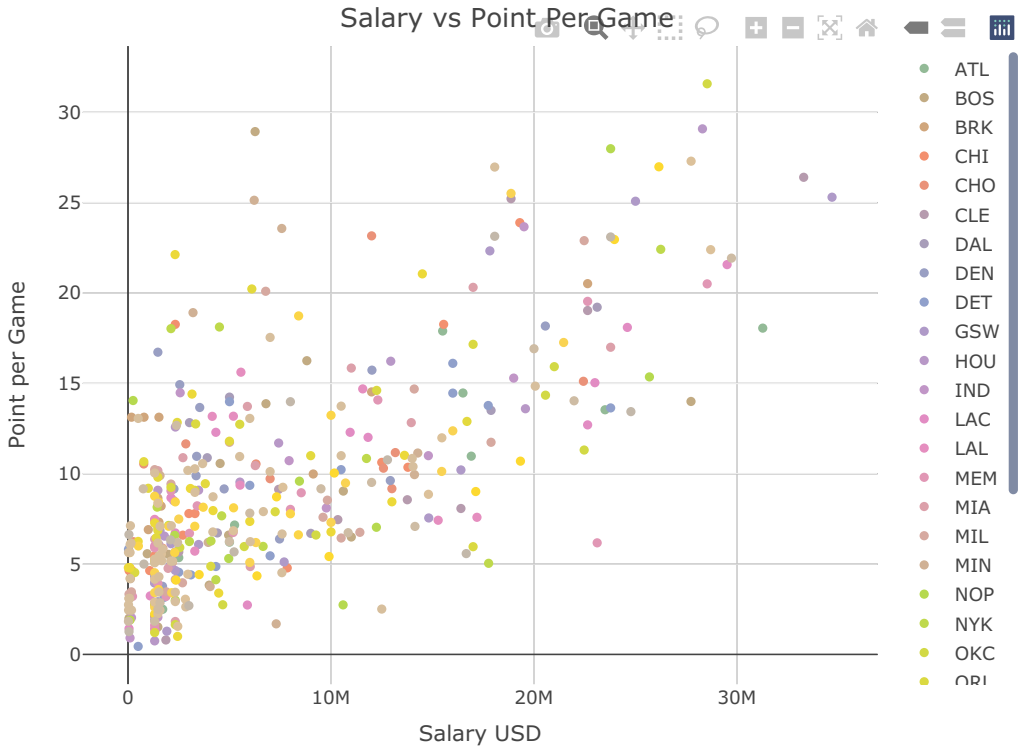
```
## The following object is masked from 'package:graphics':
##
##     layout
```

```
plot_ly(data = per_game_stats, x = ~Salary17_18, y = ~PPG, color = ~Team,
        hoverinfo = "text",
        text = ~paste("Player: ", Player,
                      "<br>Salary: ", format(Salary17_18, big.mark = ","),"$",
                      "<br>PPG: ", round(PPG, digits = 3),
                      "<br>Team: ", Team)) %>%
  layout(
    title = "Salary vs Point Per Game",
    xaxis = list(title = "Salary USD"),
    yaxis = list(title = "Point per Game")
  )
```

```
## No trace type specified:
##   Based on info supplied, a 'scatter' trace seems appropriate.
##   Read more about this trace type -> https://plotly.com/r/reference/#scatter
```

```
## No scatter mode specifed:
##   Setting the mode to markers
##   Read more about this attribute -> https://plotly.com/r/reference/#scatter-mode
```

```
## Warning in RColorBrewer::brewer.pal(N, "Set2"): n too large, allowed maximum for palette Set2 is 8
## Returning the palette you asked for with that many colors
## Warning in RColorBrewer::brewer.pal(N, "Set2"): n too large, allowed maximum for palette Set2 is 8
## Returning the palette you asked for with that many colors
```



If you hover on the points in this plot, you can check the player names. As you can tell the most paid player is Stephen Curry.

Linear regression:

```
library(ggplot2)
per_game_stats %>%
  ggplot(aes(x = Salary17_18, y = PPG)) +
  geom_point() +
  geom_smooth(method = "lm")
```

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

You can see a general upward trend, which means that as players score more points per game, their salaries tend to be higher.

The blue line is a trend line that shows this overall pattern, and the shaded area around it represents the range where most data points are likely to fall, showing confidence in this trend.

In simple terms, players who score more points usually earn higher salaries.

```
stats_salary_regression <-
  per_game_stats %>% select(Salary17_18, MPG:SPG)
lm(Salary17_18~., data=stats_salary_regression)
```

```
##
## Call:
## lm(formula = Salary17_18 ~ ., data = stats_salary_regression)
##
## Coefficients:
## (Intercept)          MPG          PPG          APG          RPG         TOPG
##    -2792909        30565       686815      1059087       916087     -2709447
##         BPG          SPG
##      470136       631255
```

Scoring points, assisting, and rebounding contribute the most to higher salaries. Turnovers (losing possession) have a large negative impact on salary.

Defensive stats like blocks and steals also contribute positively but to a lesser extent than scoring and playmaking skills.

d. Data Preprocessing

Earlier in Correlation analysis we saw that even turnovers have a bit of positive impact over a player's salary and we related it to a player's aggressive playstyle, if a player is involved more in ball movements he tends to commit turn overs, so here we can a create a variable called as Aggressiveness.

If a player's turnovers per game are equal to or above this average, they're labeled as "Yes" for Aggressiveness (meaning they turn over the ball relatively often).

Apart from this we can also observe that MPG (minutes per game) also have a positive relation with the salary. Players who generally trusted by their coaches tend to play more on court and tend to score more which boosts their salaries, Hence we can create another variable called Trusted.

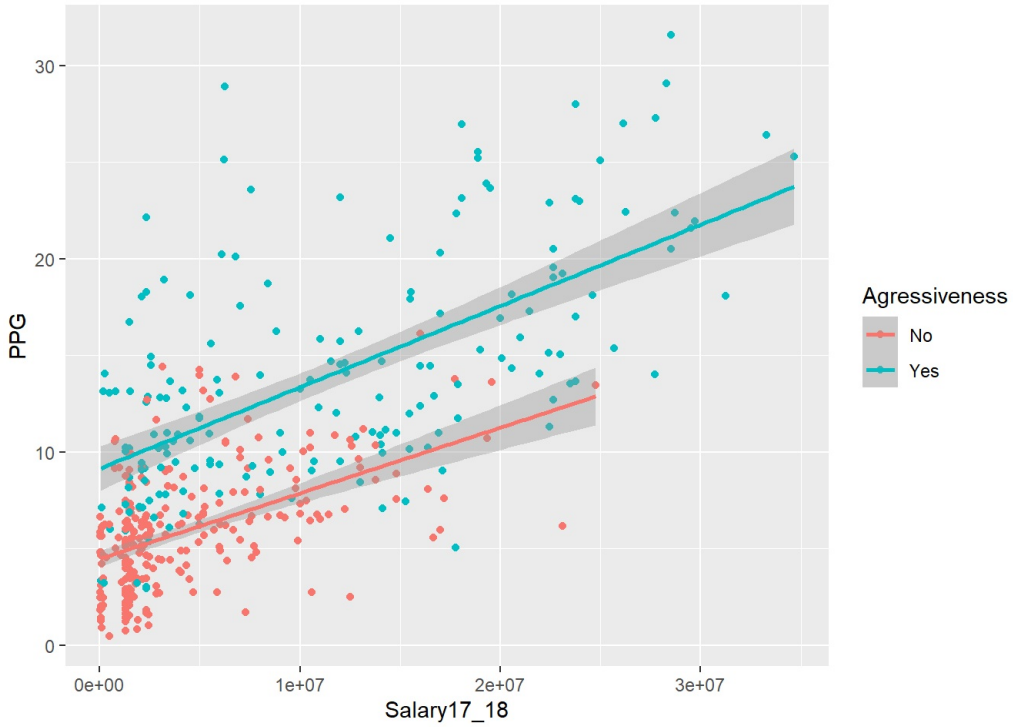
If a player's minutes per game are equal to or above this average, they're labeled as "Yes" for Trust (meaning the coach plays them often, indicating trust).

```
avg.minutes <- mean(per_game_stats$MPG)
avg.turnover <- mean(per_game_stats$TOPG)
per_game_stats$Trusted <- as.factor(ifelse(per_game_stats$MPG >= avg.minutes, "Yes", "No"))
per_game_stats$Agressiveness <- as.factor(ifelse(per_game_stats$TOPG >= avg.turnover, "Yes", "No"))
head(per_game_stats)
```

```
##            Player Team       MPG       PPG       APG      RPG      TOPG
## 1    A.J. Hammons  DAL  7.409091  2.181818 0.1818182 1.636364 0.4545455
## 2    Aaron Brooks  IND 13.753846  4.953846 1.9230769 1.061538 1.0153846
## 3    Aaron Gordon  ORL 28.725000 12.737500 1.8750000 5.062500 1.1125000
## 4 Al-Farouq Aminu  POR 29.065574  8.721311 1.6229508 7.393443 1.5409836
## 5      Al Horford  BOS 32.250000 14.000000 4.9558824 6.823529 1.7058824
## 6    Al Jefferson  IND 14.106061  8.106061 0.8636364 4.212121 0.5000000
##         BPG        SPG Salary17_18 Trusted Agressiveness
## 1 0.5909091 0.04545455     1312611      No            No
## 2 0.1384615 0.38461538     2116955      No            No
## 3 0.5000000 0.80000000     5504420     Yes            No
## 4 0.7213115 0.98360656     7319035     Yes           Yes
## 5 1.2794118 0.76470588    27734405     Yes           Yes
## 6 0.2424242 0.28787879     9769821      No            No
```

```
per_game_stats %>%
  ggplot(aes(x = Salary17_18, y = PPG, colour = Agressiveness)) +
  geom_point() +
  geom_smooth(method="lm")
```

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



The data points are color-coded by “Aggressiveness,” which is based on whether the player has an above-average or below-average number of turnovers (TOPG).

Here, players labeled “Yes” in teal have higher turnovers (more aggressive), while players labeled “No” in red have fewer turnovers. The trend lines for each group show the general relationship between salary and points per game for players with different levels of aggressiveness.

Positive Salary-Performance Trend: As salary increases, players generally score more points per game, which indicates that higher-paid players tend to perform better in terms of scoring.

Aggressiveness Impact: The teal line (for aggressive players) is above the red line, suggesting that players who are more aggressive (higher turnovers) tend to have slightly higher PPG at a given salary level compared to less aggressive players.

```
per_game_stats$TrustedYes <- ifelse(per_game_stats$Trusted == "Yes", 1, 0)
per_game_stats$AgressivenessYes <- ifelse(per_game_stats$Agressiveness == "Yes", 1, 0)

head(per_game_stats)
```

```
##            Player Team       MPG       PPG       APG      RPG      TOPG
## 1    A.J. Hammons  DAL  7.409091  2.181818 0.1818182 1.636364 0.4545455
## 2    Aaron Brooks  IND 13.753846  4.953846 1.9230769 1.061538 1.0153846
## 3    Aaron Gordon  ORL 28.725000 12.737500 1.8750000 5.062500 1.1125000
## 4 Al-Farouq Aminu  POR 29.065574  8.721311 1.6229508 7.393443 1.5409836
## 5      Al Horford  BOS 32.250000 14.000000 4.9558824 6.823529 1.7058824
## 6    Al Jefferson  IND 14.106061  8.106061 0.8636364 4.212121 0.5000000
##         BPG        SPG Salary17_18 Trusted Agressiveness TrustedYes
## 1 0.5909091 0.04545455     1312611      No            No          0
## 2 0.1384615 0.38461538     2116955      No            No          0
## 3 0.5000000 0.80000000     5504420     Yes            No          1
## 4 0.7213115 0.98360656     7319035     Yes           Yes          1
## 5 1.2794118 0.76470588    27734405     Yes           Yes          1
## 6 0.2424242 0.28787879     9769821      No            No          0
##   AgressivenessYes
## 1                0
## 2                0
## 3                0
## 4                1
## 5                1
## 6                0
```

```
library(dplyr)
per_game_stats <- per_game_stats %>% select(-Trusted, -Agressiveness)
head(per_game_stats)
```

```
##            Player Team       MPG       PPG       APG      RPG      TOPG
## 1    A.J. Hammons  DAL  7.409091  2.181818 0.1818182 1.636364 0.4545455
## 2    Aaron Brooks  IND 13.753846  4.953846 1.9230769 1.061538 1.0153846
## 3    Aaron Gordon  ORL 28.725000 12.737500 1.8750000 5.062500 1.1125000
## 4 Al-Farouq Aminu  POR 29.065574  8.721311 1.6229508 7.393443 1.5409836
## 5      Al Horford  BOS 32.250000 14.000000 4.9558824 6.823529 1.7058824
## 6    Al Jefferson  IND 14.106061  8.106061 0.8636364 4.212121 0.5000000
##         BPG        SPG Salary17_18 TrustedYes AgressivenessYes
## 1 0.5909091 0.04545455     1312611          0                0
## 2 0.1384615 0.38461538     2116955          0                0
## 3 0.5000000 0.80000000     5504420          1                0
## 4 0.7213115 0.98360656     7319035          1                1
## 5 1.2794118 0.76470588    27734405          1                1
## 6 0.2424242 0.28787879     9769821          0                0
```

Let's re arrange the columns in a proper order.

```
library(dplyr)
per_game_stats <- per_game_stats %>%
  select(Player, Team, MPG, PPG, APG, RPG, TOPG, BPG, SPG, TrustedYes, AgressivenessYes, Salary17_18)
head(per_game_stats)
```

```
##            Player Team       MPG       PPG       APG      RPG      TOPG
## 1    A.J. Hammons  DAL  7.409091  2.181818 0.1818182 1.636364 0.4545455
## 2    Aaron Brooks  IND 13.753846  4.953846 1.9230769 1.061538 1.0153846
## 3    Aaron Gordon  ORL 28.725000 12.737500 1.8750000 5.062500 1.1125000
## 4 Al-Farouq Aminu  POR 29.065574  8.721311 1.6229508 7.393443 1.5409836
## 5      Al Horford  BOS 32.250000 14.000000 4.9558824 6.823529 1.7058824
## 6    Al Jefferson  IND 14.106061  8.106061 0.8636364 4.212121 0.5000000
##         BPG        SPG TrustedYes AgressivenessYes Salary17_18
## 1 0.5909091 0.04545455          0                0     1312611
## 2 0.1384615 0.38461538          0                0     2116955
## 3 0.5000000 0.80000000          1                0     5504420
## 4 0.7213115 0.98360656          1                1     7319035
## 5 1.2794118 0.76470588          1                1    27734405
## 6 0.2424242 0.28787879          0                0     9769821
```

The dataset appears clean and does not require extensive preprocessing. Player statistics such as PPG, MPG, and other performance metrics are inherently meaningful and directly interpretable in their current form, making scaling unnecessary. These values represent individual player performances, and altering them would distort their real-world significance.

Additionally, there are no apparent missing values or extreme outliers based on the summary statistics, indicating that the data is ready for analysis without major cleaning. However, as part of preprocessing, dummy variables will be created for categorical variables like Trusted and Aggressiveness to ensure compatibility with machine learning models that require numerical inputs.

####e.Clustering

Pre-processing

We can exclude non numeric data for clustering like, team, Player, etc

```
data_for_clustering <- per_game_stats %>%
  select(MPG, PPG, APG, RPG, TOPG, BPG, SPG, TrustedYes, AgressivenessYes, Salary17_18)
```

Determining the optimal number of clusters

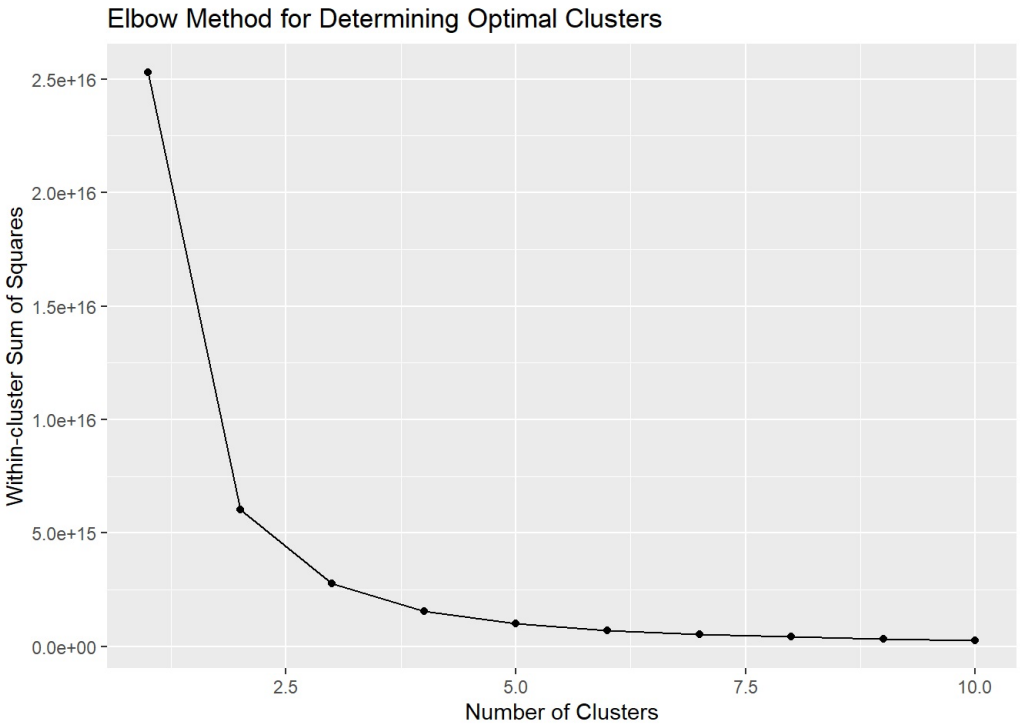
1.Elbow Method

```
library(ggplot2)
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
set.seed(123)
wss <- sapply(1:10, function(k) {
  kmeans(data_for_clustering, centers = k, nstart = 25)$tot.withinss
})

ggplot(data.frame(k = 1:10, wss = wss), aes(x = k, y = wss)) +
  geom_line() +
  geom_point() +
  labs(title = "Elbow Method for Determining Optimal Clusters", x = "Number of Clusters", y = "Within-cluster Sum
of Squares")
```

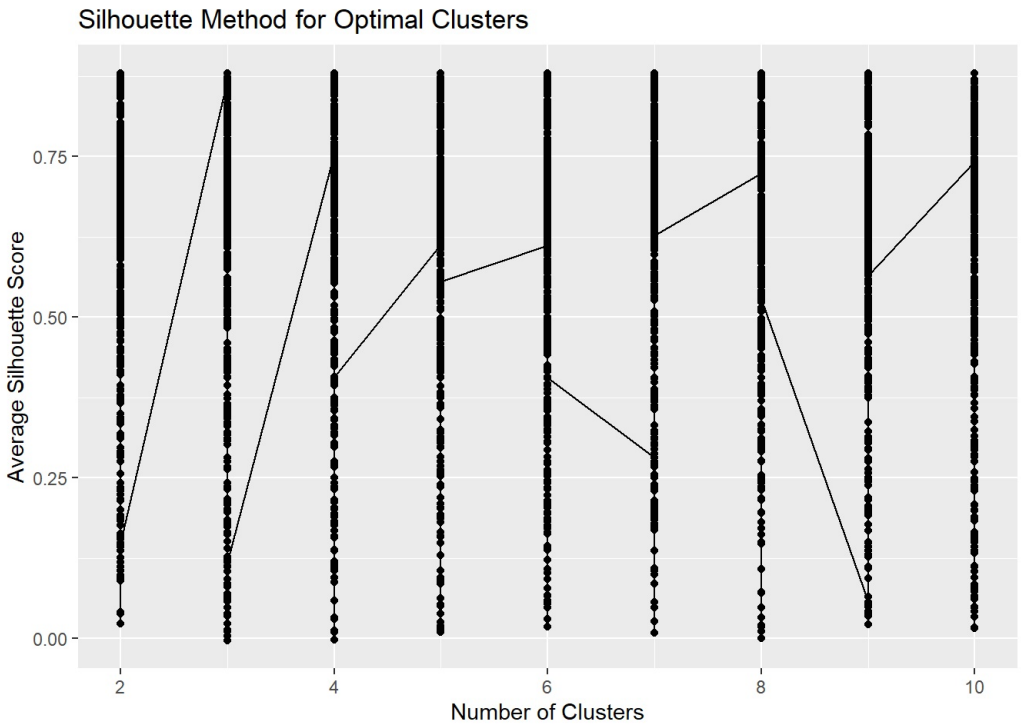


The optimal number of clusters appears to be 3. This is because the WSS continues to decrease after k=3, but the reduction become less steep at k = 3, indicating adding more clusters does not improve the clustering significantly.

2.Silhouette Method:

```
library(cluster)
sil_scores <- sapply(2:10, function(k) {
  km <- kmeans(data_for_clustering, centers = k, nstart = 25)
  silhouette(km$cluster, dist(data_for_clustering))[, 3]
})

sil_plot <- ggplot(data.frame(k = 2:10, silhouette = sapply(sil_scores, mean)), aes(x = k, y = silhouette)) +
  geom_line() + geom_point() +
  labs(title = "Silhouette Method for Optimal Clusters", x = "Number of Clusters", y = "Average Silhouette Score"
)
print(sil_plot)
```



Here we can observe that, 3 is the optimal number of clusters as it has the highest Average Silhouette Score.

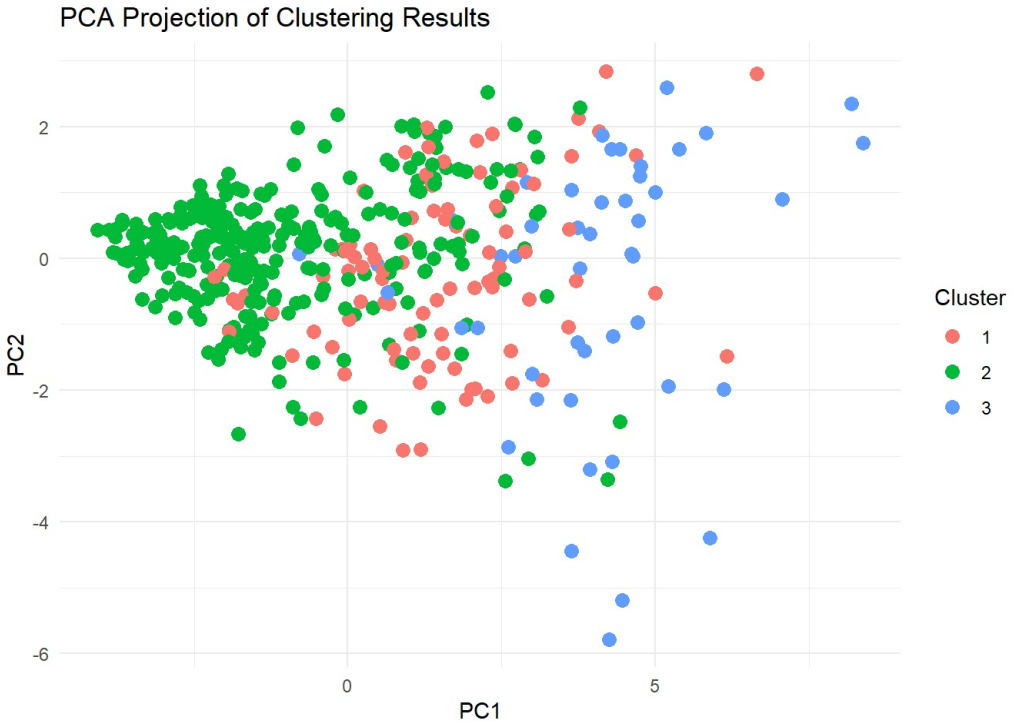
f.Clustering

```
set.seed(123)
kmeans_result <- kmeans(data_for_clustering, centers = 3, nstart = 25)
per_game_stats$Cluster <- as.factor(kmeans_result$cluster)
```

Now let's visualize the clusters with PCA.

```
library(ggplot2)
library(FactoMineR)

pca_result <- prcomp(data_for_clustering, scale. = TRUE)
pca_df <- data.frame(pca_result$x, Cluster = as.factor(kmeans_result$cluster))
ggplot(pca_df, aes(x = PC1, y = PC2, color = Cluster)) +
  geom_point(size = 3) +
  labs(title = "PCA Projection of Clustering Results", x = "PC1", y = "PC2") +
  theme_minimal()
```



observations:

Cluster 1 (RED) seems to form a more compact group

Cluster 2 (GREEN) shows a more spread out group

Cluster 3 (Blue) is more separated and spread, indicating that the players in this cluster may have a different set of characteristics compared to the others.

```
table(per_game_stats$Cluster, per_game_stats$TrustedYes)
```

```
##
##       0   1
##   1  16  79
##   2 205  95
##   3   1  46
```

This table shows how the variable TrustedYes is distributed across the different clusters.

1. Cluster 1 has 16 players with TrustedYes = 0 and 70 with TrustedYes = 1, indicating that most of the players in this cluster are trusted by their coach
2. Cluster 2 has 205 players with TrustedYes = 0 and 95 with TrustedYes = 1, indicating that this cluster has a higher number of non-trusted players.
3. Cluster 3 has 1 players with TrustedYes = 0 and 46 with TrustedYes = 1, indicating that this cluster has almost entirely trusted players.

This suggests that cluster 3 may be the cluster most associated with trusted players, while cluster 2 contains a more balanced mix, and cluster 1 is skewed towards trusted players.

```
table(per_game_stats$Cluster, per_game_stats$AgressivenessYes)
```

```
##
##       0   1
##   1  40  55
##   2 220  80
##   3   4  43
```

This table shows how the variable AgressivenessYes is distributed across the different clusters.

1. Cluster 1 has 40 players with AgressivenessYes = 0 and 55 players with AgressivenessYes = 1, showing a fairly balanced distribution.
2. cluster 2 has 220 players with AgressivenessYes = 0 and 80 with AgressivenessYes = 1, indicating that this cluster has a higher number of non-aggressive players.
3. cluster 3 has a 4 players with AgressivenessYes = 0 and 43 with AgressivenessYes = 1, suggesting that it contains a high proportion of aggressive players.

Cluster 3 has the most aggressive players, while cluster 2 has a higher proportion of non-aggressive players.

```
stats_salary_regression <-
  per_game_stats %>% select(MPG, PPG, APG, RPG, TOPG, BPG, SPG, TrustedYes,AgressivenessYes, Salary17_18 )
lm(Salary17_18~ TrustedYes * AgressivenessYes, data=stats_salary_regression)
```

```
##
## Call:
## lm(formula = Salary17_18 ~ TrustedYes * AgressivenessYes, data = stats_salary_regression)
##
## Coefficients:
##                 (Intercept)                   TrustedYes
##                     2914582                      5125780
##           AgressivenessYes  TrustedYes:AgressivenessYes
##                      969783                      3518647
```

Intercept (2,914,582): This is the baseline salary for a player who is neither trusted nor aggressive.

TrustedYes (5,125,780): Players trusted by the coach (i.e., those who play more minutes) earn significantly more, even when adjusting for other factors.

AggressivenessYes (969,783): More aggressive players (those with higher turnovers) also have a slight salary increase.

TrustedYes(3,518,647): When a player is both trusted and aggressive, their salary further increases significantly.

Players who are both trusted by their coach (play more minutes) and play aggressively (higher turnovers) tend to have higher salaries. The trend lines in the plot confirm that aggressive players tend to score more points per game at similar salary levels.

f. Classification

stepwise model:

Here let's include all the variables, second order terms and interaction terms in the model and then perform stepwise regression to identify which combination comes out as the best ones.

```
library(caret)
```

```
## Loading required package: lattice
```

```
model2 <- lm(Salary17_18 ~ (MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes + AgressivenessYes)^2 +
                        I(MPG^2) + I(PPG^2) + I(APG^2) + I(RPG^2) + I(TOPG^2) + I(BPG^2) + I(SPG^2) +
                        I(TrustedYes^2) + I(AgressivenessYes^2), data = per_game_stats,trControl = train_control)
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
model_stepwise <- step(model2, direction = "both", trace = 1)
```

```
## Start:  AIC=13668.5
## Salary17_18 ~ (MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes)^2 + I(MPG^2) + I(PPG^2) + I(APG^2) + I(RPG^2) +
##     I(TOPG^2) + I(BPG^2) + I(SPG^2) + I(TrustedYes^2) + I(AgressivenessYes^2)
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13668.5
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(MPG^2) + I(PPG^2) + I(APG^2) + I(RPG^2) +
##     I(TOPG^2) + I(BPG^2) + I(SPG^2) + I(TrustedYes^2) + MPG:PPG +
##     MPG:APG + MPG:RPG + MPG:TOPG + MPG:BPG + MPG:SPG + MPG:TrustedYes +
##     MPG:AgressivenessYes + PPG:APG + PPG:RPG + PPG:TOPG + PPG:BPG +
##     PPG:SPG + PPG:TrustedYes + PPG:AgressivenessYes + APG:RPG +
##     APG:TOPG + APG:BPG + APG:SPG + APG:TrustedYes + APG:AgressivenessYes +
##     RPG:TOPG + RPG:BPG + RPG:SPG + RPG:TrustedYes + RPG:AgressivenessYes +
##     TOPG:BPG + TOPG:SPG + TOPG:TrustedYes + TOPG:AgressivenessYes +
##     BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes + SPG:TrustedYes +
##     SPG:AgressivenessYes + TrustedYes:AgressivenessYes
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13668.5
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(MPG^2) + I(PPG^2) + I(APG^2) + I(RPG^2) +
##     I(TOPG^2) + I(BPG^2) + I(SPG^2) + MPG:PPG + MPG:APG + MPG:RPG +
##     MPG:TOPG + MPG:BPG + MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes +
##     PPG:APG + PPG:RPG + PPG:TOPG + PPG:BPG + PPG:SPG + PPG:TrustedYes +
##     PPG:AgressivenessYes + APG:RPG + APG:TOPG + APG:BPG + APG:SPG +
##     APG:TrustedYes + APG:AgressivenessYes + RPG:TOPG + RPG:BPG +
##     RPG:SPG + RPG:TrustedYes + RPG:AgressivenessYes + TOPG:BPG +
##     TOPG:SPG + TOPG:TrustedYes + TOPG:AgressivenessYes + BPG:SPG +
##     BPG:TrustedYes + BPG:AgressivenessYes + SPG:TrustedYes +
##     SPG:AgressivenessYes + TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - MPG:RPG                      1 1.5015e+10 9.3645e+15 13666
## - APG:RPG                      1 2.8404e+10 9.3645e+15 13666
## - PPG:TrustedYes               1 1.3977e+11 9.3646e+15 13666
## - MPG:PPG                      1 2.3557e+11 9.3647e+15 13666
## - TOPG:TrustedYes              1 2.9463e+11 9.3648e+15 13666
## - RPG:TOPG                     1 3.3107e+11 9.3648e+15 13666
## - PPG:APG                      1 3.4344e+11 9.3648e+15 13666
## - APG:TOPG                     1 4.8873e+11 9.3650e+15 13666
## - APG:BPG                      1 1.1574e+12 9.3657e+15 13667
## - I(TOPG^2)                    1 1.3109e+12 9.3658e+15 13667
## - I(MPG^2)                     1 1.3810e+12 9.3659e+15 13667
## - RPG:AgressivenessYes         1 1.7359e+12 9.3662e+15 13667
## - TOPG:AgressivenessYes        1 2.3868e+12 9.3669e+15 13667
## - MPG:TOPG                     1 2.7048e+12 9.3672e+15 13667
## - I(RPG^2)                     1 2.7343e+12 9.3672e+15 13667
## - I(PPG^2)                     1 3.9300e+12 9.3684e+15 13667
## - PPG:RPG                      1 4.7660e+12 9.3693e+15 13667
## - RPG:BPG                      1 6.3516e+12 9.3708e+15 13667
## - MPG:TrustedYes               1 7.2547e+12 9.3717e+15 13667
## - I(SPG^2)                     1 8.9559e+12 9.3734e+15 13667
## - I(BPG^2)                     1 1.3121e+13 9.3776e+15 13667
## - TrustedYes:AgressivenessYes  1 1.3275e+13 9.3778e+15 13667
## - MPG:BPG                      1 1.6638e+13 9.3811e+15 13667
## - MPG:AgressivenessYes         1 1.7470e+13 9.3820e+15 13667
## - TOPG:SPG                     1 1.8126e+13 9.3826e+15 13667
## - I(APG^2)                     1 1.9578e+13 9.3841e+15 13667
## - RPG:TrustedYes               1 2.0582e+13 9.3851e+15 13668
## - SPG:TrustedYes               1 2.4923e+13 9.3894e+15 13668
## - SPG:AgressivenessYes         1 2.5822e+13 9.3903e+15 13668
## - APG:TrustedYes               1 2.9161e+13 9.3937e+15 13668
## - BPG:TrustedYes               1 2.9270e+13 9.3938e+15 13668
## - RPG:SPG                      1 3.0774e+13 9.3953e+15 13668
## - PPG:TOPG                     1 3.6942e+13 9.4014e+15 13668
## - BPG:AgressivenessYes         1 3.8096e+13 9.4026e+15 13668
## - APG:AgressivenessYes         1 4.1643e+13 9.4061e+15 13668
## <none>                                      9.3645e+15 13668
## - TOPG:BPG                     1 4.5347e+13 9.4098e+15 13669
## - BPG:SPG                      1 4.6122e+13 9.4106e+15 13669
## - MPG:SPG                      1 6.7645e+13 9.4321e+15 13670
## - APG:SPG                      1 7.1384e+13 9.4359e+15 13670
## - PPG:SPG                      1 9.3356e+13 9.4578e+15 13671
## - PPG:BPG                      1 9.3645e+13 9.4581e+15 13671
## - MPG:APG                      1 9.8702e+13 9.4632e+15 13671
## - PPG:AgressivenessYes         1 1.0033e+14 9.4648e+15 13671
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13666.5
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(MPG^2) + I(PPG^2) + I(APG^2) + I(RPG^2) +
##     I(TOPG^2) + I(BPG^2) + I(SPG^2) + MPG:PPG + MPG:APG + MPG:TOPG +
##     MPG:BPG + MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes +
##     PPG:APG + PPG:RPG + PPG:TOPG + PPG:BPG + PPG:SPG + PPG:TrustedYes +
##     PPG:AgressivenessYes + APG:RPG + APG:TOPG + APG:BPG + APG:SPG +
##     APG:TrustedYes + APG:AgressivenessYes + RPG:TOPG + RPG:BPG +
##     RPG:SPG + RPG:TrustedYes + RPG:AgressivenessYes + TOPG:BPG +
##     TOPG:SPG + TOPG:TrustedYes + TOPG:AgressivenessYes + BPG:SPG +
##     BPG:TrustedYes + BPG:AgressivenessYes + SPG:TrustedYes +
##     SPG:AgressivenessYes + TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - APG:RPG                      1 2.2567e+10 9.3645e+15 13664
## - PPG:TrustedYes               1 1.3614e+11 9.3646e+15 13664
## - MPG:PPG                      1 2.5269e+11 9.3648e+15 13664
## - TOPG:TrustedYes              1 2.9711e+11 9.3648e+15 13664
## - PPG:APG                      1 3.3164e+11 9.3648e+15 13664
## - RPG:TOPG                     1 3.9702e+11 9.3649e+15 13664
## - APG:TOPG                     1 4.8127e+11 9.3650e+15 13664
## - APG:BPG                      1 1.1431e+12 9.3657e+15 13665
## - I(TOPG^2)                    1 1.3426e+12 9.3659e+15 13665
## - I(MPG^2)                     1 1.6173e+12 9.3661e+15 13665
## - RPG:AgressivenessYes         1 1.7265e+12 9.3662e+15 13665
## - TOPG:AgressivenessYes        1 2.3886e+12 9.3669e+15 13665
## - MPG:TOPG                     1 2.7010e+12 9.3672e+15 13665
## - I(RPG^2)                     1 3.0282e+12 9.3675e+15 13665
## - I(PPG^2)                     1 3.9830e+12 9.3685e+15 13665
## - RPG:BPG                      1 6.3702e+12 9.3709e+15 13665
## - MPG:TrustedYes               1 7.2413e+12 9.3718e+15 13665
## - PPG:RPG                      1 7.8801e+12 9.3724e+15 13665
## - I(SPG^2)                     1 9.0281e+12 9.3735e+15 13665
## - TrustedYes:AgressivenessYes  1 1.3291e+13 9.3778e+15 13665
## - I(BPG^2)                     1 1.3519e+13 9.3780e+15 13665
## - MPG:AgressivenessYes         1 1.7514e+13 9.3820e+15 13665
## - TOPG:SPG                     1 1.8183e+13 9.3827e+15 13665
## - I(APG^2)                     1 1.9703e+13 9.3842e+15 13665
## - SPG:TrustedYes               1 2.4942e+13 9.3895e+15 13666
## - MPG:BPG                      1 2.5267e+13 9.3898e+15 13666
## - SPG:AgressivenessYes         1 2.5818e+13 9.3903e+15 13666
## - APG:TrustedYes               1 2.9200e+13 9.3937e+15 13666
## - RPG:TrustedYes               1 3.1523e+13 9.3960e+15 13666
## - RPG:SPG                      1 3.2854e+13 9.3974e+15 13666
## - BPG:TrustedYes               1 3.4066e+13 9.3986e+15 13666
## - PPG:TOPG                     1 3.7622e+13 9.4021e+15 13666
## - BPG:AgressivenessYes         1 3.8428e+13 9.4029e+15 13666
## - APG:AgressivenessYes         1 4.1674e+13 9.4062e+15 13666
## <none>                                      9.3645e+15 13666
## - BPG:SPG                      1 5.0061e+13 9.4146e+15 13667
## - TOPG:BPG                     1 5.1271e+13 9.4158e+15 13667
## - MPG:SPG                      1 6.8224e+13 9.4327e+15 13668
## - APG:SPG                      1 7.1388e+13 9.4359e+15 13668
## + MPG:RPG                      1 1.5015e+10 9.3645e+15 13668
## - PPG:SPG                      1 9.4009e+13 9.4585e+15 13669
## - PPG:AgressivenessYes         1 1.0037e+14 9.4649e+15 13669
## - MPG:APG                      1 1.0054e+14 9.4650e+15 13669
## - PPG:BPG                      1 1.1130e+14 9.4758e+15 13670
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13664.5
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(MPG^2) + I(PPG^2) + I(APG^2) + I(RPG^2) +
##     I(TOPG^2) + I(BPG^2) + I(SPG^2) + MPG:PPG + MPG:APG + MPG:TOPG +
##     MPG:BPG + MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes +
##     PPG:APG + PPG:RPG + PPG:TOPG + PPG:BPG + PPG:SPG + PPG:TrustedYes +
##     PPG:AgressivenessYes + APG:TOPG + APG:BPG + APG:SPG + APG:TrustedYes +
##     APG:AgressivenessYes + RPG:TOPG + RPG:BPG + RPG:SPG + RPG:TrustedYes +
##     RPG:AgressivenessYes + TOPG:BPG + TOPG:SPG + TOPG:TrustedYes +
##     TOPG:AgressivenessYes + BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes +
##     SPG:TrustedYes + SPG:AgressivenessYes + TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - PPG:TrustedYes               1 1.2878e+11 9.3647e+15 13662
## - MPG:PPG                      1 2.4265e+11 9.3648e+15 13662
## - TOPG:TrustedYes              1 2.8131e+11 9.3648e+15 13662
## - PPG:APG                      1 3.3836e+11 9.3649e+15 13662
## - APG:TOPG                     1 4.5875e+11 9.3650e+15 13662
## - RPG:TOPG                     1 9.1740e+11 9.3654e+15 13662
## - I(TOPG^2)                    1 1.3709e+12 9.3659e+15 13663
## - APG:BPG                      1 1.3873e+12 9.3659e+15 13663
## - I(MPG^2)                     1 1.5949e+12 9.3661e+15 13663
## - RPG:AgressivenessYes         1 1.7141e+12 9.3662e+15 13663
## - TOPG:AgressivenessYes        1 2.3957e+12 9.3669e+15 13663
## - MPG:TOPG                     1 2.6964e+12 9.3672e+15 13663
## - I(PPG^2)                     1 3.9687e+12 9.3685e+15 13663
## - I(RPG^2)                     1 4.2967e+12 9.3688e+15 13663
## - RPG:BPG                      1 6.3632e+12 9.3709e+15 13663
## - MPG:TrustedYes               1 7.2979e+12 9.3718e+15 13663
## - PPG:RPG                      1 7.8581e+12 9.3724e+15 13663
## - I(SPG^2)                     1 9.0295e+12 9.3736e+15 13663
## - TrustedYes:AgressivenessYes  1 1.3327e+13 9.3779e+15 13663
## - I(BPG^2)                     1 1.4235e+13 9.3788e+15 13663
## - MPG:AgressivenessYes         1 1.7586e+13 9.3821e+15 13663
## - TOPG:SPG                     1 1.9341e+13 9.3839e+15 13663
## - I(APG^2)                     1 2.2048e+13 9.3866e+15 13664
## - SPG:TrustedYes               1 2.5007e+13 9.3895e+15 13664
## - MPG:BPG                      1 2.5704e+13 9.3902e+15 13664
## - SPG:AgressivenessYes         1 2.5888e+13 9.3904e+15 13664
## - APG:TrustedYes               1 3.0659e+13 9.3952e+15 13664
## - RPG:TrustedYes               1 3.2392e+13 9.3969e+15 13664
## - BPG:TrustedYes               1 3.4309e+13 9.3988e+15 13664
## - RPG:SPG                      1 3.4845e+13 9.3994e+15 13664
## - PPG:TOPG                     1 3.7628e+13 9.4022e+15 13664
## - BPG:AgressivenessYes         1 3.8697e+13 9.4032e+15 13664
## - APG:AgressivenessYes         1 4.2121e+13 9.4067e+15 13664
## <none>                                      9.3645e+15 13664
## - BPG:SPG                      1 5.0733e+13 9.4153e+15 13665
## - TOPG:BPG                     1 5.9491e+13 9.4240e+15 13665
## - MPG:SPG                      1 6.8269e+13 9.4328e+15 13666
## - APG:SPG                      1 7.8252e+13 9.4428e+15 13666
## + APG:RPG                      1 2.2567e+10 9.3645e+15 13666
## + MPG:RPG                      1 9.1775e+09 9.3645e+15 13666
## - PPG:SPG                      1 9.4076e+13 9.4586e+15 13667
## - PPG:AgressivenessYes         1 1.0057e+14 9.4651e+15 13667
## - MPG:APG                      1 1.0058e+14 9.4651e+15 13667
## - PPG:BPG                      1 1.1143e+14 9.4760e+15 13668
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13662.51
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(MPG^2) + I(PPG^2) + I(APG^2) + I(RPG^2) +
##     I(TOPG^2) + I(BPG^2) + I(SPG^2) + MPG:PPG + MPG:APG + MPG:TOPG +
##     MPG:BPG + MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes +
##     PPG:APG + PPG:RPG + PPG:TOPG + PPG:BPG + PPG:SPG + PPG:AgressivenessYes +
##     APG:TOPG + APG:BPG + APG:SPG + APG:TrustedYes + APG:AgressivenessYes +
##     RPG:TOPG + RPG:BPG + RPG:SPG + RPG:TrustedYes + RPG:AgressivenessYes +
##     TOPG:BPG + TOPG:SPG + TOPG:TrustedYes + TOPG:AgressivenessYes +
##     BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes + SPG:TrustedYes +
##     SPG:AgressivenessYes + TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - MPG:PPG                      1 1.2435e+11 9.3648e+15 13660
## - TOPG:TrustedYes              1 1.9580e+11 9.3649e+15 13660
## - PPG:APG                      1 3.5038e+11 9.3650e+15 13660
## - APG:TOPG                     1 4.3992e+11 9.3651e+15 13660
## - RPG:TOPG                     1 9.4189e+11 9.3656e+15 13661
## - APG:BPG                      1 1.3415e+12 9.3660e+15 13661
## - I(TOPG^2)                    1 1.4094e+12 9.3661e+15 13661
## - I(MPG^2)                     1 1.5049e+12 9.3662e+15 13661
## - RPG:AgressivenessYes         1 1.7919e+12 9.3665e+15 13661
## - TOPG:AgressivenessYes        1 2.3215e+12 9.3670e+15 13661
## - MPG:TOPG                     1 3.2150e+12 9.3679e+15 13661
## - I(PPG^2)                     1 3.8649e+12 9.3685e+15 13661
## - I(RPG^2)                     1 4.4516e+12 9.3691e+15 13661
## - RPG:BPG                      1 6.2518e+12 9.3709e+15 13661
## - PPG:RPG                      1 7.9969e+12 9.3727e+15 13661
## - I(SPG^2)                     1 9.0416e+12 9.3737e+15 13661
## - MPG:TrustedYes               1 1.0344e+13 9.3750e+15 13661
## - TrustedYes:AgressivenessYes  1 1.3525e+13 9.3782e+15 13661
## - I(BPG^2)                     1 1.4208e+13 9.3789e+15 13661
## - MPG:AgressivenessYes         1 1.9173e+13 9.3838e+15 13661
## - TOPG:SPG                     1 2.0040e+13 9.3847e+15 13662
## - I(APG^2)                     1 2.2053e+13 9.3867e+15 13662
## - SPG:TrustedYes               1 2.4879e+13 9.3895e+15 13662
## - SPG:AgressivenessYes         1 2.5855e+13 9.3905e+15 13662
## - MPG:BPG                      1 2.6646e+13 9.3913e+15 13662
## - APG:TrustedYes               1 3.2221e+13 9.3969e+15 13662
## - RPG:TrustedYes               1 3.2286e+13 9.3969e+15 13662
## - RPG:SPG                      1 3.4717e+13 9.3994e+15 13662
## - BPG:TrustedYes               1 3.4725e+13 9.3994e+15 13662
## - PPG:TOPG                     1 3.7500e+13 9.4022e+15 13662
## - BPG:AgressivenessYes         1 3.8787e+13 9.4034e+15 13662
## - APG:AgressivenessYes         1 4.1996e+13 9.4067e+15 13662
## <none>                                      9.3647e+15 13662
## - BPG:SPG                      1 5.0619e+13 9.4153e+15 13663
## - TOPG:BPG                     1 6.0215e+13 9.4249e+15 13663
## - MPG:SPG                      1 6.8141e+13 9.4328e+15 13664
## - APG:SPG                      1 7.8543e+13 9.4432e+15 13664
## + PPG:TrustedYes               1 1.2878e+11 9.3645e+15 13664
## + APG:RPG                      1 1.5215e+10 9.3646e+15 13664
## + MPG:RPG                      1 7.2262e+09 9.3647e+15 13664
## - PPG:SPG                      1 9.4350e+13 9.4590e+15 13665
## - MPG:APG                      1 1.0330e+14 9.4680e+15 13665
## - PPG:AgressivenessYes         1 1.1137e+14 9.4760e+15 13666
## - PPG:BPG                      1 1.1360e+14 9.4783e+15 13666
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13660.51
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(MPG^2) + I(PPG^2) + I(APG^2) + I(RPG^2) +
##     I(TOPG^2) + I(BPG^2) + I(SPG^2) + MPG:APG + MPG:TOPG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:APG +
##     PPG:RPG + PPG:TOPG + PPG:BPG + PPG:SPG + PPG:AgressivenessYes +
##     APG:TOPG + APG:BPG + APG:SPG + APG:TrustedYes + APG:AgressivenessYes +
##     RPG:TOPG + RPG:BPG + RPG:SPG + RPG:TrustedYes + RPG:AgressivenessYes +
##     TOPG:BPG + TOPG:SPG + TOPG:TrustedYes + TOPG:AgressivenessYes +
##     BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes + SPG:TrustedYes +
##     SPG:AgressivenessYes + TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - TOPG:TrustedYes              1 1.8670e+11 9.3650e+15 13658
## - PPG:APG                      1 4.1392e+11 9.3652e+15 13658
## - APG:TOPG                     1 4.6939e+11 9.3653e+15 13658
## - RPG:TOPG                     1 8.4681e+11 9.3656e+15 13659
## - APG:BPG                      1 1.2688e+12 9.3661e+15 13659
## - I(TOPG^2)                    1 1.3248e+12 9.3661e+15 13659
## - I(MPG^2)                     1 1.5064e+12 9.3663e+15 13659
## - RPG:AgressivenessYes         1 1.7445e+12 9.3665e+15 13659
## - TOPG:AgressivenessYes        1 2.5184e+12 9.3673e+15 13659
## - I(RPG^2)                     1 4.5352e+12 9.3693e+15 13659
## - MPG:TOPG                     1 4.6731e+12 9.3695e+15 13659
## - RPG:BPG                      1 6.2332e+12 9.3710e+15 13659
## - I(PPG^2)                     1 7.8778e+12 9.3727e+15 13659
## - PPG:RPG                      1 8.1343e+12 9.3729e+15 13659
## - I(SPG^2)                     1 9.1621e+12 9.3739e+15 13659
## - MPG:TrustedYes               1 1.0221e+13 9.3750e+15 13659
## - TrustedYes:AgressivenessYes  1 1.3492e+13 9.3783e+15 13659
## - I(BPG^2)                     1 1.4238e+13 9.3790e+15 13659
## - MPG:AgressivenessYes         1 1.9091e+13 9.3839e+15 13659
## - TOPG:SPG                     1 2.1243e+13 9.3860e+15 13660
## - I(APG^2)                     1 2.1939e+13 9.3867e+15 13660
## - SPG:TrustedYes               1 2.4757e+13 9.3895e+15 13660
## - SPG:AgressivenessYes         1 2.6134e+13 9.3909e+15 13660
## - MPG:BPG                      1 2.7086e+13 9.3919e+15 13660
## - APG:TrustedYes               1 3.2136e+13 9.3969e+15 13660
## - RPG:TrustedYes               1 3.2230e+13 9.3970e+15 13660
## - RPG:SPG                      1 3.4637e+13 9.3994e+15 13660
## - BPG:TrustedYes               1 3.4806e+13 9.3996e+15 13660
## - BPG:AgressivenessYes         1 3.9952e+13 9.4047e+15 13660
## - PPG:TOPG                     1 4.1601e+13 9.4064e+15 13660
## - APG:AgressivenessYes         1 4.2063e+13 9.4068e+15 13660
## <none>                                      9.3648e+15 13660
## - BPG:SPG                      1 5.0510e+13 9.4153e+15 13661
## - TOPG:BPG                     1 6.0183e+13 9.4250e+15 13661
## - MPG:SPG                      1 7.2160e+13 9.4369e+15 13662
## - APG:SPG                      1 7.8712e+13 9.4435e+15 13662
## + MPG:PPG                      1 1.2435e+11 9.3647e+15 13662
## + MPG:RPG                      1 2.0650e+10 9.3648e+15 13662
## + APG:RPG                      1 1.1672e+10 9.3648e+15 13662
## + PPG:TrustedYes               1 1.0479e+10 9.3648e+15 13662
## - MPG:APG                      1 1.0535e+14 9.4701e+15 13664
## - PPG:SPG                      1 1.1119e+14 9.4760e+15 13664
## - PPG:BPG                      1 1.1430e+14 9.4791e+15 13664
## - PPG:AgressivenessYes         1 1.1640e+14 9.4812e+15 13664
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13658.52
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(MPG^2) + I(PPG^2) + I(APG^2) + I(RPG^2) +
##     I(TOPG^2) + I(BPG^2) + I(SPG^2) + MPG:APG + MPG:TOPG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:APG +
##     PPG:RPG + PPG:TOPG + PPG:BPG + PPG:SPG + PPG:AgressivenessYes +
##     APG:TOPG + APG:BPG + APG:SPG + APG:TrustedYes + APG:AgressivenessYes +
##     RPG:TOPG + RPG:BPG + RPG:SPG + RPG:TrustedYes + RPG:AgressivenessYes +
##     TOPG:BPG + TOPG:SPG + TOPG:AgressivenessYes + BPG:SPG + BPG:TrustedYes +
##     BPG:AgressivenessYes + SPG:TrustedYes + SPG:AgressivenessYes +
##     TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - APG:TOPG                     1 4.8242e+11 9.3655e+15 13656
## - PPG:APG                      1 4.9023e+11 9.3655e+15 13656
## - RPG:TOPG                     1 9.5968e+11 9.3659e+15 13657
## - APG:BPG                      1 1.2931e+12 9.3663e+15 13657
## - I(TOPG^2)                    1 1.3005e+12 9.3663e+15 13657
## - I(MPG^2)                     1 1.6787e+12 9.3667e+15 13657
## - RPG:AgressivenessYes         1 1.6907e+12 9.3667e+15 13657
## - TOPG:AgressivenessYes        1 2.4158e+12 9.3674e+15 13657
## - I(RPG^2)                     1 4.5624e+12 9.3695e+15 13657
## - RPG:BPG                      1 6.4570e+12 9.3714e+15 13657
## - I(PPG^2)                     1 7.9625e+12 9.3729e+15 13657
## - MPG:TOPG                     1 8.0629e+12 9.3730e+15 13657
## - PPG:RPG                      1 8.4017e+12 9.3734e+15 13657
## - I(SPG^2)                     1 9.0593e+12 9.3740e+15 13657
## - MPG:TrustedYes               1 1.0039e+13 9.3750e+15 13657
## - I(BPG^2)                     1 1.5517e+13 9.3805e+15 13657
## - TrustedYes:AgressivenessYes  1 1.7744e+13 9.3827e+15 13657
## - MPG:AgressivenessYes         1 1.9305e+13 9.3843e+15 13657
## - TOPG:SPG                     1 2.1930e+13 9.3869e+15 13658
## - I(APG^2)                     1 2.2290e+13 9.3873e+15 13658
## - SPG:TrustedYes               1 2.4640e+13 9.3896e+15 13658
## - SPG:AgressivenessYes         1 2.6352e+13 9.3913e+15 13658
## - MPG:BPG                      1 2.9989e+13 9.3950e+15 13658
## - RPG:TrustedYes               1 3.3102e+13 9.3981e+15 13658
## - RPG:SPG                      1 3.4844e+13 9.3998e+15 13658
## - BPG:TrustedYes               1 3.6029e+13 9.4010e+15 13658
## - BPG:AgressivenessYes         1 3.9777e+13 9.4047e+15 13658
## - PPG:TOPG                     1 4.1778e+13 9.4067e+15 13658
## - APG:AgressivenessYes         1 4.2261e+13 9.4072e+15 13658
## <none>                                      9.3650e+15 13658
## - APG:TrustedYes               1 4.4841e+13 9.4098e+15 13659
## - BPG:SPG                      1 5.1080e+13 9.4161e+15 13659
## - TOPG:BPG                     1 6.0620e+13 9.4256e+15 13659
## - MPG:SPG                      1 7.1973e+13 9.4369e+15 13660
## - APG:SPG                      1 8.0356e+13 9.4453e+15 13660
## + TOPG:TrustedYes              1 1.8670e+11 9.3648e+15 13660
## + MPG:PPG                      1 1.1525e+11 9.3649e+15 13660
## + MPG:RPG                      1 2.6494e+10 9.3649e+15 13660
## + APG:RPG                      1 3.7239e+09 9.3650e+15 13660
## + PPG:TrustedYes               1 1.9928e+07 9.3650e+15 13660
## - PPG:SPG                      1 1.1100e+14 9.4760e+15 13662
## - MPG:APG                      1 1.1454e+14 9.4795e+15 13662
## - PPG:AgressivenessYes         1 1.1648e+14 9.4815e+15 13662
## - PPG:BPG                      1 1.1727e+14 9.4822e+15 13662
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13656.55
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(MPG^2) + I(PPG^2) + I(APG^2) + I(RPG^2) +
##     I(TOPG^2) + I(BPG^2) + I(SPG^2) + MPG:APG + MPG:TOPG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:APG +
##     PPG:RPG + PPG:TOPG + PPG:BPG + PPG:SPG + PPG:AgressivenessYes +
##     APG:BPG + APG:SPG + APG:TrustedYes + APG:AgressivenessYes +
##     RPG:TOPG + RPG:BPG + RPG:SPG + RPG:TrustedYes + RPG:AgressivenessYes +
##     TOPG:BPG + TOPG:SPG + TOPG:AgressivenessYes + BPG:SPG + BPG:TrustedYes +
##     BPG:AgressivenessYes + SPG:TrustedYes + SPG:AgressivenessYes +
##     TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - PPG:APG                      1 1.9708e+11 9.3657e+15 13655
## - RPG:TOPG                     1 1.0724e+12 9.3665e+15 13655
## - I(MPG^2)                     1 1.5912e+12 9.3670e+15 13655
## - RPG:AgressivenessYes         1 1.6351e+12 9.3671e+15 13655
## - APG:BPG                      1 1.6764e+12 9.3671e+15 13655
## - TOPG:AgressivenessYes        1 1.9814e+12 9.3674e+15 13655
## - I(RPG^2)                     1 4.3696e+12 9.3698e+15 13655
## - RPG:BPG                      1 6.7727e+12 9.3722e+15 13655
## - MPG:TOPG                     1 7.7339e+12 9.3732e+15 13655
## - I(TOPG^2)                    1 8.2905e+12 9.3737e+15 13655
## - I(SPG^2)                     1 8.6874e+12 9.3741e+15 13655
## - PPG:RPG                      1 8.8313e+12 9.3743e+15 13655
## - I(PPG^2)                     1 8.9927e+12 9.3744e+15 13655
## - MPG:TrustedYes               1 1.0194e+13 9.3756e+15 13655
## - I(BPG^2)                     1 1.6560e+13 9.3820e+15 13655
## - TrustedYes:AgressivenessYes  1 1.8171e+13 9.3836e+15 13655
## - MPG:AgressivenessYes         1 1.9739e+13 9.3852e+15 13656
## - TOPG:SPG                     1 2.5352e+13 9.3908e+15 13656
## - SPG:TrustedYes               1 2.5554e+13 9.3910e+15 13656
## - SPG:AgressivenessYes         1 2.7871e+13 9.3933e+15 13656
## - MPG:BPG                      1 3.2261e+13 9.3977e+15 13656
## - RPG:TrustedYes               1 3.3311e+13 9.3988e+15 13656
## - RPG:SPG                      1 3.4730e+13 9.4002e+15 13656
## - BPG:TrustedYes               1 3.5548e+13 9.4010e+15 13656
## - BPG:AgressivenessYes         1 3.9295e+13 9.4047e+15 13656
## - I(APG^2)                     1 3.9996e+13 9.4054e+15 13656
## <none>                                      9.3655e+15 13656
## - APG:AgressivenessYes         1 4.3223e+13 9.4087e+15 13657
## - APG:TrustedYes               1 4.6768e+13 9.4122e+15 13657
## - BPG:SPG                      1 5.0618e+13 9.4161e+15 13657
## - PPG:TOPG                     1 5.2181e+13 9.4176e+15 13657
## - TOPG:BPG                     1 6.9170e+13 9.4346e+15 13658
## - MPG:SPG                      1 7.2512e+13 9.4380e+15 13658
## - APG:SPG                      1 8.0613e+13 9.4461e+15 13658
## + APG:TOPG                     1 4.8242e+11 9.3650e+15 13658
## + TOPG:TrustedYes              1 1.9973e+11 9.3653e+15 13658
## + MPG:PPG                      1 1.4378e+11 9.3653e+15 13658
## + MPG:RPG                      1 2.5564e+10 9.3654e+15 13658
## + APG:RPG                      1 7.0298e+09 9.3654e+15 13658
## + PPG:TrustedYes               1 2.7875e+09 9.3655e+15 13658
## - PPG:SPG                      1 1.1130e+14 9.4768e+15 13660
## - MPG:APG                      1 1.1521e+14 9.4807e+15 13660
## - PPG:BPG                      1 1.1679e+14 9.4822e+15 13660
## - PPG:AgressivenessYes         1 1.1804e+14 9.4835e+15 13660
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13654.56
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(MPG^2) + I(PPG^2) + I(APG^2) + I(RPG^2) +
##     I(TOPG^2) + I(BPG^2) + I(SPG^2) + MPG:APG + MPG:TOPG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:RPG +
##     PPG:TOPG + PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:BPG +
##     APG:SPG + APG:TrustedYes + APG:AgressivenessYes + RPG:TOPG +
##     RPG:BPG + RPG:SPG + RPG:TrustedYes + RPG:AgressivenessYes +
##     TOPG:BPG + TOPG:SPG + TOPG:AgressivenessYes + BPG:SPG + BPG:TrustedYes +
##     BPG:AgressivenessYes + SPG:TrustedYes + SPG:AgressivenessYes +
##     TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - RPG:TOPG                     1 1.1092e+12 9.3668e+15 13653
## - APG:BPG                      1 1.5482e+12 9.3672e+15 13653
## - RPG:AgressivenessYes         1 1.6194e+12 9.3673e+15 13653
## - I(MPG^2)                     1 1.6196e+12 9.3673e+15 13653
## - TOPG:AgressivenessYes        1 2.0186e+12 9.3677e+15 13653
## - I(RPG^2)                     1 4.2385e+12 9.3699e+15 13653
## - RPG:BPG                      1 6.9016e+12 9.3726e+15 13653
## - MPG:TOPG                     1 8.1566e+12 9.3738e+15 13653
## - I(SPG^2)                     1 8.4942e+12 9.3741e+15 13653
## - I(TOPG^2)                    1 8.7083e+12 9.3744e+15 13653
## - I(PPG^2)                     1 9.2592e+12 9.3749e+15 13653
## - PPG:RPG                      1 9.6370e+12 9.3753e+15 13653
## - MPG:TrustedYes               1 1.0161e+13 9.3758e+15 13653
## - I(BPG^2)                     1 1.6504e+13 9.3822e+15 13653
## - TrustedYes:AgressivenessYes  1 1.7980e+13 9.3836e+15 13653
## - MPG:AgressivenessYes         1 1.9591e+13 9.3852e+15 13654
## - SPG:TrustedYes               1 2.5368e+13 9.3910e+15 13654
## - TOPG:SPG                     1 2.5392e+13 9.3910e+15 13654
## - SPG:AgressivenessYes         1 2.8179e+13 9.3938e+15 13654
## - RPG:TrustedYes               1 3.4016e+13 9.3997e+15 13654
## - MPG:BPG                      1 3.4151e+13 9.3998e+15 13654
## - RPG:SPG                      1 3.5261e+13 9.4009e+15 13654
## - BPG:TrustedYes               1 3.7744e+13 9.4034e+15 13654
## - BPG:AgressivenessYes         1 4.0092e+13 9.4057e+15 13654
## - I(APG^2)                     1 4.1116e+13 9.4068e+15 13654
## <none>                                      9.3657e+15 13655
## - APG:AgressivenessYes         1 4.3279e+13 9.4089e+15 13655
## - APG:TrustedYes               1 4.8112e+13 9.4138e+15 13655
## - BPG:SPG                      1 5.0534e+13 9.4162e+15 13655
## - TOPG:BPG                     1 7.1206e+13 9.4369e+15 13656
## - MPG:SPG                      1 7.3843e+13 9.4395e+15 13656
## - PPG:TOPG                     1 7.9325e+13 9.4450e+15 13656
## + TOPG:TrustedYes              1 2.4793e+11 9.3654e+15 13656
## + PPG:APG                      1 1.9708e+11 9.3655e+15 13656
## + APG:TOPG                     1 1.8926e+11 9.3655e+15 13656
## + MPG:PPG                      1 1.8259e+11 9.3655e+15 13656
## + MPG:RPG                      1 1.3299e+10 9.3656e+15 13657
## + PPG:TrustedYes               1 5.7733e+09 9.3656e+15 13657
## + APG:RPG                      1 1.0364e+09 9.3656e+15 13657
## - APG:SPG                      1 8.6783e+13 9.4524e+15 13657
## - PPG:AgressivenessYes         1 1.1835e+14 9.4840e+15 13658
## - PPG:BPG                      1 1.2137e+14 9.4870e+15 13658
## - PPG:SPG                      1 1.3151e+14 9.4972e+15 13659
## - MPG:APG                      1 1.9011e+14 9.5558e+15 13661
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13652.61
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(MPG^2) + I(PPG^2) + I(APG^2) + I(RPG^2) +
##     I(TOPG^2) + I(BPG^2) + I(SPG^2) + MPG:APG + MPG:TOPG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:RPG +
##     PPG:TOPG + PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:BPG +
##     APG:SPG + APG:TrustedYes + APG:AgressivenessYes + RPG:BPG +
##     RPG:SPG + RPG:TrustedYes + RPG:AgressivenessYes + TOPG:BPG +
##     TOPG:SPG + TOPG:AgressivenessYes + BPG:SPG + BPG:TrustedYes +
##     BPG:AgressivenessYes + SPG:TrustedYes + SPG:AgressivenessYes +
##     TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - APG:BPG                      1 1.4991e+12 9.3683e+15 13651
## - I(MPG^2)                     1 1.7599e+12 9.3685e+15 13651
## - TOPG:AgressivenessYes        1 2.2854e+12 9.3690e+15 13651
## - RPG:AgressivenessYes         1 3.6081e+12 9.3704e+15 13651
## - I(RPG^2)                     1 4.3847e+12 9.3711e+15 13651
## - RPG:BPG                      1 6.7756e+12 9.3735e+15 13651
## - I(TOPG^2)                    1 7.7670e+12 9.3745e+15 13651
## - MPG:TOPG                     1 8.5949e+12 9.3754e+15 13651
## - I(SPG^2)                     1 9.0159e+12 9.3758e+15 13651
## - I(PPG^2)                     1 9.3269e+12 9.3761e+15 13651
## - PPG:RPG                      1 1.0279e+13 9.3770e+15 13651
## - MPG:TrustedYes               1 1.0375e+13 9.3771e+15 13651
## - I(BPG^2)                     1 1.6113e+13 9.3829e+15 13651
## - TrustedYes:AgressivenessYes  1 1.7947e+13 9.3847e+15 13652
## - MPG:AgressivenessYes         1 1.9871e+13 9.3866e+15 13652
## - SPG:TrustedYes               1 2.5403e+13 9.3922e+15 13652
## - TOPG:SPG                     1 2.6124e+13 9.3929e+15 13652
## - SPG:AgressivenessYes         1 2.8370e+13 9.3951e+15 13652
## - RPG:TrustedYes               1 3.2907e+13 9.3997e+15 13652
## - MPG:BPG                      1 3.3156e+13 9.3999e+15 13652
## - RPG:SPG                      1 3.6104e+13 9.4029e+15 13652
## - BPG:TrustedYes               1 3.7100e+13 9.4039e+15 13652
## - BPG:AgressivenessYes         1 3.8983e+13 9.4057e+15 13652
## <none>                                      9.3668e+15 13653
## - I(APG^2)                     1 4.3453e+13 9.4102e+15 13653
## - APG:AgressivenessYes         1 4.4710e+13 9.4115e+15 13653
## - APG:TrustedYes               1 4.8817e+13 9.4156e+15 13653
## - BPG:SPG                      1 4.9674e+13 9.4164e+15 13653
## - MPG:SPG                      1 7.3347e+13 9.4401e+15 13654
## - TOPG:BPG                     1 8.1227e+13 9.4480e+15 13654
## - PPG:TOPG                     1 8.1691e+13 9.4485e+15 13654
## + RPG:TOPG                     1 1.1092e+12 9.3657e+15 13655
## + APG:RPG                      1 4.2003e+11 9.3663e+15 13655
## + TOPG:TrustedYes              1 3.9335e+11 9.3664e+15 13655
## + APG:TOPG                     1 2.3990e+11 9.3665e+15 13655
## + PPG:APG                      1 2.3385e+11 9.3665e+15 13655
## + MPG:RPG                      1 9.1131e+10 9.3667e+15 13655
## + MPG:PPG                      1 5.0223e+10 9.3667e+15 13655
## + PPG:TrustedYes               1 6.6670e+08 9.3668e+15 13655
## - APG:SPG                      1 9.0318e+13 9.4571e+15 13655
## - PPG:AgressivenessYes         1 1.2444e+14 9.4912e+15 13656
## - PPG:SPG                      1 1.3341e+14 9.5002e+15 13657
## - PPG:BPG                      1 1.3484e+14 9.5016e+15 13657
## - MPG:APG                      1 1.9187e+14 9.5586e+15 13660
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13650.68
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(MPG^2) + I(PPG^2) + I(APG^2) + I(RPG^2) +
##     I(TOPG^2) + I(BPG^2) + I(SPG^2) + MPG:APG + MPG:TOPG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:RPG +
##     PPG:TOPG + PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG +
##     APG:TrustedYes + APG:AgressivenessYes + RPG:BPG + RPG:SPG +
##     RPG:TrustedYes + RPG:AgressivenessYes + TOPG:BPG + TOPG:SPG +
##     TOPG:AgressivenessYes + BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes +
##     SPG:TrustedYes + SPG:AgressivenessYes + TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - I(MPG^2)                     1 1.5141e+12 9.3698e+15 13649
## - TOPG:AgressivenessYes        1 1.7904e+12 9.3700e+15 13649
## - RPG:AgressivenessYes         1 3.4474e+12 9.3717e+15 13649
## - I(RPG^2)                     1 4.0577e+12 9.3723e+15 13649
## - RPG:BPG                      1 6.2239e+12 9.3745e+15 13649
## - MPG:TOPG                     1 8.8921e+12 9.3772e+15 13649
## - I(SPG^2)                     1 9.6523e+12 9.3779e+15 13649
## - I(PPG^2)                     1 9.7113e+12 9.3780e+15 13649
## - PPG:RPG                      1 9.9740e+12 9.3782e+15 13649
## - I(TOPG^2)                    1 1.0346e+13 9.3786e+15 13649
## - MPG:TrustedYes               1 1.0810e+13 9.3791e+15 13649
## - TrustedYes:AgressivenessYes  1 1.9199e+13 9.3875e+15 13650
## - MPG:AgressivenessYes         1 2.1070e+13 9.3893e+15 13650
## - I(BPG^2)                     1 2.1302e+13 9.3896e+15 13650
## - SPG:TrustedYes               1 2.6679e+13 9.3949e+15 13650
## - TOPG:SPG                     1 2.8869e+13 9.3971e+15 13650
## - SPG:AgressivenessYes         1 3.0909e+13 9.3992e+15 13650
## - RPG:TrustedYes               1 3.1770e+13 9.4000e+15 13650
## - BPG:TrustedYes               1 3.5602e+13 9.4039e+15 13650
## - RPG:SPG                      1 3.6403e+13 9.4047e+15 13650
## - BPG:AgressivenessYes         1 3.7899e+13 9.4062e+15 13650
## - MPG:BPG                      1 4.1339e+13 9.4096e+15 13651
## <none>                                      9.3683e+15 13651
## - APG:AgressivenessYes         1 4.3336e+13 9.4116e+15 13651
## - I(APG^2)                     1 4.8333e+13 9.4166e+15 13651
## - BPG:SPG                      1 4.8857e+13 9.4171e+15 13651
## - APG:TrustedYes               1 5.3774e+13 9.4220e+15 13651
## - MPG:SPG                      1 7.8377e+13 9.4466e+15 13652
## + APG:BPG                      1 1.4991e+12 9.3668e+15 13653
## + RPG:TOPG                     1 1.0601e+12 9.3672e+15 13653
## + APG:TOPG                     1 5.7929e+11 9.3677e+15 13653
## + TOPG:TrustedYes              1 4.0676e+11 9.3679e+15 13653
## + PPG:APG                      1 9.2338e+10 9.3682e+15 13653
## + MPG:RPG                      1 8.9598e+10 9.3682e+15 13653
## + MPG:PPG                      1 7.3764e+09 9.3683e+15 13653
## + APG:RPG                      1 5.5695e+08 9.3683e+15 13653
## + PPG:TrustedYes               1 4.5983e+08 9.3683e+15 13653
## - PPG:TOPG                     1 8.6653e+13 9.4549e+15 13653
## - APG:SPG                      1 8.8853e+13 9.4571e+15 13653
## - TOPG:BPG                     1 9.8147e+13 9.4664e+15 13653
## - PPG:AgressivenessYes         1 1.2851e+14 9.4968e+15 13655
## - PPG:SPG                      1 1.3195e+14 9.5002e+15 13655
## - PPG:BPG                      1 1.3506e+14 9.5033e+15 13655
## - MPG:APG                      1 2.4834e+14 9.6166e+15 13660
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13648.75
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(PPG^2) + I(APG^2) + I(RPG^2) + I(TOPG^2) +
##     I(BPG^2) + I(SPG^2) + MPG:APG + MPG:TOPG + MPG:BPG + MPG:SPG +
##     MPG:TrustedYes + MPG:AgressivenessYes + PPG:RPG + PPG:TOPG +
##     PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:TrustedYes +
##     APG:AgressivenessYes + RPG:BPG + RPG:SPG + RPG:TrustedYes +
##     RPG:AgressivenessYes + TOPG:BPG + TOPG:SPG + TOPG:AgressivenessYes +
##     BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes + SPG:TrustedYes +
##     SPG:AgressivenessYes + TrustedYes:AgressivenessYes
##
##                              Df  Sum of Sq        RSS   AIC
## - TOPG:AgressivenessYes       1 1.3533e+12 9.3711e+15 13647
## - RPG:AgressivenessYes        1 3.6304e+12 9.3734e+15 13647
## - I(RPG^2)                    1 3.8098e+12 9.3736e+15 13647
## - RPG:BPG                     1 6.3122e+12 9.3761e+15 13647
## - MPG:TOPG                    1 7.4372e+12 9.3772e+15 13647
## - I(SPG^2)                    1 8.6212e+12 9.3784e+15 13647
## - PPG:RPG                     1 1.0768e+13 9.3805e+15 13647
## - I(TOPG^2)                   1 1.1215e+13 9.3810e+15 13647
## - I(PPG^2)                    1 1.3031e+13 9.3828e+15 13647
## - TrustedYes:AgressivenessYes 1 1.8430e+13 9.3882e+15 13648
## - I(BPG^2)                    1 2.1020e+13 9.3908e+15 13648
## - MPG:AgressivenessYes        1 2.1320e+13 9.3911e+15 13648
## - SPG:TrustedYes              1 2.5341e+13 9.3951e+15 13648
## - TOPG:SPG                    1 2.7912e+13 9.3977e+15 13648
## - SPG:AgressivenessYes        1 3.1907e+13 9.4017e+15 13648
## - RPG:TrustedYes              1 3.2099e+13 9.4019e+15 13648
## - BPG:TrustedYes              1 3.4698e+13 9.4045e+15 13648
## - RPG:SPG                     1 3.7390e+13 9.4072e+15 13648
## - BPG:AgressivenessYes        1 3.7793e+13 9.4076e+15 13648
## <none>                                     9.3698e+15 13649
## - MPG:TrustedYes              1 4.2526e+13 9.4123e+15 13649
## - APG:AgressivenessYes        1 4.3139e+13 9.4129e+15 13649
## - MPG:BPG                     1 4.4722e+13 9.4145e+15 13649
## - I(APG^2)                    1 4.8087e+13 9.4179e+15 13649
## - BPG:SPG                     1 5.2863e+13 9.4226e+15 13649
## - APG:TrustedYes              1 5.4718e+13 9.4245e+15 13649
## + I(MPG^2)                    1 1.5141e+12 9.3683e+15 13651
## + APG:BPG                     1 1.2534e+12 9.3685e+15 13651
## + RPG:TOPG                    1 1.1914e+12 9.3686e+15 13651
## + TOPG:TrustedYes             1 6.4223e+11 9.3691e+15 13651
## + MPG:RPG                     1 4.7163e+11 9.3693e+15 13651
## + APG:TOPG                    1 4.4924e+11 9.3693e+15 13651
## + MPG:PPG                     1 3.3284e+11 9.3694e+15 13651
## - MPG:SPG                     1 8.4875e+13 9.4546e+15 13651
## + PPG:APG                     1 1.2261e+11 9.3697e+15 13651
## + PPG:TrustedYes              1 2.0228e+10 9.3698e+15 13651
## + APG:RPG                     1 1.8100e+05 9.3698e+15 13651
## - APG:SPG                     1 8.8782e+13 9.4586e+15 13651
## - PPG:TOPG                    1 9.6768e+13 9.4665e+15 13651
## - TOPG:BPG                    1 1.0529e+14 9.4751e+15 13652
## - PPG:AgressivenessYes        1 1.3019e+14 9.5000e+15 13653
## - PPG:SPG                     1 1.3053e+14 9.5003e+15 13653
## - PPG:BPG                     1 1.4509e+14 9.5149e+15 13654
## - MPG:APG                     1 2.4990e+14 9.6197e+15 13658
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13646.81
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(PPG^2) + I(APG^2) + I(RPG^2) + I(TOPG^2) +
##     I(BPG^2) + I(SPG^2) + MPG:APG + MPG:TOPG + MPG:BPG + MPG:SPG +
##     MPG:TrustedYes + MPG:AgressivenessYes + PPG:RPG + PPG:TOPG +
##     PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:TrustedYes +
##     APG:AgressivenessYes + RPG:BPG + RPG:SPG + RPG:TrustedYes +
##     RPG:AgressivenessYes + TOPG:BPG + TOPG:SPG + BPG:SPG + BPG:TrustedYes +
##     BPG:AgressivenessYes + SPG:TrustedYes + SPG:AgressivenessYes +
##     TrustedYes:AgressivenessYes
##
##                              Df  Sum of Sq        RSS   AIC
## - RPG:AgressivenessYes         1 3.4497e+12 9.3746e+15 13645
## - I(RPG^2)                     1 4.2658e+12 9.3754e+15 13645
## - RPG:BPG                      1 5.9588e+12 9.3771e+15 13645
## - MPG:TOPG                     1 6.2668e+12 9.3774e+15 13645
## - I(SPG^2)                     1 9.3737e+12 9.3805e+15 13645
## - PPG:RPG                      1 9.8940e+12 9.3810e+15 13645
## - I(PPG^2)                     1 1.2960e+13 9.3841e+15 13645
## - TrustedYes:AgressivenessYes  1 1.8938e+13 9.3901e+15 13646
## - I(TOPG^2)                    1 1.9499e+13 9.3906e+15 13646
## - I(BPG^2)                     1 2.0131e+13 9.3913e+15 13646
## - TOPG:SPG                     1 2.6746e+13 9.3979e+15 13646
## - SPG:TrustedYes               1 2.6788e+13 9.3979e+15 13646
## - MPG:AgressivenessYes         1 2.9446e+13 9.4006e+15 13646
## - SPG:AgressivenessYes         1 3.1236e+13 9.4024e+15 13646
## - RPG:TrustedYes               1 3.1269e+13 9.4024e+15 13646
## - BPG:TrustedYes               1 3.4456e+13 9.4056e+15 13646
## - BPG:AgressivenessYes         1 3.6440e+13 9.4076e+15 13646
## - RPG:SPG                      1 3.7601e+13 9.4087e+15 13647
## <none>                                      9.3711e+15 13647
## - MPG:BPG                      1 4.3380e+13 9.4145e+15 13647
## - MPG:TrustedYes               1 4.3466e+13 9.4146e+15 13647
## - APG:AgressivenessYes         1 4.6460e+13 9.4176e+15 13647
## - I(APG^2)                     1 4.7619e+13 9.4187e+15 13647
## - BPG:SPG                      1 5.2092e+13 9.4232e+15 13647
## - APG:TrustedYes               1 5.3704e+13 9.4248e+15 13647
## + RPG:TOPG                     1 1.3913e+12 9.3697e+15 13649
## + TOPG:AgressivenessYes        1 1.3533e+12 9.3698e+15 13649
## + I(MPG^2)                     1 1.0770e+12 9.3700e+15 13649
## + APG:BPG                      1 8.8401e+11 9.3702e+15 13649
## + MPG:RPG                      1 4.2754e+11 9.3707e+15 13649
## + PPG:APG                      1 1.6515e+11 9.3710e+15 13649
## + MPG:PPG                      1 9.8102e+10 9.3710e+15 13649
## + APG:TOPG                     1 7.4171e+10 9.3711e+15 13649
## + APG:RPG                      1 6.1227e+10 9.3711e+15 13649
## + TOPG:TrustedYes              1 2.7378e+10 9.3711e+15 13649
## + PPG:TrustedYes               1 1.7689e+10 9.3711e+15 13649
## - APG:SPG                      1 8.7457e+13 9.4586e+15 13649
## - MPG:SPG                      1 8.8666e+13 9.4598e+15 13649
## - TOPG:BPG                     1 1.0504e+14 9.4762e+15 13650
## - PPG:TOPG                     1 1.1293e+14 9.4841e+15 13650
## - PPG:SPG                      1 1.3310e+14 9.5042e+15 13651
## - PPG:BPG                      1 1.4374e+14 9.5149e+15 13652
## - PPG:AgressivenessYes         1 1.7402e+14 9.5451e+15 13653
## - MPG:APG                      1 2.5383e+14 9.6250e+15 13657
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13644.98
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(PPG^2) + I(APG^2) + I(RPG^2) + I(TOPG^2) +
##     I(BPG^2) + I(SPG^2) + MPG:APG + MPG:TOPG + MPG:BPG + MPG:SPG +
##     MPG:TrustedYes + MPG:AgressivenessYes + PPG:RPG + PPG:TOPG +
##     PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:TrustedYes +
##     APG:AgressivenessYes + RPG:BPG + RPG:SPG + RPG:TrustedYes +
##     TOPG:BPG + TOPG:SPG + BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes +
##     SPG:TrustedYes + SPG:AgressivenessYes + TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - I(RPG^2)                     1 2.6680e+12 9.3772e+15 13643
## - MPG:TOPG                     1 7.4564e+12 9.3820e+15 13643
## - RPG:BPG                      1 7.9417e+12 9.3825e+15 13643
## - PPG:RPG                      1 7.9681e+12 9.3825e+15 13643
## - I(SPG^2)                     1 9.8488e+12 9.3844e+15 13643
## - I(PPG^2)                     1 1.3526e+13 9.3881e+15 13644
## - TrustedYes:AgressivenessYes  1 1.9321e+13 9.3939e+15 13644
## - I(TOPG^2)                    1 1.9429e+13 9.3940e+15 13644
## - I(BPG^2)                     1 2.2191e+13 9.3968e+15 13644
## - SPG:TrustedYes               1 2.6436e+13 9.4010e+15 13644
## - TOPG:SPG                     1 2.7370e+13 9.4019e+15 13644
## - SPG:AgressivenessYes         1 3.2144e+13 9.4067e+15 13644
## - MPG:AgressivenessYes         1 3.3631e+13 9.4082e+15 13645
## - RPG:SPG                      1 3.6862e+13 9.4114e+15 13645
## - BPG:TrustedYes               1 4.0573e+13 9.4151e+15 13645
## <none>                                      9.3746e+15 13645
## - APG:AgressivenessYes         1 4.3411e+13 9.4180e+15 13645
## - MPG:BPG                      1 4.4999e+13 9.4196e+15 13645
## - RPG:TrustedYes               1 4.6602e+13 9.4212e+15 13645
## - I(APG^2)                     1 4.6857e+13 9.4214e+15 13645
## - MPG:TrustedYes               1 5.1721e+13 9.4263e+15 13645
## - BPG:SPG                      1 5.3605e+13 9.4282e+15 13646
## - APG:TrustedYes               1 5.4772e+13 9.4293e+15 13646
## + RPG:AgressivenessYes         1 3.4497e+12 9.3711e+15 13647
## + RPG:TOPG                     1 3.4404e+12 9.3711e+15 13647
## - BPG:AgressivenessYes         1 8.2293e+13 9.4569e+15 13647
## + I(MPG^2)                     1 1.2531e+12 9.3733e+15 13647
## + TOPG:AgressivenessYes        1 1.1725e+12 9.3734e+15 13647
## + APG:BPG                      1 7.7346e+11 9.3738e+15 13647
## + MPG:RPG                      1 5.5748e+11 9.3740e+15 13647
## + APG:RPG                      1 4.6423e+11 9.3741e+15 13647
## + MPG:PPG                      1 3.4212e+11 9.3742e+15 13647
## + PPG:APG                      1 1.7539e+11 9.3744e+15 13647
## + APG:TOPG                     1 7.6574e+10 9.3745e+15 13647
## + TOPG:TrustedYes              1 4.4791e+10 9.3745e+15 13647
## + PPG:TrustedYes               1 2.2501e+10 9.3746e+15 13647
## - MPG:SPG                      1 8.7412e+13 9.4620e+15 13647
## - APG:SPG                      1 8.8936e+13 9.4635e+15 13647
## - TOPG:BPG                     1 1.0595e+14 9.4805e+15 13648
## - PPG:TOPG                     1 1.1076e+14 9.4853e+15 13648
## - PPG:SPG                      1 1.3146e+14 9.5060e+15 13649
## - PPG:BPG                      1 1.4037e+14 9.5149e+15 13650
## - PPG:AgressivenessYes         1 1.7082e+14 9.5454e+15 13651
## - MPG:APG                      1 2.5329e+14 9.6279e+15 13655
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13643.1
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(PPG^2) + I(APG^2) + I(TOPG^2) + I(BPG^2) +
##     I(SPG^2) + MPG:APG + MPG:TOPG + MPG:BPG + MPG:SPG + MPG:TrustedYes +
##     MPG:AgressivenessYes + PPG:RPG + PPG:TOPG + PPG:BPG + PPG:SPG +
##     PPG:AgressivenessYes + APG:SPG + APG:TrustedYes + APG:AgressivenessYes +
##     RPG:BPG + RPG:SPG + RPG:TrustedYes + TOPG:BPG + TOPG:SPG +
##     BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes + SPG:TrustedYes +
##     SPG:AgressivenessYes + TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - MPG:TOPG                     1 7.4697e+12 9.3847e+15 13642
## - I(SPG^2)                     1 9.5675e+12 9.3868e+15 13642
## - PPG:RPG                      1 9.7569e+12 9.3870e+15 13642
## - I(PPG^2)                     1 1.2521e+13 9.3898e+15 13642
## - I(TOPG^2)                    1 1.8271e+13 9.3955e+15 13642
## - TrustedYes:AgressivenessYes  1 1.8916e+13 9.3962e+15 13642
## - SPG:TrustedYes               1 2.6411e+13 9.4037e+15 13642
## - TOPG:SPG                     1 2.7636e+13 9.4049e+15 13642
## - SPG:AgressivenessYes         1 3.2569e+13 9.4098e+15 13643
## - MPG:AgressivenessYes         1 3.2866e+13 9.4101e+15 13643
## - I(BPG^2)                     1 3.4654e+13 9.4119e+15 13643
## - BPG:TrustedYes               1 3.8176e+13 9.4154e+15 13643
## - RPG:BPG                      1 3.9598e+13 9.4168e+15 13643
## - RPG:SPG                      1 4.0604e+13 9.4178e+15 13643
## <none>                                      9.3772e+15 13643
## - APG:AgressivenessYes         1 4.4270e+13 9.4215e+15 13643
## - MPG:BPG                      1 4.4416e+13 9.4217e+15 13643
## - RPG:TrustedYes               1 4.4826e+13 9.4221e+15 13643
## - I(APG^2)                     1 4.9884e+13 9.4271e+15 13643
## - MPG:TrustedYes               1 5.3431e+13 9.4307e+15 13644
## - APG:TrustedYes               1 5.3914e+13 9.4312e+15 13644
## - BPG:SPG                      1 5.4901e+13 9.4321e+15 13644
## + RPG:TOPG                     1 2.9549e+12 9.3743e+15 13645
## + I(RPG^2)                     1 2.6680e+12 9.3746e+15 13645
## - BPG:AgressivenessYes         1 8.2840e+13 9.4601e+15 13645
## + RPG:AgressivenessYes         1 1.8519e+12 9.3754e+15 13645
## + TOPG:AgressivenessYes        1 1.5585e+12 9.3757e+15 13645
## + APG:RPG                      1 1.4326e+12 9.3758e+15 13645
## + MPG:RPG                      1 1.0460e+12 9.3762e+15 13645
## + I(MPG^2)                     1 9.6762e+11 9.3763e+15 13645
## + APG:BPG                      1 5.7952e+11 9.3767e+15 13645
## + MPG:PPG                      1 1.5123e+11 9.3771e+15 13645
## + PPG:APG                      1 8.1530e+10 9.3772e+15 13645
## + PPG:TrustedYes               1 3.2442e+10 9.3772e+15 13645
## + APG:TOPG                     1 2.7519e+10 9.3772e+15 13645
## + TOPG:TrustedYes              1 1.5699e+10 9.3772e+15 13645
## - APG:SPG                      1 8.8882e+13 9.4661e+15 13645
## - MPG:SPG                      1 8.8936e+13 9.4662e+15 13645
## - PPG:TOPG                     1 1.0848e+14 9.4857e+15 13646
## - TOPG:BPG                     1 1.1064e+14 9.4879e+15 13646
## - PPG:SPG                      1 1.3076e+14 9.5080e+15 13647
## - PPG:BPG                      1 1.4755e+14 9.5248e+15 13648
## - PPG:AgressivenessYes         1 1.7077e+14 9.5480e+15 13649
## - MPG:APG                      1 2.5064e+14 9.6279e+15 13653
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13641.45
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(PPG^2) + I(APG^2) + I(TOPG^2) + I(BPG^2) +
##     I(SPG^2) + MPG:APG + MPG:BPG + MPG:SPG + MPG:TrustedYes +
##     MPG:AgressivenessYes + PPG:RPG + PPG:TOPG + PPG:BPG + PPG:SPG +
##     PPG:AgressivenessYes + APG:SPG + APG:TrustedYes + APG:AgressivenessYes +
##     RPG:BPG + RPG:SPG + RPG:TrustedYes + TOPG:BPG + TOPG:SPG +
##     BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes + SPG:TrustedYes +
##     SPG:AgressivenessYes + TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - I(SPG^2)                     1 8.9867e+12 9.3937e+15 13640
## - PPG:RPG                      1 1.1763e+13 9.3965e+15 13640
## - I(PPG^2)                     1 1.2801e+13 9.3975e+15 13640
## - I(TOPG^2)                    1 1.6978e+13 9.4017e+15 13640
## - TOPG:SPG                     1 2.0354e+13 9.4051e+15 13640
## - TrustedYes:AgressivenessYes  1 2.2491e+13 9.4072e+15 13640
## - SPG:AgressivenessYes         1 2.8618e+13 9.4133e+15 13641
## - I(BPG^2)                     1 3.0667e+13 9.4154e+15 13641
## - SPG:TrustedYes               1 3.1205e+13 9.4159e+15 13641
## - RPG:BPG                      1 3.5295e+13 9.4200e+15 13641
## - MPG:BPG                      1 3.8695e+13 9.4234e+15 13641
## <none>                                      9.3847e+15 13642
## - APG:AgressivenessYes         1 4.4334e+13 9.4290e+15 13642
## - RPG:SPG                      1 4.5230e+13 9.4299e+15 13642
## - BPG:TrustedYes               1 4.5307e+13 9.4300e+15 13642
## - MPG:TrustedYes               1 5.0310e+13 9.4350e+15 13642
## - BPG:SPG                      1 5.1866e+13 9.4366e+15 13642
## - RPG:TrustedYes               1 5.1989e+13 9.4367e+15 13642
## - MPG:AgressivenessYes         1 5.3346e+13 9.4381e+15 13642
## - APG:TrustedYes               1 5.5797e+13 9.4405e+15 13642
## - I(APG^2)                     1 5.8970e+13 9.4437e+15 13642
## + MPG:TOPG                     1 7.4697e+12 9.3772e+15 13643
## - BPG:AgressivenessYes         1 7.8166e+13 9.4629e+15 13643
## + TOPG:TrustedYes              1 4.0562e+12 9.3807e+15 13643
## - APG:SPG                      1 8.1741e+13 9.4665e+15 13643
## + RPG:TOPG                     1 3.4888e+12 9.3812e+15 13643
## + RPG:AgressivenessYes         1 2.7213e+12 9.3820e+15 13643
## + I(RPG^2)                     1 2.6813e+12 9.3820e+15 13643
## + APG:BPG                      1 1.5068e+12 9.3832e+15 13643
## + PPG:APG                      1 1.4966e+12 9.3832e+15 13643
## + APG:TOPG                     1 1.1849e+12 9.3835e+15 13643
## + MPG:PPG                      1 1.1712e+12 9.3835e+15 13643
## + MPG:RPG                      1 4.2411e+11 9.3843e+15 13643
## + APG:RPG                      1 3.2617e+11 9.3844e+15 13643
## + TOPG:AgressivenessYes        1 2.4440e+11 9.3845e+15 13643
## + PPG:TrustedYes               1 3.5003e+10 9.3847e+15 13644
## + I(MPG^2)                     1 1.2404e+10 9.3847e+15 13644
## - MPG:SPG                      1 1.0616e+14 9.4909e+15 13644
## - TOPG:BPG                     1 1.1080e+14 9.4955e+15 13645
## - PPG:TOPG                     1 1.3397e+14 9.5187e+15 13646
## - PPG:BPG                      1 1.5455e+14 9.5393e+15 13647
## - PPG:SPG                      1 1.7419e+14 9.5589e+15 13648
## - PPG:AgressivenessYes         1 1.7773e+14 9.5624e+15 13648
## - MPG:APG                      1 2.5526e+14 9.6400e+15 13651
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13639.88
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(PPG^2) + I(APG^2) + I(TOPG^2) + I(BPG^2) +
##     MPG:APG + MPG:BPG + MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes +
##     PPG:RPG + PPG:TOPG + PPG:BPG + PPG:SPG + PPG:AgressivenessYes +
##     APG:SPG + APG:TrustedYes + APG:AgressivenessYes + RPG:BPG +
##     RPG:SPG + RPG:TrustedYes + TOPG:BPG + TOPG:SPG + BPG:SPG +
##     BPG:TrustedYes + BPG:AgressivenessYes + SPG:TrustedYes +
##     SPG:AgressivenessYes + TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - PPG:RPG                      1 1.0154e+13 9.4039e+15 13638
## - I(PPG^2)                     1 1.3888e+13 9.4076e+15 13638
## - TOPG:SPG                     1 1.7597e+13 9.4113e+15 13639
## - I(TOPG^2)                    1 1.9390e+13 9.4131e+15 13639
## - TrustedYes:AgressivenessYes  1 2.1834e+13 9.4155e+15 13639
## - SPG:AgressivenessYes         1 2.6160e+13 9.4199e+15 13639
## - SPG:TrustedYes               1 2.9286e+13 9.4230e+15 13639
## - I(BPG^2)                     1 2.9800e+13 9.4235e+15 13639
## - RPG:BPG                      1 3.4713e+13 9.4284e+15 13640
## - MPG:BPG                      1 3.8838e+13 9.4325e+15 13640
## - APG:AgressivenessYes         1 4.2060e+13 9.4358e+15 13640
## <none>                                      9.3937e+15 13640
## - MPG:TrustedYes               1 4.4227e+13 9.4379e+15 13640
## - BPG:TrustedYes               1 4.4261e+13 9.4380e+15 13640
## - RPG:SPG                      1 4.9411e+13 9.4431e+15 13640
## - RPG:TrustedYes               1 4.9504e+13 9.4432e+15 13640
## - BPG:SPG                      1 4.9869e+13 9.4436e+15 13640
## - MPG:AgressivenessYes         1 5.4407e+13 9.4481e+15 13640
## - APG:TrustedYes               1 5.4503e+13 9.4482e+15 13640
## - I(APG^2)                     1 5.9699e+13 9.4534e+15 13641
## - APG:SPG                      1 7.2757e+13 9.4665e+15 13641
## + I(SPG^2)                     1 8.9867e+12 9.3847e+15 13642
## - BPG:AgressivenessYes         1 7.7805e+13 9.4715e+15 13642
## + MPG:TOPG                     1 6.8890e+12 9.3868e+15 13642
## + RPG:TOPG                     1 4.4620e+12 9.3892e+15 13642
## + RPG:AgressivenessYes         1 3.1463e+12 9.3906e+15 13642
## + TOPG:TrustedYes              1 2.8438e+12 9.3909e+15 13642
## + I(RPG^2)                     1 2.4070e+12 9.3913e+15 13642
## + PPG:APG                      1 2.2922e+12 9.3914e+15 13642
## + APG:BPG                      1 2.0728e+12 9.3916e+15 13642
## + MPG:PPG                      1 1.6738e+12 9.3920e+15 13642
## + APG:TOPG                     1 8.4807e+11 9.3929e+15 13642
## + MPG:RPG                      1 5.2573e+11 9.3932e+15 13642
## + APG:RPG                      1 4.5841e+11 9.3932e+15 13642
## + I(MPG^2)                     1 1.4959e+11 9.3936e+15 13642
## + TOPG:AgressivenessYes        1 4.3040e+10 9.3937e+15 13642
## + PPG:TrustedYes               1 2.1334e+09 9.3937e+15 13642
## - MPG:SPG                      1 1.0539e+14 9.4991e+15 13643
## - TOPG:BPG                     1 1.0616e+14 9.4999e+15 13643
## - PPG:TOPG                     1 1.3888e+14 9.5326e+15 13644
## - PPG:BPG                      1 1.5303e+14 9.5467e+15 13645
## - PPG:SPG                      1 1.7044e+14 9.5641e+15 13646
## - PPG:AgressivenessYes         1 1.8467e+14 9.5784e+15 13646
## - MPG:APG                      1 2.4695e+14 9.6406e+15 13649
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13638.36
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(PPG^2) + I(APG^2) + I(TOPG^2) + I(BPG^2) +
##     MPG:APG + MPG:BPG + MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes +
##     PPG:TOPG + PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG +
##     APG:TrustedYes + APG:AgressivenessYes + RPG:BPG + RPG:SPG +
##     RPG:TrustedYes + TOPG:BPG + TOPG:SPG + BPG:SPG + BPG:TrustedYes +
##     BPG:AgressivenessYes + SPG:TrustedYes + SPG:AgressivenessYes +
##     TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - I(PPG^2)                     1 1.3584e+13 9.4174e+15 13637
## - TOPG:SPG                     1 1.5225e+13 9.4191e+15 13637
## - TrustedYes:AgressivenessYes  1 1.9003e+13 9.4229e+15 13637
## - I(TOPG^2)                    1 1.9130e+13 9.4230e+15 13637
## - SPG:AgressivenessYes         1 2.5922e+13 9.4298e+15 13638
## - SPG:TrustedYes               1 2.6378e+13 9.4302e+15 13638
## - BPG:TrustedYes               1 3.5508e+13 9.4394e+15 13638
## - I(BPG^2)                     1 3.7136e+13 9.4410e+15 13638
## - RPG:TrustedYes               1 3.9366e+13 9.4432e+15 13638
## - RPG:BPG                      1 4.0331e+13 9.4442e+15 13638
## <none>                                      9.4039e+15 13638
## - MPG:BPG                      1 4.2767e+13 9.4466e+15 13638
## - APG:AgressivenessYes         1 4.3393e+13 9.4472e+15 13638
## - MPG:TrustedYes               1 4.6621e+13 9.4505e+15 13638
## - APG:TrustedYes               1 5.0702e+13 9.4546e+15 13639
## - MPG:AgressivenessYes         1 5.1574e+13 9.4554e+15 13639
## - I(APG^2)                     1 5.6223e+13 9.4601e+15 13639
## - BPG:SPG                      1 5.6781e+13 9.4606e+15 13639
## - APG:SPG                      1 6.6475e+13 9.4703e+15 13640
## - RPG:SPG                      1 6.8541e+13 9.4724e+15 13640
## + PPG:RPG                      1 1.0154e+13 9.3937e+15 13640
## + MPG:TOPG                     1 8.7293e+12 9.3951e+15 13640
## + I(SPG^2)                     1 7.3778e+12 9.3965e+15 13640
## + MPG:RPG                      1 6.4777e+12 9.3974e+15 13640
## + TOPG:TrustedYes              1 4.6333e+12 9.3992e+15 13640
## + I(RPG^2)                     1 4.2090e+12 9.3996e+15 13640
## - BPG:AgressivenessYes         1 8.2144e+13 9.4860e+15 13640
## + APG:BPG                      1 1.8891e+12 9.4020e+15 13640
## + PPG:APG                      1 1.2012e+12 9.4027e+15 13640
## + APG:TOPG                     1 9.6824e+11 9.4029e+15 13640
## + RPG:AgressivenessYes         1 9.5276e+11 9.4029e+15 13640
## + APG:RPG                      1 8.0959e+11 9.4030e+15 13640
## + MPG:PPG                      1 7.3736e+11 9.4031e+15 13640
## + TOPG:AgressivenessYes        1 5.5264e+11 9.4033e+15 13640
## + RPG:TOPG                     1 4.2361e+11 9.4034e+15 13640
## + PPG:TrustedYes               1 7.7355e+10 9.4038e+15 13640
## + I(MPG^2)                     1 2.5718e+10 9.4038e+15 13640
## - TOPG:BPG                     1 1.0204e+14 9.5059e+15 13641
## - MPG:SPG                      1 1.0866e+14 9.5125e+15 13641
## - PPG:TOPG                     1 1.2970e+14 9.5336e+15 13642
## - PPG:BPG                      1 1.5848e+14 9.5623e+15 13644
## - PPG:SPG                      1 1.7050e+14 9.5744e+15 13644
## - PPG:AgressivenessYes         1 1.8071e+14 9.5846e+15 13645
## - MPG:APG                      1 2.3855e+14 9.6424e+15 13647
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13636.99
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(APG^2) + I(TOPG^2) + I(BPG^2) + MPG:APG +
##     MPG:BPG + MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes +
##     PPG:TOPG + PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG +
##     APG:TrustedYes + APG:AgressivenessYes + RPG:BPG + RPG:SPG +
##     RPG:TrustedYes + TOPG:BPG + TOPG:SPG + BPG:SPG + BPG:TrustedYes +
##     BPG:AgressivenessYes + SPG:TrustedYes + SPG:AgressivenessYes +
##     TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - I(TOPG^2)                    1 8.6971e+12 9.4261e+15 13635
## - TOPG:SPG                     1 9.3615e+12 9.4268e+15 13635
## - TrustedYes:AgressivenessYes  1 1.6538e+13 9.4340e+15 13636
## - SPG:AgressivenessYes         1 2.2735e+13 9.4402e+15 13636
## - SPG:TrustedYes               1 2.4066e+13 9.4415e+15 13636
## - BPG:TrustedYes               1 3.4595e+13 9.4520e+15 13637
## - RPG:TrustedYes               1 3.4677e+13 9.4521e+15 13637
## - I(BPG^2)                     1 3.5401e+13 9.4528e+15 13637
## - MPG:BPG                      1 3.5757e+13 9.4532e+15 13637
## - APG:AgressivenessYes         1 3.9189e+13 9.4566e+15 13637
## - RPG:BPG                      1 4.0363e+13 9.4578e+15 13637
## <none>                                      9.4174e+15 13637
## - BPG:SPG                      1 4.8450e+13 9.4659e+15 13637
## - APG:TrustedYes               1 4.9277e+13 9.4667e+15 13637
## - MPG:AgressivenessYes         1 5.4159e+13 9.4716e+15 13638
## - I(APG^2)                     1 5.6316e+13 9.4738e+15 13638
## - APG:SPG                      1 5.7806e+13 9.4752e+15 13638
## - MPG:TrustedYes               1 6.4012e+13 9.4815e+15 13638
## - RPG:SPG                      1 6.4351e+13 9.4818e+15 13638
## + I(PPG^2)                     1 1.3584e+13 9.4039e+15 13638
## + PPG:RPG                      1 9.8506e+12 9.4076e+15 13638
## + MPG:TOPG                     1 8.9714e+12 9.4085e+15 13639
## - BPG:AgressivenessYes         1 7.7034e+13 9.4945e+15 13639
## + I(SPG^2)                     1 8.3750e+12 9.4091e+15 13639
## + MPG:RPG                      1 6.1670e+12 9.4113e+15 13639
## + TOPG:TrustedYes              1 5.7008e+12 9.4117e+15 13639
## + MPG:PPG                      1 4.4749e+12 9.4130e+15 13639
## + I(RPG^2)                     1 2.8331e+12 9.4146e+15 13639
## + APG:BPG                      1 2.2905e+12 9.4151e+15 13639
## + APG:TOPG                     1 2.0427e+12 9.4154e+15 13639
## + PPG:TrustedYes               1 1.8596e+12 9.4156e+15 13639
## + RPG:AgressivenessYes         1 1.5035e+12 9.4159e+15 13639
## + APG:RPG                      1 9.7131e+11 9.4165e+15 13639
## + PPG:APG                      1 7.3870e+11 9.4167e+15 13639
## + TOPG:AgressivenessYes        1 6.5417e+11 9.4168e+15 13639
## + I(MPG^2)                     1 6.5016e+11 9.4168e+15 13639
## + RPG:TOPG                     1 2.3035e+11 9.4172e+15 13639
## - TOPG:BPG                     1 9.1419e+13 9.5089e+15 13639
## - MPG:SPG                      1 1.0795e+14 9.5254e+15 13640
## - PPG:BPG                      1 1.4584e+14 9.5633e+15 13642
## - PPG:TOPG                     1 1.4671e+14 9.5641e+15 13642
## - PPG:AgressivenessYes         1 1.8491e+14 9.6024e+15 13644
## - PPG:SPG                      1 1.8553e+14 9.6030e+15 13644
## - MPG:APG                      1 2.2637e+14 9.6438e+15 13646
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13635.4
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(APG^2) + I(BPG^2) + MPG:APG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG +
##     PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:TrustedYes +
##     APG:AgressivenessYes + RPG:BPG + RPG:SPG + RPG:TrustedYes +
##     TOPG:BPG + TOPG:SPG + BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes +
##     SPG:TrustedYes + SPG:AgressivenessYes + TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - TOPG:SPG                     1 1.5012e+13 9.4411e+15 13634
## - TrustedYes:AgressivenessYes  1 1.8077e+13 9.4442e+15 13634
## - SPG:TrustedYes               1 2.3028e+13 9.4492e+15 13634
## - SPG:AgressivenessYes         1 2.6291e+13 9.4524e+15 13635
## - I(BPG^2)                     1 3.0514e+13 9.4567e+15 13635
## - MPG:BPG                      1 3.1698e+13 9.4578e+15 13635
## - RPG:TrustedYes               1 3.3352e+13 9.4595e+15 13635
## - RPG:BPG                      1 3.5502e+13 9.4616e+15 13635
## - BPG:TrustedYes               1 3.6919e+13 9.4631e+15 13635
## - APG:AgressivenessYes         1 3.7283e+13 9.4634e+15 13635
## <none>                                      9.4261e+15 13635
## - BPG:SPG                      1 4.6076e+13 9.4722e+15 13636
## - APG:TrustedYes               1 4.7847e+13 9.4740e+15 13636
## - MPG:AgressivenessYes         1 5.0666e+13 9.4768e+15 13636
## - RPG:SPG                      1 6.3013e+13 9.4891e+15 13636
## - MPG:TrustedYes               1 6.4020e+13 9.4902e+15 13636
## - APG:SPG                      1 6.7654e+13 9.4938e+15 13637
## - BPG:AgressivenessYes         1 7.4084e+13 9.5002e+15 13637
## + APG:TOPG                     1 1.0125e+13 9.4160e+15 13637
## + PPG:RPG                      1 9.7857e+12 9.4164e+15 13637
## + I(SPG^2)                     1 9.6484e+12 9.4165e+15 13637
## + I(TOPG^2)                    1 8.6971e+12 9.4174e+15 13637
## + MPG:TOPG                     1 7.5906e+12 9.4185e+15 13637
## + MPG:RPG                      1 6.6445e+12 9.4195e+15 13637
## + APG:BPG                      1 4.5401e+12 9.4216e+15 13637
## + I(PPG^2)                     1 3.1517e+12 9.4230e+15 13637
## + APG:RPG                      1 2.9478e+12 9.4232e+15 13637
## + I(RPG^2)                     1 2.2983e+12 9.4238e+15 13637
## + TOPG:TrustedYes              1 2.0447e+12 9.4241e+15 13637
## + RPG:AgressivenessYes         1 1.3890e+12 9.4247e+15 13637
## + PPG:TrustedYes               1 1.2379e+12 9.4249e+15 13637
## + MPG:PPG                      1 1.0469e+12 9.4251e+15 13637
## + RPG:TOPG                     1 9.9156e+11 9.4251e+15 13637
## + TOPG:AgressivenessYes        1 5.0245e+11 9.4256e+15 13637
## + PPG:APG                      1 2.9068e+11 9.4258e+15 13637
## + I(MPG^2)                     1 2.3176e+11 9.4259e+15 13637
## - I(APG^2)                     1 9.2197e+13 9.5183e+15 13638
## - TOPG:BPG                     1 9.4239e+13 9.5204e+15 13638
## - MPG:SPG                      1 1.0094e+14 9.5271e+15 13638
## - PPG:BPG                      1 1.4670e+14 9.5728e+15 13640
## - PPG:SPG                      1 1.7721e+14 9.6034e+15 13642
## - PPG:AgressivenessYes         1 1.8557e+14 9.6117e+15 13642
## - MPG:APG                      1 2.1799e+14 9.6441e+15 13644
## - PPG:TOPG                     1 2.6228e+14 9.6884e+15 13646
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13634.1
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(APG^2) + I(BPG^2) + MPG:APG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG +
##     PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:TrustedYes +
##     APG:AgressivenessYes + RPG:BPG + RPG:SPG + RPG:TrustedYes +
##     TOPG:BPG + BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes +
##     SPG:TrustedYes + SPG:AgressivenessYes + TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - SPG:AgressivenessYes         1 1.4402e+13 9.4555e+15 13633
## - TrustedYes:AgressivenessYes  1 1.8134e+13 9.4593e+15 13633
## - SPG:TrustedYes               1 2.3015e+13 9.4642e+15 13633
## - RPG:TrustedYes               1 3.2760e+13 9.4739e+15 13634
## - MPG:BPG                      1 3.3229e+13 9.4744e+15 13634
## - BPG:TrustedYes               1 3.4140e+13 9.4753e+15 13634
## - I(BPG^2)                     1 3.4493e+13 9.4756e+15 13634
## - RPG:BPG                      1 3.6872e+13 9.4780e+15 13634
## - APG:AgressivenessYes         1 3.9295e+13 9.4804e+15 13634
## - BPG:SPG                      1 3.9715e+13 9.4809e+15 13634
## <none>                                      9.4411e+15 13634
## - MPG:AgressivenessYes         1 4.5109e+13 9.4863e+15 13634
## - APG:TrustedYes               1 4.8019e+13 9.4892e+15 13634
## - APG:SPG                      1 5.2745e+13 9.4939e+15 13635
## - MPG:TrustedYes               1 6.0150e+13 9.5013e+15 13635
## - RPG:SPG                      1 6.3186e+13 9.5043e+15 13635
## + APG:TOPG                     1 1.6858e+13 9.4243e+15 13635
## + TOPG:SPG                     1 1.5012e+13 9.4261e+15 13635
## + I(TOPG^2)                    1 1.4347e+13 9.4268e+15 13635
## - BPG:AgressivenessYes         1 7.3441e+13 9.5146e+15 13636
## + PPG:RPG                      1 7.3922e+12 9.4338e+15 13636
## + APG:BPG                      1 7.3841e+12 9.4338e+15 13636
## + I(SPG^2)                     1 7.1170e+12 9.4340e+15 13636
## + MPG:RPG                      1 4.7193e+12 9.4364e+15 13636
## + TOPG:AgressivenessYes        1 2.8318e+12 9.4383e+15 13636
## + I(RPG^2)                     1 2.4937e+12 9.4387e+15 13636
## + APG:RPG                      1 1.5702e+12 9.4396e+15 13636
## + RPG:AgressivenessYes         1 1.1306e+12 9.4400e+15 13636
## + RPG:TOPG                     1 7.7978e+11 9.4404e+15 13636
## + PPG:TrustedYes               1 7.0591e+11 9.4404e+15 13636
## + MPG:TOPG                     1 2.5689e+11 9.4409e+15 13636
## + PPG:APG                      1 2.2651e+11 9.4409e+15 13636
## + I(MPG^2)                     1 2.0873e+11 9.4409e+15 13636
## + I(PPG^2)                     1 1.3335e+11 9.4410e+15 13636
## + MPG:PPG                      1 6.8860e+10 9.4411e+15 13636
## + TOPG:TrustedYes              1 4.5390e+10 9.4411e+15 13636
## - TOPG:BPG                     1 9.4280e+13 9.5354e+15 13636
## - MPG:SPG                      1 9.6649e+13 9.5378e+15 13637
## - I(APG^2)                     1 9.7437e+13 9.5386e+15 13637
## - PPG:BPG                      1 1.5195e+14 9.5931e+15 13639
## - PPG:AgressivenessYes         1 1.7062e+14 9.6118e+15 13640
## - PPG:SPG                      1 2.0365e+14 9.6448e+15 13642
## - MPG:APG                      1 2.0611e+14 9.6473e+15 13642
## - PPG:TOPG                     1 2.9848e+14 9.7396e+15 13646
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13632.78
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(APG^2) + I(BPG^2) + MPG:APG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG +
##     PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:TrustedYes +
##     APG:AgressivenessYes + RPG:BPG + RPG:SPG + RPG:TrustedYes +
##     TOPG:BPG + BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes +
##     SPG:TrustedYes + TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - SPG:TrustedYes               1 1.6046e+13 9.4716e+15 13632
## - TrustedYes:AgressivenessYes  1 1.9932e+13 9.4755e+15 13632
## - RPG:TrustedYes               1 2.9881e+13 9.4854e+15 13632
## - BPG:TrustedYes               1 3.2438e+13 9.4880e+15 13632
## - I(BPG^2)                     1 3.3961e+13 9.4895e+15 13632
## - MPG:BPG                      1 3.5868e+13 9.4914e+15 13632
## - RPG:BPG                      1 3.6535e+13 9.4921e+15 13632
## - BPG:SPG                      1 3.9482e+13 9.4950e+15 13633
## - APG:TrustedYes               1 4.2013e+13 9.4976e+15 13633
## <none>                                      9.4555e+15 13633
## - RPG:SPG                      1 5.5233e+13 9.5108e+15 13633
## - APG:AgressivenessYes         1 5.7219e+13 9.5128e+15 13633
## - MPG:TrustedYes               1 6.8356e+13 9.5239e+15 13634
## + APG:TOPG                     1 1.7180e+13 9.4384e+15 13634
## + I(TOPG^2)                    1 1.4408e+13 9.4411e+15 13634
## + SPG:AgressivenessYes         1 1.4402e+13 9.4411e+15 13634
## - BPG:AgressivenessYes         1 7.3733e+13 9.5293e+15 13634
## + APG:BPG                      1 8.7318e+12 9.4468e+15 13634
## + PPG:RPG                      1 8.2999e+12 9.4472e+15 13634
## + I(SPG^2)                     1 6.4103e+12 9.4491e+15 13634
## + MPG:RPG                      1 6.3194e+12 9.4492e+15 13634
## - MPG:AgressivenessYes         1 8.2039e+13 9.5376e+15 13635
## + TOPG:SPG                     1 3.1225e+12 9.4524e+15 13635
## + I(RPG^2)                     1 2.7494e+12 9.4528e+15 13635
## + APG:RPG                      1 2.6404e+12 9.4529e+15 13635
## + TOPG:AgressivenessYes        1 2.3385e+12 9.4532e+15 13635
## + RPG:AgressivenessYes         1 1.3166e+12 9.4542e+15 13635
## + RPG:TOPG                     1 9.6928e+11 9.4546e+15 13635
## + I(MPG^2)                     1 7.0147e+11 9.4548e+15 13635
## + MPG:TOPG                     1 6.8054e+11 9.4549e+15 13635
## + PPG:TrustedYes               1 4.0062e+11 9.4551e+15 13635
## + PPG:APG                      1 2.3185e+11 9.4553e+15 13635
## + I(PPG^2)                     1 2.0990e+11 9.4553e+15 13635
## + TOPG:TrustedYes              1 2.0699e+11 9.4553e+15 13635
## + MPG:PPG                      1 2.0371e+11 9.4553e+15 13635
## - MPG:SPG                      1 8.9027e+13 9.5446e+15 13635
## - TOPG:BPG                     1 9.4830e+13 9.5504e+15 13635
## - APG:SPG                      1 9.8870e+13 9.5544e+15 13635
## - I(APG^2)                     1 1.2518e+14 9.5807e+15 13637
## - PPG:BPG                      1 1.5397e+14 9.6095e+15 13638
## - PPG:SPG                      1 1.8981e+14 9.6454e+15 13640
## - PPG:AgressivenessYes         1 2.0890e+14 9.6644e+15 13640
## - MPG:APG                      1 2.1935e+14 9.6749e+15 13641
## - PPG:TOPG                     1 3.0342e+14 9.7590e+15 13645
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13631.53
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(APG^2) + I(BPG^2) + MPG:APG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG +
##     PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:TrustedYes +
##     APG:AgressivenessYes + RPG:BPG + RPG:SPG + RPG:TrustedYes +
##     TOPG:BPG + BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes +
##     TrustedYes:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - TrustedYes:AgressivenessYes  1 1.9177e+13 9.4908e+15 13630
## - MPG:BPG                      1 2.9605e+13 9.5012e+15 13631
## - APG:TrustedYes               1 3.0399e+13 9.5020e+15 13631
## - RPG:TrustedYes               1 3.3813e+13 9.5054e+15 13631
## - I(BPG^2)                     1 3.4829e+13 9.5064e+15 13631
## - RPG:BPG                      1 3.5759e+13 9.5074e+15 13631
## <none>                                      9.4716e+15 13632
## - BPG:SPG                      1 4.4801e+13 9.5164e+15 13632
## - BPG:TrustedYes               1 4.6437e+13 9.5180e+15 13632
## - APG:AgressivenessYes         1 6.0279e+13 9.5319e+15 13632
## - RPG:SPG                      1 6.3998e+13 9.5356e+15 13632
## - MPG:TrustedYes               1 6.4673e+13 9.5363e+15 13632
## + APG:TOPG                     1 1.9547e+13 9.4520e+15 13633
## + SPG:TrustedYes               1 1.6046e+13 9.4555e+15 13633
## + I(TOPG^2)                    1 1.3255e+13 9.4583e+15 13633
## - BPG:AgressivenessYes         1 7.3173e+13 9.5448e+15 13633
## + APG:BPG                      1 1.1107e+13 9.4605e+15 13633
## + SPG:AgressivenessYes         1 7.4323e+12 9.4642e+15 13633
## - MPG:AgressivenessYes         1 7.9053e+13 9.5506e+15 13633
## - MPG:SPG                      1 7.9087e+13 9.5507e+15 13633
## + PPG:RPG                      1 5.9630e+12 9.4656e+15 13633
## + I(SPG^2)                     1 5.3769e+12 9.4662e+15 13633
## + TOPG:SPG                     1 4.8149e+12 9.4668e+15 13633
## + MPG:RPG                      1 2.9857e+12 9.4686e+15 13633
## + I(RPG^2)                     1 2.5954e+12 9.4690e+15 13633
## + TOPG:AgressivenessYes        1 2.3946e+12 9.4692e+15 13633
## + APG:RPG                      1 2.2360e+12 9.4694e+15 13633
## + MPG:TOPG                     1 1.5167e+12 9.4701e+15 13634
## + RPG:AgressivenessYes         1 1.3951e+12 9.4702e+15 13634
## + RPG:TOPG                     1 5.1049e+11 9.4711e+15 13634
## + I(MPG^2)                     1 5.0903e+11 9.4711e+15 13634
## + PPG:TrustedYes               1 3.9874e+11 9.4712e+15 13634
## + TOPG:TrustedYes              1 1.9127e+11 9.4714e+15 13634
## + I(PPG^2)                     1 8.0207e+10 9.4715e+15 13634
## + MPG:PPG                      1 4.0493e+10 9.4716e+15 13634
## + PPG:APG                      1 3.3653e+10 9.4716e+15 13634
## - APG:SPG                      1 8.9265e+13 9.5609e+15 13634
## - TOPG:BPG                     1 8.9852e+13 9.5614e+15 13634
## - I(APG^2)                     1 1.1967e+14 9.5913e+15 13635
## - PPG:BPG                      1 1.4804e+14 9.6196e+15 13636
## - PPG:SPG                      1 1.7773e+14 9.6493e+15 13638
## - MPG:APG                      1 2.0338e+14 9.6750e+15 13639
## - PPG:AgressivenessYes         1 2.0830e+14 9.6799e+15 13639
## - PPG:TOPG                     1 2.9161e+14 9.7632e+15 13643
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13630.42
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(APG^2) + I(BPG^2) + MPG:APG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG +
##     PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:TrustedYes +
##     APG:AgressivenessYes + RPG:BPG + RPG:SPG + RPG:TrustedYes +
##     TOPG:BPG + BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - APG:TrustedYes               1 1.4432e+13 9.5052e+15 13629
## - MPG:BPG                      1 2.3509e+13 9.5143e+15 13630
## - RPG:TrustedYes               1 2.5607e+13 9.5164e+15 13630
## - I(BPG^2)                     1 3.0619e+13 9.5214e+15 13630
## - RPG:BPG                      1 3.2248e+13 9.5230e+15 13630
## - BPG:SPG                      1 4.0005e+13 9.5308e+15 13630
## <none>                                      9.4908e+15 13630
## - MPG:TrustedYes               1 4.9139e+13 9.5399e+15 13631
## - RPG:SPG                      1 5.6879e+13 9.5477e+15 13631
## - BPG:TrustedYes               1 5.7983e+13 9.5488e+15 13631
## + APG:TOPG                     1 2.4266e+13 9.4665e+15 13631
## - MPG:AgressivenessYes         1 6.4833e+13 9.5556e+15 13631
## + TrustedYes:AgressivenessYes  1 1.9177e+13 9.4716e+15 13632
## + SPG:TrustedYes               1 1.5292e+13 9.4755e+15 13632
## + I(TOPG^2)                    1 1.5178e+13 9.4756e+15 13632
## + APG:BPG                      1 1.5009e+13 9.4758e+15 13632
## - APG:AgressivenessYes         1 7.1690e+13 9.5625e+15 13632
## - MPG:SPG                      1 7.4333e+13 9.5651e+15 13632
## + SPG:AgressivenessYes         1 8.8120e+12 9.4820e+15 13632
## + I(SPG^2)                     1 5.0718e+12 9.4857e+15 13632
## - BPG:AgressivenessYes         1 8.1817e+13 9.5726e+15 13632
## + TOPG:SPG                     1 4.3569e+12 9.4864e+15 13632
## + PPG:RPG                      1 3.9603e+12 9.4868e+15 13632
## + MPG:TOPG                     1 2.7634e+12 9.4880e+15 13632
## + TOPG:AgressivenessYes        1 2.5217e+12 9.4883e+15 13632
## + APG:RPG                      1 2.1348e+12 9.4886e+15 13632
## + RPG:AgressivenessYes         1 2.0987e+12 9.4887e+15 13632
## + I(RPG^2)                     1 2.0583e+12 9.4887e+15 13632
## + PPG:TrustedYes               1 1.9276e+12 9.4888e+15 13632
## + I(MPG^2)                     1 1.5681e+12 9.4892e+15 13632
## + TOPG:TrustedYes              1 1.3145e+12 9.4895e+15 13632
## + MPG:RPG                      1 7.9818e+11 9.4900e+15 13632
## + PPG:APG                      1 6.9926e+11 9.4901e+15 13632
## + MPG:PPG                      1 6.4178e+11 9.4901e+15 13632
## + RPG:TOPG                     1 1.9058e+11 9.4906e+15 13632
## + I(PPG^2)                     1 1.7251e+10 9.4908e+15 13632
## - TOPG:BPG                     1 8.8664e+13 9.5794e+15 13632
## - APG:SPG                      1 9.5817e+13 9.5866e+15 13633
## - I(APG^2)                     1 1.3008e+14 9.6209e+15 13634
## - PPG:BPG                      1 1.4535e+14 9.6361e+15 13635
## - PPG:SPG                      1 1.7703e+14 9.6678e+15 13637
## - MPG:APG                      1 1.8438e+14 9.6752e+15 13637
## - PPG:AgressivenessYes         1 2.0100e+14 9.6918e+15 13638
## - PPG:TOPG                     1 2.8905e+14 9.7798e+15 13642
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13629.09
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(APG^2) + I(BPG^2) + MPG:APG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG +
##     PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:AgressivenessYes +
##     RPG:BPG + RPG:SPG + RPG:TrustedYes + TOPG:BPG + BPG:SPG +
##     BPG:TrustedYes + BPG:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - RPG:TrustedYes               1 2.0596e+13 9.5258e+15 13628
## - MPG:BPG                      1 2.2941e+13 9.5281e+15 13628
## - I(BPG^2)                     1 3.0927e+13 9.5361e+15 13628
## - RPG:BPG                      1 3.2124e+13 9.5373e+15 13629
## - BPG:SPG                      1 3.7257e+13 9.5425e+15 13629
## <none>                                      9.5052e+15 13629
## - RPG:SPG                      1 5.3024e+13 9.5582e+15 13630
## - MPG:TrustedYes               1 5.4952e+13 9.5602e+15 13630
## + APG:TOPG                     1 2.6871e+13 9.4783e+15 13630
## + APG:BPG                      1 1.7806e+13 9.4874e+15 13630
## - BPG:TrustedYes               1 6.8977e+13 9.5742e+15 13630
## + APG:TrustedYes               1 1.4432e+13 9.4908e+15 13630
## + I(TOPG^2)                    1 1.3270e+13 9.4919e+15 13630
## - MPG:AgressivenessYes         1 7.3350e+13 9.5786e+15 13630
## - MPG:SPG                      1 7.5699e+13 9.5809e+15 13631
## + SPG:AgressivenessYes         1 6.5674e+12 9.4986e+15 13631
## + SPG:TrustedYes               1 5.5935e+12 9.4996e+15 13631
## + TOPG:SPG                     1 5.1739e+12 9.5000e+15 13631
## + I(SPG^2)                     1 5.1047e+12 9.5001e+15 13631
## + PPG:RPG                      1 3.6801e+12 9.5015e+15 13631
## + TrustedYes:AgressivenessYes  1 3.2111e+12 9.5020e+15 13631
## + RPG:AgressivenessYes         1 2.5533e+12 9.5027e+15 13631
## + PPG:APG                      1 2.0066e+12 9.5032e+15 13631
## + MPG:TOPG                     1 1.8904e+12 9.5033e+15 13631
## + APG:RPG                      1 1.7628e+12 9.5034e+15 13631
## + I(RPG^2)                     1 1.7020e+12 9.5035e+15 13631
## + TOPG:AgressivenessYes        1 1.4031e+12 9.5038e+15 13631
## + PPG:TrustedYes               1 1.2741e+12 9.5039e+15 13631
## + MPG:RPG                      1 1.0870e+12 9.5041e+15 13631
## + TOPG:TrustedYes              1 1.0840e+12 9.5041e+15 13631
## + I(MPG^2)                     1 1.0038e+11 9.5051e+15 13631
## + RPG:TOPG                     1 3.0586e+10 9.5052e+15 13631
## + I(PPG^2)                     1 9.3618e+09 9.5052e+15 13631
## + MPG:PPG                      1 9.0375e+09 9.5052e+15 13631
## - TOPG:BPG                     1 8.9234e+13 9.5944e+15 13631
## - APG:SPG                      1 9.1849e+13 9.5971e+15 13631
## - BPG:AgressivenessYes         1 9.6423e+13 9.6016e+15 13632
## - APG:AgressivenessYes         1 1.3083e+14 9.6360e+15 13633
## - I(APG^2)                     1 1.3155e+14 9.6368e+15 13633
## - PPG:BPG                      1 1.4868e+14 9.6539e+15 13634
## - MPG:APG                      1 1.7187e+14 9.6771e+15 13635
## - PPG:SPG                      1 1.8236e+14 9.6876e+15 13636
## - PPG:AgressivenessYes         1 1.9657e+14 9.7018e+15 13636
## - PPG:TOPG                     1 2.7771e+14 9.7829e+15 13640
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13628.05
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(APG^2) + I(BPG^2) + MPG:APG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG +
##     PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:AgressivenessYes +
##     RPG:BPG + RPG:SPG + TOPG:BPG + BPG:SPG + BPG:TrustedYes +
##     BPG:AgressivenessYes
##
##                              Df  Sum of Sq        RSS   AIC
## - RPG:BPG                     1 2.4080e+13 9.5499e+15 13627
## - MPG:BPG                     1 2.5603e+13 9.5514e+15 13627
## - I(BPG^2)                    1 2.6794e+13 9.5526e+15 13627
## - BPG:SPG                     1 2.8418e+13 9.5542e+15 13627
## - RPG:SPG                     1 3.5206e+13 9.5610e+15 13628
## - MPG:TrustedYes              1 4.0444e+13 9.5662e+15 13628
## <none>                                     9.5258e+15 13628
## - BPG:TrustedYes              1 4.8696e+13 9.5745e+15 13628
## + APG:TOPG                    1 2.4951e+13 9.5009e+15 13629
## - MPG:SPG                     1 6.2779e+13 9.5886e+15 13629
## + RPG:TrustedYes              1 2.0596e+13 9.5052e+15 13629
## + RPG:AgressivenessYes        1 1.5498e+13 9.5103e+15 13629
## + APG:BPG                     1 1.3015e+13 9.5128e+15 13629
## - MPG:AgressivenessYes        1 7.4532e+13 9.6003e+15 13630
## + I(TOPG^2)                   1 1.1670e+13 9.5141e+15 13630
## + APG:TrustedYes              1 9.4218e+12 9.5164e+15 13630
## + SPG:TrustedYes              1 8.7424e+12 9.5171e+15 13630
## + MPG:RPG                     1 7.2499e+12 9.5186e+15 13630
## + TOPG:SPG                    1 5.7435e+12 9.5201e+15 13630
## + SPG:AgressivenessYes        1 4.6596e+12 9.5211e+15 13630
## + I(SPG^2)                    1 4.3883e+12 9.5214e+15 13630
## + MPG:TOPG                    1 3.5320e+12 9.5223e+15 13630
## + TOPG:TrustedYes             1 2.5560e+12 9.5232e+15 13630
## + TrustedYes:AgressivenessYes 1 1.8627e+12 9.5239e+15 13630
## + PPG:APG                     1 1.8435e+12 9.5240e+15 13630
## + RPG:TOPG                    1 1.7163e+12 9.5241e+15 13630
## + I(RPG^2)                    1 8.1937e+11 9.5250e+15 13630
## + MPG:PPG                     1 3.5216e+11 9.5254e+15 13630
## + I(MPG^2)                    1 3.1239e+11 9.5255e+15 13630
## + TOPG:AgressivenessYes       1 2.7606e+11 9.5255e+15 13630
## + APG:RPG                     1 1.7304e+11 9.5256e+15 13630
## + PPG:RPG                     1 6.3383e+10 9.5257e+15 13630
## + I(PPG^2)                    1 4.9227e+10 9.5258e+15 13630
## + PPG:TrustedYes              1 3.4592e+10 9.5258e+15 13630
## - BPG:AgressivenessYes        1 8.9921e+13 9.6157e+15 13630
## - TOPG:BPG                    1 9.1722e+13 9.6175e+15 13630
## - APG:SPG                     1 1.0066e+14 9.6265e+15 13631
## - APG:AgressivenessYes        1 1.1728e+14 9.6431e+15 13632
## - I(APG^2)                    1 1.3300e+14 9.6588e+15 13632
## - PPG:BPG                     1 1.4617e+14 9.6720e+15 13633
## - MPG:APG                     1 1.7249e+14 9.6983e+15 13634
## - PPG:SPG                     1 1.8130e+14 9.7071e+15 13634
## - PPG:AgressivenessYes        1 1.9227e+14 9.7181e+15 13635
## - PPG:TOPG                    1 2.8196e+14 9.8078e+15 13639
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13627.17
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(APG^2) + I(BPG^2) + MPG:APG + MPG:BPG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG +
##     PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:AgressivenessYes +
##     RPG:SPG + TOPG:BPG + BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - I(BPG^2)                     1 8.4322e+12 9.5583e+15 13626
## - BPG:SPG                      1 3.7370e+13 9.5873e+15 13627
## - RPG:SPG                      1 4.3118e+13 9.5930e+15 13627
## - BPG:TrustedYes               1 4.3296e+13 9.5932e+15 13627
## <none>                                      9.5499e+15 13627
## - MPG:TrustedYes               1 4.4260e+13 9.5941e+15 13627
## - MPG:BPG                      1 5.1604e+13 9.6015e+15 13628
## + RPG:BPG                      1 2.4080e+13 9.5258e+15 13628
## + APG:TOPG                     1 1.7511e+13 9.5324e+15 13628
## - MPG:AgressivenessYes         1 6.9521e+13 9.6194e+15 13628
## + RPG:TrustedYes               1 1.2552e+13 9.5373e+15 13629
## + I(RPG^2)                     1 1.0216e+13 9.5397e+15 13629
## + APG:TrustedYes               1 1.0208e+13 9.5397e+15 13629
## + RPG:AgressivenessYes         1 8.8331e+12 9.5410e+15 13629
## - MPG:SPG                      1 7.8034e+13 9.6279e+15 13629
## + SPG:TrustedYes               1 7.6445e+12 9.5422e+15 13629
## + I(TOPG^2)                    1 7.1081e+12 9.5428e+15 13629
## + TOPG:SPG                     1 6.3194e+12 9.5436e+15 13629
## + SPG:AgressivenessYes         1 4.8610e+12 9.5450e+15 13629
## + APG:BPG                      1 4.7501e+12 9.5451e+15 13629
## + I(SPG^2)                     1 3.7010e+12 9.5462e+15 13629
## + TOPG:TrustedYes              1 2.8915e+12 9.5470e+15 13629
## + APG:RPG                      1 1.8279e+12 9.5481e+15 13629
## + TrustedYes:AgressivenessYes  1 1.3202e+12 9.5486e+15 13629
## + MPG:TOPG                     1 1.1767e+12 9.5487e+15 13629
## + MPG:RPG                      1 8.7588e+11 9.5490e+15 13629
## + PPG:APG                      1 7.5660e+11 9.5491e+15 13629
## + PPG:RPG                      1 5.5608e+11 9.5493e+15 13629
## + TOPG:AgressivenessYes        1 3.8793e+11 9.5495e+15 13629
## + RPG:TOPG                     1 3.8586e+11 9.5495e+15 13629
## + I(MPG^2)                     1 8.8105e+10 9.5498e+15 13629
## + I(PPG^2)                     1 4.1152e+10 9.5498e+15 13629
## + MPG:PPG                      1 1.2955e+10 9.5499e+15 13629
## + PPG:TrustedYes               1 1.2069e+10 9.5499e+15 13629
## - BPG:AgressivenessYes         1 8.6883e+13 9.6368e+15 13629
## - TOPG:BPG                     1 9.9227e+13 9.6491e+15 13630
## - APG:SPG                      1 1.0627e+14 9.6562e+15 13630
## - APG:AgressivenessYes         1 1.1781e+14 9.6677e+15 13631
## - I(APG^2)                     1 1.3481e+14 9.6847e+15 13631
## - PPG:BPG                      1 1.6259e+14 9.7125e+15 13633
## - MPG:APG                      1 1.7019e+14 9.7201e+15 13633
## - PPG:AgressivenessYes         1 1.8644e+14 9.7363e+15 13634
## - PPG:SPG                      1 2.0727e+14 9.7571e+15 13635
## - PPG:TOPG                     1 2.9668e+14 9.8466e+15 13639
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13625.56
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(APG^2) + MPG:APG + MPG:BPG + MPG:SPG +
##     MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG + PPG:BPG +
##     PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:AgressivenessYes +
##     RPG:SPG + TOPG:BPG + BPG:SPG + BPG:TrustedYes + BPG:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - BPG:SPG                      1 3.0129e+13 9.5884e+15 13625
## - RPG:SPG                      1 4.2226e+13 9.6005e+15 13626
## - MPG:BPG                      1 4.3179e+13 9.6015e+15 13626
## <none>                                      9.5583e+15 13626
## - MPG:TrustedYes               1 4.5682e+13 9.6040e+15 13626
## - BPG:TrustedYes               1 5.6003e+13 9.6143e+15 13626
## + APG:TOPG                     1 1.9637e+13 9.5387e+15 13627
## - MPG:AgressivenessYes         1 7.1085e+13 9.6294e+15 13627
## + RPG:TrustedYes               1 1.2994e+13 9.5453e+15 13627
## + APG:BPG                      1 1.0543e+13 9.5478e+15 13627
## + APG:TrustedYes               1 1.0346e+13 9.5480e+15 13627
## + RPG:AgressivenessYes         1 8.7250e+12 9.5496e+15 13627
## + I(BPG^2)                     1 8.4322e+12 9.5499e+15 13627
## + SPG:TrustedYes               1 8.1565e+12 9.5502e+15 13627
## + TOPG:SPG                     1 7.5648e+12 9.5507e+15 13627
## + I(RPG^2)                     1 7.5008e+12 9.5508e+15 13627
## + I(TOPG^2)                    1 5.8335e+12 9.5525e+15 13627
## + RPG:BPG                      1 5.7185e+12 9.5526e+15 13627
## + SPG:AgressivenessYes         1 4.5618e+12 9.5538e+15 13627
## + TOPG:TrustedYes              1 4.0192e+12 9.5543e+15 13627
## - MPG:SPG                      1 8.3329e+13 9.6416e+15 13627
## + I(SPG^2)                     1 3.1280e+12 9.5552e+15 13627
## + PPG:RPG                      1 1.0280e+12 9.5573e+15 13628
## + PPG:APG                      1 1.0072e+12 9.5573e+15 13628
## + TrustedYes:AgressivenessYes  1 9.6238e+11 9.5574e+15 13628
## + MPG:TOPG                     1 6.7199e+11 9.5576e+15 13628
## + APG:RPG                      1 6.1019e+11 9.5577e+15 13628
## + MPG:RPG                      1 5.6357e+11 9.5578e+15 13628
## + TOPG:AgressivenessYes        1 1.5113e+11 9.5582e+15 13628
## + RPG:TOPG                     1 1.0857e+11 9.5582e+15 13628
## + I(MPG^2)                     1 7.4969e+10 9.5582e+15 13628
## + I(PPG^2)                     1 2.0627e+10 9.5583e+15 13628
## + PPG:TrustedYes               1 3.6299e+09 9.5583e+15 13628
## + MPG:PPG                      1 8.7576e+07 9.5583e+15 13628
## - BPG:AgressivenessYes         1 8.8791e+13 9.6471e+15 13628
## - TOPG:BPG                     1 9.1166e+13 9.6495e+15 13628
## - APG:SPG                      1 1.0279e+14 9.6611e+15 13628
## - APG:AgressivenessYes         1 1.2308e+14 9.6814e+15 13629
## - I(APG^2)                     1 1.2731e+14 9.6856e+15 13629
## - PPG:BPG                      1 1.5769e+14 9.7160e+15 13631
## - PPG:AgressivenessYes         1 1.8260e+14 9.7409e+15 13632
## - MPG:APG                      1 2.0028e+14 9.7586e+15 13633
## - PPG:SPG                      1 2.0990e+14 9.7682e+15 13633
## - PPG:TOPG                     1 2.9213e+14 9.8504e+15 13637
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13624.95
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(APG^2) + MPG:APG + MPG:BPG + MPG:SPG +
##     MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG + PPG:BPG +
##     PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:AgressivenessYes +
##     RPG:SPG + TOPG:BPG + BPG:TrustedYes + BPG:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - RPG:SPG                      1 1.8257e+13 9.6067e+15 13624
## - MPG:BPG                      1 2.1829e+13 9.6103e+15 13624
## <none>                                      9.5884e+15 13625
## + BPG:SPG                      1 3.0129e+13 9.5583e+15 13626
## - TOPG:BPG                     1 6.6228e+13 9.6547e+15 13626
## - MPG:TrustedYes               1 6.7101e+13 9.6555e+15 13626
## - MPG:AgressivenessYes         1 6.7695e+13 9.6561e+15 13626
## + I(RPG^2)                     1 1.7106e+13 9.5713e+15 13626
## - BPG:TrustedYes               1 7.0657e+13 9.6591e+15 13626
## + RPG:BPG                      1 1.5265e+13 9.5732e+15 13626
## + SPG:TrustedYes               1 1.0670e+13 9.5778e+15 13626
## + APG:TrustedYes               1 9.2209e+12 9.5792e+15 13626
## + APG:TOPG                     1 9.2196e+12 9.5792e+15 13626
## - BPG:AgressivenessYes         1 7.8460e+13 9.6669e+15 13626
## - MPG:SPG                      1 8.1554e+13 9.6700e+15 13627
## + PPG:RPG                      1 4.9724e+12 9.5835e+15 13627
## + RPG:TrustedYes               1 4.9460e+12 9.5835e+15 13627
## + SPG:AgressivenessYes         1 4.5314e+12 9.5839e+15 13627
## + RPG:AgressivenessYes         1 4.1073e+12 9.5843e+15 13627
## + TOPG:TrustedYes              1 3.6638e+12 9.5848e+15 13627
## + I(TOPG^2)                    1 3.5690e+12 9.5849e+15 13627
## + TOPG:SPG                     1 3.5298e+12 9.5849e+15 13627
## + I(SPG^2)                     1 2.2058e+12 9.5862e+15 13627
## + MPG:RPG                      1 1.9908e+12 9.5865e+15 13627
## + RPG:TOPG                     1 1.5761e+12 9.5869e+15 13627
## + PPG:APG                      1 1.3125e+12 9.5871e+15 13627
## + I(BPG^2)                     1 1.1909e+12 9.5873e+15 13627
## + TrustedYes:AgressivenessYes  1 7.7205e+11 9.5877e+15 13627
## + MPG:TOPG                     1 6.4047e+11 9.5878e+15 13627
## + APG:RPG                      1 5.3888e+11 9.5879e+15 13627
## + APG:BPG                      1 1.8631e+11 9.5883e+15 13627
## + I(MPG^2)                     1 1.6169e+11 9.5883e+15 13627
## + TOPG:AgressivenessYes        1 6.6052e+10 9.5884e+15 13627
## + I(PPG^2)                     1 2.5138e+10 9.5884e+15 13627
## + MPG:PPG                      1 2.2387e+10 9.5884e+15 13627
## + PPG:TrustedYes               1 1.3072e+09 9.5884e+15 13627
## - APG:SPG                      1 8.8058e+13 9.6765e+15 13627
## - APG:AgressivenessYes         1 1.2271e+14 9.7112e+15 13629
## - I(APG^2)                     1 1.2901e+14 9.7174e+15 13629
## - PPG:BPG                      1 1.3774e+14 9.7262e+15 13629
## - PPG:AgressivenessYes         1 1.7464e+14 9.7631e+15 13631
## - MPG:APG                      1 1.7651e+14 9.7649e+15 13631
## - PPG:SPG                      1 1.9761e+14 9.7861e+15 13632
## - PPG:TOPG                     1 2.6335e+14 9.8518e+15 13635
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13623.79
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(APG^2) + MPG:APG + MPG:BPG + MPG:SPG +
##     MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG + PPG:BPG +
##     PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:AgressivenessYes +
##     TOPG:BPG + BPG:TrustedYes + BPG:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## - MPG:BPG                      1 3.0269e+13 9.6370e+15 13623
## <none>                                      9.6067e+15 13624
## - MPG:TrustedYes               1 5.8085e+13 9.6648e+15 13624
## + I(RPG^2)                     1 2.5182e+13 9.5815e+15 13625
## - BPG:TrustedYes               1 6.2902e+13 9.6696e+15 13625
## - MPG:SPG                      1 6.3352e+13 9.6701e+15 13625
## - MPG:AgressivenessYes         1 6.3611e+13 9.6703e+15 13625
## + RPG:SPG                      1 1.8257e+13 9.5884e+15 13625
## + PPG:RPG                      1 1.4100e+13 9.5926e+15 13625
## + RPG:BPG                      1 1.3506e+13 9.5932e+15 13625
## + SPG:TrustedYes               1 1.2489e+13 9.5942e+15 13625
## - BPG:AgressivenessYes         1 7.5236e+13 9.6819e+15 13625
## - TOPG:BPG                     1 7.6211e+13 9.6829e+15 13625
## + APG:TOPG                     1 1.0818e+13 9.5959e+15 13625
## + RPG:TOPG                     1 1.0693e+13 9.5960e+15 13625
## + APG:TrustedYes               1 9.4749e+12 9.5972e+15 13625
## + MPG:RPG                      1 8.6781e+12 9.5980e+15 13625
## + TOPG:SPG                     1 7.0638e+12 9.5996e+15 13626
## + BPG:SPG                      1 6.1590e+12 9.6005e+15 13626
## + TOPG:TrustedYes              1 4.1811e+12 9.6025e+15 13626
## + I(SPG^2)                     1 4.1345e+12 9.6026e+15 13626
## + I(TOPG^2)                    1 3.8757e+12 9.6028e+15 13626
## + I(BPG^2)                     1 3.0415e+12 9.6037e+15 13626
## + APG:BPG                      1 2.4489e+12 9.6043e+15 13626
## + APG:RPG                      1 2.0024e+12 9.6047e+15 13626
## + SPG:AgressivenessYes         1 1.6004e+12 9.6051e+15 13626
## + MPG:TOPG                     1 1.2868e+12 9.6054e+15 13626
## + PPG:APG                      1 1.0059e+12 9.6057e+15 13626
## + TrustedYes:AgressivenessYes  1 6.0185e+11 9.6061e+15 13626
## + RPG:TrustedYes               1 2.3715e+11 9.6065e+15 13626
## + RPG:AgressivenessYes         1 1.3855e+11 9.6066e+15 13626
## + TOPG:AgressivenessYes        1 8.6826e+10 9.6066e+15 13626
## + I(MPG^2)                     1 2.0696e+10 9.6067e+15 13626
## + PPG:TrustedYes               1 1.1809e+10 9.6067e+15 13626
## + I(PPG^2)                     1 9.8169e+09 9.6067e+15 13626
## + MPG:PPG                      1 5.5853e+08 9.6067e+15 13626
## - APG:SPG                      1 1.1595e+14 9.7227e+15 13627
## - APG:AgressivenessYes         1 1.1902e+14 9.7257e+15 13627
## - PPG:BPG                      1 1.4072e+14 9.7474e+15 13628
## - I(APG^2)                     1 1.4223e+14 9.7489e+15 13628
## - MPG:APG                      1 1.6421e+14 9.7709e+15 13629
## - PPG:AgressivenessYes         1 1.6734e+14 9.7740e+15 13629
## - PPG:SPG                      1 2.0086e+14 9.8076e+15 13631
## - PPG:TOPG                     1 2.4642e+14 9.8531e+15 13633
## - RPG                          1 4.7769e+14 1.0084e+16 13643
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
##
## Step:  AIC=13623.18
## Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG + SPG + TrustedYes +
##     AgressivenessYes + I(APG^2) + MPG:APG + MPG:SPG + MPG:TrustedYes +
##     MPG:AgressivenessYes + PPG:TOPG + PPG:BPG + PPG:SPG + PPG:AgressivenessYes +
##     APG:SPG + APG:AgressivenessYes + TOPG:BPG + BPG:TrustedYes +
##     BPG:AgressivenessYes
##
##                               Df  Sum of Sq        RSS   AIC
## <none>                                     9.6370e+15 13623
## + I(RPG^2)                     1 4.0755e+13 9.5962e+15 13623
## + RPG:BPG                      1 3.2462e+13 9.6045e+15 13624
## - TOPG:BPG                     1 5.5278e+13 9.6922e+15 13624
## - MPG:AgressivenessYes         1 5.6598e+13 9.6936e+15 13624
## + MPG:BPG                      1 3.0269e+13 9.6067e+15 13624
## - MPG:SPG                      1 5.8008e+13 9.6950e+15 13624
## - BPG:AgressivenessYes         1 5.9268e+13 9.6962e+15 13624
## + RPG:SPG                      1 2.6697e+13 9.6103e+15 13624
## + MPG:RPG                      1 2.6005e+13 9.6110e+15 13624
## + PPG:RPG                      1 1.8795e+13 9.6182e+15 13624
## + RPG:TOPG                     1 1.5239e+13 9.6217e+15 13624
## + APG:TOPG                     1 1.4769e+13 9.6222e+15 13624
## - MPG:TrustedYes               1 7.3838e+13 9.7108e+15 13625
## + APG:TrustedYes               1 9.2494e+12 9.6277e+15 13625
## + TOPG:SPG                     1 9.0400e+12 9.6279e+15 13625
## + TOPG:TrustedYes              1 7.5608e+12 9.6294e+15 13625
## + APG:BPG                      1 6.4859e+12 9.6305e+15 13625
## + SPG:TrustedYes               1 4.9391e+12 9.6320e+15 13625
## + I(SPG^2)                     1 4.8687e+12 9.6321e+15 13625
## + PPG:APG                      1 3.3978e+12 9.6336e+15 13625
## + SPG:AgressivenessYes         1 3.0211e+12 9.6339e+15 13625
## + APG:RPG                      1 2.6198e+12 9.6343e+15 13625
## + I(MPG^2)                     1 1.8825e+12 9.6351e+15 13625
## + I(TOPG^2)                    1 1.3308e+12 9.6356e+15 13625
## + MPG:PPG                      1 3.8245e+11 9.6366e+15 13625
## + RPG:TrustedYes               1 3.5643e+11 9.6366e+15 13625
## + MPG:TOPG                     1 1.8714e+11 9.6368e+15 13625
## + I(PPG^2)                     1 6.7831e+10 9.6369e+15 13625
## + BPG:SPG                      1 6.4782e+10 9.6369e+15 13625
## + RPG:AgressivenessYes         1 5.7643e+10 9.6369e+15 13625
## + PPG:TrustedYes               1 5.3417e+10 9.6369e+15 13625
## + TOPG:AgressivenessYes        1 3.4254e+10 9.6369e+15 13625
## + TrustedYes:AgressivenessYes  1 1.9170e+09 9.6370e+15 13625
## + I(BPG^2)                     1 1.3607e+09 9.6370e+15 13625
## - PPG:BPG                      1 1.1230e+14 9.7493e+15 13626
## - APG:AgressivenessYes         1 1.1261e+14 9.7496e+15 13626
## - APG:SPG                      1 1.1865e+14 9.7556e+15 13627
## - I(APG^2)                     1 1.4611e+14 9.7831e+15 13628
## - MPG:APG                      1 1.5359e+14 9.7906e+15 13628
## - PPG:AgressivenessYes         1 1.5778e+14 9.7948e+15 13628
## - BPG:TrustedYes               1 1.9502e+14 9.8320e+15 13630
## - PPG:SPG                      1 1.9892e+14 9.8359e+15 13630
## - PPG:TOPG                     1 2.3613e+14 9.8731e+15 13632
## - RPG                          1 4.8769e+14 1.0125e+16 13643
```

```
summary(model_stepwise)
```

```
##
## Call:
## lm(formula = Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG +
##     SPG + TrustedYes + AgressivenessYes + I(APG^2) + MPG:APG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG +
##     PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:AgressivenessYes +
##     TOPG:BPG + BPG:TrustedYes + BPG:AgressivenessYes, data = per_game_stats,
##     trControl = train_control)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -14937985  -2273358   -461127   2318966  18582921
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -543468    1186172  -0.458  0.64707
## MPG                    180653     133652   1.352  0.17721
## PPG                    -59682     251745  -0.237  0.81272
## APG                   -1969636    1092244  -1.803  0.07206 .
## RPG                    818296     177918   4.599 5.63e-06 ***
## TOPG                   -391774    1612970  -0.243  0.80821
## BPG                     -73317    1612103  -0.045  0.96375
## SPG                    2663406    3261544   0.817  0.41462
## TrustedYes           -11599960    4748465  -2.443  0.01498 *
## AgressivenessYes       5902362    3551177   1.662  0.09725 .
## I(APG^2)                283784     112728   2.517  0.01219 *
## MPG:APG                 137917      53434   2.581  0.01019 *
## MPG:SPG                -281745     177621  -1.586  0.11345
## MPG:TrustedYes          392843     219513   1.790  0.07424 .
## MPG:AgressivenessYes   -296519     189250  -1.567  0.11791
## PPG:TOPG               -274828      85875  -3.200  0.00148 **
## PPG:BPG                -363323     164619  -2.207  0.02785 *
## PPG:SPG                 672283     228872   2.937  0.00349 **
## PPG:AgressivenessYes    733673     280450   2.616  0.00922 **
## APG:SPG               -1325717     584378  -2.269  0.02380 *
## APG:AgressivenessYes  -1885903     853314  -2.210  0.02764 *
## TOPG:BPG               1979001    1278068   1.548  0.12228
## BPG:TrustedYes         6520298    2241878   2.908  0.00383 **
## BPG:AgressivenessYes  -3907032    2436786  -1.603  0.10961
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4802000 on 418 degrees of freedom
## Multiple R-squared:  0.6192, Adjusted R-squared:  0.5983
## F-statistic: 29.56 on 23 and 418 DF,  p-value: < 2.2e-16
```

Best model

```
stepwise_model <- lm(formula = Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG +
    SPG + TrustedYes + AgressivenessYes + I(APG^2) + MPG:APG +
    MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG +
    PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:AgressivenessYes +
    TOPG:BPG + BPG:TrustedYes + BPG:AgressivenessYes, data = per_game_stats,
    trControl = train_control)
```

```
## Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
##  extra argument 'trControl' will be disregarded
```

```
summary(stepwise_model)
```

```
##
## Call:
## lm(formula = Salary17_18 ~ MPG + PPG + APG + RPG + TOPG + BPG +
##     SPG + TrustedYes + AgressivenessYes + I(APG^2) + MPG:APG +
##     MPG:SPG + MPG:TrustedYes + MPG:AgressivenessYes + PPG:TOPG +
##     PPG:BPG + PPG:SPG + PPG:AgressivenessYes + APG:SPG + APG:AgressivenessYes +
##     TOPG:BPG + BPG:TrustedYes + BPG:AgressivenessYes, data = per_game_stats,
##     trControl = train_control)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -14937985  -2273358   -461127   2318966  18582921
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)           -543468    1186172  -0.458  0.64707
## MPG                    180653     133652   1.352  0.17721
## PPG                    -59682     251745  -0.237  0.81272
## APG                   -1969636    1092244  -1.803  0.07206 .
## RPG                    818296     177918   4.599 5.63e-06 ***
## TOPG                   -391774    1612970  -0.243  0.80821
## BPG                     -73317    1612103  -0.045  0.96375
## SPG                    2663406    3261544   0.817  0.41462
## TrustedYes           -11599960    4748465  -2.443  0.01498 *
## AgressivenessYes       5902362    3551177   1.662  0.09725 .
## I(APG^2)                283784     112728   2.517  0.01219 *
## MPG:APG                 137917      53434   2.581  0.01019 *
## MPG:SPG                -281745     177621  -1.586  0.11345
## MPG:TrustedYes          392843     219513   1.790  0.07424 .
## MPG:AgressivenessYes   -296519     189250  -1.567  0.11791
## PPG:TOPG               -274828      85875  -3.200  0.00148 **
## PPG:BPG                -363323     164619  -2.207  0.02785 *
## PPG:SPG                 672283     228872   2.937  0.00349 **
## PPG:AgressivenessYes    733673     280450   2.616  0.00922 **
## APG:SPG               -1325717     584378  -2.269  0.02380 *
## APG:AgressivenessYes  -1885903     853314  -2.210  0.02764 *
## TOPG:BPG               1979001    1278068   1.548  0.12228
## BPG:TrustedYes         6520298    2241878   2.908  0.00383 **
## BPG:AgressivenessYes  -3907032    2436786  -1.603  0.10961
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 4802000 on 418 degrees of freedom
## Multiple R-squared:  0.6192, Adjusted R-squared:  0.5983
## F-statistic: 29.56 on 23 and 418 DF,  p-value: < 2.2e-16
```

Model Performance: The model's r-squared value is 0.6192, and the adjusted r-squared is 0.5983. This means that the predictors in the model explain about 61.92% of the variation in player salaries, showing that it captures a substantial portion of the influencing factors.

Key Predictors: Certain variables, like rebounds per game (RPG), show a strong positive relationship with salary. This indicates that players who excel at rebounding tend to earn more.

Interaction Effects: The model includes interaction terms such as PPG (points per game and steals per game) and PPG (points per game and aggressive play style). These significant interactions demonstrate that specific skill combinations can increase player salaries.

Second-Order Terms: The model includes second-order terms like I(APG2)I(APG^2)I(APG2) (assists per game squared). These terms help capture non-linear relationships, where the effect of assists on salary changes at higher levels.

Model Robustness: The strong F-statistic (p-value < 2.2e-16) confirms that the model, as a whole, is statistically significant and the predictors contribute meaningful information for salary prediction.

RandomForest

```
library(dplyr)
rf_data <- per_game_stats %>% select( MPG, PPG, APG, RPG, TOPG, BPG, SPG, TrustedYes, AgressivenessYes, Salary17_
18)
head(rf_data)
```

```
##         MPG       PPG       APG      RPG      TOPG       BPG        SPG
## 1  7.409091  2.181818 0.1818182 1.636364 0.4545455 0.5909091 0.04545455
## 2 13.753846  4.953846 1.9230769 1.061538 1.0153846 0.1384615 0.38461538
## 3 28.725000 12.737500 1.8750000 5.062500 1.1125000 0.5000000 0.80000000
## 4 29.065574  8.721311 1.6229508 7.393443 1.5409836 0.7213115 0.98360656
## 5 32.250000 14.000000 4.9558824 6.823529 1.7058824 1.2794118 0.76470588
## 6 14.106061  8.106061 0.8636364 4.212121 0.5000000 0.2424242 0.28787879
##   TrustedYes AgressivenessYes Salary17_18
## 1          0               0     1312611
## 2          0               0     2116955
## 3          1               0     5504420
## 4          1               1     7319035
## 5          1               1    27734405
## 6          0               0     9769821
```

```
library(randomForest)
```

```
## randomForest 4.7-1.2
```

```
## Type rfNews() to see new features/changes/bug fixes.
```

```
##
## Attaching package: 'randomForest'
```

```
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
rf_model <- randomForest(Salary17_18 ~ ., data = rf_data, ntree = 500)

print(rf_model)
```

```
##
## Call:
##  randomForest(formula = Salary17_18 ~ ., data = rf_data, ntree = 500)
##                Type of random forest: regression
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##           Mean of squared residuals: 2.66485e+13
##                     % Var explained: 53.46
```

```
predictions <- predict(rf_model, rf_data)

rmse <- sqrt(mean((predictions - rf_data$Salary17_18)^2))
cat("RMSE: ", rmse, "\n")
```

```
## RMSE:  2350629
```

```
rsq <- 1 - sum((predictions - rf_data$Salary17_18)^2) / sum((rf_data$Salary17_18 - mean(rf_data$Salary17_18))^2)
cat("R-squared: ", rsq, "\n")
```

```
## R-squared:  0.9035066
```

Mean of Squared Residuals: The mean squared residual (error) is 2.637338e+13 which gives an idea of how well the model is fitting the data (lower values are better).

The model explains 53.94% of the variance in the target variable (Salary17_18). This indicates that the model captures just over half of the variability in the target variable, which is useful but not perfect.

RMSE (Root Mean Squared Error): The RMSE is 2,334,881, which measures the average magnitude of the error between the predicted and actual values. This indicates how far off, on average, the predictions are from the actual values.

R-squared: The R^2 value of 0.9048 is very high, indicating that about 90.48% of the variance in salary can be explained by the model's features.

Accuracy comparison

let's create a predicted columns for both the models

```
predict_salary_rf <- function(point, minutes, turn_over, rebounds, assists, block, steal, TrustedYes, Agressivene
ssYes) {
  predict(rf_model, data.frame(PPG = point, MPG = minutes, TOPG = turn_over, RPG = rebounds, SPG = steal,
                               APG = assists, BPG = block, TrustedYes = TrustedYes, AgressivenessYes = Agressiven
essYes))
}

per_game_stats$rf_Predicted_Salary <- mapply(predict_salary_rf,
                                             point = per_game_stats$PPG,
                                             minutes = per_game_stats$MPG,
                                             turn_over = per_game_stats$TOPG,
                                             rebounds = per_game_stats$RPG,
                                             assists = per_game_stats$APG,
                                             block = per_game_stats$BPG,
                                             steal = per_game_stats$SPG,
                                             TrustedYes = per_game_stats$TrustedYes,
                                             AgressivenessYes = per_game_stats$AgressivenessYes)


predict_salary_stepwise <- function(point, minutes, turn_over, rebounds, assists, block, steal, TrustedYes, Agres
sivenessYes) {
  predict(stepwise_model, data.frame(PPG = point, MPG = minutes, TOPG = turn_over, RPG = rebounds, SPG = steal,
                               APG = assists, BPG = block, TrustedYes = TrustedYes, AgressivenessYes = Agressiven
essYes))
}

per_game_stats$stepwise_Predicted_Salary <- mapply(predict_salary_stepwise,
                                             point = per_game_stats$PPG,
                                             minutes = per_game_stats$MPG,
                                             turn_over = per_game_stats$TOPG,
                                             rebounds = per_game_stats$RPG,
                                             assists = per_game_stats$APG,
                                             block = per_game_stats$BPG,
                                             steal = per_game_stats$SPG,
                                             TrustedYes = per_game_stats$TrustedYes,
                                             AgressivenessYes = per_game_stats$AgressivenessYes)

head(per_game_stats)
```

```
##            Player Team       MPG       PPG       APG      RPG      TOPG
## 1    A.J. Hammons  DAL  7.409091  2.181818 0.1818182 1.636364 0.4545455
## 2    Aaron Brooks  IND 13.753846  4.953846 1.9230769 1.061538 1.0153846
## 3    Aaron Gordon  ORL 28.725000 12.737500 1.8750000 5.062500 1.1125000
## 4 Al-Farouq Aminu  POR 29.065574  8.721311 1.6229508 7.393443 1.5409836
## 5      Al Horford  BOS 32.250000 14.000000 4.9558824 6.823529 1.7058824
## 6    Al Jefferson  IND 14.106061  8.106061 0.8636364 4.212121 0.5000000
##         BPG        SPG TrustedYes AgressivenessYes Salary17_18 Cluster
## 1 0.5909091 0.04545455          0                0     1312611       2
## 2 0.1384615 0.38461538          0                0     2116955       2
## 3 0.5000000 0.80000000          1                0     5504420       2
## 4 0.7213115 0.98360656          1                1     7319035       2
## 5 1.2794118 0.76470588          1                1    27734405       3
## 6 0.2424242 0.28787879          0                0     9769821       1
##   rf_Predicted_Salary stepwise_Predicted_Salary
## 1             1482978                   1491947
## 2             1865646                   1496801
## 3             6844990                  10643725
## 4             9303577                  10398142
## 5            20956951                  18707029
## 6             6776402                   4218798
```

```
library(ggplot2)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
```

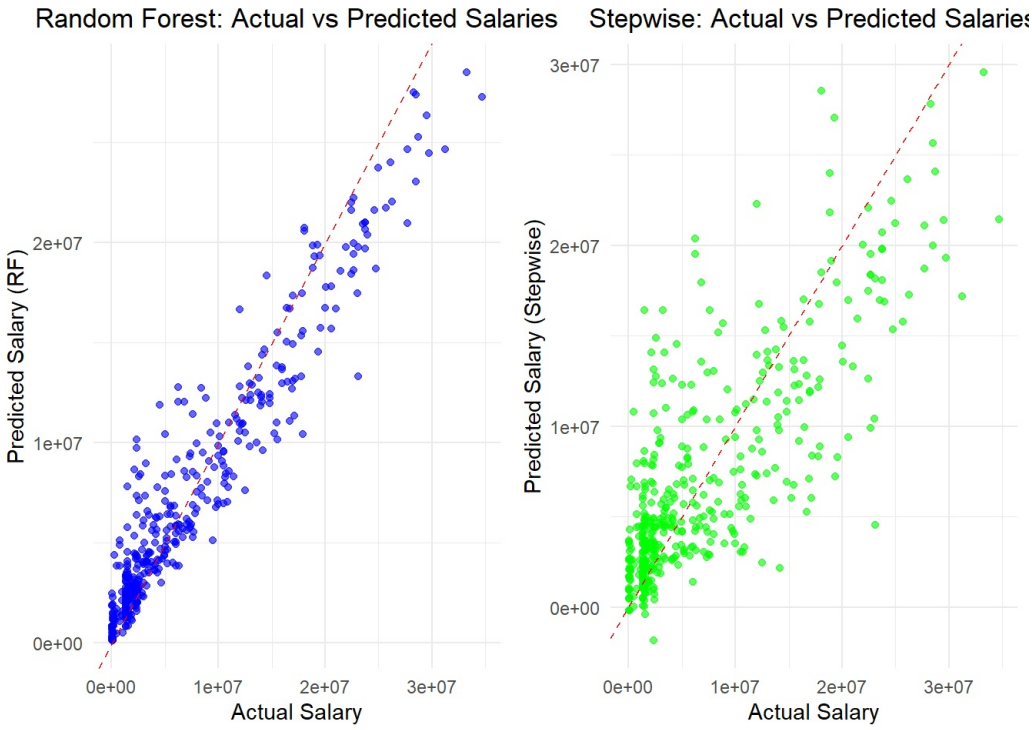
```
## The following object is masked from 'package:randomForest':
##
##     combine
```

```
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
plot_rf <- ggplot(per_game_stats) +
  geom_point(aes(x = Salary17_18, y = rf_Predicted_Salary), color = "blue", alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Random Forest: Actual vs Predicted Salaries", x = "Actual Salary", y = "Predicted Salary (RF)") +
  theme_minimal() + theme(plot.title = element_text(hjust = 0.5))

plot_stepwise <- ggplot(per_game_stats) +
  geom_point(aes(x = Salary17_18, y = stepwise_Predicted_Salary), color = "green", alpha = 0.6) +
  geom_abline(slope = 1, intercept = 0, color = "red", linetype = "dashed") +
  labs(title = "Stepwise: Actual vs Predicted Salaries", x = "Actual Salary", y = "Predicted Salary (Stepwise)")
+
  theme_minimal() + theme(plot.title = element_text(hjust = 0.5))

grid.arrange(plot_rf, plot_stepwise, ncol = 2)
```



Random Forest:

The predicted values closely align with the actual values, especially for low-to-mid-range salaries, suggesting high accuracy. Slight deviations occur at the higher salary range.

Stepwise Regression:

The predictions are more dispersed around the red line, indicating lower accuracy and higher variance. It struggles with higher salaries, showing significant underprediction.

Overall, the Random Forest model outperforms Stepwise Regression in terms of predictive accuracy and consistency.

g. Evaluation

```
summary(rf_data$Salary17_18)
```

```
##     Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
##    17224  1518316  3519283  7106258 10812701 34682550
```

Let's set the median as the threshold for our classification of salaries.

```
threshold <- 3519283
rf_data$Salary_Class <- ifelse(rf_data$Salary17_18 > threshold, "High Salary", "Low Salary")
rf_data$Salary_Class <- factor(rf_data$Salary_Class)

predictions <- predict(rf_model, rf_data)
predicted_class <- ifelse(predictions > threshold, "High Salary", "Low Salary")
predicted_class <- factor(predicted_class, levels = c("Low Salary", "High Salary"))
```

A salary class column is created based on weather the salary is above or below the threshold.

The rf_model is used to make the predictions.

Predictions are also classified as high or low based on the same threshold.

```
conf_matrix <- table(Predicted = predicted_class, Actual = rf_data$Salary_Class)
print(conf_matrix)
```

```
##             Actual
## Predicted     High Salary Low Salary
##   Low Salary            2        171
##   High Salary         219         50
```

The confusion matrix indicates that the model has a relatively high number of false positives (219) for “High Salary” and true negatives (167) for “Low Salary.” The recall is quite high (0.99), meaning it successfully identifies most players with “High Salary,” but the precision (0.80) suggests that it misclassifies some players as “High Salary” when they actually belong in the “Low Salary” category.

```
TP <- conf_matrix["High Salary", "High Salary"]
TN <- conf_matrix["Low Salary", "Low Salary"]
FP <- conf_matrix["High Salary", "Low Salary"]
FN <- conf_matrix["Low Salary", "High Salary"]

precision <- TP / (TP + FP)
recall <- TP / (TP + FN)

cat("Precision: ", precision, "\n")
```

```
## Precision:  0.8141264
```

```
cat("Recall: ", recall, "\n")
```

```
## Recall:  0.9909502
```

Precision (0.80): This means that 80% of the players predicted as “High Salary” actually have a high salary. It’s a measure of the model’s accuracy in identifying “High Salary” players without making too many mistakes (false positives).

Recall (0.99): This indicates that the model correctly identified 99% of the actual “High Salary” players. It’s a measure of the model’s ability to capture most of the true “High Salary” players, even if it makes some false positives.

The model has a very high recall, meaning it successfully identifies most high-salary players. However, with a precision of 80%, it occasionally misclassifies lower salary players as high salary. While the model excels at recognizing high salary players, improving its precision by reducing false positives would lead to better performance overall.

```
library(pROC)
```

```
## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'
```

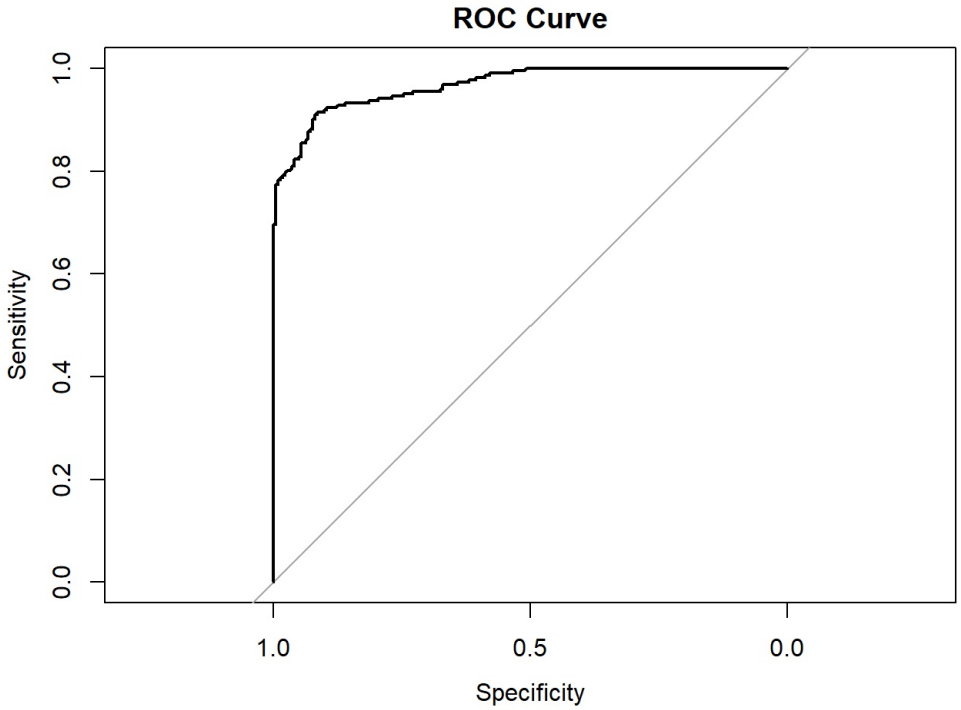
```
## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
probabilities <- predict(rf_model, rf_data, type = "response")
roc_curve <- roc(rf_data$Salary_Class, probabilities)
```

```
## Setting levels: control = High Salary, case = Low Salary
```

```
## Setting direction: controls > cases
```

```
plot(roc_curve, main = "ROC Curve")
```



Curve Shape: The curve is close to the top-left corner, indicating strong model performance. Diagonal Reference Line: The diagonal represents random guessing. Your curve being above this line shows the model performs better than random.

Area Under Curve (AUC): Though not explicitly mentioned, the shape suggests a high AUC, reflecting good discrimination between high and low salary classes.

The model has excellent sensitivity and precision (as seen in prior metrics), and the ROC curve confirms its ability to distinguish between salary classes effectively. However, the imbalance in misclassification rates (notably, high false positives for “High Salary”) should be considered for refinement.

Report

1.Introduction

This project involves building a predictive model to estimate NBA player salaries for the 2017- 2018 season. The prediction is based on player performance data from the 2016-2017 season, allowing us to assess how past performance influences future earnings.

Objective:

To develop a model that can predict the 2017-2018 season salaries of NBA players using their statistics from the previous season. Understanding the relationship between player performance and salary can provide insights into the factors that NBA teams consider when compensating players. It highlights which performance metrics are most valued and how they contribute to a player’s market worth, guiding both management decisions and player career strategies.

2.Data Gathering and Integration

Two primary datasets were used:

a.Stats dataset: Contains detailed performance metrics for NBA players from 1950 to 2017.

b.Salary_1718 dataset: Includes salaries of NBA players for the 2017-2018 season. The datasets were filtered to focus on the 2016-2017 season stats and merged based on player names.

3.Data Cleaning and Preprocessing

1.Removal of unnecessary columns (e.g., Team).

2.Handling of 32 NA values by replacing them with column means.

3.Creation of per-game statistics (e.g., PPG, MPG, APG) to normalize player performance.

4.Introduction of derived variables: “Aggressiveness”: Based on above-average turnovers per game. “Trusted”: Based on above-average minutes per game.

4.Data Exploration

Correlation analysis revealed:

Strongest correlations with salary: PPG > MPG > TOPG > RPG > PER > SPG > APG

Positive correlation between turnovers and salary, interpreted as a sign of player involvement Visualization of PPG vs. salary showed a general upward trend, indicating higher salaries for players with higher scoring averages.

5.Clustering Analysis

The optimal number of clusters was determined using two methods:

Elbow Method: Suggested 3 clusters as optimal.

Silhouette Method: Confirmed 3 clusters as optimal.

Clustering observations revealed:

- Cluster 1 (Red): Compact group, mostly trusted players
- Cluster 2 (Green): Spread-out group, mix of trusted and non-trusted players
- Cluster 3 (Blue): Separated group, almost entirely trusted and aggressive players

6.Classification Models

Two classification models were developed:

Stepwise Regression Model:

R-squared: 0.6192

Adjusted R-squared: 0.5983

Key predictors: RPG, PPG interactions, APG squared

Random Forest Model:

Mean Squared Residuals: 2.637338e+13

Variance explained: 53.94%

R-squared: 0.9048

The Random Forest model outperformed the Stepwise Regression model in terms of predictive accuracy and consistency.

7.Model Evaluation

A threshold of $351,983 (median salary) was set to classify salaries as high or low. The Random Forest model was evaluated using various metrics:

Precision: 0.80

Recall: 0.99

ROC curve: Showed strong model performance with a curve close to the top-left corner The model excelled at identifying high-salary players but occasionally misclassified lower salary players as high salary.

.8.Conclusion

This project demonstrated the effectiveness of machine learning techniques in predicting NBA player salaries based on performance metrics. The Random Forest model showed superior performance, explaining about 90% of the variance in salaries. The analysis highlighted the importance of scoring, rebounding, and playing time in determining a player's market value. Future refinements could focus on reducing false positives in high salary predictions to improve overall model accuracy.

i. Reflection

This Project emphasized the importance of data preparation, from cleaning to visualization, and taught the nuances of unsupervised and supervised techniques. Through hands-on work, I strengthened my R programming skills and gained deeper insights into clustering and classification. The journey underscored the value of metrices like precision and Roc in complementing accuracy for model evaluation.

Fundamental of Data Science as a course has provided a solid foundation for data science principles and practices. The hands-on experience with working with real world data as assignments and projects has been invaluable, bridging the gap between theory and applications. As I move forward in my data science journey, I am confident that these skills and experiences will serve as a strong basis of tackling more complex challenges and contributing meaningfully to my future projects.