Final-Project-2

2024-12-05

```
knitr::opts_chunk$set(echo = TRUE)

# Load necessary packages
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 4.4.2
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.4.2

## Warning: package 'tibble' was built under R version 4.4.2

## Warning: package 'tidyr' was built under R version 4.4.2

## Warning: package 'readr' was built under R version 4.4.2

## Warning: package 'purrr' was built under R version 4.4.2

## Warning: package 'dplyr' was built under R version 4.4.2

## Warning: package 'forcats' was built under R version 4.4.2

## Warning: package 'lubridate' was built under R version 4.4.2

## -- Attaching core tidyverse packages ------------------------ tidyverse 2.0.0 --
## v dplyr     1.1.4     v readr     2.1.5
## v forcats   1.0.0     v stringr   1.5.1
## v lubridate 1.9.3     v tibble    3.2.1
## v purrr     1.0.2     v tidyr     1.3.1
## -- Conflicts ---------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggcorrplot)
```

```
## Warning: package 'ggcorrplot' was built under R version 4.4.2
```

```r
library(olsrr)
```

```
## Warning: package 'olsrr' was built under R version 4.4.2
```

```
##
## Attaching package: 'olsrr'
##
## The following object is masked from 'package:datasets':
##
##     rivers
```

```r
library(readxl)
```

```
## Warning: package 'readxl' was built under R version 4.4.2
```

```r
library(gridExtra)
```

```
## Warning: package 'gridExtra' was built under R version 4.4.2
```

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

```r
library(tinytex)
```

```
## Warning: package 'tinytex' was built under R version 4.4.2
```

```r
# Load data
data <- read_excel("C:/Users/jknod/Desktop/STAT 461/Final Project/Dataset/Dataset461.xlsx")

# Remove teamName, teamID and teamDivision
data <- data[,-1:-2]
```

Data Visualization

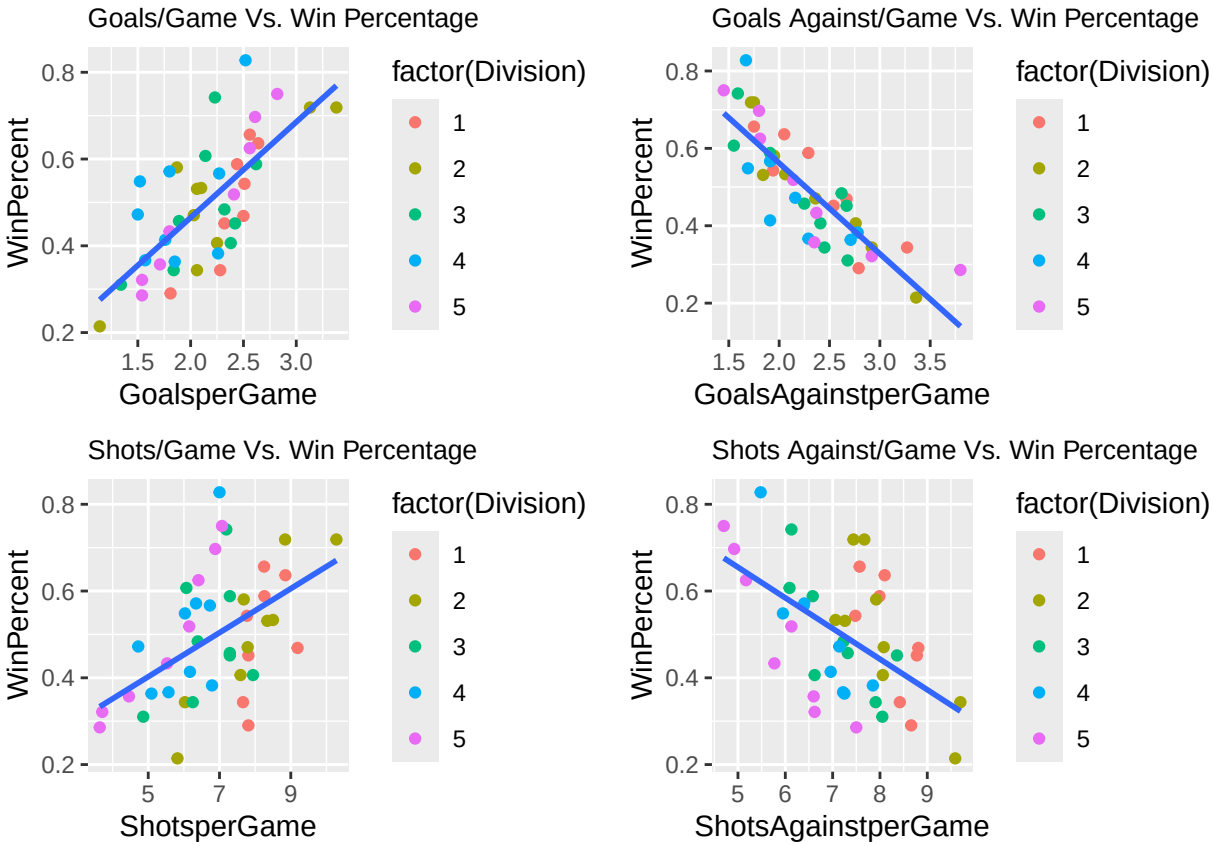
Scatterplots

```r
# Plot Independent Variables Against WinPercent

# Goals/Game
plot1 <- ggplot(data, aes(GoalsperGame, WinPercent)) +
  geom_point(aes(color = factor(Division))) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Goals/Game Vs. Win Percentage") +
```

```
  theme(plot.title = element_text(size = 10))

# Goals Against/Game
plot2 <- ggplot(data, aes(GoalsAgainstperGame, WinPercent)) +
  geom_point(aes(color = factor(Division))) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Goals Against/Game Vs. Win Percentage") +
  theme(plot.title = element_text(size = 10))

# Shots/Game
plot3 <- ggplot(data, aes(ShotsperGame, WinPercent)) +
  geom_point(aes(color = factor(Division))) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Shots/Game Vs. Win Percentage") +
  theme(plot.title = element_text(size = 10))

# Shots Against/Game
plot4 <- ggplot(data, aes(ShotsAgainstperGame, WinPercent)) +
  geom_point(aes(color = factor(Division))) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Shots Against/Game Vs. Win Percentage") +
  theme(plot.title = element_text(size = 10))

# Shooting Percentage
plot5 <- ggplot(data, aes(ShootingPercent, WinPercent)) +
  geom_point(aes(color = factor(Division))) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Shooting Percentage Vs. Win Percentage") +
  theme(plot.title = element_text(size = 10))

# Assists/Game
plot6 <- ggplot(data, aes(AssistsperGame, WinPercent)) +
  geom_point(aes(color = factor(Division))) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Assists/Game Vs. Win Percentage") +
  theme(plot.title = element_text(size = 10))

# Saves/Game
plot7 <- ggplot(data, aes(SavesperGame, WinPercent)) +
  geom_point(aes(color = factor(Division))) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Saves/Game Vs. Win Percentage") +
  theme(plot.title = element_text(size = 10))

# SavePercent
plot8 <- ggplot(data, aes(SavePercent, WinPercent)) +
  geom_point(aes(color = factor(Division))) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Save Percentage Vs. Win Percentage") +
  theme(plot.title = element_text(size = 10))

# Demos/Game
plot9 <- ggplot(data, aes(DemosperGame, WinPercent)) +
```

```
  geom_point(aes(color = factor(Division))) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Demos/Game Vs. Win Percentage") +
  theme(plot.title = element_text(size = 10))

# Demos Against/Game
plot10 <- ggplot(data, aes(DemosAgainstperGame, WinPercent)) +
  geom_point(aes(color = factor(Division))) +
  geom_smooth(method = "lm", se = FALSE) +
  labs(title = "Demos Against/Game Vs. Win Percentage") +
  theme(plot.title = element_text(size = 10))

# Display all plots in a grid arrangement (2 rows and 2 columns)
grid.arrange(plot1, plot2, plot3, plot4,
             ncol = 2, nrow = 2)
```

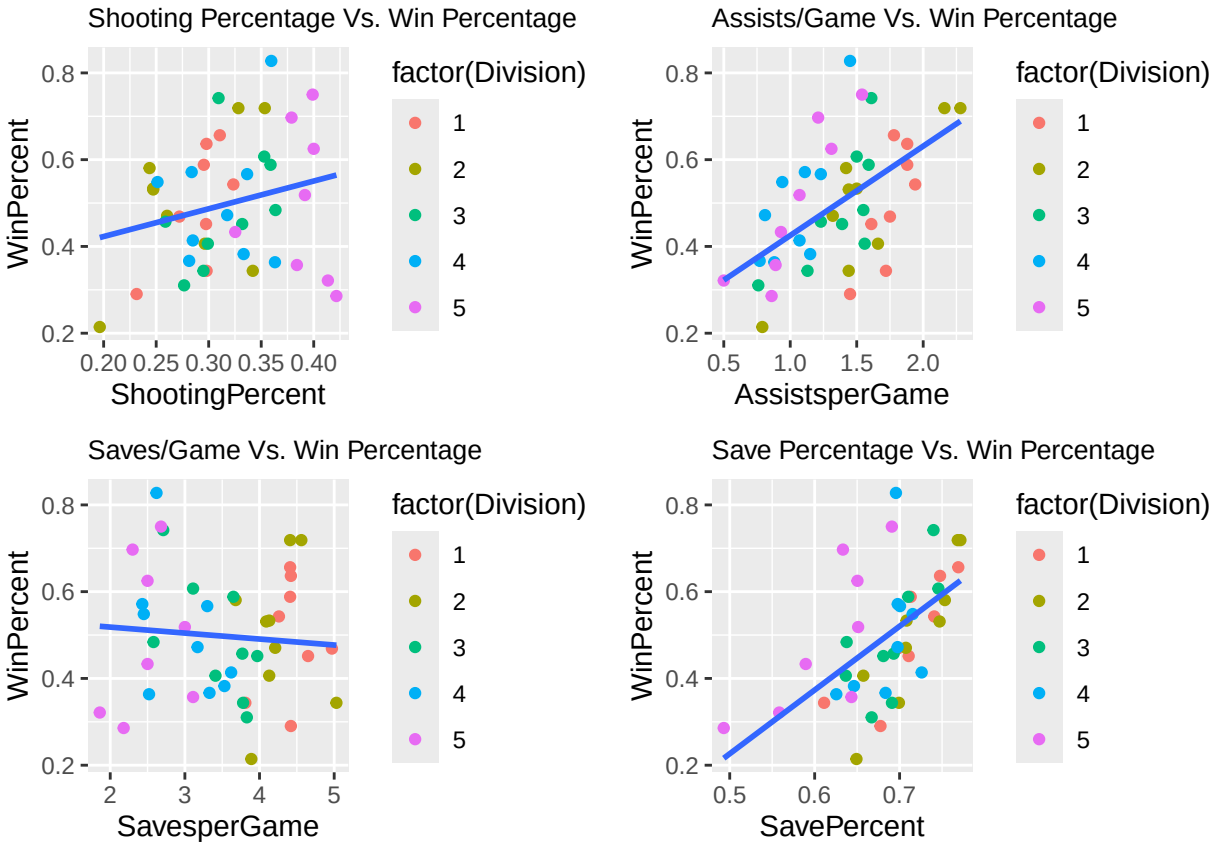
```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
grid.arrange(plot5, plot6, plot7, plot8,
             ncol = 2, nrow = 2)
```

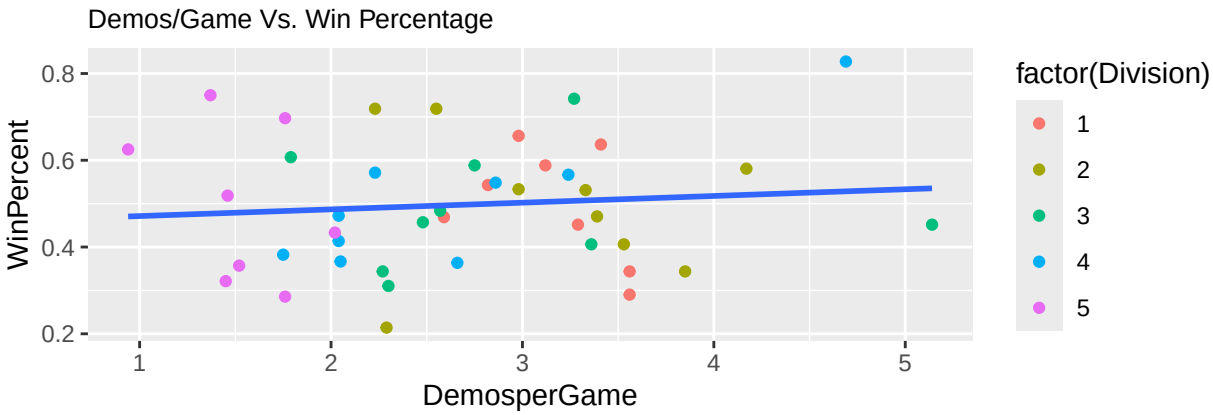
```
## `geom_smooth()` using formula = 'y ~ x'
```

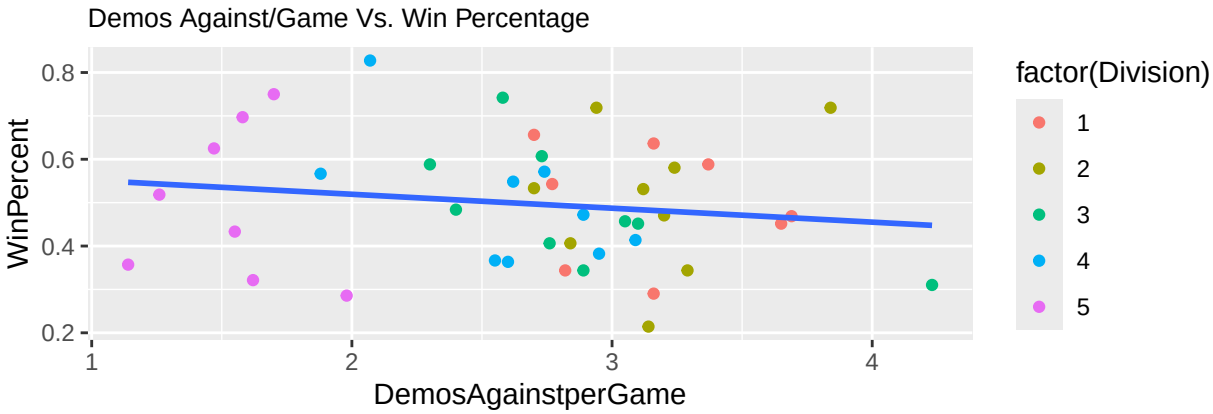
```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```




```
grid.arrange(plot9, plot10,
             ncol = 1, nrow = 2)
```

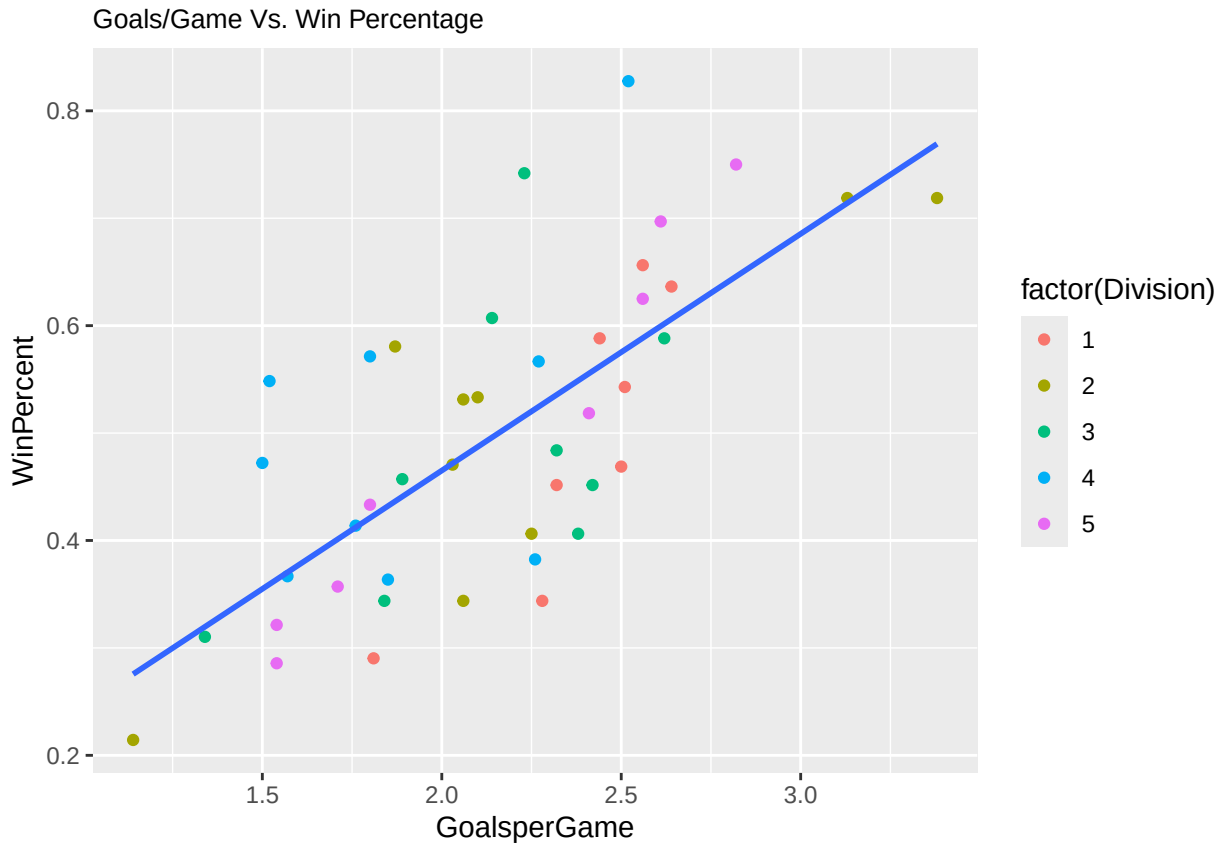
```
## `geom_smooth()` using formula = 'y ~ x'
## `geom_smooth()` using formula = 'y ~ x'
```



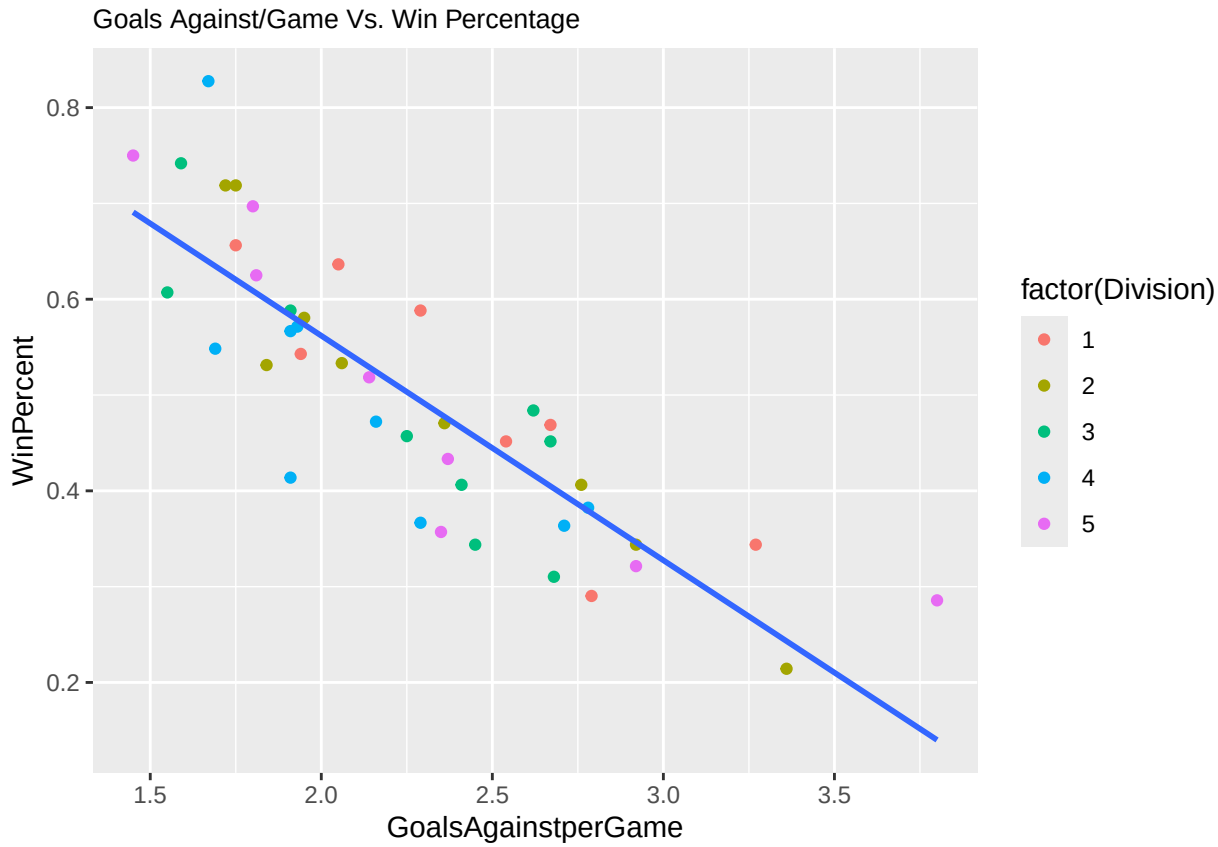



```
# Store all the plots in a list
plot_list <- list(plot1, plot2, plot3, plot4, plot5, plot6, plot7, plot8, plot9, plot10)

# Loop through the list to display each plot in its own window
for (i in seq_along(plot_list)) {
  print(plot_list[[i]])  # Print each plot
}
```

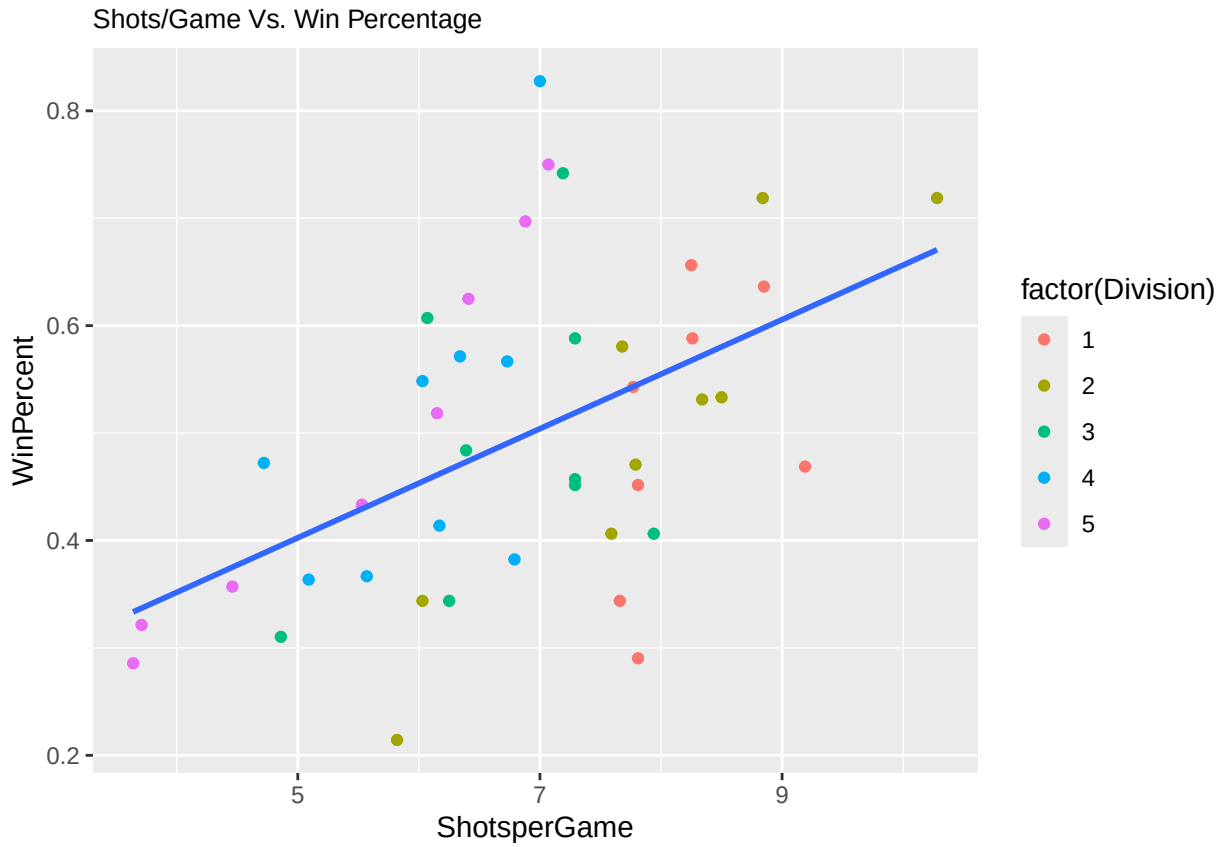
```
## `geom_smooth()` using formula = 'y ~ x'
```



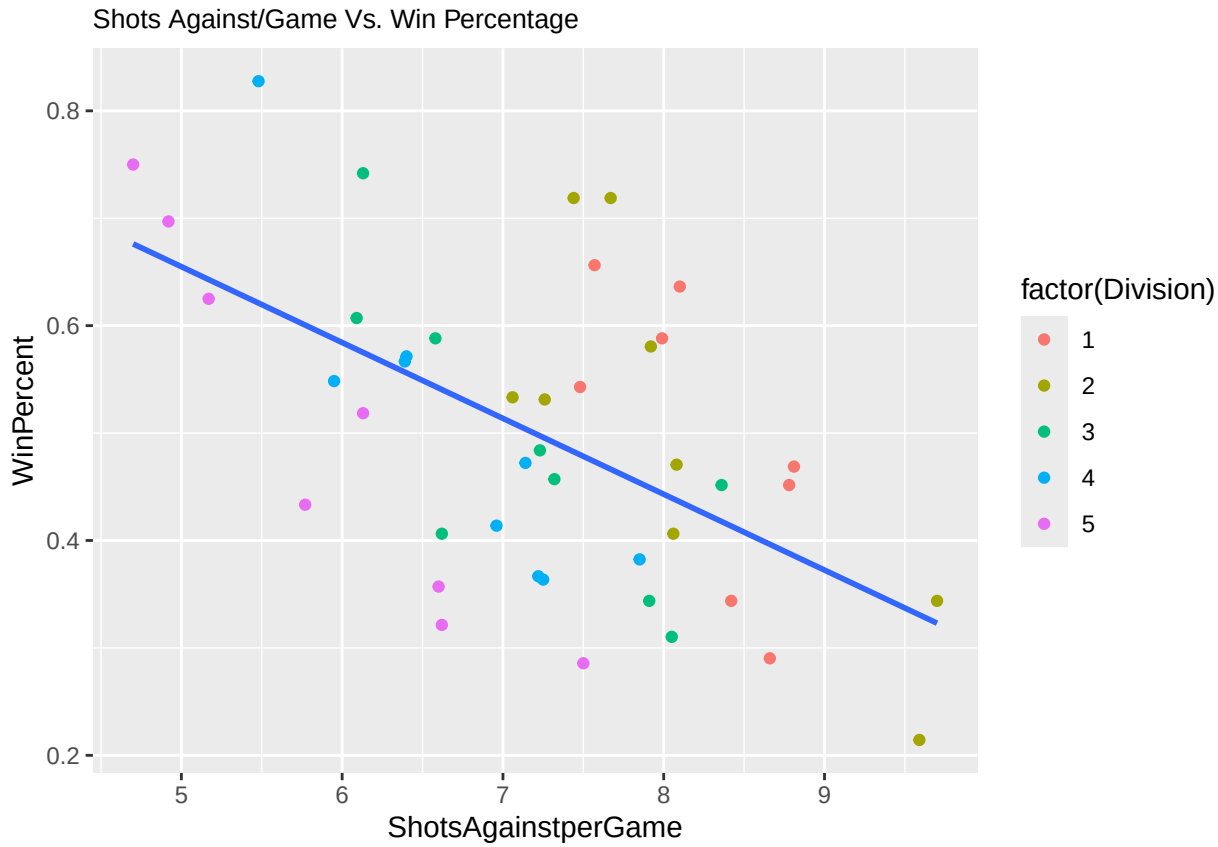

```
## `geom_smooth()` using formula = 'y ~ x'
```



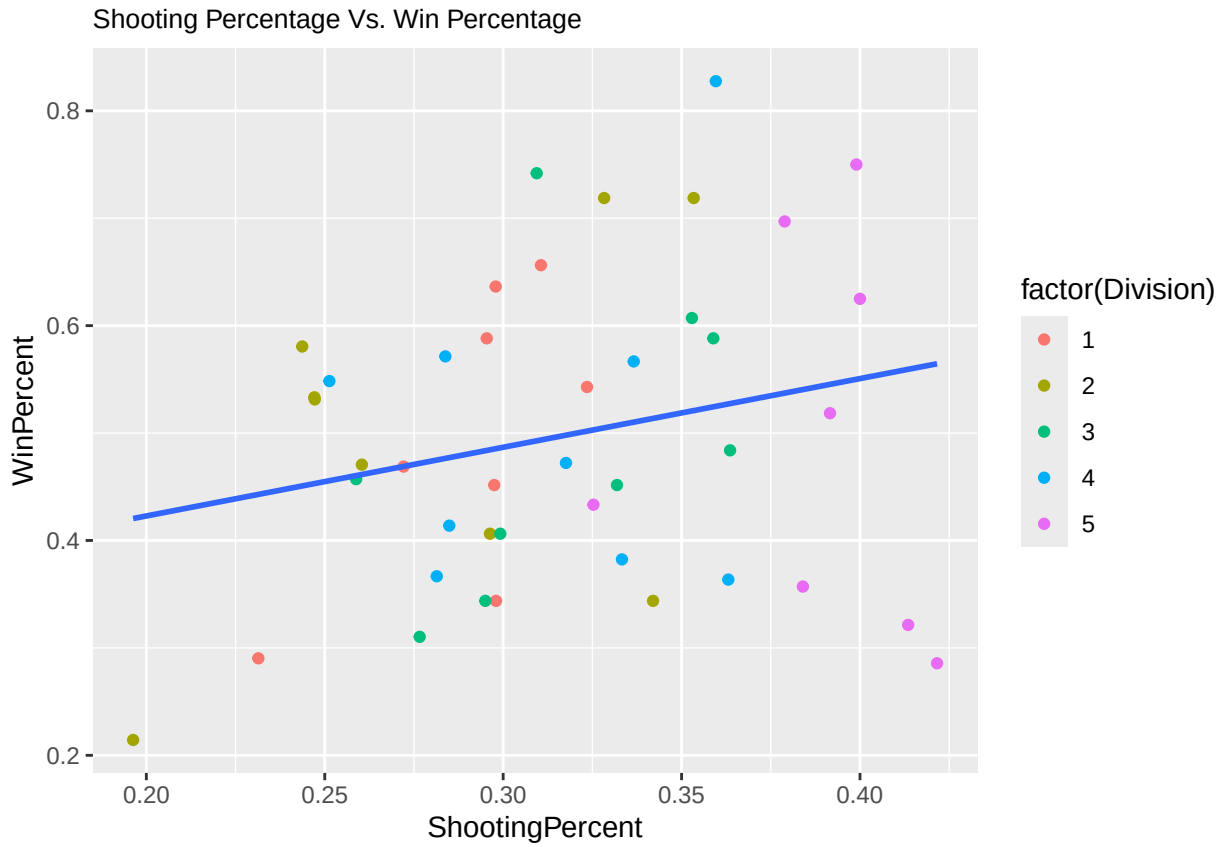

```
## `geom_smooth()` using formula = 'y ~ x'
```



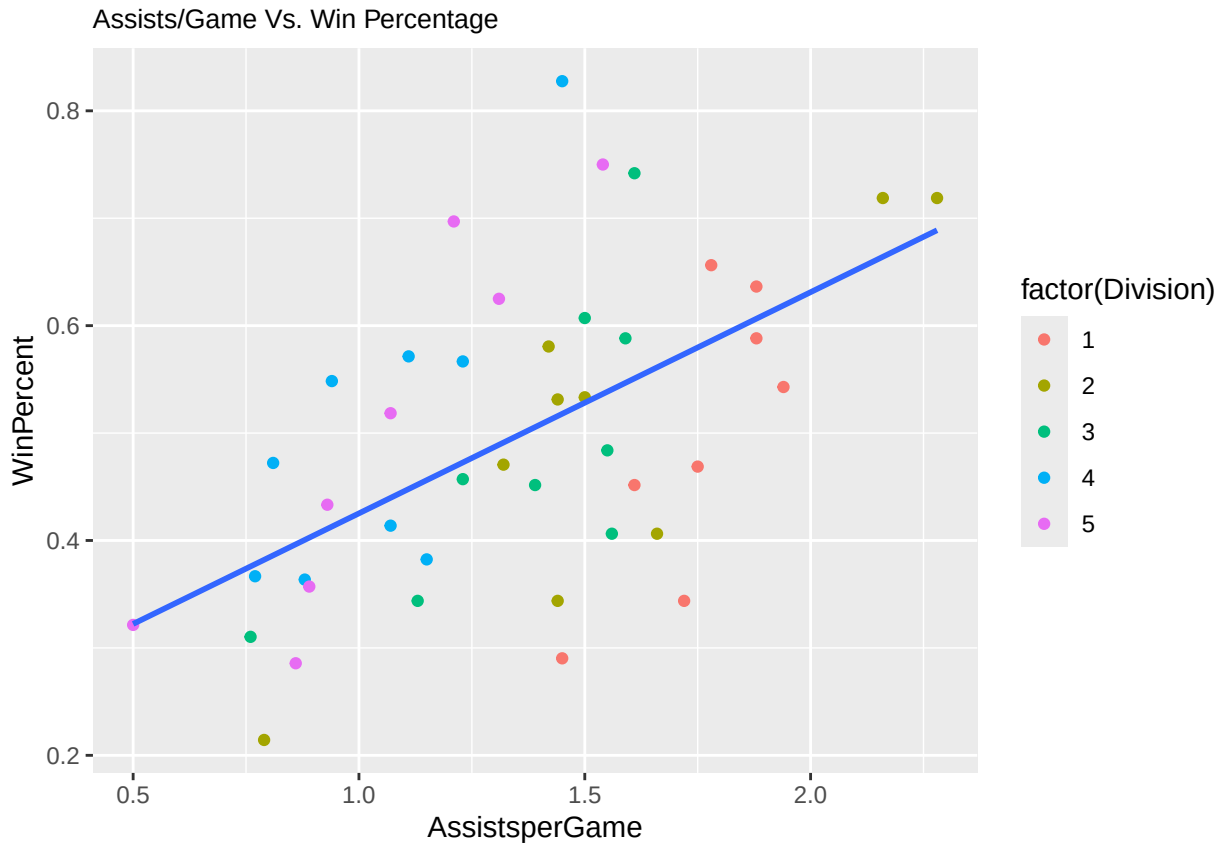

```
## `geom_smooth()` using formula = 'y ~ x'
```



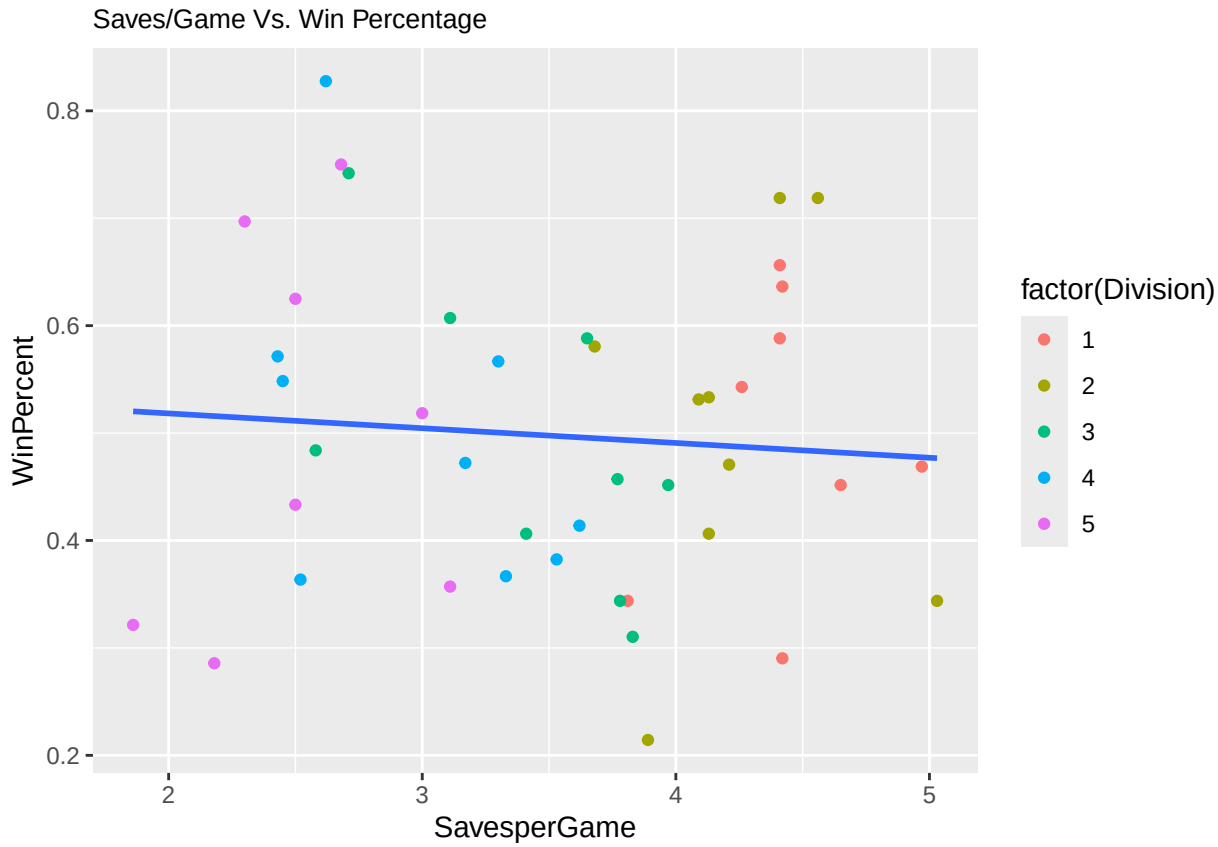

```
## `geom_smooth()` using formula = 'y ~ x'
```



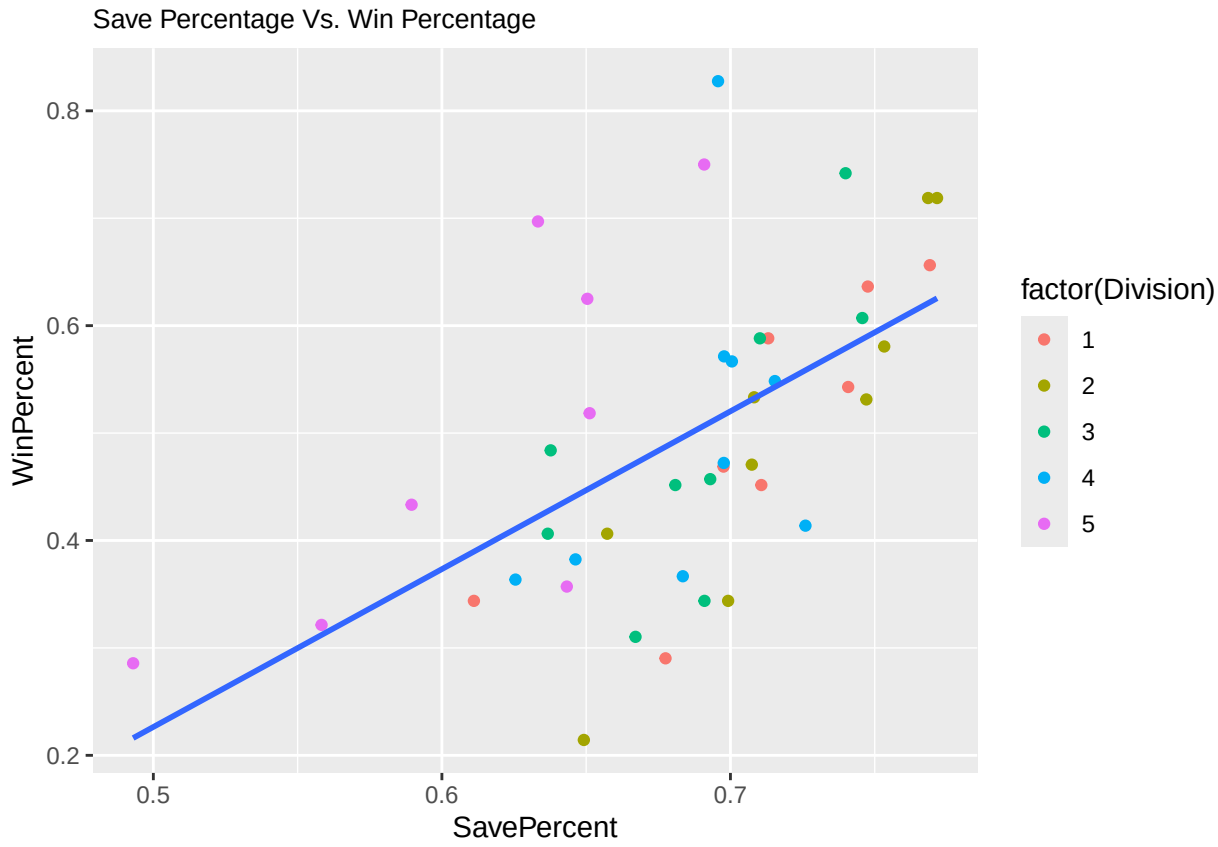

```
## `geom_smooth()` using formula = 'y ~ x'
```



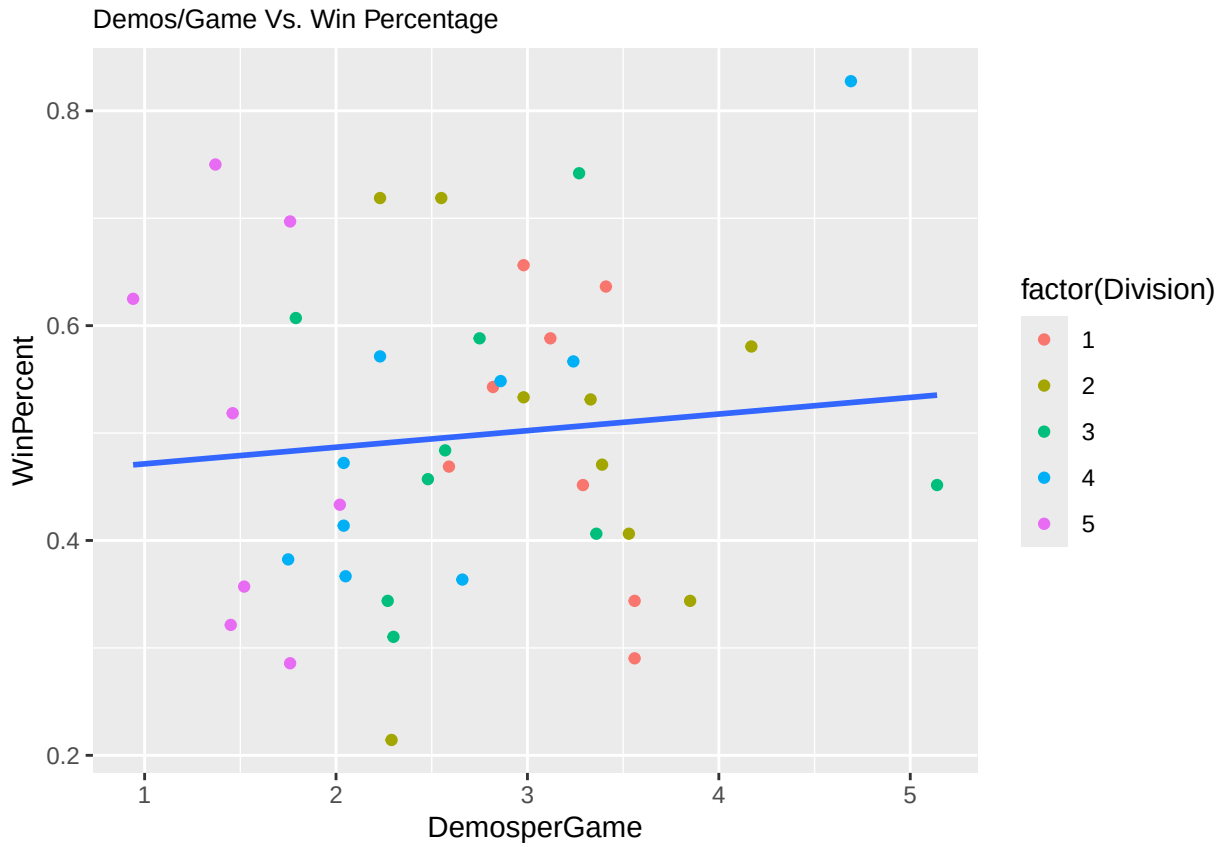

```
## `geom_smooth()` using formula = 'y ~ x'
```



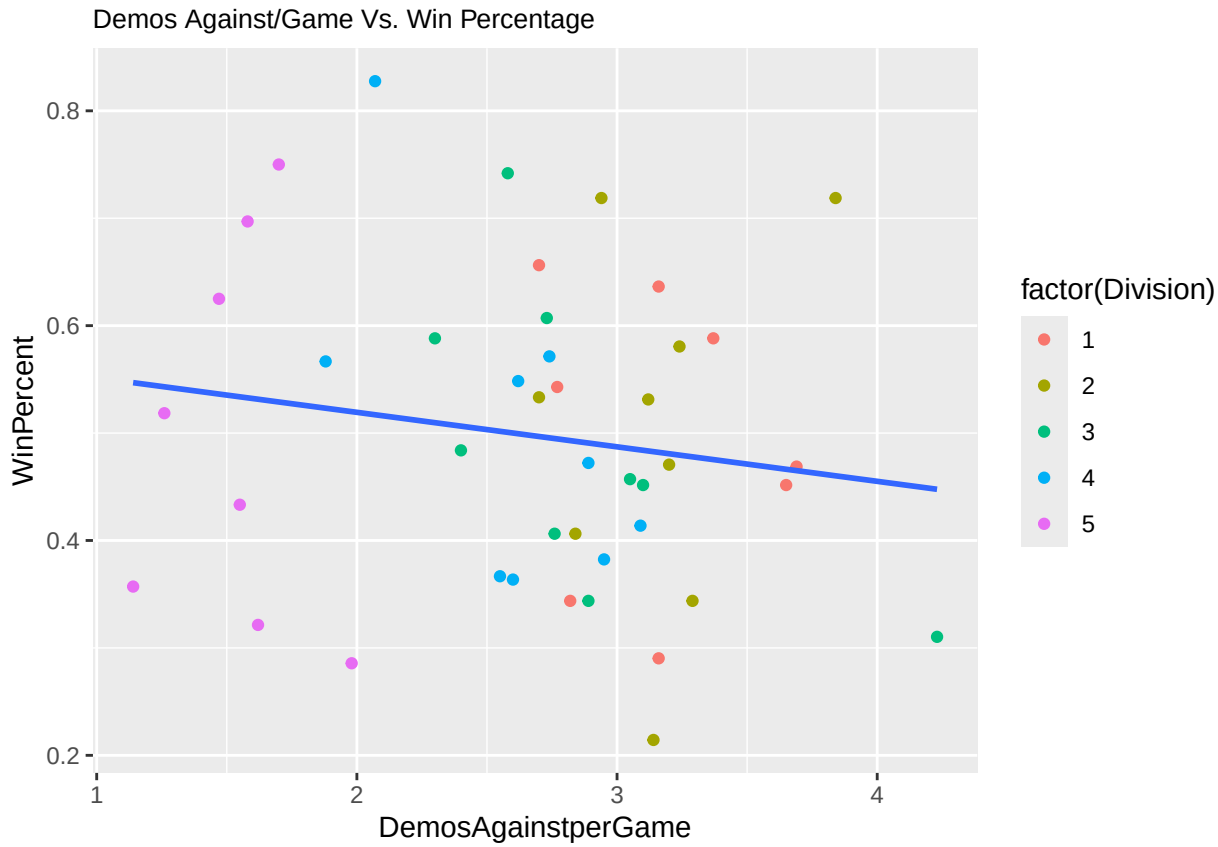

```
## `geom_smooth()` using formula = 'y ~ x'
```




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




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




```r
# Get t-statistic, p-value, R-Square and adj. R-Square
response_var <- "WinPercent"

# Perform linear regressions and extract relevant statistics
results <- lapply(setdiff(names(data), response_var), function(var) {
  formula <- as.formula(paste(response_var, "~", var))
  model <- lm(formula, data = data)
  model_summary <- summary(model)

  # Extract t-statistic, p-value, R^2, and adjusted R^2
  list(
    t_statistic = round(coef(model_summary)[2, "t value"], 3),
    p_value = round(coef(model_summary)[2, "Pr(>|t|)"], 3),
    r_squared = round(model_summary$r.squared, 3),
    adj_r_squared = round(model_summary$adj.r.squared, 3)
  )
})

names(results) <- setdiff(names(data), response_var)

# Display the extracted statistics
for (var_name in names(results)) {
  cat("=== Results for", var_name, "===\n")
  cat("T-statistic:", results[[var_name]]$t_statistic, "\n")
  cat("P-value:", results[[var_name]]$p_value, "\n")
  cat("R-squared:", results[[var_name]]$r_squared, "\n")
```

```
  cat("Adjusted R-squared:", results[[var_name]]$adj_r_squared, "\n\n")
}
```

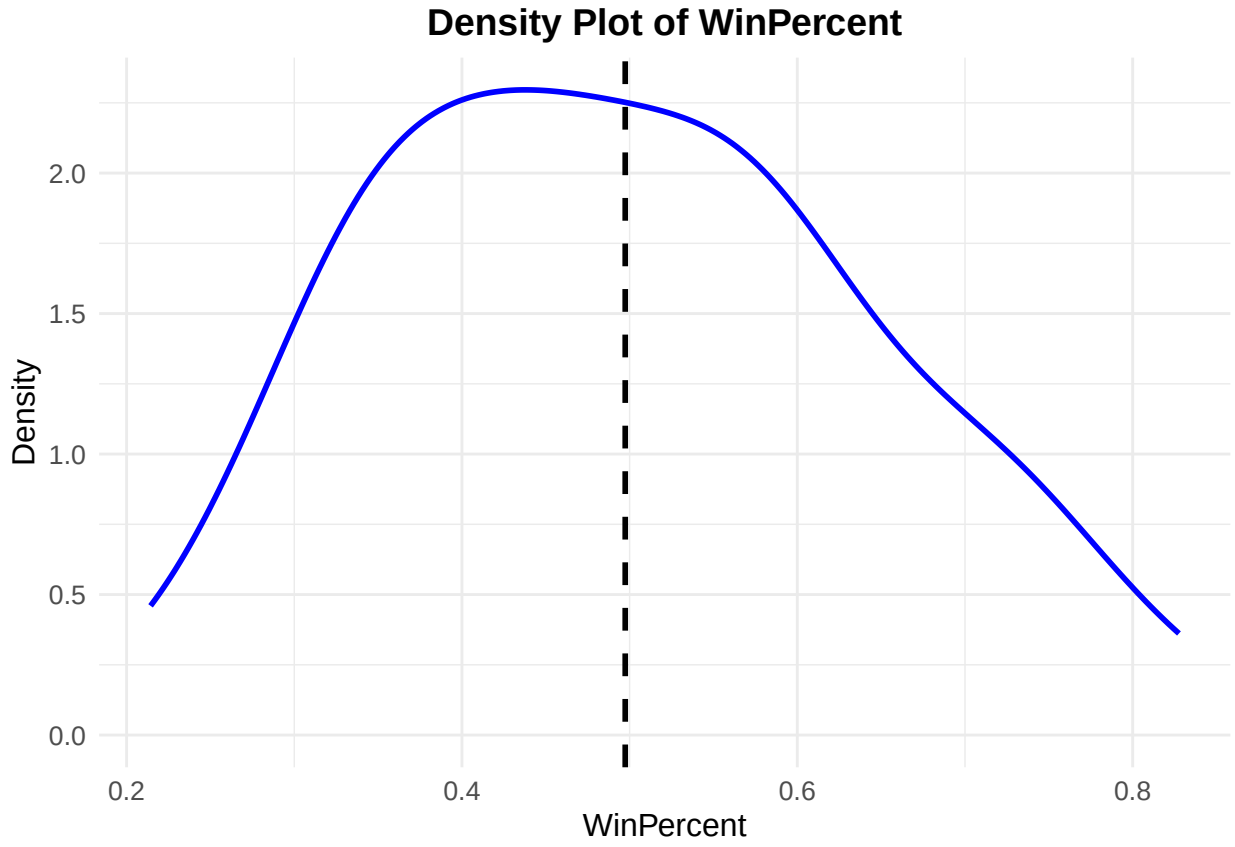
```
## === Results for GoalsperGame ===
## T-statistic: 6.52
## P-value: 0
## R-squared: 0.509
## Adjusted R-squared: 0.497
##
## === Results for ShotsperGame ===
## T-statistic: 3.719
## P-value: 0.001
## R-squared: 0.252
## Adjusted R-squared: 0.234
##
## === Results for ShootingPercent ===
## T-statistic: 1.529
## P-value: 0.134
## R-squared: 0.054
## Adjusted R-squared: 0.031
##
## === Results for AssistsperGame ===
## T-statistic: 4.418
## P-value: 0
## R-squared: 0.322
## Adjusted R-squared: 0.306
##
## === Results for SavesperGame ===
## T-statistic: -0.502
## P-value: 0.618
## R-squared: 0.006
## Adjusted R-squared: -0.018
##
## === Results for GoalsAgainstperGame ===
## T-statistic: -10.338
## P-value: 0
## R-squared: 0.723
## Adjusted R-squared: 0.716
##
## === Results for ShotsAgainstperGame ===
## T-statistic: -4.327
## P-value: 0
## R-squared: 0.314
## Adjusted R-squared: 0.297
##
## === Results for SavePercent ===
## T-statistic: 4.507
## P-value: 0
## R-squared: 0.331
## Adjusted R-squared: 0.315
##
## === Results for DemosperGame ===
## T-statistic: 0.617
```

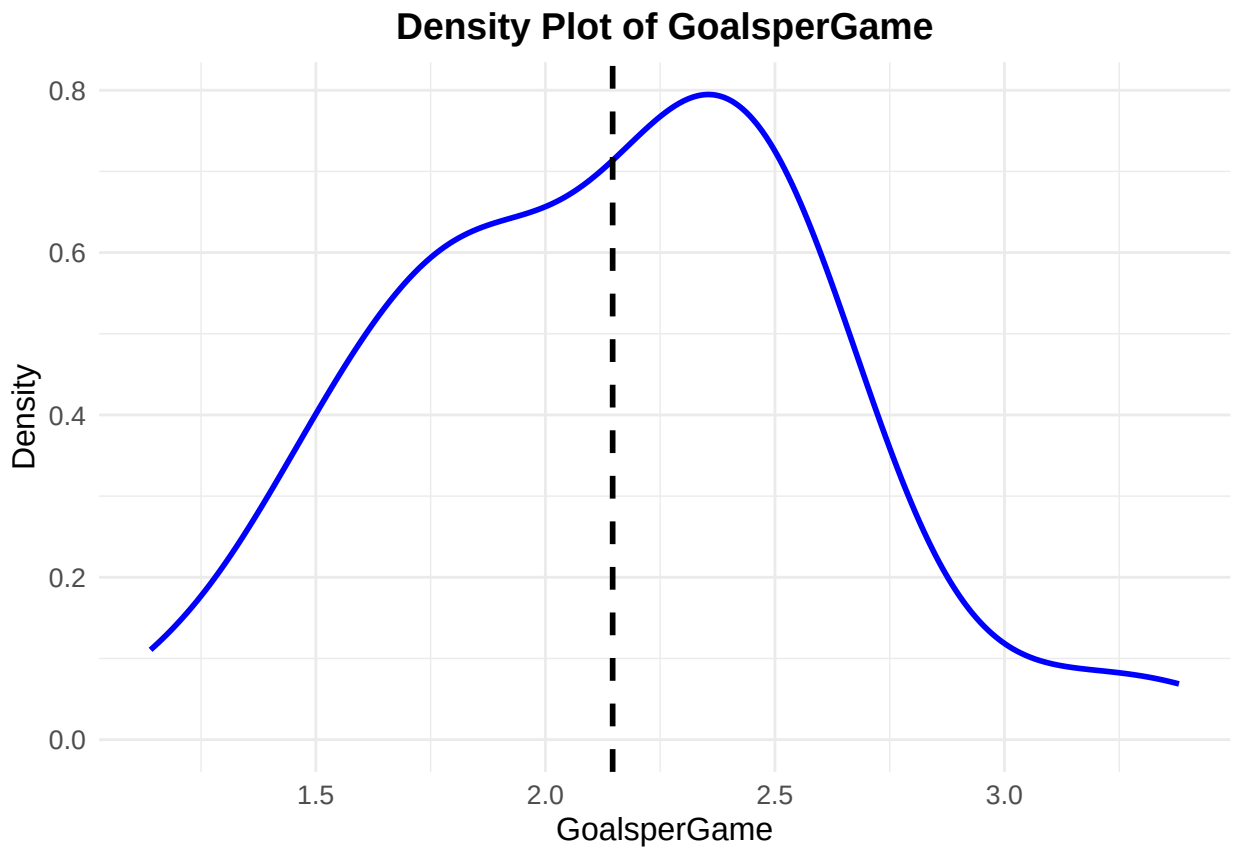
```
## P-value: 0.541
## R-squared: 0.009
## Adjusted R-squared: -0.015
##
## === Results for DemosAgainstperGame ===
## T-statistic: -1.009
## P-value: 0.319
## R-squared: 0.024
## Adjusted R-squared: 0
##
## === Results for Division ===
## T-statistic: 0.011
## P-value: 0.991
## R-squared: 0
## Adjusted R-squared: -0.024
```

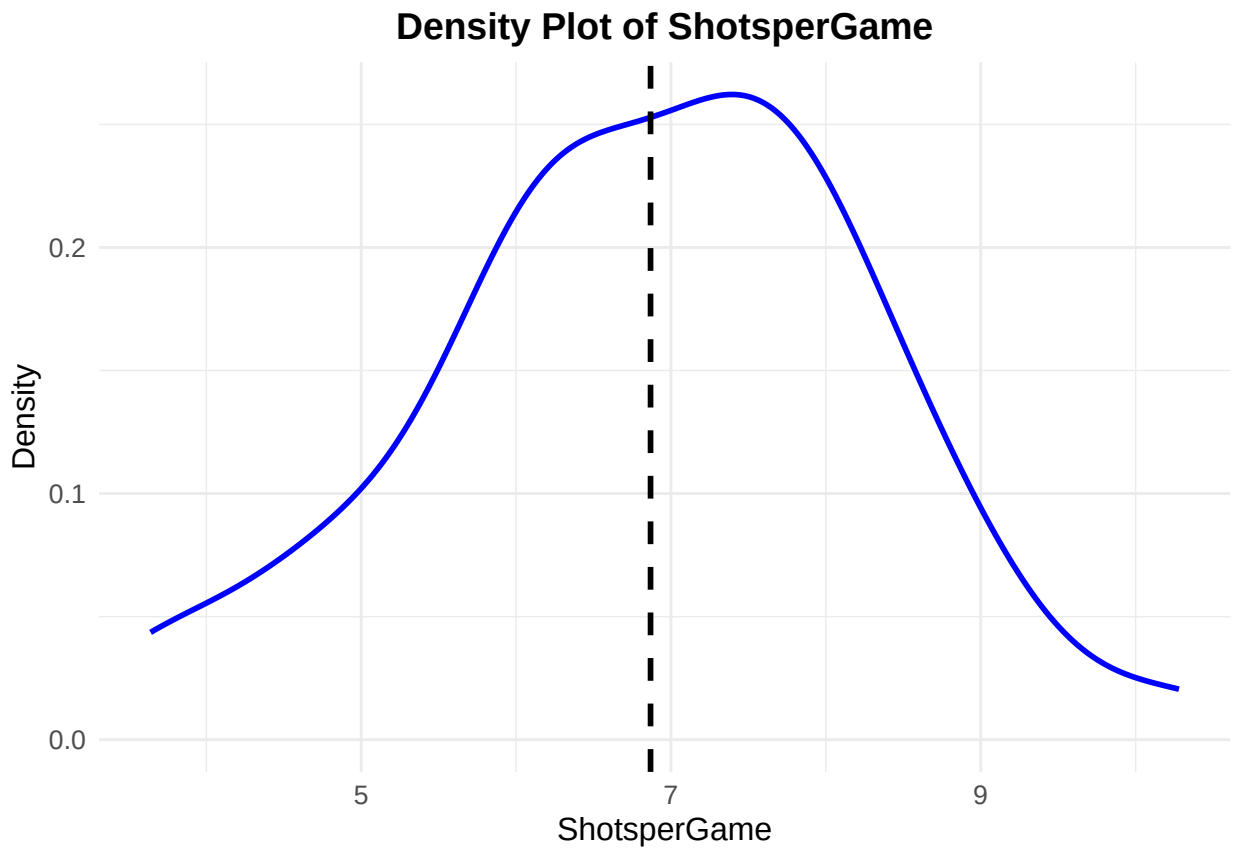
```
# Density Plots

# List of variables to plot
variables <- c(
  "WinPercent", "GoalsperGame", "ShotsperGame", "ShootingPercent",
  "AssistsperGame", "SavesperGame", "GoalsAgainstperGame",
  "ShotsAgainstperGame", "SavePercent", "DemosperGame", "DemosAgainstperGame"
)

# Loop through variables to create density plots
for (var in variables) {
  p <- ggplot(data, aes_string(x = var)) +  # Use aes_string for variable names
    geom_density(color = "blue", linewidth = 1) +  # Outline-only density plot
    geom_vline(
      aes(xintercept = mean(.data[[var]])),
      color = "black",  # Black mean line
      linewidth = 1,
      linetype = "dashed"  # Dashed line for clarity
    ) +
    labs(
      title = paste("Density Plot of", var),
      x = var,
      y = "Density"
    ) +
    theme_minimal() +  # Clean and modern theme
    theme(
      plot.title = element_text(hjust = 0.5, size = 14, face = "bold"),  # Center and style title
      axis.title = element_text(size = 12),  # Increase axis title size
      axis.text = element_text(size = 10)   # Adjust axis text size
    )

  # Print each plot
  print(p)
}
```

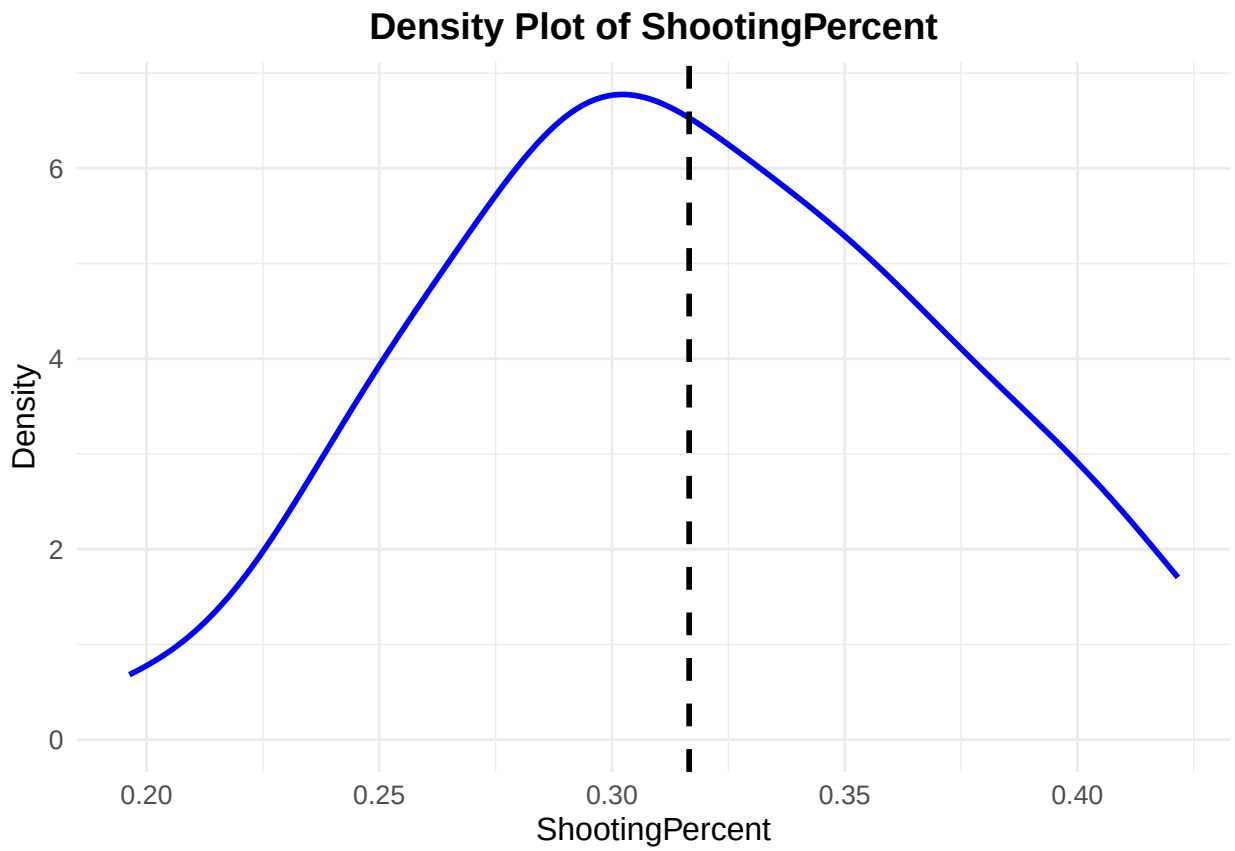
```
## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
```

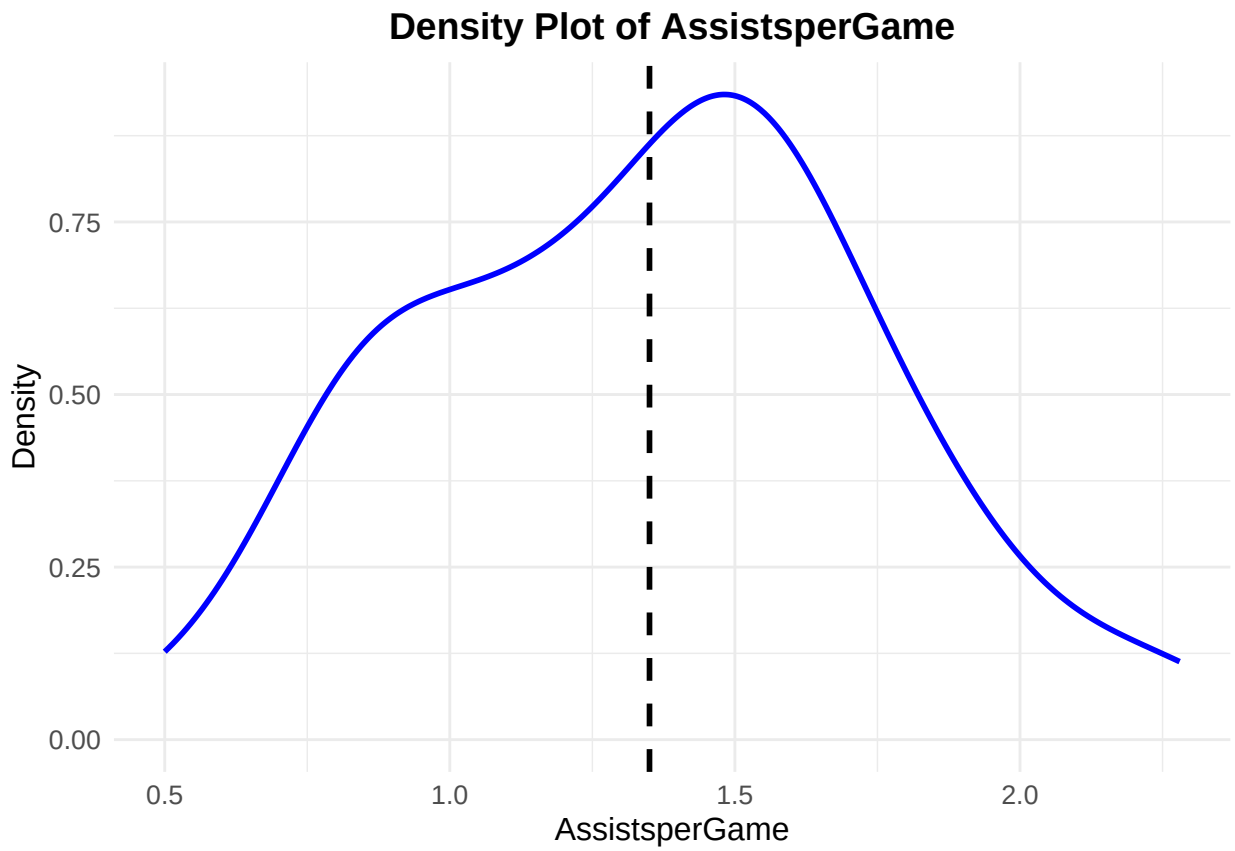
```
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```

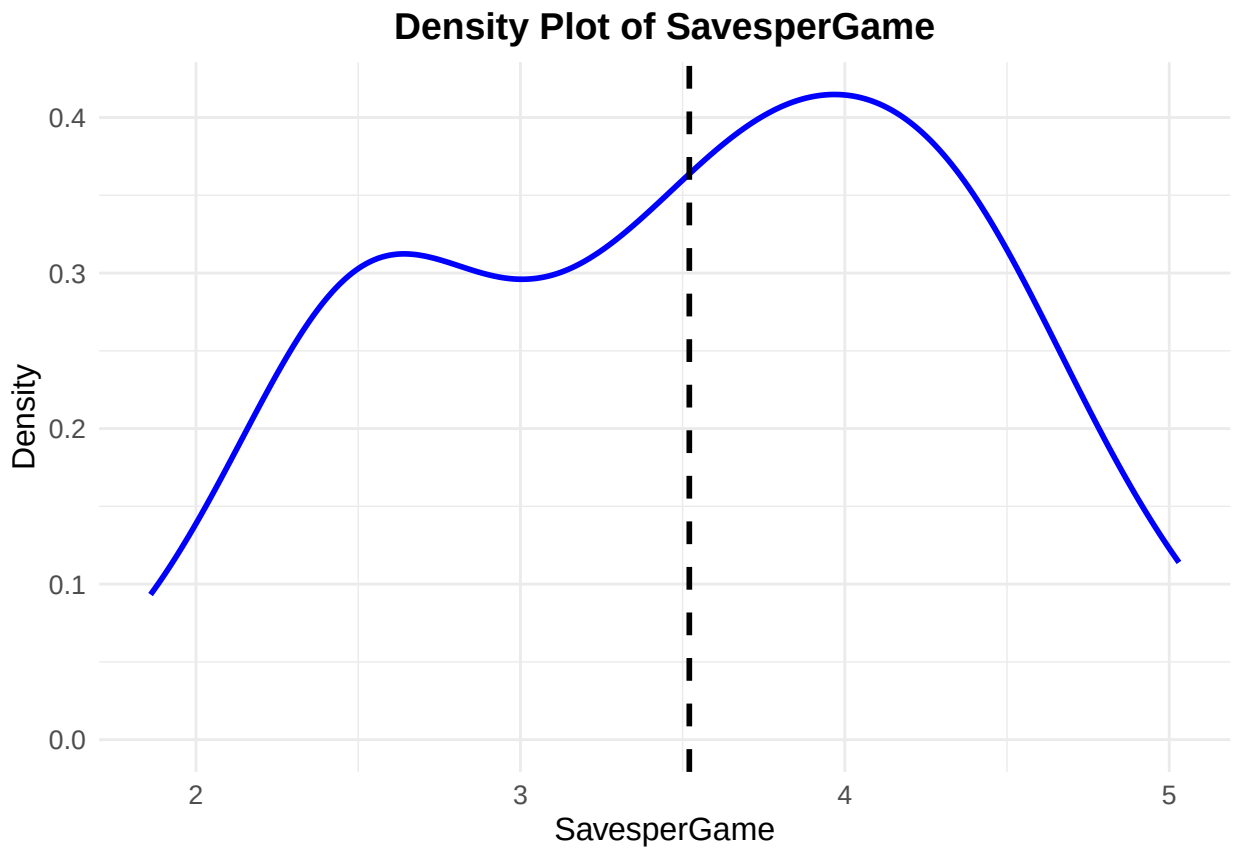


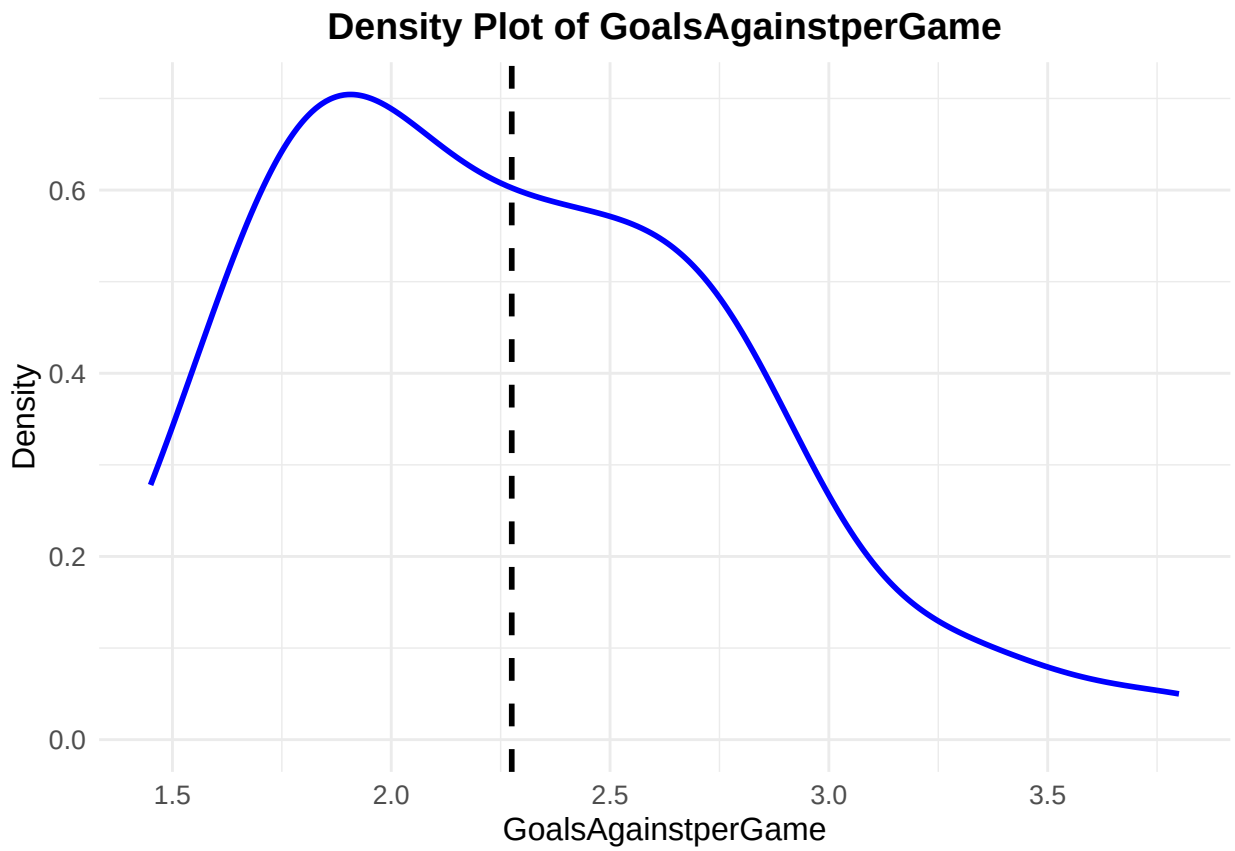

Density Plot of GoalsperGame
0.8
0.6
0.4
0.2
0.0
Density
1.5
2.0
2.5
3.0
GoalsperGame

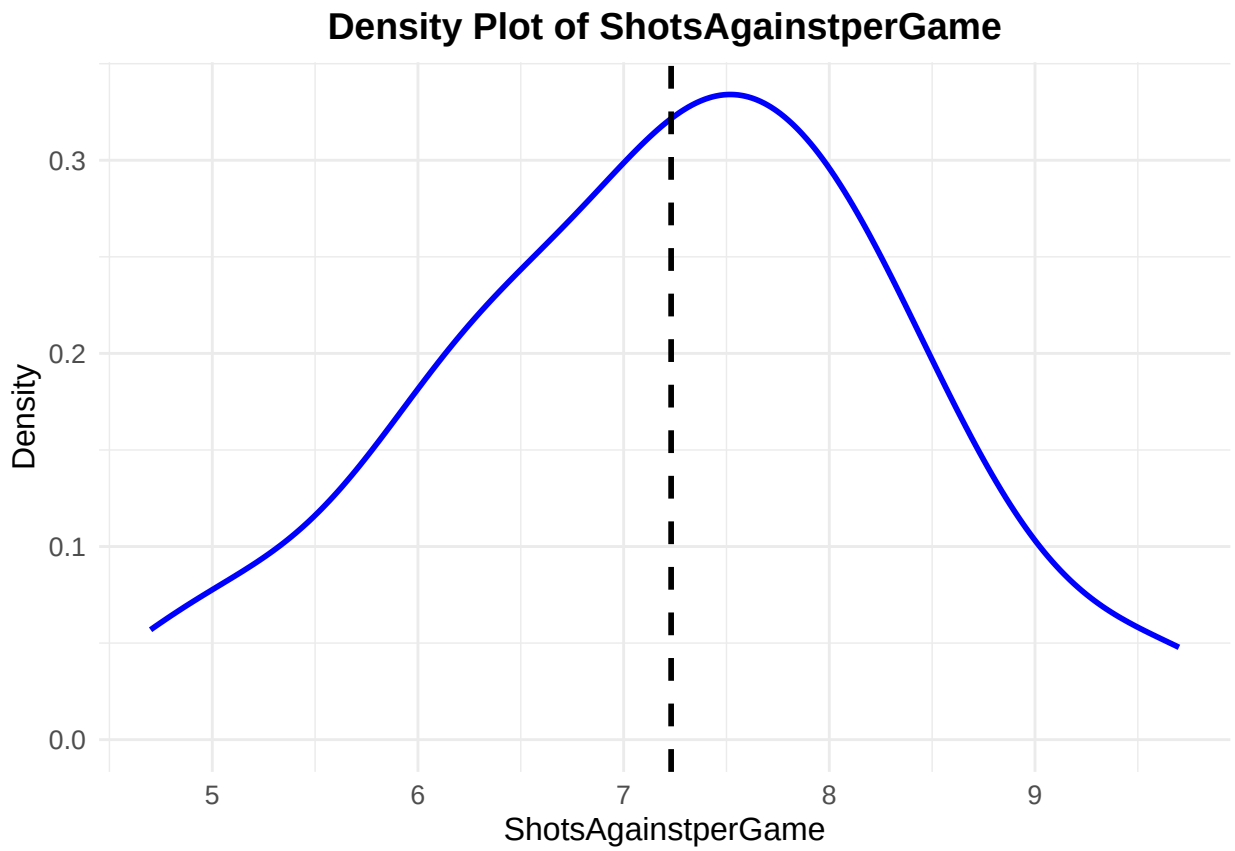

Density Plot of ShotsperGame
Density
0.2
0.1
0.0
5
7
9
ShotsperGame

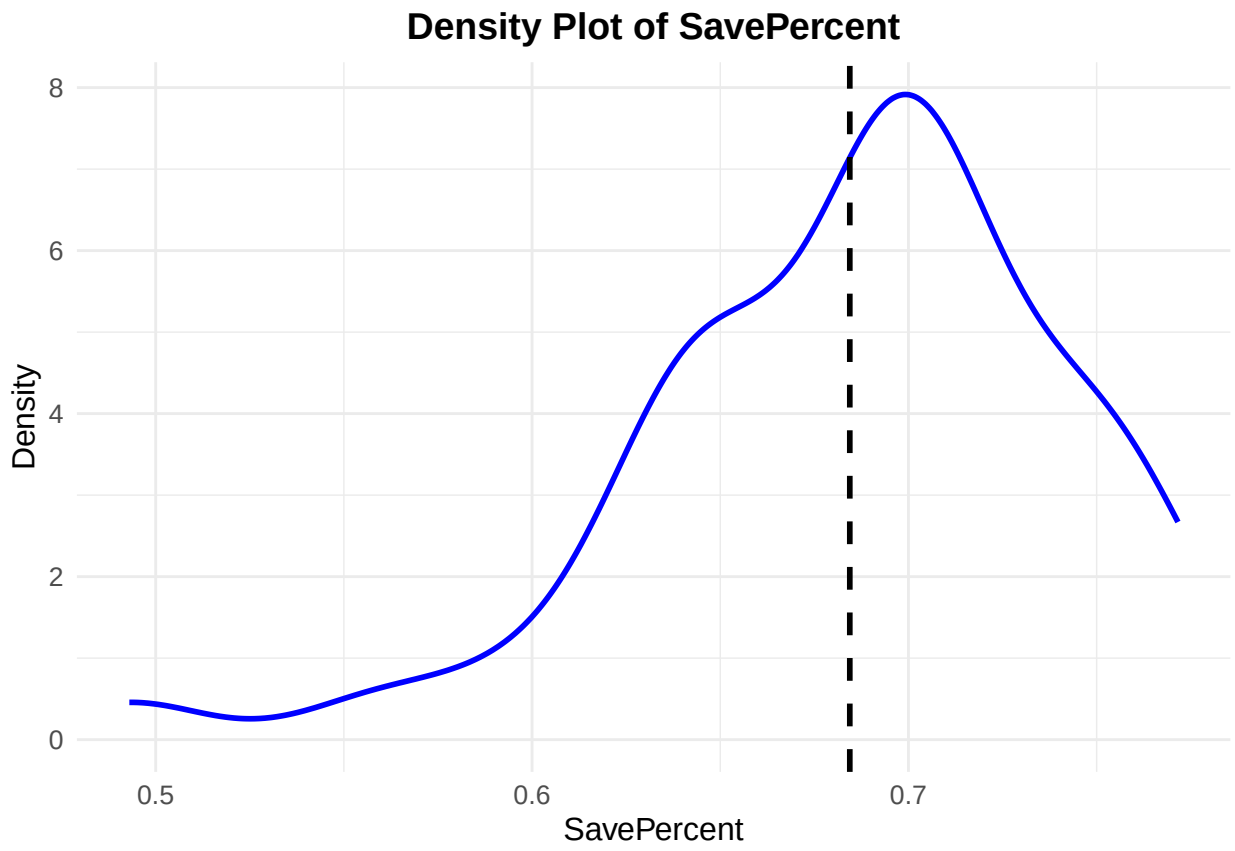

Density Plot of ShootingPercent
Density
0
2
4
6
0.20
0.25
0.30
0.35
0.40
ShootingPercent

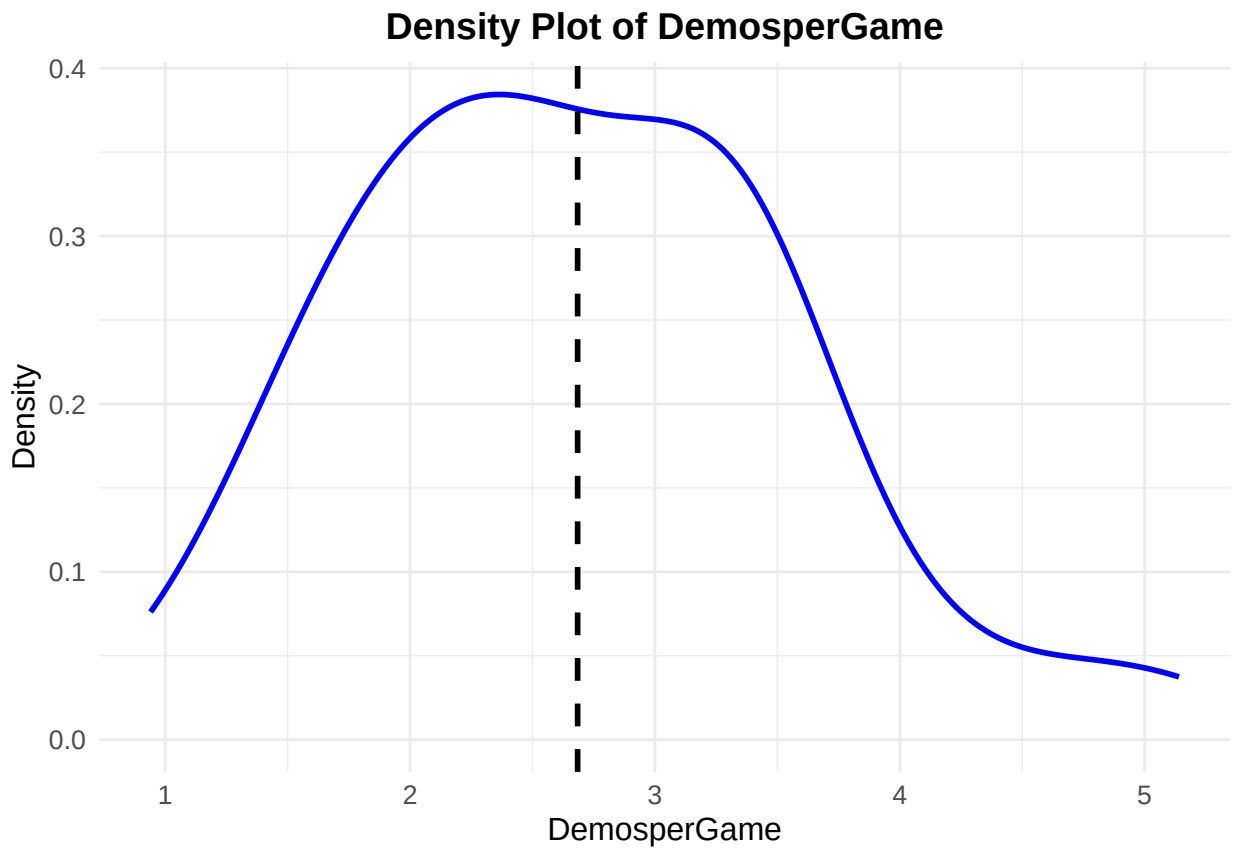

Density Plot of AssistsperGame
Density
0.75
0.50
0.25
0.00
0.5
1.0
1.5
2.0
AssistsperGame

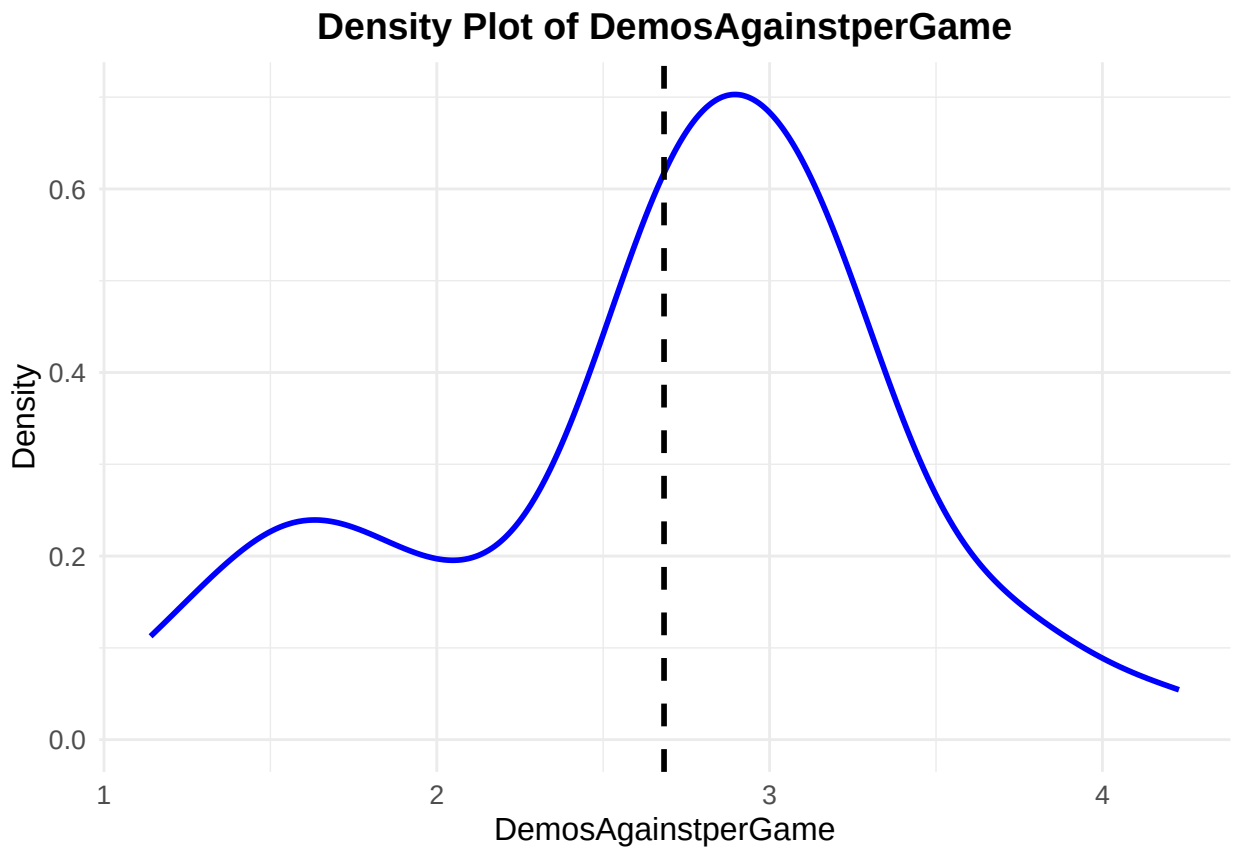

Density Plot of SavesperGame
0.4
0.3
0.2
0.1
0.0
Density
2
3
4
5
SavesperGame


Density Plot of GoalsAgainstperGame
Density
0.0
0.2
0.4
0.6
1.5
2.0
2.5
3.0
3.5
GoalsAgainstperGame


Density Plot of ShotsAgainstperGame
Density
0.0
0.1
0.2
0.3
5
6
7
8
9
ShotsAgainstperGame


Density Plot of SavePercent
Density
8
6
4
2
0
0.5
0.6
0.7
SavePercent


Density Plot of DemosperGame
0.4
0.3
0.2
0.1
0.0
Density
1
2
3
4
5
DemosperGame



Summary Statistics

```
# Display summary statistics for data
cat("Win Percentage:\n")
```

```
## Win Percentage:
```

```
summary(data$WinPercent)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.2143  0.3745  0.4722  0.4974  0.5882  0.8276
```

```
cat("Goals/Game:\n")
```

```
## Goals/Game:
```

```
summary(data$GoalsperGame)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.140   1.805   2.230   2.147   2.470   3.380
```

```r
cat("Goals Against/Game:\n")
```

```
## Goals Against/Game:
```

```r
summary(data$GoalsAgainstperGame)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.450   1.875   2.250   2.275   2.670   3.800
```

```r
cat("Shots/Game:\n")
```

```
## Shots/Game:
```

```r
summary(data$ShotsperGame)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.640   6.050   7.000   6.868   7.800  10.280
```

```r
cat("Shots Against/Game:\n")
```

```
## Shots Against/Game:
```

```r
summary(data$ShotsAgainstperGame)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   4.700   6.490   7.260   7.231   8.020   9.700
```

```r
cat("Shooting Percentage:\n")
```

```
## Shooting Percentage:
```

```r
summary(data$ShootingPercent)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.1963  0.2826  0.3106  0.3166  0.3562  0.4216
```

```r
cat("Assists/Game:\n")
```

```
## Assists/Game:
```

```r
summary(data$AssistsperGame)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.50    1.07    1.42    1.35    1.60    2.28
```

```r
cat("Saves/Game:\n")
```

```
## Saves/Game:
```

```r
summary(data$SavesperGame)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.860   2.695   3.650   3.521   4.170   5.030
```

```r
cat("Save Percentage:\n")
```

```
## Save Percentage:
```

```r
summary(data$SavePercent)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##  0.4930  0.6498  0.6957  0.6844  0.7143  0.7716
```

```r
cat("Demos/Game:\n")
```

```
## Demos/Game:
```

```r
summary(data$DemosperGame)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.940   2.040   2.590   2.684   3.310   5.140
```

```r
cat("Demos Against/Game:\n")
```

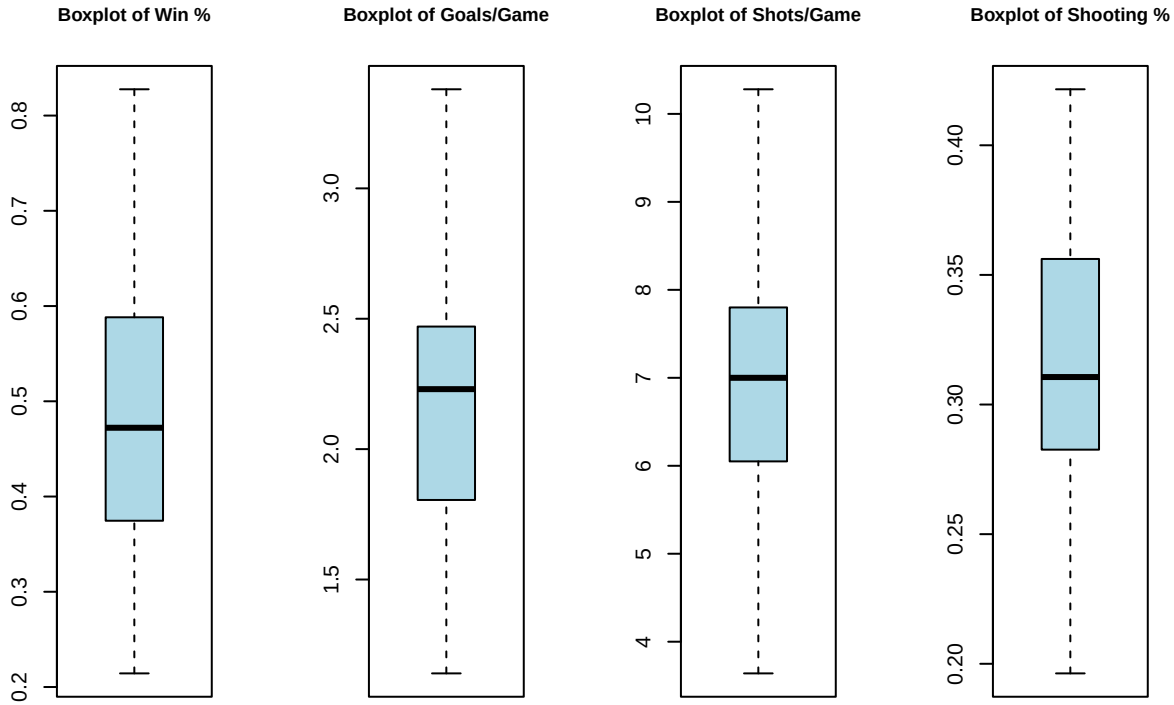
```
## Demos Against/Game:
```

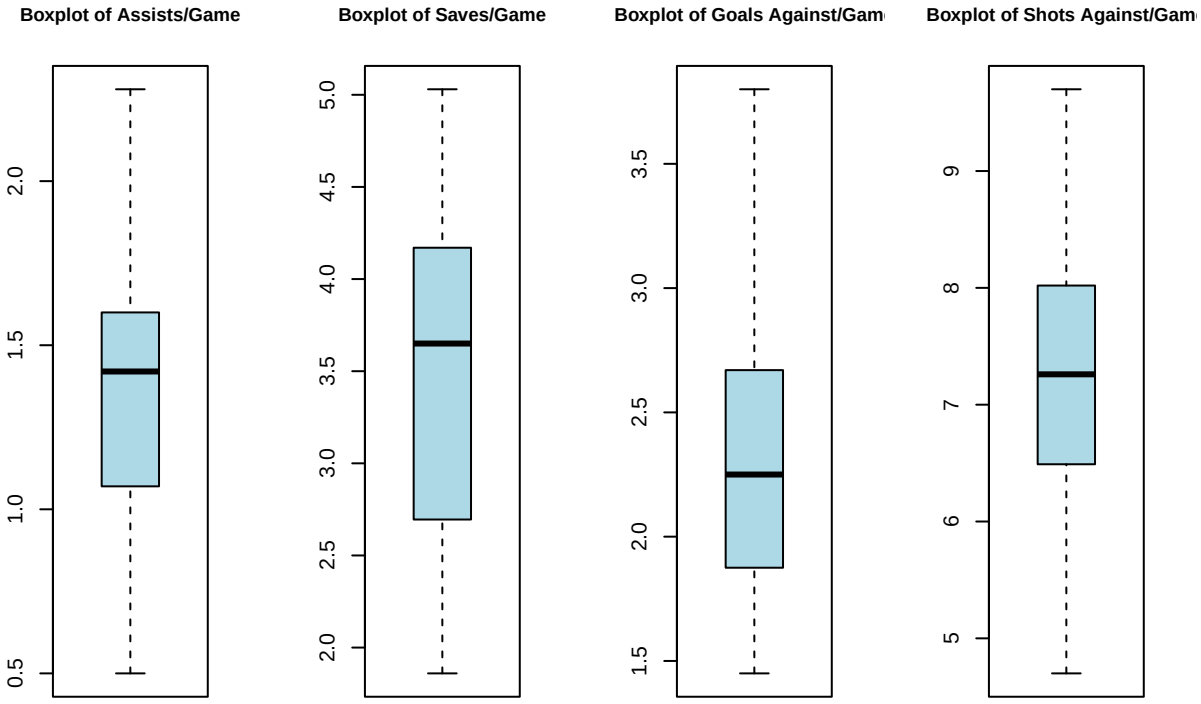
```r
summary(data$DemosAgainstperGame)
```

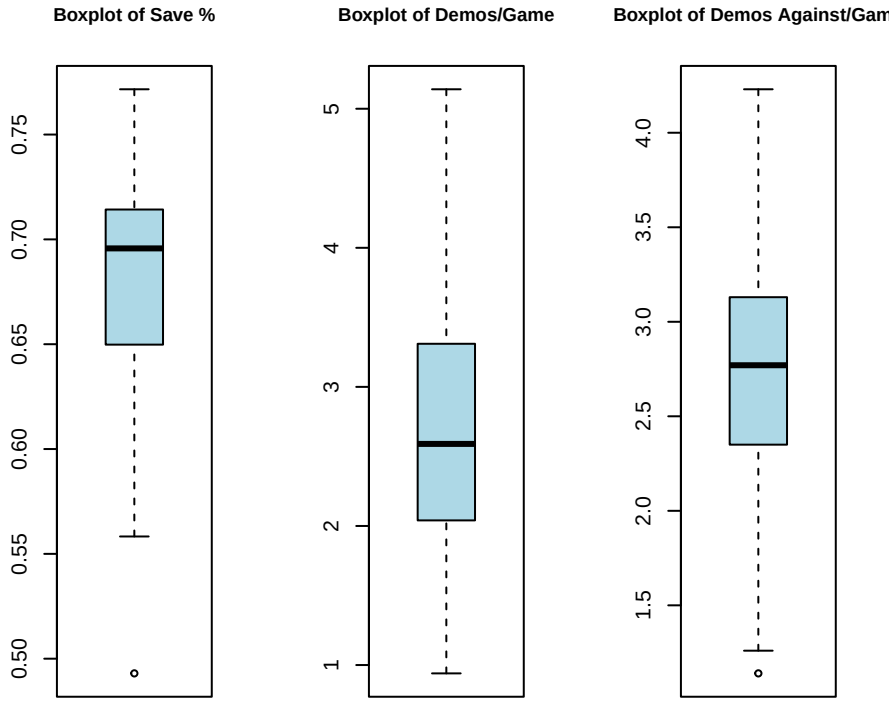
```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.140   2.350   2.770   2.683   3.130   4.230
```

```r
# List of variable names for titles and labels
par(mfrow = c(1,4))
variable_names <- c("Win %",
                    "Goals/Game",
                    "Shots/Game",
                    "Shooting %",
                    "Assists/Game",
                    "Saves/Game",
                    "Goals Against/Game",
                    "Shots Against/Game",
                    "Save %",
```

```
                    "Demos/Game",
                    "Demos Against/Game")

# Display the boxplots for the variables
for (i in 1:length(variable_names)) {
  boxplot(data[[i]],
          main = paste("Boxplot of", variable_names[i]),
          cex.main = 0.85,
          col = "lightblue")
}
```

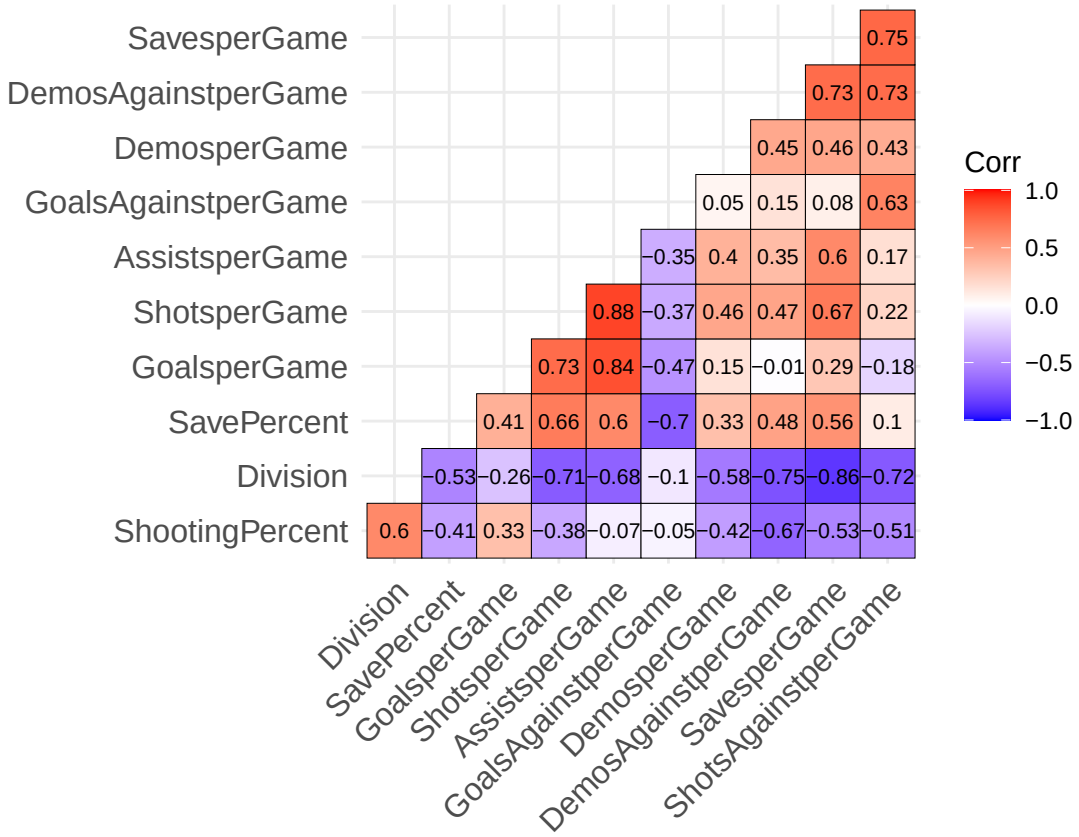



Boxplot of Assists/Game
0.5
1.0
1.5
2.0
Boxplot of Saves/Game
2.0
2.5
3.0
3.5
4.0
4.5
5.0
Boxplot of Goals Against/Gam
1.5
2.0
2.5
3.0
3.5
Boxplot of Shots Against/Gam
5
6
7
8
9



Addressing Multicollinearity

```
# Correlation matrix excluding response variable
corr <- round(cor(data %>% select(-WinPercent)), 2)

# Create the correlation plot with ggplot
ggcorrplot(corr, hc.order = TRUE, type = "lower", outline.col = "black",
           lab = TRUE, lab_size = 2.75)
```




```
# Use Stepwise Selection Method to generate best model
data <- data %>% select(-Division)
# Full Model
model.all <- lm(WinPercent ~ ., data = data)

# Forwards Model
model.forwards.olsrr <- ols_step_forward_p(model.all, trace = 0)
print(model.forwards.olsrr)
```

```
##
##
##                                 Stepwise Summary
## -------------------------------------------------------------------------------------
## Step    Variable                 AIC         SBC         SBIC        R2       Adj. R2
## -------------------------------------------------------------------------------------
##  0      Base Model              -40.569     -37.047     -165.897    0.00000    0.00000
##  1      GoalsAgainstperGame     -93.727     -88.443     -218.630    0.72272    0.71596
##  2      GoalsperGame           -118.247    -111.203     -241.467    0.85036    0.84287
##  3      SavesperGame           -123.406    -114.600     -245.754    0.87331    0.86356
##  4      DemosperGame           -129.552    -118.985     -250.024    0.89517    0.88414
##  5      AssistsperGame         -130.915    -118.586     -250.010    0.90306    0.88996
##  6      ShotsAgainstperGame    -130.941    -116.852     -248.747    0.90752    0.89211
## -------------------------------------------------------------------------------------
##
## Final Model Output
## ------------------
```

```
##
##                          Model Summary
## ----------------------------------------------------------------
## R                       0.953       RMSE                 0.044
## R-Squared               0.908       MSE                  0.002
## Adj. R-Squared          0.892       Coef. Var            9.630
## Pred R-Squared          0.847       AIC               -130.941
## MAE                     0.035       SBC               -116.852
## ----------------------------------------------------------------
##  RMSE: Root Mean Square Error
##  MSE: Mean Square Error
##  MAE: Mean Absolute Error
##  AIC: Akaike Information Criteria
##  SBC: Schwarz Bayesian Criteria
##
##                                ANOVA
## -------------------------------------------------------------------
##                Sum of
##               Squares        DF    Mean Square      F         Sig.
## -------------------------------------------------------------------
## Regression      0.810         6          0.135    58.878    0.0000
## Residual        0.083        36          0.002
## Total           0.893        42
## -------------------------------------------------------------------
##
##                                    Parameter Estimates
## ----------------------------------------------------------------------------------------------------
##               model      Beta    Std. Error    Std. Beta      t        Sig      lower      upper
## ----------------------------------------------------------------------------------------------------
##         (Intercept)     0.633         0.087                 7.291    0.000     0.457      0.810
## GoalsAgainstperGame    -0.199         0.030       -0.722    -6.743    0.000    -0.259     -0.139
##        GoalsperGame     0.115         0.039        0.372     2.949    0.006     0.036      0.194
##        SavesperGame    -0.084         0.025       -0.477    -3.339    0.002    -0.135     -0.033
##        DemosperGame     0.018         0.010        0.112     1.799    0.080    -0.002      0.038
##      AssistsperGame     0.073         0.051        0.202     1.437    0.159    -0.030      0.176
## ShotsAgainstperGame     0.030         0.023        0.240     1.318    0.196    -0.016      0.077
## ----------------------------------------------------------------------------------------------------
```

```
# Backwards Model
model.backwards.olsrr <- ols_step_backward_p(model.all, trace = 0)
print(model.backwards.olsrr)
```

```
##
##
##                                  Stepwise Summary
## ------------------------------------------------------------------------------------
## Step    Variable                 AIC         SBC         SBIC        R2       Adj. R2
## ------------------------------------------------------------------------------------
##  0      Full Model            -123.843    -102.708    -238.545    0.90944    0.88114
##  1      SavePercent           -125.800    -106.426    -241.215    0.90935    0.88462
##  2      GoalsperGame          -127.650    -110.038    -243.828    0.90903    0.88763
##  3      DemosAgainstperGame   -129.293    -113.442    -246.308    0.90827    0.88993
## ------------------------------------------------------------------------------------
##
```

```
## Final Model Output
## ------------------
##
##                            Model Summary
## ----------------------------------------------------------------
## R                        0.953       RMSE                 0.044
## R-Squared                0.908       MSE                  0.002
## Adj. R-Squared           0.890       Coef. Var            9.727
## Pred R-Squared           0.841       AIC                -129.293
## MAE                      0.035       SBC                -113.442
## ----------------------------------------------------------------
##  RMSE: Root Mean Square Error
##  MSE: Mean Square Error
##  MAE: Mean Absolute Error
##  AIC: Akaike Information Criteria
##  SBC: Schwarz Bayesian Criteria
##
##                                ANOVA
## --------------------------------------------------------------------
##                Sum of
##               Squares        DF    Mean Square      F         Sig.
## --------------------------------------------------------------------
## Regression      0.811         7          0.116    49.509    0.0000
## Residual        0.082        35          0.002
## Total           0.893        42
## --------------------------------------------------------------------
##
##                                    Parameter Estimates
## ----------------------------------------------------------------------------------------------------
##               model      Beta    Std. Error    Std. Beta      t        Sig      lower     upper
## ----------------------------------------------------------------------------------------------------
##         (Intercept)     0.385         0.161                 2.385    0.023     0.057     0.712
##        ShotsperGame     0.039         0.017        0.381    2.227    0.032     0.003     0.074
##     ShootingPercent     0.769         0.262        0.279    2.934    0.006     0.237     1.302
##      AssistsperGame     0.067         0.057        0.186    1.192    0.241    -0.047     0.182
##        SavesperGame    -0.086         0.027       -0.490   -3.158    0.003    -0.141    -0.031
## GoalsAgainstperGame    -0.211         0.031       -0.767   -6.729    0.000    -0.275    -0.148
## ShotsAgainstperGame     0.035         0.024        0.275    1.450    0.156    -0.014     0.083
##        DemosperGame     0.018         0.010        0.109    1.675    0.103    -0.004     0.039
## ----------------------------------------------------------------------------------------------------
```

```r
# Both Model
model.both.olsrr <- ols_step_both_p(model.all, trace = 0)
print(model.both.olsrr)
```

```
##
##
##                                    Stepwise Summary
## ------------------------------------------------------------------------------------------
## Step    Variable                     AIC         SBC         SBIC        R2       Adj. R2
## ------------------------------------------------------------------------------------------
##  0      Base Model                -40.569     -37.047    -165.897    0.00000    0.00000
##  1      GoalsAgainstperGame (+)   -93.727     -88.443    -218.630    0.72272    0.71596
##  2      GoalsperGame (+)          -118.247    -111.203    -241.467    0.85036    0.84287
```

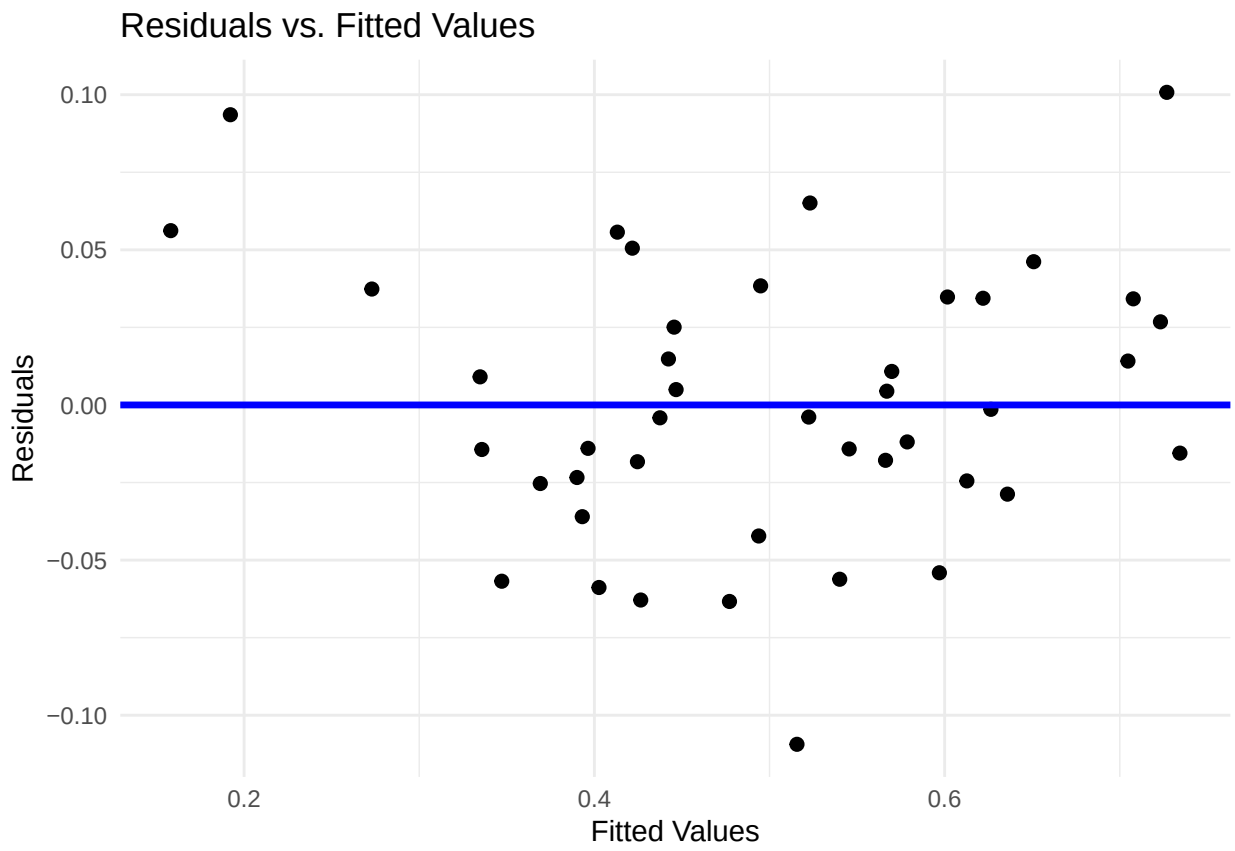
```
##  3      SavesperGame (+)           -123.406    -114.600    -245.754    0.87331    0.86356
##  4      DemosperGame (+)           -129.552    -118.985    -250.024    0.89517    0.88414
##  5      AssistsperGame (+)         -130.915    -118.586    -250.010    0.90306    0.88996
## ---------------------------------------------------------------------------------------
##
## Final Model Output
## ------------------
##
##                          Model Summary
## ------------------------------------------------------------------
## R                       0.950       RMSE                  0.045
## R-Squared               0.903       MSE                   0.002
## Adj. R-Squared          0.890       Coef. Var             9.725
## Pred R-Squared          0.855       AIC                -130.915
## MAE                     0.036       SBC                -118.586
## ------------------------------------------------------------------
##  RMSE: Root Mean Square Error
##  MSE: Mean Square Error
##  MAE: Mean Absolute Error
##  AIC: Akaike Information Criteria
##  SBC: Schwarz Bayesian Criteria
##
##                                ANOVA
## ---------------------------------------------------------------------
##                Sum of
##               Squares        DF    Mean Square      F         Sig.
## ---------------------------------------------------------------------
## Regression      0.806         5          0.161    68.933    0.0000
## Residual        0.087        37          0.002
## Total           0.893        42
## ---------------------------------------------------------------------
##
##                                    Parameter Estimates
## ------------------------------------------------------------------------------------------------
##               model      Beta    Std. Error    Std. Beta       t        Sig      lower     upper
## ------------------------------------------------------------------------------------------------
##         (Intercept)     0.700         0.071                  9.801    0.000     0.555     0.845
## GoalsAgainstperGame    -0.167         0.017       -0.605    -10.003    0.000    -0.201    -0.133
##        GoalsperGame     0.093         0.036        0.301      2.613    0.013     0.021     0.165
##        SavesperGame    -0.056         0.013       -0.317     -4.164    0.000    -0.083    -0.029
##        DemosperGame     0.021         0.010        0.130      2.130    0.040     0.001     0.041
##      AssistsperGame     0.087         0.050        0.240      1.735    0.091    -0.015     0.189
## ------------------------------------------------------------------------------------------------
```

```r
# Best Possible Model
model.best <- ols_step_best_subset(model.all, trace = 0)
print(model.best)
```

```
##                                                                  Best Subsets Regression
## ---------------------------------------------------------------------------------------------------------
## Model Index    Predictors
## ---------------------------------------------------------------------------------------------------------
##      1         GoalsAgainstperGame
##      2         GoalsperGame GoalsAgainstperGame
```

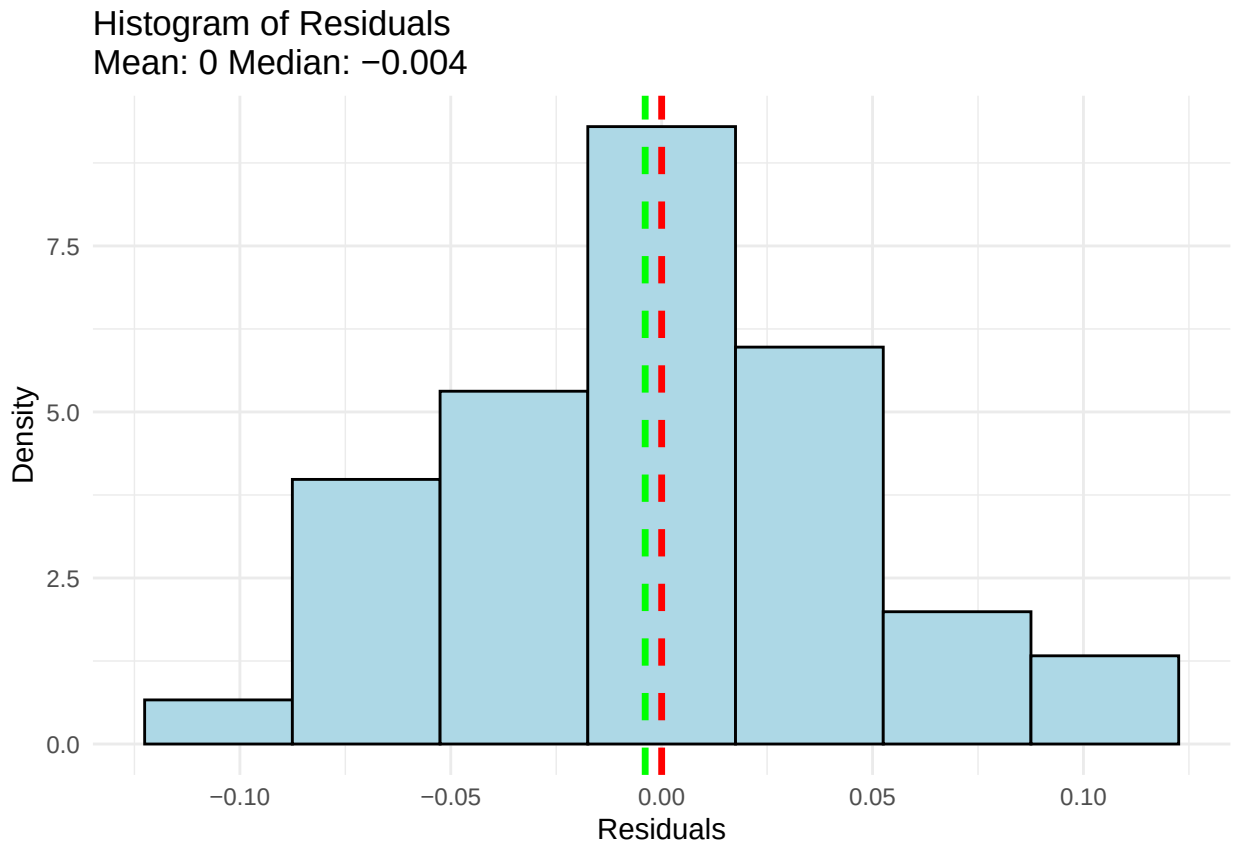
```
##      3         GoalsperGame SavesperGame SavePercent
##      4         GoalsperGame SavesperGame GoalsAgainstperGame DemosperGame
##      5         GoalsperGame AssistsperGame SavesperGame GoalsAgainstperGame DemosperGame
##      6         GoalsperGame AssistsperGame SavesperGame GoalsAgainstperGame ShotsAgainstperGame Demos
##      7         ShotsperGame ShootingPercent AssistsperGame SavesperGame GoalsAgainstperGame SavePerc
##      8         ShotsperGame ShootingPercent AssistsperGame SavesperGame GoalsAgainstperGame ShotsAga
##      9         GoalsperGame ShotsperGame ShootingPercent AssistsperGame SavesperGame GoalsAgainstper
##     10         GoalsperGame ShotsperGame ShootingPercent AssistsperGame SavesperGame GoalsAgainstper
## ----------------------------------------------------------------------------------------------------
##
##                                                Subsets Regression Summary
## ----------------------------------------------------------------------------------------------------
##                        Adj.        Pred
## Model    R-Square    R-Square    R-Square     C(p)         AIC         SBIC          SBC         MSEP
## ----------------------------------------------------------------------------------------------------
##   1        0.7227      0.7160      0.6844    58.9747     -93.7267    -218.6296     -88.4431    0.259
##   2        0.8504      0.8429      0.8192    15.8760    -118.2474    -241.4666    -111.2026    0.143
##   3        0.8840      0.8751      0.8564     5.9931    -127.1929    -248.8347    -118.3869    0.114
##   4        0.8952      0.8841      0.8513     4.0408    -129.5523    -250.0240    -118.9851    0.106
##   5        0.9031      0.8900      0.8555     3.2548    -130.9146    -250.0101    -118.5862    0.100
##   6        0.9075      0.8921      0.8467     3.6779    -130.9412    -248.7472    -116.8516    0.099
##   7        0.9084      0.8901      0.8436     5.3746    -129.3421    -246.3350    -113.4913    0.101
##   8        0.9090      0.8876      0.8287     7.1434    -127.6503    -243.8276    -110.0383    0.103
##   9        0.9093      0.8846      0.8026     9.0320    -125.7996    -241.2146    -106.4264    0.106
##  10        0.9094      0.8811       0.791    11.0000    -123.8426    -238.5451    -102.7082    0.109
## ----------------------------------------------------------------------------------------------------
## AIC: Akaike Information Criteria
##  SBIC: Sawa's Bayesian Information Criteria
##  SBC: Schwarz Bayesian Criteria
##  MSEP: Estimated error of prediction, assuming multivariate normality
##  FPE: Final Prediction Error
##  HSP: Hocking's Sp
##  APC: Amemiya Prediction Criteria
```

```r
# Generate residual plots
model.final <- lm(WinPercent ~ GoalsAgainstperGame + GoalsperGame + SavesperGame + DemosperGame + Assis

# Residuals vs Fitted Values Plot
ggplot(data = data.frame(
  Fitted = model.final$fitted.values,
  Residuals = model.final$residuals
), aes(x = Fitted, y = Residuals)) +
  geom_point(color = "black", size = 2) +
  geom_hline(yintercept = 0, color = "blue", linetype = "solid", linewidth = 1.2) +
  labs(
    title = "Residuals vs. Fitted Values",
    x = "Fitted Values",
    y = "Residuals"
  ) +
  theme_minimal()
```

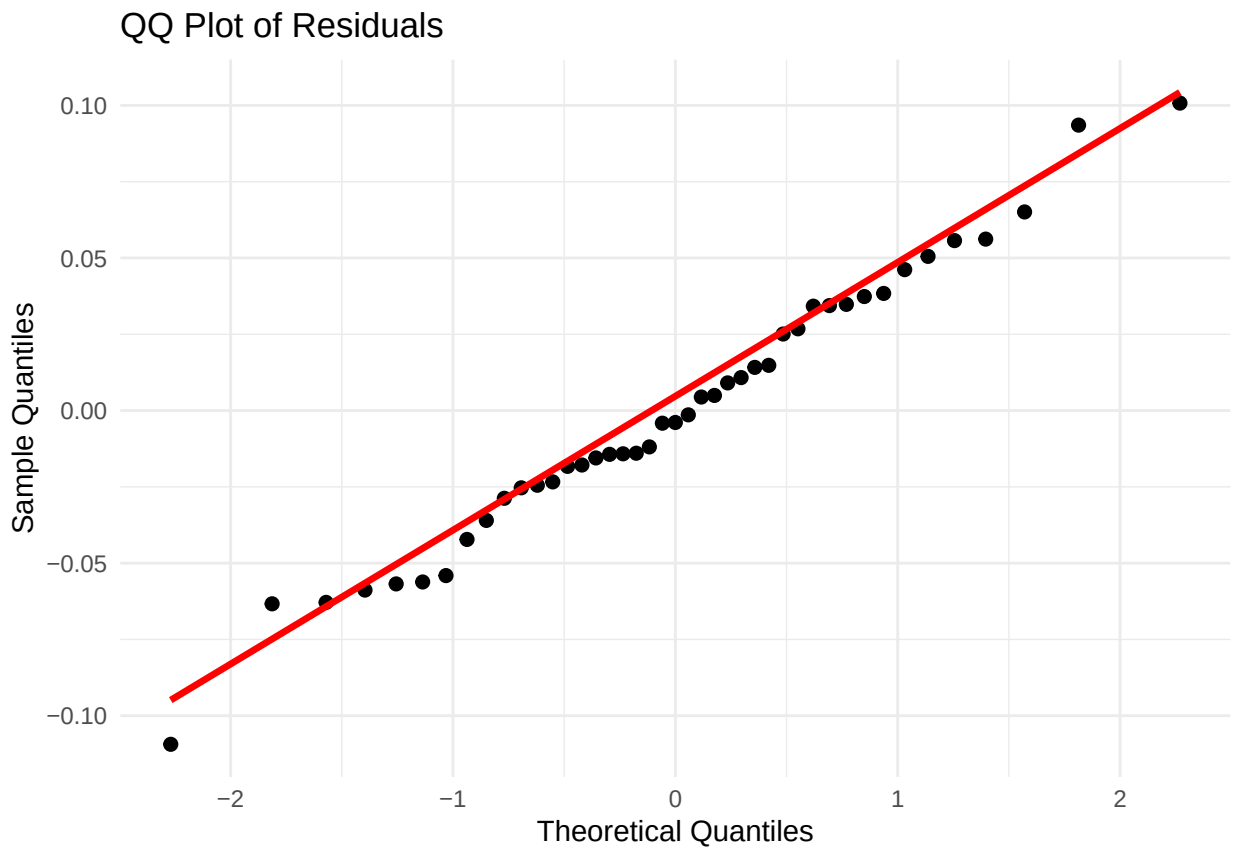



```
# Histogram of Residuals with Mean Line
ggplot(data = data.frame(Residuals = model.final$residuals), aes(x = Residuals)) +
  geom_histogram(aes(y = ..density..), bins = 7, fill = "lightblue", color = "black") +
  geom_vline(aes(xintercept = mean(Residuals)), color = "red", linetype = "dashed", linewidth = 1.2) +
  geom_vline(aes(xintercept = median(Residuals)), color = "green", linetype = "dashed", linewidth = 1.2
  labs(
    title = paste(
      "Histogram of Residuals\nMean:", round(mean(model.final$residuals), 3),
      "Median:", round(median(model.final$residuals), 3)
    ),
    x = "Residuals",
    y = "Density"
  ) +
  theme_minimal()
```

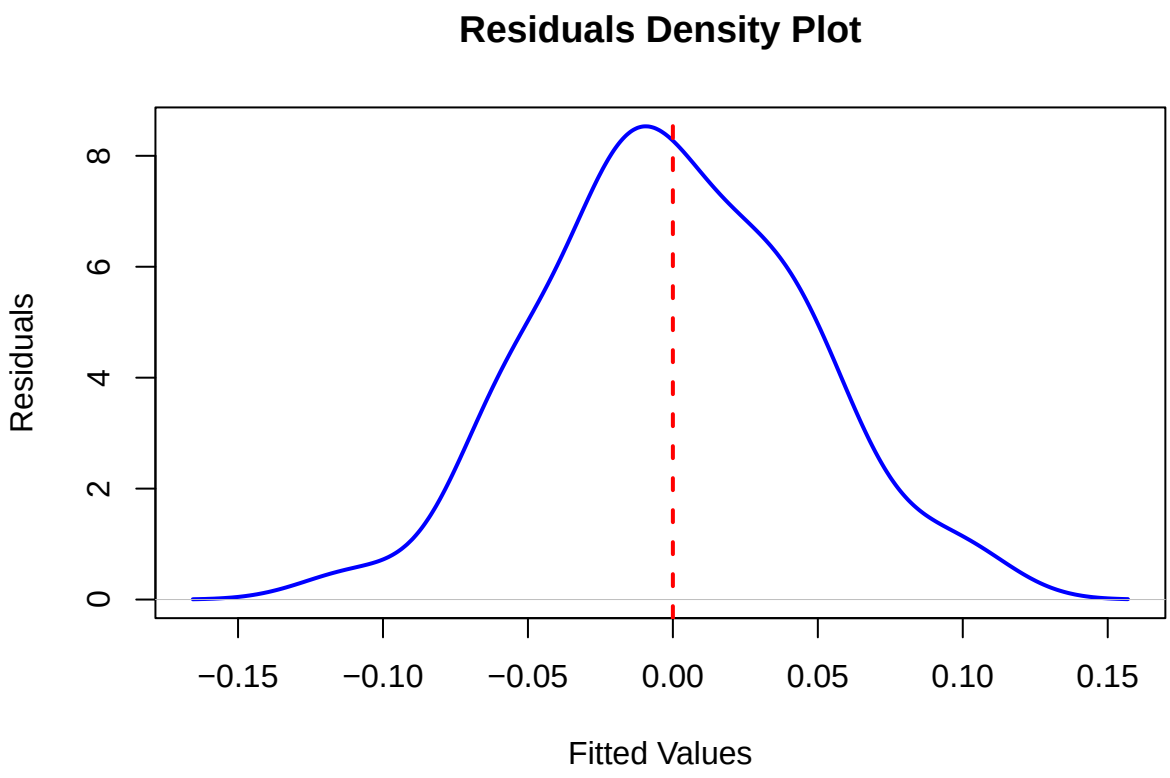
```
## Warning: The dot-dot notation (‘..density..‘) was deprecated in ggplot2 3.4.0.
## i Please use ‘after_stat(density)‘ instead.
## This warning is displayed once every 8 hours.
## Call ‘lifecycle::last_lifecycle_warnings()‘ to see where this warning was
## generated.
```




```r
# QQ Plot of Residuals
ggplot(data = data.frame(Residuals = model.final$residuals), aes(sample = Residuals)) +
  stat_qq(color = "black", size = 2) +
  stat_qq_line(color = "red", linetype = "solid", linewidth = 1.2) +
  labs(
    title = "QQ Plot of Residuals",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  theme_minimal()
```




```
plot(density(model.final$residuals),
     main = "Residuals Density Plot",
     col = "blue", lwd = 2,
     xlab = "Fitted Values",
     ylab = "Residuals")
abline(v = mean(model.final$residuals), col = "red", lwd = 2, lty = 2)
```




```r
model.log <- lm(log(WinPercent + 1) ~ GoalsAgainstperGame + GoalsperGame + SavesperGame + DemosperGame +

# Residuals vs Fitted Values Plot
ggplot(data = data.frame(
  Fitted = model.log$fitted.values,
  Residuals = model.log$residuals
), aes(x = Fitted, y = Residuals)) +
  geom_point(color = "black", size = 2) +
  geom_hline(yintercept = 0, color = "blue", linetype = "solid", linewidth = 1.2) +
  labs(
    title = "Residuals vs. Fitted Values",
    x = "Fitted Values",
    y = "Residuals"
  ) +
  theme_minimal()

# Histogram of Residuals with Mean Line
ggplot(data = data.frame(Residuals = model.log$residuals), aes(x = Residuals)) +
  geom_histogram(aes(y = ..density..), bins = 7, fill = "lightblue", color = "black") +
  geom_vline(aes(xintercept = mean(Residuals)), color = "red", linetype = "dashed", linewidth = 1.2) +
  geom_vline(aes(xintercept = median(Residuals)), color = "green", linetype = "dashed", linewidth = 1.2)
  labs(
    title = paste(
      "Histogram of Residuals\nMean:", round(mean(model.log$residuals), 3),
      "Median:", round(median(model.log$residuals), 3)
    ),
```

```
    x = "Residuals",
    y = "Density"
  ) +
  theme_minimal()

# QQ Plot of Residuals
ggplot(data = data.frame(Residuals = model.log$residuals), aes(sample = Residuals)) +
  stat_qq(color = "black", size = 2) +
  stat_qq_line(color = "red", linetype = "solid", linewidth = 1.2) +
  labs(
    title = "QQ Plot of Residuals",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  theme_minimal()

plot(density(model.log$residuals),
     main = "Residuals Density Plot",
     col = "blue", lwd = 2,
     xlab = "Fitted Values",
     ylab = "Residuals")
abline(v = mean(model.log$residuals), col = "red", lwd = 2, lty = 2)
```

```
print(anova(model.final))
```

```
## Analysis of Variance Table
##
## Response: WinPercent
##                      Df  Sum Sq Mean Sq  F value    Pr(>F)
## GoalsAgainstperGame   1 0.64539 0.64539 281.3325 < 2.2e-16 ***
## GoalsperGame          1 0.11398 0.11398  49.6836 2.834e-08 ***
## SavesperGame          1 0.02049 0.02049   8.9336  0.005021 **
## DemosperGame          1 0.01953 0.01953   8.5114  0.006046 **
## AssistsperGame        1 0.00704 0.00704   3.0692  0.088304 .
## ShotsAgainstperGame   1 0.00399 0.00399   1.7373  0.195812
## Residuals            36 0.08259 0.00229
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
print(model.final)
```

```
##
## Call:
## lm(formula = WinPercent ~ GoalsAgainstperGame + GoalsperGame +
##     SavesperGame + DemosperGame + AssistsperGame + ShotsAgainstperGame,
##     data = data)
##
## Coefficients:
##         (Intercept)  GoalsAgainstperGame         GoalsperGame
##             0.63345             -0.19919              0.11485
##        SavesperGame         DemosperGame       AssistsperGame
##            -0.08389              0.01802              0.07308
```

```
## ShotsAgainstperGame
##             0.03027
```

```
print(summary(model.final))
```

```
##
## Call:
## lm(formula = WinPercent ~ GoalsAgainstperGame + GoalsperGame +
##     SavesperGame + DemosperGame + AssistsperGame + ShotsAgainstperGame,
##     data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.109372 -0.024902 -0.003904  0.034299  0.100759
##
## Coefficients:
##                     Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.63345    0.08688   7.291 1.37e-08 ***
## GoalsAgainstperGame -0.19919    0.02954  -6.743 7.15e-08 ***
## GoalsperGame         0.11485    0.03895   2.949  0.00557 **
## SavesperGame        -0.08389    0.02513  -3.339  0.00197 **
## DemosperGame         0.01802    0.01002   1.799  0.08048 .
## AssistsperGame       0.07308    0.05084   1.437  0.15925
## ShotsAgainstperGame  0.03027    0.02297   1.318  0.19581
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.0479 on 36 degrees of freedom
## Multiple R-squared:  0.9075, Adjusted R-squared:  0.8921
## F-statistic: 58.88 on 6 and 36 DF,  p-value: < 2.2e-16
```

```
print(anova(model.all))
```

```
## Analysis of Variance Table
##
## Response: WinPercent
##                     Df  Sum Sq Mean Sq  F value    Pr(>F)
## GoalsperGame         1 0.45461 0.45461 179.8814 1.107e-14 ***
## ShotsperGame         1 0.00072 0.00072   0.2845  0.597425
## ShootingPercent      1 0.01930 0.01930   7.6367  0.009404 **
## AssistsperGame       1 0.00208 0.00208   0.8227  0.371166
## SavesperGame         1 0.14731 0.14731  58.2870 1.063e-08 ***
## GoalsAgainstperGame  1 0.17249 0.17249  68.2519 1.943e-09 ***
## ShotsAgainstperGame  1 0.00829 0.00829   3.2808  0.079492 .
## SavePercent          1 0.00025 0.00025   0.1004  0.753351
## DemosperGame         1 0.00661 0.00661   2.6136  0.115770
## DemosAgainstperGame  1 0.00047 0.00047   0.1866  0.668649
## Residuals           32 0.08087 0.00253
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
print(model.all)
```

```
##
## Call:
## lm(formula = WinPercent ~ ., data = data)
##
## Coefficients:
##         (Intercept)         GoalsperGame         ShotsperGame
##            0.246584             0.035716             0.029149
##     ShootingPercent       AssistsperGame         SavesperGame
##            0.611907             0.057719            -0.084205
## GoalsAgainstperGame  ShotsAgainstperGame          SavePercent
##           -0.167712             0.017447             0.264025
##        DemosperGame  DemosAgainstperGame
##            0.017926             0.009456
```

```r
print(summary(model.all))
```

```
##
## Call:
## lm(formula = WinPercent ~ ., data = data)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.108457 -0.027271  0.001167  0.031683  0.100999
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)          0.246584   1.120915   0.220  0.82728
## GoalsperGame         0.035716   0.115552   0.309  0.75926
## ShotsperGame         0.029149   0.040242   0.724  0.47412
## ShootingPercent      0.611907   0.801953   0.763  0.45104
## AssistsperGame       0.057719   0.063873   0.904  0.37293
## SavesperGame        -0.084205   0.028625  -2.942  0.00603 **
## GoalsAgainstperGame -0.167712   0.201698  -0.832  0.41185
## ShotsAgainstperGame  0.017447   0.068164   0.256  0.79963
## SavePercent          0.264025   1.475516   0.179  0.85912
## DemosperGame         0.017926   0.010937   1.639  0.11100
## DemosAgainstperGame  0.009456   0.021890   0.432  0.66865
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.05027 on 32 degrees of freedom
## Multiple R-squared:  0.9094, Adjusted R-squared:  0.8811
## F-statistic: 32.13 on 10 and 32 DF,  p-value: 6.947e-14
```

```r
# Residuals vs Fitted Values Plot
ggplot(data = data.frame(
  Fitted = model.all$fitted.values,
  Residuals = model.all$residuals
), aes(x = Fitted, y = Residuals)) +
  geom_point(color = "black", size = 2) +
```

```
  geom_hline(yintercept = 0, color = "blue", linetype = "solid", linewidth = 1.2) +
  labs(
    title = "Residuals vs. Fitted Values",
    x = "Fitted Values",
    y = "Residuals"
  ) +
  theme_minimal()

# Histogram of Residuals with Mean Line
ggplot(data = data.frame(Residuals = model.all$residuals), aes(x = Residuals)) +
  geom_histogram(aes(y = ..density..), bins = 7, fill = "lightblue", color = "black") +
  geom_vline(aes(xintercept = mean(Residuals)), color = "red", linetype = "dashed", linewidth = 1.2) +
  geom_vline(aes(xintercept = median(Residuals)), color = "green", linetype = "dashed", linewidth = 1.2
  labs(
    title = paste(
      "Histogram of Residuals\nMean:", round(mean(model.all$residuals), 3),
      "Median:", round(median(model.all$residuals), 3)
    ),
    x = "Residuals",
    y = "Density"
  ) +
  theme_minimal()


# QQ Plot of Residuals
ggplot(data = data.frame(Residuals = model.all$residuals), aes(sample = Residuals)) +
  stat_qq(color = "black", size = 2) +
  stat_qq_line(color = "red", linetype = "solid", linewidth = 1.2) +
  labs(
    title = "QQ Plot of Residuals",
    x = "Theoretical Quantiles",
    y = "Sample Quantiles"
  ) +
  theme_minimal()


plot(density(model.all$residuals),
     main = "Residuals Density Plot",
     col = "blue", lwd = 2,
     xlab = "Fitted Values",
     ylab = "Residuals")
abline(v = mean(model.all$residuals), col = "red", lwd = 2, lty = 2)
```

```
# Create a model that can be used for inferences (Remove multicollinearity)
data.infer <- data %>% select(-AssistsperGame)

model.infer <- lm(WinPercent ~ GoalsAgainstperGame + GoalsperGame + SavesperGame + DemosperGame + Shots

print(anova(model.infer))
```

```
## Analysis of Variance Table
##
## Response: WinPercent
```

```
##                      Df  Sum Sq Mean Sq  F value    Pr(>F)
## GoalsAgainstperGame   1 0.64539 0.64539 273.4542 < 2.2e-16 ***
## GoalsperGame          1 0.11398 0.11398  48.2923 3.312e-08 ***
## SavesperGame          1 0.02049 0.02049   8.6835   0.00553 **
## DemosperGame          1 0.01953 0.01953   8.2730   0.00664 **
## ShotsAgainstperGame   1 0.00629 0.00629   2.6637   0.11114
## Residuals            37 0.08733 0.00236
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
print(model.infer)
```

```
##
## Call:
## lm(formula = WinPercent ~ GoalsAgainstperGame + GoalsperGame +
##     SavesperGame + DemosperGame + ShotsAgainstperGame, data = data.infer)
##
## Coefficients:
##         (Intercept)  GoalsAgainstperGame         GoalsperGame
##             0.57849             -0.20954              0.16144
##        SavesperGame         DemosperGame  ShotsAgainstperGame
##            -0.07918              0.02200              0.03718
```

```
print(summary(model.infer))
```

```
##
## Call:
## lm(formula = WinPercent ~ GoalsAgainstperGame + GoalsperGame +
##     SavesperGame + DemosperGame + ShotsAgainstperGame, data = data.infer)
##
## Residuals:
##       Min        1Q    Median        3Q       Max
## -0.101470 -0.029231 -0.002466  0.031901  0.109874
##
## Coefficients:
##                      Estimate Std. Error t value Pr(>|t|)
## (Intercept)           0.57849    0.07913   7.311 1.10e-08 ***
## GoalsAgainstperGame -0.20954    0.02906  -7.211 1.49e-08 ***
## GoalsperGame          0.16144    0.02191   7.370 9.17e-09 ***
## SavesperGame         -0.07918    0.02527  -3.134  0.00337 **
## DemosperGame          0.02200    0.00977   2.251  0.03040 *
## ShotsAgainstperGame   0.03718    0.02278   1.632  0.11114
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04858 on 37 degrees of freedom
## Multiple R-squared:  0.9022, Adjusted R-squared:  0.889
## F-statistic: 68.27 on 5 and 37 DF,  p-value: < 2.2e-16
```

```
options(width = 60)
cat("Partial R-Square Values:\n")
```

```
## Partial R-Square Values:
```

```r
library(sensemakr)
```

```
## Warning: package 'sensemakr' was built under R version
## 4.4.2
```

```
## See details in:
```

```
## Carlos Cinelli and Chad Hazlett (2020). Making Sense of Sensitivity: Extending Omitted Variable Bias
```

```r
partial_r2(model.infer)
```

```
##         (Intercept) GoalsAgainstperGame        GoalsperGame
##           0.59091709          0.58426656          0.59479624
##        SavesperGame        DemosperGame ShotsAgainstperGame
##           0.20974784          0.12047614          0.06715791
```