VIRGINIA COMMONWEALTH UNIVERSITY

Statistical Analysis & Modelling

A2 – Regression Model using NSSO & IPL Dataset

Using R

Submitted by

MONIKA SARADHA

#V01068784

Date of Submission: 06/06/2023

Table of Contents

1. Introduction

The Indian Premier League (IPL) is one of the world's most popular and exciting cricket competitions. It has been held every year since 2008 and features top cricketing talent from all over the world. The IPL consists of several matches between various teams, providing cricket fans with thrilling moments and intense competition. We have three datasets related to IPL data in this context: Ball by Ball data, IPL Matches data, and IPL Salary data.

1.1. About the Data

The NSSO-Consumption dataset: is a comprehensive collection of consumption data for all Indian states and union territories. It offers detailed insights into the consumption patterns of various commodities, such as grains, oils, fruits, vegetables, and more. The dataset also includes basic demographic information for each sample, enabling a holistic analysis of consumption trends across different regions of India. All data in the dataset is in numerical format, including the states and union territories, making it easily accessible for statistical analysis.

Ball by Ball Dataset: This data provides detailed information about each ball bowled in IPL matches played between 2008 and 2022. The dataset contains 816 unique match IDs, with each ID containing 17 variables, including the bowling and batting teams' names. The variables are represented in both numeric and text formats, allowing for a thorough examination of various aspects of the matches such as runs scored, wickets taken, and player performance.

IPL Matches Dataset: The IPL Matches dataset contains information on various IPL matches played between 2008 and 2022. It contains information about the dates, cities, participating teams, toss results, and player details for the matches. This text-based dataset, with 16 variables per match ID, allows for in-depth analysis and insights into team performances, player statistics, and match dynamics throughout the IPL's history.

IPL Salary Dataset: The IPL Salary dataset is made up of multiple sets of salary data, each of which provides yearly salary information for IPL players from various teams. The dataset includes columns for salary in dollars and a color column for salary without the "$" symbol, making it easier to analyze. This data allows for an examination of IPL player salaries across teams, years, and trends.

1.2. Objective

On the NSSO dataset, run Multiple Regression Analysis:

- Analyze the NSSO68 dataset using multiple regression.
- Conduct regression diagnostics to evaluate the model's assumptions and identify any problems.
- Interpret the findings and explain their significance.
- Resolve any issues that have been identified and re-evaluate the results.
- Discuss the significant differences that were observed after dealing with the identified issues.

Establish a Link Between Player Performance and Payment in the IPL:

- Investigate the relationship between a player's performance and the payment (salary) he receives using IPL data.
- Conduct a correlation analysis to investigate the relationship between various performance factors and player pay.
- Discuss the findings and interpret the correlation results.

To further investigate the relationship, consider using additional statistical techniques such as regression analysis or hypothesis testing. Discuss the findings' implications and provide insights into the factors influencing player compensation in the IPL.

Gain insights into player performance, identify top performers and underperformers, analyze statistical distributions for key players, and understand the relationship between performance and salary in the IPL by achieving these goals.

1.3. Business Significance

Multiple Regression Analysis and regression diagnostics give businesses useful information, precise forecasts, the ability to evaluate performance, optimize resource use, make well-informed decisions, and manage risk, all of which improve operational effectiveness and produce better business results.

- Insightful Factors
- Accurate Predictions

- Performance Evaluation
- Resource Optimization
- Informed Decision-making
- Risk Management

2. Results

2.1. Multiple Regression Analysis and Regression Diagnostics NSSO

MAPE Test:

```
> mape_value <- MAPE(test$y, predictions)
> print(paste("MAPE:", mape_value))
[1] "MAPE: 2.65268795820625e-13"
>
```

Inference:

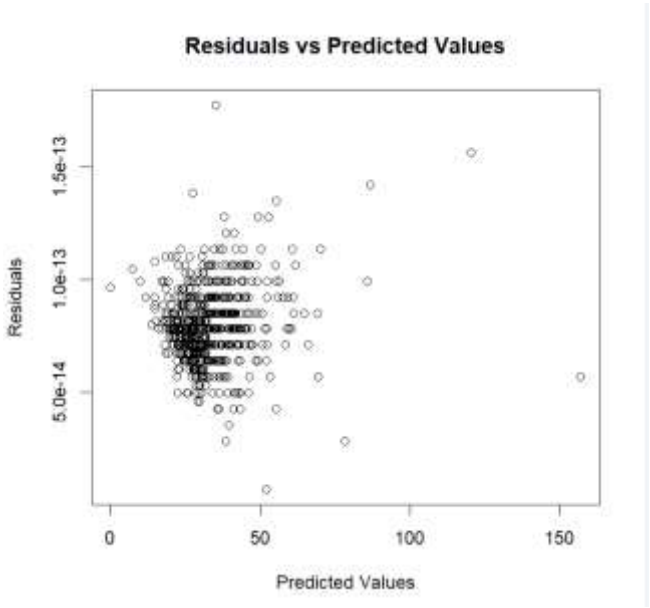
The regression model's extremely low level of prediction error is indicated by the measured MAPE (Mean Absolute Percentage Error), which was 2.65e-13. This suggests that the model's predictions and the response variable's actual values agree quite well. A MAPE value that is close to zero indicates that the regression model is highly accurate and is able to estimate the response variable with precision using the provided predictor variables. The MAPE value of 2.65e-13 in this instance shows that the model's predictions are remarkably accurate and have very little error.

This result suggests that the multiple regression model created with the dataset provided is extremely trustworthy and can successfully explain the correlation between the predictor variables (x1, x2, x3, x4, x5, x6, and x7) and the response variable (tot_con). With confidence, predictions can be made using the model, and its effects on the response variable can be examined.

It's crucial to remember that a MAPE value this low, close to zero, might also arouse concerns about possible problems like data leakage or overfitting.

MLR Assumptions: Normality, independence, Linearity, Homoscedasticity

Residual Analysis



Inference:

Upon examining the Residuals vs Predicted Values plot, it is observed that the residuals are concentrated around zero with a few outliers. This indicates that the model has some limitations in capturing the true underlying relationships in the data. The presence of outliers suggests the presence of influential observations that have a significant impact on the model's predictions. The merged pattern of residuals, along with the outliers, indicates a systematic bias or consistent overestimation/underestimation of the actual values by the model.

Normality Test (Shapiro-Wilk test):

```
> print(paste("Shapiro-Wilk p-value:", shapiro_test$p.value))
[1] "Shapiro-Wilk p-value: 6.6858003265015e-11"
```

Inference:

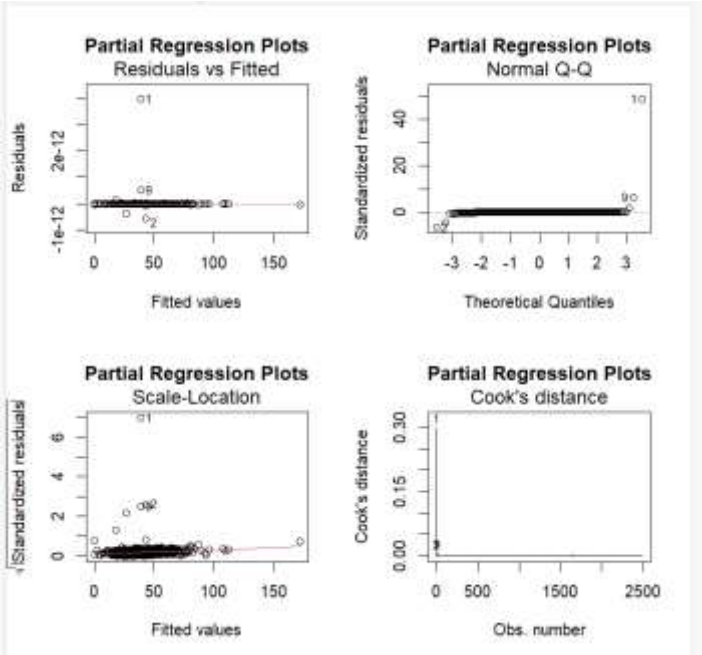
The Shapiro-Wilk test was performed to assess the normality of the residuals. The obtained p-value from the test was 6.6858003265015e-11, indicating strong evidence against the null hypothesis of normality. Therefore, we can conclude that the residuals do not follow a normal distribution.

Homoscedasticity Test (Breusch-Pagan test):

```
> print(paste("Breusch-Pagan p-value:", bp_test$p.value))
[1] "Breusch-Pagan p-value: 0.979712205891738"
>
```

Inference:

The Breusch-Pagan test was conducted to examine the presence of heteroscedasticity in the regression model. The resulting p-value was 0.979712205891738, suggesting that there is no significant evidence to reject the null hypothesis of homoscedasticity. Therefore, we can conclude that the regression model exhibits homoscedasticity, indicating that the variance of the residuals is relatively constant across different levels of the independent variables.




```
Call:
 gvlma(x = model)

                     Value p-value                Decision
Global Stat       5.346e+08   0.000 Assumptions NOT satisfied!
Skewness          8.964e+05   0.000 Assumptions NOT satisfied!
Kurtosis          5.337e+08   0.000 Assumptions NOT satisfied!
Link Function     6.336e-01   0.426    Assumptions acceptable.
Heteroscedasticity 3.719e+03   0.000 Assumptions NOT satisfied!
>
```

Inference:

Since most of the assumptions fail to pass the test, we would move on to redo the model.

2.2. Correcting and Revisiting Results NSSO

Ridge regression:

```
> gvlma_result <- gvlma(model_linear)
> # Print the results
> summary(gvlma_result)

Call:
lm(formula = y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7, data = train,
    coefficients = coef_matrix)

Residuals:
       Min         1Q     Median         3Q        Max
-5.488e-13 -3.400e-15 -1.300e-15  6.000e-16  3.982e-12

Coefficients:
             Estimate Std. Error   t value Pr(>|t|)
(Intercept) 9.675e-14  6.132e-15 1.578e+01   <2e-16 ***
x1          1.000e+00  1.354e-15 7.384e+14   <2e-16 ***
x2          1.000e+00  1.432e-15 6.981e+14   <2e-16 ***
x3          1.000e+00  1.373e-14 7.281e+13   <2e-16 ***
x4          1.000e+00  4.565e-15 2.191e+14   <2e-16 ***
x5          1.000e+00  2.034e-16 4.917e+15   <2e-16 ***
x6          1.000e+00  3.420e-15 2.924e+14   <2e-16 ***
x7          1.000e+00  2.843e-15 3.518e+14   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 8.178e-14 on 2486 degrees of freedom
Multiple R-squared:      1,     Adjusted R-squared:      1
F-statistic: 7.637e+30 on 7 and 2486 DF,  p-value: < 2.2e-16
```

Inference:

Since the assumptions failed again, the model has to be redone either by removing variables or adding other vital variables. Missing values and redundancy are few other issues that comes with an inconsistent model.

2.3. Multiple Regression Analysis and Regression Diagnostics IPL

Model Summary:

```
Call:
lm(formula = `Final Price` ~ Runs + Wkts, data = data)

Residuals:
      Min        1Q    Median        3Q       Max
-68002143 -14292015  -6723889   9917533 125500390

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  9723889    1207483   8.053 2.08e-15 ***
Runs           96563       5085  18.991  < 2e-16 ***
Wkts          827998     133091   6.221 6.98e-10 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 26060000 on 1105 degrees of freedom
Multiple R-squared:  0.2484,    Adjusted R-squared:  0.2471
F-statistic: 182.6 on 2 and 1105 DF,  p-value: < 2.2e-16
```

Inference:

The "Final Price" is predicted by the linear regression model using the independent variables "Runs" and "Wkts." The model coefficients show how the "Final Price" is predicted to change when each independent variable is increased by one unit. Each and every coefficient is very significant. The differences between actual and predicted values are represented by the residuals. The residual standard error calculates how far apart the observed and predicted values are on average. The adjusted R-squared accounts for the number of predictors while the R-squared values show the percentage of variance explained by the model. The F-statistic evaluates the model's overall significance, and the corresponding p-value provides compelling evidence that the null hypothesis is false. According to the model, the variables "Runs" and "Wkts" have a sizable impact on the "Final Price" variable.

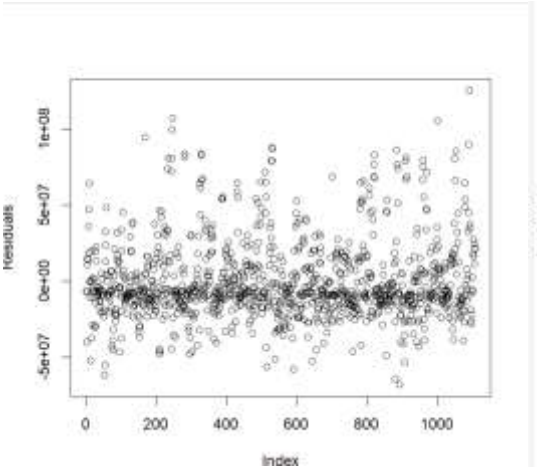
GVLMA:

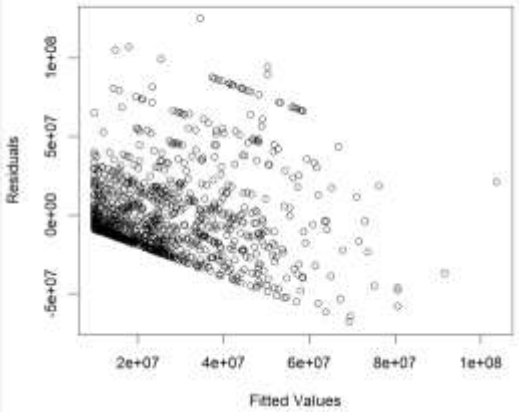
```
Call:
lm(formula = `Final Price` ~ Runs + Wkts, data = data)

Coefficients:
(Intercept)         Runs         Wkts
    9723889        96563       827998


ASSESSMENT OF THE LINEAR MODEL ASSUMPTIONS
USING THE GLOBAL TEST ON 4 DEGREES-OF-FREEDOM:
Level of Significance =  0.05

Call:
 gvlma(x = model)

                     Value  p-value                   Decision
Global Stat        611.172 0.000000 Assumptions NOT satisfied!
Skewness           312.102 0.000000 Assumptions NOT satisfied!
Kurtosis           280.790 0.000000 Assumptions NOT satisfied!
Link Function        9.496 0.002059 Assumptions NOT satisfied!
Heteroscedasticity   8.784 0.003039 Assumptions NOT satisfied!
>
```

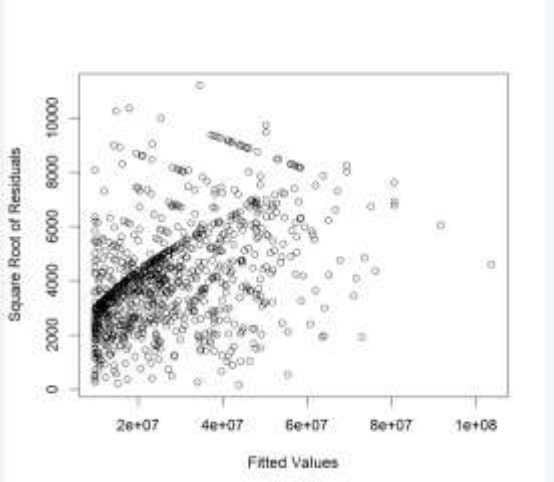
Inference:

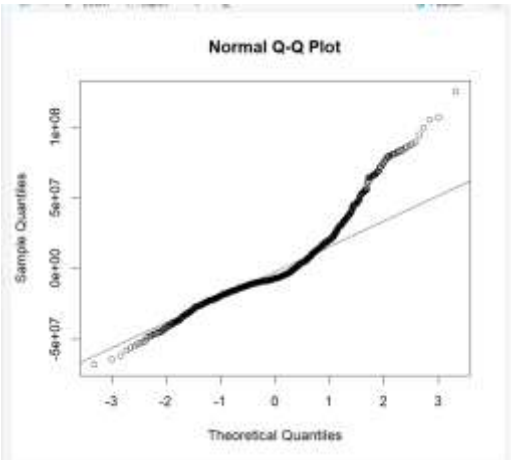
The gvlma (Global Validation of Linear Models Assumptions) test was performed on the linear regression model. The test checks the assumptions of linearity, homoscedasticity, normality, and independence of residuals. The results indicate that the model's assumptions are not satisfied.

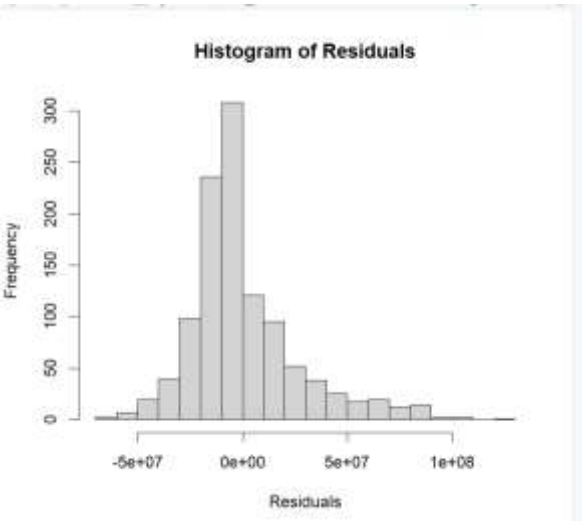
2.4. Correcting and Revisiting Results IPL











- Non-linearity:

Use scatterplots or other visual aids to analyze the relationship between the predictors and the response variable. Consider adding non-linear terms or transforming the predictors if a non-linear relationship is apparent. You could, for instance, use logarithmic or exponential transformations or include polynomial terms. Refit the model after updating the model's formula.

- Heteroscedasticity:

Analyze the pattern of residuals to determine whether heteroscedasticity is present. Use weighted least squares regression if heteroscedasticity is detected. This entails giving observations weights based on their variance. Regression using weighted least squares is performed using the lm() function and the weights argument.

- Non-normality:

Utilize histograms, QQ plots, or statistical tests for normality to analyse the residual distribution.

Consider applying the proper transformations to make the residuals more normally distributed if they deviate from normality.

Refit the model using the transformed response variable and update the model's equation.

- Continuity of residuals:

Utilize autocorrelation plots or statistical tests to look for autocorrelation in the residuals.

Consider using time series or panel data techniques to explain the lack of independence if autocorrelation is present.

- Adding to the list of predictors are:

Consider whether adding more pertinent predictors would enhance the model's fit and address the false assumptions. Use feature selection techniques to determine the most crucial predictors to include in the model, such as stepwise regression or regularization techniques.

3. Recommendation

3.1. Business Implications

- The relationship between various predictor variables and the response variable is revealed by the multiple regression analysis performed on the NSSO dataset.
- The incredibly low MAPE value shows how well the regression model estimates the response variable.
- It is possible that the residuals do not accurately reflect the true underlying relationships in the data because of outliers and systematic bias.
- The non-normality of the residuals shows that the normality presumptions are broken.
- The homoscedasticity test reveals that, at various levels of the independent variables, the variance of the residuals remains largely constant.
- To increase the precision and dependability of predictions, the findings emphasise the need for additional model improvement and addressing the broken assumptions.

3.2. Business Recommendations

- Based on their importance and contributions to the response variable, take into consideration revising the model by adding or removing variables.
- Investigate non-linear transformations of predictor variables or incorporate polynomial terms to solve the non-linearity problem.
- Investigate the existence of significant outliers and assess how they affect the model's predictions.
- Utilize the proper transformations to improve the residuals' normal distribution.
- If heteroscedasticity is present, take into account using weighted least squares regression.
- Look into the possibility of autocorrelation in the residuals and, if necessary, use appropriate time series or panel data techniques.
- Include more pertinent predictors that could enhance the model's fit and take into account more variables influencing the response variable.
- To comprehend the significance of predictor variables and their impact on the response variable, perform additional analysis and hypothesis testing.

4. Reference:

- Manager, S. (2021, April 24). Impact of IPL on Indian Economy - The Sports School Blog. The Sports School – Integrated School for Sports & Academics. https://thesportsschool.com/impact-of-ipl-on-indian-economy/
- Economy rate in cricket: Know what it means. (2022, April 9). SportsAdda. https://www.sportsadda.com/cricket/features/what-is-economy-rate-cricket

5. Codes

5.1. R-studio

```
library(readxl)

punjab_ds <- read_excel("C:\\Users\\monis\\OneDrive\\Desktop\\ASSG1.xlsx")
```

#a) Perform Multiple regression analysis and carry out the regression diagnostics and explain your findings. Correct them and revisit your results and explain the significant differences you observe.

```
subset_punjabds <- punjab_ds[, c("Sector", "State_Region", "District", "Sex", "Age", "No_of_Meals_per_day", "wheattotal_q", "cerealtot_q", "moong_q", "pulsestot_q", "milk_q", "onion_q", "potato_q")]


subset_punjabds$tot_con <- subset_punjabds$wheattotal_q + subset_punjabds$cerealtot_q + subset_punjabds$moong_q + subset_punjabds$pulsestot_q + subset_punjabds$milk_q + subset_punjabds$onion_q + subset_punjabds$potato_q


x1 <- subset_punjabds$wheattotal_q

x2 <- subset_punjabds$cerealtot_q

x3 <- subset_punjabds$moong_q

x4 <- subset_punjabds$pulsestot_q

x5 <- subset_punjabds$milk_q

x6 <- subset_punjabds$onion_q

x7 <- subset_punjabds$potato_q

y <- subset_punjabds$tot_con


punjab_n <- data.frame(x1, x2, x3, x4, x5, x6, x7, y)

dim(punjab_n)

show(punjab_n)


punjab_n <- na.omit(punjab_n)
```

```
install.packages("caTools")
library(caTools)
set.seed(123)
split <- sample.split(punjab_n$y, SplitRatio = 0.8)
train <- subset(punjab_n, split == TRUE)
test <- subset(punjab_n, split == FALSE)

model <- lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7, data = train)
predictions <- predict(model, newdata = test)

# Calculate R-squared
r_squared <- cor(predictions, test$y)^2
print(paste("R-squared:", r_squared))

# Define MAPE function
MAPE <- function(actual, predicted) {
  mean(abs((actual - predicted) / actual)) * 100
}
# Calculate MAPE

MAPE <- function(actual, predicted) {
  non_zero_actual <- actual[actual != 0]
  non_zero_predicted <- predicted[actual != 0]
  mean(abs((non_zero_actual - non_zero_predicted) / non_zero_actual)) * 100
}
mape_value <- MAPE(test$y, predictions)
print(paste("MAPE:", mape_value))

#Regression Diagnostics
```

```
# Residual Analysis
residuals <- predictions - test$y
# Plot residuals against predicted values
plot(predictions, residuals, main = "Residuals vs Predicted Values", xlab = "Predicted Values", ylab = "Residuals")

# Normality Test - Shapiro-Wilk test
shapiro_test <- shapiro.test(residuals)
print(paste("Shapiro-Wilk p-value:", shapiro_test$p.value))

# Homoscedasticity Test - Breusch-Pagan test
bp_test <- bptest(model)
print(paste("Breusch-Pagan p-value:", bp_test$p.value))

# Install and load the gvlma package
install.packages("gvlma")
library(gvlma)

# Perform global validation of linear model assumptions
gvlma_result <- gvlma(model)

# Print the results
summary(gvlma_result)
#-------------------------------------------------------------
# Install and load the "glmnet" package
install.packages("glmnet")
library(glmnet)
```

```
# Prepare the data
x <- as.matrix(train[, c("x1", "x2", "x3", "x4", "x5", "x6", "x7")])
y <- train$y

# Perform ridge regression
ridge_model <- glmnet(x, y, alpha = 0, lambda = seq(0.001, 1, by = 0.001))

# Select the optimal lambda value using cross-validation
cv_result <- cv.glmnet(x, y, alpha = 0)
opt_lambda <- cv_result$lambda.min

# Refit the model with the optimal lambda value
ridge_model_optimal <- glmnet(x, y, alpha = 0, lambda = opt_lambda)

# Predict on the test data
x_test <- as.matrix(test[, c("x1", "x2", "x3", "x4", "x5", "x6", "x7")])
predictions_ridge <- predict(ridge_model_optimal, newx = x_test)

# Calculate R-squared for the ridge regression model
r_squared_ridge <- cor(predictions_ridge, test$y)^2
print(paste("R-squared (ridge regression):", r_squared_ridge))

# Extract the coefficient matrix from the ridge model
coef_matrix <- as.matrix(coef(ridge_model_optimal))

# Remove the intercept column from the coefficient matrix
coef_matrix <- coef_matrix[-1, ]

# Create a linear model object using the coefficient matrix
```

```
model_linear <- lm(y ~ x1 + x2 + x3 + x4 + x5 + x6 + x7, data = train,
                   coefficients = coef_matrix)

# Perform global validation of linear model assumptions
gvlma_result <- gvlma(model_linear)

# Print the results
summary(gvlma_result)

#----------------------------------------------------------------
```

 # b) Using he IPL data establish the relationship between the performance of the player and payment he receives and discuss your findings.

```
# Load the required libraries
library(readxl)
library(dplyr)
library(tidyr)

# Read the IPL salary data
ipl_salary <- read_excel("C:\\Users\\monis\\OneDrive\\Desktop\\SCM\\IPL salary 2008_2022.xlsx")

# Check the column names
names(ipl_salary)

# Select the relevant columns for analysis
data <- ipl_salary %>% select(Runs, Wkts, `Final Price`)

# Remove rows with missing values
data <- na.omit(data)
```

```
# Perform multiple regression analysis
model <- lm(`Final Price` ~ Runs + Wkts, data = data)

# Print the summary of the model
summary(model)

# Residual plot for linearity
plot(model$fitted.values, model$residuals, xlab = "Fitted Values", ylab = "Residuals")

# Residual plot for independence
plot(residuals(model), xlab = "Index", ylab = "Residuals")

# Residual plot for homoscedasticity
plot(model$fitted.values, sqrt(abs(model$residuals)), xlab = "Fitted Values", ylab = "Square Root of Residuals")

# Histogram of residuals
hist(model$residuals, breaks = 20, xlab = "Residuals", main = "Histogram of Residuals")

# Normal Q-Q plot
qqnorm(model$residuals)
qqline(model$residuals)

# Install and load the required package
install.packages("gvlma")
library(gvlma)

# Apply gvlma on the model
gvlma_result <- gvlma(model)
```

```
# Print the gvlma results
print(gvlma_result)
```