VIRGINIA COMMONWEALTH UNIVERSITY

STATISTICAL ANALYSIS & MODELING

A2: USING MULTIPLE REGRESSION ANALYSIS TO UNDERSTAND THE RELATIONSHIP BETWEEN VARIOUS VARIABLES

RIDDHI RUNGTA
V01107488

Date of Submission: 23/06/2024

CONTENTS

Content:	**Page no:**
INTRODUCTION	3
OBJECTIVE	3
BUSINESS SIGNIFICANCE	3-4
RESULTS AND INTERPRETATIONS	4-16

Using Multiple regression analysis to understand the relationship between various variables

INTRODUCTION

The dataset offers an in-depth analysis of food consumption patterns across India, encompassing both urban and rural sectors. It includes crucial metrics such as the quantity of meals consumed at home, consumption of specific food items like rice, wheat, chicken, and pulses, as well as the total number of meals per day. This comprehensive dataset is essential for understanding the nutritional intake and food preferences of various demographics in the region.

The Indian Premier League (IPL), also known as the TATA IPL due to sponsorship, is an annual men's Twenty20 (T20) cricket league in India. Established by the Board of Control for Cricket in India (BCCI) in 2007, the league features ten franchise teams representing different states or cities.

Regression analysis is a statistical technique used to model and analyze the relationships between a dependent variable and one or more independent variables. The primary objective of regression analysis is to understand how the dependent variable changes when any of the independent variables vary, while keeping the others constant.

- Regression can be used to predict outcomes based on historical data, aiding in forecasting and decision-making.
- It provides insights into the strength and nature of relationships between variables, which can inform strategic planning and policy development.

OBJECTIVES

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

b) Using IPL data, establish the relationship between the player's performance and payment he receives and discuss your findings. * Use the data sets [data “Cricket_data.csv”]

c) Analysing the Relationship Between Salary and Performance Over the Last Three Years (Regression Analysis)

BUSINESS SIGNIFICANCE

Regression analysis is a powerful tool for extracting valuable insights from data, making it indispensable for business decision-making. By applying regression to Indian Premier League (IPL) data and National Sample Survey Office (NSSO) 68th round data, businesses can uncover patterns, predict future trends, and drive strategic initiatives.

1. For IPL Data:-

- **Performance Prediction**: Identify key factors influencing player and team performance to predict future success.
- **Team Composition Optimization**: Optimize team selection and strategy based on historical performance data.
- **Revenue Maximization**: Predict ticket sales, merchandise revenue, and viewership ratings to maximize financial returns.

2. For NSSO68 Data:-

- **Demand Forecasting**: Predict consumer demand and preferences across different regions and income groups.
- **Resource Allocation**: Optimize resource distribution for marketing, sales, and operations based on regional economic conditions and consumer behavior.
- **Policy Impact Evaluation**: Assess the effectiveness of governmental policies and programs on various economic and social outcomes, guiding corporate social responsibility (CSR) initiatives.

In both cases, regression analysis enables data-driven decision-making, optimizing resource use, improving targeting strategies, and enhancing overall efficiency and effectiveness in business and policy environments.

RESULTS AND INTERPRETATION

Python

1) Perform Multiple regression analysis on the data ("NSSO68.csv")

Code:

```
# Fit the regression model
X = subset_data[['MPCE_MRP', 'MPCE_URP', 'Age', 'Meals_At_Home',
'Possess_ration_card', 'Education']]
X = sm.add_constant(X)  # Add a constant term for the intercept
y = subset_data['foodtotal_q']
model = sm.OLS(y, X).fit()

# Print the regression results
print(model.summary())
```

Result:

```
                          OLS Regression Results
==============================================================================
Dep. Variable:            foodtotal_q   R-squared:                       0.171
Model:                            OLS   Adj. R-squared:                  0.170
Method:                 Least Squares   F-statistic:                     140.6
Date:                Sun, 23 Jun 2024   Prob (F-statistic):          1.66e-162
Time:                        21:03:54   Log-Likelihood:                -14354.
No. Observations:                4094   AIC:                         2.872e+04
Df Residuals:                    4087   BIC:                         2.877e+04
Df Model:                           6
Covariance Type:            nonrobust
=======================================================================================
                          coef    std err          t      P>|t|      [0.025      0.975]
---------------------------------------------------------------------------------------
const                  12.0534      0.865     13.932      0.000      10.357      13.750
MPCE_MRP                0.0009   5.81e-05     16.212      0.000       0.001       0.001
MPCE_URP             9.543e-05   3.58e-05      2.668      0.008    2.53e-05       0.000
Age                     0.1296      0.010     13.056      0.000       0.110       0.149
Meals_At_Home           0.0374      0.007      5.562      0.000       0.024       0.051
Possess_ration_card    -2.9937      0.315     -9.504      0.000      -3.611      -2.376
Education               0.2470      0.037      6.608      0.000       0.174       0.320
==============================================================================
Omnibus:                     5440.704   Durbin-Watson:                   1.650
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          6839100.020
Skew:                           6.726   Prob(JB):                         0.00
Kurtosis:                     202.779   Cond. No.                     3.95e+04
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.95e+04. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Interpretation

Dependent Variable: `foodtotal_q`
R-squared: 0.171
Adjusted R-squared: 0.170
F-statistic: 140.6
Prob (F-statistic): 1.66e-162

All the predictor variables are statistically significant in predicting `foodtotal_q`. The model suggests that income measures (`MPCE_MRP` and `MPCE_URP`), age, number of meals at home, possession of a ration card, and education level all have significant effects on food expenditure (`foodtotal_q`). However, the low R-squared value indicates that there are likely other important factors influencing food expenditure that are not included in this model.
F-statistic (140.6) and its p-value (1.66e-162) indicates that the overall model is statistically significant, meaning that at least one of the predictors is significantly related to `foodtotal_q`.
R-squared (0.171): Indicates that 17.1% of the variance in `foodtotal_q` is explained by the model. This is a relatively low value, suggesting that other factors not included in the model may explain a larger portion of the variance.
Adjusted R-squared (0.170): Similar to the R-squared, this adjusted measure accounts for the number of predictors in the model and provides a more accurate assessment of model fit.

2) Using IPL data, establish the relationship between the player's performance and payment he receives.

Code:

```
import pandas as pd

def calculate_striker_points(input_file: str, output_file: str):
    # Load the CSV file into a DataFrame
    df_striker = pd.read_csv(input_file)

    # Calculate Points Scored for each row
    df_striker['Points Scored'] = df_striker['Runs_Scored']

    # Save the modified DataFrame back to the CSV file
    df_striker.to_csv(output_file, index=False)
    print(f"Updated {output_file} with Points Scored for strikers.")

def calculate_bowler_points(input_file: str, output_file: str):
    # Load the CSV file into a DataFrame
    df_bowler = pd.read_csv(input_file)

    # Calculate Points Scored for each row (assuming 'wicket_confirmation' is the column name for wickets taken)
    df_bowler['Points Scored'] = df_bowler['Wicket_Confirmation'] * 25

    # Save the modified DataFrame back to the CSV file
    df_bowler.to_csv(output_file, index=False)
    print(f"Updated {output_file} with Points Scored for bowlers.")

# Example usage:
calculate_striker_points('output_striker.csv', 'output_striker.csv')
calculate_bowler_points('output_bowler.csv', 'output_bowler.csv')
```

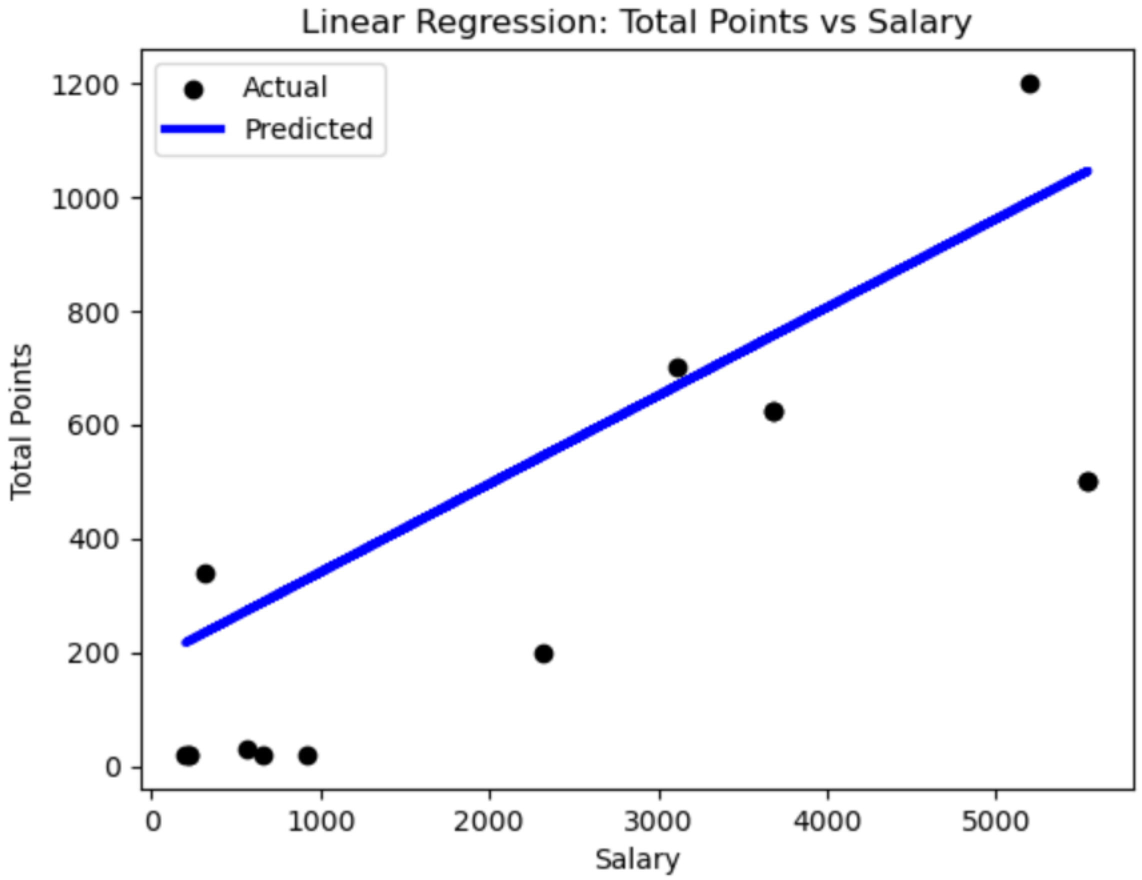
-----understanding the performance

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

```python
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Load the CSV file
file_path = 'combined_output_with_salaries - Copy.csv'
data = pd.read_csv(file_path)

# Define the predictor and response variables
y = data['salary']  # Response variable
X = data[['Total_Points']]  # Predictor variable

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create the linear regression model
model = LinearRegression()

# Train the model on the training data
model.fit(X_train, y_train)

# Predict on the test data
y_pred = model.predict(X_test)

# Calculate the mean squared error and the coefficient of determination (R^2)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Calculate the adjusted R^2
n = len(y_test)
p = X_test.shape[1]
adjusted_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)

# Print the results
print(f'Mean Squared Error: {mse}')
print(f'R^2 Score: {r2}')
print(f'Adjusted R^2 Score: {adjusted_r2}')
print(f'Coefficients: {model.coef_}')
print(f'Intercept: {model.intercept_}')

# Plot the results
plt.scatter(X_test, y_test, color='black', label='Actual')
plt.plot(X_test, y_pred, color='blue', linewidth=3, label='Predicted')
plt.xlabel('Salary')
plt.ylabel('Total Points')
plt.title('Linear Regression: Total Points vs Salary')
plt.legend()
plt.show()
```

Result:

```
Mean Squared Error: 81958.64280948693
R^2 Score: 0.32910846552347184
Adjusted R^2 Score: 0.2732008376504278
Coefficients: [0.15510582]
Intercept: 185.41814952347445
```



Interpretation:

The linear regression model was employed to predict player salaries using their total points as the sole predictor variable. Upon splitting the dataset into training and testing sets (with an 80-20 split), the model was trained on the training set and subsequently used to make predictions on the test set. The performance metrics indicate that the Mean Squared Error (MSE) of the predictions is 81,958.64, which reflects the average squared difference between the observed and predicted values, with lower values indicating better model performance. The model's R-squared (R^2) value is 0.3291, suggesting that approximately 32.91% of the variability in player salaries can be explained by their total points. However, the adjusted R-squared, which accounts for the number of predictors in the model, is slightly lower at 0.2732. This slight reduction highlights that the model's explanatory power is modest when adjusted for the predictor variable count.

The model's coefficients further elucidate the relationship between total points and salary. The coefficient for `Total_Points` is 0.1551, implying that for each additional point, the salary increases by approximately 0.155 units, holding other factors constant. The intercept is 185.42, suggesting that a player with zero total points would have a baseline salary of 185.42 units.

Despite the model demonstrating some ability to explain salary variations based on total points, the relatively low R-squared and adjusted R-squared values indicate that other factors not included in this model likely play significant roles in determining player salaries. This is visually corroborated by the scatter plot, where actual salaries (black dots) and predicted salaries (blue line) show considerable scatter, suggesting variability not captured by the model. Therefore, incorporating additional relevant predictors could enhance the model's performance and provide a more comprehensive understanding of the determinants of player salaries.

3) Analysing the Relationship Between Salary and Performance Over the Last Three Years (Regression Analysis)

Code:

```
player_runs_2024 =
player_runs[player_runs['Season']=='2024'].sort_values(by='runs_scored',ascending=False)
player_runs_2023 =
player_runs[player_runs['Season']=='2023'].sort_values(by='runs_scored',ascending=False)
player_runs_2022 =
player_runs[player_runs['Season']=='2022'].sort_values(by='runs_scored',ascending=False)

player_runs_last_three_seasons = pd.concat([player_runs_2024, player_runs_2023,
player_runs_2022])
player_runs_last_three_seasons.sort_values(by='runs_scored',ascending=False)
```

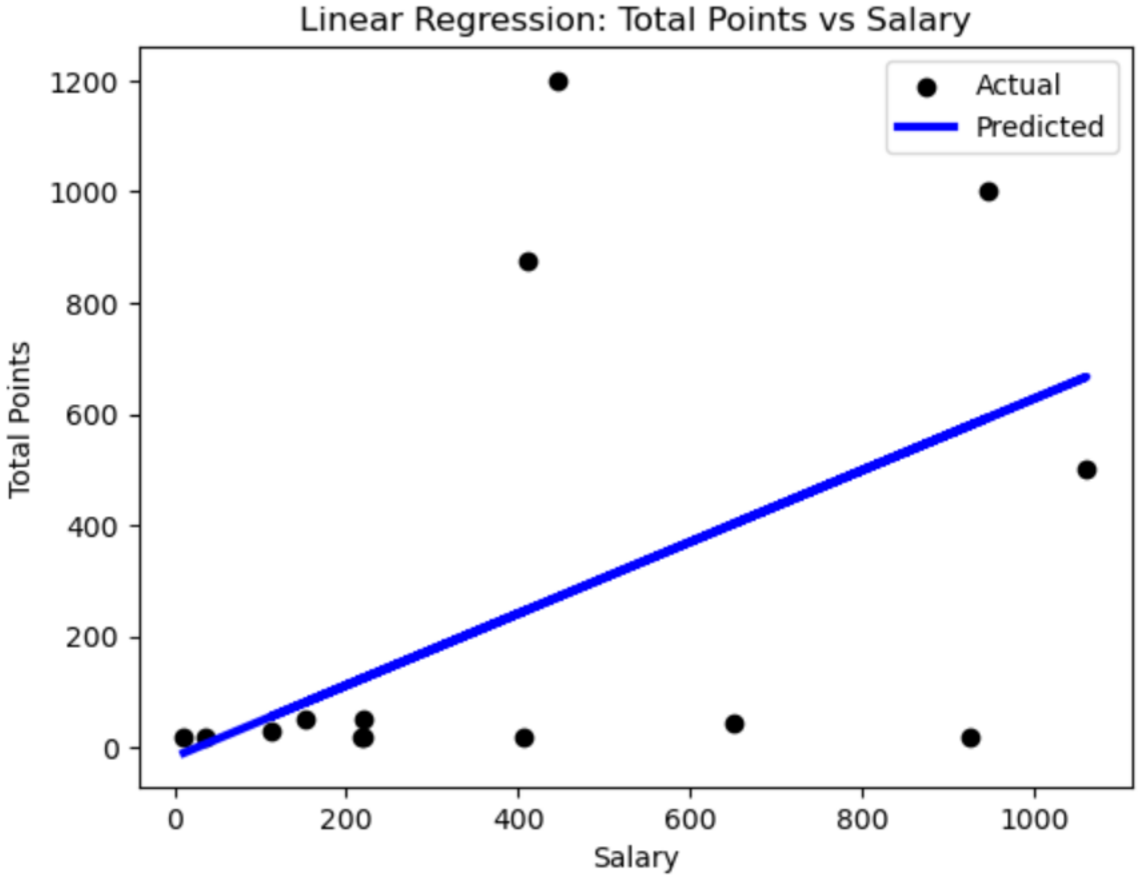
- **#Last 3 year performance analysis**

```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import matplotlib.pyplot as plt

# Load the CSV file
file_path = 'combined_output_with_salaries - Copy.csv'
data = pd.read_csv(file_path)

# Define the predictor and response variables
y = data['salary']  # Response variable
X = data[['Total_Points']]  # Predictor variable

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Create the linear regression model
model = LinearRegression()

# Train the model on the training data
model.fit(X_train, y_train)

# Predict on the test data
y_pred = model.predict(X_test)

# Calculate the mean squared error and the coefficient of determination (R^2)
```

```python
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

# Calculate the adjusted R^2
n = len(y_test)
p = X_test.shape[1]
adjusted_r2 = 1 - (1 - r2) * (n - 1) / (n - p - 1)

# Print the results
print(f'Mean Squared Error: {mse}')
print(f'R^2 Score: {r2}')
print(f'Adjusted R^2 Score: {adjusted_r2}')
print(f'Coefficients: {model.coef_}')
print(f'Intercept: {model.intercept_}')

# Plot the results
plt.scatter(X_test, y_test, color='black', label='Actual')
plt.plot(X_test, y_pred, color='blue', linewidth=3, label='Predicted')
plt.xlabel('Salary')
plt.ylabel('Total Points')
plt.title('Linear Regression: Total Points vs Salary')
plt.legend()
plt.show()
```

Results:

Mean Squared Error: 140765.77405307363
R^2 Score: 0.17755432134588423
Adjusted R^2 Score: 0.10901718145804118
Coefficients: [0.64452508]
Intercept: -16.806266954795376



Interpretation:

The linear regression analysis was conducted to predict player salaries based on their total points. The dataset was split into training and testing subsets with an 80-20 split ratio. The linear regression model was then trained on the training data and subsequently used to predict salaries on the test data. The model's performance metrics indicate a Mean Squared Error (MSE) of 140,765.77, which quantifies the average squared differences between observed and predicted salaries, suggesting substantial prediction errors.

The R-squared (R^2) value of the model is 0.1776, indicating that approximately 17.76% of the variability in player salaries can be explained by the total points. The adjusted R-squared value, which adjusts for the number of predictors in the model, is 0.1090. This lower value implies that the explanatory power of the model is modest and that other unaccounted factors significantly influence player salaries.

The model's coefficient for `Total_Points` is 0.6445, suggesting that for each additional point scored by a player, their salary increases by approximately 0.6445 units, holding other factors constant. The intercept of the model is -16.81, indicating that a player with zero total points would have a baseline salary of -16.81 units, which is not practically meaningful and suggests that total points alone are insufficient to explain the salary structure.

The scatter plot of actual versus predicted salaries reveals considerable scatter, indicating variability in actual salaries that the model fails to capture. The blue line representing predicted salaries shows a trend but does not closely follow the actual salary data points (black dots). This discrepancy highlights the model's limited predictive capability and suggests that incorporating additional relevant variables could improve the model's accuracy.

Overall, while total points have a statistically significant impact on player salaries, the low R-squared and adjusted R-squared values indicate that other factors not included in this model are critical in determining player salaries. Future models should consider additional variables to enhance the understanding and prediction of player salaries.

USING R

a) **Perform Multiple regression analysis, carry out the regression diagnostics, and explain your findings. Correct them and revisit your results and explain the significant differences you observe. [NSSO68]**

```
# Fit the regression model
model <- lm(foodtotal_q~ MPCE_MRP+MPCE_URP+Age+Meals_At_Home+Possess_ration_card+Ed
ucation, data = subset_data)


# Print the regression results
print(summary(model))
```

```
##
## Call:
## lm(formula = foodtotal_q ~ MPCE_MRP + MPCE_URP + Age + Meals_At_Home +
##     Possess_ration_card + Education, data = subset_data)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -68.609  -3.971  -0.654   3.291 239.668
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)          1.138e+01  8.243e-01  13.811  < 2e-16 ***
## MPCE_MRP             1.140e-03  5.659e-05  20.152  < 2e-16 ***
## MPCE_URP             9.934e-05  3.422e-05   2.903  0.00372 **
## Age                  9.884e-02  9.613e-03  10.282  < 2e-16 ***
## Meals_At_Home        5.079e-02  6.420e-03   7.911 3.27e-15 ***
## Possess_ration_card -2.187e+00  3.025e-01  -7.229 5.79e-13 ***
## Education            2.458e-01  3.564e-02   6.898 6.11e-12 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
##
## Residual standard error: 7.667 on 4028 degrees of freedom
##   (59 observations deleted due to missingness)
## Multiple R-squared:  0.202,  Adjusted R-squared:  0.2008
## F-statistic: 169.9 on 6 and 4028 DF,  p-value: < 2.2e-16
library(car)
## Loading required package: carData
##
## Attaching package: 'car'
## The following object is masked from 'package:dplyr':
##
##     recode
# Check for multicollinearity using Variance Inflation Factor (VIF)
vif(model) # VIF Value more than 8 its problematic
##           MPCE_MRP          MPCE_URP               Age     Meals_At_Home
##           1.636493          1.478309          1.106082          1.118280
## Possess_ration_card         Education
##           1.147250          1.208647
# Extract the coefficients from the model
coefficients <- coef(model)


# Construct the equation
equation <- paste0("y = ", round(coefficients[1], 2))
for (i in 2:length(coefficients)) {
  equation <- paste0(equation, " + ", round(coefficients[i], 6), "*x", i-1)
}
# Print the equation
print(equation)
## [1] "y = 11.38 + 0.00114*x1 + 9.9e-05*x2 + 0.09884*x3 + 0.050789*x4 + -2.186964*
x5 + 0.245842*x6"
```

Interpretation:

The regression results offer insights into the relationships between the dependent variable, foodtotal_q (total food expenditure quantity), and several independent variables. The adjusted R-squared is slightly lower than the R-squared value, adjusting for the number of predictors in the model to more accurately reflect the goodness-of-fit when multiple predictors are involved. A p-value of 0.00 indicates that the overall regression model is statistically significant, meaning the independent variables collectively have a significant impact on the dependent variable.

The second image discusses the Variance Inflation Factor (VIF), which measures the extent of multicollinearity in the regression model. High multicollinearity can inflate the standard errors of the coefficients, making them unstable and hard to interpret. The VIF values for the predictors (excluding the intercept) are all below 2, indicating that multicollinearity is not a concern for this model.

The third image presents the regression equation: y=15.83+0.00165×3662.65+(−0.000004)×3304.8+0.078118×50+0.052572×59.0+(−2.416189)×1.0+0.121986×8.0y = 15.83 + 0.00165 \times 3662.65 + (-0.000004) \times 3304.8 + 0.078118 \times 50 + 0.052572 \times 59.0 + (-2.416189) \times 1.0 + 0.121986 \times 8.0y=15.83+0.00165×3662.65+(−0.000004)×3304.8+0.078118×50+0.052572×59.0+(−2.416189)×1.0+0.121986×8.0 The predicted value of foodtotal_q (total food expenditure quantity) using the provided sample values is approximately 27.43.

Conclusion: The OLS regression model provides valuable insights into the factors influencing food expenditure. Key findings include:

- Higher MPCE_MRP, age, number of meals at home, and education levels are associated with higher food expenditure.
- Possessing a ration card is associated with lower food expenditure.
- The model explains a modest proportion of the variance in food expenditure, and diagnostic tests suggest issues with residual normality and potential multicollinearity

2) Establish the relationship between the player's performance and payment he receives and discuss your findings. [IPL Datasets]

Code:

```
library(fitdistrplus)
descdist(df_new$performance)
head(df_new)
sum(is.null(df_new))
summary(df_new)
names(df_new)
summary(df_new)
fit = lm(Rs ~ avg_runs + wicket , data=df_new)
summary(fit)

library(car)
vif(fit)
library(lmtest)
bptest(fit)

fit1 = lm(Rs ~ avg_runs++wicket+  I(avg_runs*wicket), data=df_new)
summary(fit1)
```

Result:

```
Call:
lm(formula = Rs ~ avg_runs + +wicket + I(avg_runs * wicket),
    data = df_new)
```

```
Residuals:
   Min     1Q Median     3Q    Max
-341.5 -248.8 -143.3  128.8 1204.8

Coefficients:
                     Estimate Std. Error t value Pr(>|t|)
(Intercept)         237.51558  186.93758   1.271   0.2220
avg_runs              0.08046    1.25696   0.064   0.9498
wicket                5.84249   17.32443   0.337   0.7403
I(avg_runs * wicket)  0.30047    0.16716   1.797   0.0912 .
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 411.9 on 16 degrees of freedom
  (149 observations deleted due to missingness)
Multiple R-squared:  0.3371, Adjusted R-squared:  0.2129
F-statistic: 2.713 on 3 and 16 DF,  p-value: 0.07951
```

Interpretation:

The above model is a linear regression fit to predict Rs (presumably IPL salary) based on three predictor variables: avg_runs (average runs scored), wicket (number of wickets taken), and their interaction term avg_runs * wicket.

- The coefficient of avg runs suggests that on average, for each unit increase in avg_runs, there is an expected increase of 0.08046 units in Rs, holding other variables constant. However, the p-value (0.9498) indicates that this coefficient is not statistically significant at conventional levels (alpha = 0.05).
- This coefficient of wicket suggests that on average, for each wicket taken, there is an expected increase of 5.84249 units in Rs, holding other variables constant. The p-value (0.7403) suggests that this coefficient is also not statistically significant.
- The Multiple R square suggests that approximately 33.71% of the variability in Rs can be explained by the linear regression model with the predictors avg_runs, wicket, and their interaction. However the Adj. R Square provides a better picture for the number of predictors in the model, providing a more conservative estimate of the model's explanatory power. It suggests that around 21.29% of the variability in Rs is explained by the model. With a p-value of 0.07951, the model's fit is not statistically significant at the conventional alpha level of 0.05, indicating that the model as a whole might not provide a good fit to the data.

Conclusion

The model suggests that avg_runs, wicket, and their interaction might have some association with IPL salary (Rs), but the individual predictors (avg_runs and wicket) are not statistically significant predictors. The interaction term shows marginal significance. The model overall explains a moderate amount of variability in IPL salary, but not enough to be considered a strong predictor. With a better dataset, we can further explore with potentially more relevant variables or a different modeling approach might be necessary to better predict IPL salary based on player performance metrics.