Predictive modelling with linear regression

Lego Set Dataset

Student Name: - Niravkumar Purohit

Student ID: - (8923245)

Subject: - Multivariate Statistics

Subject Code:- Stat8030

Professor: Bip Thapa

Index

INTRODUCTION

This project aims to analyze a dataset containing information about Lego sets. The dataset includes various predictors such as set names, reviews, product IDs, countries, and more. The main objective is to build a regression model to predict the list price of lego set based on these variables. The dataset is obtained from Kaggle and by examining the data and applying regression analysis techniques, we can predict the list price of lego sets.

1. Gathering Data

To do the analysis the data set is gathered from Kaggle website and the name of the dataset is Lego sets. List prices (the dependent variable) and different predictors, including set names, customer reviews, product IDs, and nations, are all included in the data regarding Lego sets. The dataset is appropriate for regression analysis because it includes a quantitative response (list price) and a number of predictors.

Dependent Variable: list price,

Independent Variables: Set Names, Review, Product id, country, etc.

Variable	Type
prod_id	Categorical
piece_count	Numeric
play_star_rating	Numeric
num_reviews	Numeric

review_difficulty	Categorical
star_rating	Numeric
country	Categorical
Age	Categorical
List_price	Numerical
Prod_desc	Categorical
Val_star_ratings	Numerical
Theme_name	Categorical
Set_names	Categorical

Descriptive analytics

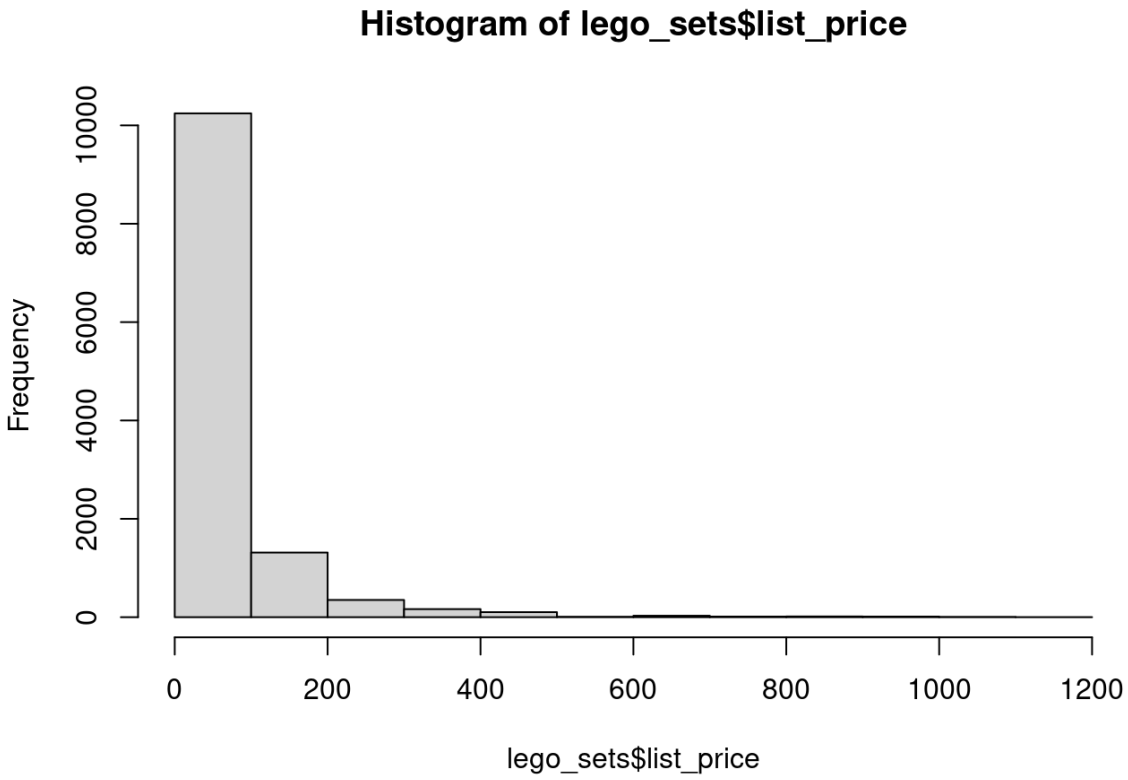
The minimum, Q1, median, Q3 and maximum values of the variables are given below.

```
    ages             list_price        num_reviews      piece_count
Length:12261       Min.   :   2.272   Min.   :  1.00   Min.   :   1.0
Class :character   1st Qu.:  19.990   1st Qu.:  2.00   1st Qu.:  97.0
Mode  :character   Median :  36.588   Median :  6.00   Median : 216.0
                   Mean   :  65.142   Mean   : 16.83   Mean   : 493.4
                   3rd Qu.:  70.192   3rd Qu.: 13.00   3rd Qu.: 544.0
                   Max.   :1104.870   Max.   :367.00   Max.   :7541.0
                                      NA's   :1620
play_star_rating  prod_desc          prod_id         prod_long_desc
Min.   :1.000    Length:12261       Min.   :    630   Length:12261
1st Qu.:4.000    Class :character   1st Qu.:  21034   Class :character
Median :4.500    Mode  :character   Median :  42069   Mode  :character
Mean   :4.338                       Mean   :  59837
3rd Qu.:4.800                       3rd Qu.:  70922
Max.   :5.000                       Max.   :2000431
NA's   :1775
review_difficulty    set_name          star_rating     theme_name
Length:12261       Length:12261       Min.   :1.800   Length:12261
Class :character   Class :character   1st Qu.:4.300   Class :character
Mode  :character   Mode  :character   Median :4.700   Mode  :character
                                      Mean   :4.514
                                      3rd Qu.:5.000
                                      Max.   :5.000
                                      NA's   :1620
val_star_rating   country
Min.   :1.000   Length:12261
1st Qu.:4.000   Class :character
Median :4.300   Mode  :character
Mean   :4.229
3rd Qu.:4.700
Max.   :5.000
```

```
NA's   :1795
```

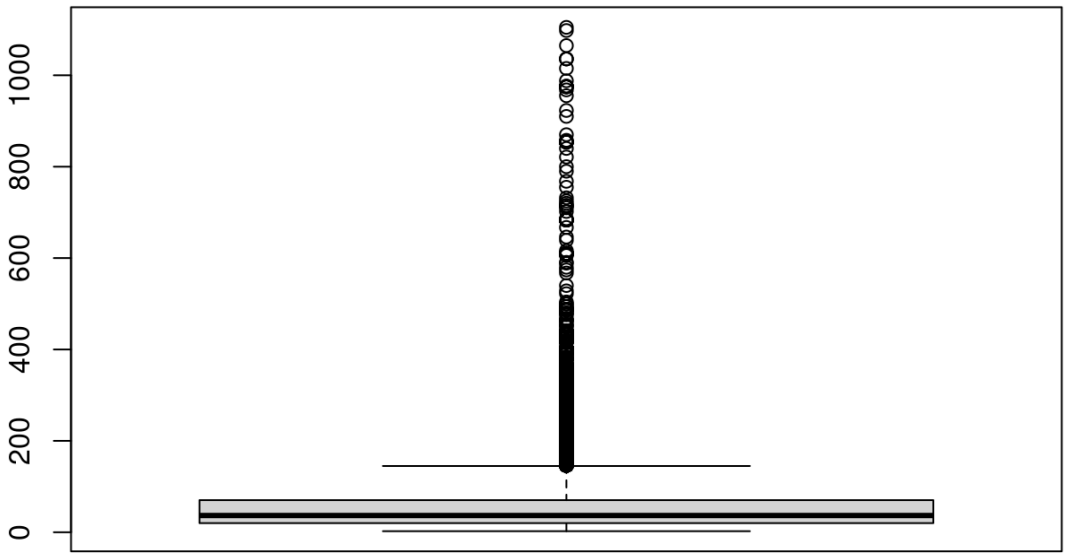
Histogram

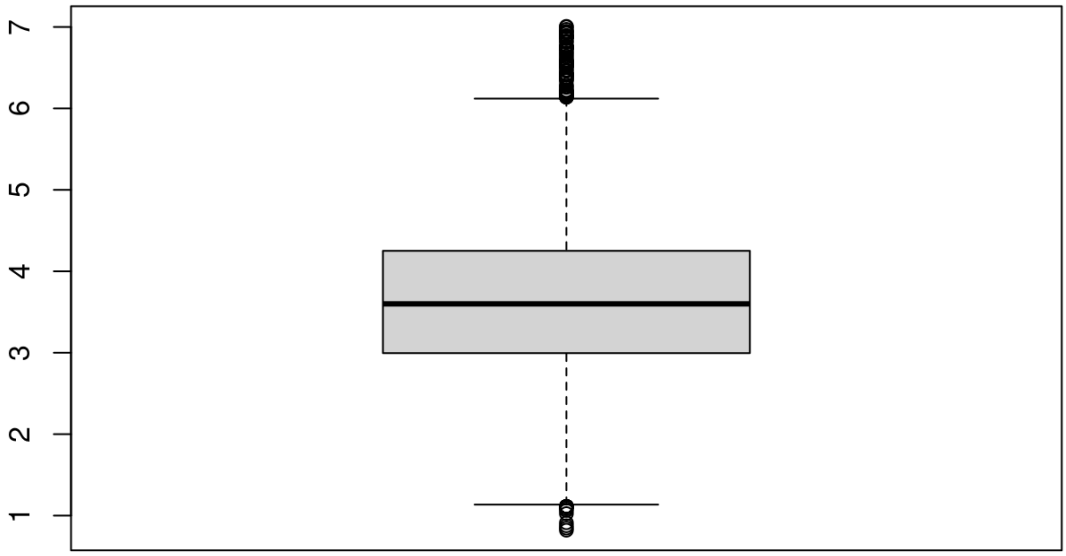
This is the graphical representation of the list price having x as list price and y as frequency.



Boxplot

This is the boxplot representation of list price. It helps to detect the outliers. We have applied the log functions to make this boxplot.





Initializing Modelling

In our first modelling, we took the following variables to create a model as they have high impact on the list price of Lego sets. The following predictors, in our opinion, are important:

- Set Names: The popularity or uniqueness of a Lego set might affect its price.
- Review: Higher-rated sets may command higher prices due to increased customer satisfaction.
- Product ID: Different product lines or categories could influence pricing.
- Country: Lego prices may vary across regions due to factors like local demand and distribution costs.

On the other hand, we do not choose factors like the Lego age, Prod description, theme name etc. These factors could add more noise to the model and are unlikely to have a direct effect on List price

The dataset was cleaned and prepared before we ran a linear regression model using the correct predictors. The regression's coefficients are shown in the table below:

```
                  list_price num_reviews piece_count play_star_rating   pro
d_id
list_price         1.0000000          NA   0.8696299                NA 0.388
6331
num_reviews               NA           1          NA                NA
NA
piece_count        0.8696299          NA   1.0000000                NA 0.217
7165
play_star_rating          NA          NA          NA                 1
NA
prod_id            0.3886331          NA   0.2177165                NA 1.000
0000
star_rating               NA          NA          NA                NA
NA
val_star_rating           NA          NA          NA                NA
NA
                  star_rating val_star_rating
list_price                 NA              NA
num_reviews                NA              NA
piece_count                NA              NA
play_star_rating           NA              NA
prod_id                    NA              NA
star_rating                 1              NA
val_star_rating            NA               1
```

We have created the first model between list price and peace count. The adjusted R square value for this model is 0.7562 and RSE value is 45.41. On further we created one mode model with independent variables prod_ id, peace count, play star ratings, num reviews. On solving this we get adjusted Rsqure 0.8 which is better than model 1.

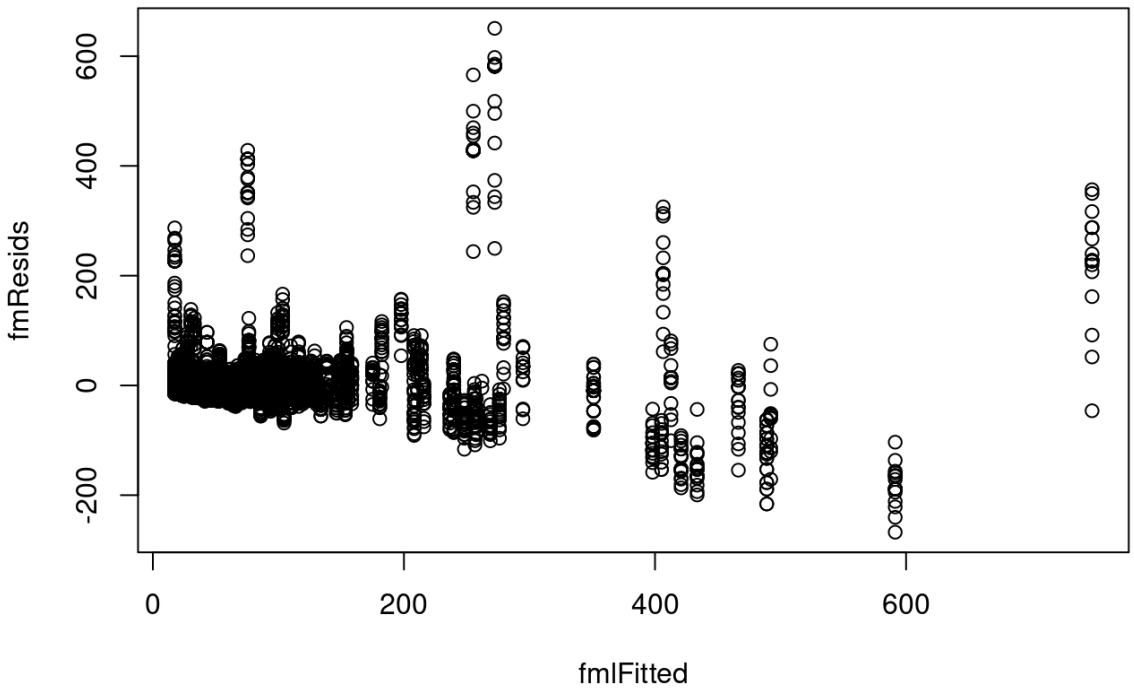
```
Residuals:
    Min      1Q  Median      3Q     Max
-267.46  -14.38   -6.45    6.97  650.69

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) 1.732e+01  4.778e-01   36.26   <2e-16 ***
piece_count 9.691e-02  4.969e-04  195.03   <2e-16
```

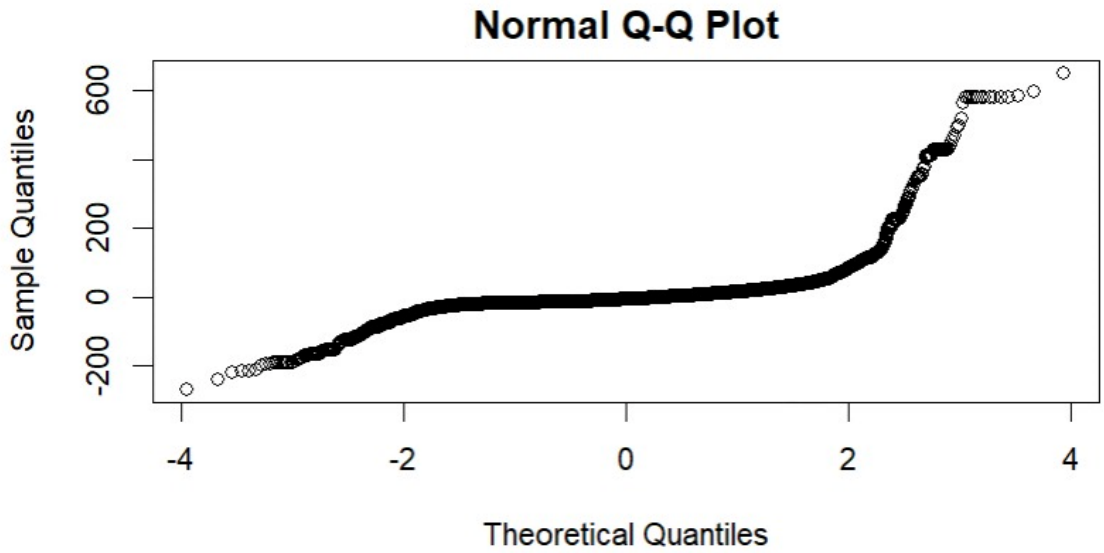
Second Model: - In second model we have created a linear model between list price and other variables such as peace count, product id and play star ratings etc. There is an increase in adjusted R-squared value, which is a good sign for a model. The values are given below.

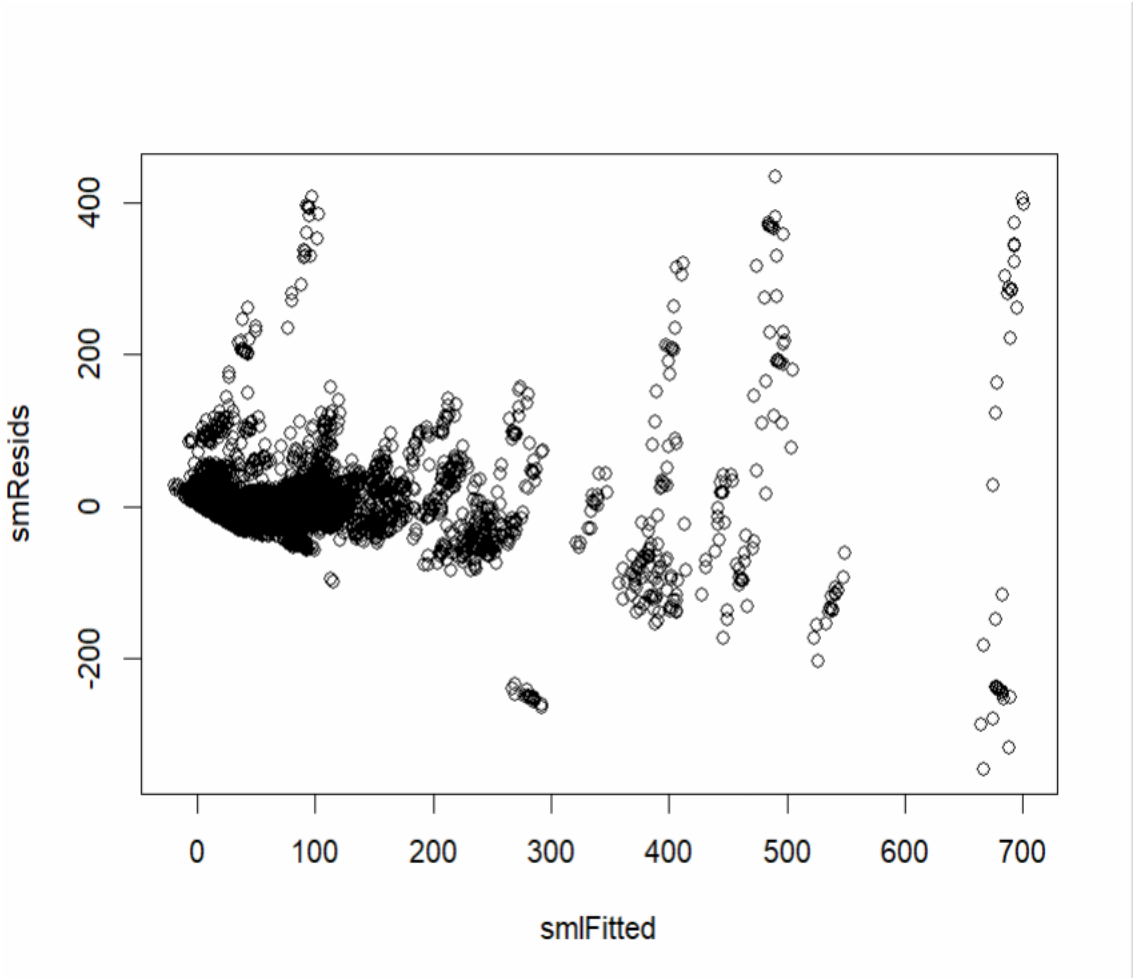
```
Residual standard error: 43.13 on 10155 degrees of freedom
  (2076 observations deleted due to missingness)
Multiple R-squared:  0.8106,   Adjusted R-squared:  0.8101
F-statistic:  1499 on 29 and 10155 DF,  p-value: < 2.2e-16
```

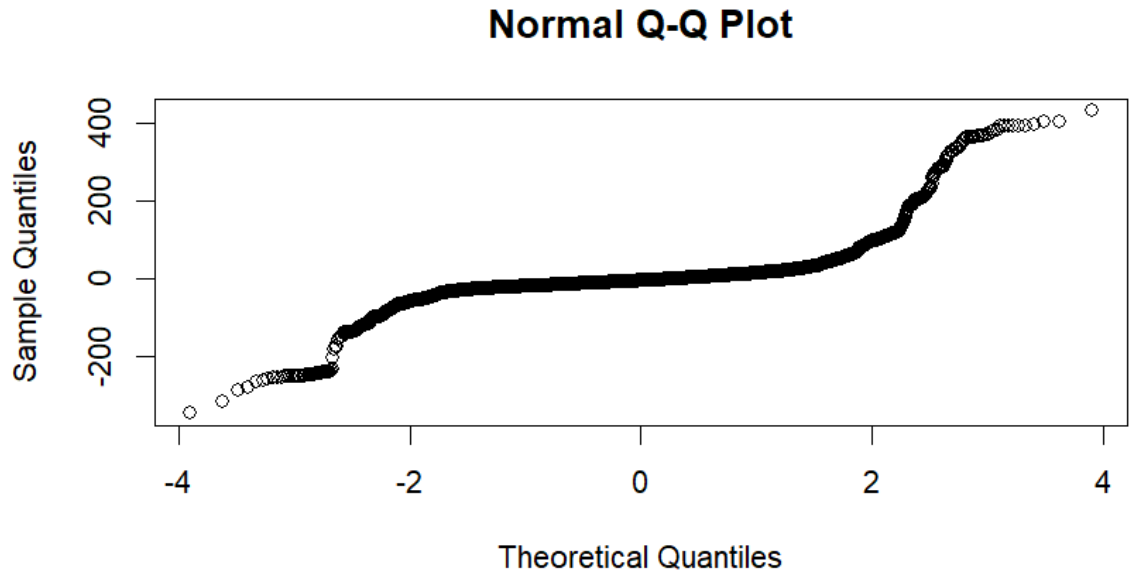
Diagnostics

Now we will the diagnose both the linear model with the help of residual vs fitted graphs and QQplot









From the above diagram it is evident that the residuals are not normally distributed.

Model Selection

In this step, we have added polynomial terms such as squared terms, interaction terms to the model. However, none of the any additional terms improved the model. Therefore, we chose to stick with the initial linear regression model as it provided a good balance between simplicity and performance.

We have selected smp model as the best model with the adjusted R-squre values 0.81, which is the highest value than other and predict the accurate value of list price.

Prediction and Summary

Now we have predicted list price value by assuming peace count 10, 20 and 30 within the confidence level 0.95 using first model on lego sets.

```
       1        2        3
18.29346 19.26259 20.23173
```

```
   fit      lwr      upr
1 18.29346 17.36180 19.22512
2 19.26259 18.33582 20.18937
3 20.23173 19.30977 21.15369
```

In summary, our regression analysis on the Lego Sets dataset revealed that set names, reviews, product IDs, and country are significant predictors of the list price of Lego sets. The initial linear regression model satisfied the assumptions of linearity, homoscedasticity, and normality. Non-linear terms were considered but did not contribute substantially to the model's performance. The selected model provides a reliable framework for predicting the list prices of Lego sets based on the given predictors.

Reference

Source: https://www.kaggle.com/datasets/mterzolo/lego-sets

Appendix

```
library(tidyverse)
library(MASS)
library(dplyr)
library(stargazer)
library(caret)
library(leaps)
library(ggplot2)
library(readr)

lego_sets <- read_csv("C:/Users/Hp/Downloads/archive (2)/lego_sets.csv")
View(lego_sets)

#descriptive analytics
str(lego_sets)
summary(lego_sets)

#histogram
hist(lego_sets$list_price)
boxplot(lego_sets$list_price,width = 0.7)
boxplot(log(lego_sets$list_price))


#correlations
numeric_data <- lego_sets[, sapply(lego_sets, is.numeric)]
cor_matrix <- cor(numeric_data)
print(cor_matrix)
cor(lego_sets$list_price,lego_sets$piece_count)
```

```
#first_model

attach(lego_sets)

fm<-lm(list_price~piece_count)

summary(fm)


sm<-
lm(list_price~prod_id+piece_count+play_star_rating+num_reviews+review_difficulty+star_rating+co
untry)

#sm <- lm(list_price ~ ., data = lego_sets)

summary(sm)


#diagnostic


fmResids <- fm$residuals

fmlFitted <- fm$fitted.value


plot(fmlFitted,fmResids)

dev.new(width = 10, height = 8)

par(mar = c(5, 5, 2, 2))



smResids <- sm$residuals

smlFitted <- sm$fitted.value

plot(smlFitted,smResids)

dev.new(width = 10, height = 8)

par(mar = c(5, 5, 2, 2))

qqnorm(fmResids)

qqnorm(smResids)


#extension

summary(sm)
```

```
smp<-
lm(list_price~prod_id+I(prod_id^2)+piece_count+prod_id:play_star_rating+play_star_rating+num_reviews+review_difficulty+star_rating+country)


summary(smp)


#Feature_selection

step<-stepAIC(smp,direction= "forward",trace=FALSE)

step$anova


step1<-stepAIC(smp,direction= "backward",trace=FALSE)

step1$anova


smnp<-
lm(list_price~prod_id+piece_count+prod_id:play_star_rating+play_star_rating+review_difficulty+star_rating+country)

summary(smnp)



summary(fm)



#prediction_model


piece_count_predictions <- data.frame(piece_count = c(10, 20, 30))

predict(fm,piece_count_predictions)


fm<-lm(list_price~piece_count)

piece_count_predictions <- data.frame(piece_count = c(10, 20, 30))


prediction_interval <- predict(fm, newdata = piece_count_predictions, interval = "confidence", level = 0.95)

prediction_interval
```