Video Game Global Sales

MIS 649 : Business Analytics

Aishwariya Chunduru
Anusha Kirigere
Himali Shewale
Sonam Desai
Vaishnavi Inamdar

San Diego State University, Fowler College of Business
Dr. Xialu Liu
May 08, 2022

Table of Contents:

1) Introduction:

The video game industry is a rapidly growing and constantly evolving sector of the entertainment industry. With the rise of advanced technology and the increasing popularity of gaming as a hobby, the video game industry has become a multi-billion dollar global market, with millions of gamers worldwide engaging with a diverse range of gaming experiences.Video games have come a long way since the early days of arcade machines and simple console games. Today's video games feature complex storylines, immersive gameplay, and stunning graphics that rival those of Hollywood blockbusters. The industry encompasses a wide variety of game genres, from action and adventure to role-playing, sports, strategy, and more.

The video game industry is not only a source of entertainment but also a significant contributor to the economy. It provides jobs to thousands of people in various fields, such as game design, programming, marketing, and publishing. The industry also has a ripple effect on related industries, such as hardware manufacturers, media outlets, and retail stores.As with any industry, the video game industry faces its own set of challenges and opportunities. It is highly competitive, with numerous companies vying for gamers' attention and dollars. Game developers must continually innovate and create compelling content to stay relevant and succeed in the marketplace. Moreover, the industry faces issues such as diversity and inclusion, cybersecurity, and intellectual property protection.

Despite these challenges, the video game industry continues to grow and evolve, with new technologies, platforms, and business models emerging regularly. As the industry continues to expand, it offers a wealth of opportunities for research, innovation, and creativity, making it an exciting field for anyone interested in gaming and entertainment.With the industry continuing to grow rapidly, there is a need for comprehensive data analysis to understand the trends and patterns that shape the video game landscape. The video game sales with ratings dataset, available on Kaggle, provides a rich collection of information on the sales, ratings, and other relevant details for thousands of video games released over the years.

2) Dataset

This dataset presents a valuable opportunity for researchers, gamers, and industry professionals to explore and analyze the characteristics of successful games, the impact of different genres, the role of ratings and reviews, and much more. By delving into this dataset, we can gain deeper insights into the complex and dynamic world of video games, informing our understanding of this important and influential aspect of modern culture.This dataset includes information on over 16,500 video games released from 1980 to 2016 across a wide variety of gaming platforms, such as PC, PlayStation, Xbox, Nintendo, and more. The dataset contains various features, such as the game title, platform, year of release, genre, publisher, sales in different regions (North America, Europe, Japan, and other regions), critic and user ratings, and rating counts. The ratings are based on a scale of 0-10, and the dataset also includes information on the number of ratings used to calculate the average.

The video game sales with ratings dataset provides a wealth of information that can be used to explore different aspects of the video game industry. For example, researchers can analyze trends in video game sales over time, examine the popularity of different genres, investigate the relationship between critic and user ratings, and much more. Game developers and publishers can also use this dataset to inform their business decisions, such as identifying which platforms and genres are most popular among gamers, and what factors contribute to successful game releases. Overall, the video game sales with ratings dataset is a valuable resource for anyone interested in the video game industry, providing a wealth of data that can be used to gain deeper insights into this dynamic and ever-evolving field.

3) Preparation and Interpretation of Data:

Data Cleaning: The process of data cleaning entails the identification and rectification or elimination of errors, inconsistencies, or inaccuracies in datasets to ensure their accuracy, completeness, and suitability for analysis. This involves various steps, such as detecting missing values, outliers, and errors, and then either removing or replacing the missing values. Correcting or discarding the errors and outliers, standardizing the format, eliminating duplicate records, reforming variable names and labels, and verifying the data units are also important steps in data cleaning. Data cleaning is essential for minimizing the possibility of making incorrect conclusions or decisions based on inaccurate or incomplete data. Moreover, it can result in more precise and dependable statistical analyses and predictive models.

In order to finalize a cleaned dataset for our analysis, we removed five variables that we deemed unnecessary for our purposes and eliminated any missing values. Our cleaned dataset now consists of 6947 observations and 13 variables. Specifically, we removed the 'NA_Sales', 'EU_Sales', 'JP_Sales', and 'Other_Sales' columns from the dataset.

Quantitative Variables	Qualitative Variables
Global Sales (Response Variable)	Name
Critic_Score*	Platform
Critic_Count*	Year_of_Release
User_Score*	Genre
User_Count	Publisher
	Developer
	Rating

Table 1: Quantitative and Qualitative Variables

Alongside the fields: Name, Platform, Year_of_Release, Genre, Publisher, NA_Sales, EU_Sales, JP_Sales, Other_Sales, Global_Sales, we have:-

- Critic_score - Aggregate score compiled by Metacritic staff
- Critic_count - The number of critics used in coming up with the Critic_score
- User_score - Score by Metacritic's subscribers
- User_count - Number of users who gave the user_score
- Developer - Party responsible for creating the game
- Rating - The ESRB ratings

Data size: 11563

4) Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a method of data analysis that concentrates on summarizing and depicting a dataset's key features. EDA is used to better understand the data and spot any patterns, trends, or connections between variables that might be present.

Plotted histograms for quantitative variables like Global sales, Critic Score, Critic Count, User Score and User count. Histograms depict the distribution of a continuous variable. They provide information about the distribution's shape, center, and spread. We may identify the following by looking at the histogram:

1. Shape: The distribution's symmetry, skewness to the left or right, bimodality, or uniformity.
2. Center: The peak or mode of the distribution determines where the distribution is focused or concentrated.

In the above histograms, Global sales, Critics Count, and User Count are all positively skewed. The ratings of Critics Score and Users Score are skewed negatively.

Figure 1.

Histograms

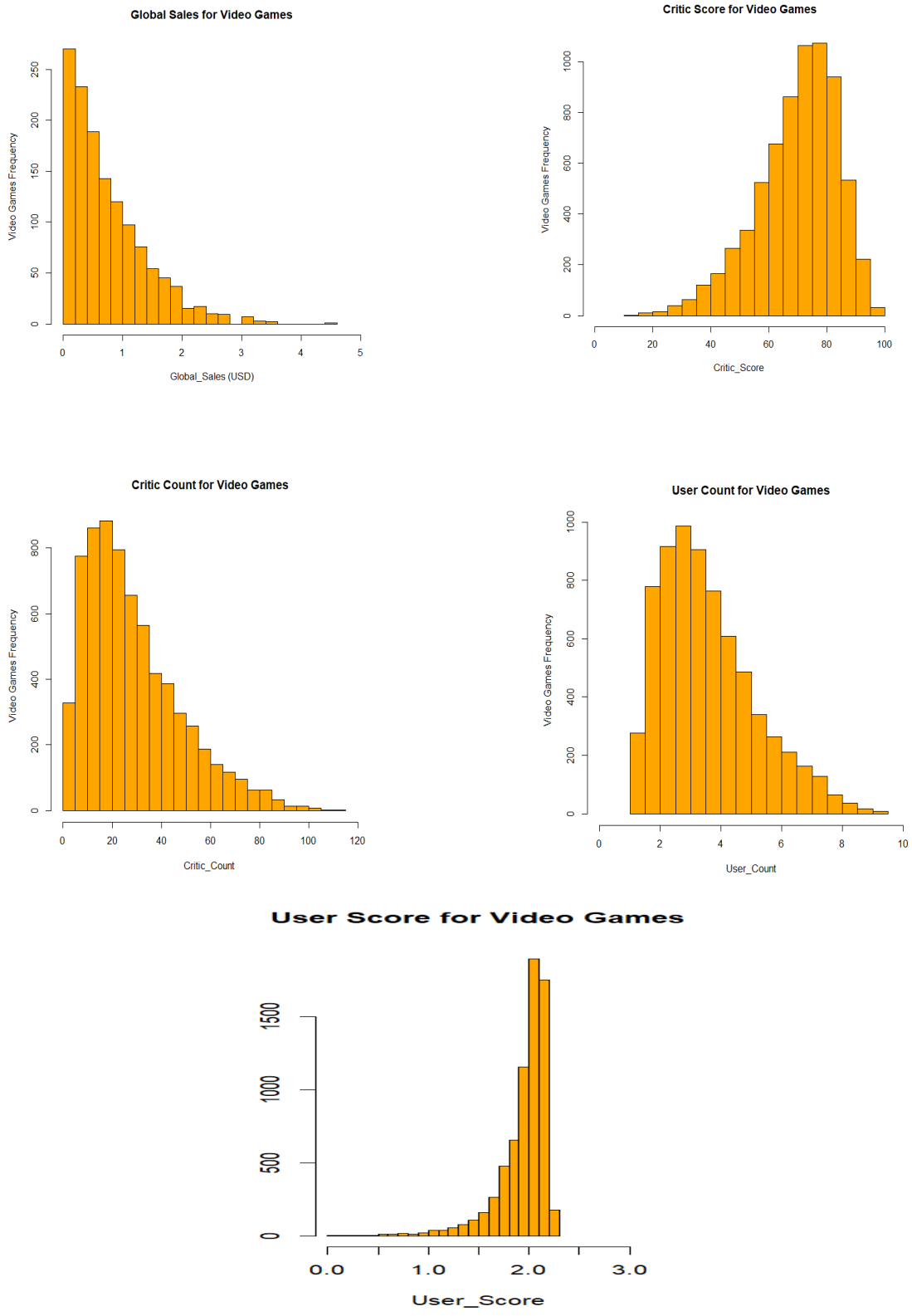
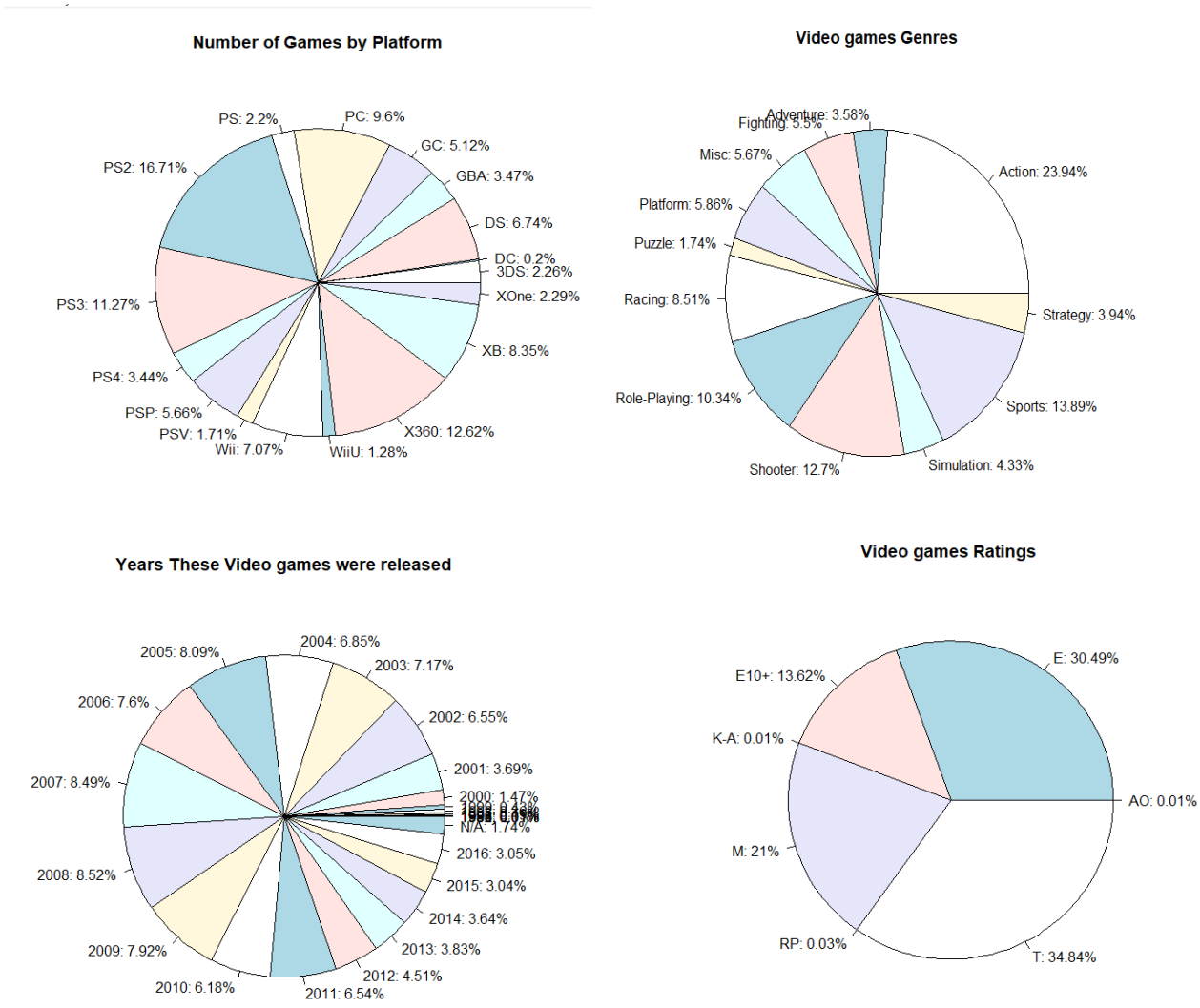


Figure 2.

Pie Charts



A pie chart is a representation of values as colored slices of a circle. The slices are labeled, and the numbers associated with each slice are also shown on the chart.

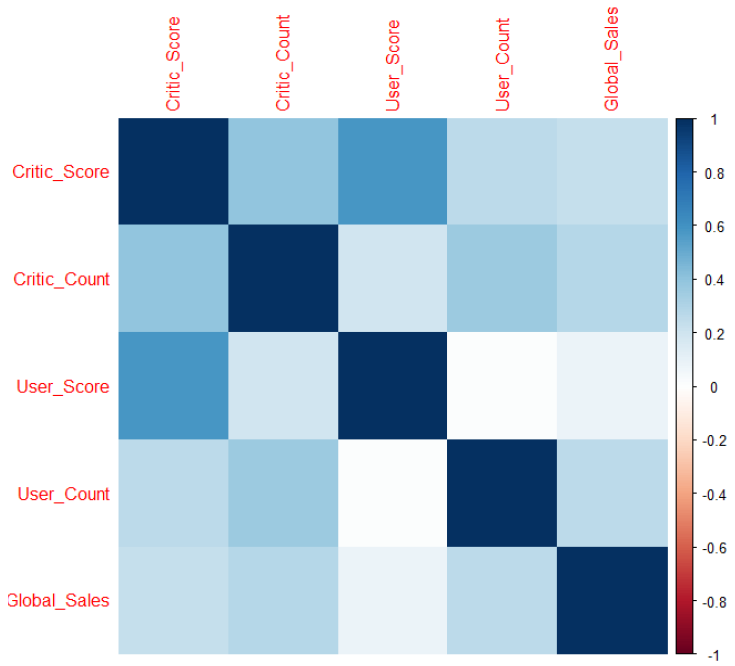
The pie chart is made in R using the pie() function, which takes a vector of positive values as input.

Here is a pie chart of game genres. We can see that the biggest slice percent is for action games, which is 23.94%, and the smallest game genres percent is for puzzle games, which is 1.74%. In Platform pie charts we have largest data for PS2 i.e.16.71% while smallest data for DC which is

0.2%. Similarly if you see ratings pie chart, the highest percentage for T rating and lowest percentage for both AO and KA ratings.

Figure 3.

Correlation Matrix

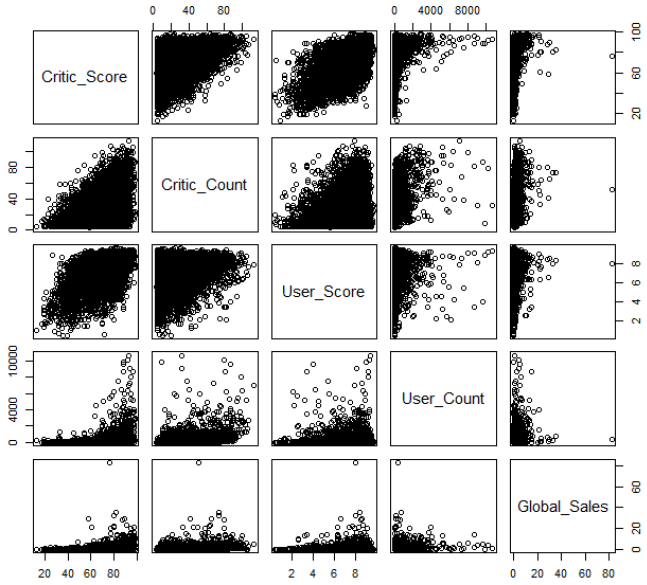


A correlation matrix is a table of correlation coefficients for a set of variables that is used to determine whether or not the variables are related. The coefficient denotes both the magnitude and the direction of the link (positive vs. negative correlations).

The purpose of correlation plots is to evaluate the association between two variables. When two variables have a positive correlation, their value will be near 1, and when they have a negative correlation, their value will be close to -1. In contrast, when two variables are not associated, their value will be close to 0. Based on the correlation analysis, it can be concluded that the User Score and Critic Score have a strong positive correlation.

Figure 4.

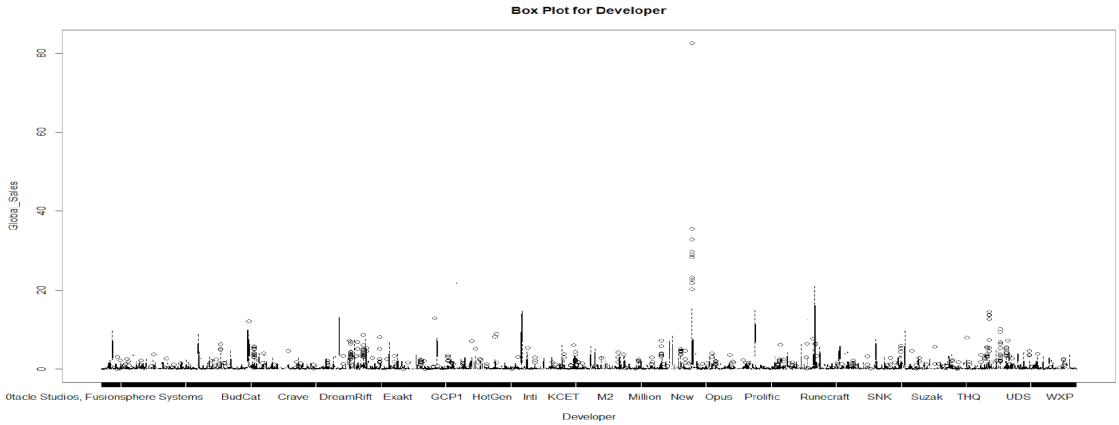
Scatter Plot

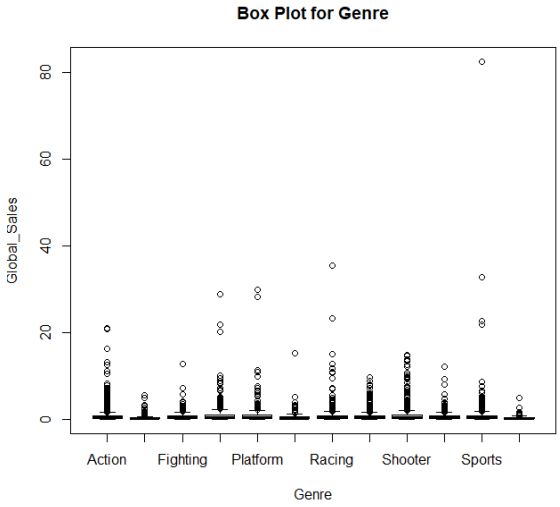


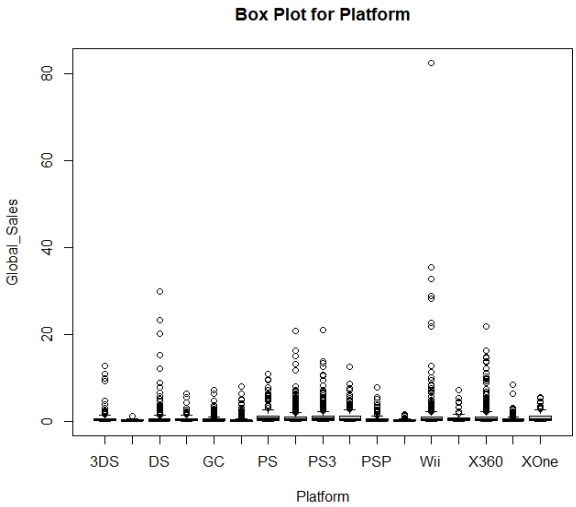
The scatter diagram compares two sets of numerical data, one variable on each axis, to see if there is a link between them. The points will fall along a line or curve if the variables are correlated. The stronger the link, the closer the spots will be to the line. The variable names are displayed in the diagonal boxes.All other boxes show a scatterplot showing the association between each pairwise variable combination. The box in the upper right corner of the matrix, for example, presents a scatterplot of numbers for global sales and critic score. The box in the upper left corner displays a scatterplot of critic score and critic count values.

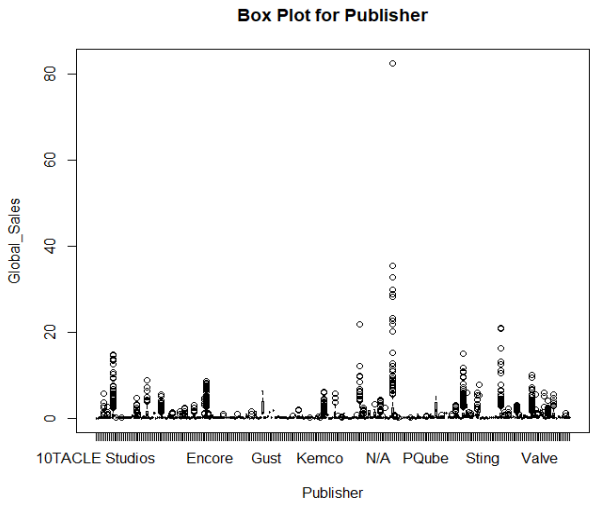
Figure 5.

Box Plots









Box plots are frequently used to depict the median, interquartile range, and any outliers in the distribution of a continuous variable. Box plots can also be used to compare continuous variable distributions across groups.Box plots aid in the visualization of the distribution of quantitative data in a field. They are also useful for comparing categorical variables or spotting outliers, if any of these are present in a dataset. In most cases, box plots visually depict the minimum value, 25th percentile (aka Q1), median (aka 50th percentile), 75th percentile (or Q3), and maximum value.

We drew a box plot with regard to our response variable, Golab Sales. For Example, Box plots for genre, platforms, publishers, and developers are available here. When it comes to genre, we can see Actine, Fighting, Platform, Racing, and the highest value with sports genre.

5) Machine Learning Algorithms

Due to the capacity of Machine Learning Algorithms to recognize patterns and trends in massive datasets, machine learning algorithms have gained significance in the field of business analytics. With the aid of these algorithms, predictive models that foretell future trends, pinpoint potential risks, and support business decision-making can be created.

Machine learning algorithms come in a wide variety, each with advantages and disadvantages. Unsupervised learning algorithms are used to identify patterns and relationships in unlabeled data, whereas supervised learning algorithms are used to make predictions based on labeled training data. Other machine learning algorithms include reinforcement learning and semi-supervised learning. Popular algorithms include decision trees, random forests, support vector machines, neural networks, logistic regression, and linear regression. In order to choose the best strategy for a particular problem and produce accurate and dependable results, it is crucial to understand the advantages and disadvantages of these algorithms.

5.1. Multivariate Regression Model

A statistical modeling method called multivariate regression is used to examine the relationship between two or more independent variables and a dependent variable. It is critical to separate the available data into two sets, training data and test data, in order to construct an accurate and trustworthy regression model.

$$Y = \beta 0 + \beta 1X1 + \beta 2X2 + \beta 3X3 + ... + \beta nXn + \varepsilon \quad \ldots\ldots \text{ Equation(1)}$$

The model is typically trained and tested in a 70:30 split. Accordingly, the model is trained using 70% of the available data, and its performance is tested and assessed using the remaining 30%. By determining the best values for the regression coefficients, the training data set is used to fit the regression model. The model is then put to the test using the test data set to gauge how accurate it is at forecasting. When all predictor variables are zero, the "Intercept" is estimated to have a coefficient of -0.8298, which means that the average global sales are predicted to fall by 0.8298 million units. The positive coefficient estimates for "Critic_Score," "Critic_Count," and "User_Count" show a correlation between rising levels of these predictor variables and rising levels of global sales. Indicating that a rise in user score is linked to a fall in global sales, the coefficient estimate for "User_Score" is negative. The model's "Multiple R-squared" value of 0.1452 shows that it accounts for 14.52% of the response variable's variability. The number of predictor variables in the model is taken into account when calculating the "Adjusted R-squared" value, which is 0.1445. The model as a whole is significant, and at least one of the predictor variables is helpful in predicting global sales, according to the "F-statistic" of 206.3 with a very low p-value.

To evaluate a regression model's goodness of fit, a residual plot is used. It contrasts the predicted values with the residuals, which are the discrepancies between the observed values of the dependent variable and the predicted values from the regression model. The following can be determined with the aid of the residual plot: The fact that the residuals take the shape of a funnel suggests that their variance is not constant over the range of the predicted values. Heteroscedasticity is what this is, and it deviates from a basic tenet of linear regression. In this situation, it might be necessary to transform the model or take another approach to modeling into account.

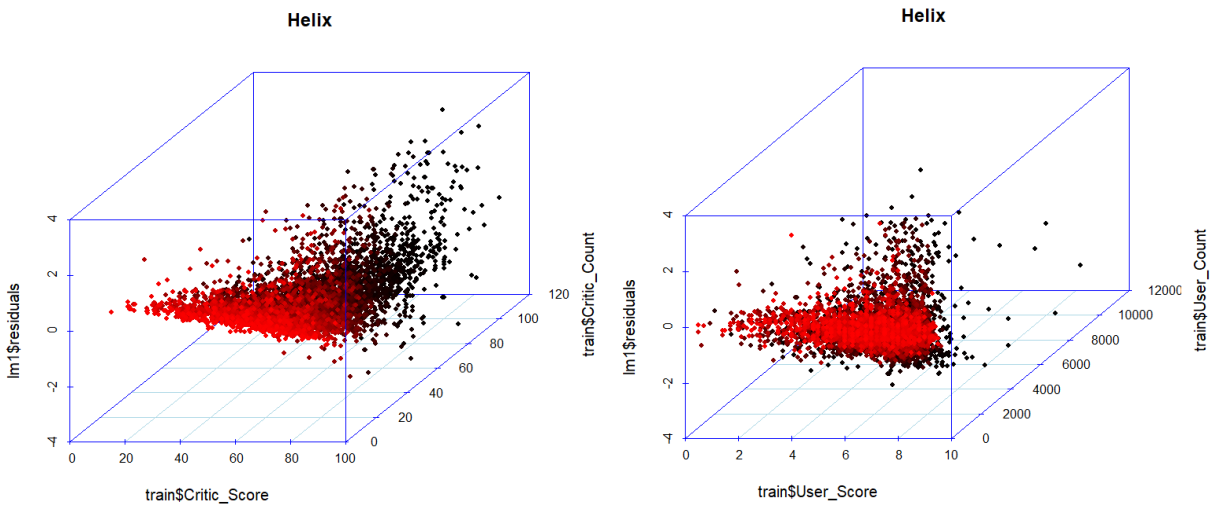


Figure 6. Residual Plots

To better fit the data in this situation, we employ a quadratic or cubic polynomial regression model. A cubic model has two curves, whereas a quadratic model only has one. We can capture more intricate relationships between the variables and make predictions that are more precise by adding more polynomial terms. It is crucial to remember that including higher-order polynomial terms can also result in overfitting of the model, where the model performs poorly on fresh data despite fitting the training data very well. As a result, it is crucial to carefully assess the model's performance on a test set before using it to make predictions.

Model	MSE
Linear	5.685017
Quadratic Polynomial	5.771518
Cubic Polynomial	5.729701

Table 2. MSE values for Polynomials

The linear regression model, which has the lowest mean squared error (MSE) of 5.685017, has the smallest difference between the predicted and actual values. The best fit for these data is the linear regression model. When choosing a regression model, it's crucial to keep other aspects in mind as well, like the model's complexity and the results' readability.

5.2. Feature Selection - Forward Stepwise and Backward Selection

The process of selecting features for predictive models is crucial because it aids in determining which features are most crucial for the target variable's prediction. Backward stepwise selection and forward stepwise selection are two frequently used feature selection techniques.

According to a predetermined criterion, such as the Akaike Information Criterion (AIC) or Bayesian Information Criterion (BIC), forward stepwise selection is a sequential feature selection method that begins with an empty model and adds the most significant feature at each step. The process keeps going until the criterion does not continue to improve. On the other hand, backward stepwise selection begins with a complete model that contains all features and eliminates the least important feature at each stage until no further improvement in the criterion is achieved. Backward selection employs AIC or BIC as the criterion to assess the model's performance, much like forward selection does.

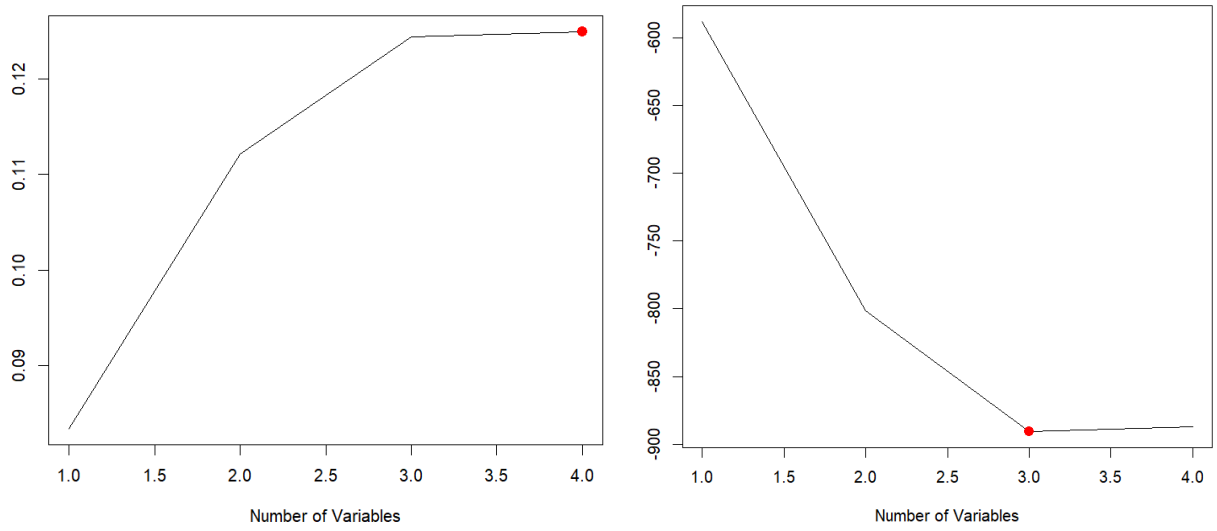


Figure 7. Maximum and minimum number of variables

The specific objective and selection criteria for the variables determine which subset is the best. The subset that contains all four variables in this situation might offer the best performance in terms of correctly fitting the model and making predictions. However, a smaller subset with fewer variables may be preferred if overfitting is a concern or if interpretability is

crucial. When choosing variables for the model, it's crucial to take the specific business context and domain knowledge into account.

The best subset of variables for the variables Critic_Score, Critic_Count, User_Score, and User_Count can be determined by comparing their BIC and adjusted R-squared values. The best subset is the one with the smallest BIC and the highest adjusted R-squared.

5.3. Ridge Regression Model

The feature selection model fits a linear model with a subset of the predictors using a subset method that employs least squares. As an alternative, we can fit a model with all p predictors by employing a method that restricts or regularizes coefficient estimates, or, to put it another way, decreases coefficient estimates in the direction of zero. It turns out that Shrinking the coefficient estimates can dramatically lower the variance of the fit.

Ridge regression is a type of regularized linear regression model that is used when the independent variables (or features) in a multivariate regression model are highly correlated. In such cases, standard linear regression can produce biased and overfitted models.

From a Ridge Regression model, one can infer the impact of each predictor on the response variable, and the extent of the impact is determined by the magnitude of the corresponding regression coefficient. We have 4 parameters that we shrink towards zero.

$$Y = \beta0 + \beta1X1 + \beta2X2 + \beta3X3 + ... + \beta nXn + \varepsilon + \lambda\Sigma(\beta i^2) \quad \ldots\ldots \text{ Equation(2)}$$

```
> bestlam

[1] 0.1357554

> mean((ridge.pred - y.test2)^2)   #test error

[1] 3.321629
```

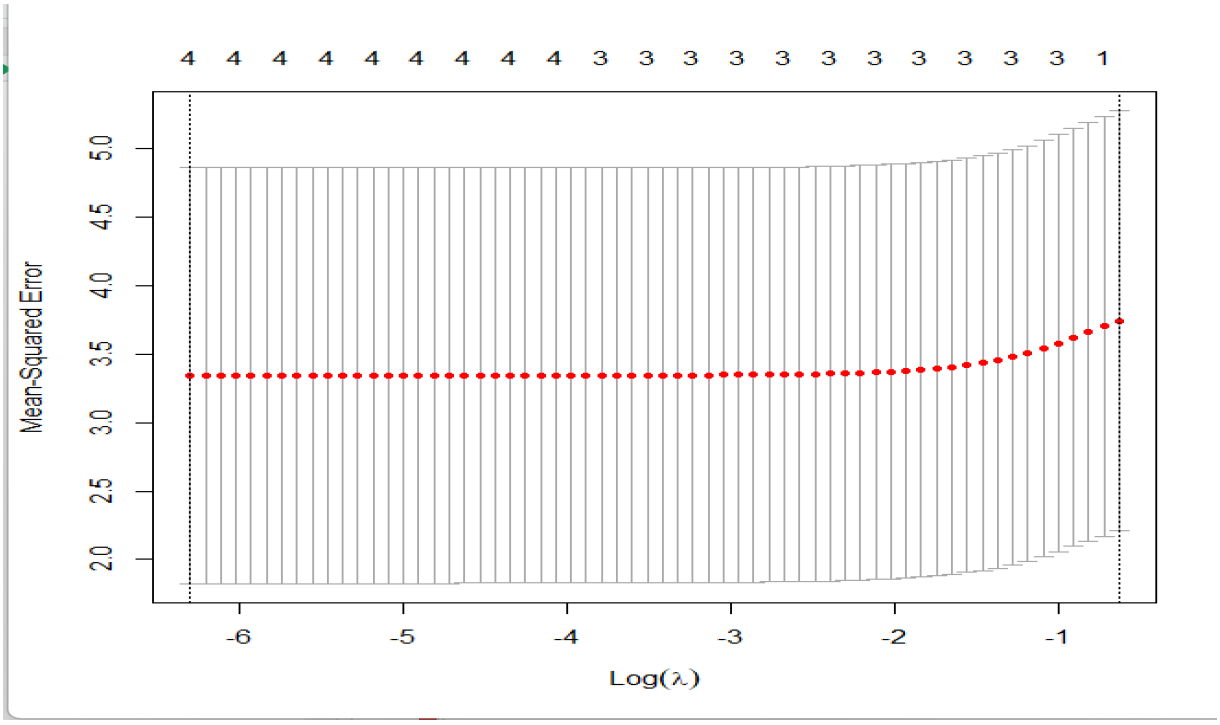


Figure 8. Ridge Regression Model

MSE for our ridge regression model is **3.321629.** For this model parameters **Critic_score** and **Critic_count** have the highest impact out of all predictors or features.

5.4. Lasso Regression Model

Although Ridge regression is a good technique, it does have a disadvantage. Unlike subset selection, ridge regression includes all p predictors in the final model. This is where Lasso comes into the picture. Lasso is an alternative model for this problem. It shrinks all coefficient estimates towards zero and some might be forced to be exactly zero. Thus, Lasso performs variable selection, that is it involves only a subset of variables.

$$Y = \beta 0 + \beta 1X1 + \beta 2X2 + .. + \beta nXn + \varepsilon + \lambda \Sigma|\beta i| \qquad \ldots\ldots \text{ Equation(3)}$$

```
> bestlam1
[1] 0.001836835
> lasso.mod = glmnet(x.train2, y.train2, alpha = 1, lambda = bestlam1)
> lasso.pred = predict(lasso.mod, s = bestlam1, newx = x.test2)
> mean((lasso.pred - y.test2)^2)  # test error
[1] 3.316236
> lasso.coef = predict(lasso.mod, type = "coefficients")
> lasso.coef
5 x 1 sparse Matrix of class "dgCMatrix"
                        s0
(Intercept)  -0.9644063504
Critic_Score  0.0196321911
Critic_Count  0.0175890540
User_Score   -0.0341119370
User_Count    0.0005028443
```

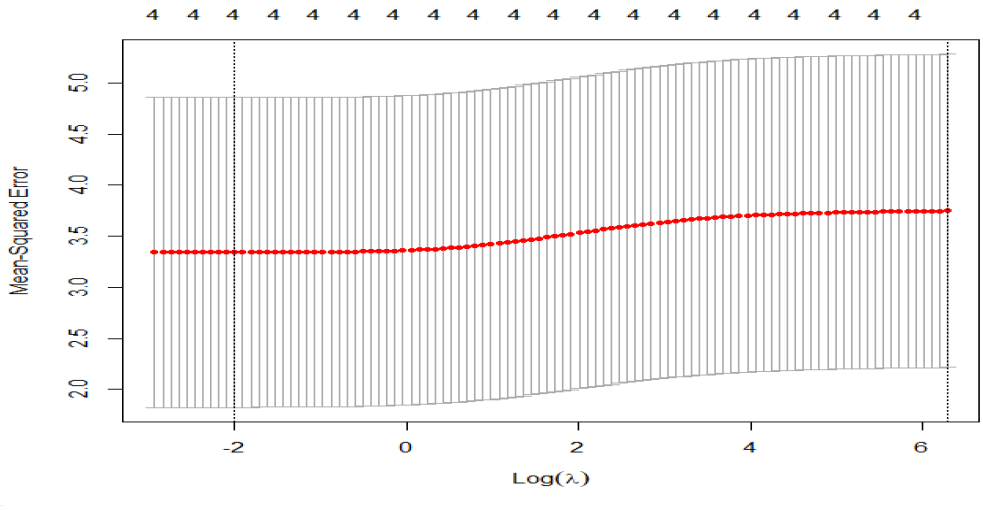


Figure 9. Lasso Regression Model

The MSE for Lasso Regression model is **3.321236.** Here parameters like **Critic_Score** and **Critic _Count** has higher impact on Global Sales.

5.5. Decision Tree

Tree-Based Methods:

In a tree based model we will be stratifying or segmenting the predictor space into different numbers of simple regions. Decision-tree methods involve creating a tree-like structure that summarizes a set of rules used to divide the predictor space into segments. Decision trees can be applied to both regression and classification problems.

Decision Tree:

- Trees are easy to explain to others, even easier than linear regression.
- Decision trees better resemble human decision-making as compared to the regression and classification techniques.
- Trees can be visually represented and can be easily understood, even by a non-expert.
- Unfortunately, trees typically lack the same level of predictive precision as some of the other regression and classification techniques.

However, **by aggregating many decision trees,** the predictive performance of trees can be substantially improved.

When **multiple trees are combined,** it can often lead to significant **enhancements in prediction** accuracy. However, this comes at the cost of losing some interpretability.

For the **Video Game Sales** dataset we have used the major predictors such as **Critic_Score, Critic_Count** and **User_Count** to predict the majority of sales of the video games.

For the **Video Game Sales** dataset, a regression tree for predicting the sales of video games based on the User count, critic scores, critic count.

> install.packages("tree")

> library(tree)

Install tree package if you have never used it before.

We created two trees with **50% of the data set and another with 70%** of the dataset.

Sample test with 50% data set yielded the following results:

```
> set.seed(151)
> index4 = sample(n, round(0.5*n))
> tree1 = tree(Global_Sales~., data = data_subset1, subset = index4)
> summary(tree1)

Regression tree:
tree(formula = Global_Sales ~ ., data = data_subset1, subset = index4)
Variables actually used in tree construction:
[1] "User_Count"   "Critic_Score" "Critic_Count"
Number of terminal nodes:  7
Residual mean deviance:  3.887 = 13480 / 3467
Distribution of residuals:
     Min.   1st Qu.    Median      Mean   3rd Qu.      Max.
-11.71000  -0.35970  -0.22970   0.00000   0.09025  70.80000
```

summary() indicates that only three variables have been used in the decision tree.

User_Count (Number of users who gave the user_score), **Critic_Count** (The number of critics used in coming up with the Critic_score), **Critic_Score**(Aggregate score compiled by Metacritic staff). The tree1 yields 7 terminal nodes with a mean test error of **3.887.**

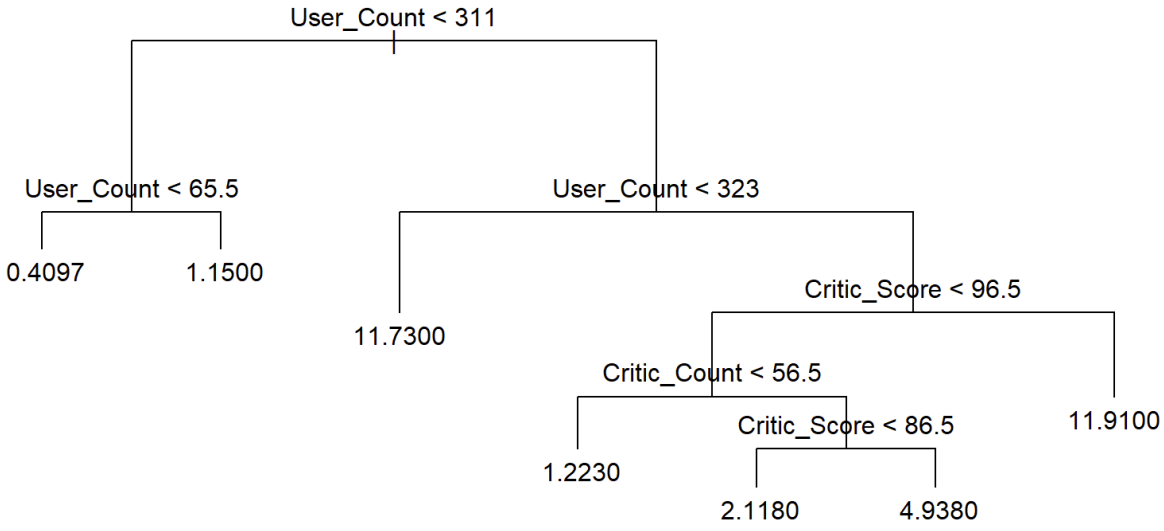


Figure 10. Decision Tree

Sample test with 70% data set yielded the following results:

```
> set.seed(10121)  #10121
> index5 = sample(n, round(0.7*n))
> tree2 = tree(Global_Sales~., data = data_subset1, subset = index5)
> summary(tree2)

Regression tree:
tree(formula = Global_Sales ~ ., data = data_subset1, subset = index5)
Variables actually used in tree construction:
[1] "User_Count"   "Critic_Score"
[3] "Critic_Count"
Number of terminal nodes:  6
Residual mean deviance:  3.482 = 16910 / 4857
Distribution of residuals:
    Min.  1st Qu.   Median
-10.8000  -0.3518  -0.2192
    Mean  3rd Qu.     Max.
  0.0000   0.1082  71.6800
```

The tree2 yields 6 terminal nodes with a mean test error of **3.482**

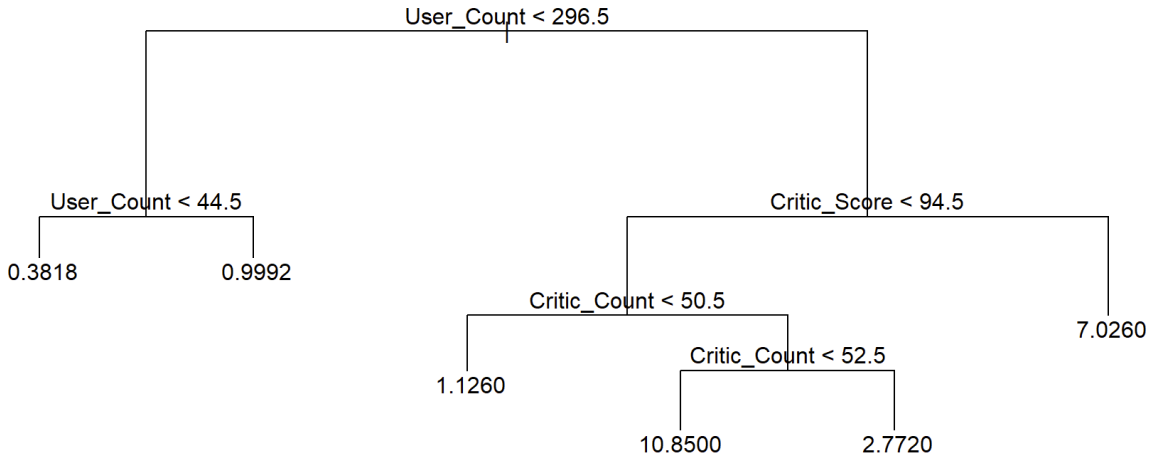


Figure 11. Decision Tree

Linear regression models:

A linear regression model**(lm5)** with global sales **data_subset1 is created with 50% dataset** included in the **index4**.

The lm5 has a mean square error of **3.321559**

```
> lm5= lm(Global_Sales~., data = data_subset1, subset = index4)
> summary(lm5)

Call:
lm(formula = Global_Sales ~ ., data = data_subset1, subset = index4)

Residuals:
   Min     1Q Median     3Q    Max
-5.414 -0.562 -0.215  0.175 81.117

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  -1.062e+00  2.000e-01  -5.312 1.15e-07 ***
Critic_Score  1.868e-02  3.400e-03   5.494 4.22e-08 ***
Critic_Count  1.843e-02  2.134e-03   8.636  < 2e-16 ***
User_Score   -1.536e-02  3.066e-02  -0.501    0.616
User_Count    7.428e-04  7.169e-05  10.362  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 2.09 on 3469 degrees of freedom
Multiple R-squared:  0.1196,    Adjusted R-squared:  0.1186
F-statistic: 117.8 on 4 and 3469 DF,  p-value: < 2.2e-16
```

A linear regression model**(lm6)** with global sales **data_subset1 is created with 70% dataset** included in the **index5**.

The lm6 has a mean square error of **3.336923**

```
Call:
lm(formula = Global_Sales ~ ., data = data_subset1, subset = index5)

Residuals:
   Min     1Q Median     3Q    Max
-6.091 -0.551 -0.218  0.179 81.184

Coefficients:
               Estimate Std. Error t value Pr(>|t|)
(Intercept)  -8.907e-01  1.602e-01  -5.561 2.82e-08 ***
Critic_Score  1.855e-02  2.693e-03   6.886 6.48e-12 ***
Critic_Count  1.926e-02  1.640e-03  11.750  < 2e-16 ***
User_Score   -4.109e-02  2.383e-02  -1.725   0.0847 .
User_Count    5.384e-04  5.007e-05  10.753  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 1.936 on 4858 degrees of freedom
Multiple R-squared:  0.1175,    Adjusted R-squared:  0.1167
F-statistic: 161.6 on 4 and 4858 DF,  p-value: < 2.2e-16
```

Two models are giving the same results: except **User_score**, all other variables are significant. R2 are close.

The predictions of Trees and linear regression models are as follows:

```
> mean((data_subset1$Global_Sales - pred.tree3)^2)
[1] 3.236701
> mean((data_subset1$Global_Sales - pred.tree4)^2)
[1] 3.192505
> mean((data_subset1$Global_Sales - pred.lm5)^2)
[1] 3.336923
> mean((data_subset1$Global_Sales - pred.lm6)^2)
[1] 3.321559
```

The Mean Square Errors of trees are different. But MSE for linear regression models are little closer, which means linear regression models are robust while trees are not.

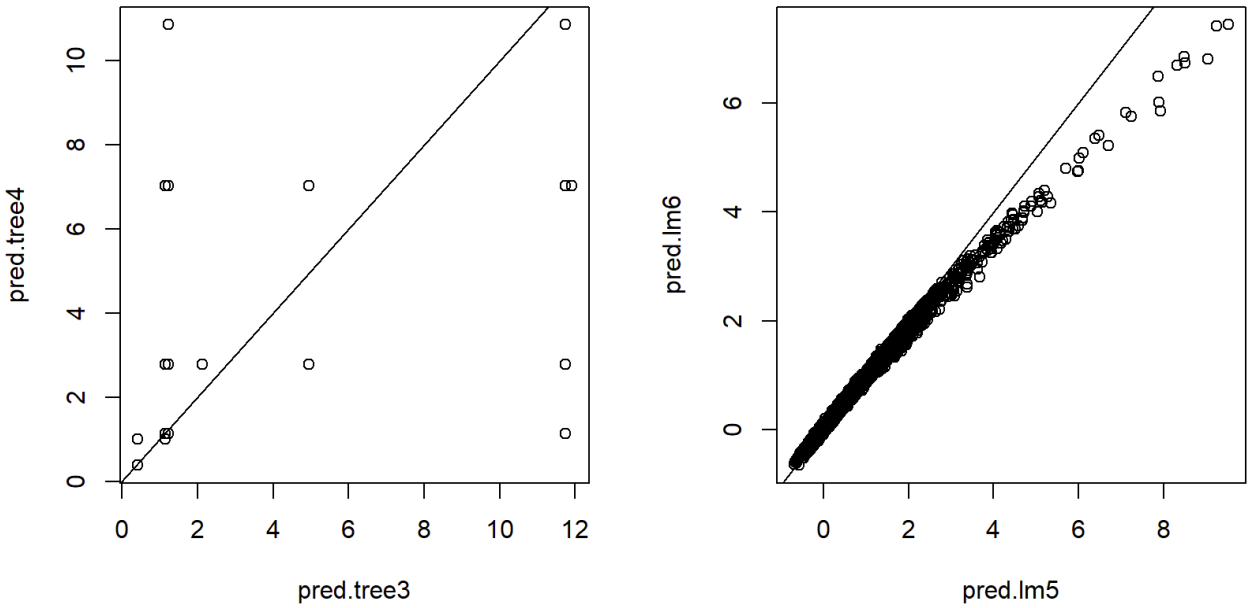


Figure 12. Residual Plot of prediction values

The plot on the left displays the predictions made by **tree4 and tree3**, while the plot on the right shows the predictions made by **lm6 and lm5**. The straight line represents **y = x.** The predictions made by trees are noticeably different, which indicates that **trees are not robust**.

5.6. Bagging

Bootstrap aggregation, also known as bagging, is a commonly used technique for decreasing the variance. We apply this to decision trees as they suffer greatly with high variance.

When a set of observations is averaged, it lowers the variance. Therefore, one way to enhance the accuracy of predictions is to obtain several training sets for the population, create a separate prediction for each training set, and average the resulting predictions. Although it is not practical to access multiple datasets, the **Bootstrap approach** allows us to use a computer to mimic the process of obtaining new data sets, so that we can have access to **'multiple training data sets'.**

To generate these **'bootstrap data sets'**, each set is produced by randomly sampling observations with **replacement from the original dataset**. Each bootstrap set has the same size

as the original dataset. As a result, some observations may appear multiple times in a given bootstrap set, while others may not appear at all.

With the **bagging approach**, we create B distinct bootstrapped training datasets. We then train our method on the b-th bootstrapped training set to generate f ∗b(x), which is the prediction for a point x. Next, we average all of the predictions to obtain ˆfbag(x), using the formula:

ˆfbag(x) = 1/B * Σ(ˆf ∗b(x)), where b ranges from 1 to B. This process is known as bagging.

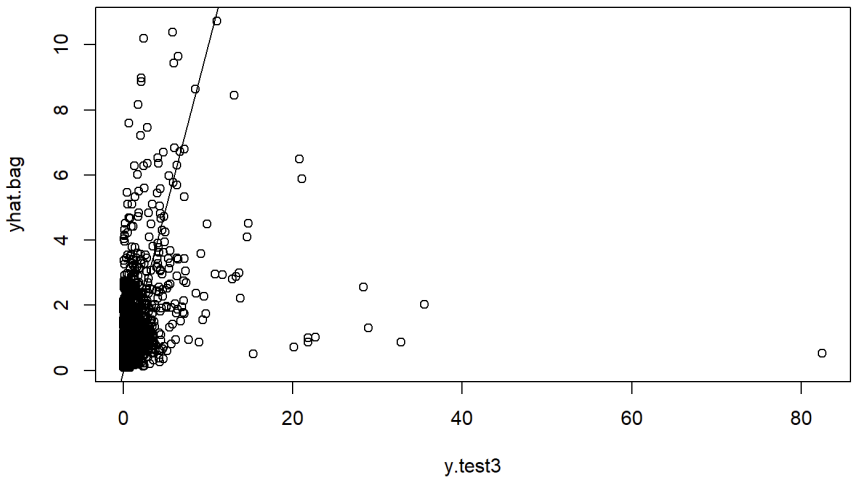


Figure 13. Bagging

5.7. Random Forest

Random forests provide an improvement over bagged trees by way of a small modification that reduces correlation between the trees. This reduces the variance when we average the trees.

As in bagging, we build a number of decision trees on bootstrapped training samples.When constructing decision trees in a random forest, **a random subset of m predictors is chosen** from the full set of p predictors each time a split is considered. The **split is only allowed to use one predictor** from the selected subset of m.

At **each split** in a random forest, a **new set of m predictors** is selected randomly, where typically **m** is approximately equal to the square root of the total number of predictors (i.e., m ≈ √p).For example in the **VideoGameSales** dataset , there are **6 predictors** in total, and approximately **3 predictors** would be **considered at each split**.

When **m = √p,** the algorithm cannot consider most of the available predictors during each split in the tree. Although this may sound strange, there is a clever reasoning behind it.

Consider a scenario where there is a single strong predictor in the dataset, alongside several other predictors that are moderately strong. In a collection of bagged trees, it is likely that most or all of the trees will use this strong predictor in their top split. As a result, all the bagged trees will look quite similar, and the predictions they produce will be highly correlated. When predictions are highly correlated, averaging them **does not significantly reduce the variance.**

To overcome this, we have Random Forest which forces each split to consider only a subset of predictors. It is a process of decorrelating and hence more reliable. The data was split randomly into a **training set and a test set**. The random forest algorithm was applied to the training set with three different values of the number of predictors considered at each split.

```
> rf

Call:
 randomForest(formula = Global_Sales ~ ., data = train3, mtry = 3,      importance = TRUE)
               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 1.352242
                    % Var explained: 52.85
> importance(rf)
               %IncMSE IncNodePurity
Critic_Score 18.215555     148.74665
Critic_Count  8.844763      44.25025
User_Score    3.766558      15.12819
User_Count   11.894910     112.30037
```

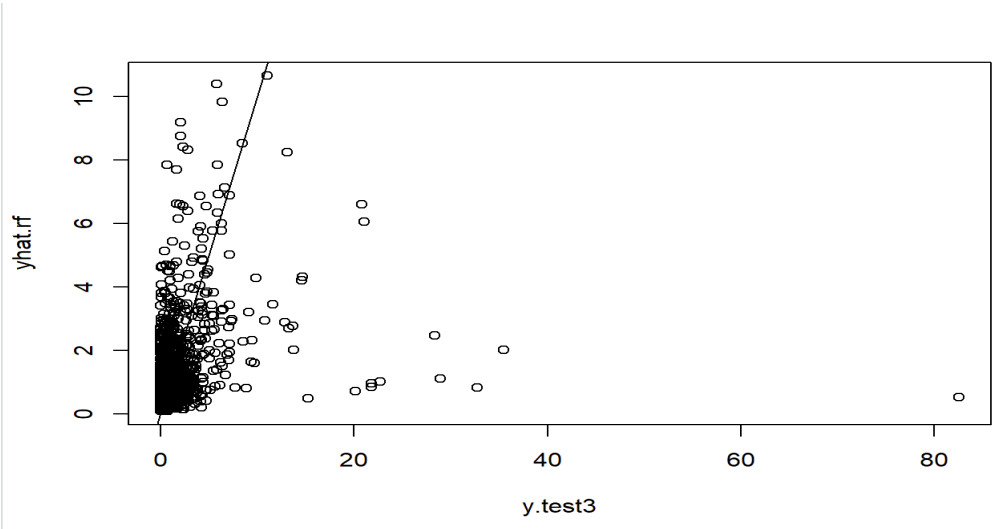


Figure 14. Random Forest Plot

Important variables:

The results indicate that across all of the trees considered in the random forest, the Global Sales of video games are highly predicted by **criticScore and the userScore are by far the two most,**

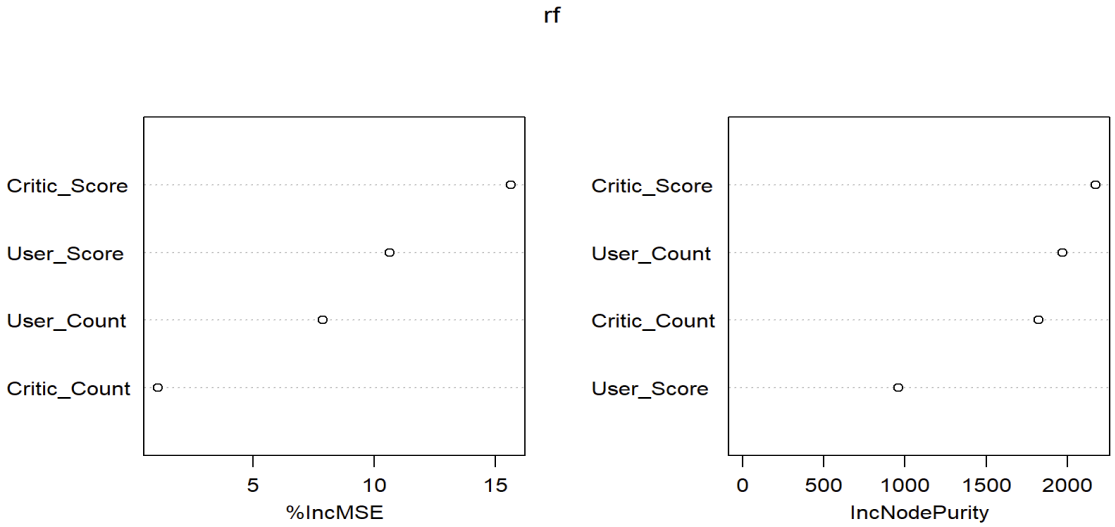


Figure 15. Important Variables in Random Forest

6. Results

Models	Test Error
Multivariate Regression	**5.685017**
Lasso Regression	**3.316236**
Ridge Regression	**3.321629**
Bagging	**4.846291**
Random Forest	**3.311559**

7. Conclusion

- In our regression analysis, the lasso model has the lowest test error with **3.321**.
- Comparing between the tree-based Decision tree models, Random forest and Bagging, the Random Forest method has the lower test error with **3.3159.**
- **Random forest** has several advantages over other models, such as its **ability to handle high-dimensional data, reduce overfitting**, and provide feature importance measures. For **prediction of global sales** we can conclude **Random forest** to be the best model.
- **Critic Score, User Score and Critic Count** has a higher impact on Global Sales.

8. References

1. Kaggle datasets https://www.kaggle.com/datasets/rush4ratio/video-game-sales-with-
2. Video Game Industry Analysis" by Simon Egenfeldt-Nielsen, Jonas Heide Smith, and Susana Pajares Tosca.
3. "Video Game Sales Data" by VGChartz: This website provides a database of video game sales data, including global sales figures and regional breakdowns. The data covers a wide range of platforms and genres, and can be filtered by date, region, or platform.