I split the cleaned data set into 80% train and 20% test sizes.

I started with a simple linear regression analysis on each of the variables excluding Sales from the other regions against Global Sales. I didn't check the assumptions of regression at this time and only checked for the significance of the variable. I converted categorical variables into dummy variables and used the most populated group as the reference group.

Platform, Genre, Publisher, Critic_Score, Critic_Count, User_Score, User_Count, Developer, and Rating are the variables that were significant for a simple linear regression model.

I recoded the Publisher and Developer columns with a prefix of P_ and D_ respectively to avoid overlap for categories such as 'Other' and 'Nintendo'.

Next, I started building a multiple linear regression model. I used all previously significant variables at the univariate analysis level in the first multiple linear regression model. I tried to remove each of them one at a time and used a partial F test to determine whether the model can be reduced. Specifically, I was testing the hypotheses:

H_0: The reduced model is better (without variable X)

H_1: The full model is better (with variable X)

I rejected the null hypothesis that the reduced model is better for all of the variables found to be significant in the univariate analysis.

Next, I tried adding variables that were not found to be significant in the univariate analysis to the model and used partial F tests to determine if I should keep them in the model.

I found that I should not include Dev_same_publisher and that I should include Years_Since_Release in the model.

Since there can be a large number of interactions in the model, I checked just a few that made sense. It makes sense to think that the Critic_Count and User_Count can interact with Critic_Score and User Score, even though the weight of the Critic_Score is inside Critic_Count and same for users. It also makes sense for Years_Since_Release to interact with User_Count and Critic_Count because the longer a game has been out, the more likely a user or critic has given a game a score. It also makes sense that Critic_Count can interact with Rating. Critics may be more likely to focus a certain Rating.

I try adding the interactions from above one at a time and use a partial F test to determine if I should include them in the model as before. I find Critic_Score * Critic_Count, Critic_Count * Years_Since_Release, and Critic_Count * Rating to be significant.

Model diagnostics indicate that homoscedasticity is violated by a predicted vs residuals plot.

By checking a scatter plot between Critic_Count, Critic_Score, User_Count, and User_Score against Global_Sales, I find that Critic_Score and User_Score have a non-linear relationship with Global_Sales. I add the quadratic term Critic_Score^2 and User_Score^2 to the model.

The residuals did not improve, so I tried a box-cox transformation. The residuals improved and the R-squared value increased substantially, but homoscedasticity was not met.

Next, I tried a natural log transformation for Global_Sales. Residuals looked the same but the R-squared value was higher by 0.09.

I also tried weighted regression, but the residuals did not improve and the R-squared value was worse. I try recoding variables, removing variables from the model, all to no avai of fixing the residuals. I decide to try three models. The first one is with all the variables I have in the full model. It Includes Critic_Score, Critic_Score_Squared, Critic_Count, User_Score, User_Score_Squared, User_Count, Platform, Genre, Publisher, Developer, Years_Since_Release, Rating, Critic_Score_Count, Critic_Count_Years, Critic_Count_E, Critic_Count_E10+, and Critic_Count_M The model is:

ln(Global_Sales) = -1.6495 - 0.0434Critic_Score + 0.0109Critic_Count + … - 0.0167User_Score_Squared

The second model has slightly better residuals but less R-squared value. It is the 'best' statistical model I could find, trading between residuals and R-squared value. It includes Critic_Score, Critic_Count, Critic_Score_Squared, Genre, Rating, and Years_Since_Release.

ln(Global_Sales) = -1.4728 – 0.0554Critic_Score – 0.5536Adventure + … + 0.0006Critic_Score_Squared

The third model has about the same residuals as Model 2, but has a different shape. The R-squared value is worse. It includes Platform, Genre, and Rating :

ln(Global_Sales) = -1.1408 – 0.1346(3DS) – 0.1935DS + … + 0.3839M

I predicted Global_Sales for the test set using the above coefficients for each model. I applied an exponential function on ln(Global_Sales) to get rid of the natural log. Next, I computed the absolute difference between the actual value and predicted value. I summed up the differences and found that Model 1 had the lowest value of 705.66 million dollars. Model 2 had 765.38 million dollars. Model 3 had 850.13 million dollars.

It turns out that the model with the highest R-squared value had the best predictive power.

Overall, it didn't seem like any of the models were any good. Since we care about prediction, it may be better that we remove large outliers because we care about having an overall good prediction. The presence of those outliers may make our predictions less accurate despite that they are valid observations.