Notes for Data Science:

Needed to group data from our 3 tables into 2 main tables for the Linear Regression.

I started by joining “Games\_Tidy” with “steam” to group together necessary columns that we need for our linear regression model. I used an inner join to avoid filling up a ton of our rows with NA values that would cause problems with our linear models. Unfortunately, this join limited our data to 140 observations before doing our train/test split. We agreed that this was the only way to move forward instead of filling in the “NA”s with random values which would skew our results far too much. We had a lot of NAs. This is a problem because we aren’t going to have a lot of data to work on but this is a proof of concept that can easily be scaled up given some bulkier data sets.

I wanted to find out which metrics correlated to the highest “Est. Revenue”. I selected columns from each table that I thought might influence the Est. Revenue and used the lm() function to find which values had the best correlation to Est. Revenue. After using this technique, we were left with Price, Est. Units Sold, steam\_rating, Reviews, positive\_ratings, and negative\_ratings that correlated to Est. Revenue. From there I split the data 80-20 and trained it using

* Model <- train(`Est. Revenue` ~ Price + `Est. Units Sold` + steam\_rating + Reviews + positive\_ratings + negative\_ratings, data = train\_data, method = "lm")

Where “train\_data” is the 80% from our split. I ended up with an HUGE RMSE…. "Root Mean Squared Error (RMSE): 86803042.0648195" and realized how much of a problem we had.   
  
So, I ended up trying other combinations of splitting up the data… a 70-30 split resulted in an RMSE of 95359476.2755914. While a 90-10 split resulted in "Root Mean Squared Error (RMSE): 65058985.6842683".

This made me think we might just be suffering from lack of data, but I figured we might have outliers or a better formula we could find.

After further EDA of my joined table even though there weren’t any NA’s, we had a lot of 0’s in est. revenue that were skewing our data. We decided to prune them to see how our model fared. This pruning helped our results but limited our data to 105 entries.

After filtering our any Est. Revenues with 0 I re-ran the model I described above with a 90-10 split to get "Root Mean Squared Error (RMSE): 54041405.2882865" still rather large but better for sure. This was the first time I realized that I might be like one of the data scientists you described when we were talking about models that were manipulating the data for the sake of a lower RMSE as opposed to changing models or just reporting the findings. I decided to consult the team to plan on our findings or to go about the data differently.

We decided from there to split up our “mega” model described above to find out whether or not individual factors play into increasing the Est Revenue with models like `Est. Revenue` ~ Price or `Est. Revenue` ~ Reviews and so on. So I made a list of many different models and created a function that would train and validate each model and gather the results. This function also took in a split in the form of a decimal from 0-1 to hone in on the best split we can use.

Based on these results (RMSE, MAE, and R^2) we decided that our model that we started with (the one up at the top with all factors) produced the lowest RMSE and MAE and the R^2 closest to 1 with a split of 60-40.

Initially our RMSE and MAE seemed way too big but upon realizing that EST. Revenue ranged from 42000 to 1.2 billion our numbers seemed a bit more rational. In the end our better model resulted in an RMSE of 48 million, MAE of 31 million and R^2 of 0.84. This puts things into perspective and doesn’t seem as bad.