**Musical Regression**

**4/8/2023**

**STAT 450 Sec 02**

For this project we will be analyzing and running multiple regression tests on songs created by artists. We want to see why some songs do as well as they do. This could be important for any up-and-coming musician or even for people who just want to impress their families at gatherings. What makes a song a good song that people want to listen to? We will attempt to figure this out throughout our tests, and discover what really matters, and what all goes into making a great, banger of a song.

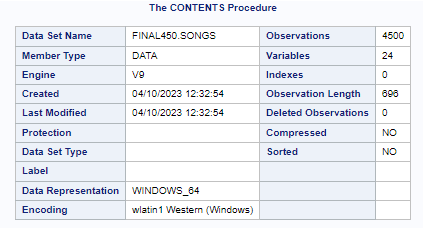
We found the data we used in this project on kaggle.com, it was put together by Salvatore Rastelli and 2 others, and was last updated February 7th, 2023. This dataset has Spotify and YouTube data for over 20,700 songs and has 26 variables. The data is in a seemingly random order with all songs by each artist grouped together. We did modify the dataset so that we could use and read it easier, first by removing some of the columns, those being Url\_spotify, Album\_type, Uri, Url\_youtube, Description, Licensed, and official\_video, as they aren’t useful for regression analysis or anything else we may do. These are the columns we did keep, starting with the easiest to explain; Track, Artist, and Album are the name of the song, artist, and album the song is on, on Spotify. Similarly, Title and Channel are the title and channel for the “official” version of the song on YouTube. Streams is the total number of plays the song has on Spotify, Duration\_ms is the length of the song in milliseconds; and Views, Likes, and Comments is the number of views, likes, and comments the YouTube video has.

The rest of the variables will come with a description directly from the creator of the dataset, as neither of us are that knowledgeable about how music is made or about Spotify’s measures they created, these descriptions are probably the best to help understand these variables. Key is the key the track is in; integers map to pitches using standard Pitch Class notation; E.g., 0 = C, 1 = C♯/D♭, 2 = D, and so on; if no key was detected, the value is -1. Tempo is the overall estimated tempo of a track in beats per minute (BPM); in musical terminology, tempo is the speed or pace of a given piece and derives directly from the average beat duration. Loudness is the overall loudness of a track in decibels (dB); loudness values are averaged across the entire track and are useful for comparing relative loudness of tracks; loudness is the quality of a sound that is the primary psychological correlate of physical strength (amplitude); values typically range between -60 and 0 db.

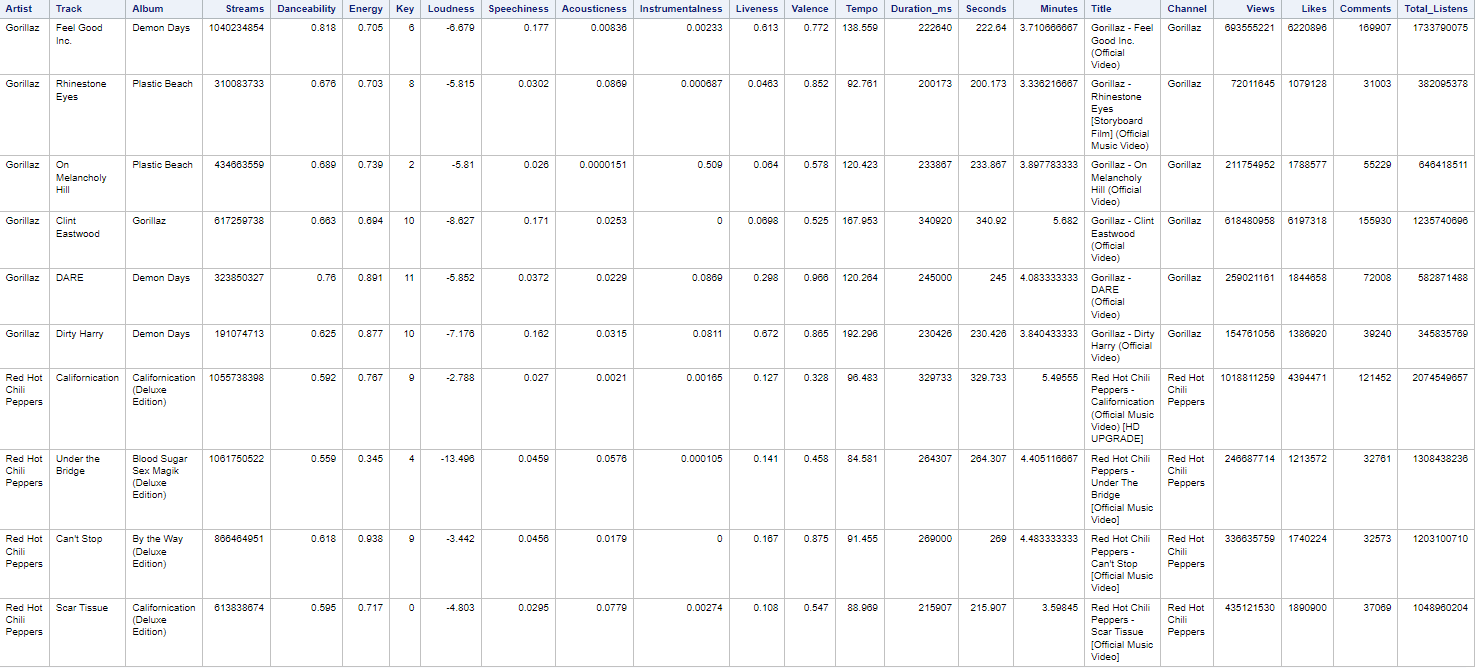
The rest of these, as far as we know, are measures that Spotify has made to add some extra information for each song, to help Spotify in recommending songs for users to listen to. Danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity; a value of 0.0 is least danceable and 1.0 is most danceable. Energy is a measure from 0.0 to 1.0 and represents a perceptual measure of intensity and activity; typically, energetic tracks feel fast, loud, and noisy; for example, death metal has high energy, while a Bach prelude scores low on the scale; perceptual features contributing to this attribute include dynamic range, perceived loudness, timbre, onset rate, and general entropy. Speechiness detects the presence of spoken words in a track; the more exclusively speech-like the recording (e.g. talk show, audio book, poetry), the closer to 1.0 the attribute value; Values above 0.66 describe tracks that are probably made entirely of spoken words, values between 0.33 and 0.66 describe tracks that may contain both music and speech, either in sections or layered, including such cases as rap music, values below 0.33 most likely represent music and other non-speech-like tracks. Acousticness is a confidence measure from 0.0 to 1.0 of whether the track is acoustic; 1.0 represents high confidence the track is acoustic. Instrumentalness predicts whether a track contains no vocals; "ooh" and "aah" sounds are treated as instrumental in this context and rap or spoken word tracks are clearly "vocal"; the closer the instrumentalness value is to 1.0, the greater likelihood the track contains no vocal content; Values above 0.5 are intended to represent instrumental tracks, but confidence is higher as the value approaches 1.0. Liveness detects the presence of an audience in the recording; higher liveness values represent an increased probability that the track was performed live; a value above 0.8 provides a strong likelihood that the track is live. And finally, Valence is a measure from 0.0 to 1.0 describing the musical positiveness conveyed by a track; tracks with high valence sound more positive (e.g., happy, cheerful, euphoric), while tracks with low valence sound more negative (e.g., sad, depressed, angry).

We also modified the data by removing any rows with empty cells in any of these numerical variables, as they wouldn’t be useful for us. This part of removing rows left us with 19,549 songs usable for regression. Then we added 3 new columns, one called Seconds, which is Duration\_ms divided by 1000 to get the duration of each song in seconds; this isn’t any different from Duration\_ms for the purposes of regression, but we decided to make it anyways. Similarly, the second is Minutes, just dividing the Seconds variable by 60 to get the duration of each song in minutes. The third is Total\_Listens, which combines Streams and Views for a larger total that we can use instead.

After further evaluation, we discovered that SAS does not perform well with regression data sets over 5,000 observations so we lowered it. We dropped it down from 19,549 songs to 4,500. We decided to sort the songs by our variable Total\_Listens and took the 4,500 songs with the most Total\_Listens since we thought these would be the most important songs to run regression analysis on.

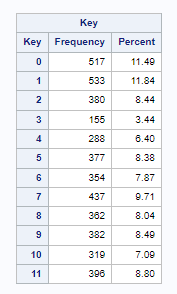


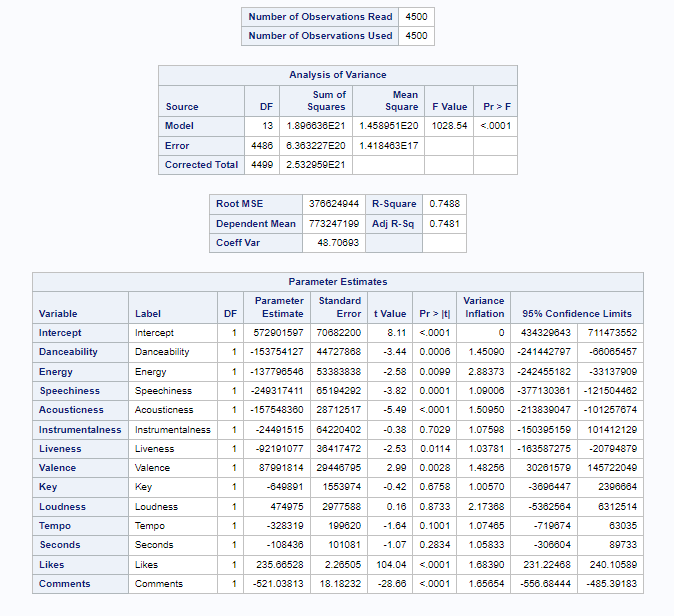
* The Print procedure shows what our data set looks like.



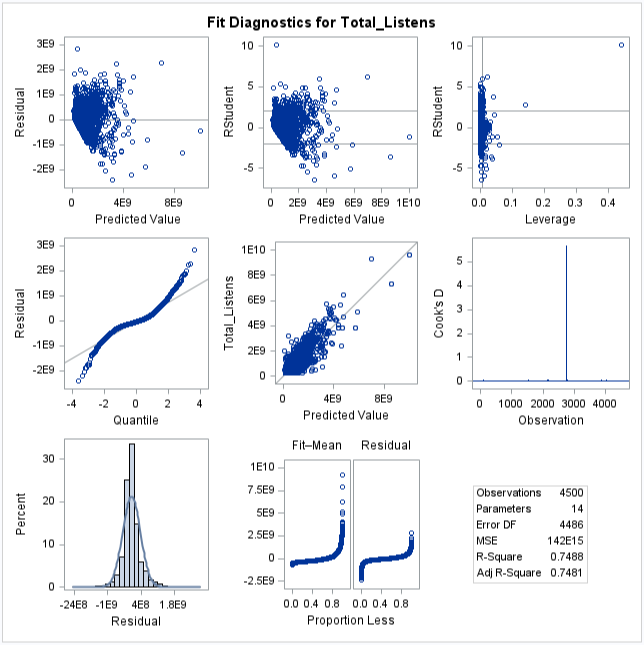


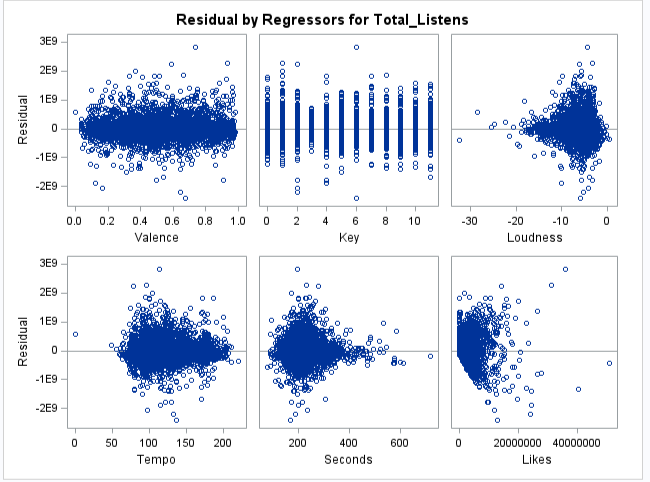
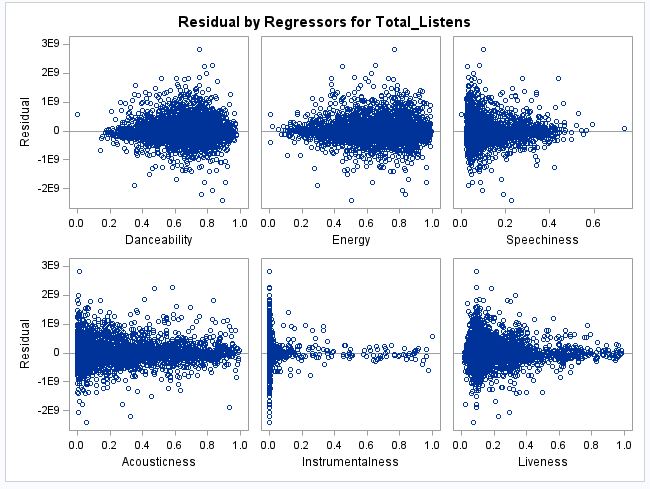
* From our means procedure we can see some things about the data, the average number of Streams, Views, and Total\_Listens are in the hundreds of millions, so we are looking at some of the most popular songs of all time. Also, the average song is in the middle in terms of Danceability, Energy, and Valence, and it is very low on the scales of Speechiness, Acousticness, Instrumentalness, and Liveness. The average song is in the key of F, probably because it is in the middle of the range. The average song is 3 minutes 47 seconds long and has a BPM of 121.45.
* Our Frequency procedure shows how many different artists we are actually working with. This data set includes many different artists from different genres, really going around the entire music circle. This can also be shown in our Key frequency chart as many genres sing in certain keys.



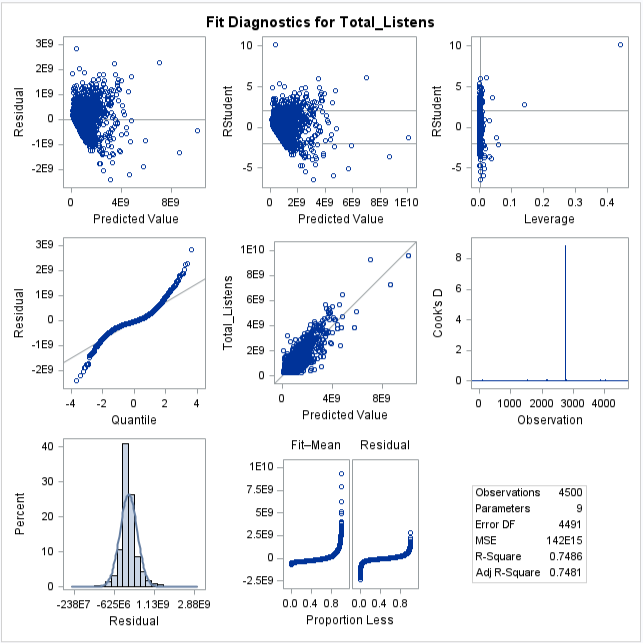
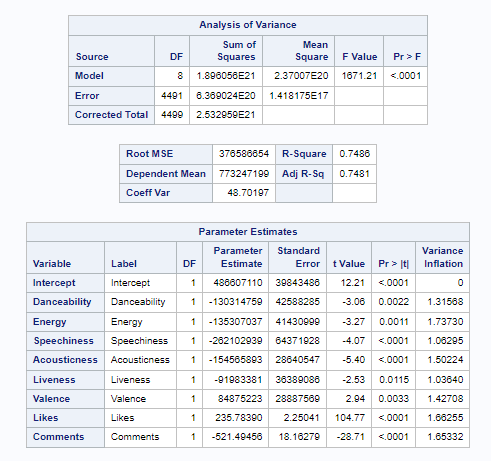


* At first glance we can see some positive information. Our R2 value is somewhat high for a data set this large which tells us that 74.88% of our data can be explained by the model. All except our variables: Instrumentalness, key, loudness, tempo, and seconds are significant in our model at a 0.05 significance level.

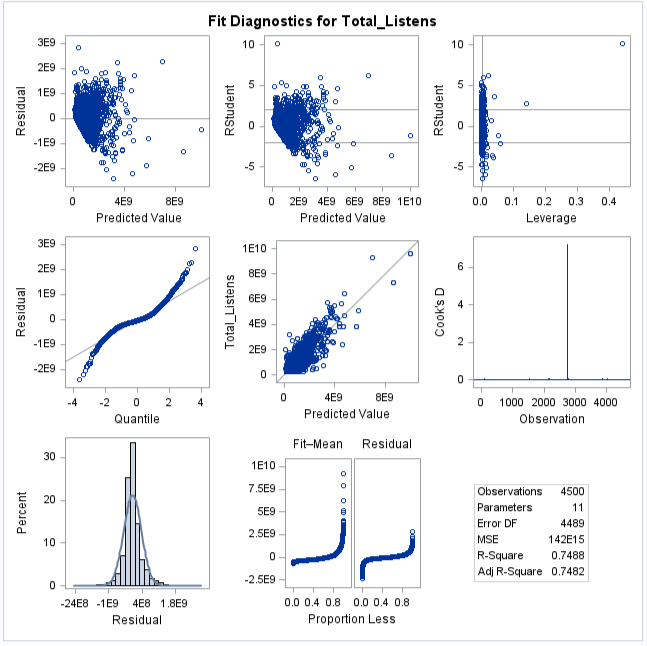
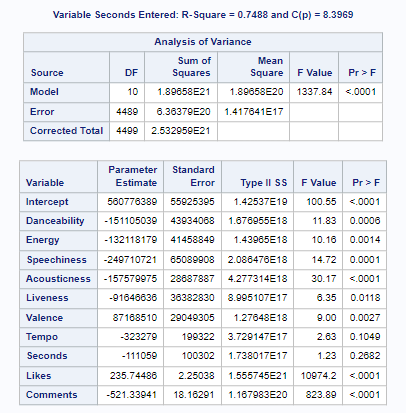




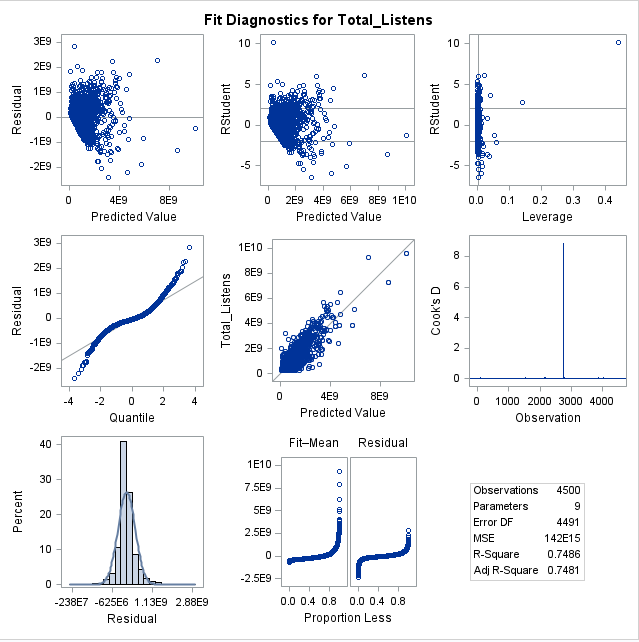
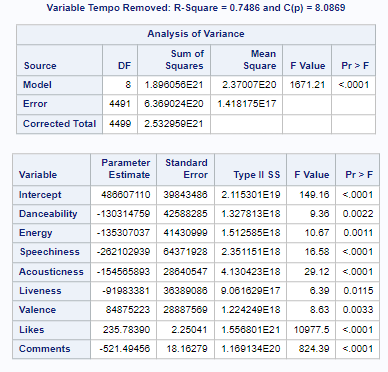
* The VIF values for all of our variables are small enough to not suggest any problems with multicollinearity. Looking at the Residual vs Regressor graphs, none of them look particularly non-linear which is good, but they aren’t without their quirks. Danceability, Energy, Liveness, Loudness, Tempo, and Seconds look Double Bow shaped to us, suggesting that they are proportions, which Danceability, Energy, and Liveness are, but the others aren’t. Speechiness, Acousticness, Instrumentalness, and Valence, which are proportions aren’t Double Bow shaped. Speechiness, Acousticness, and Instrumentalness are funnel shaped, possibly challenging the constant variance assumption. While Valence, Key, Likes, and Comments, despite Likes and Comments looking weird, are all regular or normal enough for us. While the Residual vs Predicted graph looks a little weird, if we were to remove some of the more extreme values on the right side of the graph, and just look at the majority of the values on the left, it would look pretty normal, and not raise any questions. Looking at the Normal plots, it is distinctly light-tailed. We tried a few transformations for this data, but none of them made the regression any better, it actually made it worse, so we decided to not use any form of transformation despite the fact that it might be necessary.
* Now we will try to look for the best model we can for this data, first by doing it our way by removing all of the variables deemed not significant by the individual t tests, and confirmed with the β confidence levels.



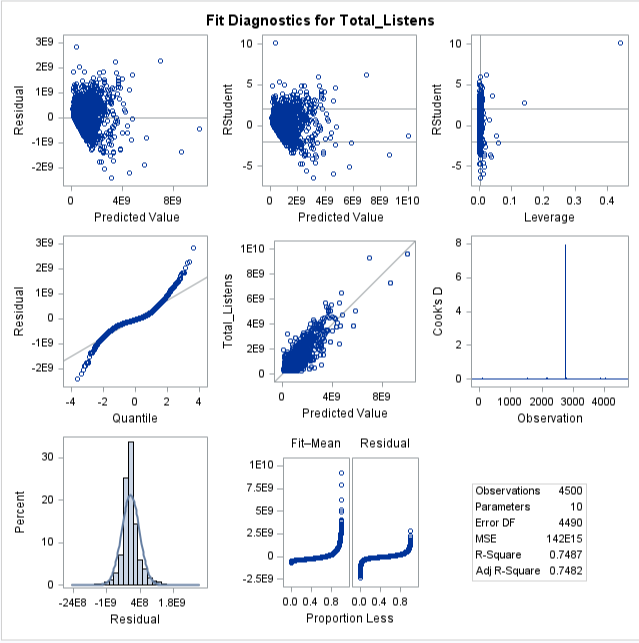
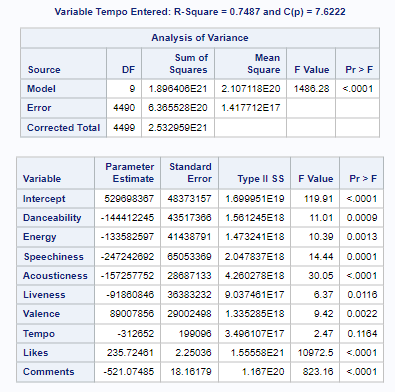
Next, using the Forward Selection technique – which adds variables one by one



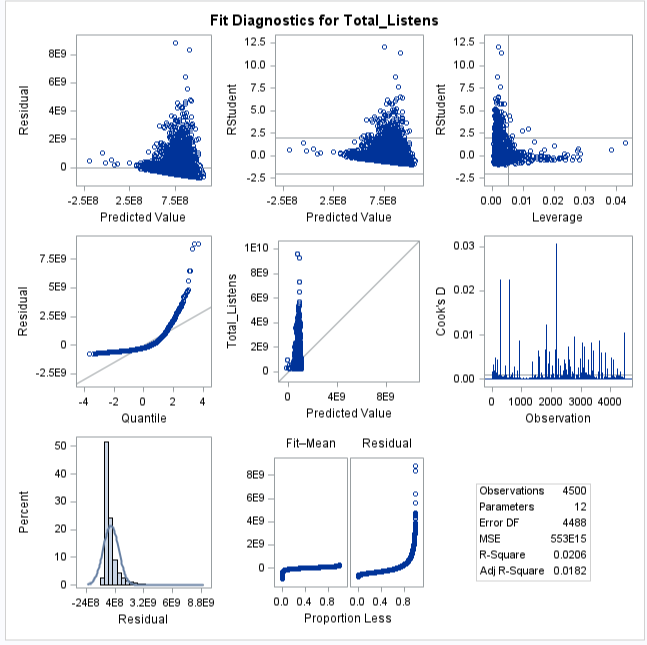
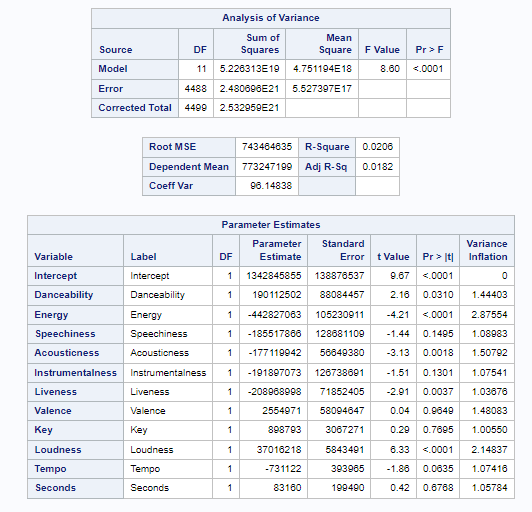
Next, using the Backward Elimination technique – which deletes variables one by one



Next, using the Stepwise technique – which is a combination of forward and backward elimination

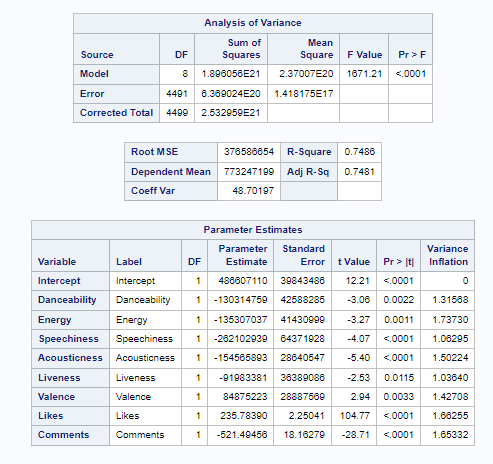


* And finally, just getting rid of two variables that we personally think are affecting the quality of the model despite being the most significant, Likes and Comments.



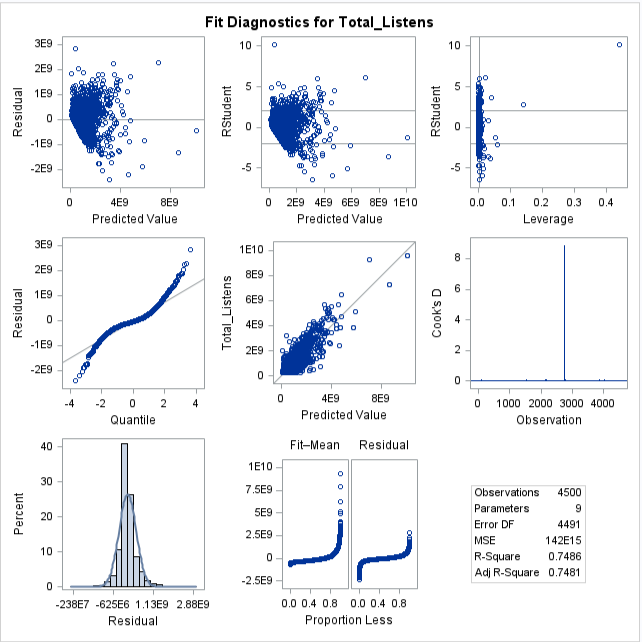
Clearly, likes and comments play a very, very important role in our model. Without the likes and comments statistics we don’t have much of a model at all. When these two variables are included, we are able to get a much better picture of our model as we are able to see what other variables can attribute to a song getting more likes and comments or vice versa.

* We decided to go with the model we created at first, and also the model that the backwards elimination technique gave us. I.E: the one where the variables Loudness, Key, Instrumentalness, Seconds, and Tempo were *removed*.



The model equation looks like Total\_Listens = 486,607,110 – 130,314,759 \* Danceability – 135,307,037 \* Energy – 262,102,939 \* Speechiness – 154,565,893 \* Acousticness – 91,983,381 \* Liveness + 84,875,223 \* Valence + 235.7839 \* Likes – 521.49456 \* Comments

Again, none of the VIF values would suggest any multicollinearity, the R2 values haven’t changed much from our original model, but are still pretty good, and while the MSRes value is really large, comparing it to the MSR value shows that it is around 200 times smaller.

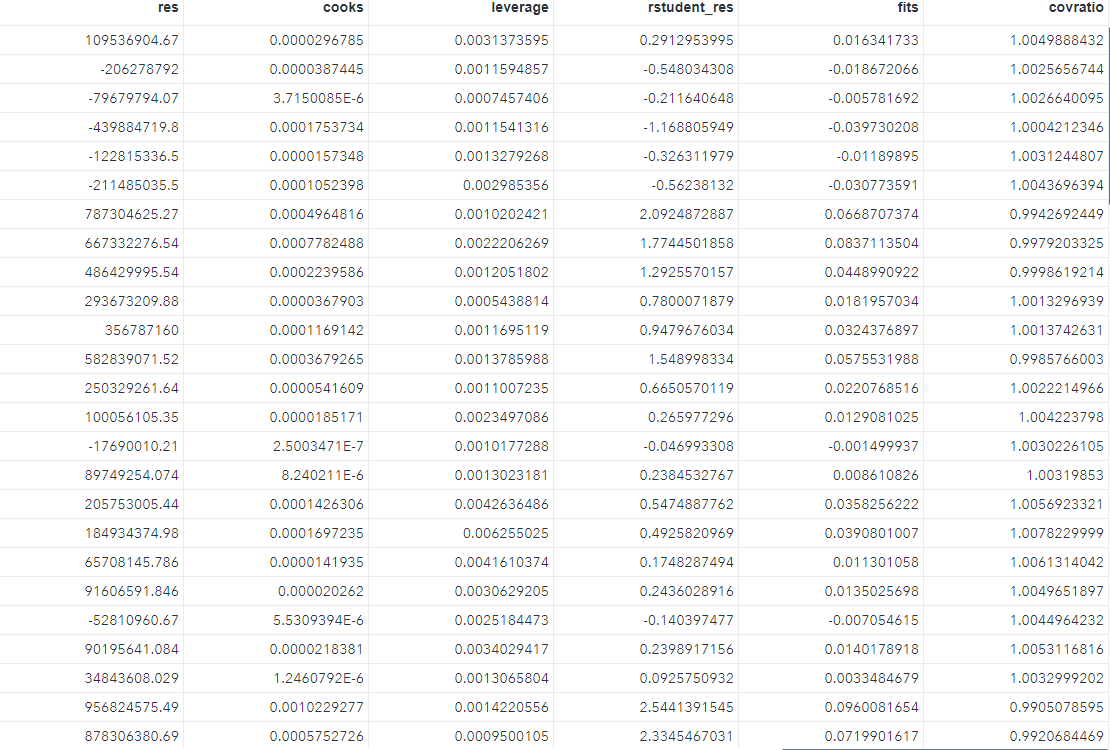


Our fit diagnostics tests look pretty similar to our original model as well. Another thing to mention is the pretty obvious influential point that we can see in both the Cook’s D and the RStudent vs Leverage graphs, which we will get to later.



This data is not in time order but assuming that it is, the RStudent vs Time Order graph looks pretty good, despite it being kind of hard to see what’s going on with this graph, it looks like a good mix of close and far away points for there to not be any correlation problem.

Now to look at the possible leverage and influential points.



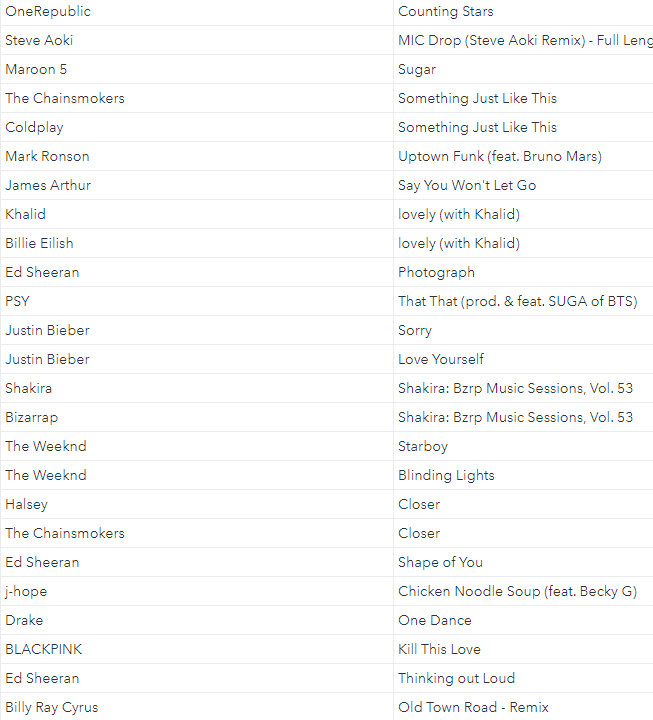
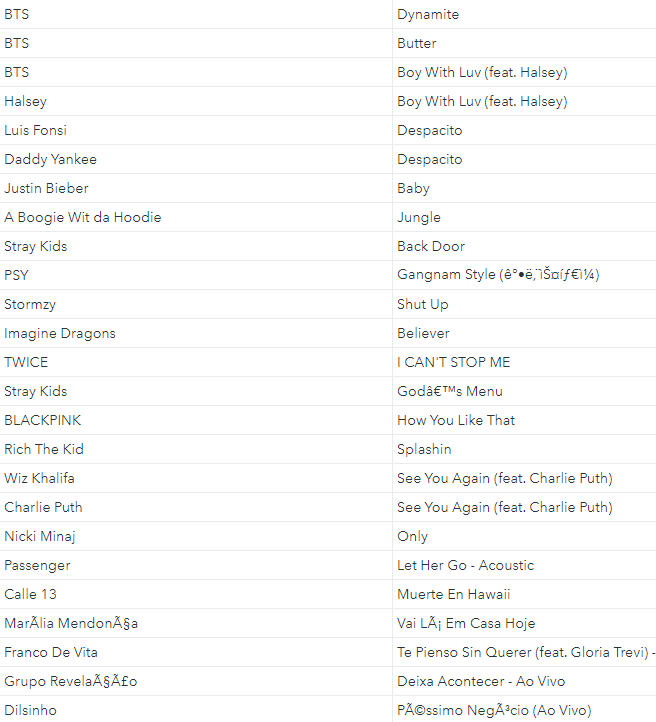
Sorting this by the leverage value, we can see that we have 246 points with a value over 2\*(9/4500) = .004. With one specifically standing out, being the one mentioned before, the one that also has a Cook’s D value of almost 9, that being observation 2753, aka the most Liked and Commented on video on all of YouTube, Dynamite by BTS.



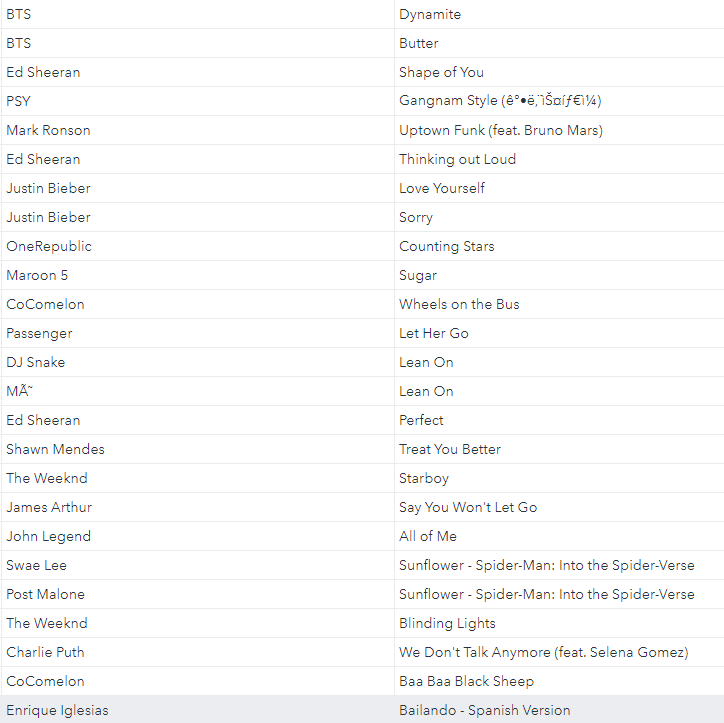
That song also has by far the highest Residual and RStudent values, with an RStudent value of 10.2, it is easily an influential point. Also, if we consider any point with an RStudent value over 3 as too high, observations 1548, 1594, 1797, 1829, 2154, 2156, 2244, 2315, 2620, 2622, 2771, 3045, 3361, 3631, 3697, 4452, and 4453 can also be considered influential points. We can’t really mention all of them by name for the purposes of space.

Sorting by Cook’s D, we see that Dynamite by BTS returns again, as the only point above 1, significantly at that, making it incredibly influential on β-hat. Since SAS doesn’t allow us to output DFBETA values, we can’t efficiently look at all the points that are influential on individual β-hat values. But since we have 1 very stand out value, we will look at that one and see how influential it is on individual β values. Using a cutoff of 3/sqrt(4500) = .0447, it is influential on every β-hat other than for Liveness, and the value is over 3 for both Likes and Comments.

Sorting by COVRATIO, we see that we have exactly 3900 values of our 45000 over 1, improving precision of the model, with the lowest being .926, which isn’t that far off from 1, but is still low enough that with 1-3\*(9/4500) = .994, it and many others are considered influential. The highest COVRATIO by a good bit is, you guessed it, held by Dynamite by BTS.



Finally, looking at DFFITS values, using a cutoff of 2\*sqrt(9/4500) = .0894, we have 155 observations that are influential on the fitted values, with the highest by far being Dynamite by BTS again.



If we were to use this model to predict future observations, which one might want to do if they were trying to make a really popular song, we can use the PRESS statistic, which is 659,179,770,000,000,000,000, and the SST value, which is 2,532,959,000,000,000,000,000, we can get the R2Prediction value. In this case 1-(PRESS/ SST) = .7398, which suggests that the model accounts for around 73.98% of the variability in predicting new observations, which is pretty good.

**Conclusion:**

It is safe to say that likes and comments account for the most listens for a song. The more people who hear it and like it, the more it will snowball. If you want to make a smash hit, just imitate BTS. In general, you probably want your song to have more music than vocals, which is interesting as we personally prefer the lyric side of music. Through our elimination techniques we found that we got rid of seconds or how long a song is. The fact that this variable was deemed insignificant is also interesting since songs range from 30 seconds to 9 minutes or some crazy range like that. Make your song energetic and danceable with high positivity, and you, my friend, have a song that people will listen to.

Code:

libname final450 "M:\My\_Private\_Files\stat450\final";

options validvarname=v7;

proc import datafile="M:\My\_Private\_Files\stat450\final\Spotify\_Youtube.xlsx"

out=final450.songs replace

dbms=xlsx;

run;

proc print data=final450.songs (obs=10);

run;

proc contents data=final450.songs;

run;

proc means data=final450.songs;

run;

proc freq data=final450.songs;

tables Artist Key / nocum;

run;

proc reg data=final450.songs;

model Total\_Listens = Danceability Energy Speechiness Acousticness Instrumentalness Liveness Valence Key Loudness Tempo Seconds Likes Comments/vif clb;

run;

proc reg data=final450.songs;

model Total\_Listens = Danceability Energy Speechiness Acousticness Liveness Valence Likes Comments / vif;

run;

proc reg data=final450.songs;

model Total\_Listens = Danceability Energy Speechiness Acousticness Instrumentalness Liveness Valence Key Loudness Tempo Seconds Likes Comments/selection=forward;

run;

proc reg data=final450.songs;

model Total\_Listens = Danceability Energy Speechiness Acousticness Instrumentalness Liveness Valence Key Loudness Tempo Seconds Likes Comments/selection=backward;

run;

proc reg data=final450.songs;

model Total\_Listens = Danceability Energy Speechiness Acousticness Instrumentalness Liveness Valence Key Loudness Tempo Seconds Likes Comments/selection=stepwise;

run;

proc reg data=final450.songs;

model Total\_Listens = Danceability Energy Speechiness Acousticness Instrumentalness Liveness Valence Key Loudness Tempo Seconds/vif;

run;

proc reg data=final450.songs outest=final450.press press;

model Total\_Listens = Danceability Energy Speechiness Acousticness Liveness Valence Likes Comments / vif;

plot rstudent.\*(obs.);

run;

proc reg data=final450.songs noprint;

model Total\_Listens = Danceability Energy Speechiness Acousticness Liveness Valence Likes Comments / influence press;

output out=final450.mainresiduals h=leverage residual=res rstudent=rstudent\_res cookd=cooks covratio=covratio dffits=fits;

run;

proc reg data=final450.songs;

model Total\_Listens = Danceability Energy Speechiness Acousticness Liveness Valence Likes Comments / influence;

run;

proc sort data=final450.mainresiduals;

by descending leverage;

run;

proc print data=final450.mainresiduals;

run;

proc sort data=final450.mainresiduals;

by descending rstudent\_res;

run;

proc print data=final450.mainresiduals;

run;

proc sort data=final450.mainresiduals;

by descending cooks;

run;

proc print data=final450.mainresiduals;

run;

proc sort data=final450.mainresiduals;

by descending covratio;

run;

proc print data=final450.mainresiduals;

run;

proc sort data=final450.mainresiduals;

by descending fits;

run;

proc print data=final450.mainresiduals;

run;

References:

Rastelli, S. (2023, March 20). *Spotify and YouTube*. Kaggle. Retrieved April 20, 2023, from <https://www.kaggle.com/datasets/salvatorerastelli/spotify-and-youtube>

Wikimedia Foundation. (2023, February 28). *Pitch class*. Wikipedia. Retrieved April 20, 2023, from <https://en.wikipedia.org/wiki/Pitch_class>