Modeling the Overall Energy in Music Tracks: Linear Regression Approach

JOEY GADBOIS
STAT 510: REGRESSION ANALYSIS

Introduction

The data for this project were scraped from Spotify's web application programming interface (API). The dataset consists of audio features of 694 tracks from various artist's across multiple genres. The variables with a short description are:

Variable	Data Type	Description
Energy (response)	Numeric	Perceptual measure of intensity and activity
Genre (1)	Categorical	Genre of music
Mode (2)	Categorical	Modality of track, 1 = major, 0 = minor
Tempo (1)	Numeric	Overall beats per minute (BPMs)
Danceability (2)	Numeric	Measures how suitable the track is for dancing
Loudness (3)	Numeric	Overall loudness of a track in decibels (dB)
Speechiness (4)	Numeric	Detects the presence of spoken works in a track
Acousticness (5)	Numeric	Measures the acoustic sound of a track
Liveness (6)	Numeric	Detects presence of an audience in the recording
Instrumentalness (7)	Numeric	Predicts presence of vocals in the track
Valence (8)	Numeric	Measure of musical positivity conveyed by a track
Duration (9)	Numeric	Length of track in milliseconds

The purpose of this study is to implement regression analysis techniques to find insights on what determines the overall energy in a track based on the overall audio features. Since the audio features are describing the entire track, linear regression is appropriate since there will be no dependence on time.

Research Questions

This project will be guided by two research questions that will be answered directly, along with a third research question for exploratory purposes of potential future research. These questions are:

1. How can linear regression be implemented in order to understand what determines the overall energy of a track in the simplest manner?
2. What variable has the largest effect on the overall energy in a track?
3. Disregarding simplicity, are there any interactions in the data that contribute to understanding the energy in a track? If so, are they intuitive?

Regression Method

In an attempt to find the desired insights from this data, a variety of regression methods can be used. To address the first research question, building a linear regression model using variable selection techniques can be used to find a starting point. The model will need to be assessed to ensure the proper assumptions are met before reliable insights can be drawn from it. If assumptions are not fulfilled, procedures such as model reduction, transformation, and influential point detection are options to aid in building a correctly specified model with proper assumptions. Answering the first research question would come from building a first-order linear regression model where all estimated coefficients are significant.

The second question follows from the first since it entails finding the variable that has the largest effect on energy. In attempt to finding this answer, a potential solution is to use the first-order model built and for each predictor in the model, take it out and conduct a general F-test against the first-order model. Since the model was built to have all significant predictors, the F-test will result in the variable being significant. Instead what can be observed is the increase in residual sum of squares to determine which of the predictors can explain the most variation in energy.

The third question can be addressed by starting at the initial starting model found from variable selection procedures. A screening can be done to determine which interaction is the most significant. Only the most significant interaction effect will be added, and the screening process will be repeated. This can be continued until there are no significant interaction terms in the scope of the model. Then the model can be reduced if necessary and assessed for linear regression model assumptions.

Regression Analysis, Results, and Interpretation

Before any research questions can be addressed, an exploratory look at the data should take place to learn about the data. From observing a scatterplot matrix, there are two variables that have notable correlations with the response, being x_3 = loudness and x_5 = acousticness. Those are the only two correlations where the magnitude is above 0.5 in the entirety of the data.

Procedures for Research Question 1:

To find an initial starting model, the variable selection procedure of choice is best subsets regression because it gives an optimal model option for each number of predictors in the data. It also provides various criteria for a decision to be made including R_a^2, Bayesian Information Criterion, and Mallows' C_p statistic. The MSE can also be calculated since it provides the residual sums of squares. The results are organized in a tibble for easier comparison of all available models. The selected subset is that of 8 predictors, since it provides the ideal results for R_a^2, MSE, and BIC. Mallows' C_p statistic is not too far from the number of predictors either. The resulting model is:

A tibble: 11 x 7

	nvars	R2	Ajd_R2	RSS	MSE	BIC	Cp
	<int>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	1	0.524	0.523	8.62	0.0125	-502.	409.
2	2	0.608	0.607	7.10	0.0103	-630.	217.
3	3	0.634	0.633	6.63	0.00960	-672.	158.
4	4	0.679	0.677	5.81	0.00844	-756.	56.6
5	5	0.698	0.696	5.47	0.00795	-792.	14.6
6	6	0.701	0.699	5.41	0.00788	-792.	9.87
7	7	0.703	0.700	5.38	0.00784	-790.	7.62
8	8	0.704	0.701	5.36	0.00782	-786.	6.92
9	9	0.704	0.701	5.36	0.00783	-780.	8.36
10	10	0.705	0.700	5.35	0.00784	-774.	10.1
11	11	0.705	0.700	5.35	0.00785	-768.	12

$$Y_i = \beta_0 + \beta_1 c_1 + \beta_2 x_1 + \beta_3 x_2 + \beta_4 x_3 + \beta_5 x_5 + \beta_6 x_6 + \beta_7 x_7 + \beta_8 x_8 + \epsilon$$

This will serve as a starting model for now, but linear regression model assumptions need to be met before it is sufficient.

Identify Potential Influential Points:

Before assessing model assumptions, influential points should be considered because they could potentially interfere with the model assumptions by heavily swaying predictions. Methods of identification will include leverages to account for predictor variables and studentized residuals to observe potential outliers in the response. An observation with a studentized residual where $|r_i| > 3$ will be considered an outlier and leverages will be considered at both twice and three times the mean. Since there are only a total of 29 potential influential points out of 694 observations, all will be dropped. After influential data points have been dropped, best subsets regression is repeated and the subset with 8 predictors was chosen again yielding the model:

```
Leverage > 3*mean : 14
Leverage > 2*mean : 28
Outliers          : 1
```

$$Y_i = \beta_0 + \beta_1 c_1 + \beta_2 x_1 + \beta_3 x_2 + \beta_4 x_3 + \beta_5 x_4 + \beta_6 x_5 + \beta_7 x_6 + \beta_8 x_8 + \epsilon$$

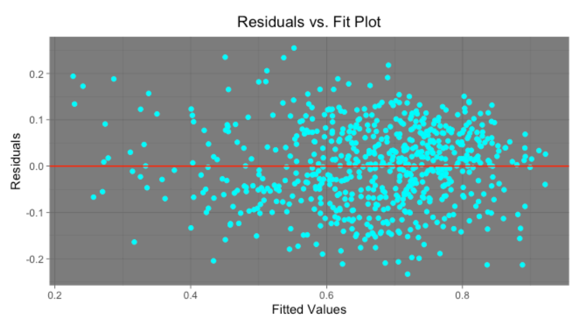
This model has an adjust coefficient of determination of $R_a^2 = 0.6828$, meaning that it can explain about 68.28% of the variation in energy. The summary table provides t-tests for the coefficient estimates for the hypotheses:

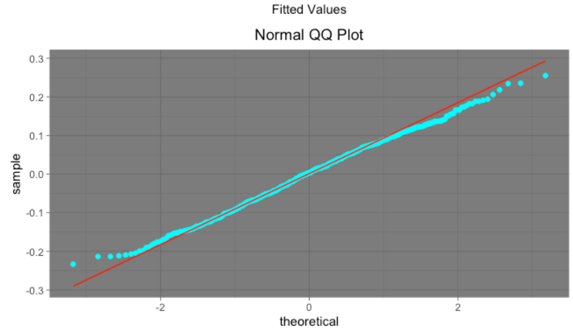
$$H_0: \beta_k = 0 \ \ vs. \ \ H_1: \beta_k \neq 0$$

Results deem the estimates for x_4 and x_6 being insignificant, meaning H_0 is accepted. The partial F-test is beneficial here, testing the hypothesis:

$$H_0: \beta_5 = \beta_7 = 0 \ \ vs. \ \ H_1: \beta_k \neq 0 \ (k = 5, 7)$$

The p-value from this test is $0.3606 > 0.05 = \alpha$ so the null hypothesis is not rejected and both variables are safely taken out of the model. The summary table of the reduced model indicates all numeric predictors are significant having a $R_a^2 = 0.6827$ thus, about 68.27% of the variation in energy can be explained by this model. Before this is deemed reliable, the assumptions must be assessed. The residuals vs. fit plot does not indicate any obvious issues with variance or linearity and the normal QQ plot seems to show a normal distribution for the residuals. This can be confirmed with the Shapiro-Wilk test, testing $H_0: e_i's \sim N(0, \sigma^2)$. The p-value is $0.1186 > 0.05 = \alpha$, so it can be concluded that the residuals are normally distributed, and the model is safe to use for inference and prediction.






```
Shapiro-Wilk Test:

Null: The variable is normally distributed
Alt : The variable is not normally distributed

        Shapiro-Wilk normality test

data:  augment(model)$.resid
W = 0.99627, p-value = 0.1186
```

Answering Research Question 1:

The first-order linear regression model with all significant predictors is the simplest way to try and understand what determines the energy in a track. This model is found to be:

$$Y_i = \beta_0 + \beta_1 c_1 + \beta_2 x_1 + \beta_3 x_2 + \beta_4 x_3 + \beta_5 x_5 + \beta_6 x_8 + \epsilon$$

```
95% Confidence Interval for Beta Coefficients:

# A tibble: 10 x 4
   Pred             Low        Est        Up
   <chr>           <dbl>      <dbl>      <dbl>
 1 (Intercept)  0.886      0.945      1.00
 2 c12         -0.0110     0.0107     0.0325
 3 c13         -0.00905    0.0157     0.0405
 4 c14         -0.0428    -0.0183     0.00622
 5 c15         -0.0407    -0.0187     0.00337
 6 x1           0.0000262  0.000275   0.000524
 7 x2          -0.239     -0.173     -0.107
 8 x3           0.0373     0.0410     0.0448
 9 x5          -0.276     -0.240     -0.205
10 x8           0.130      0.168      0.206
```

Observing the confidence intervals for the estimated coefficients, we can see that for all the numeric predictors the interval does not cover zero. This indicates that the estimates will be reliable and within these intervals 95% of the time. One thing to note is the genres, all corresponding intervals do include zero. This may be something to pay attention to for interaction terms.

Procedures for Research Question 2:

The proposed solution to determining which variable has the largest effect on the energy in a track is to conduct F-tests for a single parameter, but to observe the difference in residual sums of squares as a measure of the magnitude of effect. The difference in residual sums of squares for those that did not have the largest effect are shown in the table below:

c_1	x_1	x_2	x_5	x_8
0.0973	0.0358	0.2028	1.3354	0.5694

Answering the Question:

When x_3 is left out of the model, the difference in residual sums of squares is:

$$8.4306 - 4.9851 = 3.4455$$

```
Analysis of Variance Table

Model 1: y ~ c1 + x1 + x2 + x5 + x8
Model 2: y ~ c1 + x1 + x2 + x3 + x5 + x8
  Res.Df    RSS Df Sum of Sq      F    Pr(>F)
1    657 8.4306
2    656 4.9851  1    3.4455 453.41 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Thus, by a high margin, x_3 (loudness) is the variable that has the largest effect on energy.

Procedures for Research Question 3:

To answer the third research question using the proposed solution, this is a slightly rigorous task. The starting point is the initial starting model from best subsets regression after influential points were taken out. Recall the model:

$$Y_i = \beta_0 + \beta_1 c_1 + \beta_2 x_1 + \beta_3 x_2 + \beta_4 x_3 + \beta_5 x_4 + \beta_6 x_5 + \beta_7 x_6 + \beta_8 x_8 + \epsilon$$

```
Model:
y ~ c1 + x1 + x2 + x3 + x4 + x5 + x6 + x8
       Df Sum of Sq    RSS     AIC F value    Pr(>F)
<none>              4.9695 -3238.0
c1:x4   4  0.154410 4.8151 -3251.1  5.2110 0.0003891 ***
x1:x4   1  0.006418 4.9631 -3236.9  0.8444 0.3584896
x2:x4   1  0.062740 4.9068 -3244.5  8.3494 0.0039861 **
x3:x4   1  0.001218 4.9683 -3236.2  0.1601 0.6891963
x4:x5   1  0.159414 4.8101 -3257.8 21.6412 3.981e-06 ***
x4:x6   1  0.002491 4.9671 -3236.4  0.3275 0.5673561
x4:x8   1  0.087004 4.8825 -3247.8 11.6361 0.0006867 ***
```

Since there are so many interactions to observe, the set of them for x_4 will be displayed. The most significant interaction term is that of $x_4 x_5$, so that is the first update.

The model from above is updated and the process repeats.

After the process is repeated numerous times, the screening indicated no significant interactions after the model was updated a total of 8 times. The 8 interaction effects present in the final model in the order they were added include:

$$(x_4, x_5), (x_2, x_8), (x_2, x_3), (x_3, x_4), (x_5, x_8), (c_1, x_8), (x_4, x_8), (c_1, x_4)$$

The variables x_1 and x_6 were tested for significance for significance via the partial F-test. The model was reduced because they did not test to be significant. The final linear regression model with interaction effects has $R_a^2 = 0.7121$, meaning that it can explain about 71.21% of the variation in energy.

Residual Analysis:

The residuals vs. fit plot and the normal QQ plot both appear to be well-behaved. The Shapiro-Wilk test does not look needed, but to be safe, the test is done. The p-value $= 0.08209 > 0.05 = \alpha$ thus, the null hypothesis is accepted, and the residuals are normally distributed. The plots and testing output for this procedure can be found in the appendix, along with the summary table and confidence intervals for the final model.

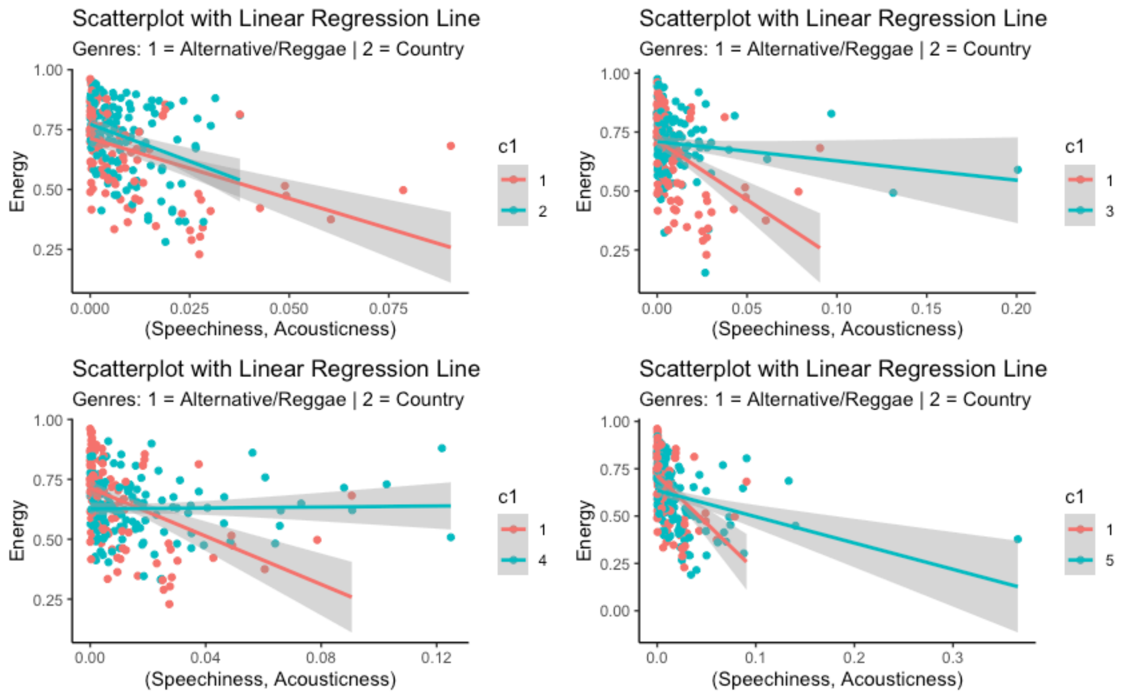
Notable Insights:

To address the different genres, the average values for each variable for each genre can be seen in the table to the right. For genres, the mapping is 1: Alternative/Reggae, 2: Country, 3: Electronic, 4: Hip Hop, 5: Pop. Country music has the highest average energy on by 0.025, which is a fair amount considering the range of the energy value. The predictors for country music are not notably extreme compared to that for other genres thus, the interactions may have a large effect. Pop music has the lowest average for energy with a value of 0.607. Hip hop has a value of 0.628 which is relatively close, but these two genres have a notably lower average energy that the other genres in the study.

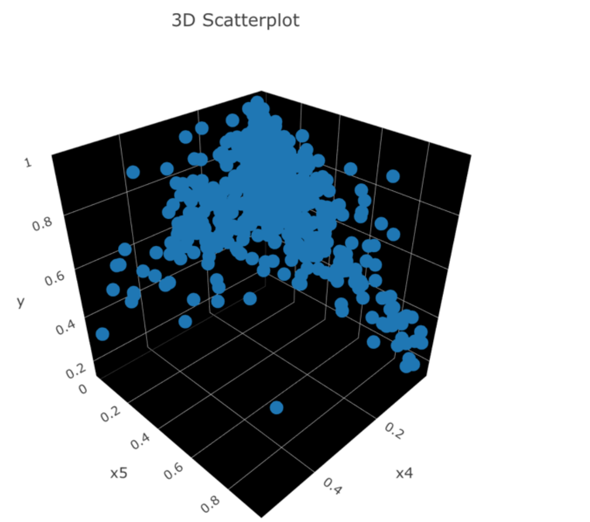
```
# A tibble: 5 x 9
  c1        n    mY   mPr   mX2   mX3    mX4   mX5   mX8
  <fct> <int> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
1 1       134 0.673 0.673 0.661 -6.37 0.0620 0.140 0.623
2 2       147 0.725 0.725 0.562 -5.23 0.0453 0.191 0.549
3 3       119 0.700 0.700 0.557 -5.26 0.0698 0.156 0.320
4 4       130 0.628 0.628 0.783 -6.13 0.196  0.101 0.465
5 5       136 0.607 0.607 0.660 -6.22 0.0797 0.257 0.471
```

To observe the two most significant interactions that are present in the model and how they differ by genres, the scatterplots for the interactions $x_4 x_5$ and $x_2 x_8$ are displayed below. The scatterplots with the linear regression trend lines are shown first, where alternative/reggae is shown in all four plots and the other genres are plotted against it for easier viewing. The first interaction being displayed is that of $x_4 x_5$ corresponding to speechiness and acousticness. It is evident that this interaction is significant in the model since the slope of the trend line is significantly different for all genres except for country. This is intuitive because the other three genres have sounds more related to sounds produced in a studio rather than sounds created from a band and multiple instruments for the most part.

Below the four scatterplots is a three-dimensional scatterplot of this interaction between speechiness and acousticness.



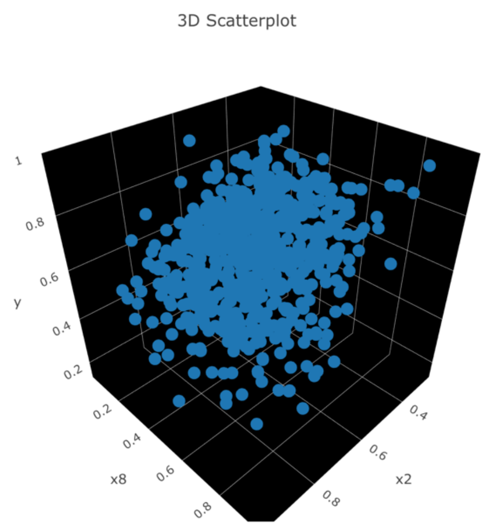
The three-dimensional scatterplot for x_4x_5, speechiness and acousticness, indicates that when observing all genres together, the highest values of energy come when speechiness and acousticness have lower values. This follows from the scatterplots for separate genres since all trend lines have a negative slope.



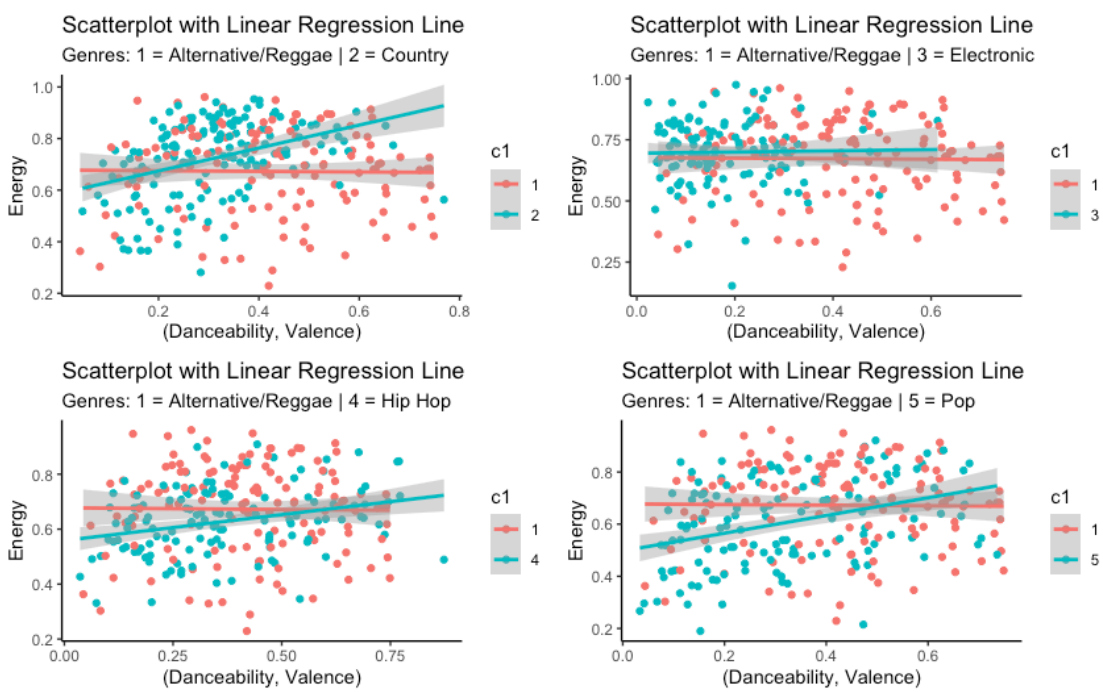
Observing the same plots for the interaction x_2x_8, corresponding to danceability and valence are displayed below. The three-dimensional scatterplot will be observed initially for this interaction.

The three-dimensional scatterplot indicates that for all genres. This plot does not give information as clearly as the one for the interaction x_4x_5. The largest energy values for the interaction x_2x_8 tend to result from lower values of x_2, where x_8 does not seem to have as significant of an effect. Although it is not as evident, the larger values of x_8 seem to result in larger energy values, but this is more dependent on x_2 having a lower value.

Visualizing this interaction on a two-dimensional plane may present insights in a more obvious manner.



Viewing x_2x_8 plotted against energy for each set of genres against alternative/reggae appears to be more insightful than the three-dimensional scatterplot. Below are the four plots, indicating that the slope of the trend lines is indeed different for almost all genres. The only genre that has a very similar slope of trendline to alternative/reggae is that of electronic dance music.



Answer to Research Question:

A simple answer to this question is yes, there are interactions in the data that help understand the energy in a track. This is also intuitive if you think about the interactions. For x_4x_5, this makes sense because music that has more acoustic sounds come more from bands that perform together. For x_2x_8 most people are likely more inclined to dance when they are in a positive environment. For x_2x_3, when it is thought of to go out dancing, the music will typically be louder. For x_3x_4, this is not as intuitive, this could potentially just be because if words are detected they have to be loud enough to be heard over the music. For x_4x_8 and x_5x_8, these kind of go hand in hand since it should be easier to convey a happy message with words, and lyrics tend to be more meaningful and present in general in acoustic music. The last two have to do with interactions with genres, meaning different genres will have different values for these features.

Thinking deeper than the question itself, note that all the interactions between two numeric variables have at least two interactions out of five variables interactions. The table below indicates that there could be some deeper more complex interactions in the data at a higher order or potentially hierarchical level.

(The numbers are the subscripts for the x_i)

2	3	4	5	8
3	2	3	4	2
8	4	5	8	4
		8		5

Notice how x_4 interacts with all others except for x_2, but they both interact with x_8. This likely indicates that there indeed more complexity to the interaction effects found between these variables.

Conclusion

The energy in music is a feature that has immense power, but why is that? This study was aimed at trying to understand what determines energy and how it varies. If it was possible to determine energy in music with a single feature, it would be the volume it is played at. The loudness was a huge factor in explaining energy, but it is far too complex to be described by one feature. It was found that the danceability, loudness, speechiness, acousticness, and valence all interacted with more than just one of each other to the point where they essentially all interact with each other, whether it be directly or indirectly. Further, the genre of music also plays a role in energy as well, because these features are a lot more or less likely to be as strong in some genres rather than others. Acousticness for example, would not likely be very strongly heard, if heard at all, in hip hop, pop, or electronic music.

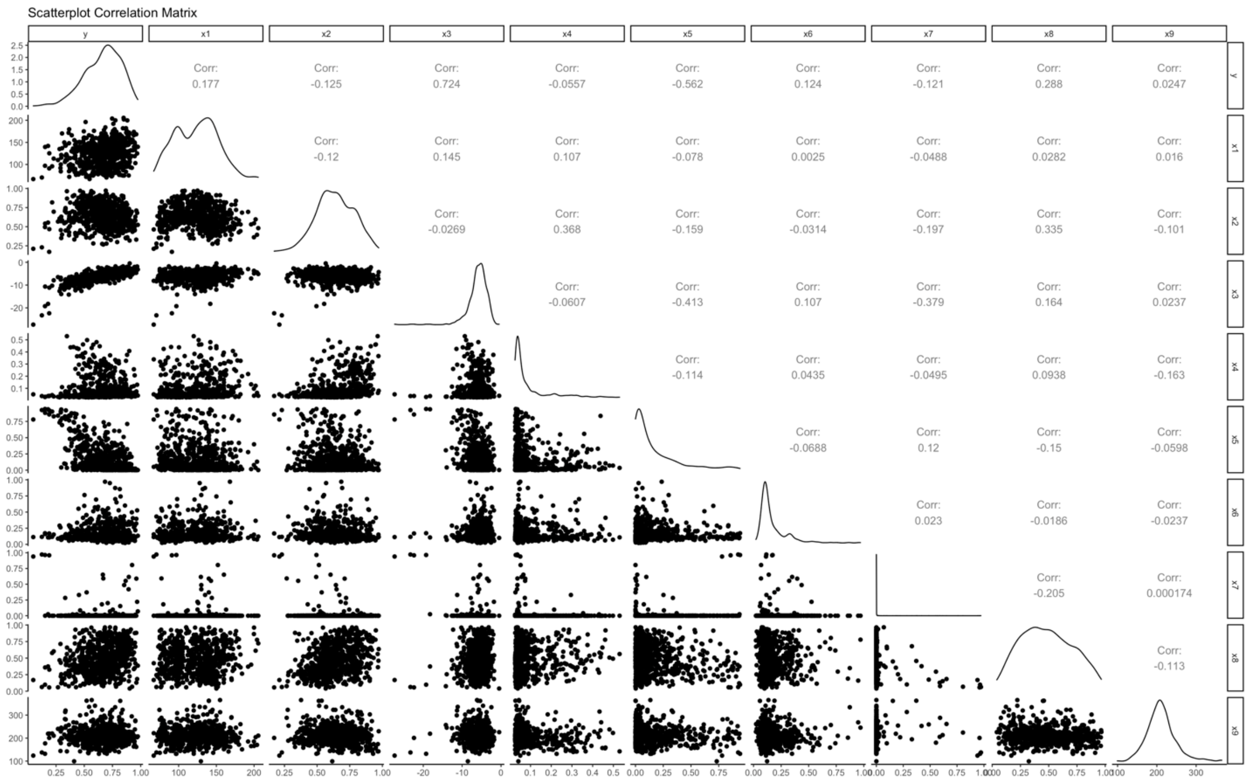
While there were many interesting observations made, it is very likely that there is far more exploration that can be done in regard to the energy in music. Interesting insights may come from exploring higher order interactions, or hierarchical relationships. Energy could potentially differ significantly when music is heard from an instrument rather than from an audio source.

Appendix

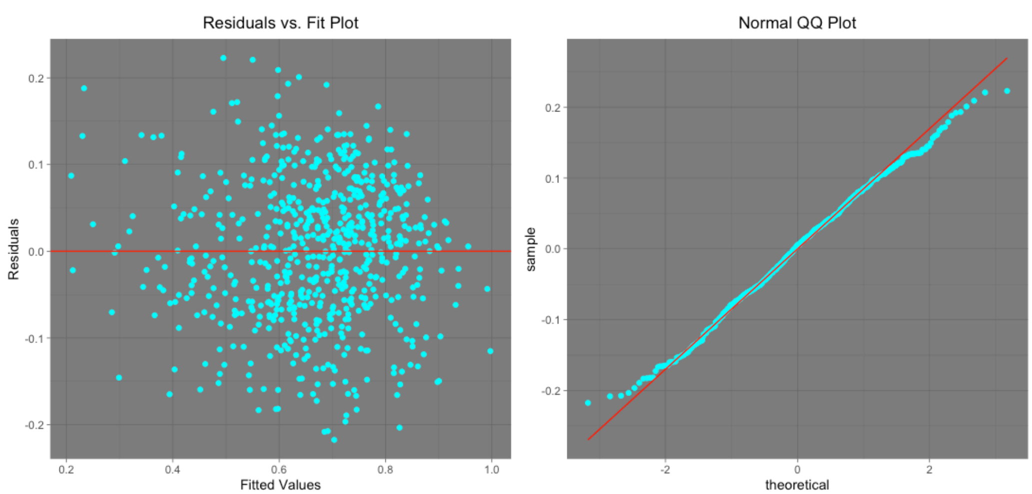
- **Section 1:** Extra Output
- **Section 2:** R Codes

Section 1:

Scatterplot Correlation Matrix



Final Model Residual Analysis:



```
Shapiro-Wilk Test:

Null: The variable is normally distributed
Alt : The variable is not normally distributed

        Shapiro-Wilk normality test

data:  augment(model)$.resid
W = 0.99594, p-value = 0.08209
```

Summary of Final Model

```
Coefficients:
             Estimate Std. Error t value Pr(>|t|)    
(Intercept)  0.958490   0.072737  13.177  < 2e-16 ***
c12         -0.017063   0.033618  -0.508 0.611943    
c13          0.078185   0.031427   2.488 0.013107 *  
c14         -0.053838   0.033864  -1.590 0.112361    
c15         -0.037936   0.030827  -1.231 0.218916    
x2          -0.251834   0.110477  -2.280 0.022964 *  
x3           0.066195   0.008478   7.808 2.38e-14 ***
x4           1.088381   0.259865   4.188 3.20e-05 ***
x5          -0.188550   0.042976  -4.387 1.34e-05 ***
x8           0.484150   0.098287   4.926 1.07e-06 ***
x4:x5        0.939777   0.240455   3.908 0.000103 ***
x2:x8       -0.422737   0.138702  -3.048 0.002400 ** 
x2:x3       -0.046883   0.013473  -3.480 0.000536 ***
x3:x4        0.055888   0.019081   2.929 0.003522 ** 
x5:x8       -0.229084   0.086078  -2.661 0.007978 ** 
c12:x8       0.075004   0.053275   1.408 0.159653    
c13:x8      -0.098761   0.067365  -1.466 0.143123    
c14:x8       0.174753   0.059094   2.957 0.003218 ** 
c15:x8       0.068685   0.050571   1.358 0.174883    
x4:x8       -0.748680   0.227716  -3.288 0.001065 ** 
c12:x4      -0.210310   0.269533  -0.780 0.435516    
c13:x4      -0.528088   0.208494  -2.533 0.011550 *  
c14:x4      -0.542693   0.169122  -3.209 0.001399 ** 
c15:x4      -0.326775   0.196197  -1.666 0.096292 .  
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.08304 on 642 degrees of freedom
Multiple R-squared:  0.7221,	Adjusted R-squared:  0.7121 
F-statistic: 72.52 on 23 and 642 DF,  p-value: < 2.2e-16
```

Confidence Intervals for β Estimates of Final Model

```
95% Confidence Interval for Beta Coefficients:

# A tibble: 24 x 4
   Pred           Low     Est      Up
   <chr>        <dbl>   <dbl>   <dbl>
 1 (Intercept)  0.816   0.958   1.10
 2 c12         -0.0831 -0.0171  0.0490
 3 c13          0.0165  0.0782  0.140
 4 c14         -0.120  -0.0538  0.0127
 5 c15         -0.0985 -0.0379  0.0226
 6 x2          -0.469  -0.252  -0.0349
 7 x3           0.0495  0.0662  0.0828
 8 x4           0.578   1.09    1.60
 9 x5          -0.273  -0.189  -0.104
10 x8           0.291   0.484   0.677
11 x4:x5        0.468   0.940   1.41
12 x2:x8       -0.695  -0.423  -0.150
13 x2:x3       -0.0733 -0.0469 -0.0204
14 x3:x4        0.0184  0.0559  0.0934
15 x5:x8       -0.398  -0.229  -0.0601
16 c12:x8      -0.0296  0.0750  0.180
17 c13:x8      -0.231  -0.0988  0.0335
18 c14:x8       0.0587  0.175   0.291
19 c15:x8      -0.0306  0.0687  0.168
20 x4:x8       -1.20   -0.749  -0.302
21 c12:x4      -0.740  -0.210   0.319
22 c13:x4      -0.938  -0.528  -0.119
23 c14:x4      -0.875  -0.543  -0.211
24 c15:x4      -0.712  -0.327   0.0585
```

Observing Data for Largest Residuals

Average values for all data:

Genres:
1: Alt/Reg | 2: Country
3: Electronic | 4: Hip Hop
5: Pop

```
# A tibble: 5 x 9
  c1        n    mY   mPr   mX2   mX3    mX4   mX5   mX8
  <fct> <int> <dbl> <dbl> <dbl> <dbl>  <dbl> <dbl> <dbl>
1 1       134 0.673 0.673 0.661 -6.37 0.0620 0.140 0.623
2 2       147 0.725 0.725 0.562 -5.23 0.0453 0.191 0.549
3 3       119 0.700 0.700 0.557 -5.26 0.0698 0.156 0.320
4 4       130 0.628 0.628 0.783 -6.13 0.196  0.101 0.465
5 5       136 0.607 0.607 0.660 -6.22 0.0797 0.257 0.471
```

Average values for data where the residuals are greater than 0.1:

```
# A tibble: 5 x 9
  c1        n    mY   mPr   mX2   mX3    mX4    mX5   mX8
  <fct> <int> <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl> <dbl>
1 1        11 0.756 0.623 0.715 -7.18 0.063  0.173  0.631
2 2        15 0.798 0.663 0.591 -5.74 0.0475 0.307  0.511
3 3        12 0.822 0.693 0.576 -4.99 0.0825 0.219  0.368
4 4        15 0.746 0.611 0.751 -6.89 0.186  0.0866 0.507
5 5        22 0.722 0.591 0.684 -6.61 0.0730 0.273  0.501
```

Average values for data where the residuals are greater than 0.15:

```
# A tibble: 5 x 9
  c1        n    mY   mPr   mX2   mX3    mX4    mX5   mX8
  <fct> <int> <dbl> <dbl> <dbl> <dbl>  <dbl>  <dbl> <dbl>
1 1         2 0.744 0.555 0.845 -7.55 0.126  0.416  0.657
2 2         3 0.821 0.619 0.629 -5.89 0.0548 0.439  0.505
3 3         2 0.823 0.654 0.568 -5.50 0.0378 0.304  0.322
4 4         3 0.723 0.558 0.877 -7.35 0.258  0.0451 0.529
5 5         4 0.712 0.519 0.566 -7.24 0.0711 0.434  0.408
```

Section 2:

```
#-------------------------------------------------------------------------------

# STAT 510 FINAL PROJECT

#-------------------------------------------------------------------------------

# starter functions
load_libraries = function(){
 # loads necessary libraries
 library(tidyverse)
 library(GGally)
 library(ggpubr)
 library(broom)
 library(leaps)
}
load_and_process_data = function(path){
 # given a path, loads data and preprocesses variables
 # this is specific to this dataset
 data = read_csv(path)
 data = data %>% select(genre:duration)
 df = data %>% mutate(y = energy,
            c1 = factor(genre), c2 = factor(mode),
            x1 = tempo, x2 = danceability, x3 = loudness,
            x4 = speechiness, x5 = acousticness,
            x6 = liveness, x7 = instrumentalness,
            x8 = valence, x9 = duration/1000)
 df = df %>% select(y:x9)
 df = df %>% mutate(c1 = if_else(c1=='Alternative/Reggae', '1',
                 if_else(c1=='Country', '2',
                    if_else(c1=='Electronic Dance', '3',
                       if_else(c1=='Hip Hop', '4', '5')))))
 df = df %>% mutate(c1 = factor(c1)) %>% print()
 return(df)
}
var_map = function(){
 # reference map for what each variable is
 cat('Response:  Y  = Energy\n\nCategorical Predictors:',
   '\n\nc1 = Genre   (1: Alt/Reg, 2: Country, 3: Electronic, 4: Hip Hop, 5: Pop)',
   '\nc2 = Mode\n\nNumeric Predictors:',
   '\n\nx1 = Tempo (BPMs)\nx2 = Danceability\nx3 = Loudness',
   '\nx4 = Speechiness\nx5 = Acousticness\nx6 = Liveness',
   '\nx7 = Instrumentalness\nx8 = Valence\nx9 = Duration (s)')
}

# variable selection functions
```

```
var_select_bestsubsets_comps = function(X, y, df){
  # performs best subsets regression
  # creates dataset to easily compare models
  # based on the values extracted
  model_subsets = regsubsets(X, y, nvmax = ncol(df))
  mods = summary(model_subsets)
  rs = mods$rss
  n = nrow(df)
  nsubsets = length(rs)
  mses = c()
  for (i in 1:nsubsets){
    mse = rs[i]/(n-i-1)
    mses = append(mses, mse)
  }
  mod_comps = tibble(nvars = 1:nsubsets, R2 = mods$rsq, Ajd_R2 = mods$adjr2,
                     RSS = rs, MSE = mses, BIC = mods$bic, Cp = mods$cp)
  mod_comps %>% print()
}
var_select_bestsubsets_predictors = function(X, y, df){
  # shows which variables are in model subsets
  nv = ncol(df)
  model_subsets = regsubsets(X, y, nvmax = nv)
  mods = summary(model_subsets)
  mods$which
}

# assessment functions
assess_res_fit = function(model){
  # Residuals vs. Fit Plot
  augment(model) %>% ggplot(., aes(x=.fitted, y=.resid)) +
    geom_point(color = 'cyan') + geom_hline(yintercept = 0, color = 'red') +
    labs(title = 'Residuals vs. Fit Plot', x = 'Fitted Values', y = 'Residuals') +
    theme_dark() + theme(plot.title = element_text(hjust = 0.5))
}
assess_norm_qq = function(model){
  # Normal QQ Plot
  augment(model) %>% ggplot(., aes(sample = .resid)) +
    stat_qq(color = 'cyan') + stat_qq_line(color = 'red') + ggtitle('Normal QQ Plot') +
    theme_dark() + theme(plot.title = element_text(hjust = 0.5))
}
assess_outliers_leverage = function(model, df){
  # finds leverages and outliers
  # returns dataframe with values to determine both
  df$hv = hatvalues(model)
  df$rs = rstandard(model)
  cat('Leverage > 3*mean :', length(which(df$hv > (3*sum(df$hv)/nrow(df)))),
      '\nLeverage > 2*mean :', length(which(df$hv > (2*sum(df$hv)/nrow(df)))),
      '\nOutliers          :', length(which(abs(df$rs) > 3)),
      '\n')
  return(df)
}

# testing and confidence intervals
test_LRT = function(alpha, full, reduced){
  # conducts likelihood ratio test to determine better fitting model
  k = length(full$coefficients) - length(reduced$coefficients)
  lr= -2*(logLik(reduced) - logLik(full))
  p = pchisq(lr, df = k, lower.tail = F)
  cat('\nLikelihood Ratio Test:',
      '\n\nNull: Smaller model has a better fit',
      '\nAlt : Larger model has a better fit',
```

```
    '\n\nTest Statistic:', lr, '\nDegrees of Freedom:', k)
  if (p < alpha){
    cat('\nThe p-value =', p, '<', alpha, '= alpha',
      '\n\nReject Null, choose larger model\n')
  }else{
    cat('\nThe p-value =', p, '>', alpha, '= alpha',
      '\n\nAccept Null, choose smaller model\n')
  }
}
test_shapiro = function(model){
  # Shapiro-Wilk Test for Normality
  cat('\nShapiro-Wilk Test:\n\nNull: The variable is normally distributed',
    '\nAlt : The variable is not normally distributed\n')
  shapiro.test(augment(model)$.resid)
}
inf_confints = function(model, level = 0.95){
  coeffs = as.data.frame(model$coefficients)
  ci = tibble(Pred = rownames(coeffs),
          Low = confint(model, level = level)[,1],
          Est = model$coefficients,
          Up = confint(model, level = level)[,2])
  cat('\n\nConfidence Interval for Beta Coefficients:\n\n')
  ci %>% print(n = Inf)
}

#-------------------------------------------------------------------------------

# FIRST STEPS

#-------------------------------------------------------------------------------

# load libraries and data
load_libraries()
data = load_and_process_data('audio_features.csv')
var_map()

# scatterplot correlation matrix
data %>%
  select(y, x1:x9) %>%
  ggpairs(title = 'Scatterplot Correlation Matrix', progress = F) +
  theme_classic()

# genre means
data %>%
  group_by(c1) %>%
  summarise(mY = mean(y),
        mX1 = mean(x1), mX2 = mean(x2), mX3 = mean(x3),
        mX4 = mean(x4), mX5 = mean(x5), mX6 = mean(x6),
        mX7 = mean(x7), mX8 = mean(x8), mX9 = mean(x9))

#-------------------------------------------------------------------------------

# BUILDING INITIAL MODEL

#-------------------------------------------------------------------------------

# best subsets regression
attach(data)
X = cbind(c1, c2, x1, x2, x3, x4, x5, x6, x7, x8, x9)
var_select_bestsubsets_comps(X, y, data)
var_select_bestsubsets_predictors(X, y, data)
```

```
detach(data)

# initial model selected from best subsets
fit = lm(y ~ c1 + x1 + x2 + x3 + x5 + x6 + x7 + x8, data = data)
summary(fit)

# inflential point detection | leverages and studentized residuals
df = assess_outliers_leverage(fit, data)
df = df %>% filter(hv < 2*sum(hv)/nrow(df), abs(rs) < 3)
df = df %>% select(y, c1, x1:x8) %>% print()

# repeat best subsets
attach(df)
X = cbind(c1, x1, x2, x3, x4, x5, x6, x7, x8)
var_select_bestsubsets_comps(X, y, df)
var_select_bestsubsets_predictors(X, y, df)
detach(df)

# initial model selected from repeated best subsets
fit = lm(y ~ c1 + x1 + x2 + x3 + x4 + x5 + x6 + x8, data = df)
summary(fit)

# residual analysis
assess_res_fit(fit)
assess_norm_qq(fit)
test_shapiro(fit)
assess_outliers_leverage(fit, df) %>% filter(hv > 3*sum(hv)/nrow(df))

# scatterplot of residuals vs. x7
df %>% ggplot(aes(x = x7, y = augment(fit)$.resid)) +
  geom_point(color = 'blue') + geom_hline(yintercept = 0, color = 'red') +
  labs(title='Residuals vs. x7', x='x7', y='Residuals') + theme_classic()

# confidence interval
inf_confints(fit, 0.95)

# reduce model
rfit = lm(y ~ c1 + x1 + x2 + x3 + x5 + x8, data = df)
summary(rfit)
anova(rfit)
test_LRT(0.05, fit, rfit)
anova(rfit, fit)

assess_res_fit(rfit)
assess_norm_qq(rfit)
test_shapiro(rfit)

#-------------------------------------------------------------------------------

# FIRST-ORDER MODEL

#-------------------------------------------------------------------------------

# model
model = lm(y ~ c1 + x1 + x2 + x3 + x5 + x8, data = df)
summary(model)
anova(model)

# residual analysis
p1 = assess_res_fit(model)
p2 = assess_norm_qq(model)
```

```
ggarrange(p1, p2, nrow = 1, ncol = 2)
test_shapiro(model)

# confidence intervals
inf_confints(model, 0.95)

# explore which data points have the largest errors and why
augment(model)
augment(model) %>% filter(.resid > 0.15)
augment(model) %>%
  group_by(c1) %>%
  summarise(n = n(), mY = mean(y), mPr = mean(.fitted),
        mX1 = mean(x1), mX2 = mean(x2), mX3 = mean(x3),
        mX4 = mean(x4), mX5 = mean(x5), mX8 = mean(x8))
augment(model) %>%
  filter(.resid > 0.1) %>%
  group_by(c1) %>%
  summarise(n = n(), mY = mean(y), mPred = mean(.fitted),
        mX1 = mean(x1), mX2 = mean(x2), mX3 = mean(x3),
        mX4 = mean(x4), mX5 = mean(x5), mX8 = mean(x8))
augment(model) %>%
  filter(.resid > 0.15) %>%
  group_by(c1) %>%
  summarise(n = n(), mY = mean(y), mPred = mean(.fitted),
        mX1 = mean(x1), mX2 = mean(x2), mX3 = mean(x3),
        mX4 = mean(x4), mX5 = mean(x5), mX8 = mean(x8))

#------------------------------------------------------------------------------

# F-tests for Finding Variable with Largest Effect

#------------------------------------------------------------------------------

# leave out c1
reg_form = y ~ x1 + x2 + x3 + x5 + x8
anova(lm(reg_form, data = df), model)

# leave out x1
reg_form = y ~ c1 + x2 + x3 + x5 + x8
anova(lm(reg_form, data = df), model)

# leave out x2
reg_form = y ~ c1 + x1 + x3 + x5 + x8
anova(lm(reg_form, data = df), model)

# leave out x3
reg_form = y ~ c1 + x1 + x2 + x5 + x8
anova(lm(reg_form, data = df), model)

# leave out x5
reg_form = y ~ c1 + x1 + x2 + x3 + x8
anova(lm(reg_form, data = df), model)

# leave out x8
reg_form = y ~ c1 + x1 + x2 + x3 + x5
anova(lm(reg_form, data = df), model)

#------------------------------------------------------------------------------

# INTERACTION EFFECTS
```

```
#-------------------------------------------------------------------------------

# list to hold models
ifit = list()

# reference
summary(fit)
var_map()

# screening interactions round 1
add1(fit, ~.+ c1*x1 + c1*x2 + c1*x3 + c1*x4 + c1*x5 + c1*x6 + c1*x8, test = 'F')
add1(fit, ~.+ x1*c1 + x1*x2 + x1*x3 + x1*x4 + x1*x5 + x1*x6 + x1*x8, test = 'F')
add1(fit, ~.+ x2*c1 + x2*x1 + x2*x3 + x2*x4 + x2*x5 + x2*x6 + x2*x8, test = 'F')
add1(fit, ~.+ x3*c1 + x3*x1 + x3*x2 + x3*x4 + x3*x5 + x3*x6 + x3*x8, test = 'F')
add1(fit, ~.+ x4*c1 + x4*x1 + x4*x2 + x4*x3 + x4*x5 + x4*x6 + x4*x8, test = 'F')
add1(fit, ~.+ x5*c1 + x5*x1 + x5*x2 + x5*x3 + x5*x4 + x5*x6 + x5*x8, test = 'F')
add1(fit, ~.+ x6*c1 + x6*x1 + x6*x2 + x6*x3 + x6*x4 + x6*x5 + x6*x8, test = 'F')
add1(fit, ~.+ x8*c1 + x8*x1 + x8*x2 + x8*x3 + x8*x4 + x8*x5 + x8*x6, test = 'F')

# update 1
ifit = append(ifit, list(update(fit, ~.+ x4*x5)))
summary(ifit[[1]])
anova(ifit[[1]])

# screening round 2
add1(ifit[[1]], ~.+ c1*x1 + c1*x2 + c1*x3 + c1*x4 + c1*x5 + c1*x6 + c1*x8, test = 'F')
add1(ifit[[1]], ~.+ x1*c1 + x1*x2 + x1*x3 + x1*x4 + x1*x5 + x1*x6 + x1*x8, test = 'F')
add1(ifit[[1]], ~.+ x2*c1 + x2*x1 + x2*x3 + x2*x4 + x2*x5 + x2*x6 + x2*x8, test = 'F')
add1(ifit[[1]], ~.+ x3*c1 + x3*x1 + x3*x2 + x3*x4 + x3*x5 + x3*x6 + x3*x8, test = 'F')
add1(ifit[[1]], ~.+ x4*c1 + x4*x1 + x4*x2 + x4*x3 + x4*x6 + x4*x8, test = 'F')
add1(ifit[[1]], ~.+ x5*c1 + x5*x1 + x5*x2 + x5*x3 + x5*x6 + x5*x8, test = 'F')
add1(ifit[[1]], ~.+ x6*c1 + x6*x1 + x6*x2 + x6*x3 + x6*x4 + x6*x5 + x6*x8, test = 'F')
add1(ifit[[1]], ~.+ x8*c1 + x8*x1 + x8*x2 + x8*x3 + x8*x4 + x8*x5 + x8*x6, test = 'F')

# update 2
ifit = append(ifit, list(update(ifit[[1]], ~.+ x2*x8)))
summary(ifit[[2]])
anova(ifit[[2]])

# screening round 3
add1(ifit[[2]], ~.+ c1*x1 + c1*x2 + c1*x3 + c1*x4 + c1*x5 + c1*x6 + c1*x8, test = 'F')
add1(ifit[[2]], ~.+ x1*c1 + x1*x2 + x1*x3 + x1*x4 + x1*x5 + x1*x6 + x1*x8, test = 'F')
add1(ifit[[2]], ~.+ x2*c1 + x2*x1 + x2*x3 + x2*x4 + x2*x5 + x2*x6, test = 'F')
add1(ifit[[2]], ~.+ x3*c1 + x3*x1 + x3*x2 + x3*x4 + x3*x5 + x3*x6 + x3*x8, test = 'F')
add1(ifit[[2]], ~.+ x4*c1 + x4*x1 + x4*x2 + x4*x3 + x4*x6 + x4*x8, test = 'F')
add1(ifit[[2]], ~.+ x5*c1 + x5*x1 + x5*x2 + x5*x3 + x5*x6 + x5*x8, test = 'F')
add1(ifit[[2]], ~.+ x6*c1 + x6*x1 + x6*x2 + x6*x3 + x6*x4 + x6*x5 + x6*x8, test = 'F')
add1(ifit[[2]], ~.+ x8*c1 + x8*x1 + x8*x3 + x8*x4 + x8*x5 + x8*x6, test = 'F')

# update 3
ifit = append(ifit, list(update(ifit[[2]], ~.+ x2*x3)))
summary(ifit[[3]])
anova(ifit[[3]])

# screening round 4
add1(ifit[[3]], ~.+ c1*x1 + c1*x2 + c1*x3 + c1*x4 + c1*x5 + c1*x6 + c1*x8, test = 'F')
add1(ifit[[3]], ~.+ x1*c1 + x1*x2 + x1*x3 + x1*x4 + x1*x5 + x1*x6 + x1*x8, test = 'F')
add1(ifit[[3]], ~.+ x2*c1 + x2*x1 + x2*x4 + x2*x5 + x2*x6, test = 'F')
add1(ifit[[3]], ~.+ x3*c1 + x3*x1 + x3*x4 + x3*x5 + x3*x6 + x3*x8, test = 'F')
add1(ifit[[3]], ~.+ x4*c1 + x4*x1 + x4*x2 + x4*x3 + x4*x6 + x4*x8, test = 'F')
add1(ifit[[3]], ~.+ x5*c1 + x5*x1 + x5*x2 + x5*x3 + x5*x6 + x5*x8, test = 'F')
```

```
add1(ifit[[3]], ~.+ x6*c1 + x6*x1 + x6*x2 + x6*x3 + x6*x4 + x6*x5 + x6*x8, test = 'F')
add1(ifit[[3]], ~.+ x8*c1 + x8*x1 + x8*x3 + x8*x4 + x8*x5 + x8*x6, test = 'F')

# update 4
ifit = append(ifit, list(update(ifit[[3]], ~.+ x3*x4)))
summary(ifit[[4]])
anova(ifit[[4]])

# screening round 5
add1(ifit[[4]], ~.+ c1*x1 + c1*x2 + c1*x3 + c1*x4 + c1*x5 + c1*x6 + c1*x8, test = 'F')
add1(ifit[[4]], ~.+ x1*c1 + x1*x2 + x1*x3 + x1*x4 + x1*x5 + x1*x6 + x1*x8, test = 'F')
add1(ifit[[4]], ~.+ x2*c1 + x2*x1 + x2*x4 + x2*x5 + x2*x6, test = 'F')
add1(ifit[[4]], ~.+ x3*c1 + x3*x1 + x3*x5 + x3*x6 + x3*x8, test = 'F')
add1(ifit[[4]], ~.+ x4*c1 + x4*x1 + x4*x2 + x4*x6 + x4*x8, test = 'F')
add1(ifit[[4]], ~.+ x5*c1 + x5*x1 + x5*x2 + x5*x3 + x5*x6 + x5*x8, test = 'F')
add1(ifit[[4]], ~.+ x6*c1 + x6*x1 + x6*x2 + x6*x3 + x6*x4 + x6*x5 + x6*x8, test = 'F')
add1(ifit[[4]], ~.+ x8*c1 + x8*x1 + x8*x3 + x8*x4 + x8*x5 + x8*x6, test = 'F')

# update 5
ifit = append(ifit, list(update(ifit[[4]], ~.+ x5*x8)))
summary(ifit[[5]])
anova(ifit[[5]])

# screening round 6
add1(ifit[[5]], ~.+ c1*x1 + c1*x2 + c1*x3 + c1*x4 + c1*x5 + c1*x6 + c1*x8, test = 'F')
add1(ifit[[5]], ~.+ x1*c1 + x1*x2 + x1*x3 + x1*x4 + x1*x5 + x1*x6 + x1*x8, test = 'F')
add1(ifit[[5]], ~.+ x2*c1 + x2*x1 + x2*x4 + x2*x5 + x2*x6, test = 'F')
add1(ifit[[5]], ~.+ x3*c1 + x3*x1 + x3*x5 + x3*x6 + x3*x8, test = 'F')
add1(ifit[[5]], ~.+ x4*c1 + x4*x1 + x4*x2 + x4*x6 + x4*x8, test = 'F')
add1(ifit[[5]], ~.+ x5*c1 + x5*x1 + x5*x2 + x5*x3 + x5*x6, test = 'F')
add1(ifit[[5]], ~.+ x6*c1 + x6*x1 + x6*x2 + x6*x3 + x6*x4 + x6*x5 + x6*x8, test = 'F')
add1(ifit[[5]], ~.+ x8*c1 + x8*x1 + x8*x3 + x8*x4 + x8*x6, test = 'F')

# update 6
ifit = append(ifit, list(update(ifit[[5]], ~.+ c1*x8)))
summary(ifit[[6]])
anova(ifit[[6]])

# screening round 7
add1(ifit[[6]], ~.+ c1*x1 + c1*x2 + c1*x3 + c1*x4 + c1*x5 + c1*x6, test = 'F')
add1(ifit[[6]], ~.+ x1*c1 + x1*x2 + x1*x3 + x1*x4 + x1*x5 + x1*x6 + x1*x8, test = 'F')
add1(ifit[[6]], ~.+ x2*c1 + x2*x1 + x2*x4 + x2*x5 + x2*x6, test = 'F')
add1(ifit[[6]], ~.+ x3*c1 + x3*x1 + x3*x5 + x3*x6 + x3*x8, test = 'F')
add1(ifit[[6]], ~.+ x4*c1 + x4*x1 + x4*x2 + x4*x6 + x4*x8, test = 'F')
add1(ifit[[6]], ~.+ x5*c1 + x5*x1 + x5*x2 + x5*x3 + x5*x6, test = 'F')
add1(ifit[[6]], ~.+ x6*c1 + x6*x1 + x6*x2 + x6*x3 + x6*x4 + x6*x5 + x6*x8, test = 'F')
add1(ifit[[6]], ~.+ x8*x1 + x8*x3 + x8*x4 + x8*x6, test = 'F')

# update 7
ifit = append(ifit, list(update(ifit[[6]], ~.+ x4*x8)))
summary(ifit[[7]])
anova(ifit[[7]])

# screening round 8
add1(ifit[[7]], ~.+ c1*x1 + c1*x2 + c1*x3 + c1*x4 + c1*x5 + c1*x6, test = 'F')
add1(ifit[[7]], ~.+ x1*c1 + x1*x2 + x1*x3 + x1*x4 + x1*x5 + x1*x6 + x1*x8, test = 'F')
add1(ifit[[7]], ~.+ x2*c1 + x2*x1 + x2*x4 + x2*x5 + x2*x6, test = 'F')
add1(ifit[[7]], ~.+ x3*c1 + x3*x1 + x3*x5 + x3*x6 + x3*x8, test = 'F')
add1(ifit[[7]], ~.+ x4*c1 + x4*x1 + x4*x2 + x4*x6, test = 'F')
add1(ifit[[7]], ~.+ x5*c1 + x5*x1 + x5*x2 + x5*x3 + x5*x6, test = 'F')
add1(ifit[[7]], ~.+ x6*c1 + x6*x1 + x6*x2 + x6*x3 + x6*x4 + x6*x5 + x6*x8, test = 'F')
```

```
add1(ifit[[7]], ~.+ x8*x1 + x8*x3 + x8*x6, test = 'F')

# update 8
ifit = append(ifit, list(update(ifit[[7]], ~.+ c1*x4)))
summary(ifit[[8]])
anova(ifit[[8]])

# screening round 9
add1(ifit[[8]], ~.+ c1*x1 + c1*x2 + c1*x3 + c1*x5 + c1*x6, test = 'F')
add1(ifit[[8]], ~.+ x1*c1 + x1*x2 + x1*x3 + x1*x4 + x1*x5 + x1*x6 + x1*x8, test = 'F')
add1(ifit[[8]], ~.+ x2*c1 + x2*x1 + x2*x4 + x2*x5 + x2*x6, test = 'F')
add1(ifit[[8]], ~.+ x3*c1 + x3*x1 + x3*x5 + x3*x6 + x3*x8, test = 'F')
add1(ifit[[8]], ~.+ x4*x1 + x4*x2 + x4*x6, test = 'F')
add1(ifit[[8]], ~.+ x5*c1 + x5*x1 + x5*x2 + x5*x3 + x5*x6, test = 'F')
add1(ifit[[8]], ~.+ x6*c1 + x6*x1 + x6*x2 + x6*x3 + x6*x4 + x6*x5 + x6*x8, test = 'F')
add1(ifit[[8]], ~.+ x8*x1 + x8*x3 + x8*x6, test = 'F')

# reduce model
rmod = lm(y ~ c1 + x2 + x3 + x4 + x5 + x8 + x4:x5 + x2:x8 + x2:x3 +
          x3:x4 + x5:x8 + c1:x8 + x4:x8 + c1:x4, data = df)
anova(rmod, ifit[[8]])
summary(rmod)

#-------------------------------------------------------------------------------

# FINAL MODEL WITH INTERACTION TERMS

#-------------------------------------------------------------------------------

# final model
model = lm(y ~ c1 + x2 + x3 + x4 + x5 + x8 + x4:x5 + x2:x8 + x2:x3 +
           x3:x4 + x5:x8 + c1:x8 + x4:x8 + c1:x4, data = df)
summary(model)

# residual analysis
p1 = assess_res_fit(model)
p2 = assess_norm_qq(model)
ggarrange(p1, p2, nrow = 1, ncol = 2)
test_shapiro(model)

df %>% ggplot(aes(x=x6, y=augment(model)$.resid)) +
  geom_point(color='blue') + geom_hline(yintercept = 0, color='red') +
  ggtitle('Residuals vs. x6') + labs(x='x6', y='Residuals') + theme_classic()

df %>% ggplot(aes(x=x6*x1, y=augment(model)$.resid)) +
  geom_point(color='blue') + geom_hline(yintercept = 0, color='red') +
  ggtitle('Residuals vs. x6*x1') + labs(x='x6*x1', y='Residuals') + theme_classic()

# confidence intervals
inf_confints(model, 0.95)

# comparing prediction vs confidence intervals
predict(model, interval = 'confidence')[1:10,]
predict(model, interval = 'prediction')[1:10,]

# inspecting errors
augment(model)
augment(model) %>% filter(.resid > 0.15)
augment(model) %>%
  group_by(c1) %>%
  summarise(n = n(), mY = mean(y), mPr = mean(.fitted),
```

```
        mX2 = mean(x2), mX3 = mean(x3), mX4 = mean(x4),
        mX5 = mean(x5), mX8 = mean(x8))
augment(model) %>%
 filter(.resid > 0.1) %>%
 group_by(c1) %>%
 summarise(n = n(), mY = mean(y), mPr = mean(.fitted),
        mX2 = mean(x2), mX3 = mean(x3), mX4 = mean(x4),
        mX5 = mean(x5), mX8 = mean(x8))
augment(model) %>%
 filter(.resid > 0.15) %>%
 group_by(c1) %>%
 summarise(n = n(), mY = mean(y), mPr = mean(.fitted),
        mX2 = mean(x2), mX3 = mean(x3), mX4 = mean(x4),
        mX5 = mean(x5), mX8 = mean(x8))

#-------------------------------------------------------------------------------

# INSIGHTS ABOUT GENRES

#-------------------------------------------------------------------------------

# average values for different genres
augment(model) %>%
 group_by(c1) %>%
 summarise(n = n(), mY = mean(y), mPr = mean(.fitted), mX2 = mean(x2),
        mX3 = mean(x3), mX4 = mean(x4), mX5 = mean(x5), mX8 = mean(x8))

# observing predictor values for highest and lowest predictions

# alternative/reggae
augment(model) %>% filter(c1 == 1) %>% arrange(desc(.fitted)) %>% print(n=20)
augment(model) %>% filter(c1 == 1) %>% arrange(.fitted) %>% print(n=20)

# country
augment(model) %>% filter(c1 == 2) %>% arrange(desc(.fitted)) %>% print(n=20)
augment(model) %>% filter(c1 == 2) %>% arrange(.fitted) %>% print(n=20)

# edm
augment(model) %>% filter(c1 == 3) %>% arrange(desc(.fitted)) %>% print(n=20)
augment(model) %>% filter(c1 == 3) %>% arrange(.fitted) %>% print(n=20)

# hip hop
augment(model) %>% filter(c1 == 4) %>% arrange(desc(.fitted)) %>% print(n=20)
augment(model) %>% filter(c1 == 4) %>% arrange(.fitted) %>% print(n=20)

# pop
augment(model) %>% filter(c1 == 5) %>% arrange(desc(.fitted)) %>% print(n=20)
augment(model) %>% filter(c1 == 5) %>% arrange(.fitted) %>% print(n=20)

#-------------------------------------------------------------------------------

# VISUALIZING INTERACTIONS

#-------------------------------------------------------------------------------

# 3D plotting interactions
library(plotly)

# to hold plots
iplots = list()
```

```
# plotting interaction x4x5 by genre
p12 = df %>% filter(c1 == 1 | c1 == 2) %>%
  ggplot(aes(x=x4*x5, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Speechiness, Acousticness)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
          subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p13 = df %>% filter(c1 == 1 | c1 == 3) %>%
  ggplot(aes(x=x4*x5, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Speechiness, Acousticness)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
          subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p14 = df %>% filter(c1 == 1 | c1 == 4) %>%
  ggplot(aes(x=x4*x5, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Speechiness, Acousticness)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
          subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p15 = df %>% filter(c1 == 1 | c1 == 5) %>%
  ggplot(aes(x=x4*x5, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Speechiness, Acousticness)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
          subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')

iplots = append(iplots, list(ggarrange(p12, p13, p14, p15, nrow = 2, ncol = 2)))
iplots[[1]]

# 3D Scatterplot for x4x5
axx = list(title='x4', showbackground=TRUE, backgroundcolor='black')
axy = list(title='x5', showbackground=TRUE, backgroundcolor='black')
axz = list(title='y', showbackground=TRUE, backgroundcolor='black')
plt3_45 = plot_ly(df, x=~x4, y=~x5, z=~y) %>% add_markers()
plt3_45 %>%
  layout(title = '3D Scatterplot',
         scene = list(xaxis=axx, yaxis=axy, zaxis=axz, aspectmode='cube'))

# plotting interaction x2x8 by genre
p12 = df %>% filter(c1 == 1 | c1 == 2) %>%
  ggplot(aes(x=x2*x8, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Danceability, Valence)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
          subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p13 = df %>% filter(c1 == 1 | c1 == 3) %>%
  ggplot(aes(x=x2*x8, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x = '(Danceability, Valence)', y = 'Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
          subtitle='Genres: 1 = Alternative/Reggae | 3 = Electronic')
p14 = df %>% filter(c1 == 1 | c1 == 4) %>%
  ggplot(aes(x=x2*x8, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x = '(Danceability, Valence)', y = 'Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
          subtitle='Genres: 1 = Alternative/Reggae | 4 = Hip Hop')
p15 = df %>% filter(c1 == 1 | c1 == 5) %>%
  ggplot(aes(x=x2*x8, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Danceability, Valence)', y='Energy') +
```

```
  ggtitle('Scatterplot with Linear Regression Line',
         subtitle='Genres: 1 = Alternative/Reggae | 5 = Pop')

iplots = append(iplots, list(ggarrange(p12, p13, p14, p15, nrow = 2, ncol = 2)))
iplots[[2]]

# 3D Scatterplot for x2x8
axx = list(title='x2', showbackground=TRUE, backgroundcolor='black')
axy = list(title='x8', showbackground=TRUE, backgroundcolor='black')
axz = list(title='y', showbackground=TRUE, backgroundcolor='black')
plt3_34 = plot_ly(df, x=~x2, y=~x8, z=~y) %>% add_markers()
plt3_34 %>%
  layout(title = '3D Scatterplot',
        scene = list(xaxis=axx, yaxis=axy, zaxis=axz, aspectmode='cube'))

# plotting interaction x2x3
p12 = df %>% filter(c1 == 1 | c1 == 2) %>%
  ggplot(aes(x=x2*x3, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Danceability, Loudness)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
         subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p13 = df %>% filter(c1 == 1 | c1 == 3) %>%
  ggplot(aes(x=x2*x3, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Danceability, Loudness)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
         subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p14 = df %>% filter(c1 == 1 | c1 == 4) %>%
  ggplot(aes(x=x2*x3, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Danceability, Loudness)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
         subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p15 = df %>% filter(c1 == 1 | c1 == 5) %>%
  ggplot(aes(x=x2*x3, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Danceability, Loudness)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
         subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')

iplots = append(iplots, list(ggarrange(p12, p13, p14, p15, nrow = 2, ncol = 2)))
iplots[[3]]

# 3D Scatterplot for x2x3
axx = list(title='x2', showbackground=TRUE, backgroundcolor='black')
axy = list(title='x3', showbackground=TRUE, backgroundcolor='black')
axz = list(title='y', showbackground=TRUE, backgroundcolor='black')
plt3_34 = plot_ly(df, x=~x2, y=~x3, z=~y) %>% add_markers()
plt3_34 %>%
  layout(title = '3D Scatterplot',
        scene = list(xaxis=axx, yaxis=axy, zaxis=axz, aspectmode='cube'))

# plotting interaction x3x4
p12 = df %>% filter(c1 == 1 | c1 == 2) %>%
  ggplot(aes(x=x3*x4, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Loudness, Speechiness)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
         subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p13 = df %>% filter(c1 == 1 | c1 == 3) %>%
```

```
  ggplot(aes(x=x4*x3, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Loudness, Speechiness)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
        subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p14 = df %>% filter(c1 == 1 | c1 == 4) %>%
  ggplot(aes(x=x4*x3, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Loudness, Speechiness)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
        subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p15 = df %>% filter(c1 == 1 | c1 == 5) %>%
  ggplot(aes(x=x4*x3, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Loudness, Speechiness)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
        subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')

iplots = append(iplots, list(ggarrange(p12, p13, p14, p15, nrow = 2, ncol = 2)))
iplots[[4]]

# 3D Scatterplot for x3x4
axx = list(title='x3', showbackground=TRUE, backgroundcolor='black')
axy = list(title='x4', showbackground=TRUE, backgroundcolor='black')
axz = list(title='y', showbackground=TRUE, backgroundcolor='black')
plt3_34 = plot_ly(df, x=~x3, y=~x4, z=~y) %>% add_markers()
plt3_34 %>%
  layout(title = '3D Scatterplot',
       scene = list(xaxis=axx, yaxis=axy, zaxis=axz, aspectmode='cube'))

# plotting interaction x5x8
p12 = df %>% filter(c1 == 1 | c1 == 2) %>%
  ggplot(aes(x=x5*x8, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Acousticness, Valence)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
        subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p13 = df %>% filter(c1 == 1 | c1 == 3) %>%
  ggplot(aes(x=x5*x8, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Acousticness, Valence)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
        subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p14 = df %>% filter(c1 == 1 | c1 == 4) %>%
  ggplot(aes(x=x5*x8, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Acousticness, Valence)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
        subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p15 = df %>% filter(c1 == 1 | c1 == 5) %>%
  ggplot(aes(x=x5*x8, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Acousticness, Valence)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
        subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')

iplots = append(iplots, list(ggarrange(p12, p13, p14, p15, nrow = 2, ncol = 2)))
iplots[[5]]

# 3D Scatterplot for x5x8
axx = list(title='x5', showbackground=TRUE, backgroundcolor='black')
```

```
axy = list(title='x8', showbackground=TRUE, backgroundcolor='black')
axz = list(title='y', showbackground=TRUE, backgroundcolor='black')
plt3_34 = plot_ly(df, x=~x5, y=~x8, z=~y) %>% add_markers()
plt3_34 %>%
  layout(title = '3D Scatterplot',
      scene = list(xaxis=axx, yaxis=axy, zaxis=axz, aspectmode='cube'))

# plotting interaction x4x8
p12 = df %>% filter(c1 == 1 | c1 == 2) %>%
  ggplot(aes(x=x4*x8, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Speechiness, Valence)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
       subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p13 = df %>% filter(c1 == 1 | c1 == 3) %>%
  ggplot(aes(x=x4*x8, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Speechiness, Valence)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
       subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p14 = df %>% filter(c1 == 1 | c1 == 4) %>%
  ggplot(aes(x=x4*x8, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Speechiness, Valence)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
       subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')
p15 = df %>% filter(c1 == 1 | c1 == 5) %>%
  ggplot(aes(x=x4*x8, y=y, color=c1)) +
  geom_point() + geom_smooth(method = 'lm') +
  theme_classic() + labs(x='(Speechiness, Valence)', y='Energy') +
  ggtitle('Scatterplot with Linear Regression Line',
       subtitle='Genres: 1 = Alternative/Reggae | 2 = Country')

iplots = append(iplots, list(ggarrange(p12, p13, p14, p15, nrow = 2, ncol = 2)))
iplots[[6]]

# 3D Scatterplot for x5x8
axx = list(title='x4', showbackground=TRUE, backgroundcolor='black')
axy = list(title='x8', showbackground=TRUE, backgroundcolor='black')
axz = list(title='y', showbackground=TRUE, backgroundcolor='black')
plt3_34 = plot_ly(df, x=~x4, y=~x8, z=~y) %>% add_markers()
plt3_34 %>%
  layout(title = '3D Scatterplot',
      scene = list(xaxis=axx, yaxis=axy, zaxis=axz, aspectmode='cube'))

#------------------------------------------------------------------
```