Effect of Energy, Danceability, Duration, and Speechiness on the Positivity of Billboard's Popular Songs from 2010 to 2019

Data analysis project for STAT 334, Spring 2022

Anagha Sikha, Sarah Ellwein, Joshua Blank, and Amanda Belden

Acknowledgements

This dataset was created by Leonardo Henrique and posted on the website Kaggle. The data comes from top songs on Billboard charts and was extracted using the website Organize Your Music.

Dedication

We would like to dedicate this analysis to Adele, for bringing down the mood.

Contents

1 Introduction

We wanted to study the trends that appear in popular music in the world. Everyone has various music tastes, but what is common between the songs that are seen as popular and are high on Billboard charts and win awards at the Grammys. We found a data set that measured the most popular songs in the world from the Billboard charts from the years of 2010 to 2019 and used Spotify and Organize Your Music website to measure the different attributes of a song including genre, year, beats per minute (BPM), energy, danceability, loudness (dB), liveness, valence, length, acousticness, speechiness, popularity, and duration. While listening to songs, we don't tend to think of these attributes, but all of these factors play a role in what makes songs enjoyable and well-known. We found it interesting to see how valence, a measure of mood, related to these other factors in order to find out what types of moods are found in the most popular songs from the past decade. Through this project we can objectively see what makes up the moods in the most popular songs, which is often seen as a subjective view. We believe this will interest readers to observe quantitative and categorical aspects of a topic like music.

Some parties that may benefit from the data and the results from the data analysis are Spotify users (or other music streaming service users) and song producers. In order to formulate playlists, streaming users could look at the data to see which songs elicit specific qualities which match the type of playlist that they want to make. For example, if someone wants to make a happy playlist, they could look for songs with high valence and high bpm. Song producers could also use the data when making decisions for creating new songs. For example, if the producer wants to write a happy song, they could see which qualities (bpm, dB, danceability, etc.) are associated with high valence and incorporate those qualities accordingly. The data only contains songs that charted in the billboard charts for their respective years. When looking through these songs, most are categorized as pop (or some variation of the pop genre). Therefore, it may not be appropriate to apply the results from the analysis to songs outside of the pop genre. There also may be a strong bias in the data towards American music and consumption as most of the songs were created and made popular in the United States.

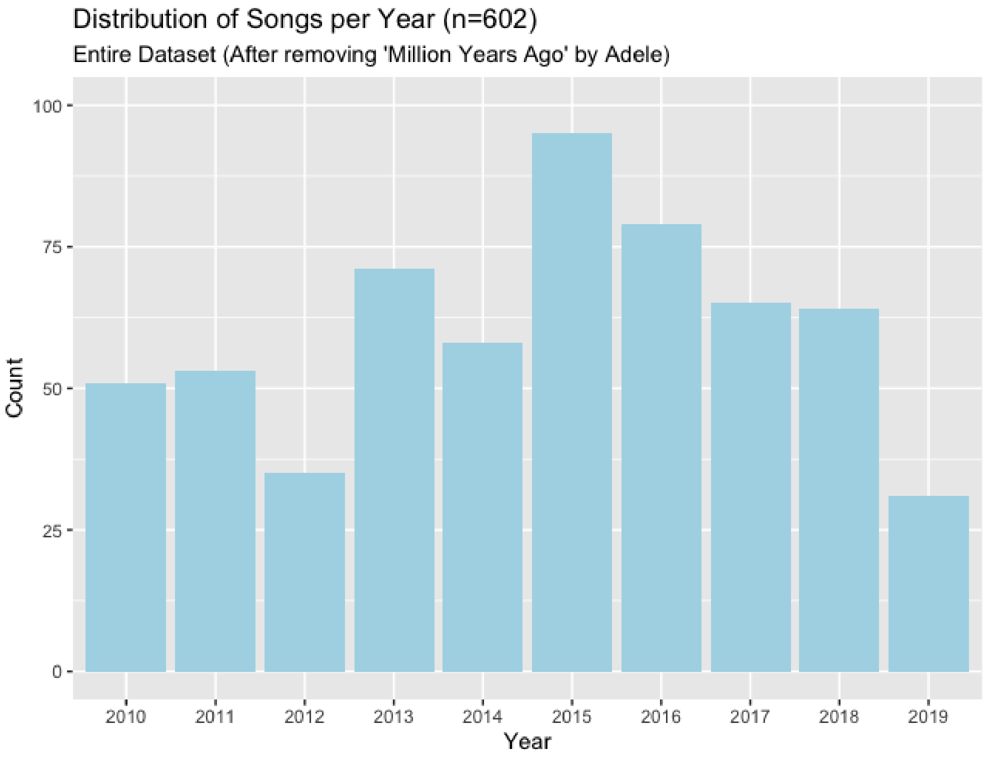
2 Materials and Methods

We selected the data from Leonardo Henrique's "Top Spotify songs from 2010-2019 - BY YEAR" dataset. Henrique gathered data from the most popular songs from Billboard charts from 2010-2019. Henrique has not specified which type of Billboard charts the data was collected from. We assume he selected data from the Billboard's Hot 100 charts, as multiple songs match with the songs on the Hot 100 charts from 2010 to 2019. The observational units we used were popular songs from each year. The primary variables of interest were genre, year, beats per minute (BPM), energy, danceability, loudness (dB), liveness, length, acousticness, speechiness, and duration. Our response variable was valence which measures the mood for the song, with higher values showing a more positive mood and lower values showing a more negative mood. These values were calculated by Spotify and then compiled into a dataset using the Organize Your Music website. To gather the data similarly to how Henrique did, we can go on the Billboard Year End charts Hot 100 songs website, and go to each year (2010 to 2019) on the dropdown menu to get the songs. Then we can create a Spotify playlist for these songs by searching the name and adding each of these songs to the playlist. On the Organize Your Music website, select "a specific playlist" and paste the URL of the spotify playlist. The website will then produce plots for each of these variables. Since we are selecting the number of songs from each chart and we are selecting which charts to use, there was no random sampling or random assignment.

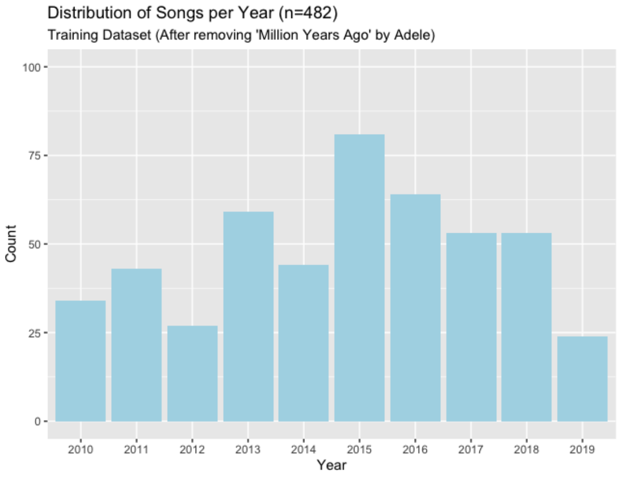
We made one edit to this dataset before beginning our formal investigations. The song "Million Years Ago " by Adele (observation 443) was an unusual observation due to its influence in all of the matrix scatter plots; therefore, we decided to omit this song from the dataset. We believe that this observation was entered in error as all but one of its values were zero.

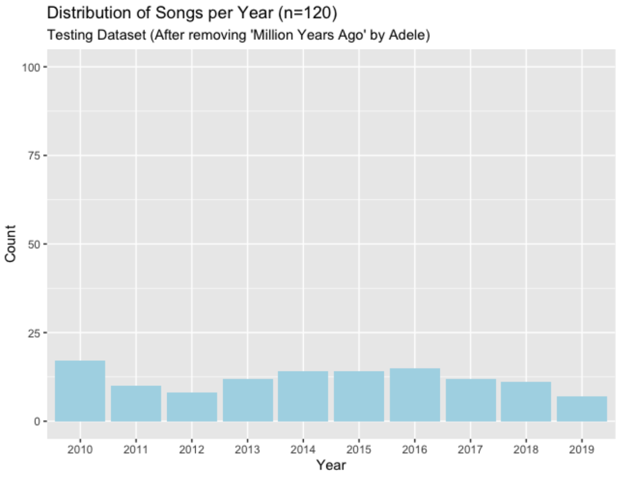
3 Splitting the Data

In our original dataset, we have a total 602 observations from years 2010 to 2019. We further explore the distribution of songs for each year.



Since there are 602 observations from the entire data set, we took 80% (482 songs) of the observations to be used for the training data and 20% (120 songs) to be used for the test data. In order to do this, we set a seed (seed number = 23173) in R to randomly select 482 out of the 602 observations for the training data. We then used the remaining 120 songs for the test data. The training data set is then used to form and compare linear models (e.g., linear regression, quadratic regression, etc.) and the parameter estimates from those models. We will apply the test data to our model and find the mean squared error between the predicted valence and the test data's actual valence. The testing error will evaluate our model's performance in predicting valence.

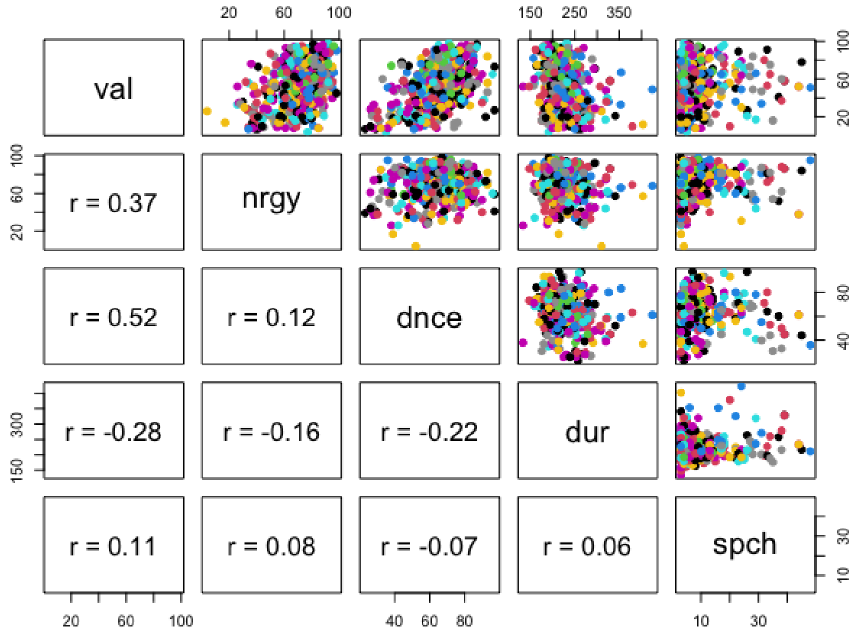




4 Data Visualization

4.1 Correlation Matrix

We chose to include the five of the most important/interesting quantitative variables in the matrix scatterplot and correlation matrix: valence, energy, danceability, duration, and speechiness. We also include a colored matrix scatterplot such that each color represents a year.

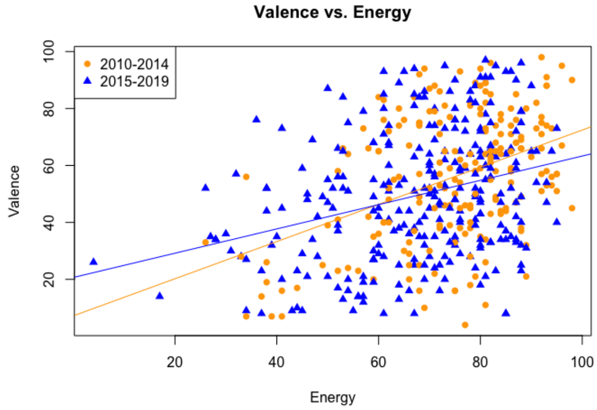


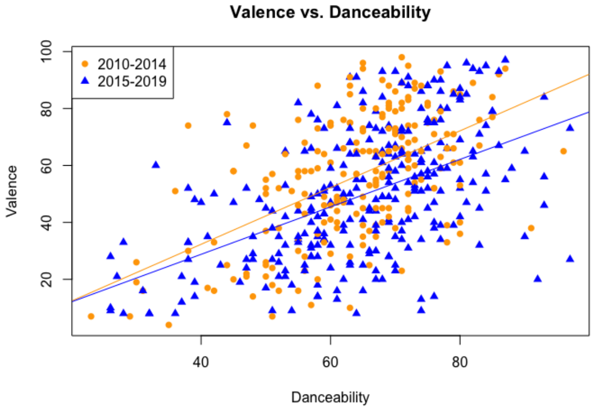
From the matrix scatterplot, it appears energy and danceability are the two predictors that are most strongly associated with the response variable, valence. They both seem to have positive, moderate linear associations with valence. Solely based on the scatterplots, it does not seem as though we will need transformations for these predictors. It appears as though danceability and energy may have a weak to moderate correlation and duration and danceability may also have a weak correlation. This matrix scatterplot suggests that as the energy and danceability of a song increases, the positivity of the song also tends to increase, which is expected.

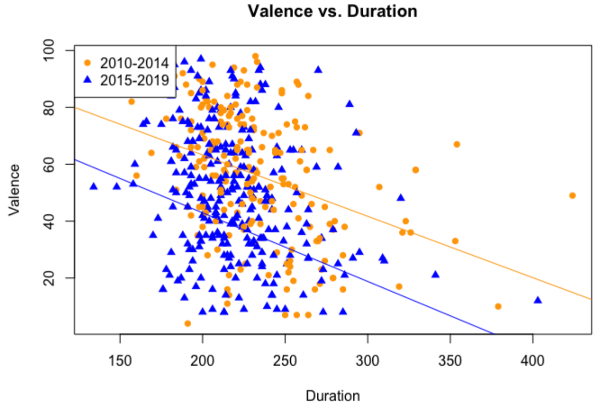
There is one unusual observation: the song “Start” by John Legend. While this observation appears to be a potential outlier for its energy value, it does not appear to be concerning.

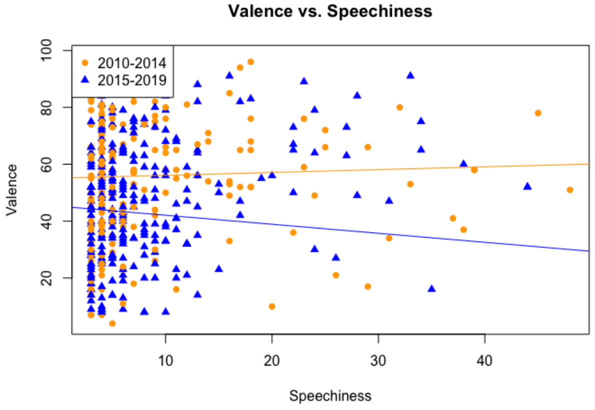
4.2 Interaction Plots

We explored four interaction plots comparing our categorical variable (years) by other quantitative variables (energy, danceability, duration, and speechiness).









There doesn't appear to be any interactions between years and other predictors. The only one that may seem of interest is the interaction between years and energy: this interaction show that as year increases, the effect of energy decreases. We suspect that the interaction between energy and years is not very significant as the difference in the slopes of the 2010-2014 and 2015-2019 lines are not very large.

5 Variable Prepocessing

5.1 Boxcox Procedure

We introduce one categorical variable called year. In the original dataset, year contained all the years that each song was released in. To make this a categorical feature, we split the year into two halves of a decade: 2010-2014 and 2015-2019.

In the beginning, we created a model with speechiness, danceability, BPM, energy, duration, genre, years, and decibels. We observed that the residual vs fitted plot for this model had unequal variance (Figure 2 left). Therefore we ran the Box-Cox procedure to see what power transformation was needed for valence. This procedure suggested to decrease the power of valence to around 0.8 (Figure 1).

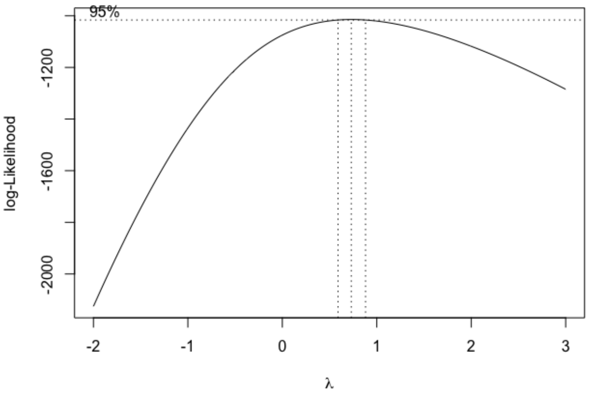


Figure 1: Boxcox Procedure

After running a new model with the power change in valence, we did not perceive a large change in the transformed residual plot (Figure 2 right). Since there was no large improvement from the transformation, we concluded it would be best to leave valence untransformed.

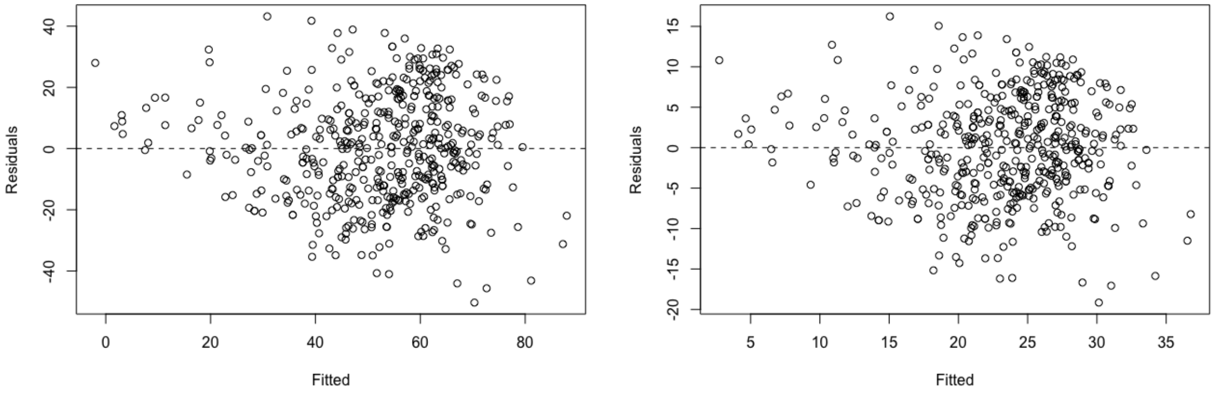


Figure 2: Residual plots for full model regression. Left: before transformation. Right: after transformation

5.2 Polynomial Regression

After looking at the matrix scatterplot, it was apparent that valence and speechiness had an unusual relationship; therefore, we decided to look at the residual scatterplot for valence vs speechiness. We observed some heteroscedasticity in the residual plots (Figure 3 left). We applied a polynomial regression to improve equal variance by modeling valence on speechiness and speechiness to the power of 1/2 and centering the terms to prevent multicollinearity. This transformation improved the residual vs fitted plot (Figure 3 right) so that the equal variance assumption is met.

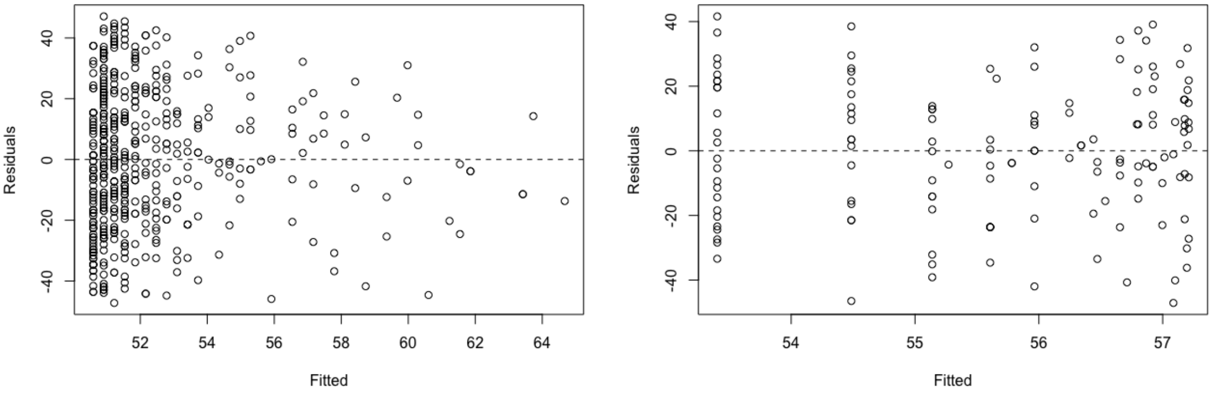
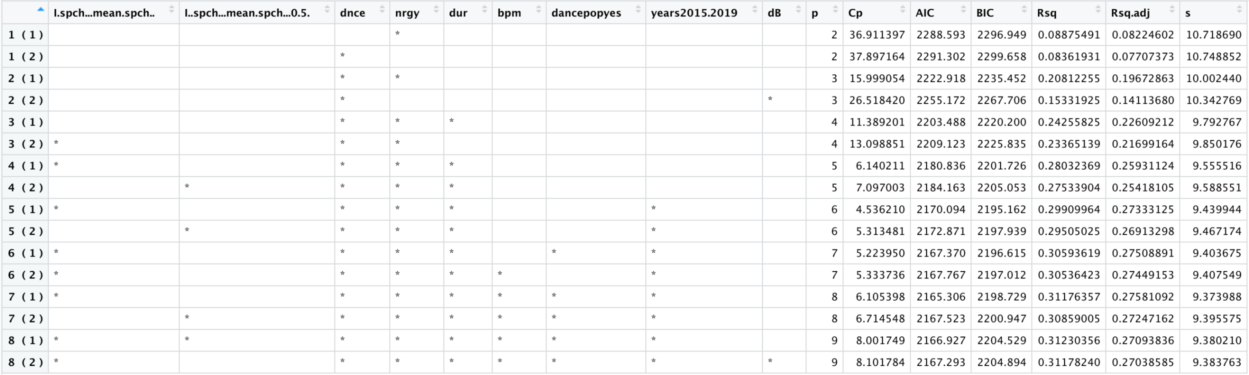


Figure 3: Residual plots for valence vs. speechiness. Left: before transformation. Right: after transformation

After this transformation, the full model residuals and normality appear reasonable. Next, we moved on to the variable selection process. We started by running the Best Subsets procedure on valence vs. centered speechiness, centered speechiness to the 1/2 power, danceability, energy, duration, BPM, genre, year, and decibel. We organized the suggested models by ascending BIC values (Figure 4).



	I.spch...mean.spch..	I..spch...mean.spch...0.5.	dnce	nrgy	dur	bpm	dancepopyes	years2015.2019	dB	p	Cp	AIC	BIC	Rsq	Rsq.adj	s
1 (1)				*						2	36.911397	2288.593	2296.949	0.08875491	0.08224602	10.718690
1 (2)			*							2	37.897164	2291.302	2299.658	0.08361931	0.07707373	10.748852
2 (1)			*	*						3	15.999054	2222.918	2235.452	0.20812255	0.19672863	10.002440
2 (2)			*						*	3	26.518420	2255.172	2267.706	0.15331925	0.14113680	10.342769
3 (1)			*	*	*					4	11.389201	2203.488	2220.200	0.24255825	0.22609212	9.792767
3 (2)	*		*	*						4	13.098851	2209.123	2225.835	0.23365139	0.21699164	9.850176
4 (1)	*		*	*	*					5	6.140211	2180.836	2201.726	0.28032369	0.25931124	9.555516
4 (2)		*	*	*	*					5	7.097003	2184.163	2205.053	0.27533904	0.25418105	9.588551
5 (1)	*		*	*	*			*		6	4.536210	2170.094	2195.162	0.29909964	0.27333125	9.439944
5 (2)		*	*	*	*			*		6	5.313481	2172.871	2197.939	0.29505025	0.26913298	9.467174
6 (1)	*		*	*	*		*	*		7	5.223950	2167.370	2196.615	0.30593619	0.27508891	9.403675
6 (2)	*		*	*	*	*		*		7	5.333736	2167.767	2197.012	0.30536423	0.27449153	9.407549
7 (1)	*		*	*	*	*	*	*		8	6.105398	2165.306	2198.729	0.31176357	0.27581092	9.373988
7 (2)		*	*	*	*	*	*	*		8	6.714548	2167.523	2200.947	0.30859005	0.27247162	9.395575
8 (1)	*	*	*	*	*	*	*	*		9	8.001749	2166.927	2204.529	0.31230356	0.27093836	9.380210
8 (2)	*		*	*	*	*	*	*	*	9	8.101784	2167.293	2204.894	0.31178240	0.27038585	9.383763

Figure 4: Method of best subsets

While analyzing for best subsets, we find that the criteria values (AIC, BIC, R^2, R^2_{adj}, s) were comparable for all models; therefore, we focused on comparing p and C_p. We decided start with model 5(2) (centered speechiness 1/2, danceability, energy, duration, and year) since $p = 6$ and $C_p = 5.313$ are the closest out of the other models while maintaining low AIC, BIC, and s values.

6 Residual Analysis

We produce residual plots, QQ plots, and histograms to analyze the L.I.N.E. assumptions for our chosen model (Figure 5). The residual vs. fitted plot for our new model does not appear to reveal any violations of linearity or equal variance due to the lack of curvature and the roughly equal spread of data points. The Normal Q-Q plot and residual histogram does not appear to reveal any violations of the normality assumption due to its approximately linear fit and having a roughly symmetric histogram of residuals.

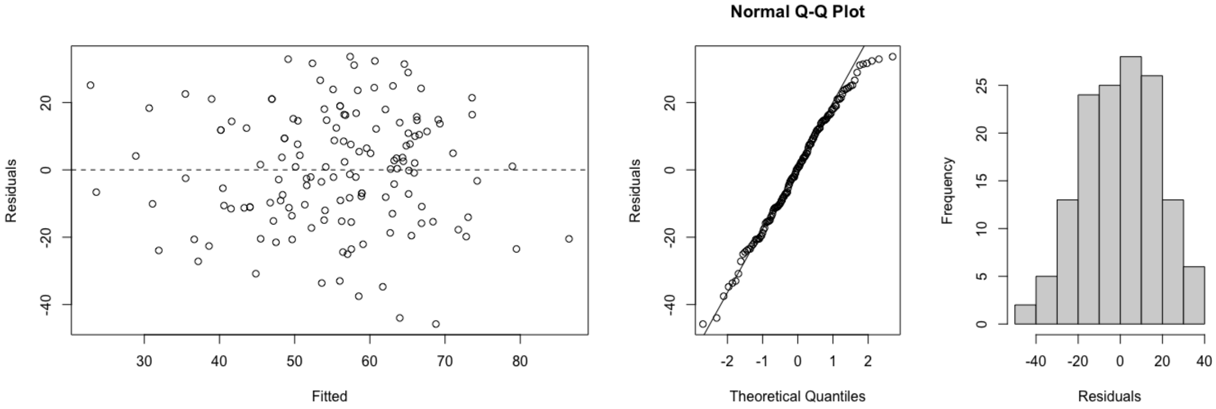


Figure 5: Left: residual plot for linear model. Right: QQ plot and histogram of residuals.

The Shapiro-Wilk normality test $p-value = 0.2422$ and the studentized Breusch-Pagan test $p-value = 0.1619$ are large, confirming our beliefs that the normality and equal variance assumptions are not violated. The VIFs for all of the explanatory variables in the model are all less than 2, indicating that multicollinearity is not a problem (Figure 6). Since there are no replicates in our model, a lack of fit test is not applicable.

$\sqrt{spch - mean(spch)}$	$dnce$	$nrgy$	dur	$year$
1.086536	1.076225	1.127197	1.129641	1.179301

Figure 6: VIFs for linear model

7 Fit a Linear Model

After fitting a linear model with our final chosen predictors, we have the resulting equation:

	Slope Coefficient	t-Statistic	p-value
Intercept	12.15	0.71	0.48
$\sqrt{spch - mean(spch)}$	2.47	2.39	0.018
$dnce$	0.53	4.97	2.02 e-6
$nrgy$	0.52	4.31	3.12e-5
dur	-0.14	-3.41	8.44e-4
$years2015 - 2019$	-6.35	-1.95	0.053

Figure 7: Linear model summary

$$\hat{val} = 12.1528 + 2.471\sqrt{spch - mean(spch)} + 0.533(dnce) + 0.520(nrgy)–0.1394(dur)–6.346(years1519)$$

For the most part, the coefficients from our model do make sense contextually. The valence of a song is a measure of the happiness of the song. Thus, we would expect danceability, energy, and speechiness to have positive coefficients since those qualities are generally associated with happiness. It also makes sense that duration would have a negative coefficient since longer songs are often slower and have the potential to be more serious. We have no clear reason to believe that there should be a difference in valence levels between songs released in the first half of the decade (2010-2014) compared to the second half (2015-2019). Therefore, it would make sense to obtain a large p-value for the "years2015-2019" dummy variable. Our $p - value = 0.053$ is borderline and may indicate that a difference does exist, however, there is not strong evidence.

Based on our model, we can make the following interpretations:

- When a song from 2010 to 2014 has a speechiness score equal to the mean speechiness score and danceability, energy, and duration all have scores equal to 0, we can be 95% confident that the mean valence score will be between -21.491 and 45.796.
- For every 1 unit increase in danceability score, we can be 95% confident that the mean increase in the valence score will be between 0.321 and 0.745 points when all other variables are held constant.
- With all other variables held constant, we can be 90% confident that the mean valence score of a song released between 2015 and 2019 will be between 0.957 and 11.736 points less than a song released between 2010 and 2014.
- An $R^2 = 0.2951$ indicates that 29.51% of the variability in valence score can be explained by our model.
- An $s = 17.71$ (or $RMSE$) indicates a typical prediction error of 17.71 units.

Since all of the explanatory variables in our model have variance inflation factors less than 2, we are fairly confident that multicollinearity is not an issue (Figure 6).

8 Statistical Inference

8.1 Overall Model Significance

We proceed to test for the overall model significance.

$$H_0 : \beta_{(ctrspch)^{1/2}} = \beta_{dnce} = \beta_{nrgy} = \beta_{dur} = \beta_{years} = 0 \qquad H_a : \text{ At least one } \beta_i \neq 0$$

Our resulting test statistic is $F = 11.38$ on 5 and 136 degrees of freedom, yielding a p-value of < 0.0001 (Figure 7). Based on the low p-value, there is strong evidence that at least one of the coefficients is not equal to zero; therefore, the overall model is statistically significant.

8.2 Partial F-Test

Next we tested to see if multiple coefficients are statistically significant (danceability, energy, duration, and year).

$$H_0 : \beta_{dnce} = \beta_{nrgy} = \beta_{dur} = \beta_{years} = 0 \qquad H_a : \text{ At least one } \beta_i \neq 0$$

Our resulting test statistic is $F = 14.104$ on 4 degrees of freedom, yielding a p-value of < 0.0001. Based on the low p-value, we have strong evidence that at least one of the coefficients for the variable in the full model is nonzero. We were able to obtain evidence that adding danceability, energy, duration, and years significantly improves the model that already included the centered, transformed speechiness. Thus we concluded that the full model does perform significantly better than the reduced model, and we can confidently stick with the full model.

8.3 Interaction Plot Significance

We decided to test the significance of an interaction between energy and year, after adjusting for the full model.

$$H_0 : \beta_{nrgy*year} = 0 \qquad H_a : \beta_{nrgy*year} \neq 0$$

Our resulting test statistic is $t = 132$, yielding a p-value of 0.89528. Based on the high p-value, we have insufficient evidence of an interaction between energy and year, after adjusting for all other variables. This matches our expectations from the coded scatterplot. We thought that the interaction between energy and years was not very significant since the difference in the slopes of the 2010-2014 and 2015-2019 lines was not very large.

8.4 Predictive Inference

We used our model to predict the valence of a song from 2014-2019 with mean values from our training data: 8.42 speechiness, 64.39 danceability, 70.74 energy, and 225.53 duration. Our model yields a valence score of 51.81 with the given bounds shown in figure 8.

Type	**Lower**	**Upper**
Confidence	44.489	59.121
Prediction	16.024	87.587

Figure 8: Confidence and Prediction Interval for valence score 51.81.

Based on the results, we can make the following interpretations:

- Given this fit, we are 95% confident that the mean valence score for all songs from 2014-2019 with the given scores is between 44.489 and 59.120.

- We are also 95% confident that an individual valence score for a song from 2014-2019 with the given scores is between 16.023 and 87.586 points.

We decided to choose the mean value of each variable because we did not want to have the risk of hidden extrapolation. Additionally, our data set focuses on popular songs from 2010-2019, so it may be subjective if we select a recent song to get predicted values, as that song may not be popular and not fall under the correct population.

We created a 95% confidence interval for our linear model (Figure 9).

Coefficients	**2.5%**	**97.5%**
Intercept	-21.49	45.79
$\sqrt{spch - mean(spch)}$	0.42	4.52
$dnce$	0.32	0.74
$nrgy$	0.28	0.76
dur	-0.22	-0.059
$years2015 - 2019$	-12.78	0.090

Figure 9: Confidence intervals for linear model coefficients.

We can make the following interpretations:

- After adjusting for the other variables, we are 95% confident that for every one point increase in centered speechiness score to the power of 1/2 , the mean valence score will increase between 0.4244 and 4.517 points, for the population of popular songs from 2010 to 2019 on Spotify.
- After adjusting for the other variables, we are 95% confident that for every one point increase in danceability score, the mean valence score will increase between 0.3205 and 0.7448 points for the population of popular songs from 2010 to 2019 on Spotify.
- After adjusting for the other variables, we are 95% confident that for every one point increase in energy score, the mean valence score will increase between 0.2814 and 0.7587 points, for the population of popular songs from 2010 to 2019 on Spotify.
- After adjusting for the other variables, we are 95% confident that for every one point increase in duration score, the mean valence score will decrease between -0.2201 and -0.0586, for the population of popular songs from 2010 to 2019 on Spotify.
- After adjusting for the other variables, we are 95% confident that the mean valence score for years 2015-2019 will decrease up to -12.782 points and increase up to 0.0895 points compared to the years 2010-2014, for the population of popular songs from 2010 to 2019 on Spotify.

In order to obtain a family-wide Confidence Coefficient of at least 95%, we need each individual interval to have a confidence level of 99.17%. We are at least 99.17% confident the 6 intervals simultaneously capture their corresponding population.

Coefficients	**0.417%**	**99.583%**
Intercept	-33.39	57.702
$\sqrt{spch - mean(spch)}$	-0.29	5.24
$dnce$	0.25	0.82
$nrgy$	0.19	0.84
dur	-0.25	-0.030
$years2015 - 2019$	-15.059	2.37

Figure 10: Confidence intervals, after Bonferroni adjustment, for linear model coefficients.

8.5 Interaction Between Two Quantitative Variables

We decided to test the significance of an interaction between energy and danceability, two quantitative variables, after adjusting for the full model. Our resulting test statistic is t = 0.62, yielding a p-value of 0.54. Based on the small t-statistic and large p-value, we can see that there is no evidence of an interaction between energy and danceability (Figure 11).

	Slope Coefficient	t-Statistic	p-value
Intercept	34.591	0.858	0.392
$\sqrt{spch - mean(spch)}$	2.511	2.416	0.017
$years2015 - 2019$	-6.394	-1.960	0.052
dur	-0.138	-3.386	0.001
$nrgy$	0.199	0.373	0.709
$dnce$	0.177	0.301	0.764
$nrgy : dnce$	0.005	0.614	0.540

Figure 11: Linear model summary with interaction term.

9 Model Validation

After finalizing, we used the testing dataset set aside to test and validate our model. We produce the following statistics on the models performance:

$$MSE = 300.441 \qquad PRESS = 46622.46$$
$$MSPE = 283.248 \qquad R^2_{adj} = 0.23$$

Since the mean squared predicted error ($MSPE$) of 283.248 is just slightly smaller than the mean squared error (MSE) 300.441, we conclude that the predictive ability of our model is acceptable.

	Slope Coefficient	t-Statistic	p-value
Intercept	2.51	0.17	0.87
$\sqrt{spch - mean(spch)}$	2.42	2.57	0.01
$dnce$	0.53	5.40	2.22e-7
$nrgy$	0.54	5.20	5.52e-7
dur	-0.11	-2.98	0.003
$years2015 - 2019$	-3.90	-1.34	0.18

Figure 12: Total linear model summary

After fitting the model for the full data set, all of the coefficients are approximately the same except for "years2015-2019" which increased significantly from -6.346 to -3.896 and had a change in its p-value from 0.053 to 0.181 (Figure 12). With the training data, the p-value was borderline so we concluded that a difference may exist between the valence of songs from 2010 to 2014 vs. 2015 to 2019. However, now with a much larger p-value we can see that the previous small value was most likely a type I error and it is probably safer to conclude that there is no evidence that a true relationship exists.

10 Conclusion

This model is valid according to the validation analysis above, where the model using the training data was capable of being fairly accurate at predicting valence values for the test data. This model is significant with an overall model utility p-value < 0.0001. The model offers several useful insights into the behavior of valence values for popular songs. Based on the coefficients from the overall model, the positivity of songs tends to increase as the duration decreases, the energy increases, the danceability increases, or the speechiness increases, after accounting for all of these variables and the year.

An important strength of this model is that all but one of the variables are significant in predicting the valence of songs. Another positive is that 29.08% of variability in valence is accounted for by this model, which taking into consideration the breadth and diversity of music, this is a decent percentage. A weakness of this model is the lack of significance of the year variable. This variable was included because it was a borderline significant variable and because it was suggested by the best subsets method (also because it was a categorical variable), however after fitting the model with the full dataset the coefficient was reduced by half and the p-value increased by quite a bit. It can also be said that 29.08% of the variation in valence being explained by this model is a weakness because it is a low percentage. While this might typically be true, it is also important to keep in mind that popular songs come from dozens of genres of music, and thus have hugely varying moods, so even 30% of variation being able to be explained is impressive. Another weakness is that the strength of the model did have to be sacrificed a little bit to ensure the assumptions were reasonably met. For example, a model with speechiness to the 1/2 power, and not centered, accounted for approximately 40% of the variability in valence, but the equal variance and normality were violated.

At the beginning of this project, the first topic of interest was what determined the popularity of songs. Because global popularity is so complex, instead the mood of popular songs became the focus of this investigation. However, a future exploration into the valence of songs and its relationship with popularity, perhaps including the genre of the songs and the regions in which they originate, would be very interesting to investigate. If the study were run again, it would be reasonable to increase the data sampling from just 2010-2019 to several more decades, which would allow for how the mood of popular songs differs from generation to generation. This might also allow year to become a significant variable, as a single decade is not much time for a significant change in popular culture to occur. Otherwise, the data collection was completed effectively and was as controlled as possible for this type of data.

In conclusion, due to the complexity of global popular culture, a perfect model would be impossible to create using only a few variables. However, with these resources, the model described in this report is a satisfactorily accurate predictor of the mood of popular songs.

An interesting study from February of 2022, published in the peer-reviewed journal Behavior Research Methods, proposed a way of measuring national mood and life satisfaction from the valence of popular songs. The article is titled “Measuring national mood with music: using machine learning to construct a measure of national valence from audio data” and it concludes that “the valence of a country’s most popular songs can provide a reliable indication of average life satisfaction in the population.” Although a different focus than this report, these results provide compelling insight into how the valence of popular songs can be a meaningful factor to explore.

11 Appendix

11.1 Appendix A

The original source of our data is from Kaggle by Leonardo Henrique.
https://www.kaggle.com/datasets/leonardopena/top-spotify-songs-from-20102019-by-year

Variables (description from original source)

title	Song's title
artist	Song's artist
top.genre	The genre of the track
year	Song's year in the Billboard
bpm	Beats per minute - the tempo of the song
nrgy	The energy of a song - the higher the value, the more energetic
dnce	Danceability - The higher the value, the easier it is to dance to this song
dB	Loudness in dB - The higher the value, the louder the song
live	Liveness - The higher the value, the more likely the song is a live recording
val	Valence - The higher the value, the more positive the mood for the song
dur	Length - The duration of the song
acous	Acousticness - The higher the value the more acoustic the song is
spch	Speechiness - The higher the value the more spoken words the song contains
pop	Popularity - The higher the value the more popular the song is

The variables **bpm**, **nrgy**, **dnce**, **dB**, **live**, **val**, **dur**, **acous**, **spch**, and **pop** are metrics derived from Spotify. More information on these variables can be found on the Spotify Developer website.
https://developer.spotify.com/documentation/web-api/reference/#/operations/get-several-audio-features

11.2 Appendix B

Full script will be given on Canvas upon submission. Code shown is separated by script used in each section.

11.2.1 Splitting the Data

```r
set.seed(23173)
numtrain = ceiling(.8*nrow(df))
train_ind = sample(nrow(df), numtrain)
traindata = df[train_ind, ]
testdata = df[-train_ind, ]

# Distribution of song data per year
# Entire Data
g <- ggplot(data = df, aes(x=factor(year)))
g + geom_bar(fill="lightblue") +
  scale_x_discrete(limits=factor(c(2010,2011,2012,2013,2014,2015,2016,2017,2018,2019))) +
  labs(title = "Distribution of Songs per Year (n=602)",
       subtitle = "Entire Dataset (After removing ’Million Years Ago’ by Adele)") +
  xlab("Year") + ylab("Count") + ylim(0,100)

# Train Data
g <- ggplot(data = traindata, aes(x=factor(year)))
g + geom_bar(fill="lightblue") +
  scale_x_discrete(limits=factor(c(2010,2011,2012,2013,2014,2015,2016,2017,2018,2019))) +
  labs(title = "Distribution of Songs per Year (n=482)",
       subtitle = "Training Dataset (After removing ’Million Years Ago’ by Adele)") +
  xlab("Year") + ylab("Count") + ylim(0,100)

# Test Data
g <- ggplot(data = testdata, aes(x=factor(year)))
g + geom_bar(fill="lightblue") +
  scale_x_discrete(limits=factor(c(2010,2011,2012,2013,2014,2015,2016,2017,2018,2019))) +
  labs(title = "Distribution of Songs per Year (n=120)",
       subtitle = "Testing Dataset (After removing ’Million Years Ago’ by Adele)") +
  xlab("Year") + ylab("Count") + ylim(0,100)
```

11.2.2 Data Visualization

```r
# Functions for adding correlation panel
# SOURCE: http://www.sthda.com/english/wiki/scatter-plot-matrices-r-base-graphs
panel.cor <- function(x, y){
  usr <- par("usr"); on.exit(par(usr))
  par(usr = c(0, 1, 0, 1))
  r <- round(cor(x, y), digits=2)
  txt <- paste0("r = ", r)
  text(0.5, 0.5, txt, cex=1.7)
}

upper.panel<-function(x, y){
  points(x,y, pch = 19, col = traindata$year)
}

# Matrix Scatterplot
pairs(traindata[,c(11,7,8,12,14)], pch=19, cex=0.5)

# Matrix Scatterplot (colored)
pairs(traindata[,c(11,7,8,12,14)],
      lower.panel = panel.cor, upper.panel = upper.panel)

# Correlation Matrix
cor(traindata[,c(11,7,8,12,14)])

# Interaction Plots

# Val ~ Nrgy
```

```
year.f=factor(traindata$years)
par(mfrow=c(1,1))
plot(val ~ nrgy, data=traindata,
     main = ’Valence vs. Energy’,
     xlab=’Energy’, ylab= ’Valence’,
     col=c(’orange’,’blue’)[year.f],
     pch=c(16,17)[year.f])
legend(’topleft’,
       legend=levels(factor(traindata$years)),
       col=c(’orange’,’blue’),
       pch=c(16,17))
modAP=lm(val ~ nrgy + year.f+nrgy*year.f, data=traindata)
summary(modAP)
#Fitted line for other
abline(7.1647,0.6525, col=’orange’)
#Fitted line for vic
abline(7.1647+13.4900,0.6525-0.2261,col=’blue’)

# Val ~ Dnce
plot(val ~ dnce, data=traindata,
     main = ’Valence vs. Danceability’,
     xlab=’Danceability’, ylab= ’Valence’,
     col=c(’orange’,’blue’)[year.f],
     pch=c(16,17)[year.f])
legend(’topleft’,
       legend=levels(factor(traindata$years)),
       col=c(’orange’,’blue’),
       pch=c(16,17))
modAP=lm(val ~ dnce + year.f+dnce*year.f, data=traindata)
summary(modAP)
#Fitted line for other
abline(-7.5084,0.9957, col=’orange’)
#Fitted line for vic
abline(-7.5084+2.9559,0.9957-0.1620,col=’blue’)

# Val ~ dur
plot(val ~ dur, data=traindata,
     main = ’Valence vs. Duration’,
     xlab=’Duration’, ylab= ’Valence’,
     col=c(’orange’,’blue’)[year.f],
     pch=c(16,17)[year.f])
legend(’topleft’,
       legend=levels(factor(traindata$years)),
       col=c(’orange’,’blue’),
       pch=c(16,17))
modAP=lm(val ~ dur + year.f+dur*year.f, data=traindata)
summary(modAP)
#Fitted line for other
abline(106.44057,-0.21586, col=’orange’)
#Fitted line for vic
abline(106.44057-15.20846,-0.21586-0.02569,col=’blue’)

# Val ~ spch
plot(val ~ spch, data=traindata,
     main = ’Valence vs. Speechiness’,
     xlab=’Speechiness’, ylab= ’Valence’,
     col=c(’orange’,’blue’)[year.f],
     pch=c(16,17)[year.f])
legend(’topleft’,
       legend=levels(factor(traindata$years)),
       col=c(’orange’,’blue’),
       pch=c(16,17))
modAP=lm(val ~ spch + year.f+spch*year.f, data=traindata)
summary(modAP)
#Fitted line for other
abline(55.11701,0.09954, col=’orange’)
#Fitted line for vic
abline(55.11701-9.87438,0.09954-0.41772,col=’blue’)
```

11.2.3 Variable Preprocessing

```
# Boxcox Procedure
old = lm(val~spch+dnce+nrgy+dur+bpm+dancepop+years+dB, data=traindata)
boxcox(val~spch+dnce+nrgy+dur+bpm+dancepop+years+dB, data=traindata, lambda=seq(0,1,by=0.1))
plot(resid(old)~fitted(old), xlab=’Fitted’, ylab=’Residuals’)
abline(h=0, lty=2)

old2 = lm(I(val^0.8)~spch+dnce+nrgy+dur+bpm+dancepop+years+dB, data=traindata)
plot(resid(old2)~fitted(old2), xlab=’Fitted’, ylab=’Residuals’)
abline(h=0, lty=2)

# Residuals: val ~ spch
test.spch=lm(val~spch,data=traindata)
plot(resid(test.spch)~fitted(test.spch), xlab="Fitted", ylab="Residuals")
abline(h=0, lty=2)

# Residuals: val ~ spch + spch1/2
test.fit2=lm(val~I(spch-mean(spch))+I((spch-mean(spch))^(1/2)),data=traindata)
plot(resid(test.fit2)~fitted(test.fit2), xlab="Fitted", ylab="Residuals")
abline(h=0, lty=2)


all=lm(val~(I(spch-mean(spch))+I((spch-mean(spch))^2))+dnce+nrgy+dur+bpm+dancepop+years+dB,
    data=traindata)
best.sub=summary(regsubsets(formula(all), data=traindata, method="exhaustive", nbest=2))
best.sub
```

11.2.4 Residual Analysis

```
# Final model
final.fit <- lm(val~I((spch-mean(spch))^0.5)+dnce+nrgy+dur+years,data=traindata)

#Residuals for final model
par(mfrow=c(1,1))
plot(resid(final.fit)~fitted(final.fit), ylab=’Residuals’, xlab=’Fitted’)
abline(h=0,lty=2)

#Normality plots for final model
par(mfrow=c(1,2))  #plot in 1 by 2 grid
qqnorm(resid(final.fit), ylab=’Residuals’)
qqline(resid(final.fit))
hist(resid(final.fit), main=’’, xlab=’Residuals’)

#Shapiro and BP Test
shapiro.test(resid(final.fit))
bptest(final.fit)
vif(final.fit)
```

11.2.5 Statistical Inference

```
# Overall model significance
summary(final.fit)

# Partial F-test
fit.full = lm(val~(I((spch-mean(spch))^(1/2)))+dnce+nrgy+dur+years, data=traindata)
fit.reduced = lm(val~(I((spch-mean(spch))^(1/2))), data=traindata)
anova(fit.reduced, fit.full)

# Interaction plot significance
interaction.fit <- lm(val~(I((spch-mean(spch))^(1/2)))+dnce+nrgy+dur+years+nrgy:years, data=
    traindata)
summary(interaction.fit)

# CI and PI for mean values
```

```
x0 <- data.frame(spch=mean(traindata$spch), dnce=mean(traindata$dnce),
                 nrgy=mean(traindata$nrgy), dur=mean(traindata$dur), years=’2010-2014’)

predict(final.fit, newdata=x0 , interval = "confidence",  level=0.95)
predict(final.fit, newdata=x0 , interval = "prediction",  level=0.95)

# CI for slope coefficient
confint(fit.full, level=0.95)

# Bonferrini Adjustment CI for slope coefficient
confint(fit.full, level=1-(1-0.95)/6)
```

11.2.6 Model Validation

```
#MSE and MSPE for model validation
ms <- summary(final.fit)
MSE = mean(ms$residuals^2)
predicted = predict(final.fit, testdata[])
for (i in 1:120) {
  predicted[i] = predict(final.fit, testdata[i,])
}
actual = testdata$val
MSPE = mean((predicted-actual)^2)

c(MSE, MSPE)

#PRESS and Predicted R^2
e_i = resid(final.fit) #residuals
h_i = hatvalues(final.fit) #leverages
SST = sum((anova(final.fit))$’Sum Sq’) #SSTotal
PRESS = sum( (e_i/(1-h_i))^2 )
Pred.Rsq = 1-PRESS/SST

PRESS
Pred.Rsq

#Model using all observations
total.fit = lm(val~I((spch-mean(spch))^0.5)+dnce+nrgy+dur+years,data=df)
summary(total.fit)
```