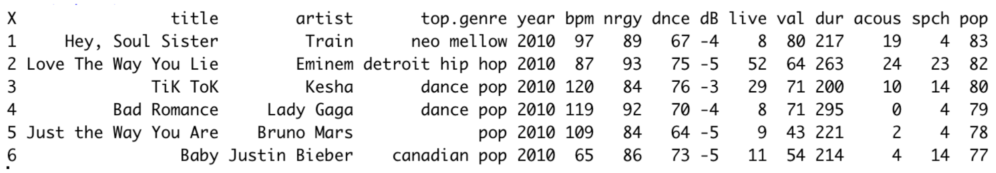
Analyzing Spotify Music Popularity from 2010 to 2019

Christina Xu
U12512757
MA 575 Linear Models (Spring 2022)
May 4, 2022

Table of Contents

I. Univariate Data Analysis



```
X                title        artist       top.genre year bpm nrgy dnce dB live val dur acous spch pop
1     Hey, Soul Sister         Train      neo mellow 2010  97   89   67 -4    8  80 217    19    4  83
2 Love The Way You Lie        Eminem detroit hip hop 2010  87   93   75 -5   52  64 263    24   23  82
3              TiK ToK         Kesha       dance pop 2010 120   84   76 -3   29  71 200    10   14  80
4          Bad Romance     Lady Gaga       dance pop 2010 119   92   70 -4    8  71 295     0    4  79
5 Just the Way You Are    Bruno Mars             pop 2010 109   84   64 -5    9  43 221     2    4  78
6                 Baby Justin Bieber    canadian pop 2010  65   86   73 -5   11  54 214     4   14  77
```

Table 1.1 Dataset

A. Hypothesis test for the true population mean danceability of top Spotify songs from 2010 to 2019

According to Spotify, danceability describes how suitable a track is for dancing based on a combination of musical elements including tempo, rhythm stability, beat strength, and overall regularity. In the dataset, danceability ranges from 0 to 97, where the higher the value, the easier it is to dance to a song. Based on logic, one would expect the mean danceability of top Spotify songs to be somewhere in the upper half of that range, so I arbitrarily claim that the true population danceability of top Spotify songs from 2010 to 2019 is 70. In the following section, I test my hypothesis with α = 0.05 because I am moderately confident in my proposed hypothesis.

Let μ be the true population mean of the danceability of top Spotify songs from 2010 to 2019

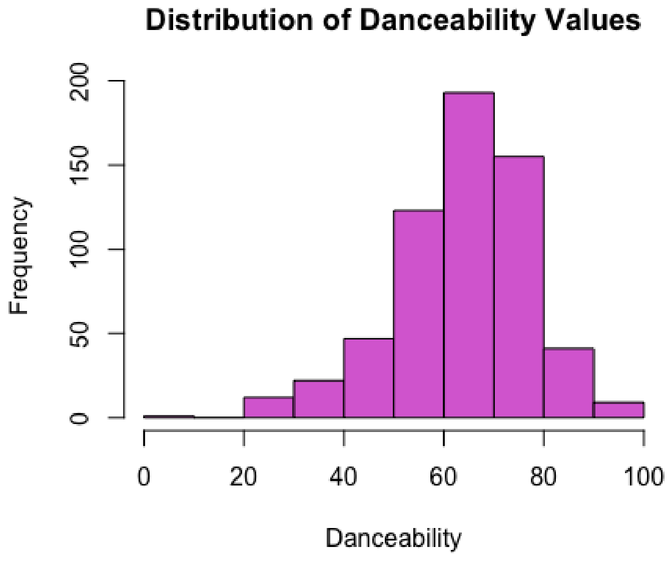
$$H_0: \mu = 70$$
$$H_1: \mu \neq 70$$

Metric Name	Metric Value
Lower Bound of Confidence Interval	63.3098
Upper Bound of Confidence Interval	65.4498
Claimed Value of Population Mean	70.0000
T statistic	-10.3157
T critical value	1.9639
p-value	4.3836e-23
alpha	0.0500

Table 1.2 Output from two tailed one-sample t-test

Reject H_0: $\mu = 70$ at a two-sided level since p value = 4.3836e-23 < $\alpha = 0.05$. There is significant evidence to suggest that the true population mean of the danceability of top Spotify songs from 2010 to 2019 is NOT 70. Based on the confidence interval, we are 95% confident that the true value of the population mean lies in the interval (63.310, 65.450). Although there is significant evidence to reject the null hypothesis, the conclusion is in line with my intuition that the mean danceability of top Spotify songs is in the upper half of the range (0,97).

These results are valid since the assumptions for a one sample t-test seem to be fulfilled: (1) independence and (2) normality. It is reasonable to assume the values of danceability are independent of each other. In other words, the value of danceability for one song does not influence the value of danceability for any other song. Furthermore, it appears that the distribution of danceability values is approximately normal because it roughly resembles a bell shape.



Graph 1.1 Distribution of Danceability Values

B. Hypothesis test for the true population standard deviation for danceability of top Spotify songs from 2010 to 2019

Since the sample standard deviation for danceability is 13.3787, I expect the population standard deviation to be similar, so I arbitrarily claim that it is 15. In the following section, I conduct a chi-square test to test my hypothesis with α = 0.05 because I am moderately confident in my claim.

Let σ be the true population standard deviation of the danceability of top Spotify songs from 2010 to 2019

$$H_0: \sigma = 15$$
$$H_1: \sigma \neq 15$$

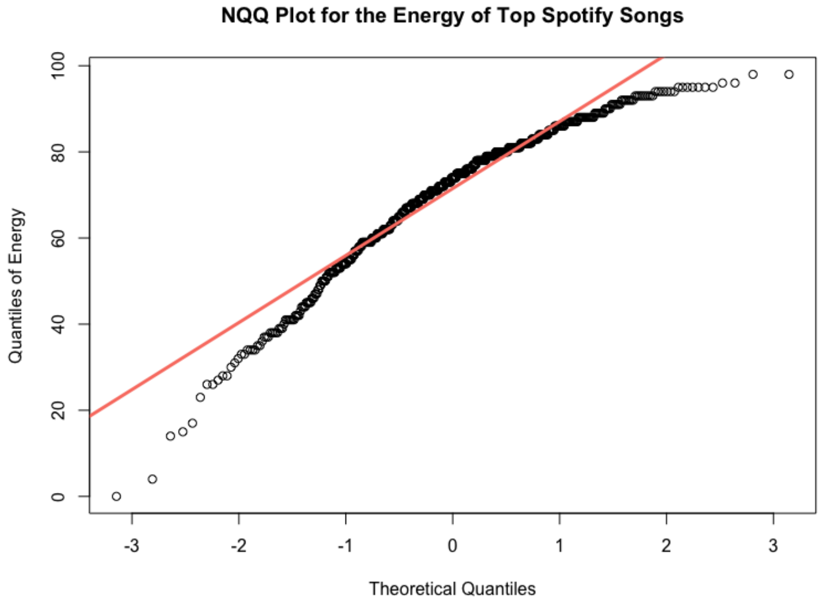
Metric Name	Metric Value
Lower Tail Bound	23.1496
Upper Tail Bound	25.9207
Claimed value of Population Standard Deviation σ	15.0000
Chi-squared test statistic	21.8837
alpha	0.0500

Table 1.3 Output from two tailed chi-square test

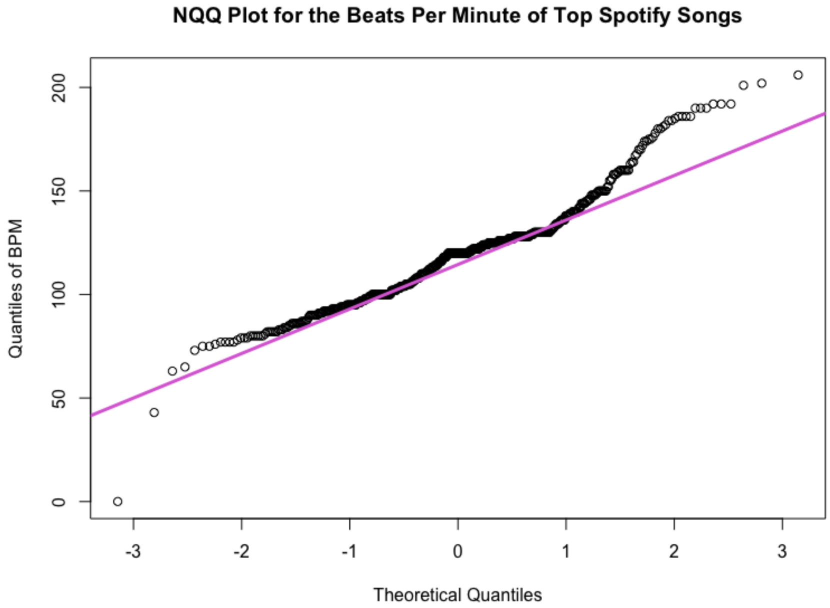
Reject H_0: σ = 15 at a two-sided level of α = 0.05 since the test statistic (21.8837) lies outside of the confidence interval (23.1496, 25.9207). According to the results of the test, 15 is not that close to 13.379. Thus, the conclusion does not match my intuition.

The same assumptions from the previous t-test hold for this chi-square test for the same reasons. Because the confidence interval for the true population value of the standard deviation suggests that it is relatively large, I am less confident in results of the previous hypothesis test since the margin of error is calculated using the sample standard deviation of danceability values.

C. Assessing the Normality of Beats per Minute and Energy Distributions



Graph 1.2: Normal Q-Q Plot for the Beats Per Minute of Top Spotify Songs



Graph 1.3: Normal Q-Q Plot for the Energy of Top Spotify Songs

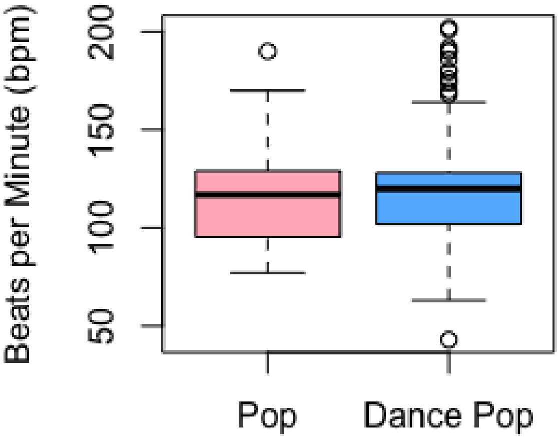
The resulting normal Q-Q plots produce some points that deviate from a line, providing evidence for non-normality. In Graph 1.1, the points form a curve that is concave up, suggesting that the distribution of bpm of top Spotify songs from 2010 to 2019 is right skewed. Conversely, in Graph 1.2, the points form a curve that is concave down, suggesting that the distribution for energy of top Spotify songs from 2010 to 2019 is left skewed.

	Correlation Coefficient
Graph 1.2	0.9769
Graph 1.3	0.9697

Table 1.4 Correlation coefficients associated with NQQ plots

However, upon further investigation into their respective correlation coefficients, they indicate that the distribution of bpm and distribution of energy are indeed normal because they exceed the threshold of 0.95 which is conventionally used for modest sized datasets.

D. Hypothesis Test for the true difference in population mean beats per minute between top pop songs and top dance pop songs on Spotify from 2010-2019



Graph 1.4 Boxplot comparing the distributions of bpm

I have no reason to believe that the true population mean bpm of pop and dance pop are the same since dance pop songs are generally more uptempo as they are intended to be played at nightclubs. This is supported by the boxplot where the sample mean bpm of dance pop songs in the dataset is slightly greater than that of pop songs. Because the sample sizes of pop and dance pop songs are unequal, 60 and 327 respectively, I assume they also have unequal variances. Thus, I will conduct a Welch's t-test with $\alpha = 0.01$ since I am confident in my claim.

Let μ_1 be the true population mean for bpm of top pop songs from 2010 to 2019
Let μ_2 be the true population mean for bpm of top dance pop songs from 2010 to 2019

$$H_0: \mu_1 = \mu_2$$
$$H_1: \mu_1 \neq \mu_2$$

Metric Name	MetricValue
Lower Bound of Confidence Interval	-10.4771
Upper Bound of Confidence Interval	7.6380
T statistic	-0.4139
T critical value	2.641
p-value	0.6801
alpha	0.0100

Table 1.5 Output from two tailed Welch's t-test

Fail to reject $H_0: \mu_1 = \mu_2$ at a two tailed level of $\alpha=0.01$ since the p-value is greater than the alpha level. There is no evidence to suggest that the true population mean bpm of top pop songs on Spotify from 2010 to 2019 is significantly different from the true population mean bpm top pop songs on Spotify from 2010 to 2019, contradicting my original claim. Furthermore, 0 is included in the confidence interval of the true population difference in means (-10.4771, 7.6380), providing additional evidence that there is no significant difference between the two.

The independence assumption is met since it is reasonable to assume that the bpm of songs both within and between genres are independent of each other. However, the results of the Welch's t-test may still be unreliable given that the normality assumption seems to be violated. The distribution of bpm for both pop and dance pop songs appears to be skewed to the left. The distribution of bpm of dance pop songs is especially problematic because there are several outliers in the upper right corner of Graph 1.4.

E. Hypothesis test for the difference in the population variance of beats per minute between top pop songs and top dance pop songs on Spotify from 2010-2019

I expect the true population variance of bpm between the two genres to be unequal given the difference in spreads between the two distributions as depicted in Graph 1.4. I will test my claim using a two tailed F test with an $\alpha = 0.01$ since I am confident in my claim.

Let σ_1^2 be the true population variance of beats per minute of top pop songs on Spotify from 2010 to 2019
Let σ_2^2 be the true population variance of beats per minute of top dance pop songs on Spotify from 2010 to 2019

$$H_0: \sigma_1^2 = \sigma_2^2$$
$$H_1: \sigma_1^2 \neq \sigma_2^2$$

Lower Bound of Confidence Interval	0.6969
Upper Bound of Confidence Interval	1.9746
F statistic for Lower Bound	0.8187
F statistic for Upper Bound	1.2215
F critical value for Lower Bound	0.5705
F critical value for Upper Bound	1.6166
p-value	0.3194
alpha	0.0100

Table 1.6 Output from two tailed F-test

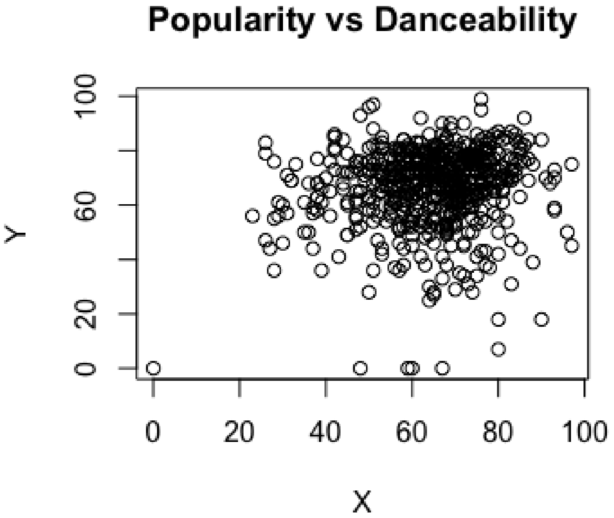
Fail to reject $H_0: \sigma_1^2 = \sigma_2^2$ at a two sided level of $\alpha = 0.01$ since the p-value is greater than alpha. There is no evidence that the true population variance for beats per minute of top pop songs is significantly different from the true population variance for top dance pop songs on Spotify from 2010-2019, contradicting my original claim. We are 99% confident that the true quotient between the true population variance of the two genres lies in the interval (0.6969, 1.9746). Since the value of 1 is included in the interval, this further supports that there is no significant difference between the true population variances of bpm between pop songs and dance songs. However, this does not invalidate the results of the previous section where I assumed unequal variance for the Welch's t-test since it still is appropriate to use due to the unequal sample sizes.

However, for the same reasons as the previous test, the results of the F-test are unreliable since the normality assumption seems to be violated because the distribution of bpm for both top pop songs and top dance pop songs is skewed.

II.A Simple Linear Regression

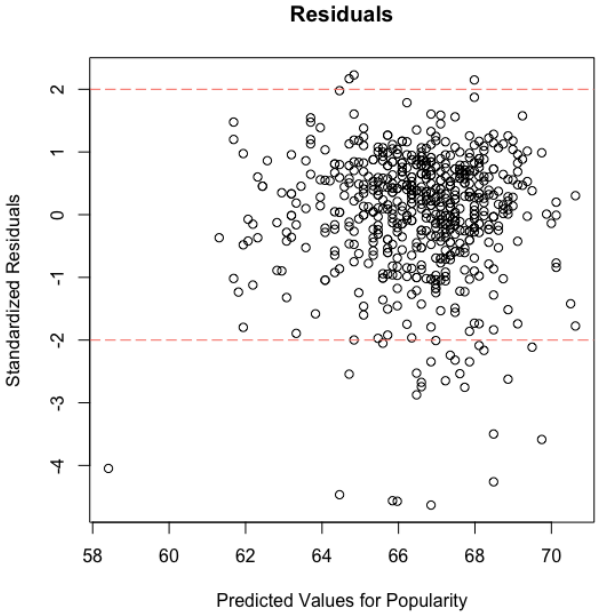
Let X or the explanatory variable be equal the danceability of a song
Let Y or the response variable be equal to the popularity of the song

I believe that X is linearly related to Y because based on common sense, the more suitable a track is for dancing, the more it is played at large group functions such as parties, and in turn the more people listen to the song.



Graph 2.1: Scatterplot of Popularity vs. Danceability

My prior belief that the popularity of a song is linearly related to its danceability is wrong since the data points on Graph 2.1 do not fall on a straight line. $E(Y \mid X = x) = B_0 + B_1 x$ and $\text{Var}(Y \mid X = x) = \sigma^2$ do not seem like reasonable assumptions, meaning that the linearity and constant variance assumptions of a linear model are violated. Furthermore, there are several extreme outliers that need to be investigated.



Graph 2.2 Plot of standardardized residuals

There is a random pattern in the standardized residual plot, however, there are several points whose standardized residuals fall outside the interval from -2 to 2, suggesting that they are outliers. Because there are so many outliers, rather than removing them, it suggests that the model is not a good fit for the data.

Let Ŷ be the predicted value of popularity for a song
Let X be the value of danceability for a song

Model: $\hat{Y} = 58.4131 + 0.1259X$

Model Parameter	Estimate	Standard Error
β_0	58.41306	2.89079975
β_1	0.125935	0.04396458

Table 2.1 SLR model parameter estimates and their associated standard errors

The estimate of β_1, the coefficient for the predictor variable, danceability is relatively small. This suggests that the predictive power of danceability for the popularity of a song is weak. The standard error for β_1 is small which means that the model estimates the unknown value of β_1 relatively precisely. The estimate of β_0, the intercept, can be interpreted as the estimated popularity of a song when its danceability value is 0. Thus, according to the model, a song that is extremely difficult to dance to would still have a popularity rating of 58.4. Because the standard error for β_0 is small, it suggests that the estimate is relatively precise.

r^2	0.0135
r^2-adj	0.0118
p-value	0.0043

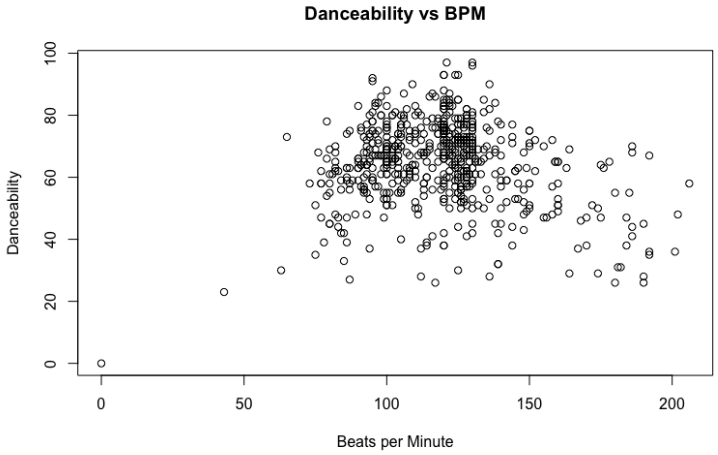
Table 2.2

According to the r^2 value, 1.350% of the total variability in popularity, the response variable, is explained by the model. After adjusting for danceability, the predictor variable, according to the r^2 adjusted value, 1.180% of the total variability in popularity, the response variable, is explained by the model. Both of these values suggest that the model fits the data poorly.

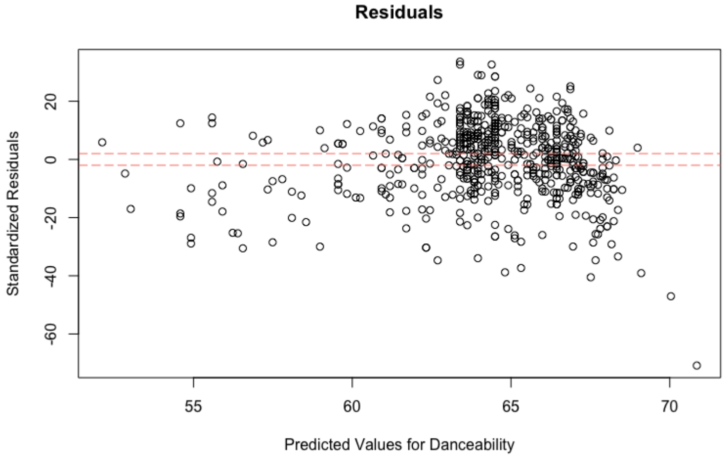
Surprising, the p-value for the F test comparing H_0: $B_1 = 0$ verses H_1: $B_1 \neq 0$ where B_1 represents the coefficient for the predictor variable, danceability, is significant because it is less that all conventional levels of α. Therefore, we reject the null hypothesis, concluding that B_1 is not equal to 0. However, the results of the F-test for the model are invalid since the assumptions for a linear model are violated. Therefore, this model has no utility and a transformation of Y and/or X needs to be made.

II.B Simple Quadratic Regression

Based on logic, it seems reasonable to assume that beats per minute (bpm) is quadratically related to danceability. It makes sense for a song to be difficult to dance to if it has an extremely slow or extremely fast tempo. The scatterplot of danceability against bpm supports my intuition because it appears that on average, songs with an extreme bpm have a smaller danceability value than songs with a moderate bpm.



Graph 2.3 Scatterplot of Danceability vs. Beats per Minute



Graph 2.4 Plot of standardized residuals

There is a clear nonrandom pattern in the standard residual plot where the spread of the data points decreases, suggesting that the constant variance assumption of a linear model is violated. Additionally, there are several points whose standardized residuals fall outside the interval from -2 to 2, suggesting that they are outliers. Both of these observations indicate that the model is not well-fitted to the data. Therefore, the model has no utility and a transformation on danceability, the response variable, and /or beats per minute, the predictor variable, needs to be made.

Let $\hat{Y}$ be estimated duration of a song
Let X be the beats per minute of a song

Model: $\hat{Y} = 70.8530 - 0.0004X^2$

Model Parameter	Estimate	Standard Error
β_0	70.8530	1.3504
β_1	-0.0004	8.000e-5

Table 2.3 Polynomial SLR model parameter estimates and their associated standard errors

As the value bpm increases by 1, the estimated value of danceability decreases by -0.0004. The magnitude of the β_1 estimate is extremely small which suggests that the predictive power of bpm for predicting danceability is weak. Because its standard error is extremely small, this is a reliable estimate for the true value of β_1. According to the estimate for β_0, when a song has a bpm of 0, its predicted value for danceability is 70.85. However, this interpretation is nonsensical because it is impossible for a song to have a bpm of 0. Its standard error is relatively small, suggesting that it is a reliable estimate for the true value of β_0.

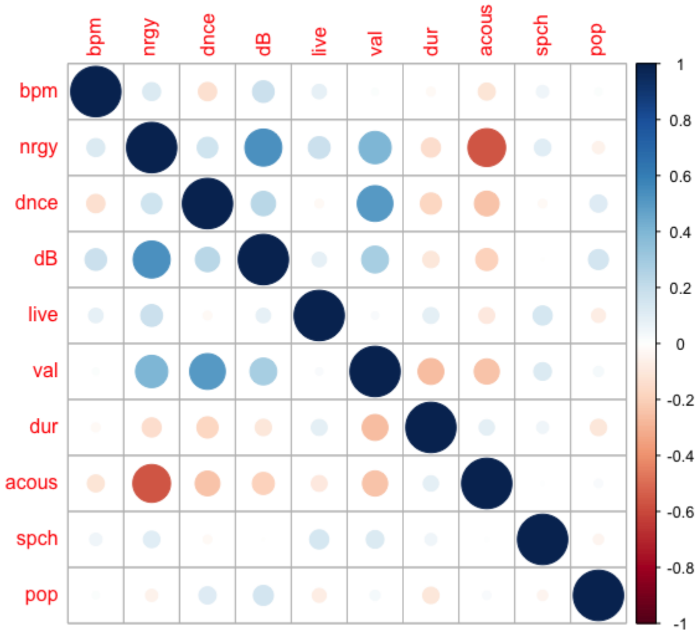
r^2	0.0433
r^2-adj	0.0417
p-value	2.0000e-7

Table 2.4

Both the r^2 and r^2-adjusted values of the model are small, 4.3300% and 4.170% , respectively. This suggests that the model fails to explain most of the variability in danceability, the response variable, indicating that it fits the data poorly.

Similar to the previous section, surprisingly, the p-value for the F test comparing H_0: $B_1 = 0$ verses H_1: $B_1 \neq 0$ where B_1 represents the coefficient for the predictor variable, bpm, is significant because it is less than all conventional values of α. Therefore, we reject the null hypothesis, concluding that B_1 is not equal to 0. However, the results of the F-test for the model are invalid since the assumptions for a linear model are violated. Therefore, this model has no utility and an additional transformation on Y and/or X needs to be made.

III. Multiple Linear Regression



Graph 3.1 Correlation matrix

Based on the correlation matrix of all the quantitative variables, it appears that the danceability, decibels, and duration of a song has the strongest linear association with its popularity. Therefore, a model containing these three variables as predictors was fitted to the data in or to predict popularity, the response variable.

Let Ŷ be the predicted value of popularity for a song
Let X1 be the predictor variable for the danceability of a song
Let X2 be the predictor variable for the decibels of a song
Let X3 be the predictor variable for the duration of a song

Model: $\hat{Y} = 72.7936 + 0.0776X1 + 0.6853X2 - 0.0331X3$

eg. According to the model the predicted value of popularity of a song when it has a danceability value of 60, -8 dB, and is 170 seconds long is 66.3328 seconds.

$$\hat{Y} = 72.7936 + 0.0776(60) + 0.6853(-8) - 0.0331(170) = 66.3328$$

Analysis of Variance

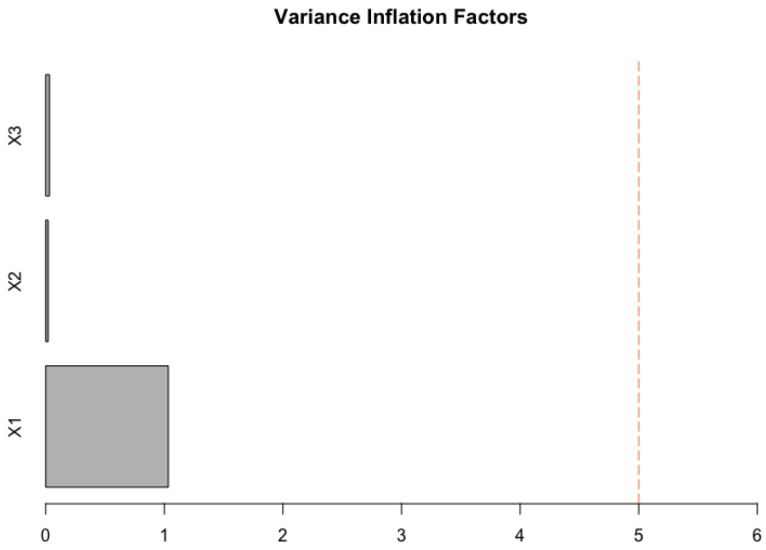
Source	Sum of Squares (SS)	Degrees of Freedom (df)	Mean Squares (MS)	F Value	p-value
Model	4713.1967	599	1571.0656	7.7031	0.00005
Error	122167.2942	600	203.9521		
Total	126880.4909	602	210.7649		

Table 3.1

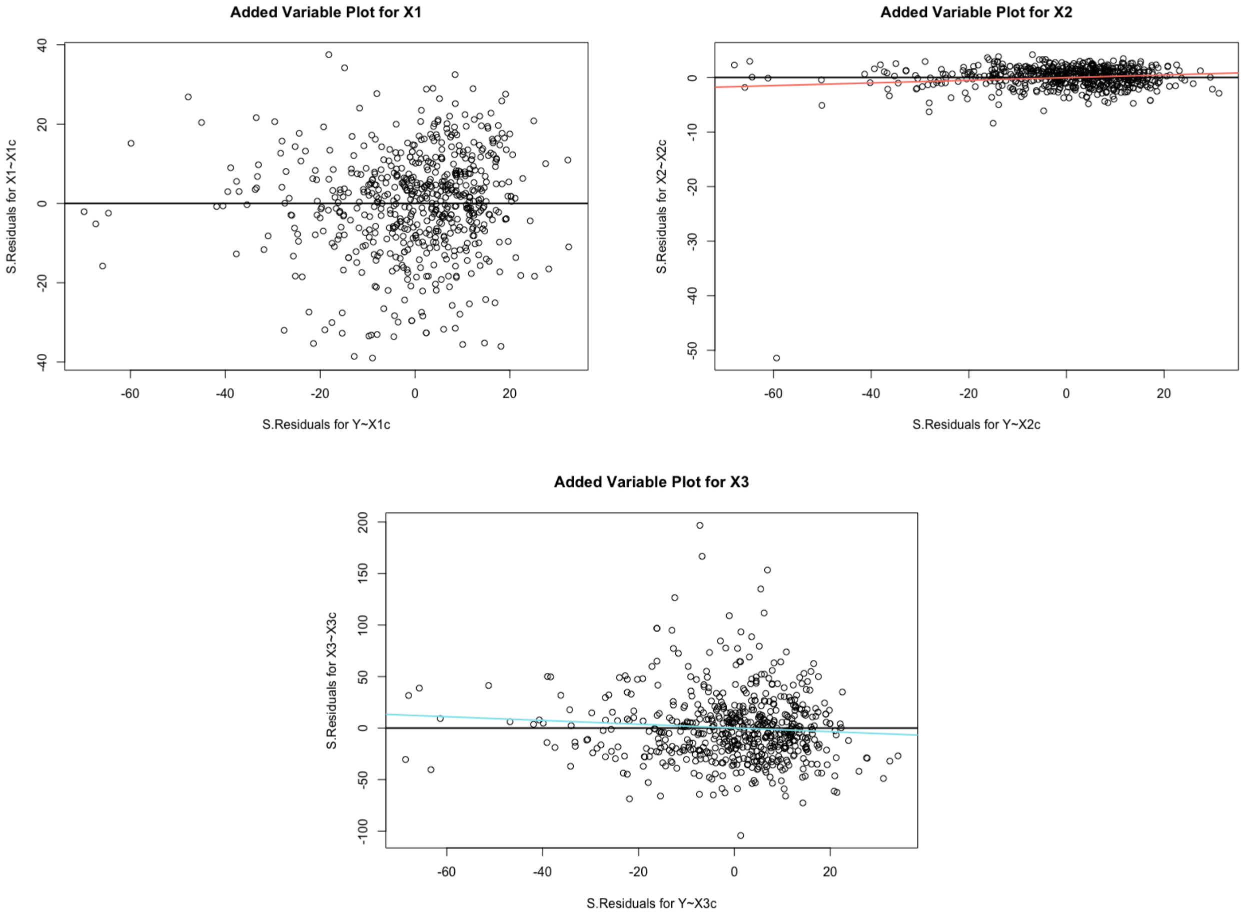
Variance Inflation Factor for Each Predictor Variable

Predictor Variable	Variance Inflation Factor
X1	1.0335
X2	0.0207
X3	0.0313

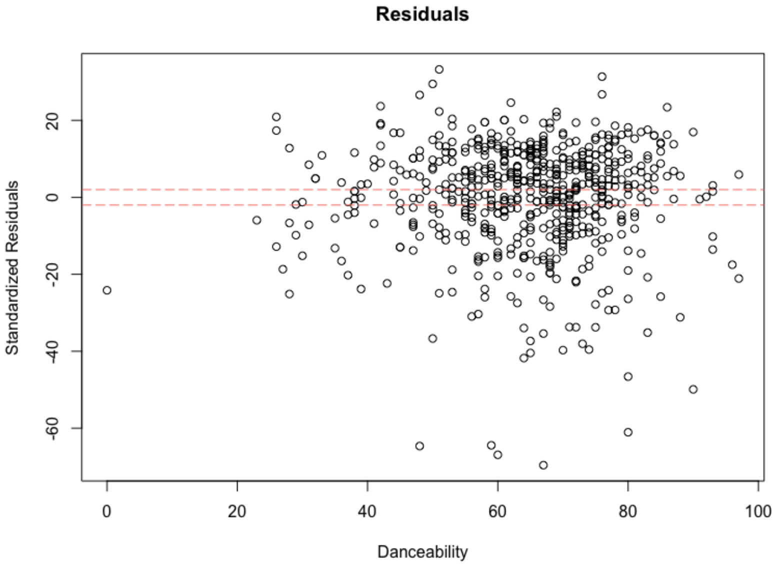
Table 3.2



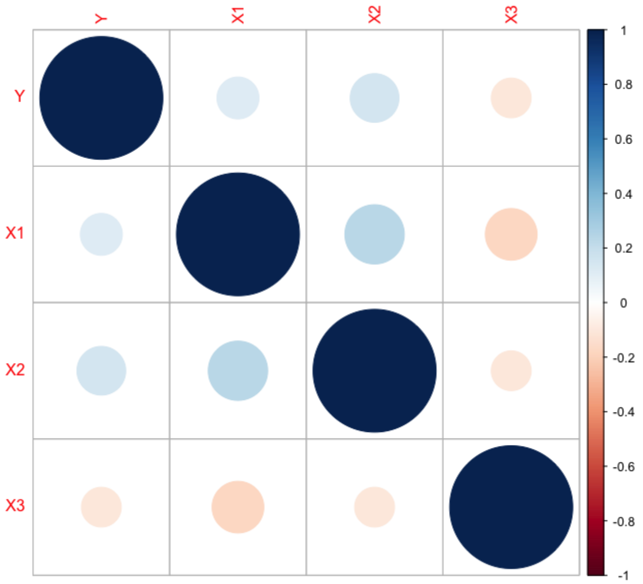
Graph 3.2 Variance inflation factor barplot for each predictor variable



Graph 3.3 Added variable plots for each predictor variable



Graph 3.4 Standardized residuals with respect to X1, the predictor variable for danceability



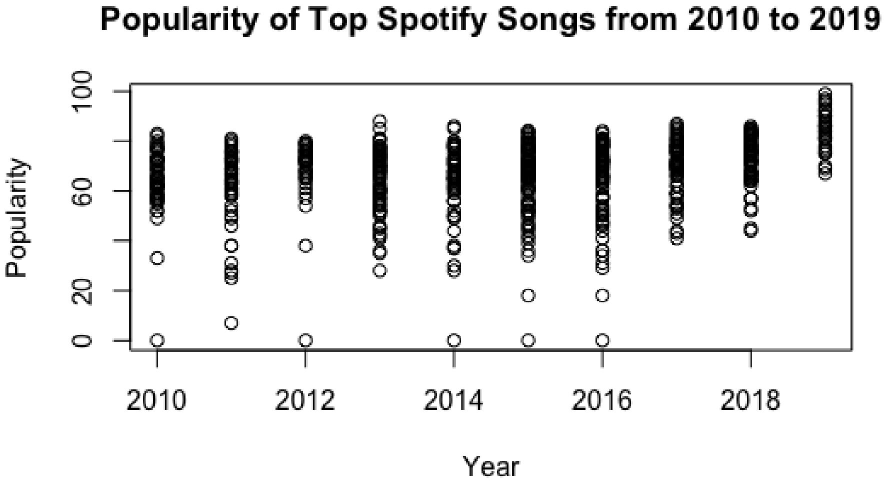
Graph 3.5 Correlation matrix between Y, X1,X2, and X3

According to the p-value for an F-test comparing H_0: $\beta_1 = \beta_2 = \beta_3 = 0$ verses H_0: At least one β_is $\neq 0$, the at least one of our beta coefficients is linearly associated with the predictor variable since 0.00005 is less than any conventional level of α. Upon viewing the correlation matrix between Y and all three variables, it gives rise to a concern of multicollinearity between X1 and X2 and X1 and X3 since the correlation between these variables is relatively high. Thus, the relationships between these variables were more closely investigated through their respective variance inflation factors. It was concluded multicollinearity does not appear to be an issue since the variance inflation factor for each of the predictor variables does not exceed a value 5, which is typically regarded as the conventional threshold for multicollinearity.

While the model deceptively seems to be a good fit for the data based on the p-value for the F-test and the lack of multicollinearity, the standardized residual plot with respect to X1 and the added variable plots suggest otherwise. Although there appears to be a random pattern in the standardized residual plot, the points are not uniformly distributed and there appears to be several points whose standardized residual falls outside of the interval (-2, 2), indicating that the constant variance assumption of a linear model is violated and there are a lot of outliers in the dataset according to the model. Furthermore, the predictive power of our model is weak as the magnitude of the beta coefficients are small both according to their estimates and their added variable plots. More specifically, the contribution of X1 to the model, adjusting for the added effects of X2 and X3, is nearly 0 given by the slope of the line of the added variable plot in the upper left corner. X2 and X3 contribute slightly more than X1 to the model after adjusting for it and X2 and X3 respectively which is given by the slope of the line of the added variable plots in the upper right corner and bottom. However, their lines nearly resemble a straight, horizontal line, meaning that their slopes are also close to 0, and thus, have weak predictive power.

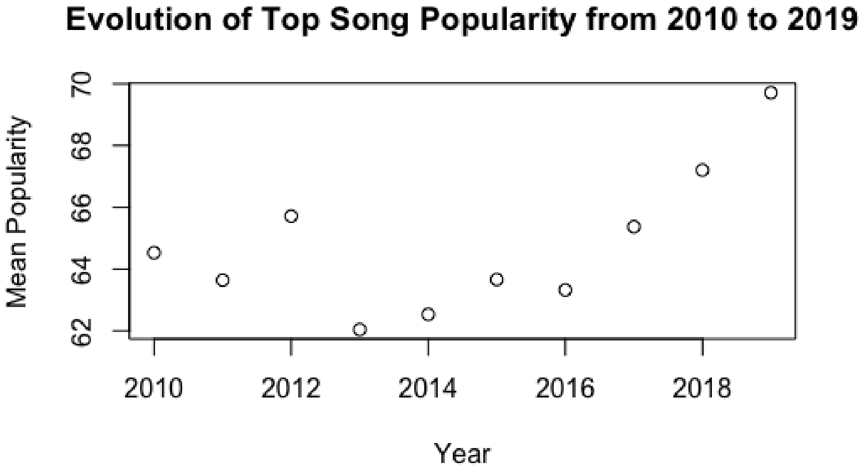
Therefore, the model has no utility as the constant variance assumption of a linear model has been violated and the model is an overall bad fit to the data since it suggests that there are a lot of outliers in the dataset and the predictor variables do not contribute much to the predictive power of the model. Since there is no evidence of multicollinearity between the predictor variables, standardizing them would not improve the model. Rather, a transformation of Y and/or the other variables should be applied.

IV. Time Series Fundamentals



Graph 4.1 Plot of the popularity of top Spotify songs from 2010 to 2019

It is difficult to perceive a trend in the popularity of top Spotify songs from 2010 to 2019 based on Graph 4.1. In order to make the trend appear more obvious, the average popularity value of top Spotify songs was calculated for each year and plotted in the following graph.



Graph 4.2 Plot depicting the evolution of the mean popularity of top Spotify songs

It appears that there is a generally increasing trend of the mean popularity of songs from 2010 to 2019. However, it does not appear to be affected by seasonality because there is only one local maximum and minimum point in the scatterplot. Consequently, it is reasonable to assume that the stationarity of the data is an irrelevant property to consider since there is no periodical behavior in the graph. Rather, it appears that a multiple cubic polynomial model might be a good fit for the data.

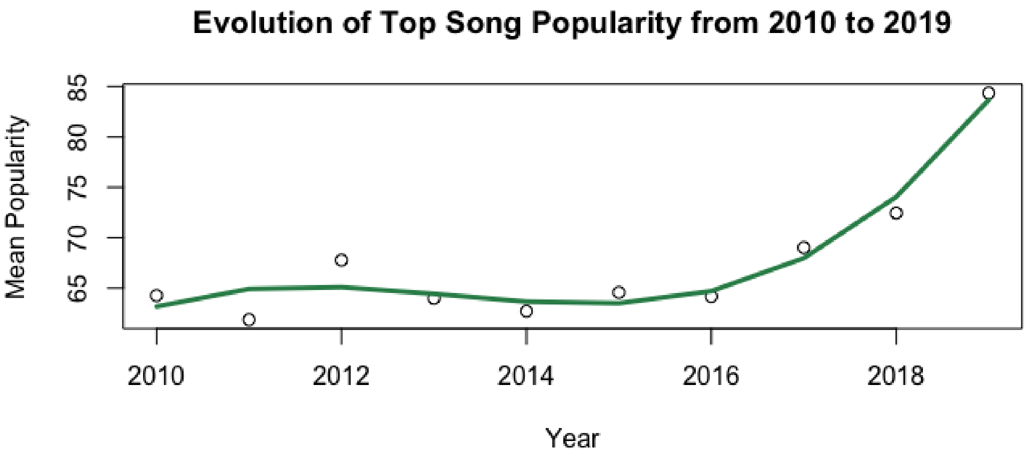
Let Y be the mean popularity of top Spotify songs from 2010-2019
Let X1 be the year associated with the mean popularity of the top Spotify songs

Proposed Model: $\hat{Y} = b + \beta_1 X_1 + \beta_2 X_1^2 + \beta_3 X_1^3$

However, because interaction terms are introduced to the model as a result of the polynomial terms, structural multicollinearity becomes a problem. Thus, the original predictor variable for year, X1, must be standardized in order to eliminate this issue.

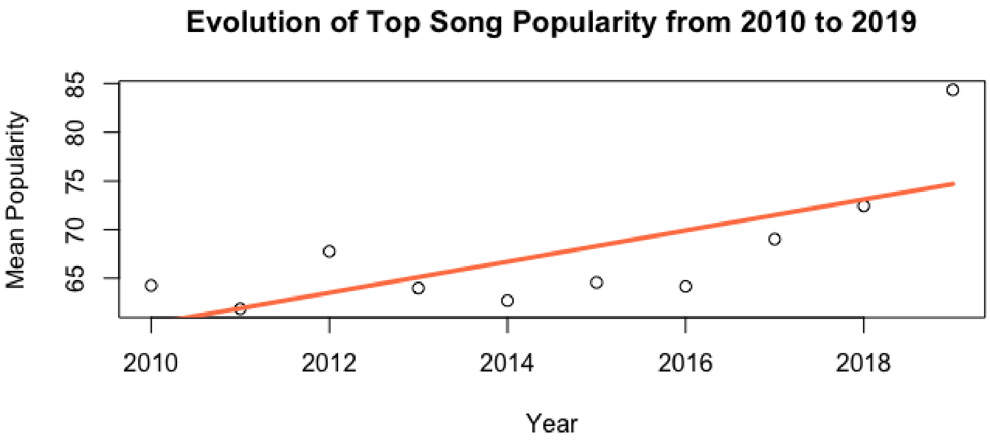
Let $Z_1 = \frac{(X_1 - \overline{X_1})}{\partial X_1}$

$$\hat{Y} = 67.5126 + 4474166.1734Z1 - 954339.04816Z1^2 + 4480179.4791Z1^3$$



Graph 4.3 Scatterplot with fitted multiple cubic polynomial regression model

$$\hat{Y} = -3149.5520 + 1.5970X$$



Graph 4.4 Scatterplot with fitted linear regression model

It is clear that the multiple cubic polynomial regression model fits the data better compared to the linear regression model. Furthermore, the beta coefficients associated with the standardized predictor variables in the polynomial regression model are large, suggesting that they are contributing greatly to the predictive power of the model whereas the beta coefficients associated with the predictor variable in the linear regression model is small, suggesting that it is not contributing much to the predictive power of the model. It would be interesting to see how well the polynomial regression model fits new data since overfitting is a potential problem.

V. Code

Section I.A

Table 1.1

```
# dataset
#https://www.kaggle.com/datasets/leonardopena/top-spotify-songs-from-20102019-by-year/code?select=top10s.csv

 head(top10s)
```

Table 1.2

```
 n <- length(top10s$dnce)
# 603
alpha <- 0.05
df <- n - 1
# 602

# Generating t-critical value
tcrit <- qt(alpha/2, df, lower.tail = F)
# 1.964

# Generating sample error
sx <- sd(top10s$dnce)
# 13.379
SE <- sx/sqrt(n)
# 0.545

# One sample hypothesis test for the pop. mean
# Based on logic, popular songs should have a high danceability
mu0 <- 70 #claimed value of the mean
xbar <- mean(top10s$dnce)
# 64.380

# Test statistic
tstat <- (xbar-mu0)/SE
# -10.316

# Pvalue
pvalue <- 2*pt(-abs(tstat), df, lower.tail=T)
```

```
# 4.384

# Margin of Error
eps <- tcrit*SE
# 1.070

# Confidence Interval
CIL <- xbar - eps
CIU <- xbar + eps
# (63.310, 65.450)

metric_name = c("CI.Lower", "CI.Upper", "Claimed.Value", "T.stat",
"T.crit",
                "pvalue", "alpha")
metric_value = c(CIL, CIU, mu0, tstat, tcrit, pvalue, alpha)

DataSummary <- data.frame(metric_name, metric_value)
```

Graph 1.1

```
# Distribution of Danceability Values
hist(top10s$dnce, col='orchid', main = 'Distribution of Danceability
Values', xlab = 'Danceability')
```

Section I.B

Table 1.3

```
# One sample hypothesis test for the pop. standard deviation
sig0 <- 15 # claimed value of the standard deviation

# Chi-squared test statistic for pop. standard deviation
cstat <- sqrt(((n-1)*sx^2/sig0^2))


# Lower and upper critical values for pop. standard deviation
ccritL <- sqrt(qchisq(alpha/2, df, lower.tail = T))
ccritU <- sqrt(qchisq(alpha/2, df, lower.tail = F))

metric_name1 <- c("Lower Tail Bound","Upper Tail Bound", "sig0",
"cstat", "alpha")
```

```
metric_value1 <- c(ccritL, ccritU, sig0, cstat, alpha)

DataSummary1 <- data.frame(metric_name1, metric_value1)
```

Section I.C

Graph 1.2

```
# Checking the normality of BPM and Energy distributions
qqnorm(top10s$bpm, ylab = "Quantiles of BPM", main = "NQQ Plot for the
Beats Per Minute of Top Spotify Songs")
qqline(top10s$bpm, col = "orchid", lwd = 3)
```

Graph 1.3

```
qqnorm(top10s$nrgy, ylab = "Quantiles of Energy", main = "NQQ Plot for
the Energy of Top Spotify Songs")
qqline(top10s$nrgy, col = "salmon", lwd = 3)
```

Table 1.4

```
# Correlation coefficients associated with NQQ plots
V <- qqnorm(top10s$bpm, plot=FALSE)
cor(V$x,V$y)
U <- qqnorm(top10s$nrgy, plot=FALSE)
cor(U$x, U$y)
```

Section I.D

Graph 1.4

```
# EDA - checking the frequency of genres
table(top10s$top.genre)
# Subsetting dataset based on genres
pop <- top10s[top10s$top.genre == 'pop',]
dancepop <- top10s[top10s$top.genre == 'dance pop',]
# Checking the distribution of bpm for pop songs and dance songs
boxplot(pop$bpm, dancepop$bpm,
        names = c('Pop', 'Dance Pop'),
        ylab = 'Beats per Minute (bpm)', col = c('lightpink',
'steelblue1'),
        horizontal = TRUE)
```

Table 1.5

```
# ---------H. Test for the Difference in Pop. Means----------
```

```
alpha1 <- 0.01

xbar1 <- mean(pop$bpm) # mean bpm of pop songs
xbar2 <- mean(dancepop$bpm) # mean bpm of dance songs
sd1 <- sd(pop$bpm)
sd2 <- sd(dancepop$bpm)
n1 <- length(pop$bpm)
n2 <- length(dancepop$bpm)

# Welch t-statistic
xbard <- xbar1 - xbar2 # difference in
wstat <- xbard/sqrt(sd1^2/n1 + sd2^2/n2)

# degrees of freedom
numerator <- ((sd1^2/n1 + sd2^2/n2)^2)
denominator <- (sd1^4/(n1^2*(n1-1)) + sd2^4/(n2^2*(n2-1)))
df <- numerator/denominator

SE1 <- sqrt((sd1^2/n1)+(sd2^2/n2))
wcrit <- qt(alpha1/2, df, lower.tail = F)
meps1 <- wcrit*SE1
pval1 <- 2*pt(-abs(wstat), df = 77.74)

metric_name2 <- c("CI.Lower", "CI.Upper", "T-stat", "T-crit",
"pvalue", "alpha")
metric_value2 <- c(xbard - meps1, xbard + meps1, wstat, wcrit, pval1,
alpha1)
DataSummary2 <- data.frame(metric_name2, metric_value2)
print(DataSummary2)
```

Section I.E

Table 1.6

```
# ---------H.Test 2 for the Difference in Pop. Variances--------
alpha2 <- 0.01

fstat <- sd1^2/sd2^2
fcritL <- qf(alpha2/2, df1 = n1-1, df2 = n2-1, lower.tail = T)
fcritR <- qf(alpha2/2, df1 = n1-1, df2 = n2 -1, lower.tail = F)
```

```
fstatL <- min(fstat, 1/fstat)
fstatR <- max(fstat, 1/fstat)

pval2 <- pf(fstatL, df1 = n1-1, df2 = n2-1, lower.tail = T) +
  pf(fstatR, df1 = n1-1, df2 = n2-1, lower.tail = F)

metric_name3 <- c("CI_Lower", "CI_Upper", "FstatL", "FstatU",
                  "FcritL", "FcritU", "pval", "alpha")
metric_value3 <- c(fcritL*fstat, fcritR*fstat, fstatL, fstatR,
                   fcritL, fcritR, pval2, alpha2)
DataSummary3 <- data.frame(metric_name3, metric_value3)
```

Section II.A

Graph 2.1

```
# Scatterplot
plot(x = top10s$dnce,
     y = top10s$pop,
     main = 'Popularity vs Danceability',
     xlab ='X',
     ylab='Y')

# SLR model
M1 <- xufunction(Sy,as.matrix(Sx))

n <- dim(top10s)[1]
Sx <- top10s[1:n,8] #Danceability
Sy <- top10s[1:n,15] #Popularity

# Beta coefficients
betahats <- M1$betahat

lines(Sx, M1$predicted, lwd = 3, col = 'lightblue')
```

Graph 2.2

```
# ANOVA
SSE <- M1$SSE
MSE <- M1$MSE
SST <- M1$SST
MST <- M1$MST
```

```
SSM <- M1$SSM
MSM <- M1$MSM

std.resid <- M1$`std. residual`

# Standardized residual plot
par(mfrow = c(1,1))
plot(Yhat,
     std.resid,
     xlab = "Predicted Values for Popularity",
     ylab = "Standardized Residuals",
     main = "Residuals")
abline(c(-2,2), col='salmon',lty='longdash',lwd=2 )
abline(-2,0,col='salmon',lty='longdash')
abline(2,0,col='salmon',lty='longdash')
```

Table 2.1

```
# Beta coefficients
betahats <- M1$betahat

# Standard errors
SE.betahat <- M1$SEbetahat
```

Table 2.2

```
# r-squared and r-squared adj
r2 <- M1$rsquared
r2adj <- M1$rsquaredadj

# H0: B1 = 0 vs H0 != 0
pvalue <- M1$`p-value`
```

<u>Section II.B</u>

Graph 2.3

```
# Scatterplot
plot(top10s$bpm, top10s$dnce, xlab = "Beats per Minute", ylab =
"Danceability", main = "Danceability vs BPM")

# Fitting the simple linear regression model
bpm <- as.matrix(top10s$bpm)
```

```
dnce <- top10s$dnce
M2 <- xufunction(dnce, bpm^2)
```

Graph 2.4

```
# Standardized residual plot
plot(M2$predicted, M2$`std. residual`,
     xlab = 'Predicted Values for Danceability',
     ylab = 'Standardized Residuals',
     main = 'Residuals')
abline(c(-2,2), col='salmon',lty='longdash',lwd=2 )
abline(-2,0,col='salmon',lty='longdash')
abline(2,0,col='salmon',lty='longdash')
```

Table 2.3

```
# Fitting the simple quadratic regression model
betahat2 <- M2$betahat
# options(scipen = 999)
SE.betahat2 <- M2$SEbetahat
```

Table 2.4

```
r2.2 <- M2$rsquared
r2adj2 <- M2$rsquaredadj
pval2 <- M2$`p-value`
```

Section III

Graph 3.1

```
# Correlation matrix between Y and all the quantitative variables
data <- data.matrix(top10s[,5:14])
R <- cor(data)
corrplot(R, method='circle')

# Fitting a multiple linear regression model
Y <- top10s$pop # popularity
X1 <- top10s$dnce # danceability
X2 <- top10s$dB # decibels
X3 <- top10s$dur # duration

M3 <- xufunction(Y, cbind(X1,X2,X3))
```

```
# Example of prediction
# X1 = 60, X2 = -8, X3 = 170
M3$betahat[1] + M3$betahat[2]*60 + M3$betahat[3]*-8 +
M3$betahat[4]*170
```

Table 3.1

```
# ANOVA
SSE <- M3$SSE
SSM <- M3$SSM
SST <- M3$SST

MSE <- M3$MSE
MSM <- M3$MSM
MST <- M3$MST

Fstat <- MSM/MSE
pval <- M3$`p-value`
```

Table 3.2

```
# Multcolinearity
MYvX1c <- xufunction(Y, cbind(X2,X3))
MX1vX1c <- xufunction(X1, cbind(X2,X3))

MYvX2c <- xufunction(Y, cbind(X1,X3))
MX2vX2c <- xufunction(X2, cbind(X1,X3))

MYvX3c <-xufunction(Y, cbind(X1,X2))
MX3vX3c <- xufunction(X3, cbind(X1,X2))

# Variation Inflation Factors
vif1 <- 1/(1-MYvX1c$rsquared)
vif2 <- 1/(1/MYvX2c$rsquared)
vif3 <- 1/(1/MYvX3c$rsquared)
```

Graph 3.2

```
# Variance inflation factor barplot for each predictor variable
vif <- c(vif1, vif2, vif3)
barplot(vif, horiz = T, main = 'Variance Inflation Factors',
        names.arg = c('X1', 'X2', 'X3'),
        xlim = c(0, max(6, max(vif))))
abline(v=5, col='coral', lty = 'longdash')
```

Graph 3.3

```
# Added Variable Plots
plot(MYvX1c$residual, MX1vX1c$residual,
     main = 'Added Variable Plot for X1',
     xlab = 'S.Residuals for Y~X1c',
     ylab = 'S.Residuals for X1~X1c')
abline(0,0,lwd=2)
abline(mean(MX1vX1c$residual)-cor(MYvX1c$residual,
MX1vX1c$residual)*sd(MX1vX1c$residual)/sd(MYvX1c$residual),
       cor(MYvX1c$residual,
MX1vX1c$residual)*sd(MX1vX1c2residual)/sd(MYvX1c$residual), col =
'seagreen', lwd = 2)

plot(MYvX2c$residual, MX2vX2c$residual,
     main = 'Added Variable Plot for X2',
     xlab = 'S.Residuals for Y~X2c',
     ylab = 'S.Residuals for X2~X2c')
abline(0,0,lwd=2)
abline(mean(MX2vX2c$residual)-cor(MYvX2c$residual,
MX2vX2c$residual)*sd(MX2vX2c$residual)/sd(MYvX2c$residual),
       cor(MYvX2c$residual,
MX2vX2c$residual)*sd(MX2vX2c$residual)/sd(MYvX2c$residual), col =
'salmon', lwd = 2)

plot(MYvX3c$residual, MX3vX3c$residual,
     main = 'Added Variable Plot for X3',
     xlab = 'S.Residuals for Y~X3c',
     ylab = 'S.Residuals for X3~X3c')
abline(0,0,lwd=2)
abline(mean(MX3vX3c$residual)-cor(MYvX3c$residual,MX3vX3c$residual)*sd
(MX3vX3c$residual)/sd(MYvX3c$residual),
       cor(MYvX3c$residual,
MX3vX3c$residual)*sd(MX3vX3c$residual)/sd(MYvX3c$residual), col =
'cadetblue2', lwd = 2)
```

Graph 3.4

```
# Standardized residual plot with respect to X1
plot(X1,M3$`std. residual`,
     main = 'Residuals',
     xlab='Danceability',
     ylab='Standardized Residuals')
abline(-2,0,col='salmon',lty='longdash')
abline(2,0,col='salmon',lty='longdash')
```

Graph 3.5

```
# Correlation matrix between Y, X1, X2, X3
data1 <- cbind(Y, X1, X2, X3)

R1 <- cor(data1)
corrplot(R1, method='circle')
```

<u>Section IV</u>

Graph 4.1

```
# Plot of the popularity of top Spotify songs from 2010 to 2019
plot(top10s$year, top10s$pop, main = 'Popularity of Top Spotify Songs
from 2010 to 2019',
     xlab = 'Year', ylab = 'Popularity')

# Calculating the average popularity of top Spotify songs for each
year
popbar <- top10s %>%  group_by(year) %>% summarise_at(vars(pop),
list(name=mean))
```

Graph 4.2

```
# Evolution of the mean popularity of top Spotify songs
plot(popbar$year,popbar$name,
     main='Evolution of Top Song Popularity from 2010 to 2019',
     xlab = 'Year',
     ylab = 'Mean Popularity')
```

Graph 4.3

```
# Multiple Polynomial Regression Model
Y <- popbar$name
X1 <- popbar$year
```

```
Z1 <- (X1 - mean(X1))/sd(X1) # Standardizing predictor variable
Z2 <- (X1^2 - mean(X1^2))/sd(X1^2)
Z3 <- (X1^3 - mean(X1^3))/sd(X1^3)

M5 <- xufunction(Y, cbind(Z1, Z2, Z3),0) # fitting the model
lines(X1, M5$predicted, lwd = 3, col = 'seagreen')
```

Graph 4.4

```
# Linear regression model
plot(popbar$year,popbar$name,
     main='Evolution of Top Song Popularity from 2010 to 2019',
     xlab = 'Year',
     ylab = 'Mean Popularity')

M6 <- xufunction(Y, as.matrix(X1)) # fitting the model
lines(X1, M6$predicted, lwd = 3, col = 'coral')

# Helper functions
xufunction <- function(Y, Xk){
  # Christina Xu, 4/1/2, MA 575 Lab 08
  n <- length(Y)
  v1s <- rep(1,n)
  X <- cbind(v1s, Xk)
  S <- svd(t(X)%*%X)
  U <- S$u
  V <- S$v
  D <- diag(S$d)

  detX <- sqrt(abs(det(t(X)%*%X)))
  # if det=0, cols and rows are colinear
  kappa <- sqrt(max(S$d)/min(S$d))
  # condition large, model is unstable

  betahat <- V%*%solve(D)%*%t(U)%*%t(X)%*%Y
  H <- X%*%V%*%solve(D)%*%t(U)%*%t(X)
  lv <- diag(H)
  Yhat <- X%*%betahat
  resid <- Y - Yhat
  SSE <- sum((resid)^2)
  p <- dim(Xk)[2]
  SST <- sd(Y)^2*(n-1)
  SSM <- SST - SSE
  # SST is not always equal SSM and SSE

  MST <- SST/(n-1)
  MSE <- SSE/(n-p-1)
  MSM <- SSM/p

  SEbetahat <- sqrt(MSE)*sqrt(diag(V%*%solve(D)%*%t(U)))
  sresid <- resid/ (sqrt(MSE)*sqrt(1-lv))
  r2 <- 1 - SSE/SST
  r2adj <- 1 - MSE/MST

  Fstat <- MSM/MSE
  pval <- pf(Fstat, p, n-p-1, lower.tail = F)


  result <- list('predicted'= Yhat,
                 'residual' = resid,
                 'std. residual' = resid,
                 'condtion' = kappa,
                 'determinant' = detX,
                 'leverage' = lv,
                 'SSE' = SSE,
                 'SSM' = SSM,
                 'SST' = SST,
                 'MSE' = MSE,
                 'MSM' = MSM,
                 'MST' = MST,
                 'p-value' = pval,
                 'betahat' = betahat,
                 'SEbetahat' = SEbetahat,
                 'rsquared' = r2,
                 'rsquaredadj' = r2adj)
  return(result)}
```