Analyzing Musical Feature Trends in Top Spotify Songs (2010–2019): A Statistical Overview

Jonah Sitorus

Dataset provided on Canvas (ProjectPhase2_ExcelDataFile_Spotify)

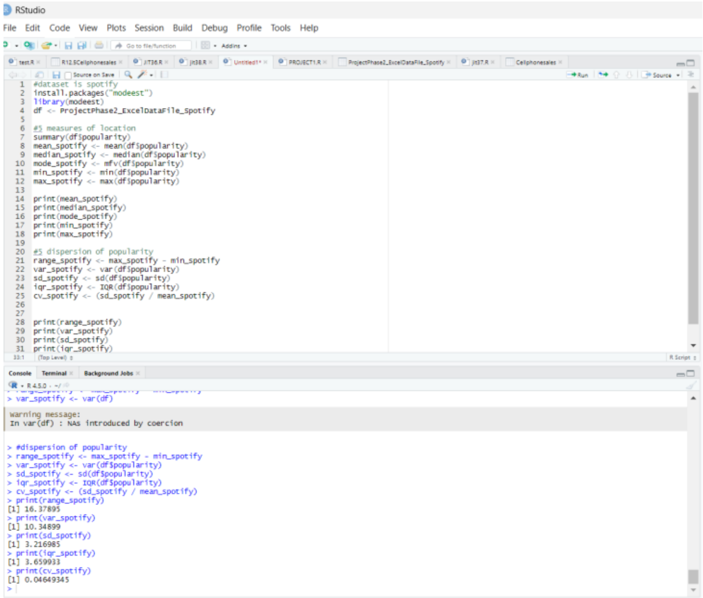
Executive Summary

In this project, I analyzed data on top Spotify songs from 2010-2019 to understand what makes a song popular. The dataset includes variables on musical features such as tempo (bpm), danceability, valence (mood), duration, year, and genre, along with a popularity score from 0 to 100. Understanding these relationships helps with identifying trends in listener preferences and better support playlist and marketing decisions.

Using R, I summarize the distribution of song popularity with descriptive statistics. I explore relationships between popularity and numerical variables using scatterplots and correlations. I built a multiple regression model with numerical predictors to explain variation in popularity and test model assumptions through residual analysis. I then extended the model by adding a categorical variable for the dance pop genre to see if this genre is associated with higher and lower popularity after adjusting for other musical attributes.

Overall, the results indicate that danceability, valence, and duration are statistically significant predictors of popularity, while year and tempo are not significant in this dataset. Residual models suggested that the assumptions were reasonably met and support the validity of this model. These findings provide an insight into the attributes that are in a top-charting Spotify song and highlights which feature is most popular and what is most geared towards listener preference.

Descriptive Statistics for Popularity

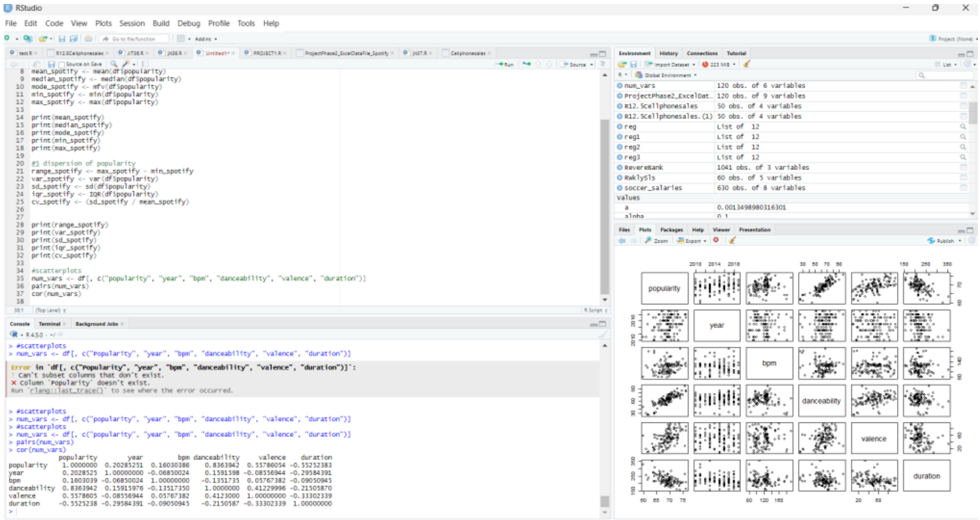


```r
#dataset is spotify
install.packages("modeest")
library(modeest)
df <- ProjectPhase2_ExcelDataFile_Spotify

#5 measures of location
summary(df$popularity)
mean_spotify <- mean(df$popularity)
median_spotify <- median(df$popularity)
mode_spotify <- mfv(df$popularity)
min_spotify <- min(df$popularity)
max_spotify <- max(df$popularity)

print(mean_spotify)
print(median_spotify)
print(mode_spotify)
print(min_spotify)
print(max_spotify)

#5 dispersion of popularity
range_spotify <- max_spotify - min_spotify
var_spotify <- var(df$popularity)
sd_spotify <- sd(df$popularity)
iqr_spotify <- IQR(df$popularity)
cv_spotify <- (sd_spotify / mean_spotify)


print(range_spotify)
print(var_spotify)
print(sd_spotify)
print(iqr_spotify)
```

```
> range_spotify <- max_spotify - min_spotify
> var_spotify <- var(df)

Warning message:
In var(df) : NAs introduced by coercion

> #dispersion of popularity
> range_spotify <- max_spotify - min_spotify
> var_spotify <- var(df$popularity)
> sd_spotify <- sd(df$popularity)
> iqr_spotify <- IQR(df$popularity)
> cv_spotify <- (sd_spotify / mean_spotify)
> print(range_spotify)
[1] 16.37895
> print(var_spotify)
[1] 10.34899
> print(sd_spotify)
[1] 3.216985
> print(iqr_spotify)
[1] 3.659933
> print(cv_spotify)
[1] 0.04649345
> 
```

In this section, the descriptive statistics for the dependent variable, Popularity. The mean popularity was approximately 69.19222 with a median of 69.41. The minimum and maximum values were 59.90868 and 76.28763. The standard deviation was 3.216985 which indicates that the popularity scores were not spread out that much. Because the distribution is skewed to the right this means that the mean is greater than the median and the best measure of the center is the median because it's not heavily influenced by the high-value outliers in the right tail.

Scatterplots and Correlation



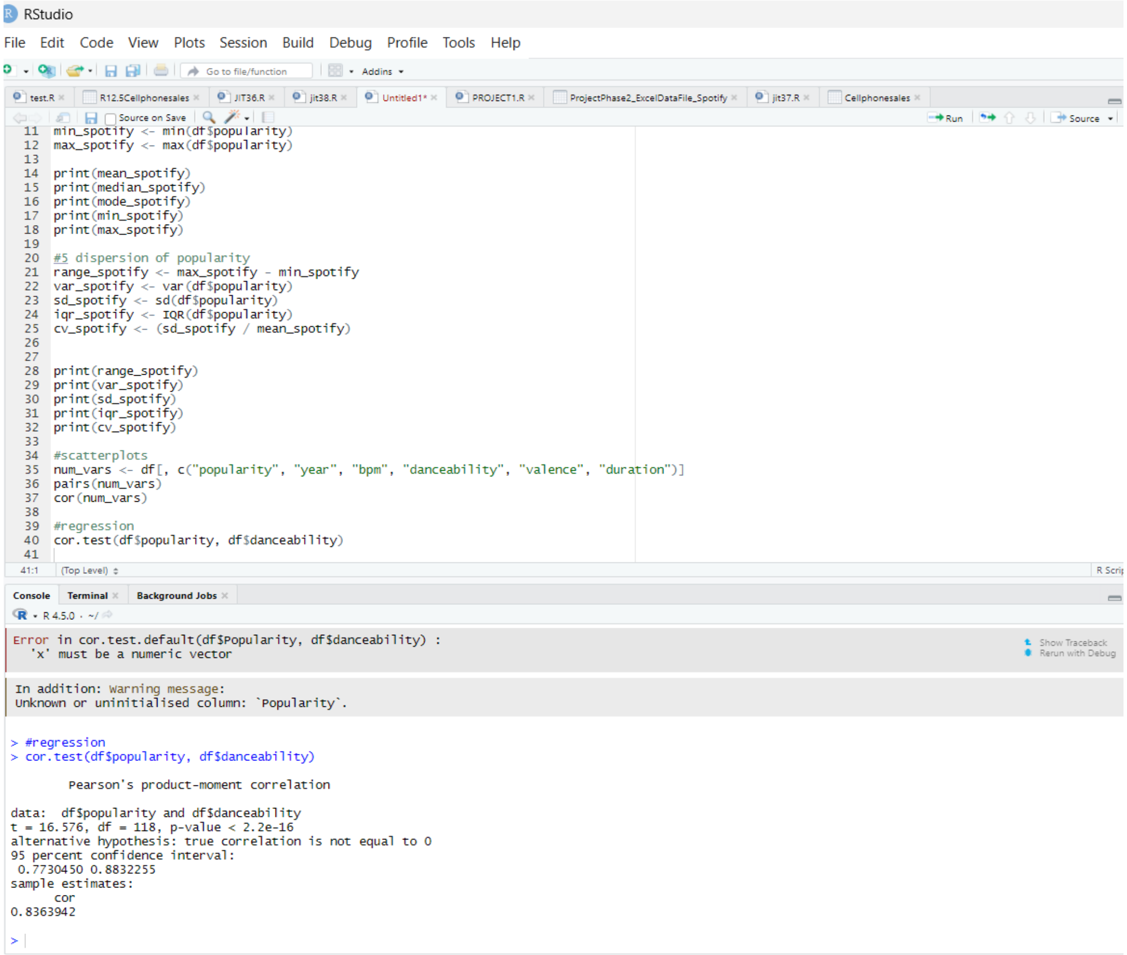
```r
mean_spotify <- mean(df$popularity)
median_spotify <- median(df$popularity)
mode_spotify <- mfv(df$popularity)
min_spotify <- min(df$popularity)
max_spotify <- max(df$popularity)

print(mean_spotify)
print(median_spotify)
print(mode_spotify)
print(min_spotify)
print(max_spotify)

#5 dispersion of popularity
range_spotify <- max_spotify - min_spotify
var_spotify <- var(df$popularity)
sd_spotify <- sd(df$popularity)
iqr_spotify <- IQR(df$popularity)
cv_spotify <- (sd_spotify / mean_spotify)


print(range_spotify)
print(var_spotify)
print(sd_spotify)
print(iqr_spotify)
print(cv_spotify)

#scatterplots
num_vars <- df[, c("popularity", "year", "bpm", "danceability", "valence", "duration")]
pairs(num_vars)
cor(num_vars)

```

```
> #scatterplots
> num_vars <- df[, c("Popularity", "year", "bpm", "danceability", "valence", "duration")]

Error in `df[, c("Popularity", "year", "bpm", "danceability", "valence", "duration")]`:
! Can't subset columns that don't exist.
✖ Column `Popularity` doesn't exist.
Run `rlang::last_trace()` to see where the error occurred.

> #scatterplots
> num_vars <- df[, c("popularity", "year", "bpm", "danceability", "valence", "duration")]
> #scatterplots
> num_vars <- df[, c("popularity", "year", "bpm", "danceability", "valence", "duration")]
> pairs(num_vars)
> cor(num_vars)
             popularity        year         bpm danceability     valence    duration
popularity    1.0000000  0.20285251  0.16030386    0.8363942  0.55786054 -0.55252383
year          0.2028525  1.00000000 -0.06850024    0.1591598 -0.08556944 -0.29584391
bpm           0.1603039 -0.06850024  1.00000000   -0.1351735  0.05767382 -0.09050945
danceability  0.8363942  0.15915976 -0.13517350    1.0000000  0.41229996 -0.21505870
valence       0.5578605 -0.08556944  0.05767382    0.4123000  1.00000000 -0.33302339
duration     -0.5525238 -0.29584391 -0.09050945   -0.2150587 -0.33302339  1.00000000
> 
```

```
num_vars                      120 obs. of 6 variables
ProjectPhase2_ExcelDat…       120 obs. of 9 variables
R12.5Cellphonesales           50 obs. of 4 variables
R12.5Cellphonesales.(1)       50 obs. of 4 variables
reg                           List of  12
reg1                          List of  12
reg2                          List of  12
reg3                          List of  12
RevereBank                    1041 obs. of 3 variables
RwklySls                      60 obs. of 5 variables
soccer_salaries               630 obs. of 8 variables
Values
  a                           0.0013498980316301
  alpha                       0.1
```

h

The scatterplot of Popularity versus Danceability appears to show a roughly linear pattern and the direction is positive, meaning that as danceability increases, popularity also increases. The strength of the relationships appears strong based on how the points cluster around a trend and that the correlation level is high at .836 and there's no obvious outliers.



```
min_spotify <- min(df$popularity)
max_spotify <- max(df$popularity)

print(mean_spotify)
print(median_spotify)
print(mode_spotify)
print(min_spotify)
print(max_spotify)

#5 dispersion of popularity
range_spotify <- max_spotify - min_spotify
var_spotify <- var(df$popularity)
sd_spotify <- sd(df$popularity)
iqr_spotify <- IQR(df$popularity)
cv_spotify <- (sd_spotify / mean_spotify)


print(range_spotify)
print(var_spotify)
print(sd_spotify)
print(iqr_spotify)
print(cv_spotify)

#scatterplots
num_vars <- df[, c("popularity", "year", "bpm", "danceability", "valence", "duration")]
pairs(num_vars)
cor(num_vars)

#regression
cor.test(df$popularity, df$danceability)

```

```
Error in cor.test.default(df$Popularity, df$danceability) :
  'x' must be a numeric vector

In addition: Warning message:
Unknown or uninitialised column: `Popularity`.

> #regression
> cor.test(df$popularity, df$danceability)

        Pearson's product-moment correlation

data:  df$popularity and df$danceability
t = 16.576, df = 118, p-value < 2.2e-16
alternative hypothesis: true correlation is not equal to 0
95 percent confidence interval:
 0.7730450 0.8832255
sample estimates:
      cor
0.8363942

>
```

Test significance of correlation (Popularity & Danceability)

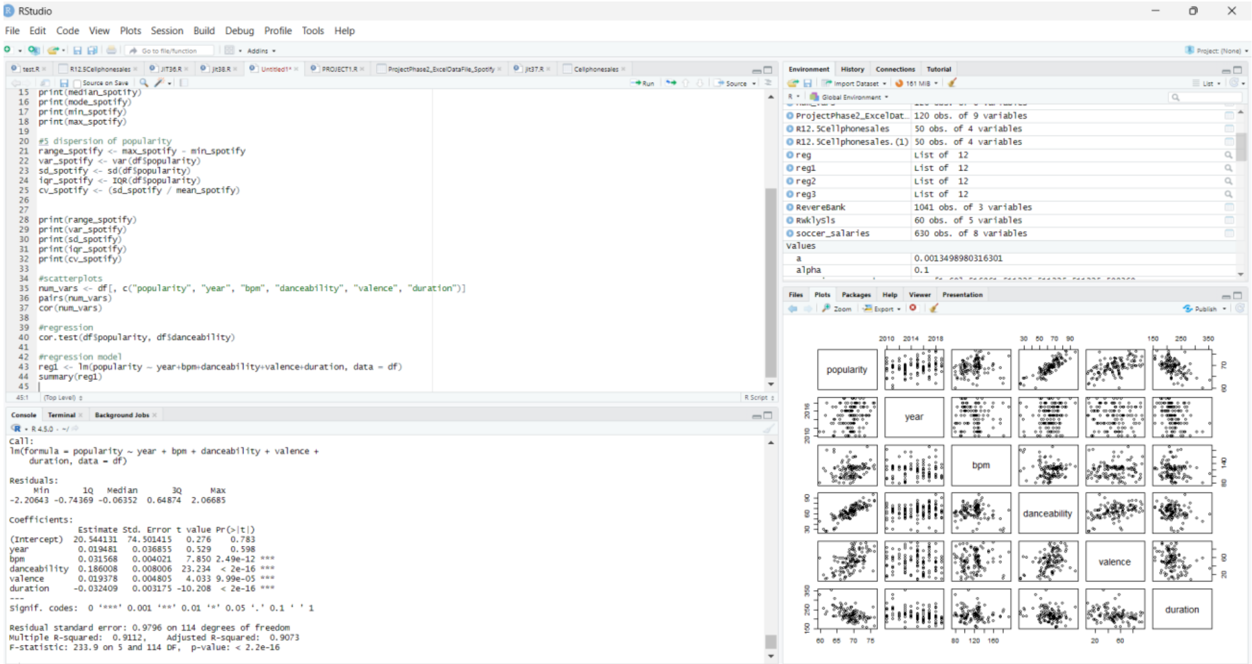
- **H_0:** ρ = 0 (no linear correlation)
- **H_a:** ρ ≠ 0 (non-zero linear correlation)

The correlation between popularity and danceability is .836. Using cor.test(), we obtain a p-value of 2.2e^-16 which at the .05 significance level, we reject the null hypothesis of zero correlation. This indicates that there is a statistically significant linear association between popularity and danceability.

Multicollinearity

For multicollinearity among the independent variables, I used the correlation matrix and pairs with correlations below 0.2 were classified as low, 0.2-0.8 as moderate and above 0.8 as extreme. In this dataset, popularity and year, popularity and bpm, popularity and duration had low multicollinearity, and popularity and valence had moderate multicollinearity and popularity and danceability and extreme multicollinearity.

Multiple Regression Model



```
print(median_spotify)
print(mode_spotify)
print(min_spotify)
print(max_spotify)

#5 dispersion of popularity
range_spotify <- max_spotify - min_spotify
var_spotify <- var(df$popularity)
sd_spotify <- sd(df$popularity)
iqr_spotify <- IQR(df$popularity)
cv_spotify <- (sd_spotify / mean_spotify)


print(range_spotify)
print(var_spotify)
print(sd_spotify)
print(iqr_spotify)
print(cv_spotify)

#scatterplots
num_vars <- df[, c("popularity", "year", "bpm", "danceability", "valence", "duration")]
pairs(num_vars)
cor(num_vars)

#regression
cor.test(df$popularity, df$danceability)

#regression model
reg1 <- lm(popularity ~ year+bpm+danceability+valence+duration, data = df)
summary(reg1)
```

```
Call:
lm(formula = popularity ~ year + bpm + danceability + valence +
    duration, data = df)

Residuals:
     Min       1Q   Median       3Q      Max
-2.20643 -0.74369 -0.06352  0.64874  2.06685

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)  20.544131  74.501415   0.276    0.783
year          0.019481   0.036855   0.529    0.598
bpm           0.031568   0.004021   7.850 2.49e-12 ***
danceability  0.186008   0.008006  23.234  < 2e-16 ***
valence       0.019378   0.004805   4.033 9.99e-05 ***
duration     -0.032409   0.003175 -10.208  < 2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9796 on 114 degrees of freedom
Multiple R-squared:  0.9112,	Adjusted R-squared:  0.9073
F-statistic: 233.9 on 5 and 114 DF,  p-value: < 2.2e-16
```

ProjectPhase2_ExcelDat…	120 obs. of 9 variables
R12.5Cellphonesales	50 obs. of 4 variables
R12.5Cellphonesales.(1)	50 obs. of 4 variables
reg	List of 12
reg1	List of 12
reg2	List of 12
reg3	List of 12
ReverеBank	1041 obs. of 3 variables
Rwklysls	60 obs. of 5 variables
soccer_salaries	630 obs. of 8 variables
Values	
a	0.0013498980316301
alpha	0.1

The r-squared for this model is 0.9112 meaning that ~91% of the variation in popularity is explained by the independent variables. This suggests the model has a strong fit to the data.

Is R-Squared Statistically Significant?

□ H_0: All slope coefficients = 0 (no linear relationship between Popularity and any predictor)

□ H_a: At least one slope coefficient ≠ 0

The F-statistic is 233.9 with a p-value of 2.2e-16 and because this p-value is less than .05 we reject the null hypothesis. This means that the regression model is statistically significant in explaining the variation in popularity.

Conclusion and interpretation

β0: if all of year, bpm, danceability, valence, and duration = 0, then estimated popularity = 20.544, not significant at the .05 level.
β1: year does not impact popularity with statistical significance at the .05 level.
β2: bpm does not impact popularity with statistical significance at the .05 level.

β3: for a one unit increase in danceability, pop increases by .186, holding

all other variables fixed, significant at the .05 level.

β4: for a one unit increase in valence, pop increases by .193, holding

all other variables fixed, significant at the .05 level.

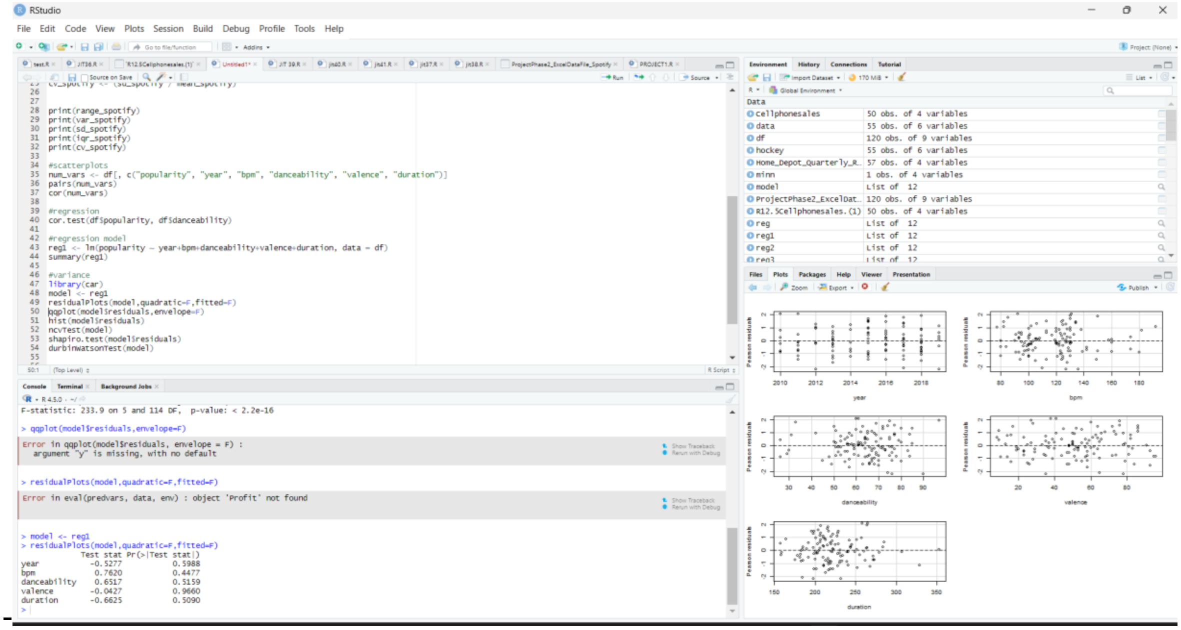
β5: for a one unit increase in duration, pop decreases by .324, holding

all other variables fixed, significant at the .05 level.

Residual Analysis

To assess whether the assumptions of simple linear regression were met, several statistical tests were used. These include graphical plots as well as formal tests for normality, homoscedasticity, and independence of residuals.

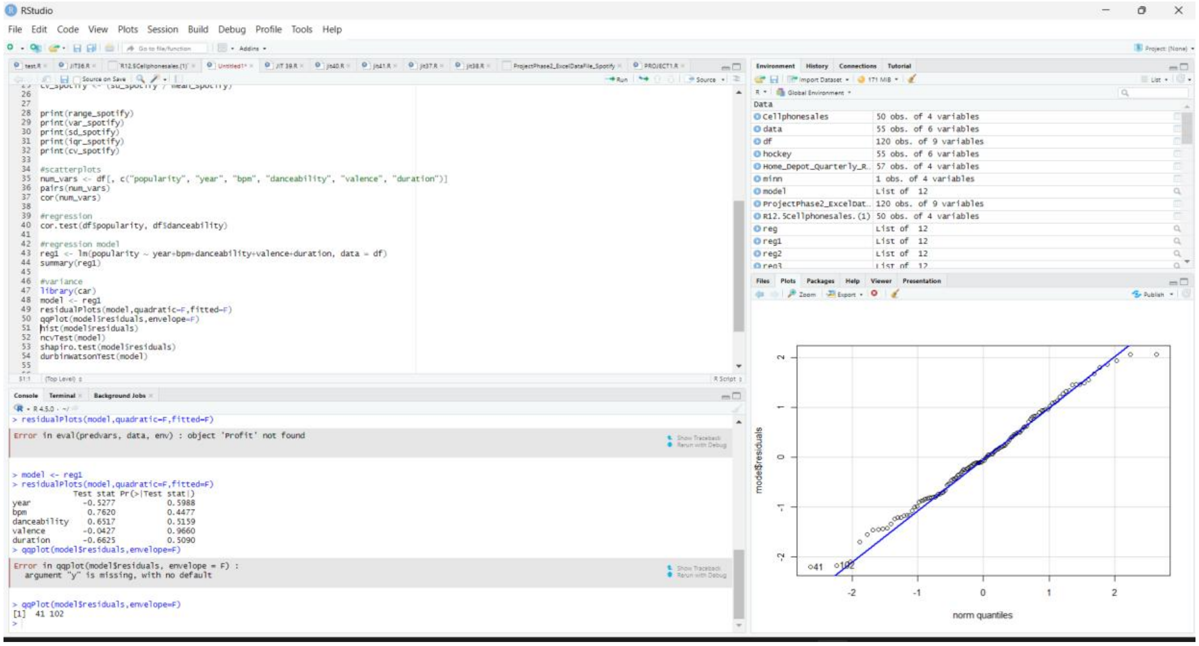
Graphs



```
print(range_spotify)
print(var_spotify)
print(sd_spotify)
print(iqr_spotify)
print(cv_spotify)

#scatterplots
num_vars <- df[, c("popularity", "year", "bpm", "danceability", "valence", "duration")]
pairs(num_vars)
cor(num_vars)

#regression
cor.test(df$popularity, df$danceability)

#regression model
reg1 <- lm(popularity ~ year+bpm+danceability+valence+duration, data = df)
summary(reg1)

#variance
library(car)
model <- reg1
residualPlots(model,quadratic=F,fitted=F)
qqplot(model$residuals,envelope=F)
hist(model$residuals)
ncvTest(model)
shapiro.test(model$residuals)
durbinWatsonTest(model)
```

```
F-statistic: 233.9 on 5 and 114 DF,  p-value: < 2.2e-16

> qqplot(model$residuals,envelope=F)
Error in qqplot(model$residuals, envelope = F) :
  argument "y" is missing, with no default

> residualPlots(model,quadratic=F,fitted=F)
Error in eval(predvars, data, env) : object 'Profit' not found

> model <- reg1
> residualPlots(model,quadratic=F,fitted=F)
             Test stat Pr(>|Test stat|)
year           -0.5277           0.5988
bpm             0.7620           0.4477
danceability    0.6517           0.5159
valence        -0.0427           0.9660
duration       -0.6625           0.5090
>
```

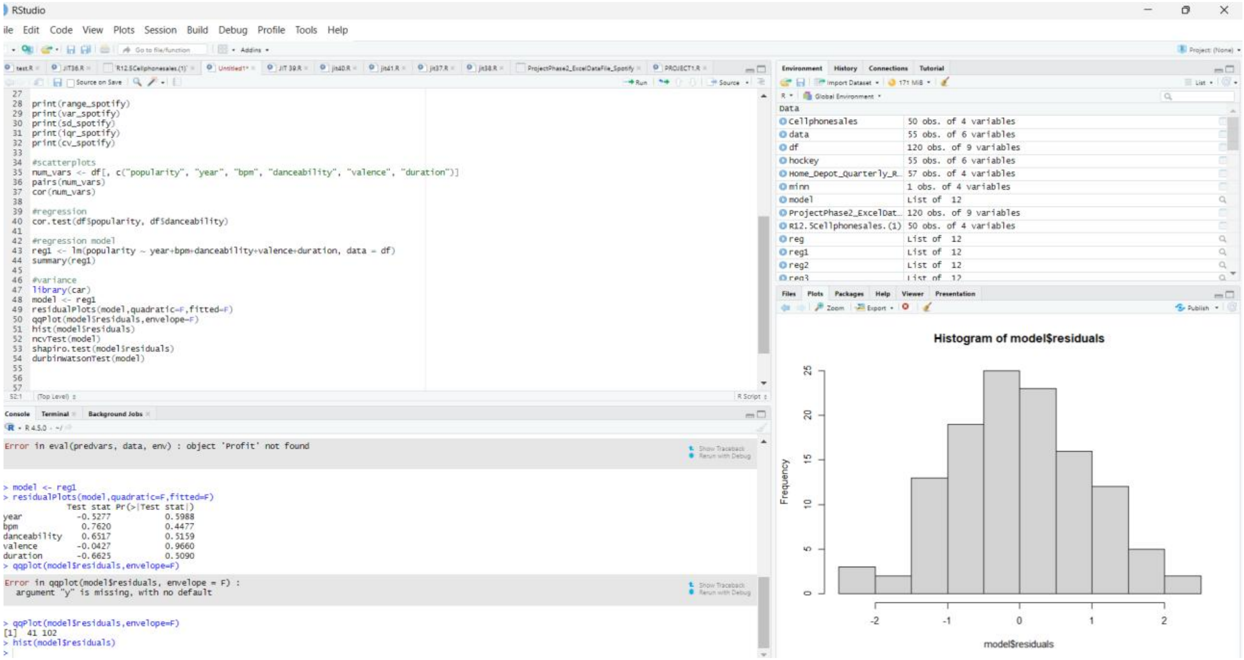
```
Data
cellphonesales              50 obs. of 4 variables
data                        55 obs. of 6 variables
df                          120 obs. of 9 variables
hockey                      55 obs. of 6 variables
Home_Depot_Quarterly_R..    57 obs. of 4 variables
minn                        1 obs. of 4 variables
model                       List of  12
ProjectPhase2_ExcelDat..    120 obs. of 9 variables
R12.5Cellphonesales.(1)     50 obs. of 4 variables
reg                         List of  12
reg1                        List of  12
reg2                        List of  12
reg3                        List of  12
```



```
print(range_spotify)
print(var_spotify)
print(sd_spotify)
print(iqr_spotify)
print(cv_spotify)

#scatterplots
num_vars <- df[, c("popularity", "year", "bpm", "danceability", "valence", "duration")]
pairs(num_vars)
cor(num_vars)

#regression
cor.test(df$popularity, df$danceability)

#regression model
reg1 <- lm(popularity ~ year+bpm+danceability+valence+duration, data = df)
summary(reg1)

#variance
library(car)
model <- reg1
residualPlots(model,quadratic=F,fitted=F)
qqPlot(model$residuals,envelope=F)
hist(model$residuals)
ncvTest(model)
shapiro.test(model$residuals)
durbinWatsonTest(model)
```

```
> residualPlots(model,quadratic=F,fitted=F)
Error in eval(predvars, data, env) : object 'Profit' not found

> model <- reg1
> residualPlots(model,quadratic=F,fitted=F)
             Test stat Pr(>|Test stat|)
year           -0.5277           0.5988
bpm             0.7620           0.4477
danceability    0.6517           0.5159
valence        -0.0427           0.9660
duration       -0.6625           0.5090
> qqplot(model$residuals,envelope=F)
Error in qqplot(model$residuals, envelope = F) :
  argument "y" is missing, with no default

> qqPlot(model$residuals,envelope=F)
[1]  41 102
>
```

```
Data
cellphonesales              50 obs. of 4 variables
data                        55 obs. of 6 variables
df                          120 obs. of 9 variables
hockey                      55 obs. of 6 variables
Home_Depot_Quarterly_R..    57 obs. of 4 variables
minn                        1 obs. of 4 variables
model                       List of  12
ProjectPhase2_ExcelDat..    120 obs. of 9 variables
R12.5Cellphonesales.(1)     50 obs. of 4 variables
reg                         List of  12
reg1                        List of  12
reg2                        List of  12
reg3                        List of  12
```



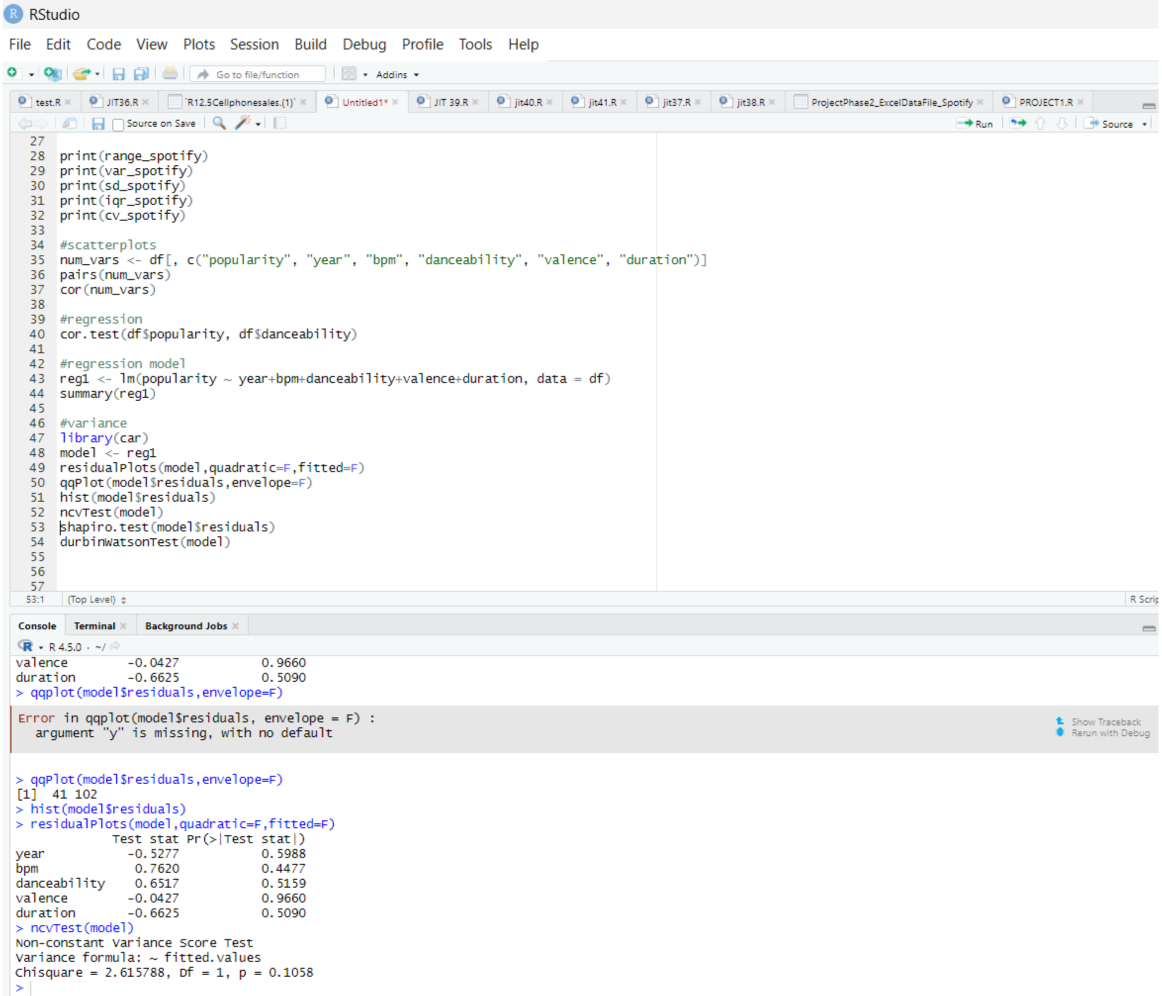
```
print(range_spotify)
print(var_spotify)
print(sd_spotify)
print(iqr_spotify)
print(cv_spotify)

#scatterplots
num_vars <- df[, c("popularity", "year", "bpm", "danceability", "valence", "duration")]
pairs(num_vars)
cor(num_vars)

#regression
cor.test(df$popularity, df$danceability)

#regression model
reg1 <- lm(popularity ~ year+bpm+danceability+valence+duration, data = df)
summary(reg1)

#variance
library(car)
model <- reg1
residualPlots(model,quadratic=F,fitted=F)
qqPlot(model$residuals,envelope=F)
hist(model$residuals)
ncvTest(model)
shapiro.test(model$residuals)
durbinWatsonTest(model)
```

```
Error in eval(predvars, data, env) : object 'Profit' not found


> model <- reg1
> residualPlots(model,quadratic=F,fitted=F)
             Test stat Pr(>|Test stat|)
year           -0.5277           0.5988
bpm             0.7620           0.4477
danceability    0.6517           0.5159
valence        -0.0427           0.9660
duration       -0.6625           0.5090
> qqplot(model$residuals,envelope=F)
Error in qqplot(model$residuals, envelope = F) :
  argument "y" is missing, with no default


> qqPlot(model$residuals,envelope=F)
[1]  41 102
> hist(model$residuals)
>
```

```
Data
Cellphonesales              50 obs. of 4 variables
data                        55 obs. of 6 variables
df                          120 obs. of 9 variables
hockey                      55 obs. of 6 variables
Home_Depot_Quarterly_R…     57 obs. of 4 variables
minn                        1 obs. of 4 variables
model                       List of  12
ProjectPhase2_ExcelDat…     120 obs. of 9 variables
R12.5Cellphonesales.(1)     50 obs. of 4 variables
reg                         List of  12
reg1                        List of  12
reg2                        List of  12
reg3                        List of  12
```

The residual plot reveals a clear pattern in year and its residual value and it slowly gets random in the other independent variables. This could mean that the linear assumption is met and that there's evidence for homoscedasticity if the plot shows a clear pattern. For the Q-Q plot, there is a strong linear pattern in the graph which could indicate normal linearity. The histogram is unimodal and approximately symmetric, indicating a normal distribution of the data.

Breusch-Pagan Test



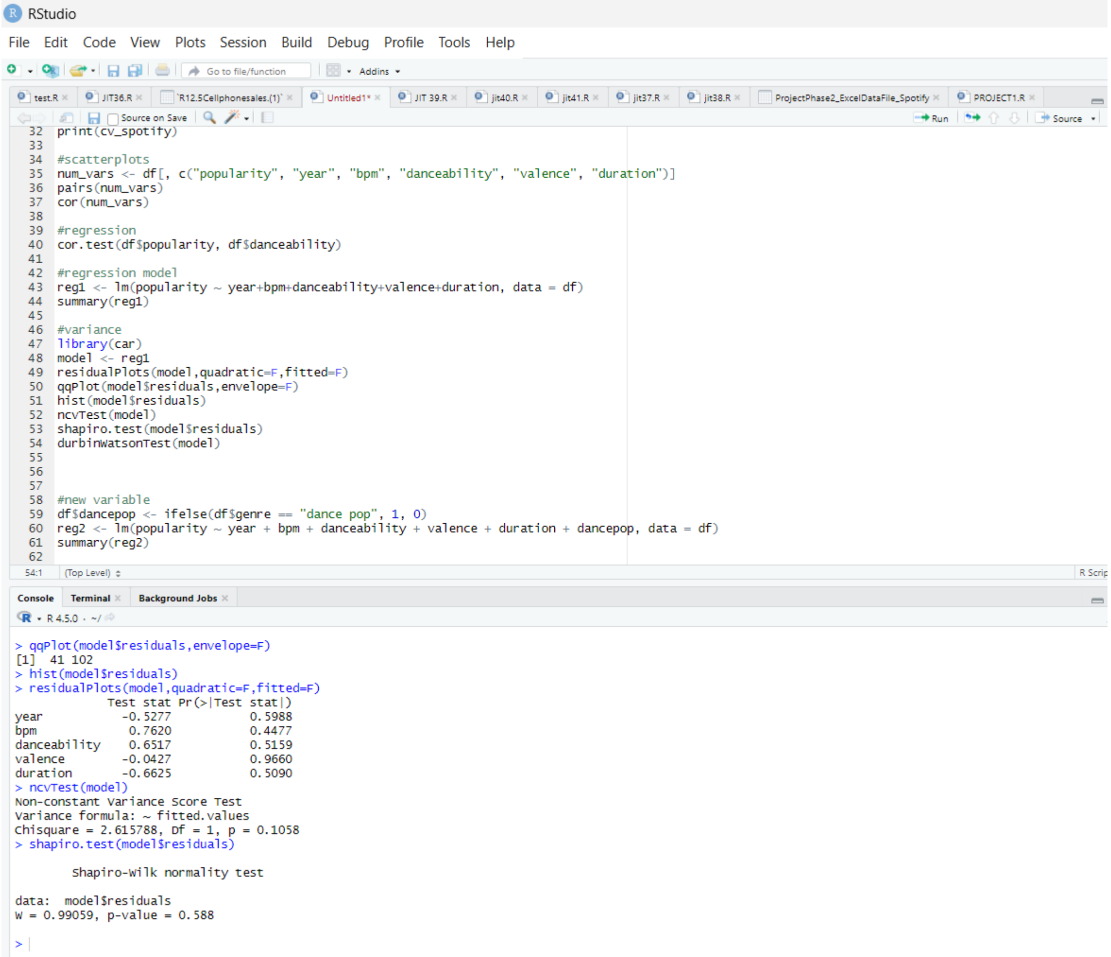
```r

print(range_spotify)
print(var_spotify)
print(sd_spotify)
print(iqr_spotify)
print(cv_spotify)

#scatterplots
num_vars <- df[, c("popularity", "year", "bpm", "danceability", "valence", "duration")]
pairs(num_vars)
cor(num_vars)

#regression
cor.test(df$popularity, df$danceability)

#regression model
reg1 <- lm(popularity ~ year+bpm+danceability+valence+duration, data = df)
summary(reg1)

#variance
library(car)
model <- reg1
residualPlots(model,quadratic=F,fitted=F)
qqPlot(model$residuals,envelope=F)
hist(model$residuals)
ncvTest(model)
shapiro.test(model$residuals)
durbinWatsonTest(model)



```

```
valence         -0.0427           0.9660
duration        -0.6625           0.5090
> qqplot(model$residuals,envelope=F)
Error in qqplot(model$residuals, envelope = F) :
  argument "y" is missing, with no default


> qqPlot(model$residuals,envelope=F)
[1]  41 102
> hist(model$residuals)
> residualPlots(model,quadratic=F,fitted=F)
             Test stat Pr(>|Test stat|)
year           -0.5277           0.5988
bpm             0.7620           0.4477
danceability    0.6517           0.5159
valence        -0.0427           0.9660
duration       -0.6625           0.5090
> ncvTest(model)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 2.615788, Df = 1, p = 0.1058
>
```

Interpretation:

- If $p > 0.05$ → Homoscedasticity holds (constant variance).
- If $p < 0.05$ → Heteroscedasticity present.
- The Breusch–Pagan test resulted in $p > 0.05$, suggesting the variance of residuals is constant, and we fail to reject that homoscedasticity is present.

Shapiro-Wilk Test



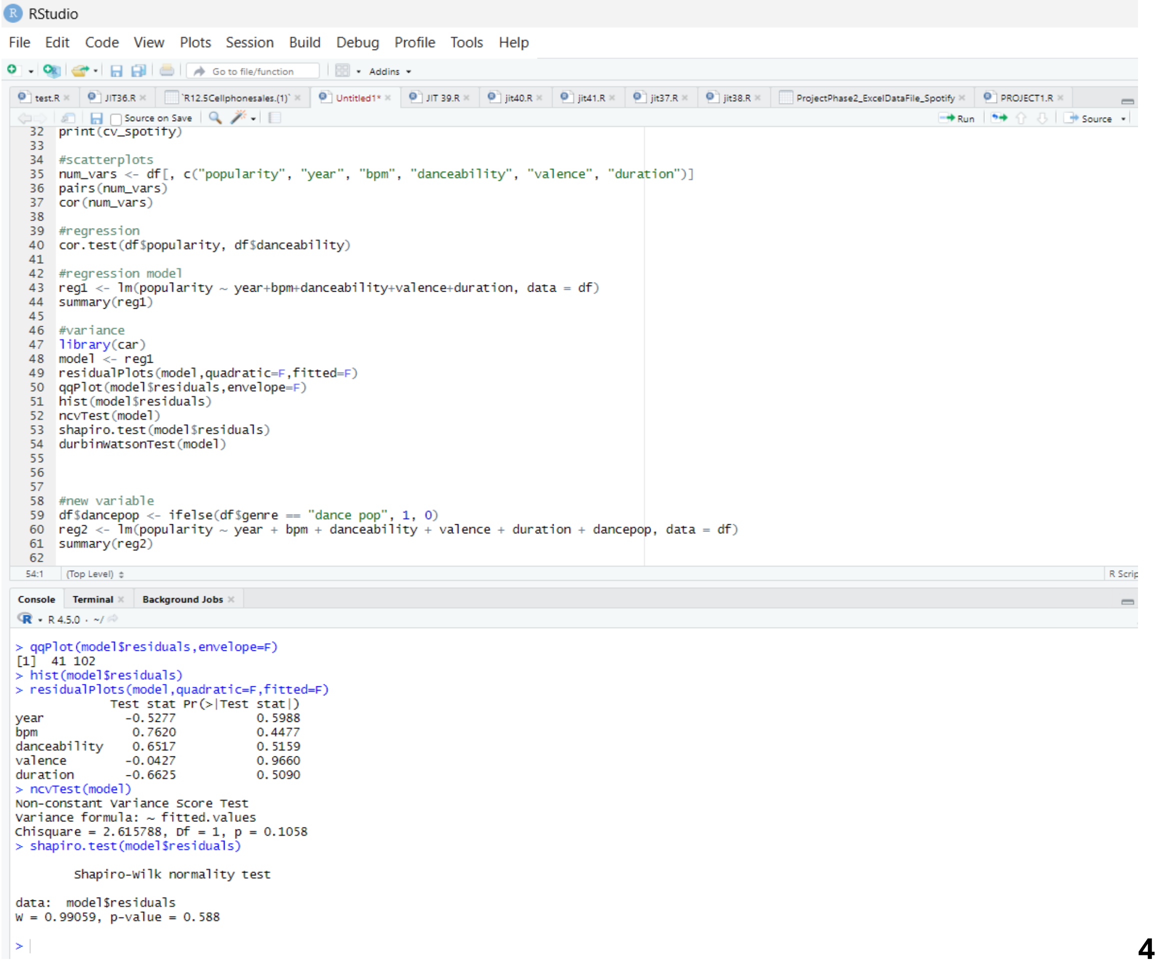
```r
print(cv_spotify)

#scatterplots
num_vars <- df[, c("popularity", "year", "bpm", "danceability", "valence", "duration")]
pairs(num_vars)
cor(num_vars)

#regression
cor.test(df$popularity, df$danceability)

#regression model
reg1 <- lm(popularity ~ year+bpm+danceability+valence+duration, data = df)
summary(reg1)

#variance
library(car)
model <- reg1
residualPlots(model,quadratic=F,fitted=F)
qqPlot(model$residuals,envelope=F)
hist(model$residuals)
ncvTest(model)
shapiro.test(model$residuals)
durbinWatsonTest(model)



#new variable
df$dancepop <- ifelse(df$genre == "dance pop", 1, 0)
reg2 <- lm(popularity ~ year + bpm + danceability + valence + duration + dancepop, data = df)
summary(reg2)

```

```
> qqPlot(model$residuals,envelope=F)
[1]  41 102
> hist(model$residuals)
> residualPlots(model,quadratic=F,fitted=F)
             Test stat Pr(>|Test stat|)
year           -0.5277           0.5988
bpm             0.7620           0.4477
danceability    0.6517           0.5159
valence        -0.0427           0.9660
duration       -0.6625           0.5090
> ncvTest(model)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 2.615788, Df = 1, p = 0.1058
> shapiro.test(model$residuals)

        Shapiro-Wilk normality test

data:  model$residuals
W = 0.99059, p-value = 0.588

> 
```

Interpretation:

- If p > 0.05 → Fail to reject normality
- If p < 0.05 → Evidence of non-normality.
- The Shapiro–Wilk test yielded p > 0.05, indicating that we fail to reject normality
- The result aligns with what is seen visually in the Q–Q plot and histogram.

Durbin–Watson Test



```
print(cv_spotify)

#scatterplots
num_vars <- df[, c("popularity", "year", "bpm", "danceability", "valence", "duration")]
pairs(num_vars)
cor(num_vars)

#regression
cor.test(df$popularity, df$danceability)

#regression model
reg1 <- lm(popularity ~ year+bpm+danceability+valence+duration, data = df)
summary(reg1)

#variance
library(car)
model <- reg1
residualPlots(model,quadratic=F,fitted=F)
qqPlot(model$residuals,envelope=F)
hist(model$residuals)
ncvTest(model)
shapiro.test(model$residuals)
durbinWatsonTest(model)



#new variable
df$dancepop <- ifelse(df$genre == "dance pop", 1, 0)
reg2 <- lm(popularity ~ year + bpm + danceability + valence + duration + dancepop, data = df)
summary(reg2)

```

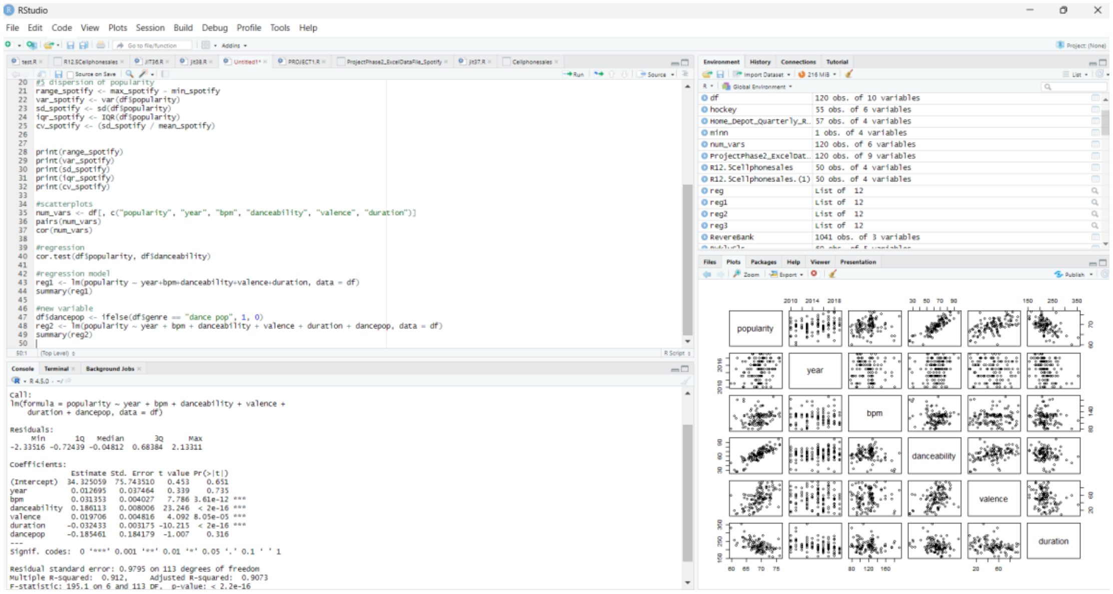
```
> qqPlot(model$residuals,envelope=F)
[1]  41 102
> hist(model$residuals)
> residualPlots(model,quadratic=F,fitted=F)
             Test stat Pr(>|Test stat|)
year           -0.5277           0.5988
bpm             0.7620           0.4477
danceability    0.6517           0.5159
valence        -0.0427           0.9660
duration       -0.6625           0.5090
> ncvTest(model)
Non-constant Variance Score Test
Variance formula: ~ fitted.values
Chisquare = 2.615788, Df = 1, p = 0.1058
> shapiro.test(model$residuals)

        Shapiro-Wilk normality test

data:  model$residuals
W = 0.99059, p-value = 0.588

> 
```

4

Interpretation:

- p > 0.05 → No significant autocorrelation (independent errors)
- p < 0.05 → Evidence of autocorrelation
- The Durbin–Watson test produced p > 0.05, indicating that there is no significant autocorrelation
- This suggests the independence assumption is met.

All diagnostic tools and statistical tests indicate that the regression assumptions are satisfied and so the regression model provides a reliable and unbiased fit for the data.



```
#5 dispersion of popularity
range_spotify <- max_spotify - min_spotify
var_spotify <- var(df$popularity)
sd_spotify <- sd(df$popularity)
iqr_spotify <- IQR(df$popularity)
cv_spotify <- (sd_spotify / mean_spotify)


print(range_spotify)
print(var_spotify)
print(sd_spotify)
print(iqr_spotify)
print(cv_spotify)

#scatterplots
num_vars <- df[, c("popularity", "year", "bpm", "danceability", "valence", "duration")]
pairs(num_vars)
cor(num_vars)

#regression
cor.test(df$popularity, df$danceability)

#regression model
reg1 <- lm(popularity ~ year+bpm+danceability+valence+duration, data = df)
summary(reg1)

#new variable
df$dancepop <- ifelse(df$genre == "dance pop", 1, 0)
reg2 <- lm(popularity ~ year + bpm + danceability + valence + duration + dancepop, data = df)
summary(reg2)

```

df	120 obs. of 10 variables
hockey	55 obs. of 6 variables
Home_Depot_Quarterly_R…	57 obs. of 4 variables
minn	1 obs. of 4 variables
num_vars	120 obs. of 6 variables
ProjectPhase2_ExcelDat…	120 obs. of 9 variables
R12.5Cellphonesales	50 obs. of 4 variables
R12.5Cellphonesales.(1)	50 obs. of 4 variables
reg	List of 12
reg1	List of 12
reg2	List of 12
reg3	List of 12
RevereBank	1041 obs. of 3 variables

```
R 4.5.0 · ~/
Call:
lm(formula = popularity ~ year + bpm + danceability + valence + 
    duration + dancepop, data = df)

Residuals:
     Min       1Q   Median       3Q      Max 
-2.33516 -0.72439 -0.04812  0.68384  2.13311 

Coefficients:
              Estimate Std. Error t value Pr(>|t|)    
(Intercept)  34.325059  75.743510   0.453    0.651    
year          0.012695   0.037464   0.339    0.735    
bpm           0.031353   0.004027   7.786 3.61e-12 ***
danceability  0.186113   0.008006  23.246  < 2e-16 ***
valence       0.019706   0.004816   4.092 8.05e-05 ***
duration     -0.032433   0.003175 -10.215  < 2e-16 ***
dancepop     -0.185461   0.184179  -1.007    0.316    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.9795 on 113 degrees of freedom
Multiple R-squared:  0.912,	Adjusted R-squared:  0.9073 
F-statistic: 195.1 on 6 and 113 DF,  p-value: < 2.2e-16
```

Extended Model Including Genre

The adjusted r-squared for the original model was .9073 and when the dancepop variable was added the adjusted r-squared became .9073. Because the adjusted r-squared was the same it didn’t improve the model’s explanatory power.

$\beta 6$: The coefficient on dance pop is -.1854 with a p-value of .316. This means that with all the variables held constant, songs classified as dance pop are predicted to have .1854 less popularity points than non-dance pop songs. The p-value is greater than 0.05, this is statistically significant at the .05 level.