What do the attributes of top the songs of the last decade tell us?

Homayoun Sadri

2022-05-12

Contents

1 Introduction

In this project, I will utilize the statistical methods we learned in class to analyze “Spotify Top 100 Songs of 2010-2019” dataset. First, I will give a brief overview of these techniques in the methodology section. Next, I will go over the dataset that I have chosen. And, in analysis and results section, I will try to ask interesting questions about the data and find statistically valid answers to them. Questions like: Have the songs gotten shorter over the years? I.e. is there a statistically significant difference between the means of duration of the songs over the years? Can we predict how popular a song is by knowing its attribute such as beats per minute, it’s dance-ability, its etc?

2 Methodoogy

2.1 Comparing The Means of More Than Two Samples

2.1.1 ANOVA

Analysis of Variance or ANOVA is a statistical technique to determine whether there is a statistically significance difference between the means of three or more samples. In other words, we would like to decide whether the factor has a significant effect on the dependent variable. In statistical terms, the null hypothesis H_0 of ANOVA states that all groups’ means are equal. And the alternative hypothesis H_1 is that H_0 is not true. In this test, the dependent variable/response is a numerical variable and the independent variable/factor is a categorical variable with two or more groups/levels.

Before carrying out ANOVA, we first need to make sure its assumptions are satisfied. ANOVA makes four assumptions:

1. All samples are independent and randomly selected from normal populations. I.e. ϵ_{ij} terms are normally distributed.
2. Homogeneity of variances of populations. I.e. populations are assumed to have equal standard deviations.
3. The factor is a categorical variable with two or more groups.
4. The response is a numerical variable.

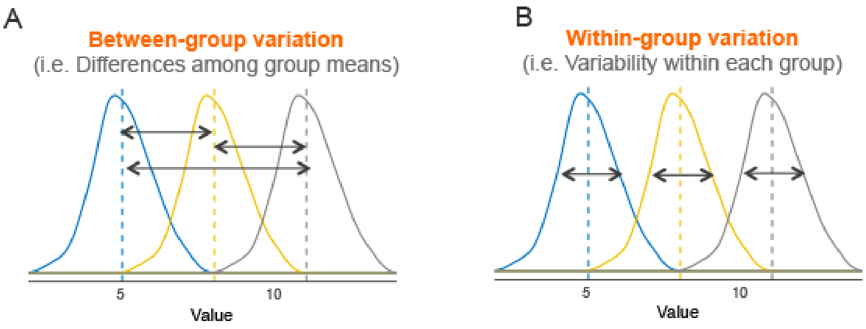
If the null hypothesis is rejected, i.e. there are at least two groups with significantly different means, we can use Tukey’s test or Bonferrani’s test to determine those groups and how their means differ.

If we are comparing the means, why is it called analysis of variance?

Assume that we have 3 groups to compare, as illustrated in the image below. The dashed line indicates the group mean. The figure shows the variation between the means of the groups (panel A) and the variation within each group (panel B), also known as residual variance.

The idea behind the ANOVA test is very simple: if the average variation between groups is large enough compared to the average variation within groups, then you could conclude that at least one group mean is not equal to the others.

Thus, it’s possible to evaluate whether the differences between the group means are significant by comparing the two variance estimates. This is why the method is called analysis of variance even though the main goal is to compare the group means.



2.1.2 Tukey's test

Tukey's test is used:

- To construct Confidence Intervals (CI) for differences of all pairs of means in such a was that intervals at the same time have a set coverage of probability.
- To find means that are significantly different from each other.

As always, we first need to check for the assumptions. They are:

1. Observations are in independent withing and among groups.
2. Groups associated with each mean in the test are normally distributed.
3. Homogeneity of variance.

Tukey's test is valid for equal sample sizes only.

2.1.3 Bonferrani's test

Bonferrani's test is used for the same purposes as Tukey's test but it has the advantage that it's valid for both equal and unequal sample sizes only.

2.1.3 Kruskal-Wallis test

This is a non-parametric method where no particular distribution for the population is assumed. It is considered the nonparametric alternative to the one-way ANOVA, and an extension of the Mann-Whitney U test to allow the comparison of more than two independent groups. The assumptions of this test are:

1. All samples are independent and randomly selected from their populations.
2. Homogeneity of the variance of the populations.
3. The factor (independent variable) is a categorical variable with two or more groups.
4. The response (dependent variable) is a numerical variable.

This test is valid for equal and unequal sample sizes.

2.2 Comparing The Means of Two Samples

When we only have two samples and aim to compare their means, we use z-test or t-test instead of ANOVA. (ANOVA gives the same result if used for two samples). Within the scope of our class, we make two assumptions for this comparison:

1. The populations are independent.
2. The populations have equal variances.

In the case the populations are normal, we use:
- The z-test: if the variance is known.
- The t-test: if the variance is not known.

And, in the case the populations are not normal, we use:
- The z-test: if the sample size $n_1 + n_2 - 2 > 30$. (Thanks to Central Limit Theorem)
- Some transformation (ex: log function) to make the data normal: if $n_1 + n_2 - 2 < 30$

2.3 Categorical Data Analysis

Here we are concerned with analysis of data that are in the form of **counts** in various categories.

2.3.1 Fisher's exact test

Fisher's exact test is a non-parametric method for comparing the proportion of categories in two different **independent** groups (categorical [nominal] variables) in a contingency table. The categorical variables should be measured **dichotomously** for 2x2 contingency table (e.g., male/female, treated/no treated, cured/no cured, etc.,). Let's list these assumptions for clarity:

1. The two variables are categorical (nominal) and data is randomly sampled.
2. The levels of variables are mutually exclusive.
3. Observations should be independent of each other.
4. Observation data should be frequency counts and not percentages, proportions or transformed data.

Unlike the chi-square test (mentioned in the next section), the Fisher's exact test is an exact test (i.e. returns exact p value) and can be applied on smaller sample sizes (<1000). This test is an alternative to the chi-square test, especially when the frequency count is < 5 for more than 20% of cells. If our sample size is larger, it is better to use chi-square test.

Finally, in the Fisher's exact test, the probability of getting results (observed frequencies) is directly calculated from hypergeometric distribution and not from using any test statistics.

2.3.2 The chi-square test of homogeneity

The chi-square test of homogeneity tests to see whether different columns (or rows) of data in a table come from the same population or not (i.e., whether the differences are consistent with being explained by sampling error alone). In other words, the test is applied to a single categorical variable from two or more different populations and tries to determine whether frequency counts are distributed identically across different populations.

The assumptions of this test are:

1. For each population, the sampling method is simple random sampling.
2. The variable under study is categorical.
3. If sample data are displayed in a contingency table (Populations x Category levels), the expected frequency count for each cell of the table is at least 5.

2.3.3 The chi-square test of independence

The chi-quare test of independence (or association) determines whether there is an association between categorical variables (i.e., whether the variables are independent or related). It is a nonparametric test.

Note that if your categorical variables represent "pre-test" and "post-test" observations, then the assumption of the independence of observations is violated. In this situation, we should use **McNemar's Test**.

The assumptions of this test are:

1. Two categorical variables.
2. Two or more categories (groups) for each variable.
3. Independence of observations.
- There is no relationship between the subjects in each group.
- The categorical variables are not "paired" in any way (e.g. pre-test/post-test observations).

4. Relatively large sample size.
- Expected frequencies for each cell are at least 1.
- Expected frequencies should be at least 5 for the majority (80%) of the cells.

2.4 Regression

2.4.1 Simple linear regression

In linear regression we consider the model:

$$y_i = \beta_0 + \beta_1 x_i + e_i \text{ for } i = 0, ..., n$$

and we fit a straight line to data as

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_i \text{ for } i = 0, ..., n.$$

$\hat{\beta}_0$ and $\hat{\beta}_1$ are the regression coefficients.

The assumptions of linear regression model are:

1. Linear relationship: between x and y.
2. Independence: the residuals are independent.
3. Homoscedasticity of errors: residuals have equal variance at every x.
4. Errors are normally distributed.

2.4.2 Multiple linear regression

Here we consider the following model

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_{p-1} x_{ip-1} + e_i \text{ for } i = 0, ..., n$$

and we fit a linear combination of the independent variables to our dependent variable as

$$y_i = \hat{\beta}_0 + \hat{\beta}_1 x_{i1} + \hat{\beta}_1 x_{i2} + ... + \hat{\beta}_{p-1} x_{ip-1} \text{ for } i = 0, ..., n.$$

3 Dataset

I will analyze "Spotify Top 100 Songs of 2010-2019" dataset. This dataset is provided by Kaggle. As the name suggests, this dataset consists of top 100 songs played on Spotify each year from 2010 to 2019. The dataset contains 17 variables and 1003 observations. 3 observations are missing values and will be removed. A summary of variables is shown the table below.

```
Column_Name <- c('title','artist' ,'genre', 'year released', 'added', 'bpm', 'nrgy',
            'dnce', 'dB', 'live', 'val', 'dur', 'acous', 'spch', 'pop', 'top year',
            'artist type')

Column_Description <-  c('Song\'s Title','Song\'s artist' ,'Genre of song',
                         'Year the song was released',
                        'Day song was added to Spotify\'s Top Hits playlist',
                        'Beats Per Minute - The tempo of the song',
                        'Energy - How energetic the song is',
                        'Danceability - How easy it is to dance to the song',
                        'Decibel - How loud the song is',
                        'How likely the song is a live recording',
                        'How positive the mood of the song is',
                        'Duration of the song', 'How acoustic the song is',
                        'The more the song is focused on spoken word',
                        'Popularity of the song (not a ranking)',
                        'Year the song was a top hit',
                        'Tells if artist is solo, duo, trio, or a band')

var_info <- data.frame(Column_Name = Column_Name,
                       Column_Description = Column_Description)


library(knitr)

kable(var_info)
```

Column_Name	Column_Description
title	Song's Title
artist	Song's artist
genre	Genre of song
year released	Year the song was released
added	Day song was added to Spotify's Top Hits playlist
bpm	Beats Per Minute - The tempo of the song
nrgy	Energy - How energetic the song is
dnce	Danceability - How easy it is to dance to the song
dB	Decibel - How loud the song is
live	How likely the song is a live recording
val	How positive the mood of the song is
dur	Duration of the song
acous	How acoustic the song is
spch	The more the song is focused on spoken word
pop	Popularity of the song (not a ranking)
top year	Year the song was a top hit
artist type	Tells if artist is solo, duo, trio, or a band

4 Analysis & Results

4.1 Comparing The Means of More Than Two Samples

4.1.1 ANOVA

The first question that I was interested to know the answer of is whether the duration of the top songs has changed over the years. I really enjoy the songs from the 80's and 90's. Those songs were longer. So I wondered if in the short span of 2010 to 2019 there has been a change in duration of the top songs. To answer this question, I choose the `top.year` variable as the factor and the `dur` variable as the response. First, I need to convert `top.year` to a factor variable since it's a numerical variable.

```
df = read.csv('/Users/homayounsadri/Downloads/archive/Spotify 2010 - 2019 Top 100.csv')

df <- na.omit(df)

df$top.year = as.factor(df$top.year)
```

Before doing ANOVA, I'll check that its assumptions hold.

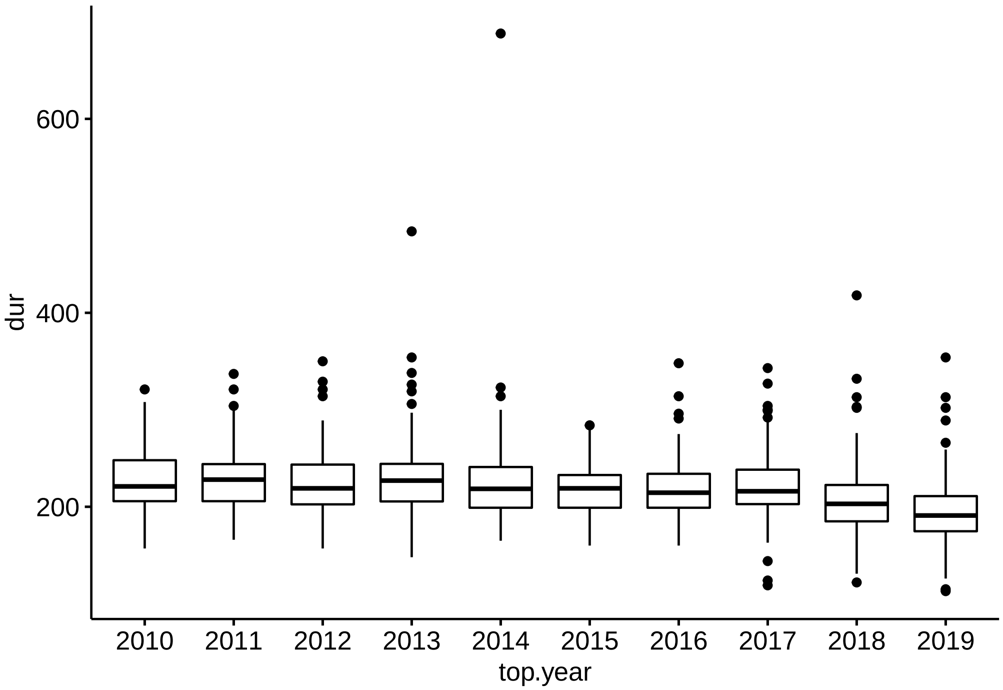
4.1.1.1 Outliers Outliers can greatly affect normality and homogeneity of variance.

```
library(dplyr)
library(rstatix)

df %>%
  group_by(top.year) %>%
  identify_outliers(dur)
```

```
## # A tibble: 44 x 19
##    top.year title   artist top.genre year.released added   bpm  nrgy  dnce    dB
##    <fct>    <chr>   <chr>  <chr>             <int> <chr> <int> <int> <int> <int>
##  1 2010     Rivers~ Sidne~ dutch ho~          2009 2022~   126    98    80    -2
##  2 2011     Lighte~ Bad M~ detroit ~          2011 2020~    90    70    68    -8
##  3 2011     Holoce~ Bon I~ eau clai~          2011 2020~   148    30    37   -15
##  4 2011     The Ed~ Lady ~ art pop            2011 2020~   128    77    58    -7
##  5 2012     Mercy   Kanye~ chicago ~          2012 2020~   140    50    56    -9
##  6 2012     Swimmi~ Kendr~ consciou~          2012 2020~    74    49    72    -8
##  7 2012     m.A.A.~ Kendr~ consciou~          2012 2020~    91    73    49    -7
##  8 2012     Anna S~ WALK ~ dance pop          2012 2021~   140    84    47    -7
##  9 2013     Lose Y~ Daft ~ electro            2013 2020~   100    66    83    -8
## 10 2013     Holy G~ JAY-Z  east coa~          2013 2020~   145    53    68    -7
## # ... with 34 more rows, and 9 more variables: live <int>, val <int>,
## #   dur <int>, acous <int>, spch <int>, pop <int>, artist.type <chr>,
## #   is.outlier <lgl>, is.extreme <lgl>
```

It seems like we have a lot of outliers. We should keep in mind that these can affect the result of our analysis. Let's check them out visually by creating some boxplots.

```
library(ggpubr)

ggboxplot(df, x = "top.year", y = "dur")
```



We have three extreme outliers in the years 2014, 2013 and 2018.

4.1.1.2 Normality The normality assumption can be checked by using one of the following two approaches:

1. Analyzing the ANOVA model residuals to check the normality for all groups together. This approach is easier and it's very handy when we have many groups or if there are few data points per group.

2. Checking normality for each group separately. This approach might be used when we have only a few groups and many data points per group.

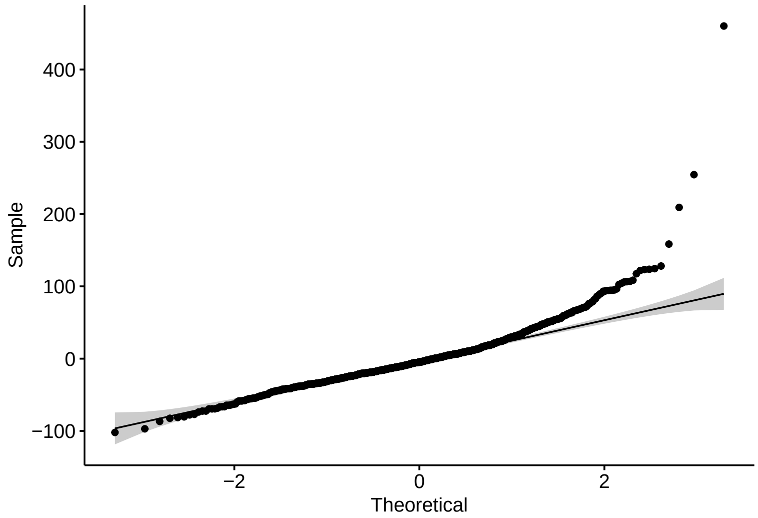
In our case, the second approach is more reasonable since we have a few groups (10) and many observations per group (100). However, I will use both methods.

1. Checking normality assumption by analyzing the model residuals

QQ plot and **Shapiro-Wilk test** of normality are used for this.

QQ plot draws the correlation between a given data and the normal distribution. Significant deviations from the line will suggest against the normality assumption.

```
library("ggpubr")

model  <- lm(dur ~ top.year, data = df)

ggqqplot(residuals(model))
```



The points in the middle of graph fall on a straight lines but at the both ends they deviate from the line. Normal Q-Q plots that exhibit this behavior usually mean our data have more extreme values than would be expected if they truly came from a Normal distribution more info here.

Now, I will also perform a significance test. There are multiple-tests available to check the normality: Kolmogorov-Smirnov (K-S) normality test and Shapiro-Wilk's test. Shapiro-Wilk's method is widely recommended for normality test and it provides better power than K-S. It is based on the correlation between the data and the corresponding normal scores. I will perform this test.

```
shapiro.test(df$dur)
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$dur
## W = 0.87772, p-value < 2.2e-16
```

From the output, p-value < 0.05 implying that the distribution of the data is significantly different from normal distribution. So, all three methods performed so far gave the same results. However, since our sample size is large (1000), we can still use ANOVA.

2. Checking normality assumption by groups

Here we compute Shapiro-Wilk test for each group level. If the data is normally distributed, the p-value should be greater than 0.05.

```
library(dplyr)

df %>%
  group_by(top.year) %>%
  shapiro_test(dur)
```

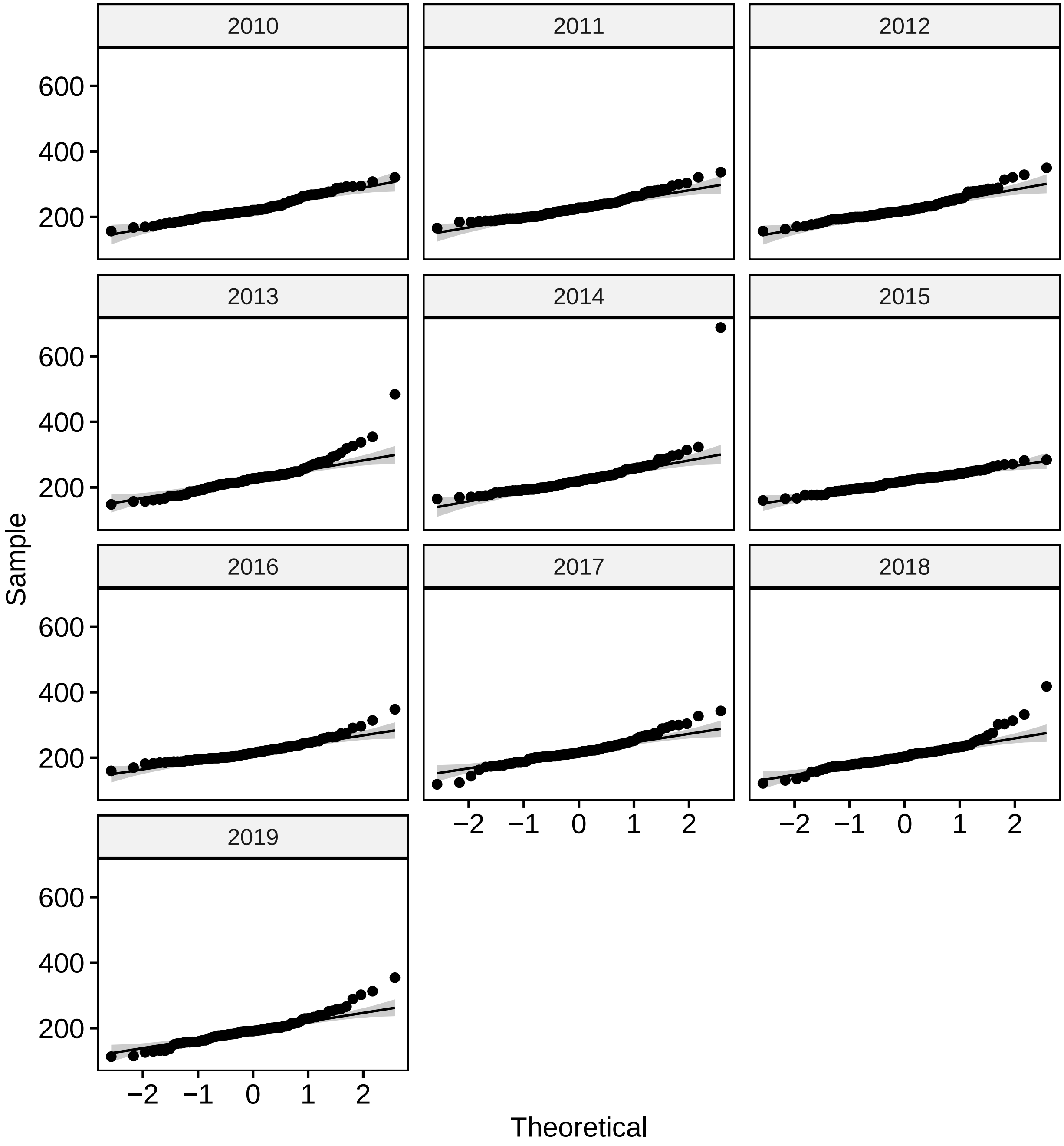
```
## # A tibble: 10 x 4
##    top.year variable statistic        p
##    <fct>    <chr>        <dbl>    <dbl>
##  1 2010     dur          0.968 1.66e- 2
##  2 2011     dur          0.954 1.51e- 3
##  3 2012     dur          0.936 1.10e- 4
##  4 2013     dur          0.867 5.51e- 8
##  5 2014     dur          0.580 1.80e-15
##  6 2015     dur          0.991 7.23e- 1
##  7 2016     dur          0.914 6.77e- 6
##  8 2017     dur          0.959 3.50e- 3
##  9 2018     dur          0.872 8.53e- 8
## 10 2019     dur          0.938 1.56e- 4
```

None of the groups are normally distributed since all the p-values are less than 0.05.

Note: If our sample size is greater than 50, the normal QQ plot is preferred because at larger sample sizes the Shapiro-Wilk test becomes very sensitive even to a minor deviation from normality.

QQ plot draws the correlation between a given data and the normal distribution. Let's create QQ plots for each group level:

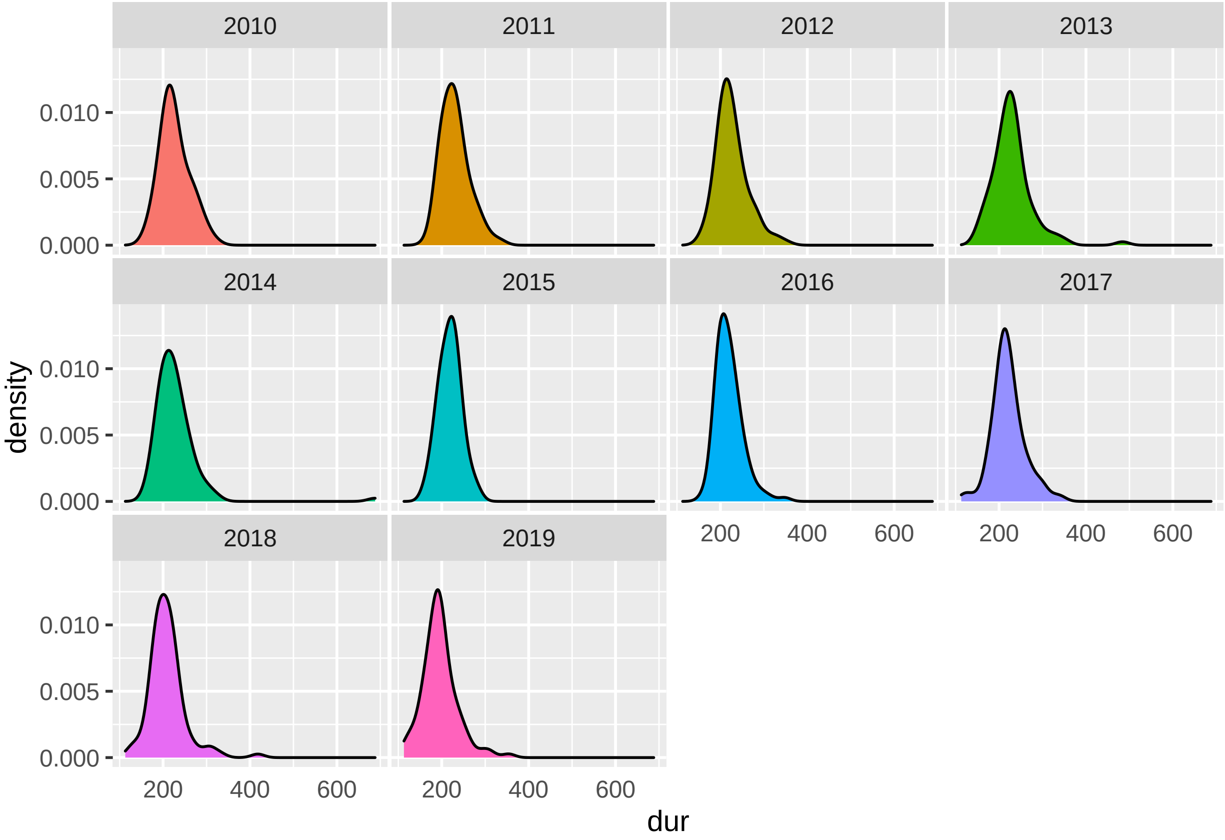
```
library("ggpubr")

ggqqplot(df, "dur", facet.by = "top.year", ncol=3)
```



The plots confirm that in fact all the groups deviate from the normal distribution.

Finally, another visual method is a density plot. This allows us to look at the distribution of data and see whether it's bell shaped.

```
library(ggplot2)
library(hrbrthemes)

ggplot(data=df, aes(x=dur, group=top.year, fill=top.year)) +
    geom_density(adjust=1.5) +
    # theme_ipsum() + #Comment out for pdf.
    facet_wrap(~top.year) +
    theme(
      legend.position="none",
      panel.spacing = unit(0.1, "lines"),
      axis.ticks.x=element_blank()
    )
```



We can see from the plots that all the distributions look like right-skewed normal distribution.

4.1.1.3 Homogeneity of Variance The `bartlett.test( )` function provides a parametric K-sample test of the equality of variances.

```
bartlett.test(dur ~ top.year, data= df)
```

```
##
##  Bartlett test of homogeneity of variances
##
## data:  dur by top.year
## Bartlett's K-squared = 95.084, df = 9, p-value < 2.2e-16
```

The `fligner.test( )` function provides a non-parametric test of the same.

```
fligner.test(dur ~ top.year, data= df)
```

```
##
##  Fligner-Killeen test of homogeneity of variances
##
## data:  dur by top.year
## Fligner-Killeen:med chi-squared = 8.9222, df = 9, p-value = 0.4445
```

Both tests point to difference in variances. However, note that the parametric test has a much lower p-value.

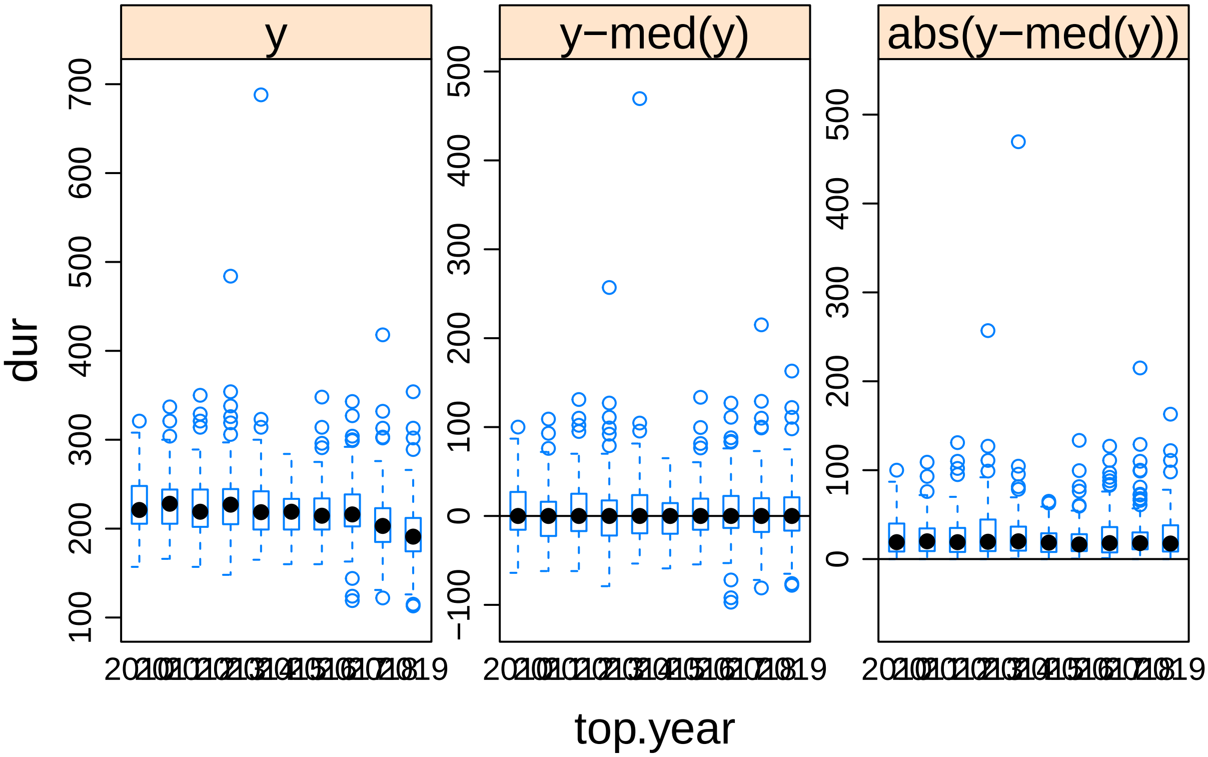
This is in line with the fact that parametric tests usually have more statistical power than their non-parametric equivalents. More information on the these two tests here.

Next, I will visually check the homogeniety of varainces. The `hovPlot( )` function in the HH package provides a graphic test of homogeneity of variances based on Brown-Forsyth.

```
library(HH)

hov(dur ~ top.year, data= df)
```

```
##
##  hov: Brown-Forsyth
##
## data:  dur
## F = 1.2648, df:top.year = 9, df:Residuals = 990, p-value = 0.252
## alternative hypothesis: variances are not identical
```

```
hovPlot(dur ~ top.year, data= df)
```



We have three panels containing boxplots for: the observed data `df$dur`, the data with the median subtracted `$df$dur - med(df$dur)`, and the absolute deviations from the median `med(df$dur)`. The Brown and Forsyth test statistic is the F statistic resulting from an ordinary one-way analysis of variance on the data points in the third panel. And the test confirms that the variances are not equal.

However as mentioned earlier since our sample size is large, we can perform ANOVA. I will then perform Kruskal-Wallis test and compare the results of the two.

```
one.way <- aov(dur ~ top.year, data = df)

summary(one.way)
```

```
##              Df  Sum Sq Mean Sq F value   Pr(>F)
## top.year      9  106635   11848   7.894 2.56e-11 ***
## Residuals   990 1485990    1501
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The p-value is (much) less than 0.05. So, there are at least two groups whose means are different (or that there is significant difference in the mean values between the three groups).

A significant one-way ANOVA is generally followed up by Tukey's **post-hoc** test to perform multiple pairwise comparisons between groups.

Since there are equal observations per group, we can use either of Tukey's test or Bonferrani's test to determine which pairs of groups have different means.

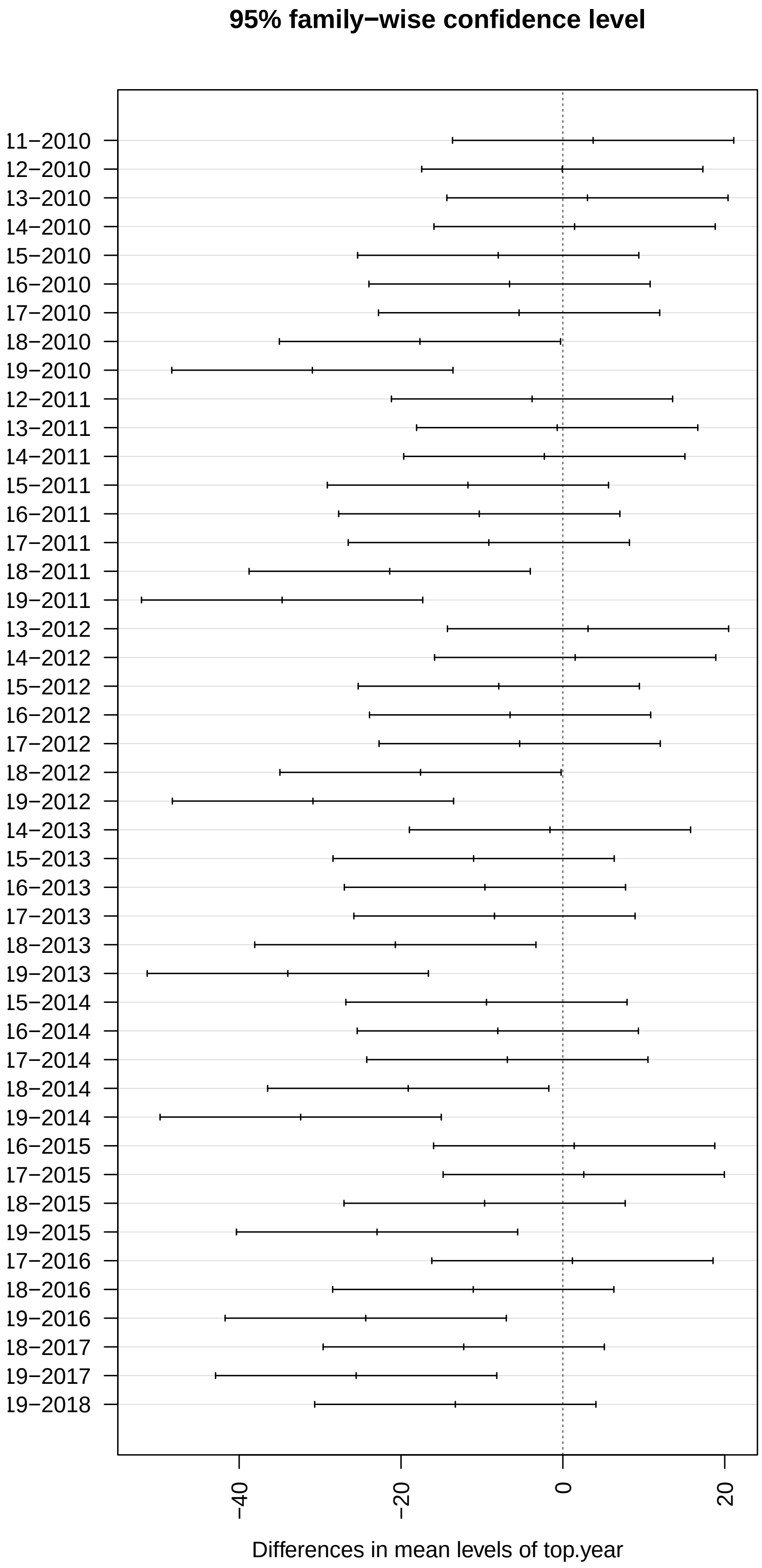
4.1.2 Tukey's Test

```
TukeyHSD(one.way, conf.level=.95)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = dur ~ top.year, data = df)
##
## $top.year
##              diff       lwr         upr     p adj
## 2011-2010   3.74 -13.63382  21.1138156 0.9996053
## 2012-2010  -0.07 -17.44382  17.3038156 1.0000000
## 2013-2010   3.04 -14.33382  20.4138156 0.9999297
## 2014-2010   1.45 -15.92382  18.8238156 0.9999999
## 2015-2010  -7.99 -25.36382   9.3838156 0.9079939
## 2016-2010  -6.59 -23.96382  10.7838156 0.9719491
## 2017-2010  -5.41 -22.78382  11.9638156 0.9929527
## 2018-2010 -17.66 -35.03382  -0.2861844 0.0426954
## 2019-2010 -30.95 -48.32382 -13.5761844 0.0000009
## 2012-2011  -3.81 -21.18382  13.5638156 0.9995408
## 2013-2011  -0.70 -18.07382  16.6738156 1.0000000
## 2014-2011  -2.29 -19.66382  15.0838156 0.9999938
## 2015-2011 -11.73 -29.10382   5.6438156 0.4988455
## 2016-2011 -10.33 -27.70382   7.0438156 0.6791154
## 2017-2011  -9.15 -26.52382   8.2238156 0.8122266
## 2018-2011 -21.40 -38.77382  -4.0261844 0.0039558
## 2019-2011 -34.69 -52.06382 -17.3161844 0.0000000
## 2013-2012   3.11 -14.26382  20.4838156 0.9999148
## 2014-2012   1.52 -15.85382  18.8938156 0.9999998
## 2015-2012  -7.92 -25.29382   9.4538156 0.9124889
## 2016-2012  -6.52 -23.89382  10.8538156 0.9738702
## 2017-2012  -5.34 -22.71382  12.0338156 0.9935971
## 2018-2012 -17.59 -34.96382  -0.2161844 0.0443919
## 2019-2012 -30.88 -48.25382 -13.5061844 0.0000010
## 2014-2013  -1.59 -18.96382  15.7838156 0.9999997
## 2015-2013 -11.03 -28.40382   6.3438156 0.5899049
## 2016-2013  -9.63 -27.00382   7.7438156 0.7616729
## 2017-2013  -8.45 -25.82382   8.9238156 0.8747845
## 2018-2013 -20.70 -38.07382  -3.3261844 0.0064421
## 2019-2013 -33.99 -51.36382 -16.6161844 0.0000000
## 2015-2014  -9.44 -26.81382   7.9338156 0.7823601
## 2016-2014  -8.04 -25.41382   9.3338156 0.9046933
## 2017-2014  -6.86 -24.23382  10.5138156 0.9635351
## 2018-2014 -19.11 -36.48382  -1.7361844 0.0181643
## 2019-2014 -32.40 -49.77382 -15.0261844 0.0000002
## 2016-2015   1.40 -15.97382  18.7738156 0.9999999
## 2017-2015   2.58 -14.79382  19.9538156 0.9999827
## 2018-2015  -9.67 -27.04382   7.7038156 0.7572125
## 2019-2015 -22.96 -40.33382  -5.5861844 0.0012497
## 2017-2016   1.18 -16.19382  18.5538156 1.0000000
## 2018-2016 -11.07 -28.44382   6.3038156 0.5847051
## 2019-2016 -24.36 -41.73382  -6.9861844 0.0004126
## 2018-2017 -12.25 -29.62382   5.1238156 0.4327797
## 2019-2017 -25.54 -42.91382  -8.1661844 0.0001539
## 2019-2018 -13.29 -30.66382   4.0838156 0.3115606
```

The pairs who have p-value < 0.05 have different means. Let's visualize this so it's easier to read.

```
plot(TukeyHSD(one.way, conf.level=.95), las = 2)
```


95% family−wise confidence level
11−2010
12−2010
13−2010
14−2010
15−2010
16−2010
17−2010
18−2010
19−2010
12−2011
13−2011
14−2011
15−2011
16−2011
17−2011
18−2011
19−2011
13−2012
14−2012
15−2012
16−2012
17−2012
18−2012
19−2012
14−2013
15−2013
16−2013
17−2013
18−2013
19−2013
15−2014
16−2014
17−2014
18−2014
19−2014
16−2015
17−2015
18−2015
19−2015
17−2016
18−2016
19−2016
18−2017
19−2017
19−2018
−40
−20
0
20
Differences in mean levels of top.year

On the plot, pairs whose confidence interval does not include zero have different means. So, 2010-2018, 2010-2019, 2011-2018, 2011-2019, 2012-2018, 2012-2019, 2013-2018, 2013-2019, 2014-2018, 2014-2019, 2015-2019, 2016-2019 and 2017-2019 have different means.

4.1.3 Kruskal-Wallis test

Now, I will perform Kruskal-Wallis test.

```
kruskal.test(dur ~ top.year, data = df)
```

```
##
##  Kruskal-Wallis rank sum test
##
## data:  dur by top.year
## Kruskal-Wallis chi-squared = 90.129, df = 9, p-value = 1.534e-15
```

As the p-value is less than the significance level 0.05, we can conclude that there are significant differences between the treatment groups. This is the result as we got from our ANOVA test.

4.2 Comparing The Means of Two samples

Now, I will pick two samples to compare their means. All the methods of comparing two samples' means that we learned in class have the following two assumptions:

1. The populations are **independent**

2. The populations have the **same variance**.

So I will check to see which pairs have equal variances.

```
num <- 0
for (level1 in levels(df$top.year)) {
  for (level2 in levels(df$top.year)) {
    if (level1 != level2) {
      res <- bartlett.test(dur[df$top.year == level1 | df$top.year == level2]
                           ~ top.year[df$top.year == level1 | df$top.year == level2],
                           data= df)
      if (res$p.value >= 0.05){
        cat(level1,'-', level2,'p-value is', res$p.value,'\n')
        num <- num + 1
      }
    }
  }
}
```

```
## 2010 - 2011 p-value is 0.7522003
## 2010 - 2012 p-value is 0.5317784
## 2010 - 2016 p-value is 0.3420933
## 2010 - 2017 p-value is 0.2292138
## 2010 - 2019 p-value is 0.07372916
## 2011 - 2010 p-value is 0.7522003
## 2011 - 2012 p-value is 0.3469292
## 2011 - 2016 p-value is 0.5255716
## 2011 - 2017 p-value is 0.1293499
## 2012 - 2010 p-value is 0.5317784
## 2012 - 2011 p-value is 0.3469292
## 2012 - 2016 p-value is 0.1157261
## 2012 - 2017 p-value is 0.5631334
## 2012 - 2018 p-value is 0.1353082
## 2012 - 2019 p-value is 0.2434974
## 2013 - 2014 p-value is 0.05032598
## 2013 - 2018 p-value is 0.2177058
## 2013 - 2019 p-value is 0.1188296
## 2014 - 2013 p-value is 0.05032598
## 2015 - 2016 p-value is 0.09992157
## 2016 - 2010 p-value is 0.3420933
## 2016 - 2011 p-value is 0.5255716
## 2016 - 2012 p-value is 0.1157261
## 2016 - 2015 p-value is 0.09992157
## 2017 - 2010 p-value is 0.2292138
## 2017 - 2011 p-value is 0.1293499
## 2017 - 2012 p-value is 0.5631334
## 2017 - 2018 p-value is 0.3589842
## 2017 - 2019 p-value is 0.5557929
## 2018 - 2012 p-value is 0.1353082
## 2018 - 2013 p-value is 0.2177058
## 2018 - 2017 p-value is 0.3589842
## 2018 - 2019 p-value is 0.7424198
## 2019 - 2010 p-value is 0.07372916
## 2019 - 2012 p-value is 0.2434974
## 2019 - 2013 p-value is 0.1188296
## 2019 - 2017 p-value is 0.5557929
## 2019 - 2018 p-value is 0.7424198
```

```
cat('There are', num, 'pairs of groups who satisfy the homogeneity of variance.')
```

```
## There are 38 pairs of groups who satisfy the homogeneity of variance.
```

I will pick the years 2010 and 2011. If these samples come from normal distributions, I can use the z-test.

```
# shapiro.test( df$dur[df$top.year == 2010 | df$top.year == 2011] )
```

```
shapiro.test( df$dur[df$top.year == 2010 ] )
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$dur[df$top.year == 2010]
## W = 0.96838, p-value = 0.01664
```

```
shapiro.test( df$dur[df$top.year == 2011] )
```

```
##
##  Shapiro-Wilk normality test
##
## data:  df$dur[df$top.year == 2011]
## W = 0.95388, p-value = 0.001509
```

They do not come from normal distributions but I still can use the z-test. If the number of samples is greater than 30 (here it's 100), central limit theorem (CLT) tells us that we can use the z-test.

However, I was not able to find a function in R that performs the two sample z-test when the variances of populations are unknown. But just as an experiment, let's try the z-test with the sample variances as population variances.

```
library(BSDA)

z.test(x = df$dur[df$top.year == 2010],
       y = df$dur[df$top.year == 2011],
       alternative = "two.sided",
       mu = 0,
       sigma.x = var(df$dur[df$top.year == 2010]),
       sigma.y = var(df$dur[df$top.year == 2011]),
       conf.level = 0.95
)
```

```
##
##  Two-sample z-Test
##
## data:  df$dur[df$top.year == 2010] and df$dur[df$top.year == 2011]
## z = -0.02475, p-value = 0.9803
## alternative hypothesis: true difference in means is not equal to 0
## 95 percent confidence interval:
##  -299.9077  292.4277
## sample estimates:
## mean of x mean of y
##    226.45    230.19
```

The p-value is greater than 0.05. So if this was a valid test (i.e. its assumptions were satisfied), we would have failed to reject the null hypothesis, meaning we would have failed to reject that the means of the two groups are not equal.

Note that the even though the t-test does not require the variances to be known, it still requires the samples to come from normal populations. So the 2010-2011 pair is not a valid pair for the t-test either. Let's see if there any two groups that come from normal distributions.

```
num <- 0
for (level in levels(df$top.year) ) {
  res <- shapiro.test( df$dur[df$top.year == 2011] )
  if (res$p.value >= 0.05) {
    cat('Group', level, 'comes from normal distribution.', '\n')
    num <- num +1
  }
}
cat("There are", num, "groups with normal population.")
```

```
## There are 0 groups with normal population.
```

4.3 Categorical Data Analysis

Now, let's try to answer this question: Are the distributions of genres the same over the years? I.e we consider genre and the years as our two categorical variables/features. To answer this question, I will perform Chi-square test of homogeneity.

```
df$top.genre = as.factor(df$top.genre)

chisq.test(df$top.year, df$top.genre)
```

```
## Warning in chisq.test(df$top.year, df$top.genre): Chi-squared approximation may
## be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  df$top.year and df$top.genre
## X-squared = 1593.5, df = 1179, p-value = 6.059e-15
```

The p-value is much less than the significance level of 0.05. So we reject the null hypothesis that there is not any correlation between the two features. However, When we don't trust the validity of the asymptotic approximation (see Warning), the permutation approach that is implemented in the `chisq.test()` function is safer to use.

```
chisq.test(df$top.year, df$top.genre, simulate.p.value = TRUE, B = 10000)
```

```
##
##  Pearson's Chi-squared test with simulated p-value (based on 10000
##  replicates)
##
## data:  df$top.year and df$top.genre
## X-squared = 1593.5, df = NA, p-value = 9.999e-05
```

Notice that the second p-value is almost 11 orders of magnitude larger that the first p-value. However, it still is less than 0.05. So we reject the null hypothesis again.

4.4 Regression

4.4.1 Multiple linear regression

4.4.1.1 Finding and fitting the best model Would it be cool if we could predict the popularity of a song just from its attributes without knowing who the singer was? or even what the genre was? So let's do it. I will use the following numeric variables to predict `pop` which represents the popularity of a song:

`bpm` : Beats Per Minute - The tempo of the song
`nrgy` : Energy - How energetic the song is
`dnce` : Danceability - How easy it is to dance to the song
`dB` : Decibel - How loud the song is
`live` : How likely the song is a live recording
`val` : How positive the mood of the song is
`dur` : Duration of the song
`acous` : How acoustic the song is
`spch` : The more the song is focused on spoken word

To decide which of these variables are useful for our multiple regression model, I will use best subset selection. We can also use forward or backward step wise selection. Best subset selection will be computationally expensive when the number of variables and number of observations are large. However, since our dataset is not very large, best subset selection gauntness we can get the best model based on the criteria that we have chosen.

This bring us to the next consideration; We need a criteria for choosing the best model. We can chose among adjusted-R^2, C_p, BIC or AIC.

Finally, we need to compute these metrics for our model on the validation dataset and not the training dataset as the model will likely overfit the training dataset. To do this, I use k-fold cross validation.

We utilize the following two helper functions for this analysis:

`get_model_formula()` allowing to access easily the formula of the models returned by the function regsubsets()

`get_cv_error()` to get the cross-validation (CV) error for a given model

```r
# id: model id
# object: regsubsets object
# data: data used to fit regsubsets
# outcome: outcome variable
get_model_formula <- function(id, object, outcome){
  # get models data
  models <- summary(object)$which[id,-1]
  # Get outcome variable
  #form <- as.formula(object$call[[2]])
  #outcome <- all.vars(form)[1]
  # Get model predictors
  predictors <- names(which(models == TRUE))
  predictors <- paste(predictors, collapse = "+")
  # Build model formula
  as.formula(paste0(outcome, "~", predictors))
}
```

```r
get_cv_error <- function(model.formula, data){
  set.seed(1)
  train.control <- trainControl(method = "cv", number = 5)
  cv <- train(model.formula, data = data, method = "lm",
              trControl = train.control)
  cv$results$RMSE
}
```

Now we can use the above defined helper functions to compute the prediction error of the different best models returned by the `regsubsets()` function.

```r
library(purrr)
library(leaps)
library(caret)

models <- regsubsets(pop~.,
                     data = df[c('pop', 'bpm', 'nrgy', 'dnce', 'dB', 'live', 'val', 'dur', 'acous', 'spc
                     nvmax = 9)

model.ids <- 1:9

cv.errors <-  map(model.ids, get_model_formula, models, 'pop') %>%
  map(get_cv_error, data = df[c('pop', 'bpm', 'nrgy', 'dnce', 'dB', 'live', 'val', 'dur', 'acous', 'spch
  unlist()

cv.errors
```

```
## [1] 8.568264 8.544193 8.514785 8.508143 8.508583 8.537637 8.546143 8.560717
## [9] 8.582918
```

Now let's pick the model with lowest corss validation error.

```r
cat('Model with', which.min(cv.errors), 'variables is the best model.')
```

```
## Model with 4 variables is the best model.
```

It can be seen that the model with 4 variables is the best model. It has the lower prediction error. The regression coefficients of this model can be extracted as follow:

```
coef(models, 4)
```

```
## (Intercept)        nrgy        live         val        spch
## 83.14716436 -0.13569329 -0.06315206  0.03680878  0.03788783
```

```
# res.sum <- summary(models)
# data.frame(
#   Adj.R2 = which.max(res.sum$adjr2),
#   CP = which.min(res.sum$cp),
#   BIC = which.min(res.sum$bic)
# )

# summary(models)$which[2,-1]
```

```
lm.fit <- lm(pop ~ nrgy + live + val + spch,
             data = df[c('pop', 'nrgy', 'live', 'val', 'spch')])

summary(lm.fit)
```

```
##
## Call:
## lm(formula = pop ~ nrgy + live + val + spch, data = df[c("pop",
##     "nrgy", "live", "val", "spch")])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.395  -4.912   0.785   5.960  21.889
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 83.14716    1.29021  64.445  < 2e-16 ***
## nrgy        -0.13569    0.01862  -7.287 6.46e-13 ***
## live        -0.06315    0.02043  -3.091  0.00205 **
## val          0.03681    0.01347   2.732  0.00641 **
## spch         0.03789    0.02923   1.296  0.19528
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.498 on 995 degrees of freedom
## Multiple R-squared:  0.07288,    Adjusted R-squared:  0.06915
## F-statistic: 19.55 on 4 and 995 DF,  p-value: 1.665e-15
```

The first step in interpreting the multiple regression analysis is to examine the F-statistic and the associated p-value, at the bottom of model summary.

Here, it can be seen that p-value of the F-statistic is much less than 0.05, which is highly significant. This means that, at least, one of the predictor variables is significantly related to the outcome variable.

Let's look at the coefficients table to see which coefficients are significant.

```
summary(lm.fit)$coefficient
```

```
##                Estimate Std. Error   t value     Pr(>|t|)
## (Intercept) 83.14716436 1.29021073 64.444639 0.000000e+00
## nrgy        -0.13569329 0.01862206 -7.286694 6.460256e-13
## live        -0.06315206 0.02042967 -3.091192 2.049010e-03
## val          0.03680878 0.01347302  2.732037 6.405861e-03
## spch         0.03788783 0.02923449  1.295998 1.952768e-01
```

For a given predictor, the t-statistic evaluates whether or not there is significant association between the predictor and the outcome variable, that is whether the beta coefficient of the predictor is significantly different from zero.

It can be seen that, changing in `nrgy`, `live` and `val` are significantly associated to changes in `pop` while changes in `spch` is not significantly associated with sales.

For a given predictor variable, the coefficient $\hat{\beta}$ can be interpreted as the average effect on y of a one unit increase in predictor, holding all other predictors fixed.

We found that `spch` is not significant in the multiple regression model. This means that, for a fixed amount of the other three variables, changes in the `spch` will not significantly affect `pop`. So we remove it and fit the model again.

```
lm.fit <- lm(pop ~ nrgy + live + val,
             data = df[c('pop', 'nrgy', 'live', 'val')])

summary(lm.fit)
```

```
##
## Call:
## lm(formula = pop ~ nrgy + live + val, data = df[c("pop", "nrgy",
##     "live", "val")])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -39.321  -4.847   0.792   5.909  21.681
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 83.65276    1.23024  67.997  < 2e-16 ***
## nrgy        -0.13867    0.01849  -7.501  1.4e-13 ***
## live        -0.06141    0.02039  -3.011  0.00267 **
## val          0.03781    0.01346   2.810  0.00505 **
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 8.501 on 996 degrees of freedom
## Multiple R-squared:  0.07132,    Adjusted R-squared:  0.06852
## F-statistic: 25.49 on 3 and 996 DF,  p-value: 6.792e-16
```

And the confidence interval of our model's coefficients are

```
confint(lm.fit)
```

```
##                   2.5 %      97.5 %
## (Intercept) 81.23860309 86.06691113
## nrgy        -0.17494413 -0.10239045
## live        -0.10142478 -0.02139133
## val          0.01141021  0.06421802
```

4.4.1.2 Model accuracy assessment The overall quality of the model can be assessed by examining the R-squared (R2) and Residual Standard Error (RSE).

R-squared R^2:

In multiple linear regression, R^2 represents the correlation coefficient between the observed values of the outcome variable y and the fitted (i.e., predicted) values of y. For this reason, the value of R will always be positive and will range from zero to one.

R^2 represents the proportion of variance, in the outcome variable y, that may be predicted by knowing the value of the x variables. An R^2 value close to 1 indicates that the model explains a large portion of the variance in the outcome variable.

However, a problem with the R^2, is that, it will always increase when more variables are added to the model, even if those variables are only weakly associated with the response. A solution is to adjust the R^2 by taking into account the number of predictor variables.

The adjustment in the "Adjusted R Square" value in the summary output is a correction for the number of x variables included in the prediction model.

Here, with `nrgy`, `live` and `val` predictor variables, the adjusted R^2 is 0.068, meaning that 6.8% of the variance in the measure of `pop` can be predicted by `nrgy`, `live` and `val`. So we need to check if assumptions of linear regression hold.

Residual Standard Error (RSE), or sigma:

The RSE estimate gives a measure of error of prediction. The lower the RSE, the more accurate the model (on the data in hand).

The error rate can be estimated by dividing the RSE by the mean outcome variable:

```
sigma(lm.fit)/mean(df$pop)
```

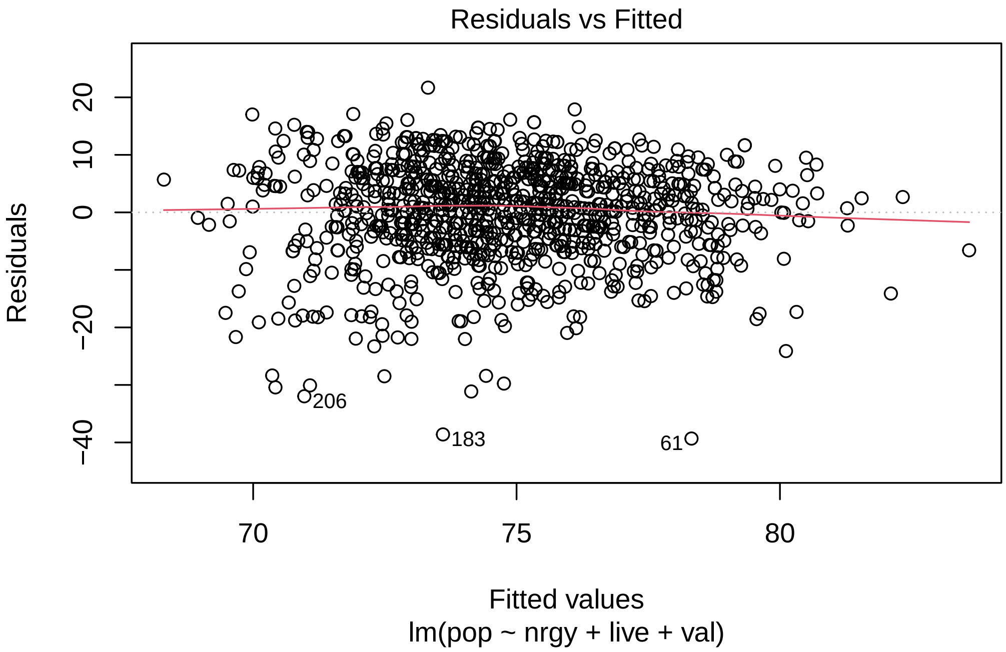
```
## [1] 0.1135854
```

The error rate is 11%.

4.4.1.3 Checking linear regression assumptions and diagnostics

4.4.1.3.A Linearity of the data The linearity assumption can be checked by inspecting the Residuals vs Fitted plot.

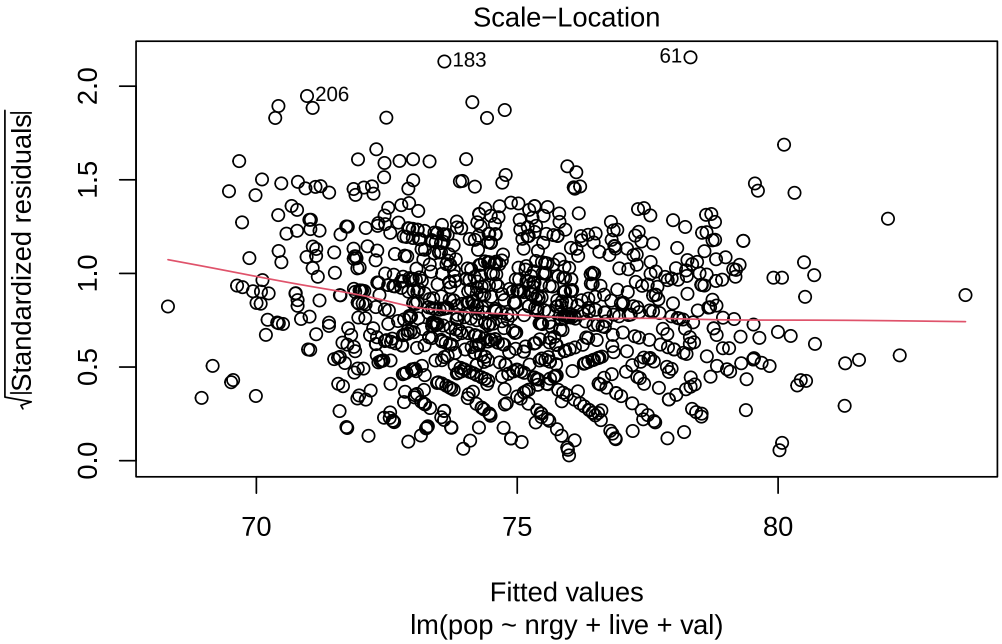
```
plot(lm.fit,1)
```



Ideally, the residual plot will show no fitted pattern. That is, the red line should be approximately horizontal at zero. The presence of a pattern may indicate a problem with some aspect of the linear model. Here, it seems like the residuals are larger for lower values of fitted line.

4.4.1.3.B Homogeneity of variance This assumption can be checked by examining the **scale-location** plot, also known as the **spread-location** plot.

```
plot(lm.fit,3)
```

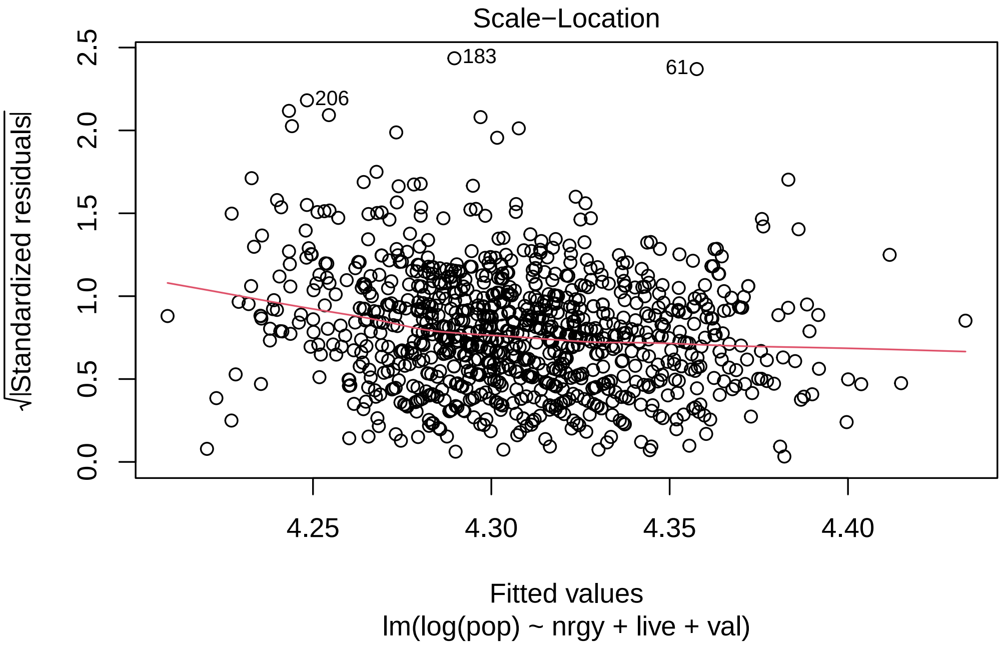


This plot shows if residuals are spread equally along the ranges of predictors. It's good if we see a horizontal line with equally spread points.

Here, this is not quite the case. It can be seen that the variability (variances) of the residual points is slightly higher for lower (approximately less than 73) fitted outcome variable, suggesting non-constant variances in the residuals errors (or heteroscedasticity).

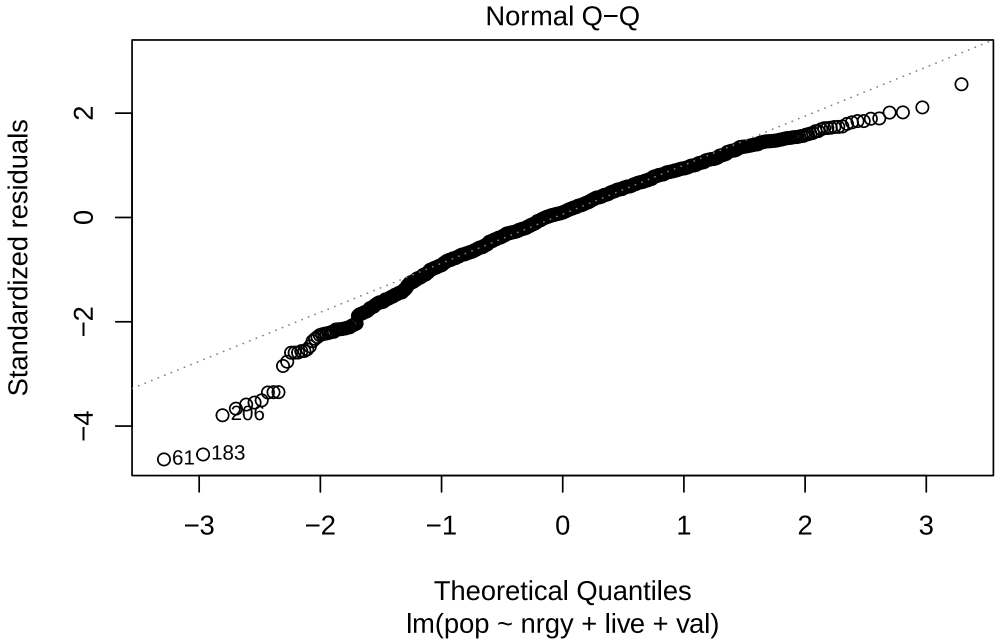
Diagnostic: A possible solution to reduce the heteroscedasticity problem is to use a log or square root transformation of the outcome variable.

```
lm.fit2 <- lm(log(pop) ~ nrgy + live + val,
              data = df[c('pop', 'nrgy', 'live', 'val')])

plot(lm.fit2, 3)
```



4.4.1.3.C Normality of residuals The QQ plot of residuals can be used to visually check the normality assumption.

```
plot(lm.fit,2)
```



The normal probability plot of residuals should approximately follow a straight line.

Here,there are significant deviations at both ends. This indicates non-normality.

4.4.1.3.D Outliers and high levarage points **Outliers:**

An outlier is a point that has an extreme outcome variable value. The presence of outliers may affect the interpretation of the model, because it increases the RSE.

Outliers can be identified by examining the **standardized residual** (or **studentized residual**), which is the residual divided by its estimated standard error. Standardized residuals can be interpreted as the number of standard errors away from the regression line.

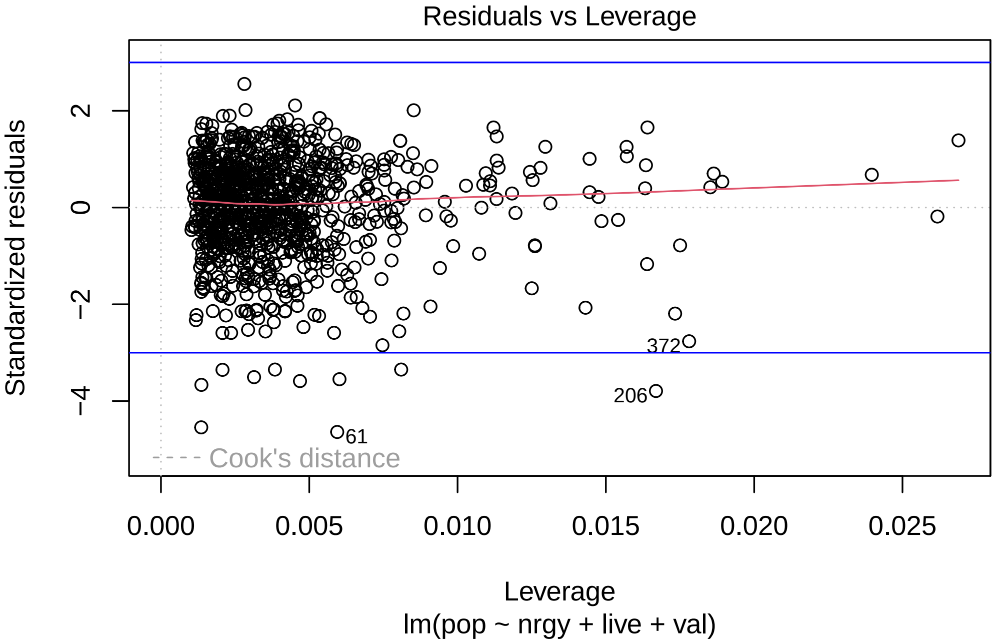
Observations whose standardized residuals are greater than 3 in absolute value are possible outliers.

High leverage points:

A data point has high leverage, if it has extreme predictor x values. This can be detected by examining the **leverage statistic** or the **hat-value**. A value of this statistic above $2(p+1)/n$ indicates an observation with high leverage; where, p is the number of predictors and n is the number of observations.

Outliers and high leverage points can be identified by inspecting the Residuals vs Leverage plot:

```
plot(lm.fit,5)
abline(h=c(-3,3), col=c("blue", "blue"))
```



```
# abline(v= 0.08, col= 'green')
```

The plot shows that there are 10 points with standardized residuals below -3.

However, there is no high leverage point in the data. That is, all data points, have a leverage statistic below $2(p+1)/n = 8/100 = 0.08$.

4.4.1.3.E Influential values An influential value is a value, which inclusion or exclusion can alter the results of the regression analysis. Such a value is associated with a large residual.

Not all outliers (or extreme data points) are influential in linear regression analysis.

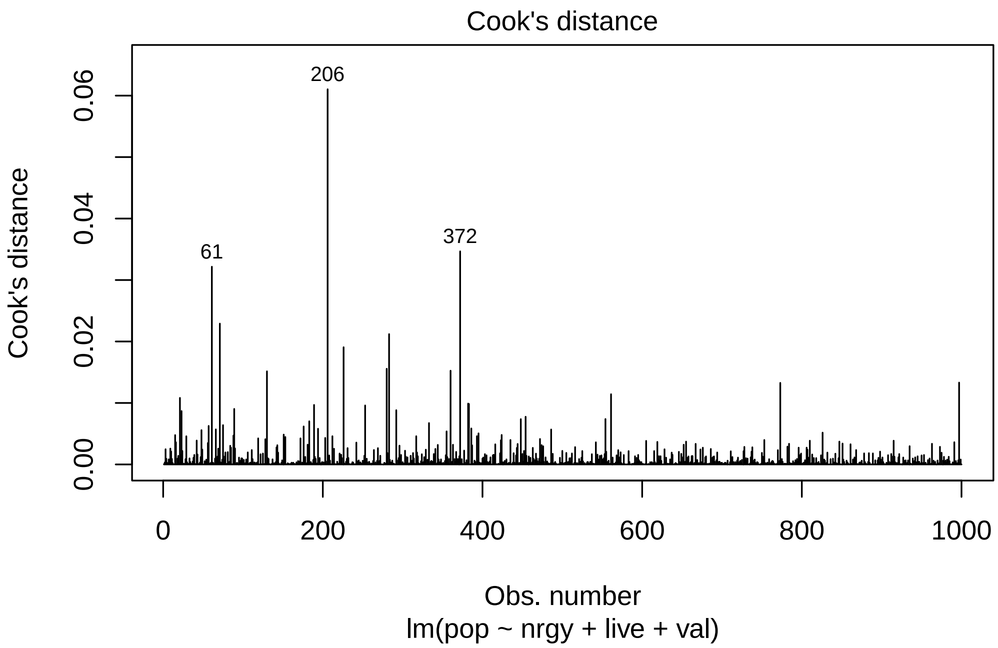
Statisticians have developed a metric called **Cook's distance** to determine the influence of a value. This metric defines influence as a combination of *leverage* and *residual size*.

A rule of thumb is that an observation has high influence if Cook's distance exceeds $4/(n-p-1)$, where n is the number of observations and p the number of predictor variables.

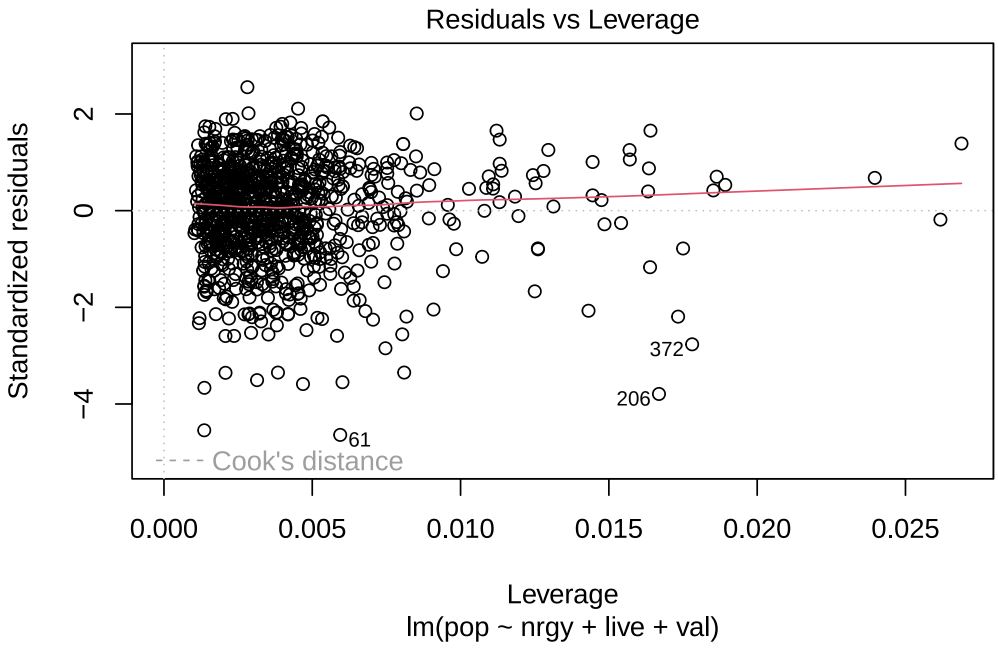
The Residuals vs Leverage plot can help us to find influential observations if any. On this plot, outlying values are generally located at the upper right corner or at the lower right corner. Those spots are the places where data points can be influential against a regression line.

The following plots illustrate the Cook's distance and the leverage of our model:

```
# Cook's distance
plot(lm.fit, 4)
```

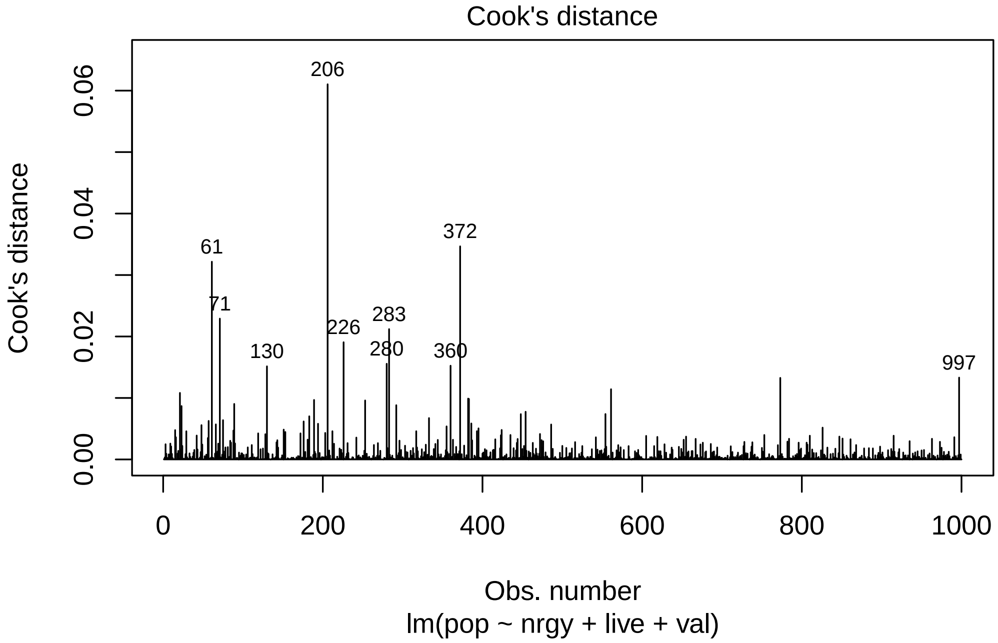


```
# Residuals vs Leverage
plot(lm.fit, 5)
```



By default, the top 3 most extreme values are labelled on the Cook's distance plot. Let's take a look at top 10 extreme values:

```
plot(lm.fit, 4, id.n = 10)
```



Let’s also take a look at these observations:

```r
library(rstatix)
library(dplyr)

lm.fit.diag.metrics <- augment(lm.fit)

lm.fit.diag.metrics %>%
  top_n(10, wt = .cooksd)
```

```
## # A tibble: 10 x 10
##      pop  nrgy  live   val .fitted .resid    .hat .sigma .cooksd .std.resid
##    <int> <int> <int> <int>   <dbl>  <dbl>   <dbl>  <dbl>   <dbl>      <dbl>
##  1    39    54    16    83    78.3  -39.3 0.00594   8.41  0.0322      -4.64
##  2    42    98    13    29    70.4  -28.4 0.00810   8.46  0.0229      -3.35
##  3    40    93    35    48    70.4  -30.4 0.00469   8.45  0.0152      -3.59
##  4    39    79    70    68    71.0  -32.0 0.0167    8.44  0.0610      -3.79
##  5    41    96    12    39    71.1  -30.1 0.00602   8.45  0.0191      -3.55
##  6    52    83    65    35    69.5  -17.5 0.0143    8.49  0.0156      -2.07
##  7    52    71    69    24    70.5  -18.5 0.0173    8.48  0.0212      -2.19
##  8    56    29    10    29    80.1  -24.1 0.00747   8.47  0.0152      -2.85
##  9    49    79    65    95    72.3  -23.3 0.0178    8.47  0.0347      -2.77
## 10    85    50    80    41    73.4   11.6 0.0269    8.50  0.0133       1.39
```

When data points have high Cook’s distance scores and are to the upper or lower right of the leverage plot, they have leverage meaning they are influential to the regression results. The regression results will be altered if we exclude those cases.

In our example, the data don’t present any influential points. Cook’s distance lines (a red dashed line) are not shown on the Residuals vs Leverage plot because all points are well inside of the Cook’s distance lines.

4.4.2 Simple linear regression

Now let's repeat the analysis in previous section. But now we would like to find the one variable that can best predict the popularity of a song and see if it satisfies the assumptions of linear regression.

```
library(purrr)
library(leaps)
library(caret)

models <- regsubsets(pop~.,
                     data = df[c('pop', 'bpm', 'nrgy', 'dnce', 'dB', 'live', 'val', 'dur',
                                 'acous', 'spch')],
                     nvmax = 1)

model.ids <- 1

cv.errors <-  map(model.ids, get_model_formula, models, 'pop') %>%
  map(get_cv_error, data = df[c('pop', 'bpm', 'nrgy', 'dnce', 'dB', 'live', 'val', 'dur',
                                'acous', 'spch')]) %>%
  unlist()

cv.errors
```

```
## [1] 8.568264
```

```
coef(models, 1)
```

```
## (Intercept)        nrgy 
##  83.8149423  -0.1291321
```

So, the variable `nrgy` which describes how energetic the song is produces the lowest cross validation error. Now let's a look take a look at the model fit.

```
lm.fit <- lm(pop ~ nrgy, data = df)

summary(lm.fit)
```

```
## 
## Call:
## lm(formula = pop ~ nrgy, data = df)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -39.130  -4.710   0.998   5.908  21.645 
## 
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 83.81494    1.21102  69.210  < 2e-16 ***
## nrgy        -0.12913    0.01698  -7.604 6.62e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## Residual standard error: 8.568 on 998 degrees of freedom
## Multiple R-squared:  0.05476,    Adjusted R-squared:  0.05381 
## F-statistic: 57.82 on 1 and 998 DF,  p-value: 6.623e-14
```

The p-value is less than 0.05.

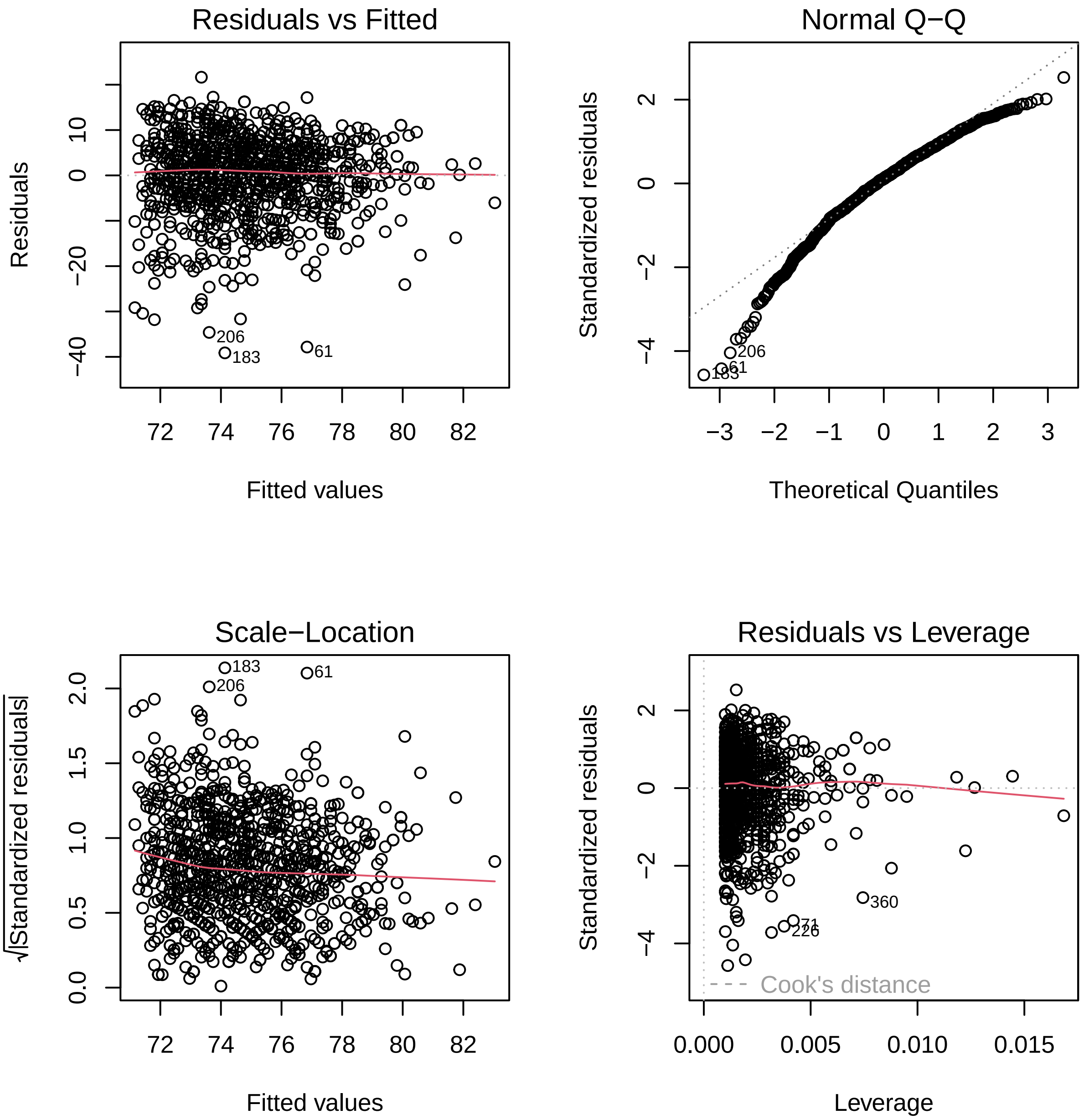
We can also obtain confidence internals for the coefficients.

```
confint(lm.fit)
```

```
##                   2.5 %      97.5 %
## (Intercept) 81.4385090 86.19137568
## nrgy        -0.1624578 -0.09580651
```

We can check the assumptions of liear regression the same way that we did for multiple linear regression. So I don't go in details. But we can have all the plots discussed for checking the assumption in a compact way as folows:

```
par(mfrow = c(2, 2))
plot(lm.fit)
```



5 Conclusion

In this project, I analyzed Spotify Top 100 Songs of 2010-2019 dataset from the standpoint of the statistical methods that we learned in class. I posed several questions about the dataset and answered them using statistical techniques. For example, I wondered if the means of durations of the songs are differnet for diffenert years, and if yes, for what pairs of years. And whether we can use different attributes of a song to predict its popularity. There are many more questions that we can ask and answer about this dataset with statistical methods used here.