# Analysis of Music Ratings and Loudness

Inwoo Choi, Brad Robinson, Rupal Gandhi

**final_report.rmd = Our main report**
**Appendix_A.rmd = Data wrangling and cleaning for the first two analysis (section 3.1 and 3.2)**
**Appendix_B.rmd = Data wrangling and cleaning for the third analysis (section 3.3)**

# 1.0 Introduction

In this report, we are interested in analyzing how users of MusicBrainz feel about the music they listen to. A user can submit ratings as well as submit informative tags for artists, albums and songs. We want to determine if the users' music preference is determined by the artists' main genre and the year the artists started. Music taste is subjective and we want to see how people feel about the artists they listen to.

In the second analysis, we will analyze loudness data provided by Acousticbrainz in order to determine whether or not music is getting louder over time. It is claimed that music is getting louder in order to attract the attention of listeners in an increasingly competitive market. This has come to be known as the "Loudness War". This is a major issue because as music gets louder during the mastering stage, the audio will lose dynamic range (the variation in the loudness in the song) and the quality and complexity of a song will decrease. While I will not measure the variation of the loudness in a song directly, I can infer that lowering of dynamic range will happen as the music gets louder during mastering as there are limits to how loud you can make a song. This means that the high loudness points in a song will cut out and be capped but parts of the song that is relatively lower in loudness will increase. The resulting song that has had it's loudness pushed way up will sound compressed, distorted, and the quality will suffer.

For the analysis we will be employing various sampling techniques to the data as the original dataset is very large. We will then use various statistical tests on the data to answer the aforementioned questions.

First, let's introduce the data:

# 2.0 Overview of the Dataset

# 2.1 MusicBrainz

https://musicbrainz.org/ (https://musicbrainz.org/)

MusicBrainz is an open-source, community-maintained encylopedia of music information. The dataset they provide is quite massive so I will only briefly gloss over some of the information that is contained and highlight the information that we will use to answer our questions.

## 2.1.1 Original Dataset

The original data contains information on the following:

- **Artists**: Bands and Artists
    - Name, sort name, IPI, aliases, type, begin and end dates, disambiguation comment, MBID
- **Release Groups**: Albums that are released in multiple versions and countries are grouped here
    - Title, artist credit, type, disambiguation comment, MBID

- **Releases**: Individual albums
    - Title, artist credit, type, status, language, date, country, label, catalog number, barcode, medium(s), disc ID(s), ASIN, disambiguation comment, MBID
- **Mediums**: Medium of distribution (ie. CD, Vinyl, Digital, etc.)
    - Format, list of tracks (title, artist credit, duration)
- **Works**
    - Title, ISWC, relationships, disambiguation comment, MBID
- **Labels**
    - Name, sort name, aliases, country, type, code, begin and end dates, disambiguation comment, MBID
- **Relationships and URLS**
    - Relationships are a way to link the above entities together and allow MusicBrainz to capture most of the data contained in the liner notes of a CD.
- **CD stubs**
    - Title, artist, barcode, disc ID, disambiguation comment

For further look at the database follow the provided link to see the database schema to see the relations:

https://musicbrainz.org/doc/MusicBrainz_Database/Schema
(https://musicbrainz.org/doc/MusicBrainz_Database/Schema)

We cannot upload the entire dataset to D2L as these files are too large but we will provide clean data files that we used for our analysis. But if you want to see the original data follow the link provided and download both **mbdump.tar.bz2** and **mbdump-derived.tar.bz2**.

https://musicbrainz.org/doc/MusicBrainz_Database/Download
(https://musicbrainz.org/doc/MusicBrainz_Database/Download)

# 2.1.2 Cleaned Data and a simple exploratory data analysis

**SEE: Appendix_A.rmd**

I have provided a separate rmd file that contains the process of merging and cleaning the data (**Appendix_A.rmd**) but in this portion of the report we will give a brief description of the process:

For the analysis of Music ratings, we were mainly interested in the Artist table which is contained in **mbdump.tar.bz2** archive. However, in order to obtain ratings and tag data we have to unzip the **mbdump-derived.tar.bz2** to obtain:

- Artist_meta which contains the review scores
- Artist_tag which contains the artist tags (genres) in terms of a numerical ID
- Tag table which contains the names of the genres in a string which references and merges the numerical tag ID

And we know that:

- The Genre is categorical data
- the years are continuous but for the purposes of comparison we can group them into categories
- Ratings are continuous and is the response variable

Now, an artist can have multiple genres associated with it, so the main genre was determined by the number of “upvotes” for a given tag (genre). The number of upvotes for a given tag is recorded in the column **ref_count** in the table **tag**. In case of a tie in upvotes, I have selected the genre/tag at random.

Refer to the referenced rmd file for more details but from here I can give a brief exploratory description of the data:

- We have over 21.6 million songs in the database (this number is given as a rough value as the number will change as MusicBrainz updates the database which they do quite frequently)
- 1.60 million artists
- around 419 distinct Genres.

# 2.2 AcousticBrainz

https://acousticbrainz.org/ (https://acousticbrainz.org/)

## 2.2.1 Brief description

AcousticBrainz is a crowd-sourced database containing low-level spectral information of 4.6 million songs. I will not go over the contents of the database as this dataset is both complex and lengthy but the following link contains the description of the information that is contained:

https://acousticbrainz.org/data (https://acousticbrainz.org/data)

As you can see, the data for just a single song is a large JSON document. However, the only value of interest is “average_loudness”.

## 2.2.2 Obtaining and cleaning data

**SEE: Appendix_B.rmd**
Unlike MusicBrainz, AcousticBrainz does not have an up-to-date data-dump as the entire low-level spectral data of all their songs is around 34GB compressed. The last data-dump is not maintained and is outdated as it only contains data up to 2015. While using an older dataset is not of concern for our analysis, downloading and working with a 34GB archive is simply not feasible.

To obtain the relevent data, we have to use WEB API methods using the ‘httr’ library for R. The following link is the documentation for the use of WEB API for AcousticBrainz:

https://acousticbrainz.readthedocs.io/api.html (https://acousticbrainz.readthedocs.io/api.html)

To understand how we obtained data through R via the ‘httr’ package refer to our rmd file (**Appendix_B.rmd**).

The WEB API needs the MBID for a given recording and once the data is obtained we can merge it with the recording.gid (contains the MBID) column in the **recording** table provided by **mbdump.tar.bz2**. The process will be described in more detail in the loudness analysis section as we first have to find a way to sample recordings by time as they are released by albums and not by individual songs. This requires the merging and cleaning of multiple tables provided by **mbdump.tar.bz2**.

Here we know that:

- Years are continuous but when we perform the Mantel-Haenszel test the years will be clustered into decades which makes them ordinal for that test
- Average_loudness is also continuous but like with years we can cluster the data into ordinal data.

# 3.0 Data Analysis

Here we will answer the questions that we have proposed earlier:

1. Are there differences in ratings for different Genres?
2. Are there differences in ratings for artists that started out in a given time period?

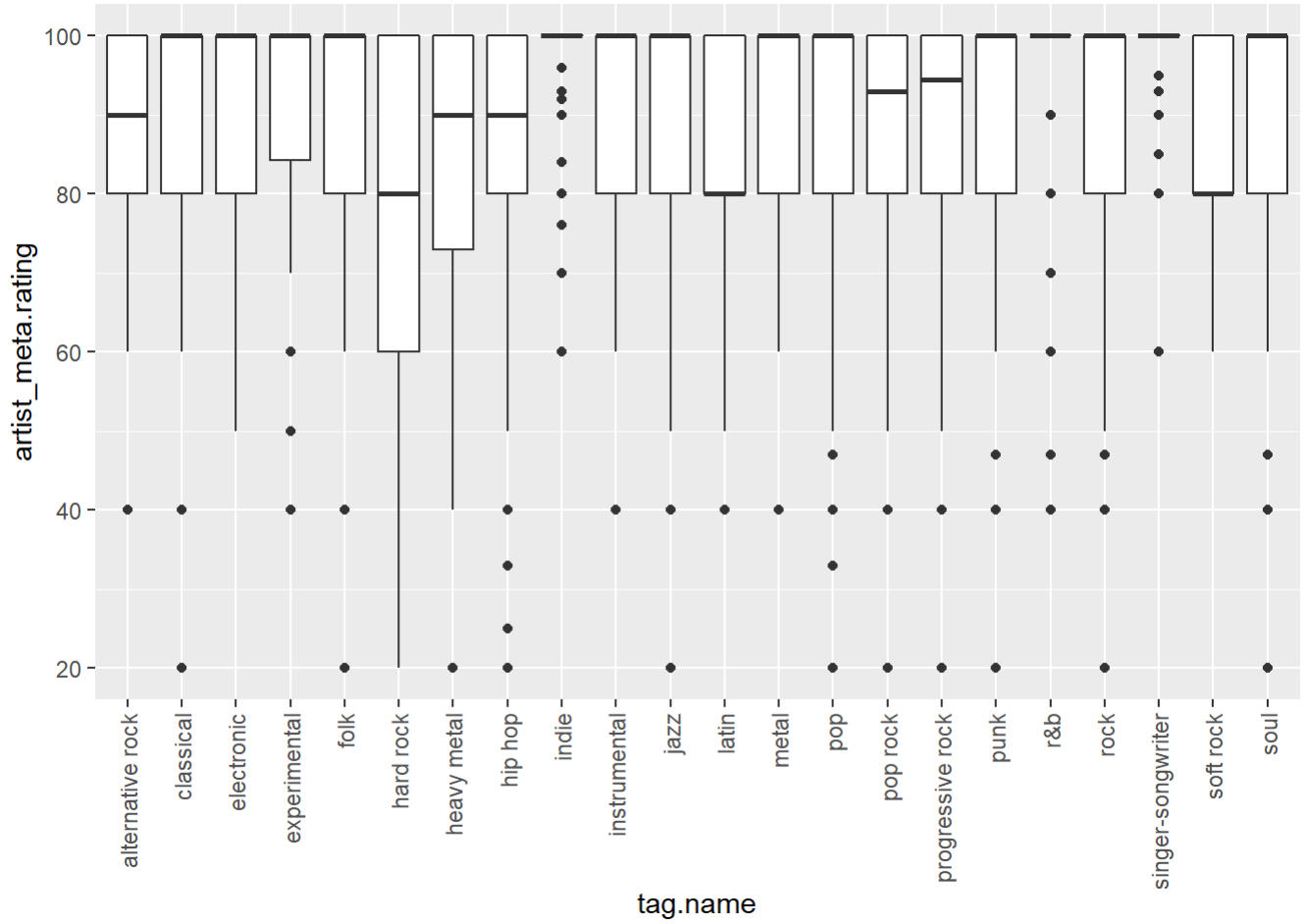
3. Is music getting louder?

# 3.1 Ratings by Genre

```
artist_data = read.csv('artist_rating_genre_feb17.csv')

#head(artist_data)
```

```
artist_box <- ggplot(artist_data, aes(x=tag.name, y=artist_meta.rating)) +
  geom_boxplot() +ggpubr::rotate_x_text()
artist_box
```




## The mean of all artist_meta.rating's is calculated:

```
mean(artist_data$artist_meta.rating)
```

```
## [1] 88.29761
```

## A simple random sample of 500 is created from the data:

```
srs_ad <- artist_data[sample(nrow(artist_data),500),]
```

## The mean of the SRS:artist_meta.rating's is calculated:

```
mean(srs_ad$artist_meta.rating)
```

```
## [1] 87.302
```

## A stratified random sample is created from the population:

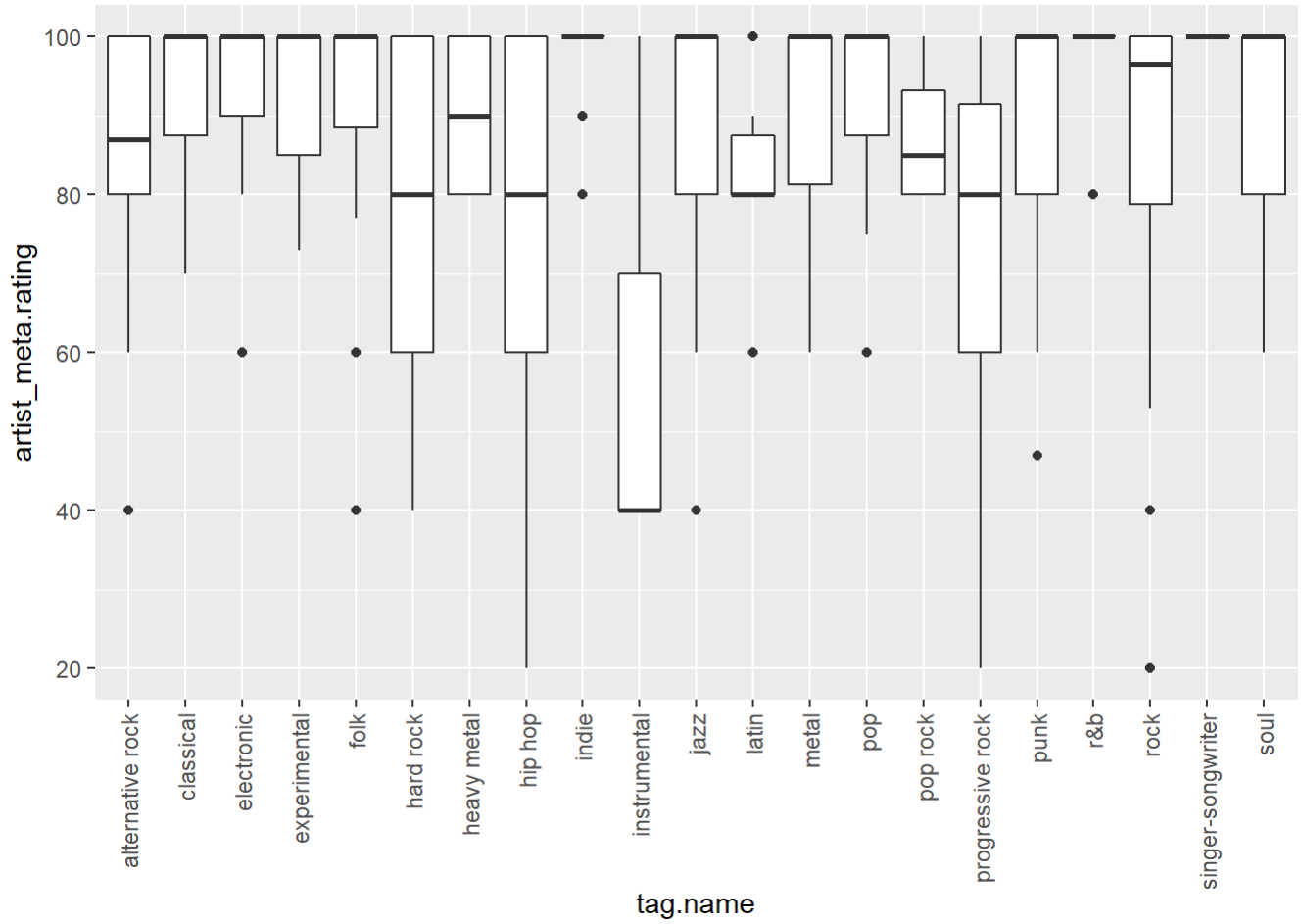
```
st=strata(artist_data,stratanames=c("tag.name"),size=c(20,20,10,20,20,20,20,20,6,20,10,20,7,20,2
0,20,20,10,9,10,20,5), method="srswor")
# extracts the observed data
df <- getdata(artist_data, st)
# see the result using a contingency table
#table(st$tag.name)

#tbl=table(st$tag.name)
head(df)
```

```
##    ï..10 artist.id      artist.name artist.sort_name artist.begin_date_year
## 11   153       293    Soul Coughing    Soul Coughing                  1993
## 14   178       394             CAKE             CAKE                  1992
## 15   183       418    Primal Scream    Primal Scream                  1982
## 38   572      1857 Marcy Playground Marcy Playground                  1997
## 56   839      2711          Warrant          Warrant                  1984
## 64  1294      9044        PJ Harvey       Harvey, PJ                  1969
##    artist.end_date_year artist.type artist.area artist.gender artist.ended
## 11                 2000           2         222            NA            t
## 14                   NA           2         222            NA            f
## 15                   NA           2         221            NA            f
## 38                   NA           2         222            NA            f
## 56                   NA           2         222            NA            f
## 64                   NA           1         221             2            f
##    artist.begin_area artist.end_area artist_meta.rating
## 11              7020              NA                100
## 14              7328              NA                 80
## 15              3855              NA                 87
## 38                NA              NA                 40
## 56             39882              NA                 75
## 64             30972              NA                 86
##    artist_meta.rating_count artist_tag.count         tag.name ID_unit
## 11                        1                2 alternative rock      11
## 14                        5                4 alternative rock      14
## 15                        3                3 alternative rock      15
## 38                        1                2 alternative rock      38
## 56                        4                1 alternative rock      56
## 64                       16                5 alternative rock      64
##          Prob Stratum
## 11 0.08032129       1
## 14 0.08032129       1
## 15 0.08032129       1
## 38 0.08032129       1
## 56 0.08032129       1
## 64 0.08032129       1
```

# A boxplot of the stratified random sample data is created:

```
z <- ggplot(srs_ad, aes(x=tag.name, y=artist_meta.rating)) +
  geom_boxplot() +ggpubr::rotate_x_text()
z
```




The plot indicates “hard rock” has a rating different from the other data.

The mean from a stratified random sample is calculated:

```
mean(df$artist_meta.rating)
```

```
## [1] 88.0634
```

The mean from stratified sampling is 89.5216 compared to the population mean of 88.29761. Since the simple random sample was 88.304 it seems the SRS performed better. However, the stratified sample is still comparable to the population mean. The mean from the simple random sample was 88.304 - effectively identical to the population mean of 88.29761 or, 88.30 rounded up.

A chi-square test of independence of the simple random sample data is made:

```
chisq.test(srs_ad$artist_meta.rating, srs_ad$tag.name, srs_ad$artist.begin_date_year, correct=FA
LSE)
```

```
## Warning in chisq.test(srs_ad$artist_meta.rating, srs_ad$tag.name,
## srs_ad$artist.begin_date_year, : Chi-squared approximation may be incorrect
```

```
##
##  Pearson's Chi-squared test
##
## data:  srs_ad$artist_meta.rating and srs_ad$tag.name
## X-squared = 547.35, df = 540, p-value = 0.4041
```

Chi square is a test of the independence of the data. Hnull: the data is independent, Halternate the data is not independent. The Chi-square test has a p-value of 1.611E-6 so we reject Hnull the data is independent and accept Halternate the data is dependent.

Anova test of the artist_meta.rating and tag.name on the simple random sample data is made:

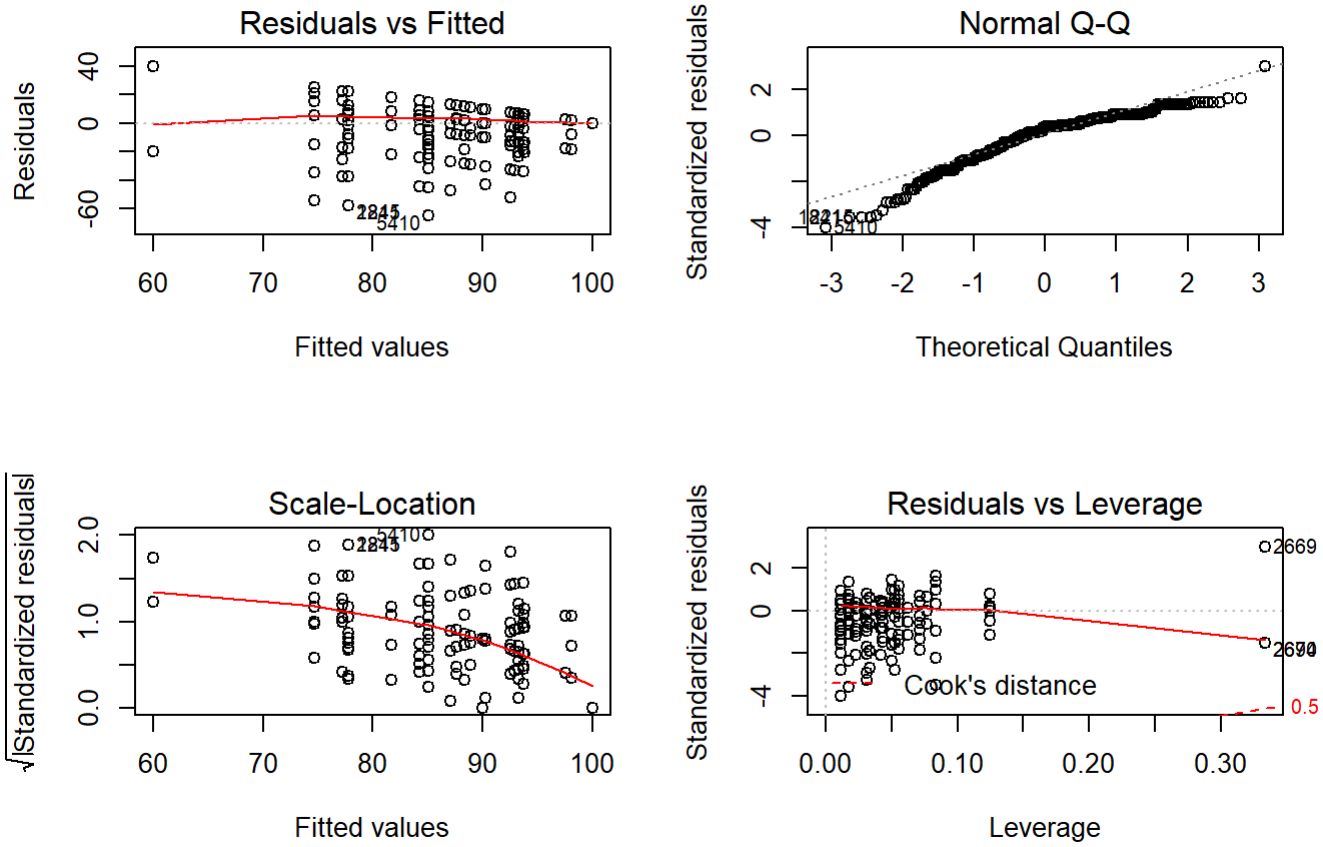
```
CRD_artist<-aov(artist_meta.rating~tag.name, data=srs_ad) #Perform ANOVA for CRD
summary(CRD_artist)
```

```
##              Df Sum Sq Mean Sq F value   Pr(>F)
## tag.name     20  21735  1086.7   4.094 1.27e-08 ***
## Residuals   479 127154   265.5
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The ANOVA test hypotheses Hnull: there is no difference in score between tag.name i.e. music genres, Halternate: there is a difference of at least one the tag.names. At a significance level of alpha=0.05 we reject Hnull and accept Halternate - there is a difference in score at least one of the tag.name.

With the results of the Anova test we can create charts to test hypotheses of equal variances, normality of residuals and leverage in the data:

```
par(mfrow=c(2,2))
plot(CRD_artist)
```




A plot of Residuals vs Fitted values demonstrates a generally structureless and random plot. There is no apparent “funnelling”. The appear to be independent and random. Further, the errors would appear to have a constant variance. The Q-Q plot demonstrates significant deviation from a line. The errors do not follow a normal distribution. A plot of residuals vs. leverage does not demonstrate leverage.

A Shapiro-Wilks test is performed to further investigate the normality of the residuals, and Breusch-Pagan to test for homoscedasticity:

```
#bartlett.test(artist_meta.rating~tag.name, data=srs_ad)
shapiro.test(residuals(CRD_artist))
```

```
##
##  Shapiro-Wilk normality test
##
## data:  residuals(CRD_artist)
## W = 0.91874, p-value = 9.143e-16
```

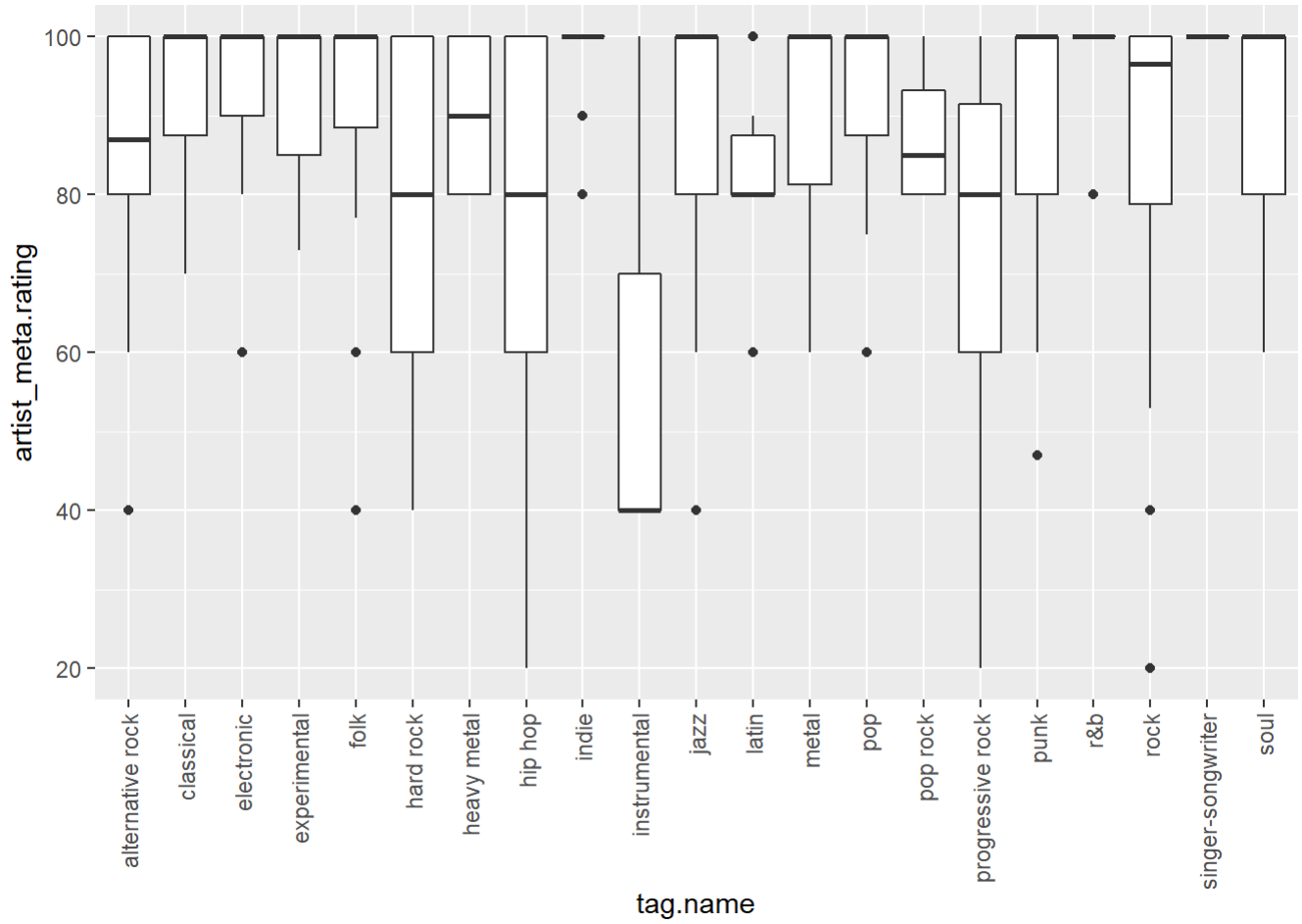
```
bptest(CRD_artist)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  CRD_artist
## BP = 51.656, df = 20, p-value = 0.0001278
```

A Shapiro-Wilk normality test was performed. In this we test the hypotheses Hnull: the residuals are normally distributed, or Halt: they residuals are not normally distributedat a significance level of alpha=0.05. The test returns a p-value of 2.2E-16 so we reject Hnull:the residuals are normally distributed and, accept Halternate: the residuals are not normally distributed. The Breusch-Pagan test has Hnull:the data is homoscedastic and Halternate: the data is heteroscedastic. In this test we fail to reject Hnull and conclude the data is homoscedastic. That is, the variance of the variables is the same or least similar.

A boxplot of the simple random sample of the data is created:

```
p <- ggplot(srs_ad, aes(x=tag.name, y=artist_meta.rating)) +
  geom_boxplot() +ggpubr::rotate_x_text()
p
```




An examination of the boxplot does demonstrate that Hard Rock and Soft Rock have mean artist_meta.rating scores significantly lower than the other genres. Despite the data failing the normality of residuals assumption a Dunn-Test and Tukey test are performed to test the differences between artist_meta.rating's

```
DT = DunnTest(artist_meta.rating~tag.name, data=srs_ad, method="none")
#DT
```

Dunn's Test can be used to pinpoint which specific means are significant from the others. Dunn's test is a "non parametric test" which can be run even though the normality hypothesis was rejected. Hnull that there is no difference between mean artist_meta.rating's and Halternate there is a difference in mean between artist_meta.rating's.

Similar to the inspection of the boxplot of means hard rock has a mean score different from each other tag.name (genre). Soft rock did not have a difference in mean rating:

###alternative rock - hard rock ###classical - hard rock
###electronic - hard rock
###experimental - hard rock ###folk - hard rock ###hard rock - indie ###hard rock - instrumental ###hard rock - jazz ###hard rock - metal ###hard rock - pop ###hard rock - pop rock ###hard rock - punk ###hard rock - r&b ###hard rock - soul

## As a comparison to the Dunn Test a Tukey test is calculated. Although, the Tukey Test is not a non-parametric test like the Dunn Test so we expect different results:

```
TukeyHSD(CRD_artist, conf.level = 0.95)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = artist_meta.rating ~ tag.name, data = srs_ad)
##
## $tag.name
##                                          diff         lwr         upr
## classical-alternative rock           9.00686499  -9.1243703 27.13810026
## electronic-alternative rock          8.70614035  -9.2534023 26.66568301
## experimental-alternative rock        9.57518797 -11.0244514 30.17482738
## folk-alternative rock                8.32072368  -8.6178472 25.25929454
## hard rock-alternative rock          -6.96052632 -25.6968324 11.77577974
## heavy metal-alternative rock         5.78947368 -15.2613929 26.84034029
## hip hop-alternative rock            -6.44266917 -21.9702316  9.08489327
## indie-alternative rock              13.91447368  -5.9300508 33.75899818
## instrumental-alternative rock      -24.21052632 -60.5448750 12.12382234
## jazz-alternative rock                2.88322368 -14.0553472 19.82179454
## latin-alternative rock              -2.54385965 -21.7805908 16.69287151
## metal-alternative rock               4.71804511 -15.8815943 25.31768452
## pop-alternative rock                 9.53947368  -6.5155371 25.59448451
## pop rock-alternative rock            3.41447368 -21.2347675 28.06371490
## progressive rock-alternative rock   -9.54385965 -31.1092285 12.02150917
## punk-alternative rock                6.02280702 -11.1248220 23.17043602
## r&b-alternative rock                13.28947368 -11.3597675 37.93871490
## rock-alternative rock                0.81220096 -13.9828757 15.60727759
## singer-songwriter-alternative rock  15.78947368  -8.8597675 40.43871490
## soul-alternative rock                4.12280702 -15.1139241 23.35953818
## electronic-classical                -0.30072464 -17.3663755 16.76492624
## experimental-classical               0.56832298 -19.2568230 20.39346891
## folk-classical                      -0.68614130 -16.6738381 15.30155552
## hard rock-classical                -15.96739130 -33.8486785  1.91389589
## heavy metal-classical               -3.21739130 -23.5109926 17.07620996
## hip hop-classical                  -15.44953416 -29.9338865 -0.96518182
## indie-classical                      4.90760870 -14.1317300 23.94694744
## instrumental-classical             -33.21739130 -69.1183127  2.68353012
## jazz-classical                      -6.12364130 -22.1113381  9.86405552
## latin-classical                    -11.55072464 -29.9556993  6.85425005
## metal-classical                     -4.28881988 -24.1139658 15.53632606
## pop-classical                        0.53260870 -14.5158012 15.58101857
## pop rock-classical                  -5.59239130 -29.5981481 18.41336550
## progressive rock-classical         -18.55072464 -39.3775439  2.27609461
## punk-classical                      -2.98405797 -19.1930816 13.22496568
## r&b-classical                        4.28260870 -19.7231481 28.28836550
## rock-classical                      -8.19466403 -21.8908518  5.50152376
## singer-songwriter-classical          6.78260870 -17.2231481 30.78836550
## soul-classical                      -4.88405797 -23.2890327 13.52091671
## experimental-electronic              0.86904762 -18.7991981 20.53729338
## folk-electronic                     -0.38541667 -16.1781343 15.40730096
## hard rock-electronic               -15.66666667 -33.3738379  2.04050453
## heavy metal-electronic              -2.91666667 -23.0570175 17.22368418
## hip hop-electronic                 -15.14880952 -29.4176547 -0.87996434
## indie-electronic                     5.20833333 -13.6675746 24.08424125
## instrumental-electronic            -32.91666667 -68.7311838  2.89785049
```

```
## jazz-electronic                    -5.82291667 -21.6156343  9.96980096
## latin-electronic                  -11.25000000 -29.4858595  6.98585954
## metal-electronic                   -3.98809524 -23.6563410 15.68015052
## pop-electronic                      0.83333333 -14.0077623 15.67442898
## pop rock-electronic                -5.29166667 -29.1680114 18.58467811
## progressive rock-electronic       -18.25000000 -38.9275211  2.42752112
## punk-electronic                    -2.68333333 -18.7000723 13.33340566
## r&b-electronic                      4.58333333 -19.2930114 28.45967811
## rock-electronic                    -7.89393939 -21.3620142  5.57413545
## singer-songwriter-electronic        7.08333333 -16.7930114 30.95967811
## soul-electronic                    -4.58333333 -22.8191929 13.65252621
## folk-experimental                  -1.25446429 -19.9950593 17.48613072
## hard rock-experimental            -16.53571429 -36.9157016  3.84427304
## heavy metal-experimental           -3.78571429 -26.3120034 18.74057488
## hip hop-experimental              -16.01785714 -33.4935521  1.45783781
## indie-experimental                  4.33928571 -17.0639821 25.74255348
## instrumental-experimental         -33.78571429 -70.9943102  3.42288164
## jazz-experimental                  -6.69196429 -25.4325593 12.04863072
## latin-experimental                -12.11904762 -32.9600300  8.72193471
## metal-experimental                 -4.85714286 -26.9623428 17.24805704
## pop-experimental                   -0.03571429 -17.9816913 17.91026274
## pop rock-experimental              -6.16071429 -32.0813588 19.75993021
## progressive rock-experimental     -19.11904762 -42.1268691  3.88877390
## punk-experimental                  -3.55238095 -22.4821432 15.37738131
## r&b-experimental                    3.71428571 -22.2063588 29.63493021
## rock-experimental                  -8.76298701 -25.5912077  8.06523367
## singer-songwriter-experimental      6.21428571 -19.7063588 32.13493021
## soul-experimental                  -5.45238095 -26.2933633 15.38860138
## hard rock-folk                    -15.28125000 -31.9520005  1.38950049
## heavy metal-folk                   -2.53125000 -21.7667313 16.70423133
## hip hop-folk                      -14.76339286 -27.7237151 -1.80307061
## indie-folk                          5.59375000 -12.3135086 23.50100858
## instrumental-folk                 -32.53125000 -67.8448402  2.78234015
## jazz-folk                          -5.43750000 -20.0587154  9.18371540
## latin-folk                        -10.86458333 -28.0958509  6.36668427
## metal-folk                         -3.60267857 -22.3432736 15.13791644
## pop-folk                            1.21875000 -12.3690434 14.80654342
## pop rock-folk                      -4.90625000 -28.0244214 18.21192142
## progressive rock-folk             -17.86458333 -37.6618026  1.93263591
## punk-folk                          -2.29791667 -17.1608214 12.56498809
## r&b-folk                            4.96875000 -18.1494214 28.08692142
## rock-folk                          -7.50852273 -19.5815940  4.56454854
## singer-songwriter-folk              7.46875000 -15.6494214 30.58692142
## soul-folk                          -4.19791667 -21.4291843 13.03335094
## heavy metal-hard rock              12.75000000  -8.0859719 33.58597188
## hip hop-hard rock                   0.51785714 -14.7171005 15.75281480
## indie-hard rock                    20.87500000   1.2585811 40.49141893
## instrumental-hard rock            -17.25000000 -53.4602697 18.96026969
## jazz-hard rock                      9.84375000  -6.8270005 26.51450049
## latin-hard rock                     4.41666667 -14.5846639 23.41799728
## metal-hard rock                    11.67857143  -8.7014159 32.05855875
## pop-hard rock                      16.50000000   0.7278057 32.27219426
## pop rock-hard rock                 10.37500000 -14.0909729 34.84097294
## progressive rock-hard rock         -2.58333333 -23.9389853 18.77231865
```

```
## punk-hard rock                       12.98333333  -3.8997920 29.86645863
## r&b-hard rock                        20.25000000  -4.2159729 44.71597294
## rock-hard rock                        7.77272727  -6.7149583 22.26041280
## singer-songwriter-hard rock          22.75000000  -1.7159729 47.21597294
## soul-hard rock                       11.08333333  -7.9179973 30.08466395
## hip hop-heavy metal                 -12.23214286 -30.2375241  5.77323842
## indie-heavy metal                     8.12500000 -13.7128962 29.96289624
## instrumental-heavy metal            -30.00000000 -67.4602917  7.46029167
## jazz-heavy metal                     -2.90625000 -22.1417313 16.32923133
## latin-heavy metal                    -8.33333333 -29.6204277 12.95376101
## metal-heavy metal                    -1.07142857 -23.5977177 21.45486059
## pop-heavy metal                       3.75000000 -14.7121757 22.21217572
## pop rock-heavy metal                 -2.37500000 -28.6556707 23.90567069
## progressive rock-heavy metal        -15.33333333 -38.7460156  8.07934896
## punk-heavy metal                      0.23333333 -19.1864952 19.65316186
## r&b-heavy metal                       7.50000000 -18.7806707 33.78067069
## rock-heavy metal                     -4.97727273 -22.3549265 12.40038109
## singer-songwriter-heavy metal        10.00000000 -16.2806707 36.28067069
## soul-heavy metal                     -1.66666667 -22.9537610 19.62042767
## indie-hip hop                        20.35714286   3.7782429 36.93604278
## instrumental-hip hop                -17.76785714 -52.4267617 16.89104737
## jazz-hip hop                          9.32589286  -3.6344294 22.28621511
## latin-hip hop                         3.89880952 -11.9475338 19.74515283
## metal-hip hop                        11.16071429  -6.3149807 28.63640924
## pop-hip hop                          15.98214286   4.2000317 27.76425399
## pop rock-hip hop                      9.85714286 -12.2480570 31.96234276
## progressive rock-hip hop             -3.10119048 -21.7054884 15.50310749
## punk-hip hop                         12.46547619  -0.7669065 25.69785885
## r&b-hip hop                          19.73214286  -2.3730570 41.83734276
## rock-hip hop                          7.25487013  -2.7425827 17.25232295
## singer-songwriter-hip hop            22.23214286   0.1269430 44.33734276
## soul-hip hop                         10.56547619  -5.2808671 26.41181949
## instrumental-indie                  -38.12500000 -74.9209185 -1.32908149
## jazz-indie                          -11.03125000 -28.9385086  6.87600858
## latin-indie                         -16.45833333 -36.5532718  3.63660516
## metal-indie                          -9.19642857 -30.5996963 12.20683920
## pop-indie                            -4.37500000 -21.4489011 12.69890113
## pop rock-indie                      -10.50000000 -35.8246879 14.82468795
## progressive rock-indie              -23.45833333 -45.7926088 -1.12405789
## punk-indie                           -7.89166667 -25.9968015 10.21346818
## r&b-indie                            -0.62500000 -25.9496879 24.69968795
## rock-indie                          -13.10227273 -28.9972097  2.79266420
## singer-songwriter-indie               1.87500000 -23.4496879 27.19968795
## soul-indie                           -9.79166667 -29.8866052 10.30327183
## jazz-instrumental                    27.09375000  -8.2198402 62.40734015
## latin-instrumental                   21.66666667 -14.8050524 58.13838575
## metal-instrumental                   28.92857143  -8.2800245 66.13716735
## pop-instrumental                     33.75000000  -1.1483931 68.64839308
## pop rock-instrumental                27.62500000 -11.9694385 67.21943849
## progressive rock-instrumental        14.66666667 -23.0851492 52.41848251
## punk-instrumental                    30.23333333  -5.1810091 65.64767573
## r&b-instrumental                     37.50000000  -2.0944385 77.09443849
## rock-instrumental                    25.02272727  -9.3142609 59.35971548
## singer-songwriter-instrumental       40.00000000   0.4055615 79.59443849
```

```
## soul-instrumental                    28.33333333  -8.1383857 64.80505241
## latin-jazz                           -5.42708333 -22.6583509 11.80418427
## metal-jazz                            1.83482143 -16.9057736 20.57541644
## pop-jazz                              6.65625000  -6.9315434 20.24404342
## pop rock-jazz                         0.53125000 -22.5869214 23.64942142
## progressive rock-jazz               -12.42708333 -32.2243026  7.37013591
## punk-jazz                             3.13958333 -11.7233214 18.00248809
## r&b-jazz                             10.40625000 -12.7119214 33.52442142
## rock-jazz                            -2.07102273 -14.1440940 10.00204854
## singer-songwriter-jazz               12.90625000 -10.2119214 36.02442142
## soul-jazz                             1.23958333 -15.9916843 18.47085094
## metal-latin                           7.26190476 -13.5790776 28.10288709
## pop-latin                            12.08333333  -4.2801863 28.44685292
## pop rock-latin                        5.95833333 -18.8929542 30.80962089
## progressive rock-latin               -7.00000000 -28.7960210 14.79602104
## punk-latin                            8.56666667  -8.8701502 26.00348350
## r&b-latin                            15.83333333  -9.0179542 40.68462089
## rock-latin                            3.35606061 -11.7732386 18.48535977
## singer-songwriter-latin              18.33333333  -6.5179542 43.18462089
## soul-latin                            6.66666667 -12.8282872 26.16162054
## pop-metal                             4.82142857 -13.1245485 22.76740559
## pop rock-metal                       -1.30357143 -27.2242159 24.61707307
## progressive rock-metal              -14.26190476 -37.2697263  8.74591676
## punk-metal                            1.30476190 -17.6250004 20.23452417
## r&b-metal                             8.57142857 -17.3492159 34.49207307
## rock-metal                           -3.90584416 -20.7340648 12.92237653
## singer-songwriter-metal              11.07142857 -14.8492159 36.99207307
## soul-metal                           -0.59523810 -21.4362204 20.24574424
## pop rock-pop                         -6.12500000 -28.6038354 16.35383537
## progressive rock-pop                -19.08333333 -38.1300674 -0.03659923
## punk-pop                             -3.51666667 -17.3641981 10.33086481
## r&b-pop                               3.75000000 -18.7288354 26.22883537
## rock-pop                             -8.72727273 -19.5257560  2.07121050
## singer-songwriter-pop                 6.25000000 -16.2288354 28.72883537
## soul-pop                             -5.41666667 -21.7801863 10.94685292
## progressive rock-pop rock           -12.95833333 -39.6528983 13.73623165
## punk-pop rock                         2.60833333 -20.6634489 25.88011555
## r&b-pop rock                          9.87500000 -19.3674308 39.11743081
## rock-pop rock                        -2.60227273 -24.1992392 18.99469372
## singer-songwriter-pop rock           12.37500000 -16.8674308 41.61743081
## soul-pop rock                         0.70833333 -24.1429542 25.55962089
## punk-progressive rock                15.56666667  -4.4097166 35.54304992
## r&b-progressive rock                 22.83333333  -3.8612317 49.52789832
## rock-progressive rock                10.35606061  -7.6414114 28.35353265
## singer-songwriter-progressive rock   25.33333333  -1.3612317 52.02789832
## soul-progressive rock                13.66666667  -8.1293544 35.46268771
## r&b-punk                              7.26666667 -16.0051156 30.53844889
## rock-punk                            -5.21060606 -17.5752754  7.15406328
## singer-songwriter-punk                9.76666667 -13.5051156 33.03844889
## soul-punk                            -1.90000000 -19.3368168 15.53681683
## rock-r&b                            -12.47727273 -34.0742392  9.11969372
## singer-songwriter-r&b                 2.50000000 -26.7424308 31.74243081
## soul-r&b                             -9.16666667 -34.0179542 15.68462089
## singer-songwriter-rock               14.97727273  -6.6196937 36.57423918
```

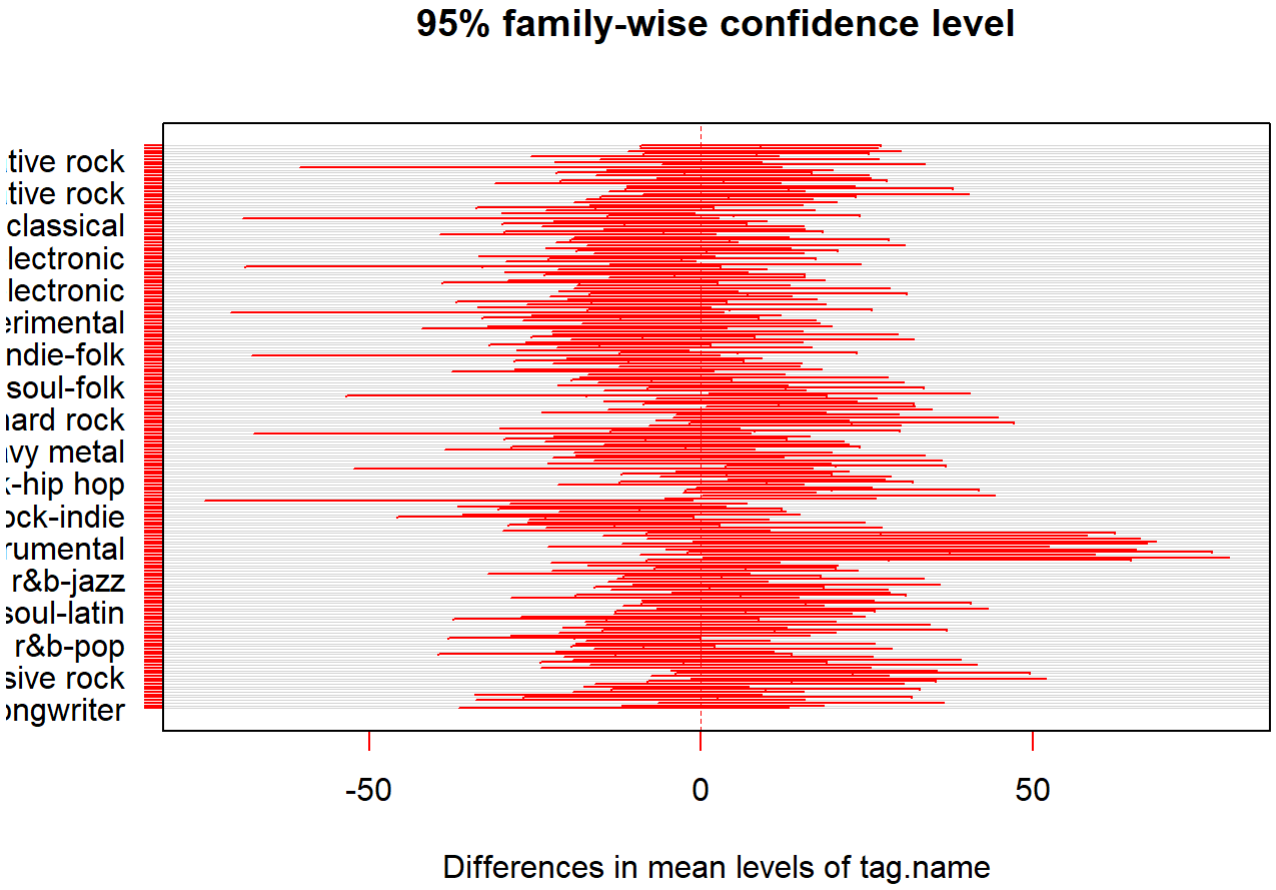
```
## soul-rock                            3.31060606 -11.8186931 18.43990523
## soul-singer-songwriter              -11.66666667 -36.5179542 13.18462089
##                                         p adj
## classical-alternative rock          0.9715633
## electronic-alternative rock         0.9780609
## experimental-alternative rock       0.9862285
## folk-alternative rock               0.9747216
## hard rock-alternative rock          0.9992161
## heavy metal-alternative rock        0.9999924
## hip hop-alternative rock            0.9965254
## indie-alternative rock              0.5958569
## instrumental-alternative rock       0.6912397
## jazz-alternative rock               1.0000000
## latin-alternative rock              1.0000000
## metal-alternative rock              0.9999997
## pop-alternative rock                0.8561126
## pop rock-alternative rock           1.0000000
## progressive rock-alternative rock   0.9922494
## punk-alternative rock               0.9996496
## r&b-alternative rock                0.9364645
## rock-alternative rock               1.0000000
## singer-songwriter-alternative rock  0.7566018
## soul-alternative rock               0.9999999
## electronic-classical                1.0000000
## experimental-classical              1.0000000
## folk-classical                      1.0000000
## hard rock-classical                 0.1530163
## heavy metal-classical               1.0000000
## hip hop-classical                   0.0223187
## indie-classical                     0.9999974
## instrumental-classical              0.1119127
## jazz-classical                      0.9988010
## latin-classical                     0.7870773
## metal-classical                     0.9999999
## pop-classical                       1.0000000
## pop rock-classical                  0.9999995
## progressive rock-classical          0.1562635
## punk-classical                      1.0000000
## r&b-classical                       1.0000000
## rock-classical                      0.8483515
## singer-songwriter-classical         0.9999881
## soul-classical                      0.9999958
## experimental-electronic             1.0000000
## folk-electronic                     1.0000000
## hard rock-electronic                0.1650839
## heavy metal-electronic              1.0000000
## hip hop-electronic                  0.0237654
## indie-electronic                    0.9999920
## instrumental-electronic             0.1190035
## jazz-electronic                     0.9992952
## latin-electronic                    0.8106759
## metal-electronic                    1.0000000
## pop-electronic                      1.0000000
## pop rock-electronic                 0.9999998
```

```
## progressive rock-electronic        0.1683802
## punk-electronic                    1.0000000
## r&b-electronic                     1.0000000
## rock-electronic                    0.8705168
## singer-songwriter-electronic       0.9999738
## soul-electronic                    0.9999983
## folk-experimental                  1.0000000
## hard rock-experimental             0.3042525
## heavy metal-experimental           1.0000000
## hip hop-experimental               0.1219877
## indie-experimental                 1.0000000
## instrumental-experimental          0.1326290
## jazz-experimental                  0.9995544
## latin-experimental                 0.8783744
## metal-experimental                 0.9999998
## pop-experimental                   1.0000000
## pop rock-experimental              0.9999994
## progressive rock-experimental      0.2615851
## punk-experimental                  1.0000000
## r&b-experimental                   1.0000000
## rock-experimental                  0.9542631
## singer-songwriter-experimental     0.9999993
## soul-experimental                  0.9999967
## hard rock-folk                     0.1219021
## heavy metal-folk                   1.0000000
## hip hop-folk                       0.0085179
## indie-folk                         0.9999407
## instrumental-folk                  0.1165223
## jazz-folk                          0.9992044
## latin-folk                         0.7803646
## metal-folk                         1.0000000
## pop-folk                           1.0000000
## pop rock-folk                      0.9999999
## progressive rock-folk              0.1400002
## punk-folk                          1.0000000
## r&b-folk                           0.9999999
## rock-folk                          0.7997750
## singer-songwriter-folk             0.9999000
## soul-folk                          0.9999990
## heavy metal-hard rock              0.8212357
## hip hop-hard rock                  1.0000000
## indie-hard rock                    0.0230300
## instrumental-hard rock             0.9818627
## jazz-hard rock                     0.8628328
## latin-hard rock                    0.9999996
## metal-hard rock                    0.8919773
## pop-hard rock                      0.0288422
## pop rock-hard rock                 0.9954170
## progressive rock-hard rock         1.0000000
## punk-hard rock                     0.4084354
## r&b-hard rock                      0.2684940
## rock-hard rock                     0.9392756
## singer-songwriter-hard rock        0.1068313
## soul-hard rock                     0.8753638
```

```
## hip hop-heavy metal               0.6562435
## indie-heavy metal                 0.9991993
## instrumental-heavy metal          0.3286920
## jazz-heavy metal                  1.0000000
## latin-heavy metal                 0.9983856
## metal-heavy metal                 1.0000000
## pop-heavy metal                   1.0000000
## pop rock-heavy metal              1.0000000
## progressive rock-heavy metal      0.7209033
## punk-heavy metal                  1.0000000
## r&b-heavy metal                   0.9999860
## rock-heavy metal                  0.9999852
## singer-songwriter-heavy metal     0.9989051
## soul-heavy metal                  1.0000000
## indie-hip hop                     0.0023431
## instrumental-hip hop              0.9607926
## jazz-hip hop                      0.5441192
## latin-hip hop                     0.9999988
## metal-hip hop                     0.7612351
## pop-hip hop                       0.0002971
## pop rock-hip hop                  0.9915129
## progressive rock-hip hop          1.0000000
## punk-hip hop                      0.0942924
## r&b-hip hop                       0.1534736
## rock-hip hop                      0.5269719
## singer-songwriter-hip hop         0.0467899
## soul-hip hop                      0.6901310
## instrumental-indie                0.0326099
## jazz-indie                        0.8125922
## latin-indie                       0.2871315
## metal-indie                       0.9945983
## pop-indie                         0.9999977
## pop rock-indie                    0.9965579
## progressive rock-indie            0.0273872
## punk-indie                        0.9935544
## r&b-indie                         1.0000000
## rock-indie                        0.2756754
## singer-songwriter-indie           1.0000000
## soul-indie                        0.9768101
## jazz-instrumental                 0.4131171
## latin-instrumental                0.8563092
## metal-instrumental                0.3864798
## pop-instrumental                  0.0722578
## pop rock-instrumental             0.6055815
## progressive rock-instrumental     0.9985440
## punk-instrumental                 0.2170980
## r&b-instrumental                  0.0894476
## rock-instrumental                 0.5183660
## singer-songwriter-instrumental    0.0444000
## soul-instrumental                 0.3880526
## latin-jazz                        0.9999326
## metal-jazz                        1.0000000
## pop-jazz                          0.9754546
## pop rock-jazz                     1.0000000
```

```
## progressive rock-jazz              0.7867780
## punk-jazz                          0.9999999
## r&b-jazz                           0.9905103
## rock-jazz                          1.0000000
## singer-songwriter-jazz             0.9133066
## soul-jazz                          1.0000000
## metal-latin                        0.9996882
## pop-latin                          0.4913162
## pop rock-latin                     0.9999993
## progressive rock-latin             0.9999087
## punk-latin                         0.9746844
## r&b-latin                          0.7648649
## rock-latin                         0.9999998
## singer-songwriter-latin            0.4932878
## soul-latin                         0.9997637
## pop-metal                          0.9999948
## pop rock-metal                     1.0000000
## progressive rock-metal             0.8042782
## punk-metal                         1.0000000
## r&b-metal                          0.9998573
## rock-metal                         0.9999996
## singer-songwriter-metal            0.9949835
## soul-metal                         1.0000000
## pop rock-pop                       0.9999935
## progressive rock-pop               0.0489054
## punk-pop                           0.9999980
## r&b-pop                            1.0000000
## rock-pop                           0.3115191
## singer-songwriter-pop              0.9999909
## soul-pop                           0.9998551
## progressive rock-pop rock          0.9777340
## punk-pop rock                      1.0000000
## r&b-pop rock                       0.9998042
## rock-pop rock                      1.0000000
## singer-songwriter-pop rock         0.9955342
## soul-pop rock                      1.0000000
## punk-progressive rock              0.3819160
## r&b-progressive rock               0.2140758
## rock-progressive rock              0.8882583
## singer-songwriter-progressive rock 0.0876774
## soul-progressive rock              0.7883527
## r&b-punk                           0.9999410
## rock-punk                          0.9957651
## singer-songwriter-punk             0.9959805
## soul-punk                          1.0000000
## rock-r&b                           0.8845583
## singer-songwriter-r&b              1.0000000
## soul-r&b                           0.9992911
## singer-songwriter-rock             0.6173182
## soul-rock                          0.9999999
## soul-singer-songwriter             0.9845925
```

```
plot(TukeyHSD(CRD_artist, conf.level = 0.95),las=1, col = "red")
```




The Tukey HSD test: Hnull -The difference between means of the artist_meta.rating means cannot be said to be different except for:

hard rock-alternative rock

indie-hard rock

pop-hard rock

hard rock-folk

hard rock-electronic

These results are different from the Dunn Test and demonstrate “hard rock” not having the Hnull hypothesis being rejected for 9 of the comparisons of hard rock to other genres.Likely, this is because the Tukey test is a “parametric” test and relies on normality of residuals. That is, we should ignore the results of the Tukey test and rely only the results of the Dunn Test.

CONCLUSIONS:

Stratified random sampling and simple random sampling gave similar results when a population mean was created. The simple random sample did seem perform slightly better in this particular instance. The Chi-Square test demonstrated the data was not independent. The ratings of the genre did depend on the genre. A aov demonstrated that there was not normality of residuals although, the data was homoscedastic. The non-parametric Dunn Test demonstrated a difference in ratings between the “hard-rock” genre and the other genre. Since the Tukey test is non-parametric it should not be used on data that did not satisfy the normality of residuals. Although, it gave similar results to the Dunn-Test.

## 3.2 Ratings by Year

I am estimating population mean, population variance and population SD for artist rating. For this I am using one stage clusetring. I have clusterred data based on decade of artist begin year. For all artist begin year before 1950, I keep all of that artists in one cluster 1950. Artists begin year 1951-1960 are in cluster 1960 and so on.

There are total 8 clusters and I am choosing 3 random clusters out of 8.

Considering all data from randomly selected cluster and estimating population mean, population variance and population standard deviation based on this sample.

Each cluster has 100 data rows, so it is equal probability & equal weight of selecting each record in sample.

I have groupped data based on artist begin year to create cluster using tableau and used copy of my data file for estimating population mean and population SD.

```
MYDATA=read.csv('Output2.csv')
head(MYDATA)
```

```
##   ï..F1 artist.id     artist.name artist.sort_name
## 1     4        17       Bob Dylan       Dylan, Bob
## 2     8        23       Tom Waits       Waits, Tom
## 3    11        29   Stevie Wonder   Wonder, Stevie
## 4    35        83 Fryderyk Chopin Chopin, Fryderyk
## 5    40        91      Bill Cosby      Cosby, Bill
## 6    43        94   John Williams   Williams, John
##   Cluster_ByArtistBeginDateYear artist.end_date_year artist.type artist.area
## 1                          1950                   NA           1         222
## 2                          1950                   NA           1         222
## 3                          1950                   NA           1         222
## 4                          1950                 1849           1         170
## 5                          1950                   NA           1         222
## 6                          1950                   NA           1         222
##   artist.gender artist.ended artist.begin_area artist.end_area
## 1             1        FALSE              5767              NA
## 2             1        FALSE              7804              NA
## 3             1        FALSE              5080              NA
## 4             1         TRUE             88730            4434
## 5             1        FALSE              7707              NA
## 6             1        FALSE               295              NA
##   artist_meta.rating n1  n2 artist_meta.rating_count artist_tag.count
## 1                 90  8 981                       29                5
## 2                 97  8 981                       12                3
## 3                 91  8 981                        7                5
## 4                 92  8 981                       10                9
## 5                 70  8 981                        2                1
## 6                 93  8 981                        9                6
##     tag.name
## 1 blues rock
## 2       rock
## 3       soul
## 4  classical
## 5     comedy
## 6  classical
```

```
# One-stage cluster sampling
cl=sampling:::cluster(MYDATA,clustername=c("Cluster_ByArtistBeginDateYear"),size=3,method="srswo
r")
mydata1=getdata(MYDATA, cl)
esti_pop_total=sum(mydata1$artist_meta.rating)*8/3
cat("Estimated population mean of Artist rating is",mean(mydata1$artist_meta.rating))
```

```
## Estimated population mean of Artist rating is 87.71667
```

# Estimating population variance using formulas:

$$v_t = \frac{1}{n-1}\sum_{i \in S}(t_i - \frac{\sum_{i \in S} t_i}{n})^2$$

$$AND$$

$$Var(\hat{\bar{y}}) = (1 - \frac{n}{N})\frac{V_t}{n \cdot M^2}$$

```
vt<-sd(c(sum(mydata1$artist_meta.rating[1:100]),sum(mydata1$artist_meta.rating[101:200]),sum(myd
ata1$artist_meta.rating[201:300])))^2
esti_variance=(1-3/10)*vt/(3*619^2)
esti_sd=esti_variance^0.5
cat("Estimated population standard deviation is",esti_sd)
```

```
## Estimated population standard deviation is 0.499956
```

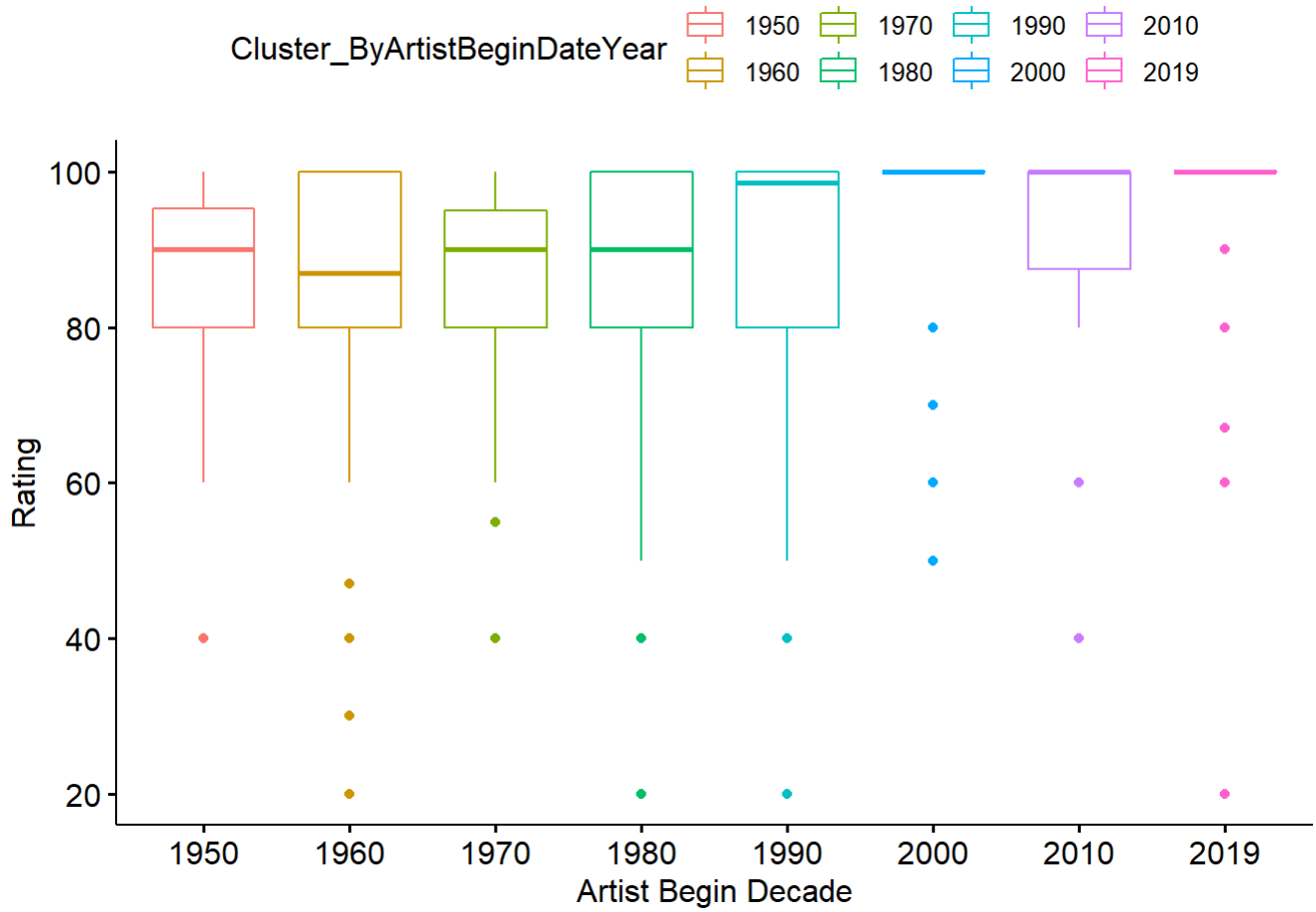
# Checking if cluster is good sampling technique for this dataset using ANOVA table.

```
#anova(within_var,within_mean)
res.aov<-aov(artist_meta.rating~Cluster_ByArtistBeginDateYear,data=MYDATA)
summary(res.aov)
```

```
##                                Df Sum Sq Mean Sq F value   Pr(>F)
## Cluster_ByArtistBeginDateYear   1  10944   10944   47.08 1.37e-11 ***
## Residuals                     798 185512     232
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

# Plotting data points on box plot, separating by cluster to check how is mean and median values of data points.

```
ggboxplot(MYDATA, x = "Cluster_ByArtistBeginDateYear", y = "artist_meta.rating",
          color = "Cluster_ByArtistBeginDateYear",
          #order = c("L1","L2","L3","L4","L5","L6","L7","L8","L9","L10"),
          ylab = "Rating", xlab = "Artist Begin Decade")
```




# Conclusion:

SSB is significantly lower than SSW. This suggest cluster sampling can be applied for estimating population mean.

# 3.3 Music Loudness by Year

## 3.3.1 Sampling technique

Songs are released by albums so the **recording** table does not contain such information. The table **release_country** contains data on the release date of a given album at a given country since an album may have multiple release dates. I simply sampled 100 albums from **release_country** and knowing that the dataset is large, I was not concerned with sampling duplicate albums that was released in multiple countries as that probability was low. I then proceeded join the sampled **release_country** to **release** then to **medium**, **track**, and then finally **recording**. The **recording** table contains the needed MBID to refer to the AcousticBrainz database using the WEB API. AcousticBrainz database does not contain information of every recording in the MusicBrainz Database unfortunately so I simply used what I had. Once I obtained average_loudness data, I simply merged it back into the previously merged dataset, cleaned it and exported as 'part33.csv'.

**SEE: Appendix_B.rmd**

## 3.3.2 Analysis

Below is the clean data that we will use for our analysis

```
newdata = read.csv('part33.csv', stringsAsFactors = FALSE) %>%
  dplyr::select(-c('X'))
```

### 3.3.2.1 Simple Regression

I will only perform a simple linear regression with only one independent variable to answer whether or not music is getting louder each year. Let:

$H_0 : \hat{\beta}_{year} = 0$

$H_a : \hat{\beta}_{year} \neq 0$

```
loudmodel = lm(average_loudness~release_country.date_year, data=newdata)
```

```
summary(loudmodel)
```

```
##
## Call:
## lm(formula = average_loudness ~ release_country.date_year, data = newdata)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7387 -0.1405  0.1270  0.2209  0.3417
##
## Coefficients:
##                              Estimate Std. Error t value Pr(>|t|)
## (Intercept)                -2.9336382  0.2525381  -11.62   <2e-16 ***
## release_country.date_year  0.0018192  0.0001268   14.35   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2941 on 21978 degrees of freedom
## Multiple R-squared:  0.009278,   Adjusted R-squared:  0.009232
## F-statistic: 205.8 on 1 and 21978 DF,  p-value: < 2.2e-16
```

```
aov(loudmodel)
```

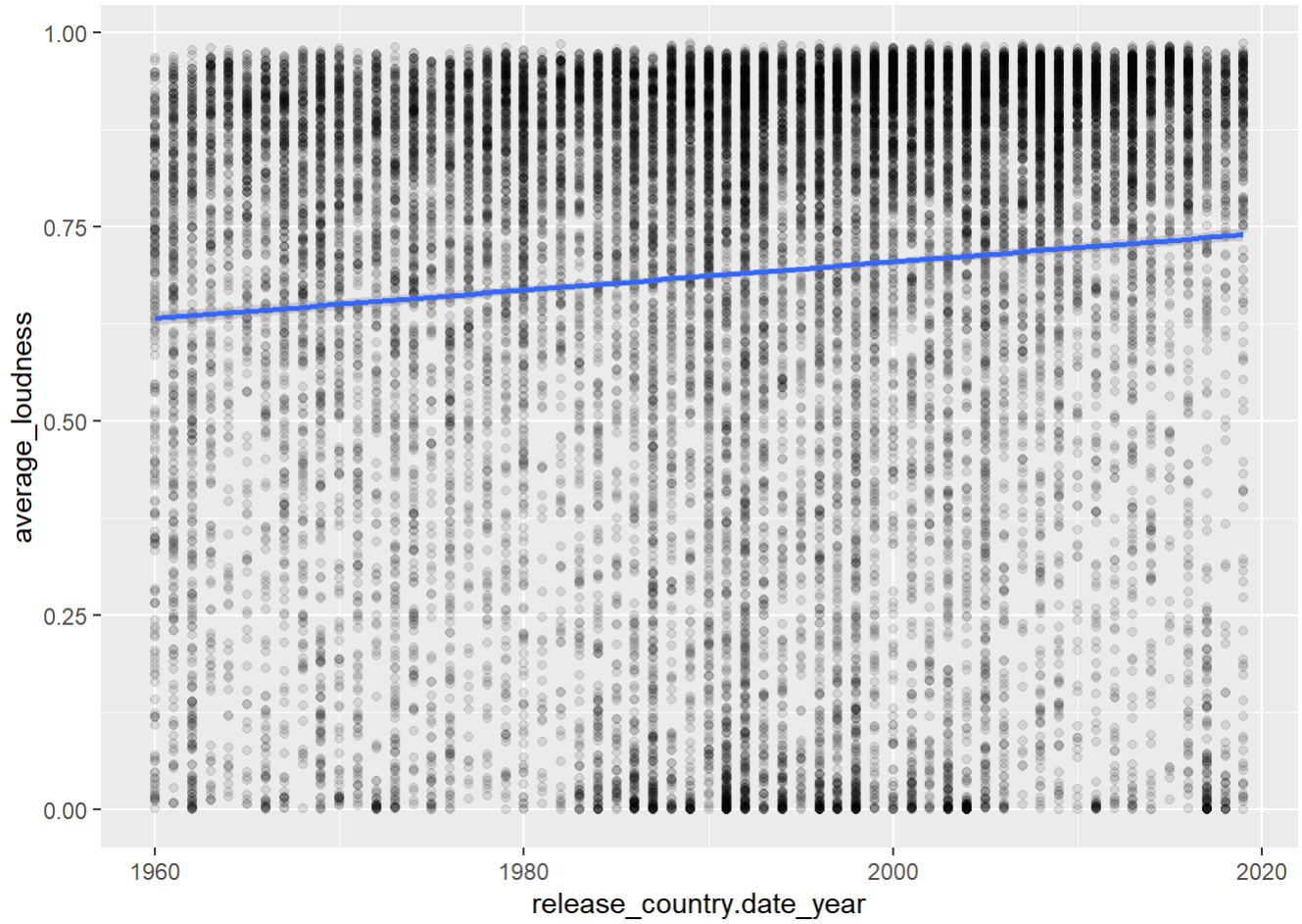
```
## Call:
##    aov(formula = loudmodel)
##
## Terms:
##                 release_country.date_year Residuals
## Sum of Squares                    17.8069 1901.5535
## Deg. of Freedom                         1     21978
##
## Residual standard error: 0.2941441
## Estimated effects may be unbalanced
```

The results indicate that the model is significant with a P-value of less than 0.05 so we can reject the null. However, the effect is very small as we have an $R^2_{adj}$ value of 0.009232.

The linear formula is:
$avg\_loudness = 0.0001268 * Year - 2.9336382$

Now, let's visualize the results.

```
ggplot(newdata, aes(x=release_country.date_year, y=average_loudness)) +geom_point(alpha=0.1) +ge
om_smooth(method='lm', level=0.95)
```




The data points are very busy and while it's hard to make out visually, our summary statistics indicate that we do see loudness increasing with time. The following is a graph which averages out the points in each year, however doing a regression on these points will not produce valid results due to a lack of normality but it makes it easier to see the trend. the non-normality will be seen later when I test the assumptions of linear regression.

```
avgdata = newdata %>%
  group_by(release_country.date_year) %>%
  summarise(mean_loud = mean(average_loudness))

ggplot(avgdata, aes(x=release_country.date_year, y=mean_loud)) +geom_point(color = 'red') +geom_
smooth(method='lm', level=0.95)
```

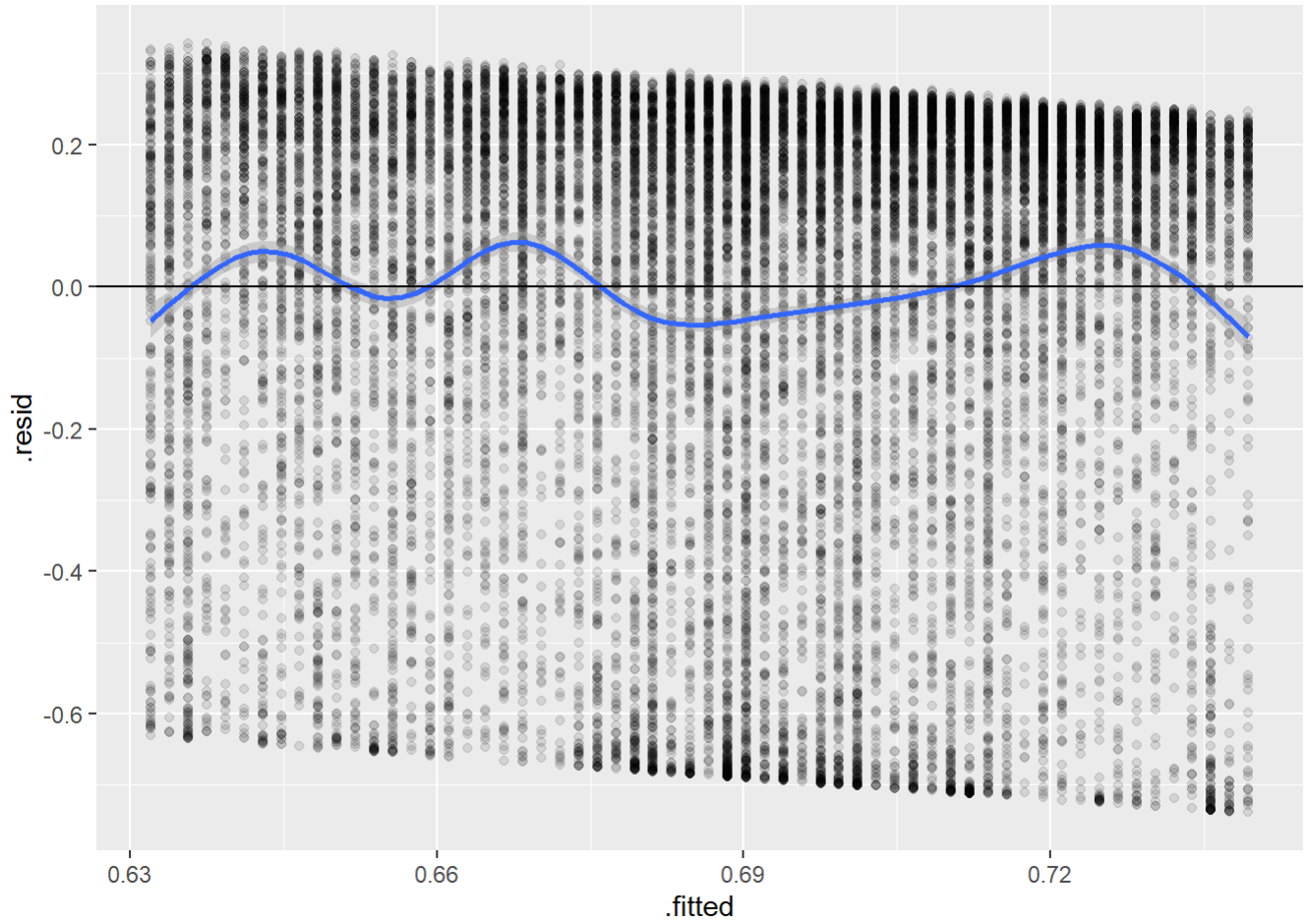

Let's perform a Breusch-Pagan test and visualize the results to see if the residuals are homoscedastic. Let:
$H_0$ : $residuals\ are\ Homoscedastic$
$H_a$ : $residuals\ are\ Heteroscedastic$

```
bptest(loudmodel)
```

```
##
##  studentized Breusch-Pagan test
##
## data:  loudmodel
## BP = 0.50449, df = 1, p-value = 0.4775
```

```
ggplot(loudmodel, aes(x=.fitted, y=.resid)) +
  geom_point(alpha = 0.1) + geom_smooth()+
  geom_hline(yintercept = 0)
```

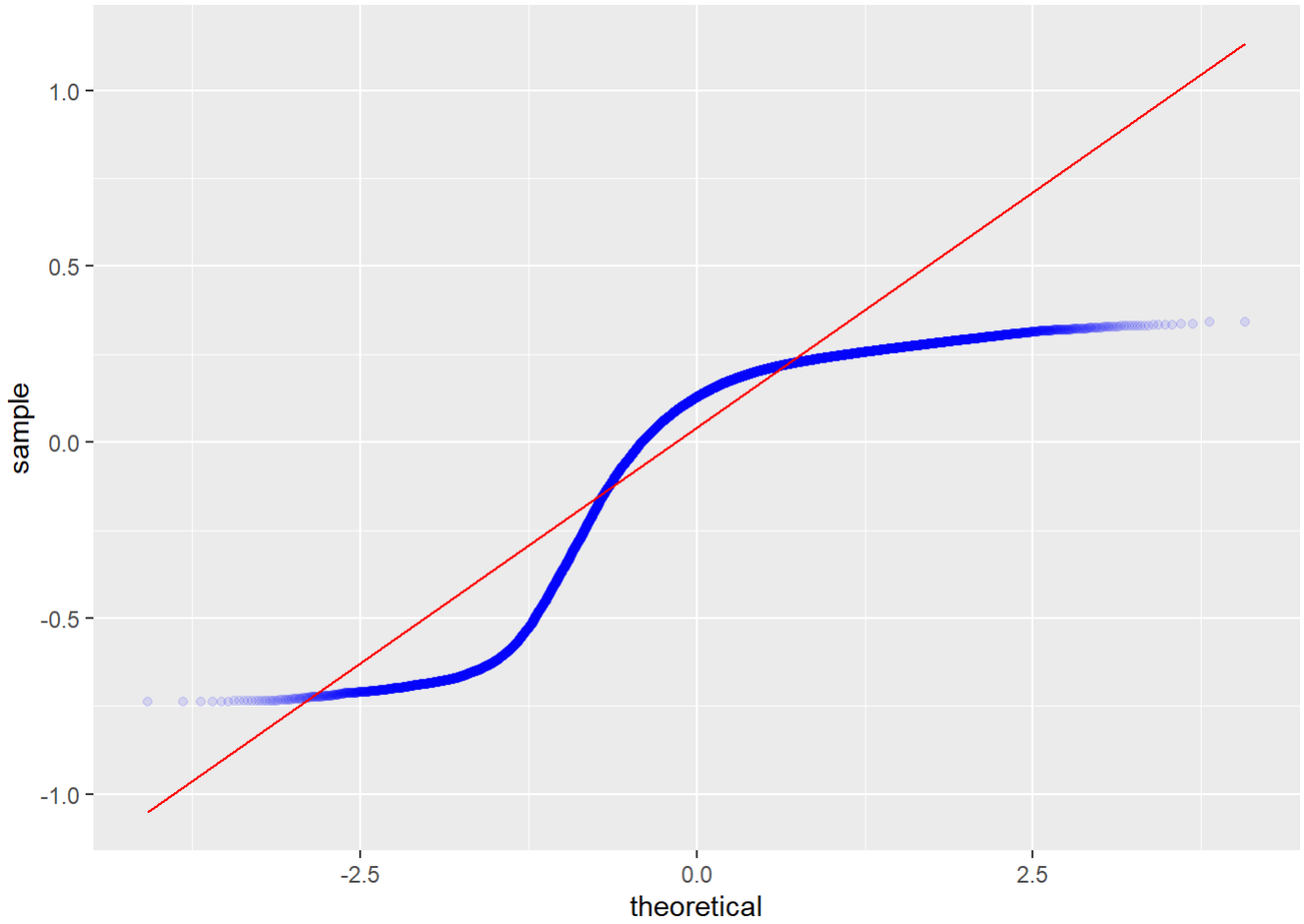
```
## `geom_smooth()` using method = 'gam' and formula 'y ~ s(x, bs = "cs")'
```




We see that the P-val is greater than 0.05 so we can accept the null. Visually we can also see that the residuals are fairly uniform.

Now let's test for normality visually.

```
ggplot(newdata, aes(sample=loudmodel$residuals)) +
  stat_qq(color = 'blue', alpha=0.1) +
  stat_qq_line(color='red')
```




The residuals are not distributed normally, however given that we have a very large sample size this requirement can be relaxed.

### 3.3.2.2 Mantel-Haenszel and Chi-Squared test

Here we will test for independence but the Mantel-Haenszel test can also give us an insight into the association between the years and loudness of music.
Year will be grouped into decades, and average_loudness will be assigned to four levels: 0,1,2,3 where higher number indicates higher loudness.

```
tabdata = newdata %>%
  mutate(y_cluster = floor((release_country.date_year-1960)/10)*10+60 ) %>%
  mutate(loud_cluster = floor(average_loudness/0.25))
```

The following is the contingency table for the ordinal data:

```
loudtable = table(tabdata$y_cluster, tabdata$loud_cluster)
loudtable
```

```
##
##          0    1    2    3
##   60   349  336  622 1352
##   70   330  330  648 1536
##   80   564  324  690 2068
##   90   947  530  834 2883
##   100  510  365  685 3173
##   110  307  177  363 2057
```

The left most column are decades. Decade 100 is the 2000's and decade 110 is the decade 2010.

Let's perform the Mantel-Haenszel test. Let:
$H_0$ : Decade and average loudness are independent
$H_a$ : Decade and average loudness are dependent

```
pears.cor(loudtable, c(1,2,3,4,5,6), c(7,8,9,10))
```

```
## Pearson correlation        MH statistic             P-Value
##          0.08587616        162.08884308          0.00000000
```

Using the Mantel-Haenszel test we can determine that the decade and loudness are dependent (P-val<0.05) and that there is an $R$ of 0.08587616 which if we square it we get $R^2$ of:

```
cat("R squared is:", 0.08587616^2)
```

```
## R squared is: 0.007374715
```

Which is very similar to the $R^2_{adj}$ that we got in the linear regression model.

The following Chi-Squared test is just an extra test that I did.

```
chisq.test(loudtable)
```

```
##
##  Pearson's Chi-squared test
##
## data:  loudtable
## X-squared = 593.26, df = 15, p-value < 2.2e-16
```

```
qchisq(c(0.95),df=15, lower.tail=TRUE)
```

```
## [1] 24.99579
```

Same conclusion, the $\chi^2$ is greater than the critical $(0.95)\chi^2$ of 24.99579.

# 4.0 Conclusion

Stratified random sampling and simple random sampling gave similar results when a population mean was created. The simple random sample did seem perform slightly better in this particular instance. The Chi-Square test demonstrated the data was not independent. The ratings of the genre did depend on the genre. A aov demonstrated that there was not normality of residuals although, the data was homoscedastic. The non-parametric Dunn Test demonstrated a difference in ratings between the “hard-rock” genre and the other genre. Since the Tukey test is non-parametric it should not be used on data that did not satisfy the normality of residuals. Although, it gave similar results to the Dunn-Test.

SSB is significantly lower than SSW. This suggest cluster sampling can be applied for estimating population mean.

# 5.0 Contribution by Each Member

Inwoo Choi

- Obtaining and cleaning data
- Part 3.3 of music loudness by year: Simple Regression and Mantel-Haenszel Test

Brad Robinson

- Part 3.1

Rupal Gandhi

- Part 3.2 : Estimating population mean, population variance and population SD using one stage clustering.