R Notebook

This is the mark-up file for the Datenanalyse 2 homework assignment.

```r
library("rio")
```

```
## Warning: package 'rio' was built under R version 3.5.3
```

```r
x <- import("https://docs.google.com/spreadsheets/d/1SWEakSjZUvvV3w8peOf5FHrGI9NTEDls3c9zETVZ5kQ/export

str(x)
```

```
## 'data.frame':    720 obs. of  31 variables:
##  $ Artist_Albums_Number             : int  0 0 1 1 1 1 1 1 1 1 ...
##  $ Artist_Albums_Tracks_Number      : int  0 0 8 8 8 8 8 8 8 8 ...
##  $ Artist_Appearances_Number        : int  9 9 2 2 2 2 2 2 2 2 ...
##  $ Artist_Appearances_Tracks_Number : int  502 502 30 30 30 30 30 30 30 30 ...
##  $ Artist_Compilations_Number       : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Artist_Compilations_Tracks_Number: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Artist_Follower                  : int  713401 713401 601346 601346 601346 601346 601346 601346 60
##  $ Artist_ID                        : chr  "2NjfBq1NflQcKSeiDooVjY" "2NjfBq1NflQcKSeiDooVjY" "1qQLhy
##  $ Artist_Popularity                : int  91 91 83 83 83 83 83 83 83 83 ...
##  $ Artist_Singles_Number            : int  3 3 15 15 15 15 15 15 15 15 ...
##  $ Artist_Singles_Tracks_Number     : int  10 10 15 15 15 15 15 15 15 15 ...
##  $ Genre                            : chr  "pop" "pop" "Hip Hop" "Hip Hop" ...
##  $ Release_Date                     : chr  "2019-05-10" "2019-07-15" "2019-10-25" "2019-08-23" ...
##  $ Streams                          : int  106824437 2327995 79193552 54619683 48552840 46784729 434
##  $ Track_Artist                     : chr  "Tones and I" "Tones and I" "Apache 207" "Apache 207" ...
##  $ Track_Duration_ms                : int  209754 200755 157093 158853 176066 163146 139693 191760 17
##  $ Track_ID                         : chr  "1rgnBhdG2JDFTbYkYRZAku" "2grAr8pWMuLWn8ZYEE9wDV" "6hw1Sy
##  $ Track_Popularity                 : int  76 72 78 77 73 75 73 75 69 69 ...
##  $ Track_Title                      : chr  "Dance Monkey" "Never Seen the Rain" "Roller" "Roller" ..
##  $ Title_Artist_Google_searches_11m : int  20904 572 8880 8880 1975 1156 3260 10880 220 568 ...
##  $ Title_Artist_Youtube_searches_11m: int  308911 7320 7660 7660 1530 990 2240 7915 154 441 ...
##  $ Title_Google_searches_11m        : int  1288732 2799 4805454 4805454 47025 33165 47709 45925 8977
##  $ Title_Youtube_searches_11m       : int  18353181 33600 3446454 3446454 32325 28872 38436 31975 593
##  $ Total_tracks                     : int  512 512 53 53 53 53 53 53 53 53 ...
##  $ Artist_Google_searches_11m       : int  299212 299212 1468281 1468281 1468281 1468281 1468281 146
##  $ Artist_Youtube_searches_11m      : int  2451500 2451500 1076400 1076400 1076400 1076400 1076400 10
##  $ commentCount                     : num  172604 2272 22183 22183 13376 ...
##  $ dislikeCount                     : int  317322 3194 27802 27802 11440 12957 10493 605 5333 4287 .
##  $ likeCount                        : int  7424686 109395 748270 748270 385252 378780 299481 24361 2
##  $ video_ID                         : chr  "qOhyYWKXF0Q" "UdRJY-jlEhQ" "Fo3DAhiNKQo" "Fo3DAhiNKQo" .
##  $ viewsCount                       :integer64 738528171 10258864 66995452 66995452 22170062 2864717
```

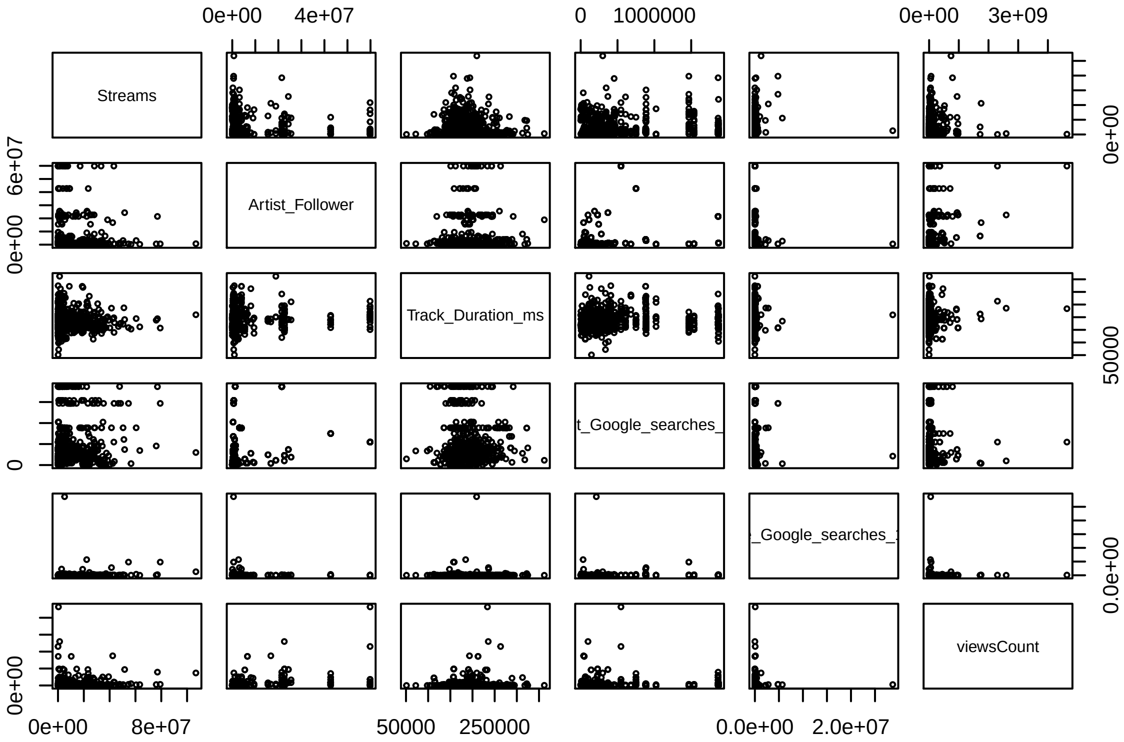
```r
x$commentCount <- as.integer(x$commentCount)
x$viewsCount <- as.numeric(x$viewsCount)
```

```r
library("dplyr")
```

```
## Warning: package 'dplyr' was built under R version 3.5.3

##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
```

```
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
drop.cols <- c('Artist_ID', 'Genre', 'Release_Date', 'Track_Artist', 'Track_ID', 'Track_Title', 'video_I
numeric_x <- select(x, -one_of(drop.cols))

keep.cols <- c('Streams', 'Artist_Follower', 'Track_Duration_ms', 'Artist_Google_searches_11m', 'Title_G
               'viewsCount')

# keep.cols <- c('Streams', 'viewsCount', 'Title_Youtube_searches_11m')

selected_pairs <- select(x, keep.cols)

pairs(selected_pairs, cex=0.5)
```



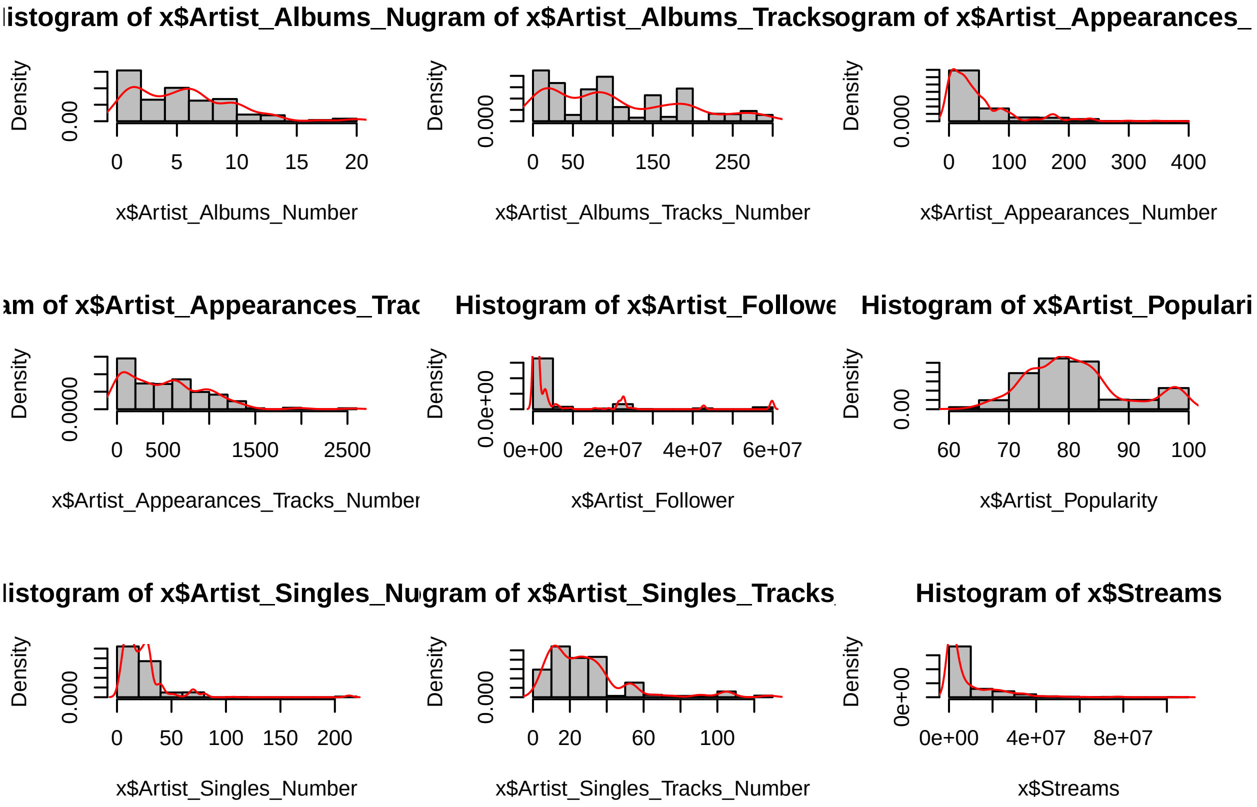
Descriptive statistics

```
summary(numeric_x)
```

```
##  Artist_Albums_Number Artist_Albums_Tracks_Number
##  Min.   : 0.000       Min.   :  0.0
##  1st Qu.: 2.000       1st Qu.: 26.0
##  Median : 5.000       Median : 86.0
##  Mean   : 5.508       Mean   :103.2
##  3rd Qu.: 8.000       3rd Qu.:159.0
##  Max.   :20.000       Max.   :299.0
##
##  Artist_Appearances_Number Artist_Appearances_Tracks_Number
##  Min.   :  0.00            Min.   :   0.0
##  1st Qu.: 12.00            1st Qu.: 140.0
##  Median : 28.00            Median : 479.0
##  Mean   : 48.43            Mean   : 526.9
##  3rd Qu.: 59.00            3rd Qu.: 786.0
##  Max.   :375.00            Max.   :2583.0
##
##  Artist_Compilations_Number Artist_Compilations_Tracks_Number
##  Min.   :0.0000             Min.   : 0.000
##  1st Qu.:0.0000             1st Qu.: 0.000
##  Median :0.0000             Median : 0.000
##  Mean   :0.1056             Mean   : 2.579
##  3rd Qu.:0.0000             3rd Qu.: 0.000
##  Max.   :2.0000             Max.   :57.000
##
##  Artist_Follower    Artist_Popularity Artist_Singles_Number
##  Min.   :    9449   Min.   :60.00     Min.   :  3.00
##  1st Qu.:  575873   1st Qu.:74.00     1st Qu.: 11.00
##  Median :  889326   Median :80.00     Median : 19.00
##  Mean   : 5710132   Mean   :81.22     Mean   : 23.18
##  3rd Qu.: 3129993   3rd Qu.:84.25     3rd Qu.: 29.00
##  Max.   :59828212   Max.   :99.00     Max.   :213.00
##
##  Artist_Singles_Tracks_Number    Streams          Track_Duration_ms
##  Min.   :  4.00               Min.   :    43688   Min.   : 51104
##  1st Qu.: 12.00               1st Qu.:   799953   1st Qu.:162634
##  Median : 26.00               Median :  3033628   Median :182656
##  Mean   : 29.01               Mean   :  8595051   Mean   :187680
##  3rd Qu.: 35.00               3rd Qu.: 11802780   3rd Qu.:204396
##  Max.   :128.00               Max.   :106824437   Max.   :361946
##
##  Track_Popularity Title_Artist_Google_searches_11m
##  Min.   : 0.00    Min.   :     0
##  1st Qu.:50.00    1st Qu.:     1
##  Median :58.00    Median :  1215
##  Mean   :58.65    Mean   : 10283
##  3rd Qu.:69.00    3rd Qu.:  7100
##  Max.   :99.00    Max.   :398000
##
##  Title_Artist_Youtube_searches_11m Title_Google_searches_11m
##  Min.   :      0                   Min.   :       0
##  1st Qu.:     10                   1st Qu.:       0
##  Median :   1292                   Median :    4666
##  Mean   :  58811                   Mean   :  106718
##  3rd Qu.:   9904                   3rd Qu.:   32263
##  Max.   :6870200                   Max.   :28689090
##
##  Title_Youtube_searches_11m  Total_tracks    Artist_Google_searches_11m
##  Min.   :        0          Min.   :  5.0   Min.   :      1
##  1st Qu.:        0          1st Qu.: 239.0   1st Qu.: 183522
##  Median :     6488          Median : 678.0   Median : 336545
##  Mean   :   662186          Mean   : 661.6   Mean   : 513368
##  3rd Qu.:   179120          3rd Qu.: 955.8   3rd Qu.: 608772
##  Max.   :134580909          Max.   :2699.0   Max.   :1871000
##
##  Artist_Youtube_searches_11m  commentCount       dislikeCount
##  Min.   :      10            Min.   :     0.0   Min.   :      2
##  1st Qu.:  270454            1st Qu.:   698.5   1st Qu.:    586
##  Median :  504090            Median :  5281.0   Median :   4594
##  Mean   : 2179428            Mean   : 32745.3   Mean   :  33026
##  3rd Qu.: 1705727            3rd Qu.: 19694.0   3rd Qu.:  18845
##  Max.   :28298181            Max.   :934238.0   Max.   :1203541
##                              NA's   :5
##    likeCount          viewsCount
##  Min.   :      32   Min.   :3.290e+03
##  1st Qu.:   24622   1st Qu.:1.698e+06
##  Median :  138002   Median :6.880e+06
##  Mean   :  808778   Mean   :7.298e+07
##  3rd Qu.:  427237   3rd Qu.:3.175e+07
##  Max.   :22120897   Max.   :4.641e+09
##
```

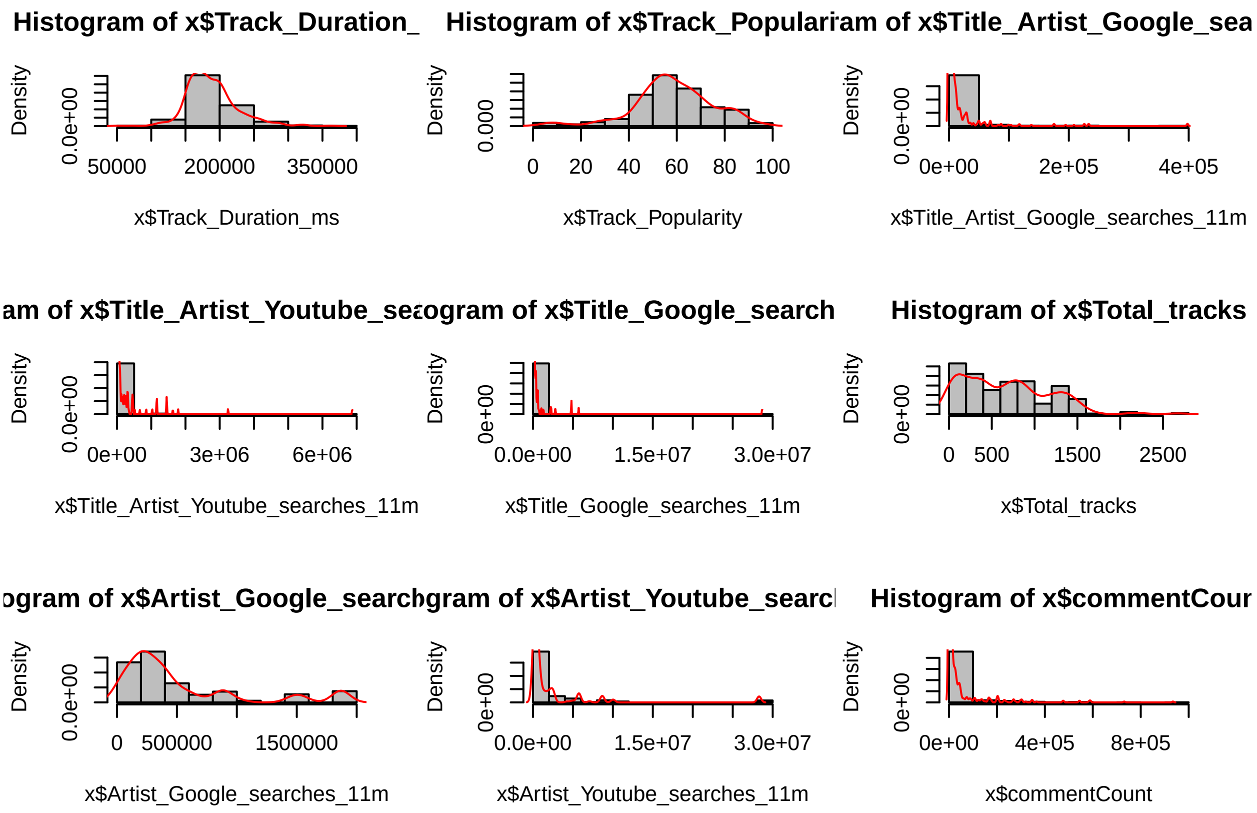
Histograms and kernel density plots of base variables

```r
par(mfrow=c(3,3))

hist(x$Artist_Albums_Number, probability = TRUE, col = "gray")
lines(density(x$Artist_Albums_Number), col = "red")

hist(x$Artist_Albums_Tracks_Number, probability = TRUE, col = "gray")
lines(density(x$Artist_Albums_Tracks_Number), col = "red")

hist(x$Artist_Appearances_Number, probability = TRUE, col = "gray")
lines(density(x$Artist_Appearances_Number), col = "red")

hist(x$Artist_Appearances_Tracks_Number, probability = TRUE, col = "gray")
lines(density(x$Artist_Appearances_Tracks_Number), col = "red")

hist(x$Artist_Follower, probability = TRUE, col = "gray")
lines(density(x$Artist_Follower), col = "red")

hist(x$Artist_Popularity, probability = TRUE, col = "gray")
lines(density(x$Artist_Popularity), col = "red")

hist(x$Artist_Singles_Number, probability = TRUE, col = "gray")
lines(density(x$Artist_Singles_Number), col = "red")

hist(x$Artist_Singles_Tracks_Number, probability = TRUE, col = "gray")
lines(density(x$Artist_Singles_Tracks_Number), col = "red")

hist(x$Streams, probability = TRUE, col = "gray")
lines(density(x$Streams), col = "red")
```

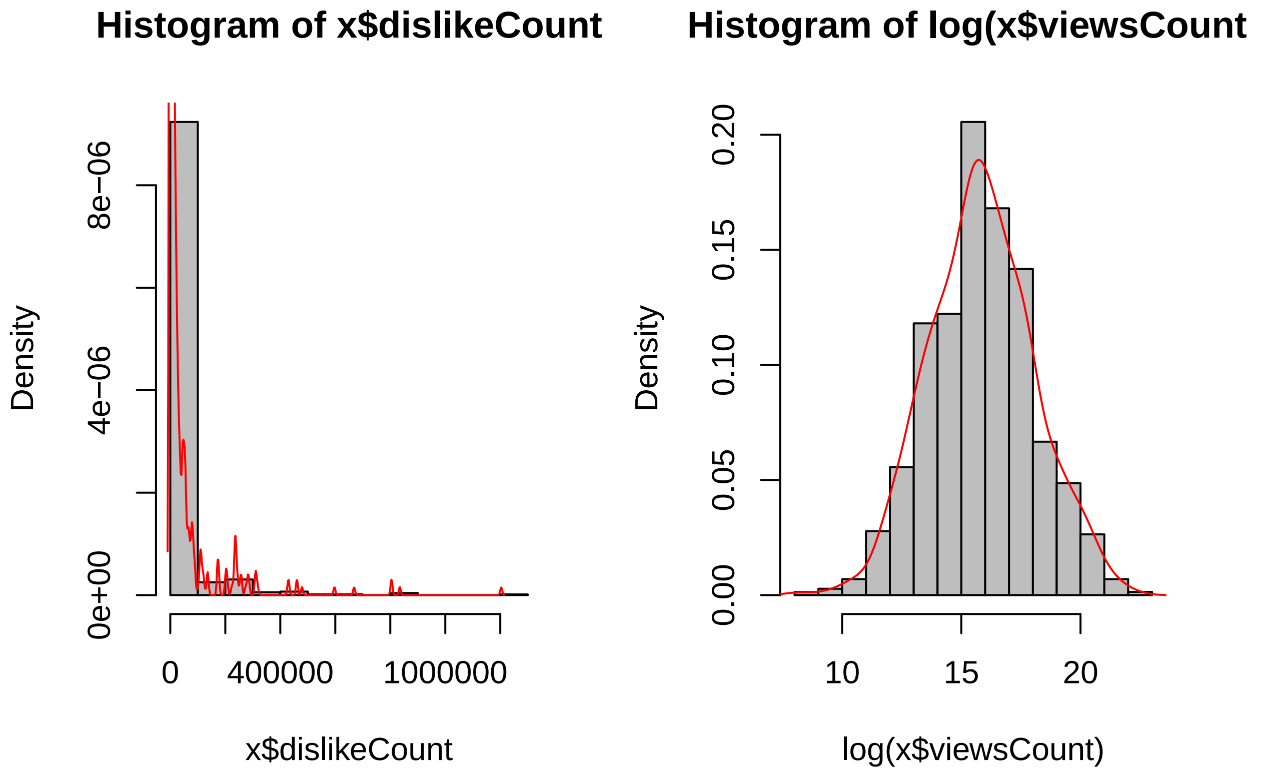


```r
par(mfrow=c(3,3))

hist(x$Track_Duration_ms, probability = TRUE, col = "gray")
lines(density(x$Track_Duration_ms), col = "red")

hist(x$Track_Popularity, probability = TRUE, col = "gray")
lines(density(x$Track_Popularity), col = "red")

hist(x$Title_Artist_Google_searches_11m, probability = TRUE, col = "gray")
lines(density(x$Title_Artist_Google_searches_11m), col = "red")

hist(x$Title_Artist_Youtube_searches_11m, probability = TRUE, col = "gray")
lines(density(x$Title_Artist_Youtube_searches_11m), col = "red")

hist(x$Title_Google_searches_11m, probability = TRUE, col = "gray")
lines(density(x$Title_Google_searches_11m), col = "red")

hist(x$Total_tracks, probability = TRUE, col = "gray")
lines(density(x$Total_tracks), col = "red")

hist(x$Artist_Google_searches_11m, probability = TRUE, col = "gray")
lines(density(x$Artist_Google_searches_11m), col = "red")

hist(x$Artist_Youtube_searches_11m, probability = TRUE, col = "gray")
lines(density(x$Artist_Youtube_searches_11m), col = "red")
```

```r
hist(x$commentCount, probability = TRUE, col = "gray")
lines(density(x$commentCount, na.rm = TRUE), col = "red")
```



```r
par(mfrow=c(1,2))

hist(x$dislikeCount, probability = TRUE, col = "gray")
lines(density(x$dislikeCount), col = "red")

hist(log(x$viewsCount), probability = TRUE, col = "gray")
lines(density(log(x$viewsCount), na.rm = TRUE), col = "red")
```



```
#hist(x$viewsCount, probability = TRUE, col = "gray")
#lines(density(x$viewsCount, na.rm = TRUE), col = "red")
```

Distribution test

```
strictly_positive_variables <- c('Artist_Follower', 'Artist_Popularity', 'Streams', 'Track_Duration_ms'
                                  'viewsCount', 'Artist_Google_searches_11m')


summary(select(x, strictly_positive_variables))
```

```
##  Artist_Follower    Artist_Popularity    Streams        
##  Min.   :    9449   Min.   :60.00     Min.   :    43688  
##  1st Qu.:  575873   1st Qu.:74.00     1st Qu.:   799953  
##  Median :  889326   Median :80.00     Median :  3033628  
##  Mean   : 5710132   Mean   :81.22     Mean   :  8595051  
##  3rd Qu.: 3129993   3rd Qu.:84.25     3rd Qu.: 11802780  
##  Max.   :59828212   Max.   :99.00     Max.   :106824437  
##  Track_Duration_ms   viewsCount         Artist_Google_searches_11m
##  Min.   : 51104    Min.   :3.290e+03   Min.   :      1           
##  1st Qu.:162634    1st Qu.:1.698e+06   1st Qu.: 183522           
##  Median :182656    Median :6.880e+06   Median : 336545           
##  Mean   :187680    Mean   :7.298e+07   Mean   : 513368           
##  3rd Qu.:204396    3rd Qu.:3.175e+07   3rd Qu.: 608772           
##  Max.   :361946    Max.   :4.641e+09   Max.   :1871000           
```

```
library("psych")
```

```
## Warning: package 'psych' was built under R version 3.5.2
```

```r
library("car")
```

```
## Warning: package 'car' was built under R version 3.5.2
```

```
## Loading required package: carData
```

```
## Warning: package 'carData' was built under R version 3.5.2
```

```
##
## Attaching package: 'car'
```

```
## The following object is masked from 'package:psych':
##
##     logit
```

```
## The following object is masked from 'package:dplyr':
##
##     recode
```

```r
ksD <- function (p, x) {
  y <- bcPower(x, p)
  ks.test(y, "pnorm", mean=mean(y), sd=sd(y))$statistic
}

oldw <- getOption("warn")
options(warn = -1)

min_values <- c()

for (column_index in 1:length(strictly_positive_variables)){

  column_name <- strictly_positive_variables[column_index]

  x_sub <- as.numeric(x[[paste(column_name)]])

  result <- optimize(ksD, c(-5,5), x=x_sub)

  min_values[column_index] <- result$minimum

  message(paste(column_index, ', minimum value is: ', result$minimum))

}
```

```
## 1 , minimum value is:  -0.205660850905614
```

```
## 2 , minimum value is:  -1.72547245696245
```

```
## 3 , minimum value is:  0.037975342271715
```

```
## 4 , minimum value is:  0.212785305428911
```

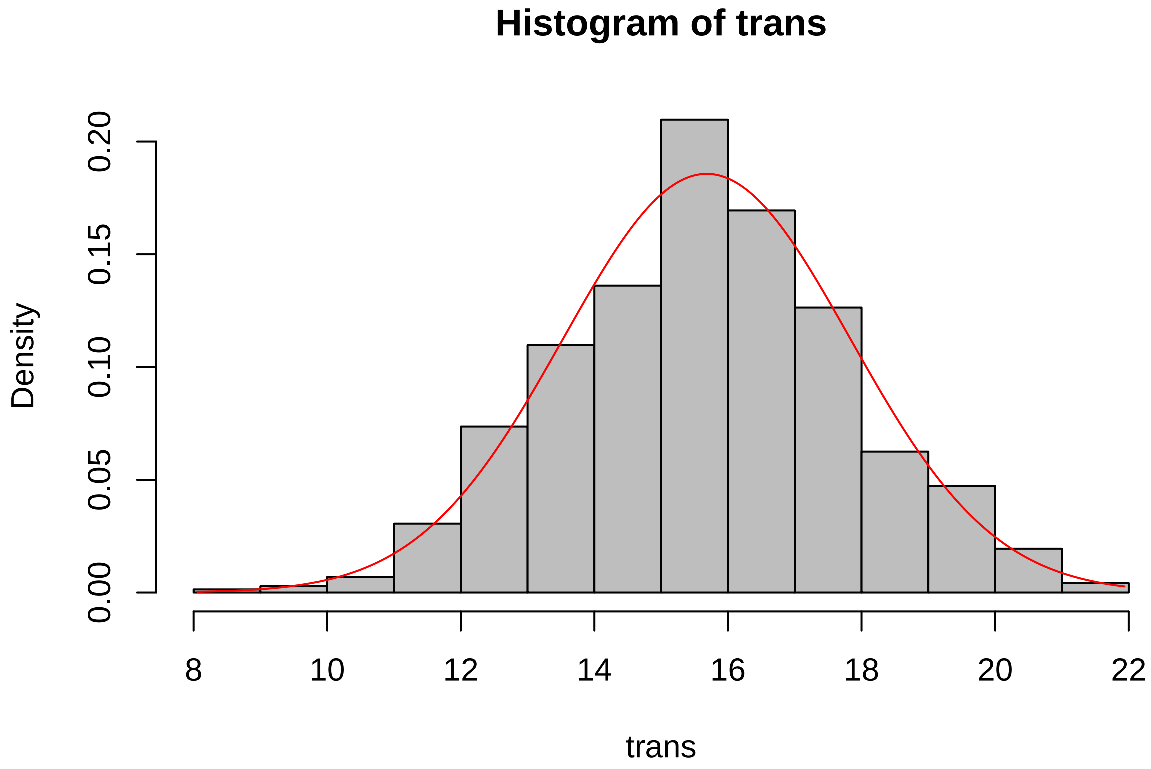
```
## 5 , minimum value is:  -0.00130968618131601
```

```
## 6 , minimum value is:  0.139522250128656
```

```r
options(warn = oldw)
```

Box-Cox transformations

```
column_index <- 5
column_name <- strictly_positive_variables[column_index]
x_sub <- as.numeric(x[[paste(column_name)]])
trans <- bcPower(x_sub, min_values[column_index])

hist(trans, col = "gray", probability = TRUE)
points(seq(min(trans), max(trans), length.out = 500),
       dnorm(seq(min(trans), max(trans), length.out = 500),
             mean(trans),sd(trans)), type = "l", col = "red")
```



```
test_statistic <- ks.test(trans, "pnorm", mean=mean(trans), sd=sd(trans))$statistic
```

```
## Warning in ks.test(trans, "pnorm", mean = mean(trans), sd = sd(trans)):
## ties should not be present for the Kolmogorov-Smirnov test
```

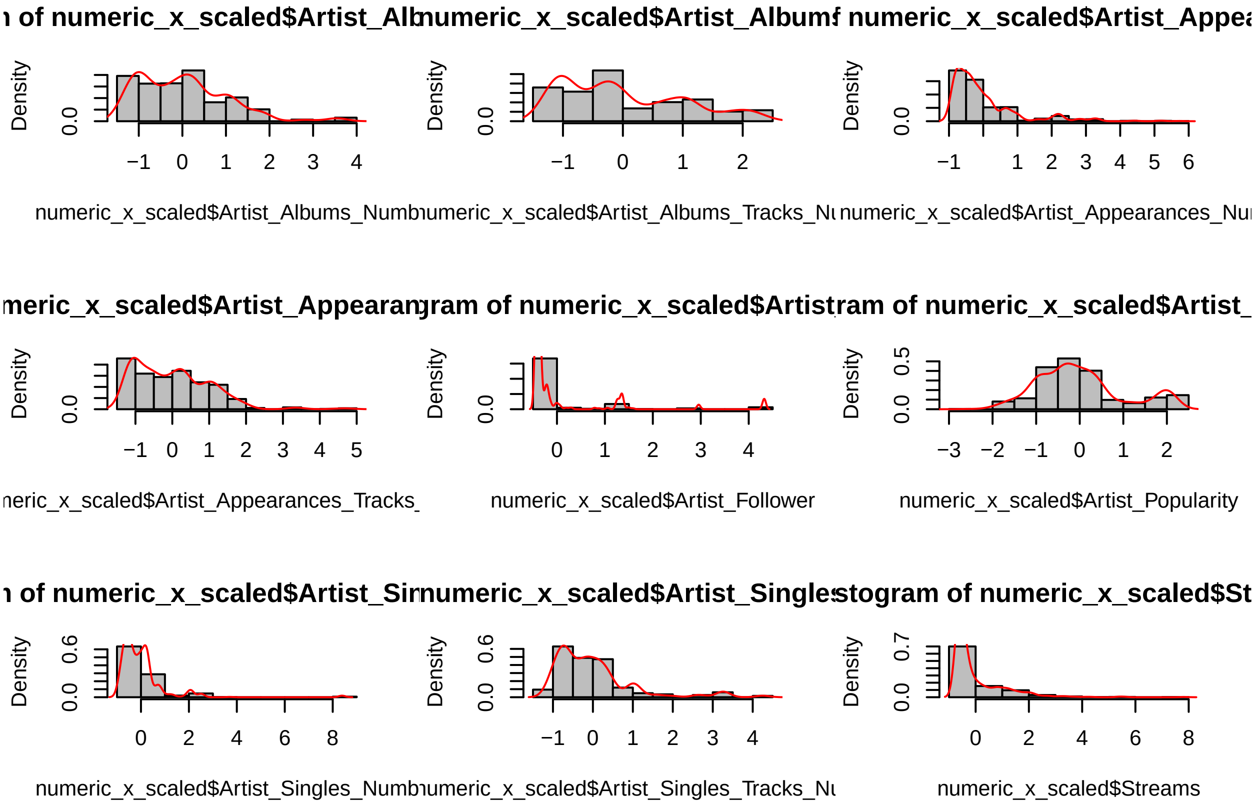
```
critical_value <- 1.3581 / sqrt (length(x_sub))

if (test_statistic > critical_value) {
message(paste("Transformed ", column_name , " is not approximately normally distributed.", test_statisti
} else {
message(paste("Transformed ", column_name , " is approximately normally distributed!", test_statistic, c
}
```

```
## Transformed  viewsCount  is approximately normally distributed! 0.0197285296674387 0.0506133986707077
```

Variable transformations

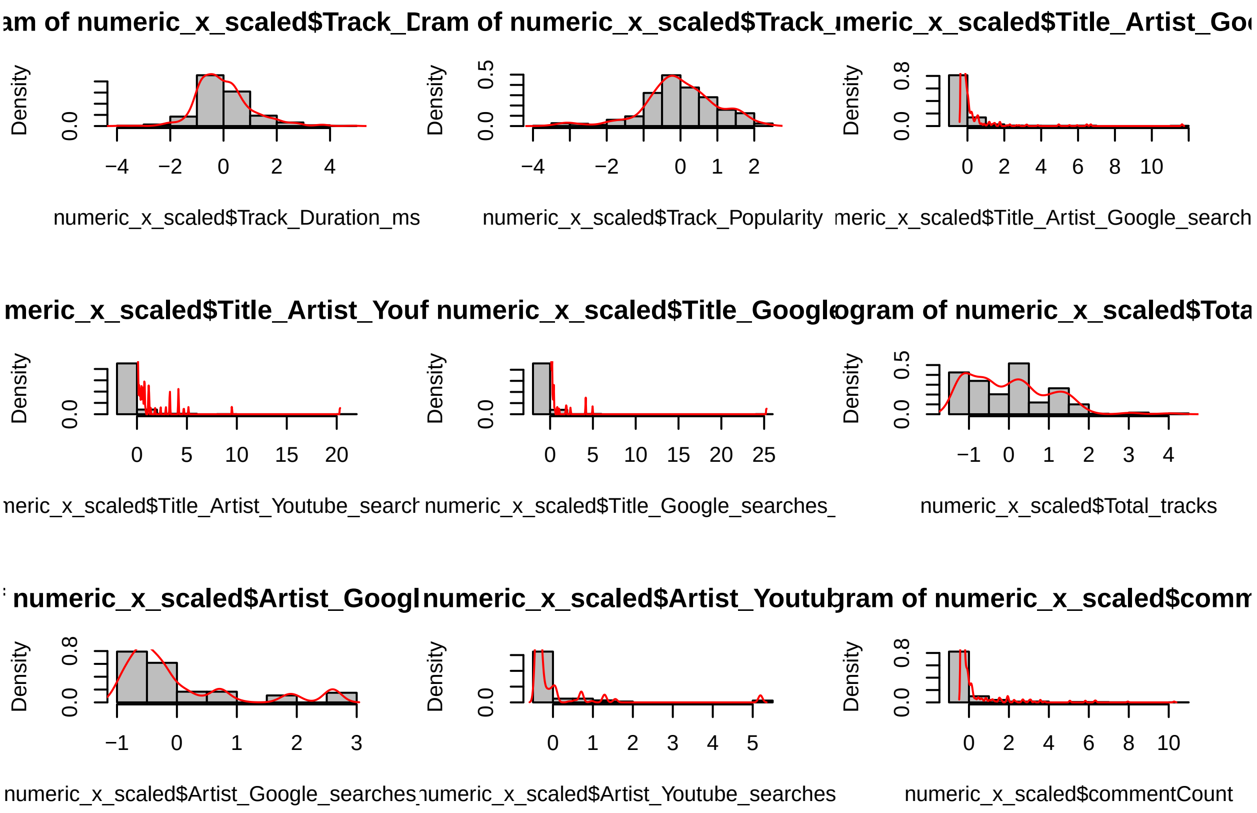
1) Z-Transformation

```r
numeric_x_scaled <- scale(numeric_x, center = TRUE, scale = TRUE)
numeric_x_scaled <- as.data.frame(numeric_x_scaled)

par(mfrow=c(3,3))

hist(numeric_x_scaled$Artist_Albums_Number, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Artist_Albums_Number), col = "red")

hist(numeric_x_scaled$Artist_Albums_Tracks_Number, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Artist_Albums_Tracks_Number), col = "red")

hist(numeric_x_scaled$Artist_Appearances_Number, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Artist_Appearances_Number), col = "red")

hist(numeric_x_scaled$Artist_Appearances_Tracks_Number, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Artist_Appearances_Tracks_Number), col = "red")

hist(numeric_x_scaled$Artist_Follower, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Artist_Follower), col = "red")

hist(numeric_x_scaled$Artist_Popularity, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Artist_Popularity), col = "red")

hist(numeric_x_scaled$Artist_Singles_Number, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Artist_Singles_Number), col = "red")

hist(numeric_x_scaled$Artist_Singles_Tracks_Number, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Artist_Singles_Tracks_Number), col = "red")

hist(numeric_x_scaled$Streams, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Streams), col = "red")
```

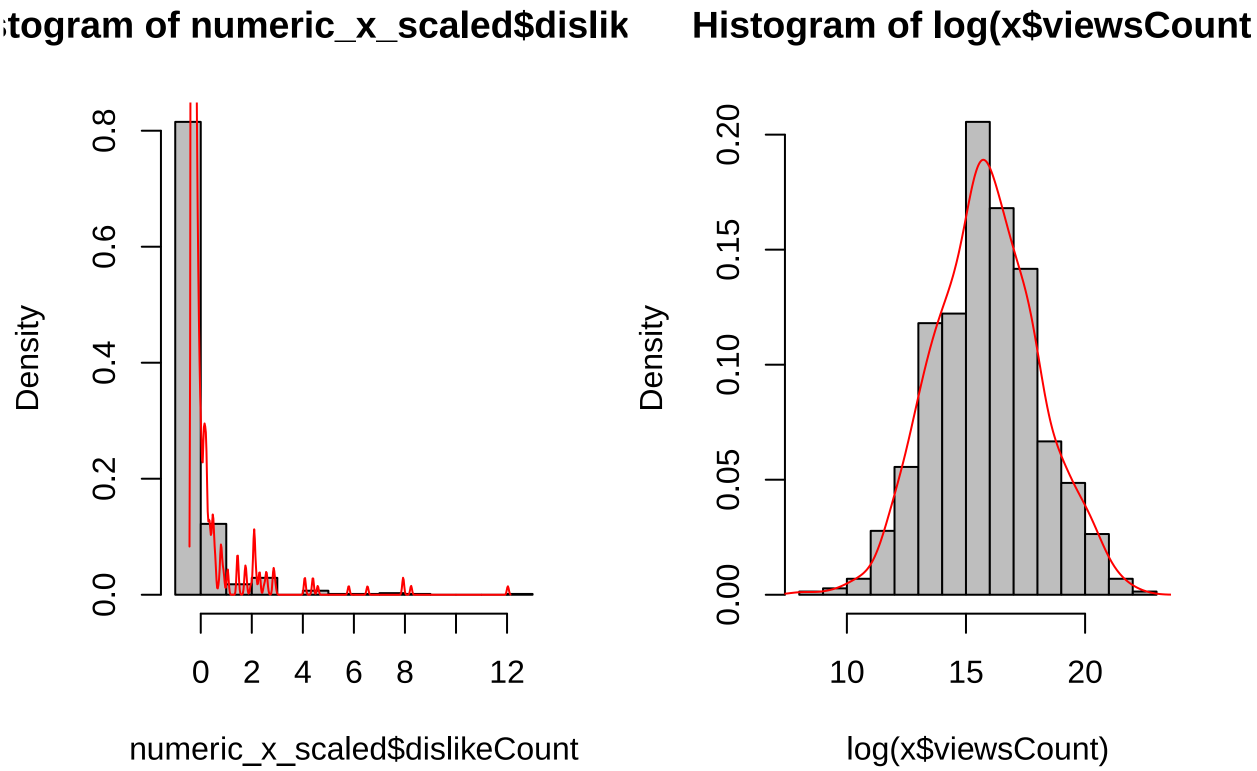


```r
par(mfrow=c(3,3))

hist(numeric_x_scaled$Track_Duration_ms, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Track_Duration_ms), col = "red")

hist(numeric_x_scaled$Track_Popularity, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Track_Popularity), col = "red")

hist(numeric_x_scaled$Title_Artist_Google_searches_11m, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Title_Artist_Google_searches_11m), col = "red")

hist(numeric_x_scaled$Title_Artist_Youtube_searches_11m, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Title_Artist_Youtube_searches_11m), col = "red")

hist(numeric_x_scaled$Title_Google_searches_11m, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Title_Google_searches_11m), col = "red")

hist(numeric_x_scaled$Total_tracks, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Total_tracks), col = "red")

hist(numeric_x_scaled$Artist_Google_searches_11m, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Artist_Google_searches_11m), col = "red")

hist(numeric_x_scaled$Artist_Youtube_searches_11m, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$Artist_Youtube_searches_11m), col = "red")
```

```r
hist(numeric_x_scaled$commentCount, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$commentCount, na.rm = TRUE), col = "red")
```



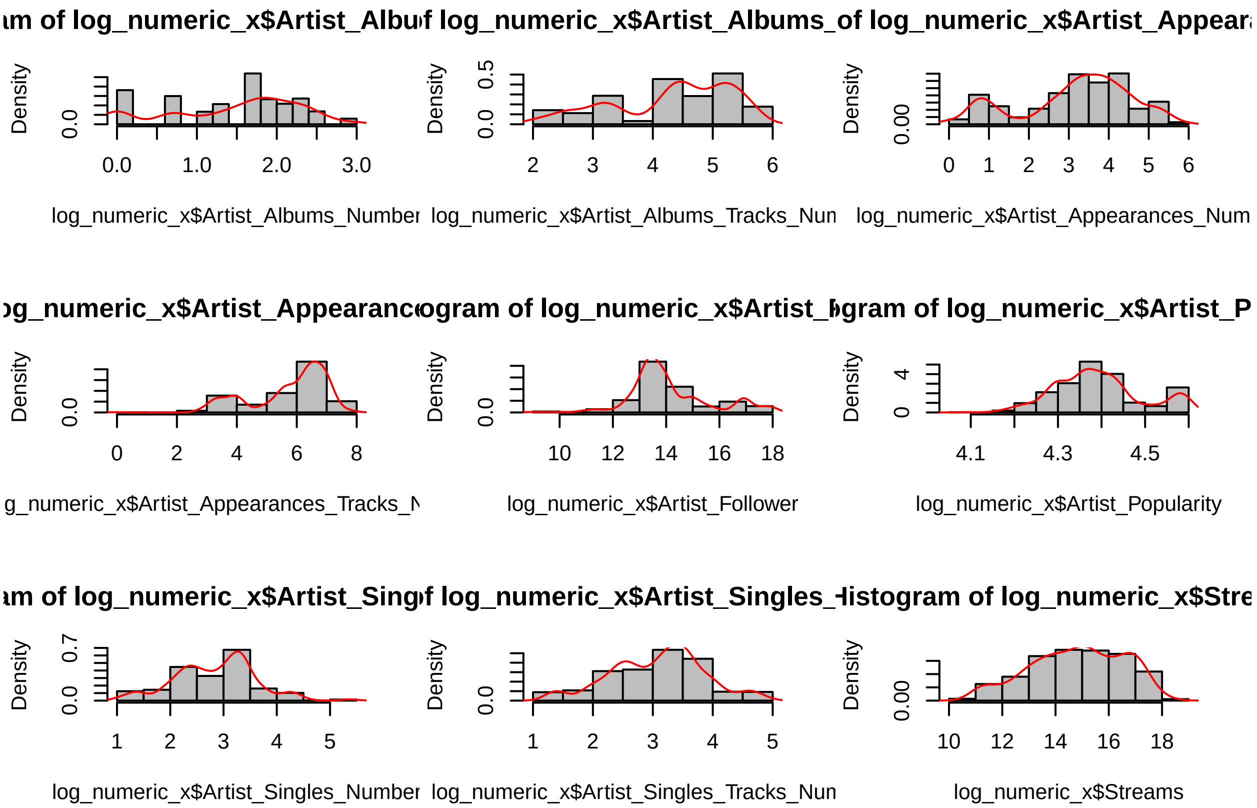
```r
par(mfrow=c(1,2))

hist(numeric_x_scaled$dislikeCount, probability = TRUE, col = "gray")
lines(density(numeric_x_scaled$dislikeCount), col = "red")

hist(log(x$viewsCount), probability = TRUE, col = "gray")
lines(density(log(x$viewsCount)), col = "red")
```



2) Log-Transformation

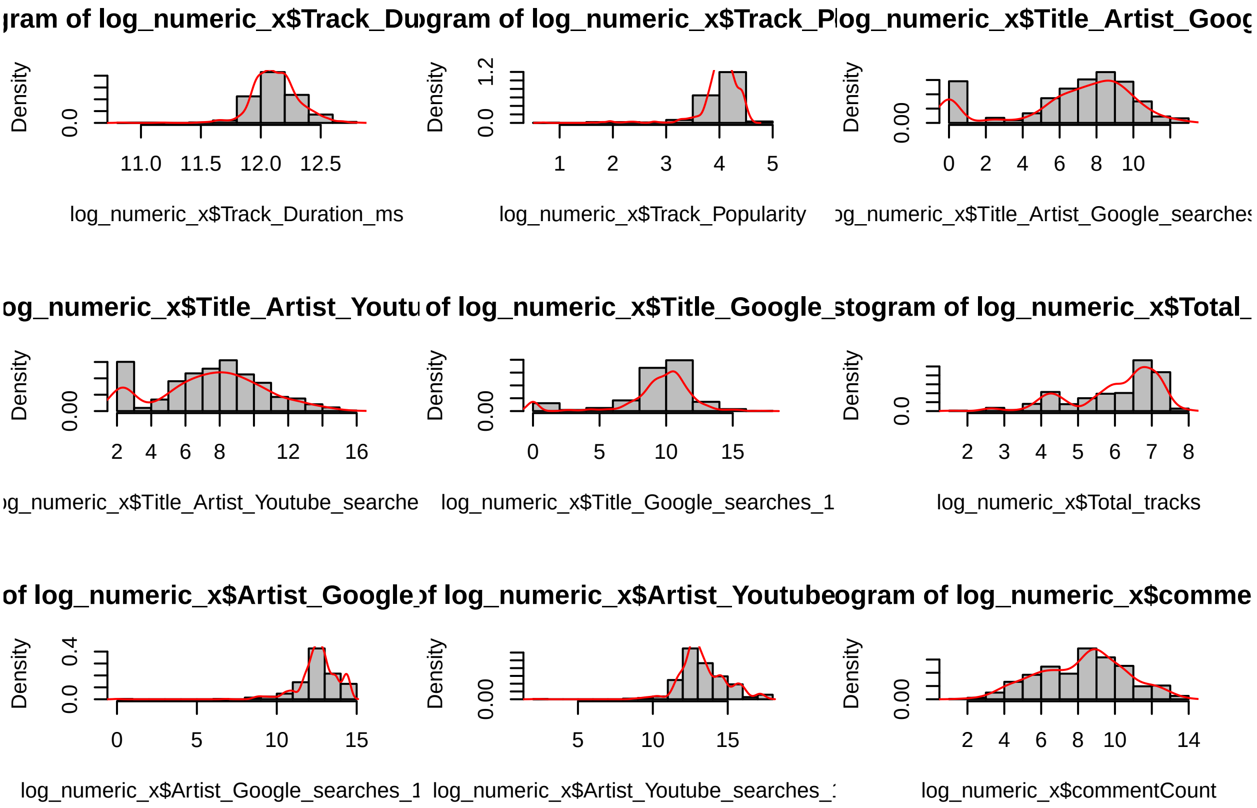
```r
log_numeric_x <- log(numeric_x)


par(mfrow=c(3,3))

hist(log_numeric_x$Artist_Albums_Number, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Artist_Albums_Number), col = "red")

hist(log_numeric_x$Artist_Albums_Tracks_Number, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Artist_Albums_Tracks_Number), col = "red")

hist(log_numeric_x$Artist_Appearances_Number, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Artist_Appearances_Number), col = "red")

hist(log_numeric_x$Artist_Appearances_Tracks_Number, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Artist_Appearances_Tracks_Number), col = "red")

hist(log_numeric_x$Artist_Follower, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Artist_Follower), col = "red")

hist(log_numeric_x$Artist_Popularity, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Artist_Popularity), col = "red")

hist(log_numeric_x$Artist_Singles_Number, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Artist_Singles_Number), col = "red")
```

```r
hist(log_numeric_x$Artist_Singles_Tracks_Number, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Artist_Singles_Tracks_Number), col = "red")

hist(log_numeric_x$Streams, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Streams), col = "red")
```



```r
par(mfrow=c(3,3))

hist(log_numeric_x$Track_Duration_ms, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Track_Duration_ms), col = "red")

hist(log_numeric_x$Track_Popularity, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Track_Popularity), col = "red")

hist(log_numeric_x$Title_Artist_Google_searches_11m, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Title_Artist_Google_searches_11m), col = "red")

hist(log_numeric_x$Title_Artist_Youtube_searches_11m, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Title_Artist_Youtube_searches_11m), col = "red")

hist(log_numeric_x$Title_Google_searches_11m, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Title_Google_searches_11m), col = "red")

hist(log_numeric_x$Total_tracks, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Total_tracks), col = "red")
```

```r
hist(log_numeric_x$Artist_Google_searches_11m, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Artist_Google_searches_11m), col = "red")

hist(log_numeric_x$Artist_Youtube_searches_11m, probability = TRUE, col = "gray")
lines(density(log_numeric_x$Artist_Youtube_searches_11m), col = "red")

hist(log_numeric_x$commentCount, probability = TRUE, col = "gray")
lines(density(log_numeric_x$commentCount, na.rm = TRUE), col = "red")
```



```r
par(mfrow=c(1,2))

hist(log_numeric_x$dislikeCount, probability = TRUE, col = "gray")
lines(density(log_numeric_x$dislikeCount), col = "red")

hist(log_numeric_x$viewsCount, probability = TRUE, col = "gray")
lines(density(log_numeric_x$viewsCount), col = "red")
```

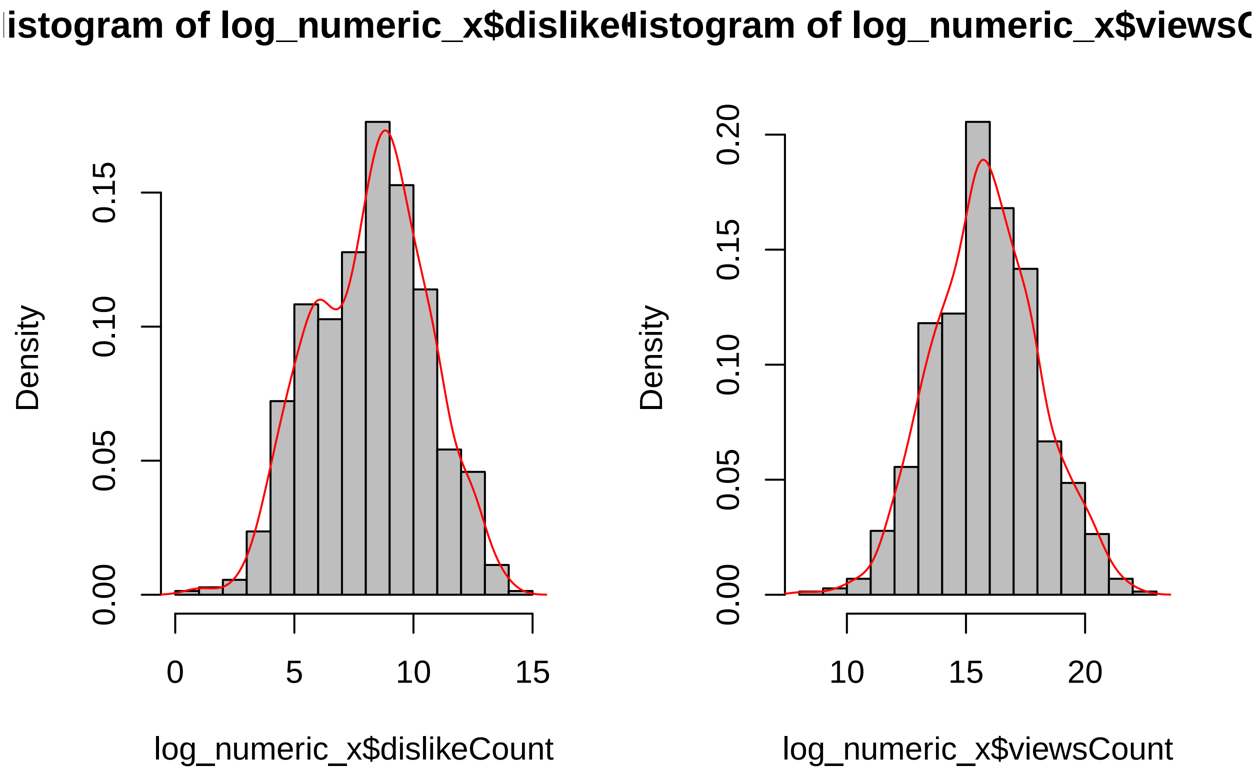
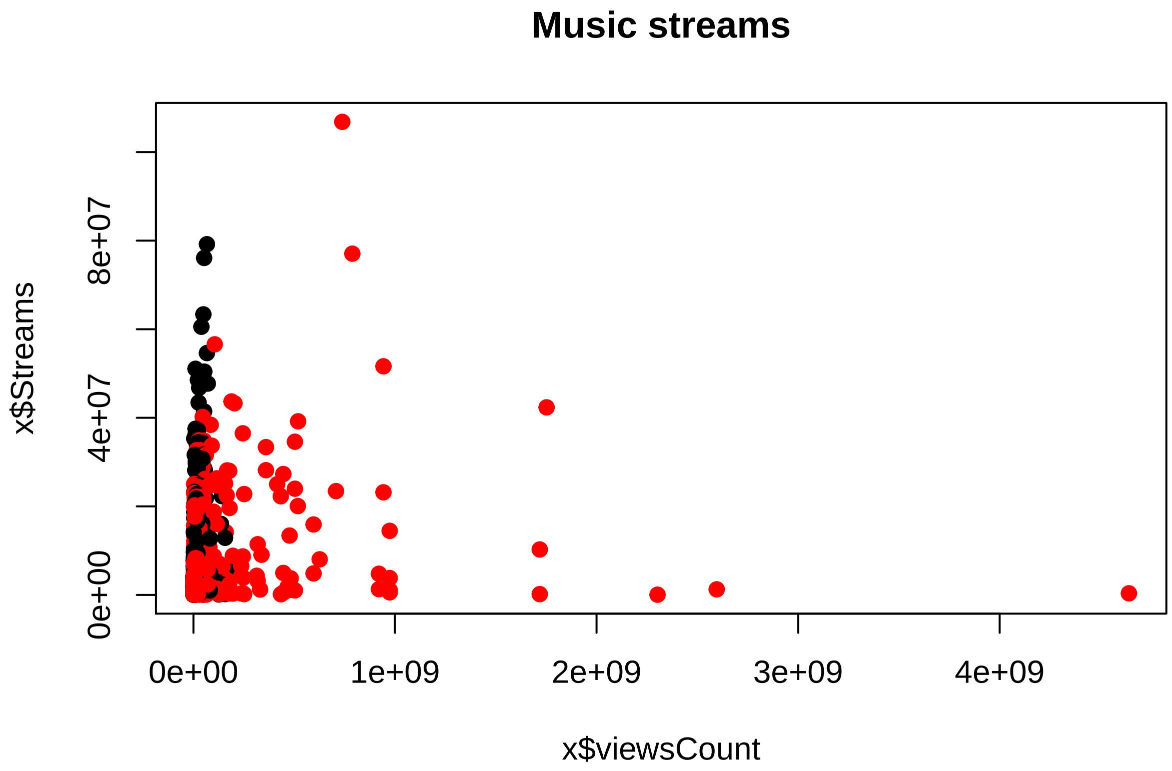


Table by genre

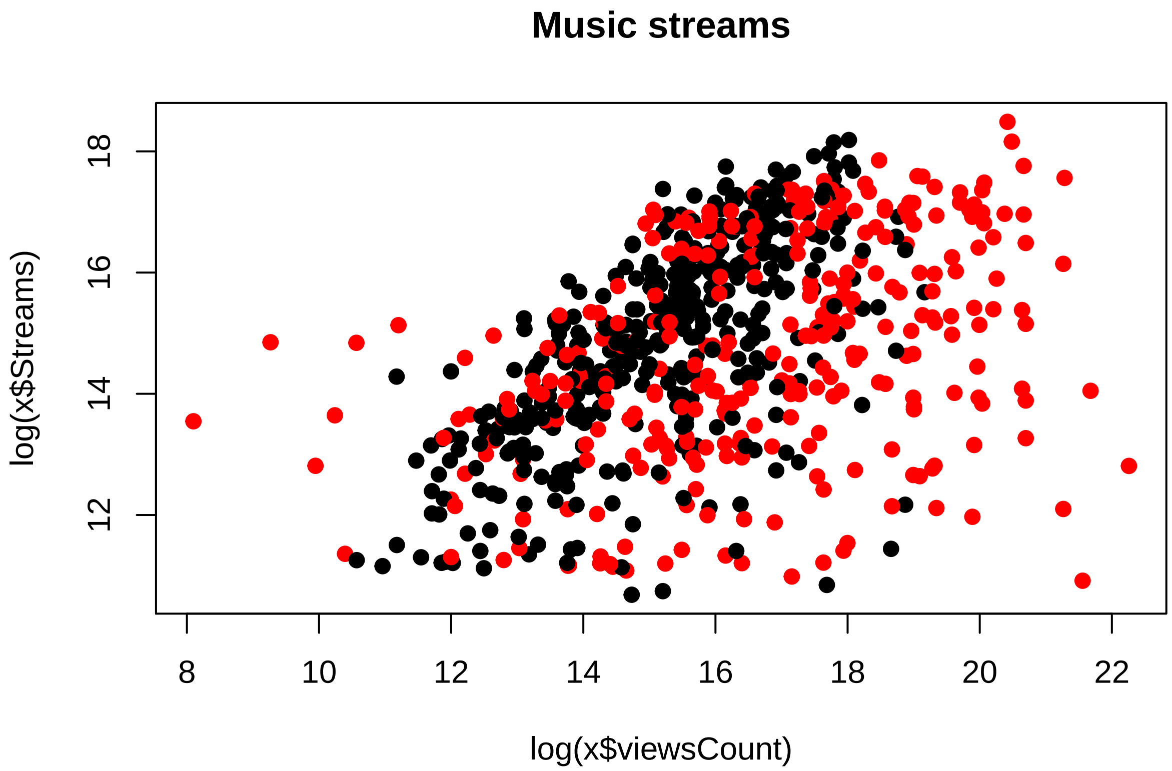
```
table(x$Genre)
```

```
##
##   dance      edm Hip Hop    house    latin    metal      pop      r&b      rap
##       4        8     411       13        2       11      140        3      126
##    rock
##       2
```

```
##1

col <- ifelse(x$Genre == "Hip Hop", "black", "red")

plot(x$viewsCount, x$Streams, main="Music streams", pch=19, col=col)
```



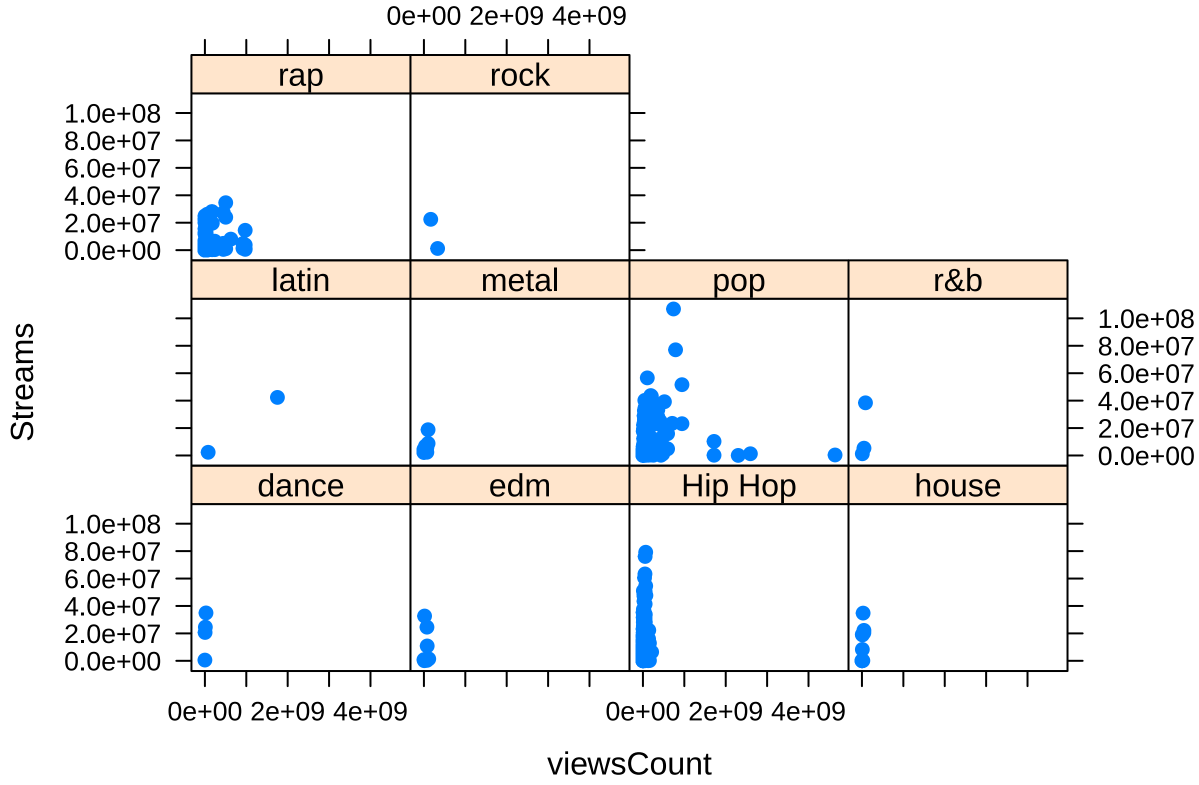
```
plot(log(x$viewsCount), log(x$Streams), main="Music streams", pch=19, col=col)
```



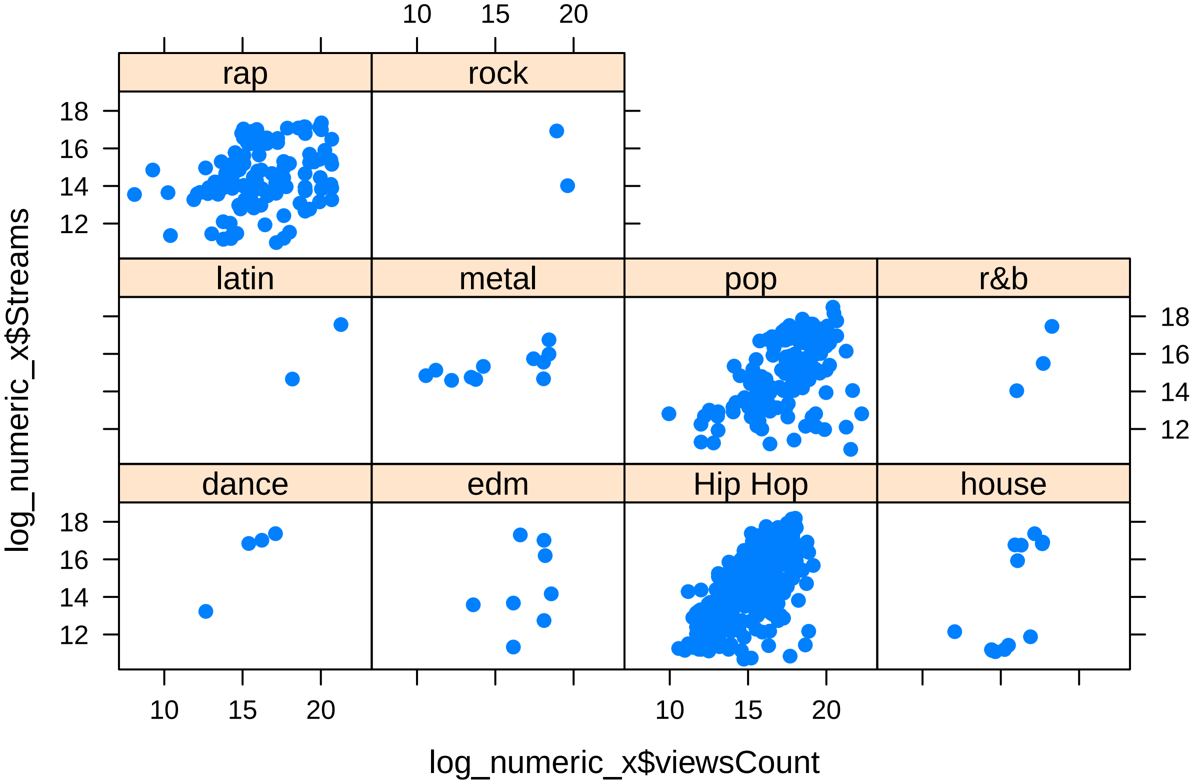
```
library("lattice")
```

```
## Warning: package 'lattice' was built under R version 3.5.1
```

```
xyplot(Streams~viewsCount|Genre, data=x, pch=19)
```



```
xyplot(log_numeric_x$Streams~log_numeric_x$viewsCount|x$Genre, pch=19)
```

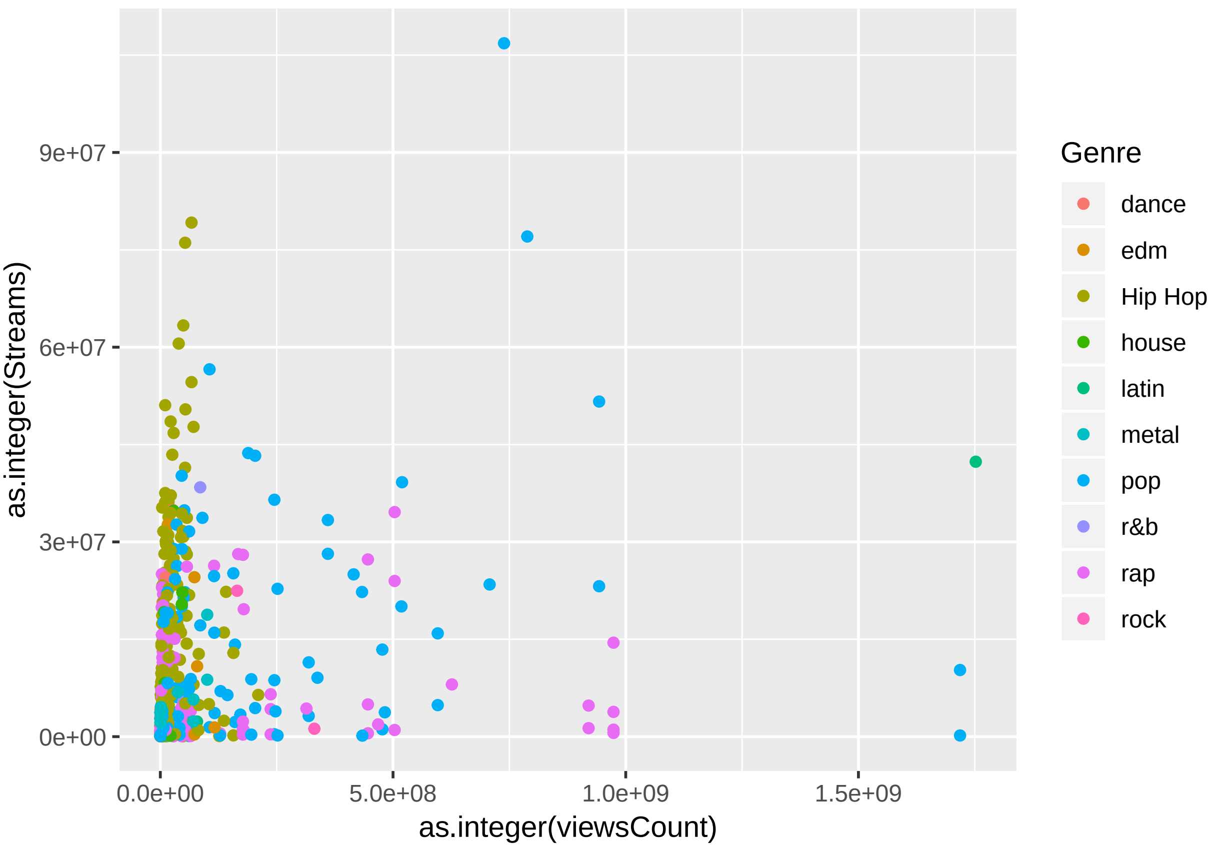


```
library("ggplot2")
```

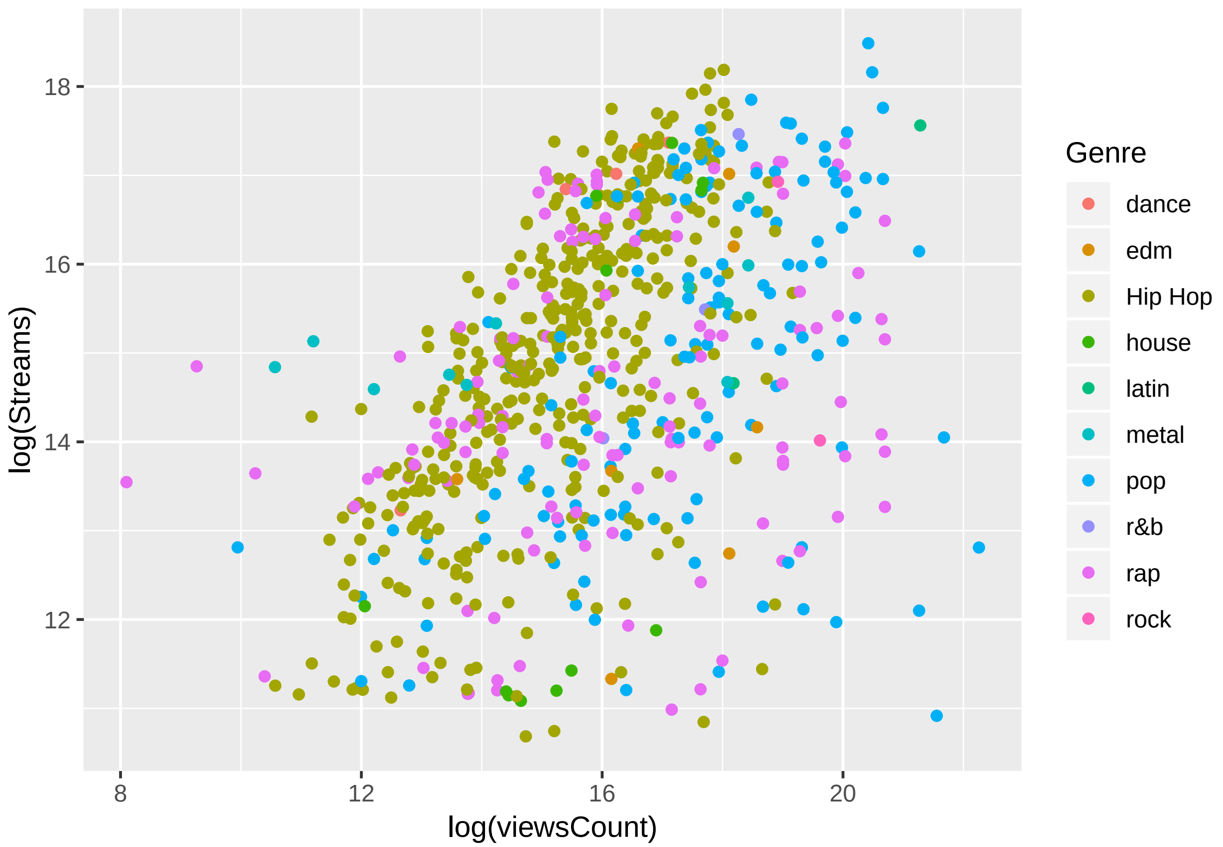
```
## Warning: package 'ggplot2' was built under R version 3.5.1

##
## Attaching package: 'ggplot2'

## The following objects are masked from 'package:psych':
##
##     %+%, alpha
```

```
d <-ggplot(x, aes(x=as.integer(viewsCount), y=as.integer(Streams), colour=Genre))
d + geom_point(shape=19)
```

```
## Warning in FUN(X[[i]], ...): NAs introduced by coercion to integer range

## Warning in FUN(X[[i]], ...): NAs introduced by coercion to integer range

## Warning: Removed 3 rows containing missing values (geom_point).
```

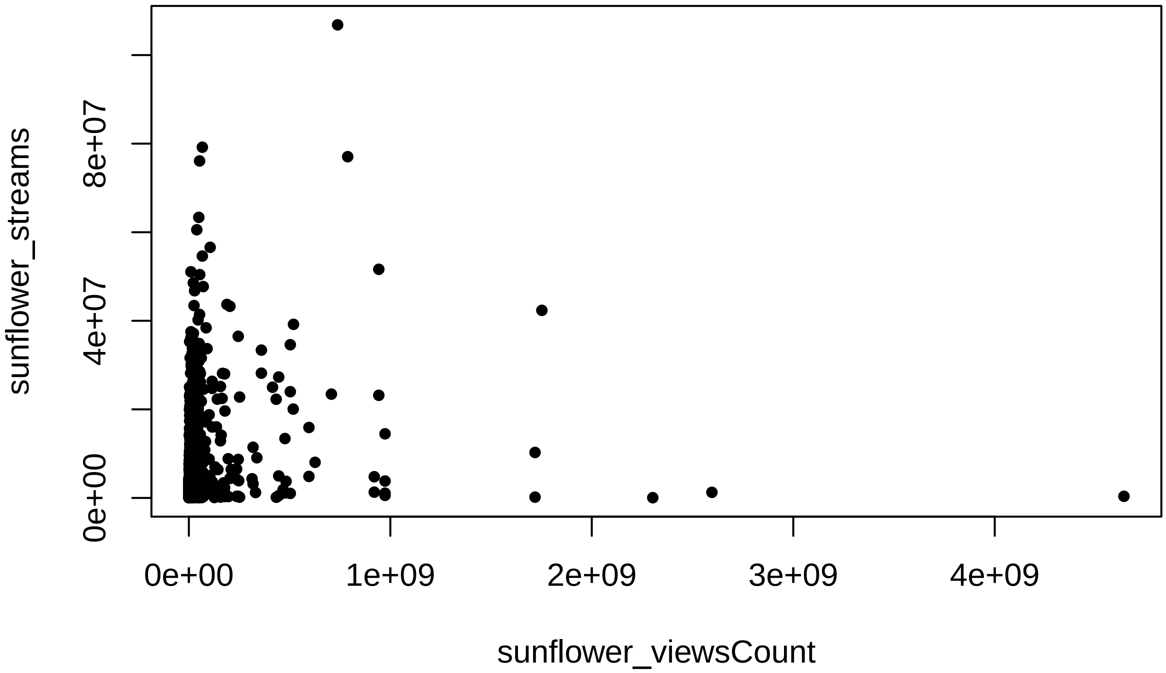


```
d <-ggplot(x, aes(x=log(viewsCount), y=log(Streams), colour=Genre))
d + geom_point(shape=19)
```



Using sunflower plot to overcome problem of overplotting.

```
sunflower_viewsCount <- 2*round(x$viewsCount/2)
sunflower_streams  <- 2*round(x$Streams/2)
sunflowerplot(sunflower_streams~sunflower_viewsCount)
```



```
library("Rmpfr")
```

```
## Warning: package 'Rmpfr' was built under R version 3.5.3
```

```
## Loading required package: gmp
```

```
## Warning: package 'gmp' was built under R version 3.5.3
```

```
##
## Attaching package: 'gmp'
```

```
## The following object is masked from 'package:rio':
##
##     factorize
```

```
## The following objects are masked from 'package:base':
##
##     %*%, apply, crossprod, matrix, tcrossprod
```

```
## C code of R package 'Rmpfr': GMP using 64 bits per limb
```

```
##
## Attaching package: 'Rmpfr'
```

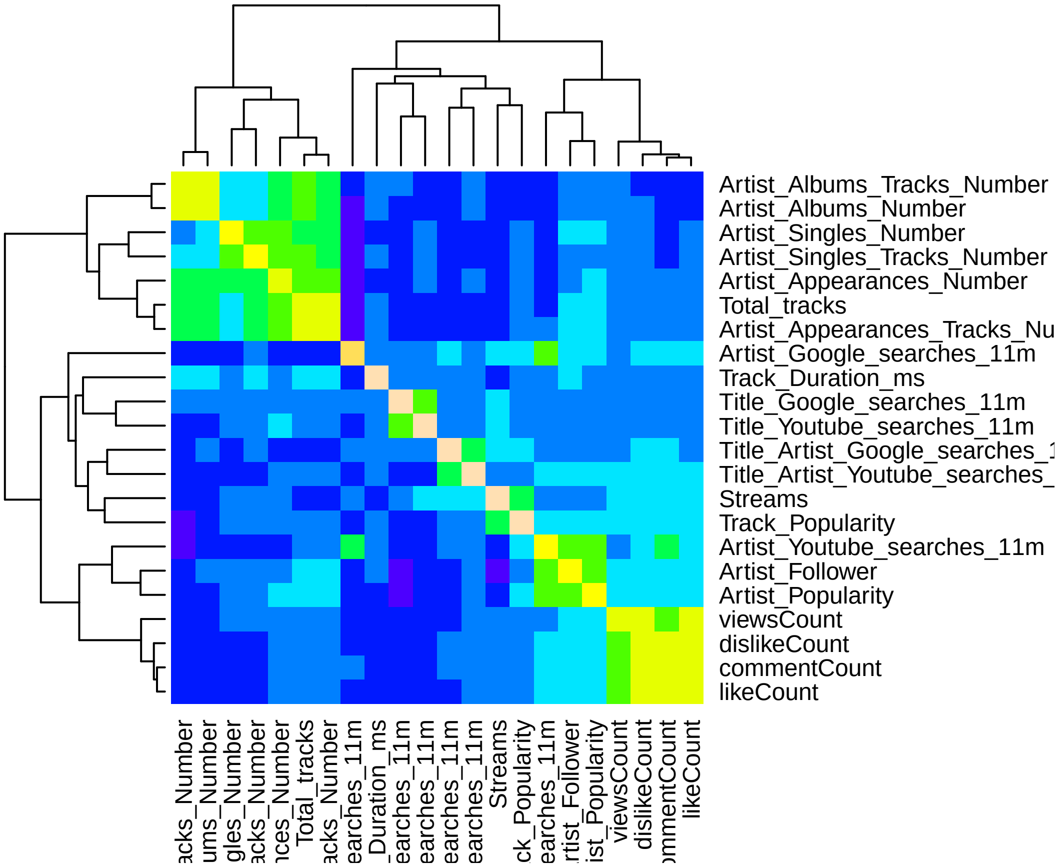
```
## The following object is masked from 'package:gmp':
##
##     outer
```

```
## The following objects are masked from 'package:stats':
##
##     dbinom, dgamma, dnorm, dpois, pnorm
## The following objects are masked from 'package:base':
##
##     cbind, pmax, pmin, rbind
```

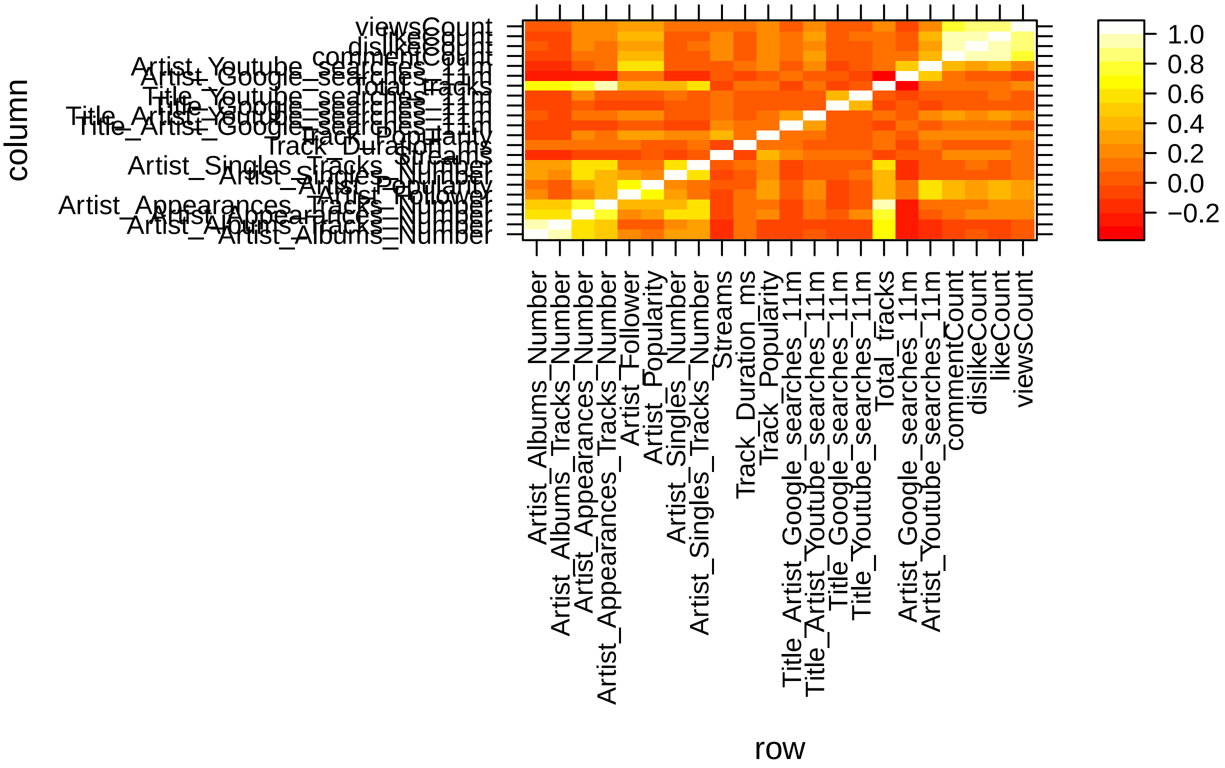
```
# (one <- mpfr(1, 120))

cor <- cor(numeric_x)
drop.cor_cols <- c('Artist_Compilations_Number', 'Artist_Compilations_Tracks_Number')
numeric_cor_x <- select(numeric_x, -one_of(drop.cor_cols))

numeric_cor_x$viewsCount <- as.numeric(numeric_cor_x$viewsCount)

str(numeric_cor_x)
```

```
## 'data.frame':    720 obs. of  22 variables:
##  $ Artist_Albums_Number              : int  0 0 1 1 1 1 1 1 1 1 ...
##  $ Artist_Albums_Tracks_Number       : int  0 0 8 8 8 8 8 8 8 8 ...
##  $ Artist_Appearances_Number         : int  9 9 2 2 2 2 2 2 2 2 ...
##  $ Artist_Appearances_Tracks_Number : int  502 502 30 30 30 30 30 30 30 30 ...
##  $ Artist_Follower                   : int  713401 713401 601346 601346 601346 601346 601346 601346 60
##  $ Artist_Popularity                 : int  91 91 83 83 83 83 83 83 83 83 ...
##  $ Artist_Singles_Number             : int  3 3 15 15 15 15 15 15 15 15 ...
##  $ Artist_Singles_Tracks_Number      : int  10 10 15 15 15 15 15 15 15 15 ...
##  $ Streams                           : int  106824437 2327995 79193552 54619683 48552840 46784729 434
##  $ Track_Duration_ms                 : int  209754 200755 157093 158853 176066 163146 139693 191760 17
##  $ Track_Popularity                  : int  76 72 78 77 73 75 73 75 69 69 ...
##  $ Title_Artist_Google_searches_11m : int  20904 572 8880 8880 1975 1156 3260 10880 220 568 ...
##  $ Title_Artist_Youtube_searches_11m: int  308911 7320 7660 7660 1530 990 2240 7915 154 441 ...
##  $ Title_Google_searches_11m         : int  1288732 2799 4805454 4805454 47025 33165 47709 45925 8977
##  $ Title_Youtube_searches_11m        : int  18353181 33600 3446454 3446454 32325 28872 38436 31975 59
##  $ Total_tracks                      : int  512 512 53 53 53 53 53 53 53 53 ...
##  $ Artist_Google_searches_11m        : int  299212 299212 1468281 1468281 1468281 1468281 1468281 146
##  $ Artist_Youtube_searches_11m       : int  2451500 2451500 1076400 1076400 1076400 1076400 1076400 10
##  $ commentCount                      : int  172604 2272 22183 22183 13376 10741 8662 303 5795 4485 ..
##  $ dislikeCount                      : int  317322 3194 27802 27802 11440 12957 10493 605 5333 4287 .
##  $ likeCount                         : int  7424686 109395 748270 748270 385252 378780 299481 24361 2
##  $ viewsCount                        : num  7.39e+08 1.03e+07 6.70e+07 6.70e+07 2.22e+07 ...
```

```
clean_cor <- cor(numeric_cor_x[complete.cases(numeric_cor_x), ])
heatmap(clean_cor, revC=T, col=topo.colors(10))
```



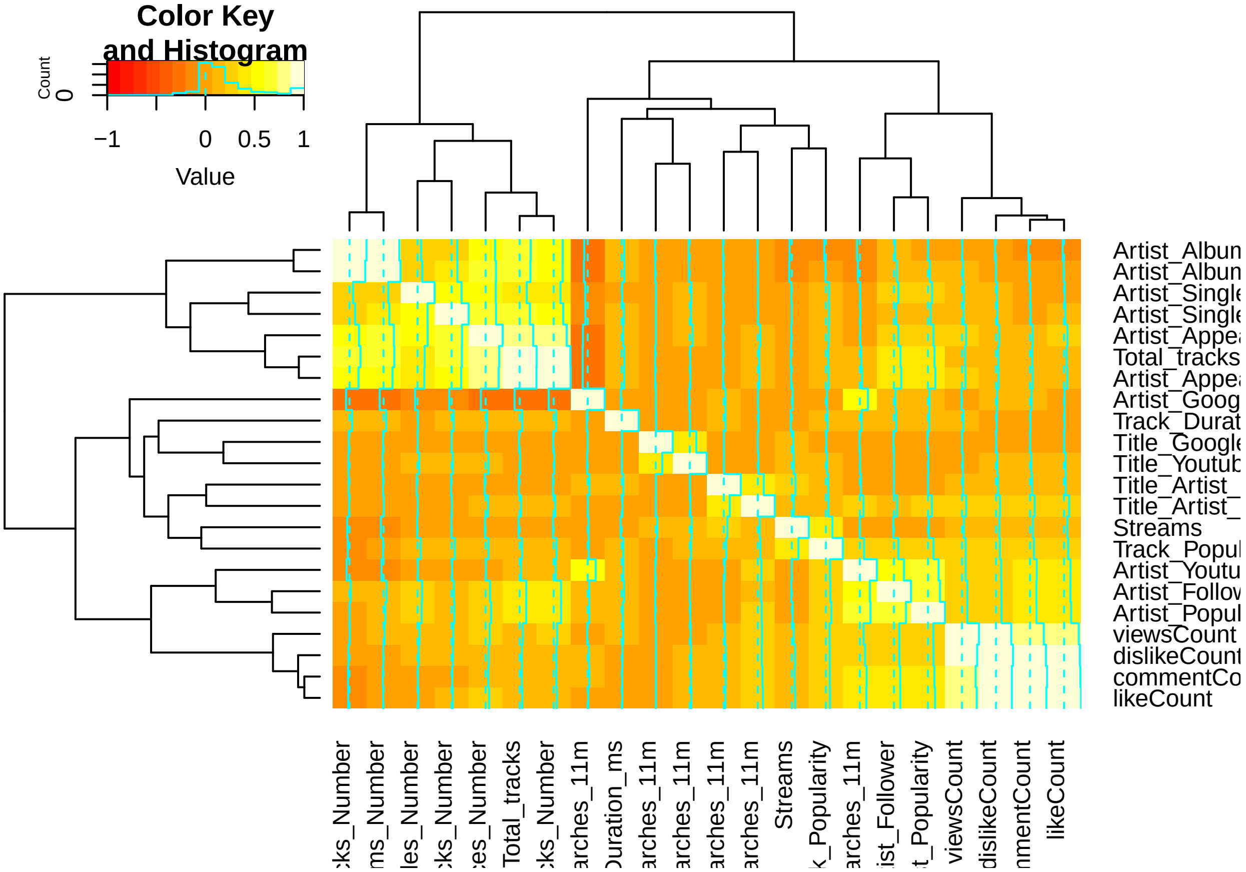
```r
library("lattice")
levelplot(clean_cor, scales=list(x=list(rot=90)), aspect = "fill", col.regions=heat.colors(100))
```



```
library("gplots")
```

```
## Warning: package 'gplots' was built under R version 3.5.2

##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```
gplots::heatmap.2(clean_cor, revC=T, na.rm=T)
```

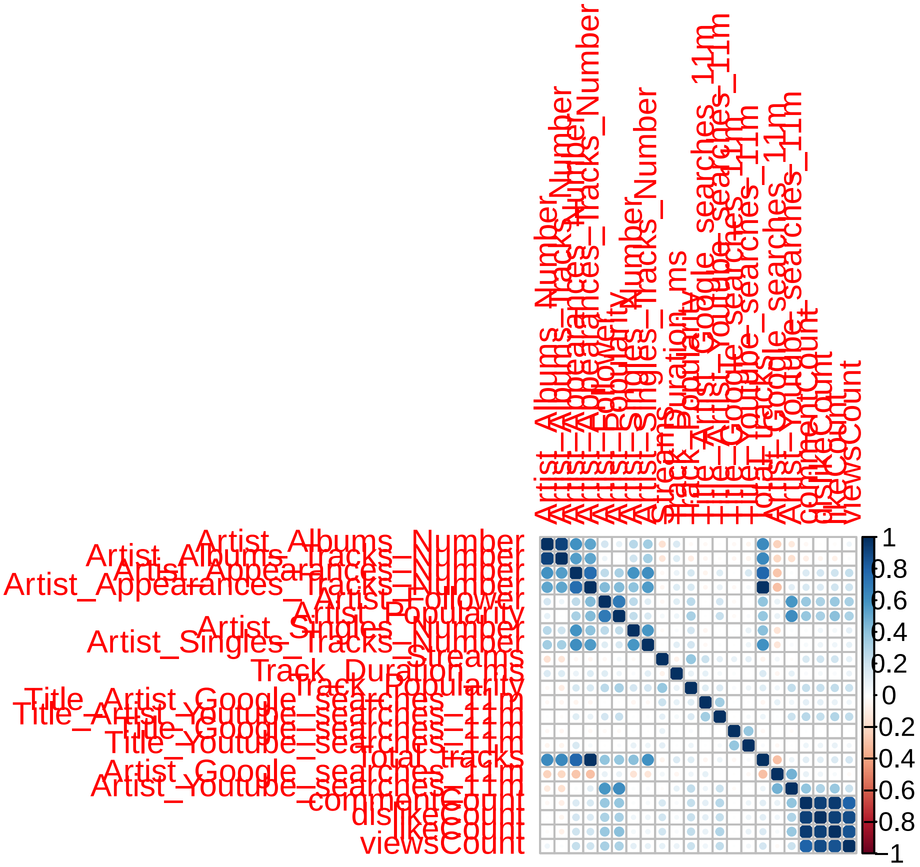


Steiger's Z test for significance of (Bravais-Pearson) correlation coefficients

```r
library(corrplot)
```

```
## corrplot 0.84 loaded
```

```r
corrplot(clean_cor, method="circle")
```



```
cor.mtest <- function(mat, ...) {
  mat <- as.matrix(mat)
  n <- ncol(mat)
  p.mat<- matrix(NA, n, n)
  diag(p.mat) <- 0
  for (i in 1:(n - 1)) {
    for (j in (i + 1):n) {
      tmp <- cor.test(mat[, i], mat[, j], ...)
      p.mat[i, j] <- p.mat[j, i] <- tmp$p.value
    }
  }
  colnames(p.mat) <- rownames(p.mat) <- colnames(mat)
  p.mat
}

# matrix of the p-value of the correlation
p.mat <- cor.mtest(clean_cor)

col <- colorRampPalette(c("#BB4444", "#EE9988", "#FFFFFF", "#77AADD", "#4477AA"))
significance_level <- 0.05

corrplot(clean_cor, method="color", col=col(200),
         type="upper", order="hclust",
         addCoef.col = "black", # Add coefficient of correlation
         tl.col="black", tl.srt=90, #Text label color and rotation
         # Combine with significance
         p.mat = p.mat, sig.level = significance_level, insig = "blank",
         # hide correlation coefficient on the principal diagonal
         diag=FALSE)
```

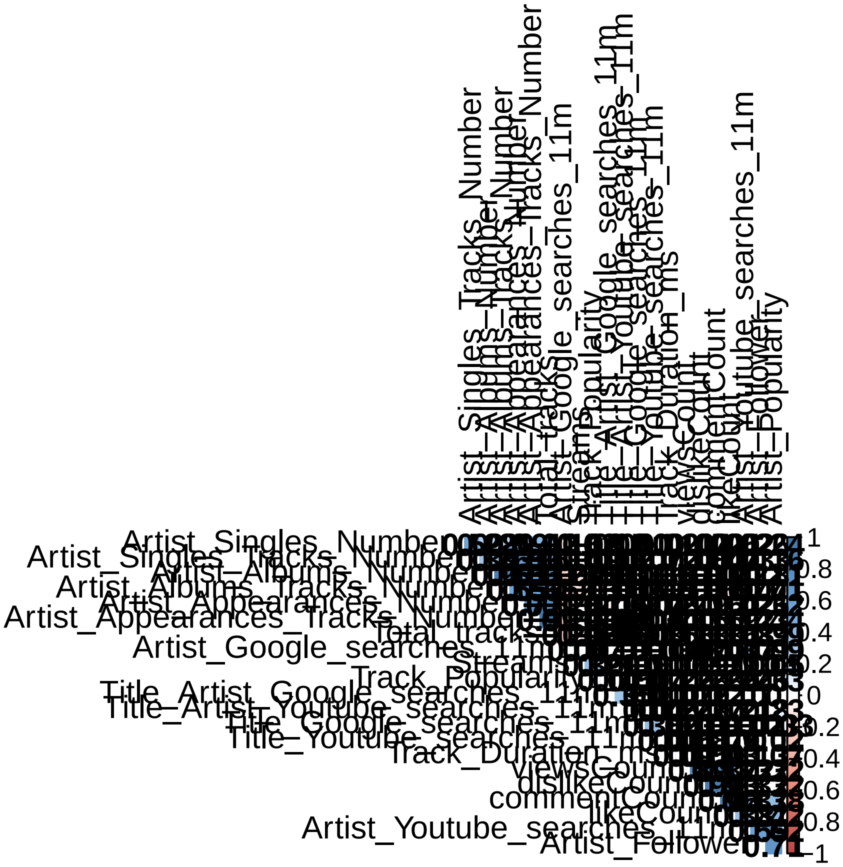
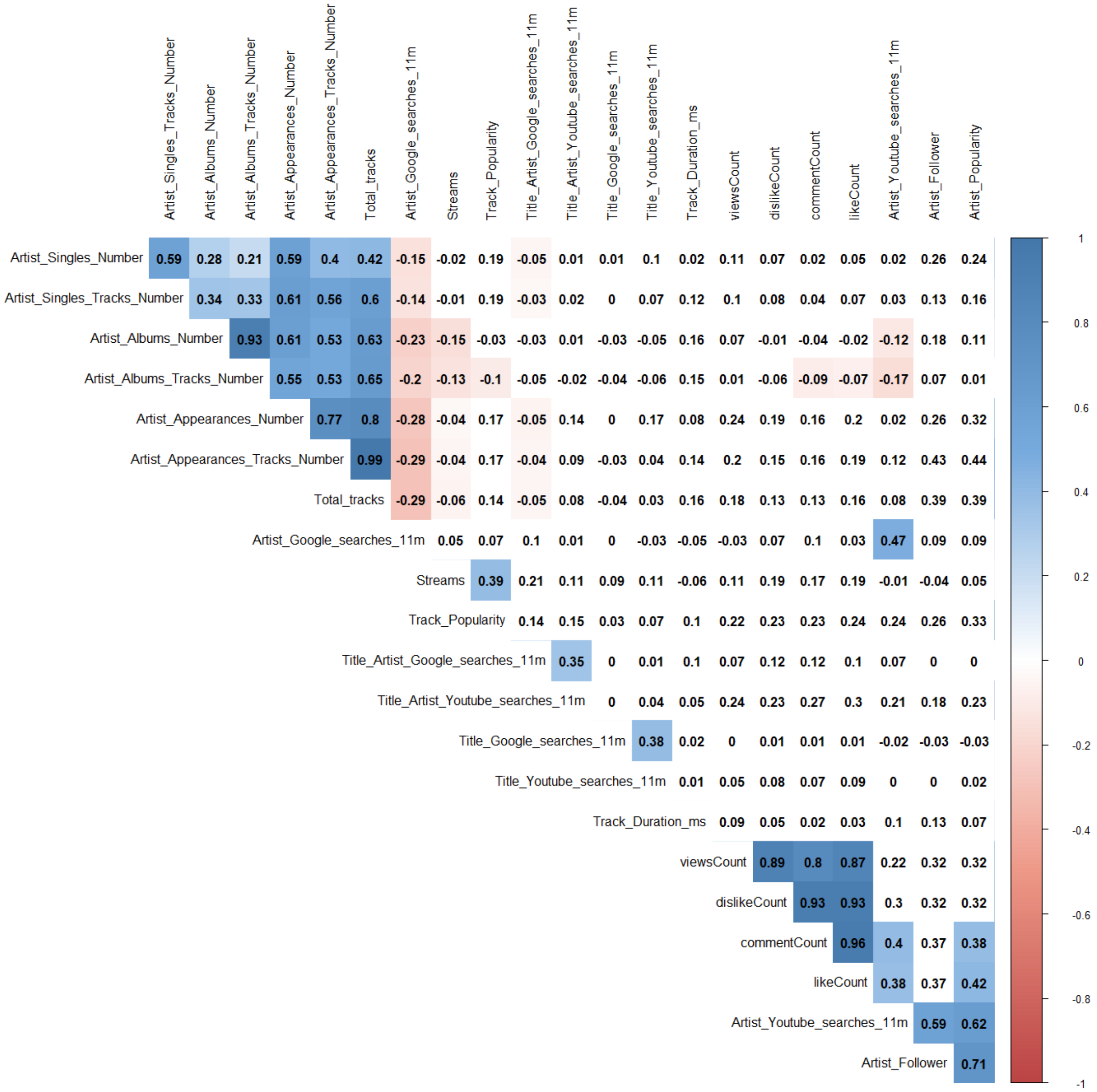


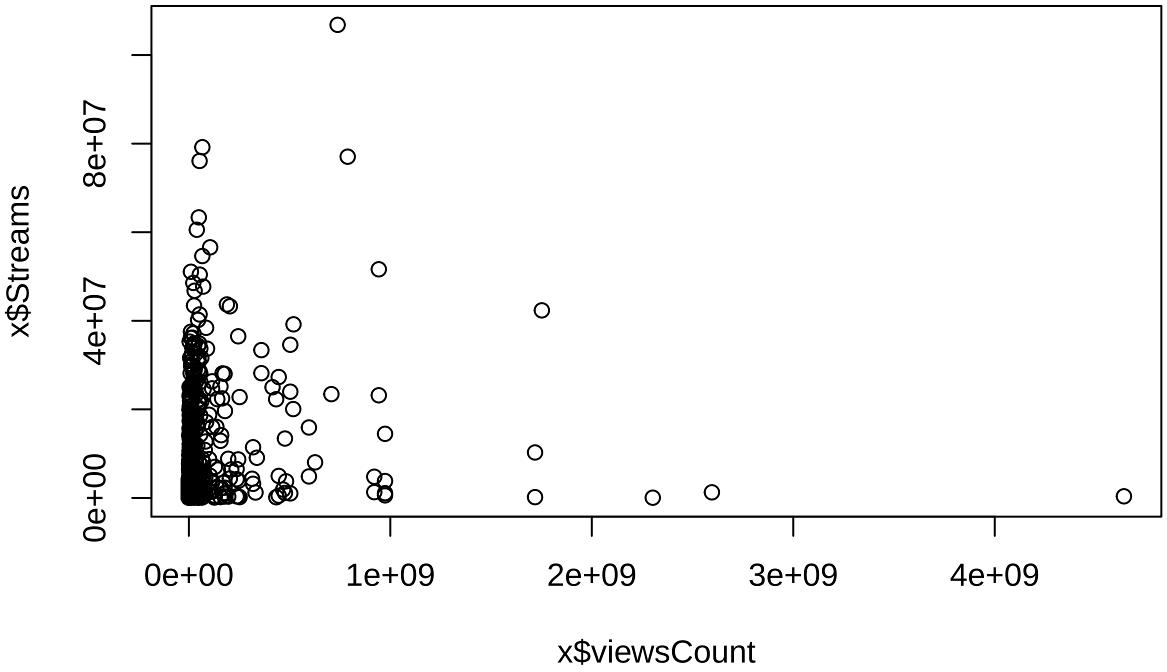
Illustration of assumption that two variables are jointly normally distributed to perform Steiger's Z test:

```
plot(x$viewsCount, x$Streams)
```

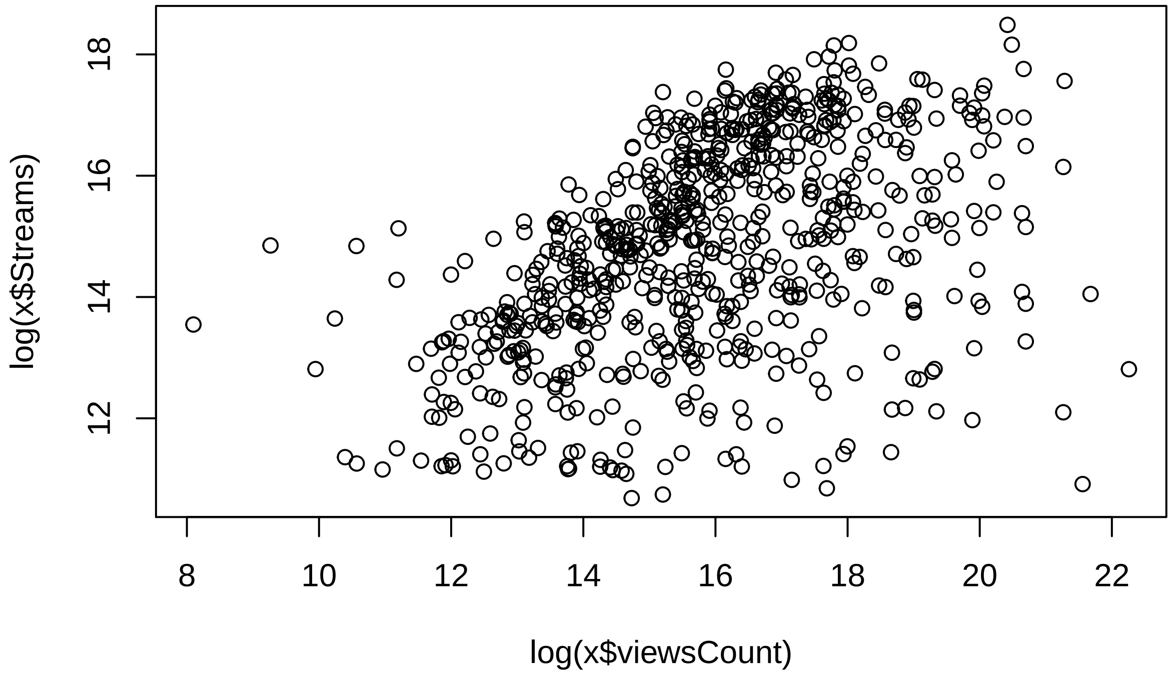


	Artist_Singles_Tracks_Number	Artist_Albums_Number	Artist_Albums_Tracks_Number	Artist_Appearances_Number	Artist_Appearances_Tracks_Number	Total_tracks	Artist_Google_searches_11m	Streams	Track_Popularity	Title_Artist_Google_searches_11m	Title_Artist_Youtube_searches_11m	Title_Google_searches_11m	Title_Youtube_searches_11m	Track_Duration_ms	viewsCount	dislikeCount	commentCount	likeCount	Artist_Youtube_searches_11m	Artist_Follower	Artist_Popularity
Artist_Singles_Number	0.59	0.28	0.21	0.59	0.4	0.42	-0.15	-0.02	0.19	-0.05	0.01	0.01	0.1	0.02	0.11	0.07	0.02	0.05	0.02	0.26	0.24
Artist_Singles_Tracks_Number		0.34	0.33	0.61	0.56	0.6	-0.14	-0.01	0.19	-0.03	0.02	0	0.07	0.12	0.1	0.08	0.04	0.07	0.03	0.13	0.16
Artist_Albums_Number			0.93	0.61	0.53	0.63	-0.23	-0.15	-0.03	-0.03	0.01	-0.03	-0.05	0.16	0.07	-0.01	-0.04	-0.02	-0.12	0.18	0.11
Artist_Albums_Tracks_Number				0.55	0.53	0.65	-0.2	-0.13	-0.1	-0.05	-0.02	-0.04	-0.06	0.15	0.01	-0.06	-0.09	-0.07	-0.17	0.07	0.01
Artist_Appearances_Number					0.77	0.8	-0.28	-0.04	0.17	-0.05	0.14	0	0.17	0.08	0.24	0.19	0.16	0.2	0.02	0.26	0.32
Artist_Appearances_Tracks_Number						0.99	-0.29	-0.04	0.17	-0.04	0.09	-0.03	0.04	0.14	0.2	0.15	0.16	0.19	0.12	0.43	0.44
Total_tracks							-0.29	-0.06	0.14	-0.05	0.08	-0.04	0.03	0.16	0.18	0.13	0.13	0.16	0.08	0.39	0.39
Artist_Google_searches_11m								0.05	0.07	0.1	0.01	0	-0.03	-0.05	-0.03	0.07	0.1	0.03	0.47	0.09	0.09
Streams									0.39	0.21	0.11	0.09	0.11	-0.06	0.11	0.19	0.17	0.19	-0.01	-0.04	0.05
Track_Popularity										0.14	0.15	0.03	0.07	0.1	0.22	0.23	0.23	0.24	0.24	0.26	0.33
Title_Artist_Google_searches_11m											0.35	0	0.01	0.1	0.07	0.12	0.12	0.1	0.07	0	0
Title_Artist_Youtube_searches_11m												0	0.04	0.05	0.24	0.23	0.27	0.3	0.21	0.18	0.23
Title_Google_searches_11m													0.38	0.02	0	0.01	0.01	0.01	-0.02	-0.03	-0.03
Title_Youtube_searches_11m														0.01	0.05	0.08	0.07	0.09	0	0	0.02
Track_Duration_ms															0.09	0.05	0.02	0.03	0.1	0.13	0.07
viewsCount																0.89	0.8	0.87	0.22	0.32	0.32
dislikeCount																	0.93	0.93	0.3	0.32	0.32
commentCount																		0.96	0.4	0.37	0.38
likeCount																			0.38	0.37	0.42
Artist_Youtube_searches_11m																				0.59	0.62
Artist_Follower																					0.71

Figure 1: Correlogram with significant correlation coefficients at $\alpha = 0.05$



```r
plot(log(x$viewsCount), log(x$Streams))
```



```r
bivariate_df <- select(x, c('Streams', 'viewsCount'))

# install.packages("normwhn.test"")

library("normwhn.test")
```

```
## Warning: package 'normwhn.test' was built under R version 3.5.2
```

```r
normality.test1(bivariate_df)
```

```
## [1] "sk"
## [1] 2.605774 9.546511
## [1] "k"
## [1]  12.86985 127.38076
## [1] "rtb1"
## [1] 2.566691 9.693132
## [1] "b2"
## [1]  12.57581 130.60922
## [1] "z1"
## [1] 16.95863 28.93775
## [1] "z2"
## [1]  -26.91954 -132.85306
## [1] "H0: data do not have skewness"
## [1] "pvalsk"
## [1]  1.661657e-64 4.002662e-184
## [1] "H0: data do not have negative skewness"
## [1] "pskneg"
## [1] 1 1
## [1] "H0: data do not have positive skewness"
## [1] "pskpos"
## [1] 0 0
## [1] "H0: data do not have kurtosis"
## [1] "pvalk"
## [1] 0 0
## [1] "H0: data do not have negative kurtosis"
## [1] "pkneg"
## [1] 6.485139e-160  0.000000e+00
## [1] "H0: data do not have positive kurtosis"
## [1] "pkpos"
## [1] 1 1
## [1] "H0: data are normally distributed"
## [1] "Ep"
##          [,1]
## [1,] 19499.59
## [1] "dof"
## [1] 4
## [1] "sig.Ep"
##      [,1]
## [1,]    0
```

```r
bivariate_df$Streams <- log(bivariate_df$Streams)
bivariate_df$viewsCount <- log(bivariate_df$viewsCount)

normality.test1(bivariate_df)
```

```
## [1] "sk"
## [1] -0.27512260  0.03938024
## [1] "k"
## [1] 2.273702 2.964363
## [1] "rtb1"
## [1] -0.4321406  0.3074930
## [1] "b2"
## [1] 2.532132 3.455444
## [1] "z1"
## [1] -4.588363  3.329138
## [1] "z2"
## [1] -5.999773  1.154060
## [1] "H0: data do not have skewness"
## [1] "pvalsk"
## [1] 4.467361e-06 8.711525e-04
## [1] "H0: data do not have negative skewness"
## [1] "pskneg"
## [1] 2.233680e-06 9.995644e-01
## [1] "H0: data do not have positive skewness"
## [1] "pskpos"
## [1] 0.9999977663 0.0004355762
## [1] "H0: data do not have kurtosis"
## [1] "pvalk"
## [1] 1.975941e-09 2.484754e-01
## [1] "H0: data do not have negative kurtosis"
## [1] "pkneg"
## [1] 9.879705e-10 8.757623e-01
## [1] "H0: data do not have positive kurtosis"
## [1] "pkpos"
## [1] 1.0000000 0.1242377
## [1] "H0: data are normally distributed"
## [1] "Ep"
##          [,1]
## [1,] 69.46536
## [1] "dof"
## [1] 4
## [1] "sig.Ep"
##              [,1]
## [1,] 2.942091e-14
```