STAT 341: Assignment 1 Solutions

QUESTION 1: Evaluating the population mid-range [20 points]

Consider the population $\mathcal{P} = \{y_1, \ldots, y_N\}$. The population mid-range is the midpoint of the range

$$a(\mathcal{P}) = a(y_1, \ldots, y_N) = \frac{y_{(1)} + y_{(N)}}{2}$$

and hence a measure of center. In this question you will investigate several of its properties.

(a) [3 points] Determine whether the mid-range is location invariant, location equivariant, or neither.

Solution:

The population $\mathcal{P}$ ordered from smallest to largest is given by

$$y_{(1)} \leq y_{(2)} \leq \cdots \leq y_{(N)}$$

Note that adding $b \in \mathbb{R}$ to every unit preserves this ordering:

$$y_{(1)} + b \leq y_{(2)} + b \leq \cdots \leq y_{(N)} + b$$

and the minimum and maximum of the translated population are respectively $y_{(1)} + b$ and $y_{(N)} + b$. Thus the mid-range of the translated population is given by:

$$\begin{aligned} a(y_1 + b, \ldots, y_N + b) &= \frac{\left(y_{(1)} + b\right) + \left(y_{(N)} + b\right)}{2} \\ &= \frac{y_{(1)} + y_{(N)} + 2b}{2} \\ &= \frac{y_{(1)} + y_{(N)}}{2} + b \\ &= a(y_1, \ldots, y_N) + b \end{aligned}$$

$\therefore$ the mid-range is location-equivariant.

(b) [3 points] Determine whether the mid-range is scale invariant, scale equivariant, or neither.

Solution:

Similarly, multiplying every unit in $\mathcal{P}$ by $m > 0$ does not effect the ordering:

$$m \times y_{(1)} \leq m \times y_{(2)} \leq \cdots \leq m \times y_{(N)}$$

and so the minimum and maximum of the scaled population are respectively $m \times y_{(1)}$ and $m \times y_{(N)}$. Thus the mid-range of the scaled population is given by:

$$a(m \times y_1, \ldots, m \times y_N) = \frac{m \times y_{(1)} \ + m \times y_{(N)}}{2} = m \times \frac{y_{(1)} + y_{(N)}}{2} = m \times a(y_1, \ldots, y_N)$$

$\therefore$ the mid-range is scale-equivariant.

(c) [3 points] Determine whether the mid-range is location-scale invariant, location-scale equivariant, or neither.

Solution:

And if we multiply every unit in $\mathcal{P}$ by $m > 0$ AND add $b \in \mathbb{R}$ the ordering is still not effected:

$$m \times y_{(1)} + b \leq m \times y_{(2)} + b \leq \cdots \leq m \times y_{(N)} + b$$

and the minimum and maximum of the translated and scaled population are respectively $m \times y_{(1)} + b$ and $m \times y_{(N)} + b$. Thus the mid-range of the translated and scaled population is given by:

$$\begin{aligned} a(m \times y_1 + b, \dots, m \times y_N + b) &= \frac{\big(m \times y_{(1)} + b\big) + \big(m \times y_{(N)} + b\big)}{2} \\ &= \frac{m \times (y_{(1)} + y_{(N)}) + 2b}{2} \\ &= m \times \frac{y_{(1)} + y_{(N)}}{2} + b \\ &= m \times a(y_1, \dots, y_N) + b \end{aligned}$$

$\therefore$ the mid-range is location-scale equivariant.

(d) [3 points] Determine whether the mid-range is replication invariant, replication equivariant, or neither.

Solution:

To investigate replication invariance/equivariance we consider the population

$$\mathcal{P}^k = \{y_1, \dots, y_N, y_1, \dots, y_N, \dots, y_1, \dots, y_N\} \equiv \{x_1, x_2, \dots, x_{Nk}\}$$

where each unit is replicated $k > 1$ times. This replicated population, ordered from smallest to largest, is given by:

$$x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(Nk)}$$

But note that the action of replication does not impact the value of the maximum or the minimum (i.e., $x_{(1)} = y_{(1)}$ and $x_{(Nk)} = y_{(N)}$) so

$$a(\mathcal{P}^k) = \frac{x_{(1)} + x_{(Nk)}}{2} = \frac{y_{(1)} + y_{(N)}}{2} = a(\mathcal{P})$$

$\therefore$ the mid-range is replication-invariant.

(e) [3 points] Derive the sensitivity curve for the mid-range, given a population $\{y_1, y_2, \dots, y_{N-1}\}$.

Solution:

The mid-range for the population $\{y_1, y_2, \dots, y_{N-1}\}$ is

$$a(y_1, \dots, y_{N-1}) = \frac{1}{2}\left[y_{(1)} + y_{(N-1)}\right]$$

The value of $a(y_1, \dots, y_{N-1}, y)$ depends on the value of y as follows:

$$a(y_1, \ldots, y_{N-1}, y) = \begin{cases} \frac{1}{2}\left[y + y_{(N-1)}\right] & \text{if} \quad y < y_{(1)} \\ \frac{1}{2}\left[y_{(1)} + y_{(N-1)}\right] & \text{if} \quad y_{(1)} \leq y \leq y_{(N-1)} \\ \frac{1}{2}\left[y_{(1)} + y\right] & \text{if} \quad y > y_{(N-1)} \end{cases}$$

Then the sensitivity curve is:

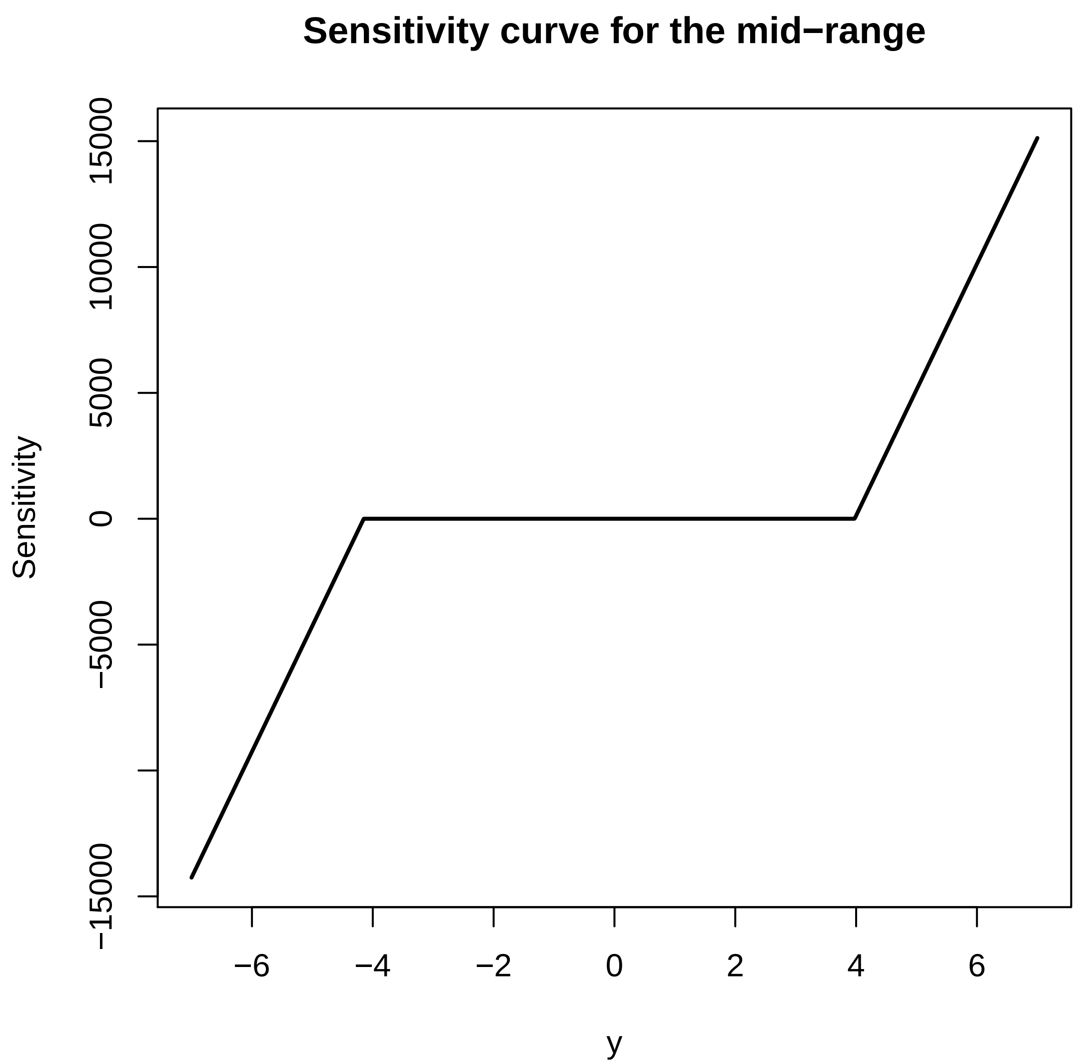
$$SC(y) = \begin{cases} \frac{N}{2}\left[y - y_{(1)}\right] & \text{if} \quad y < y_{(1)} \\ 0 & \text{if} \quad y_{(1)} \leq y \leq y_{(N-1)} \\ \frac{N}{2}\left[y - y_{(N-1)}\right] & \text{if} \quad y > y_{(N-1)} \end{cases}$$

(f) [3 points] For the population below, plot the sensitivity curve from part (e) for $y \in [-7, 7]$. You may find the `sc()` function from class useful.

```
set.seed(341)
pop <- rnorm(10000)
```

Solution:

```
mid.range <- function(pop){ mean(range(pop)) }
sc = function(y.pop, y, attr, ...) {
  N <- length(y.pop) +1
  mapply(function(y) {  N*(attr(c(y,y.pop),...) - attr(y.pop,...))  } ,y   , SIMPLIFY = FALSE )
}
y <- seq(-7, 7, 0.01)
plot(y, sc(y.pop = pop, y = y, attr = mid.range), type="l", lwd = 2,
     main="Sensitivity curve for the mid-range",
     xlab = "y",
     ylab="Sensitivity")
```



(g) [2 points] Given all that you have learned in parts (a) - (f), state one thing that is *good* about thie mid-range attribute and one thing that is *bad* about the mid-range attribute.

Solution:

Good things: It is location-scale equivariant and replication invariant.

Bad things: It is highly sensitive to extreme observations.

QUESTION 2: Write a plot-making function [5 points]

Write a function called `matrix.plot()` that takes in a single input (called `df`), that is an $N \times m$ data frame containing *numeric* data. This function should produce as its output an $m \times m$ matrix of plots where:

- the diagonal plots contain histograms of the columns of `df`
- the upper triangle of plots are scatter plots between all pairs of columns of `df`
- the lower triangle of plots report the correlation coefficients between the pairs of columns of `df`
- all plots should be labelled with the headings provided in `df`

Solution

```r
matrix.plot <- function(df){
  headers <- names(df)
  par(mfrow=c(length(headers), length(headers)), mar=c(1, 4, 4, 1))
  for(i in 1:length(headers)){
    for(j in 1:length(headers)){
      if(i == j){
        hist(df[,i], main = "", xlab = "", ylab = "", col = "gray80")
      }else if(i < j){
        plot(x = df[,j], y = df[,i],
             main = "", xlab = "", ylab = "",
             pch = 16, col = adjustcolor(col = "firebrick", alpha.f = 0.4))
      }else if(i > j){
        plot(0,0, col = "white", main = "", xlab = "", ylab = "", xaxt = "n", yaxt = "n")
        text(x = 0, y = 0, labels = paste(round(cor(df[,i],df[,j]), 4)), col = "firebrick", cex = 2)
      }
      if(i == 1){
        title(main = headers[j], font = 2, )
      }
      if(j == 1){
        title(ylab = headers[i], font.lab = 2)
      }
    }
  }
}
```

QUESTION 3: Spotify Top 30 Analysis [25 points]

Spotify, the popular music streaming service, organizes and classifies songs based on a wide range of properties (variates):

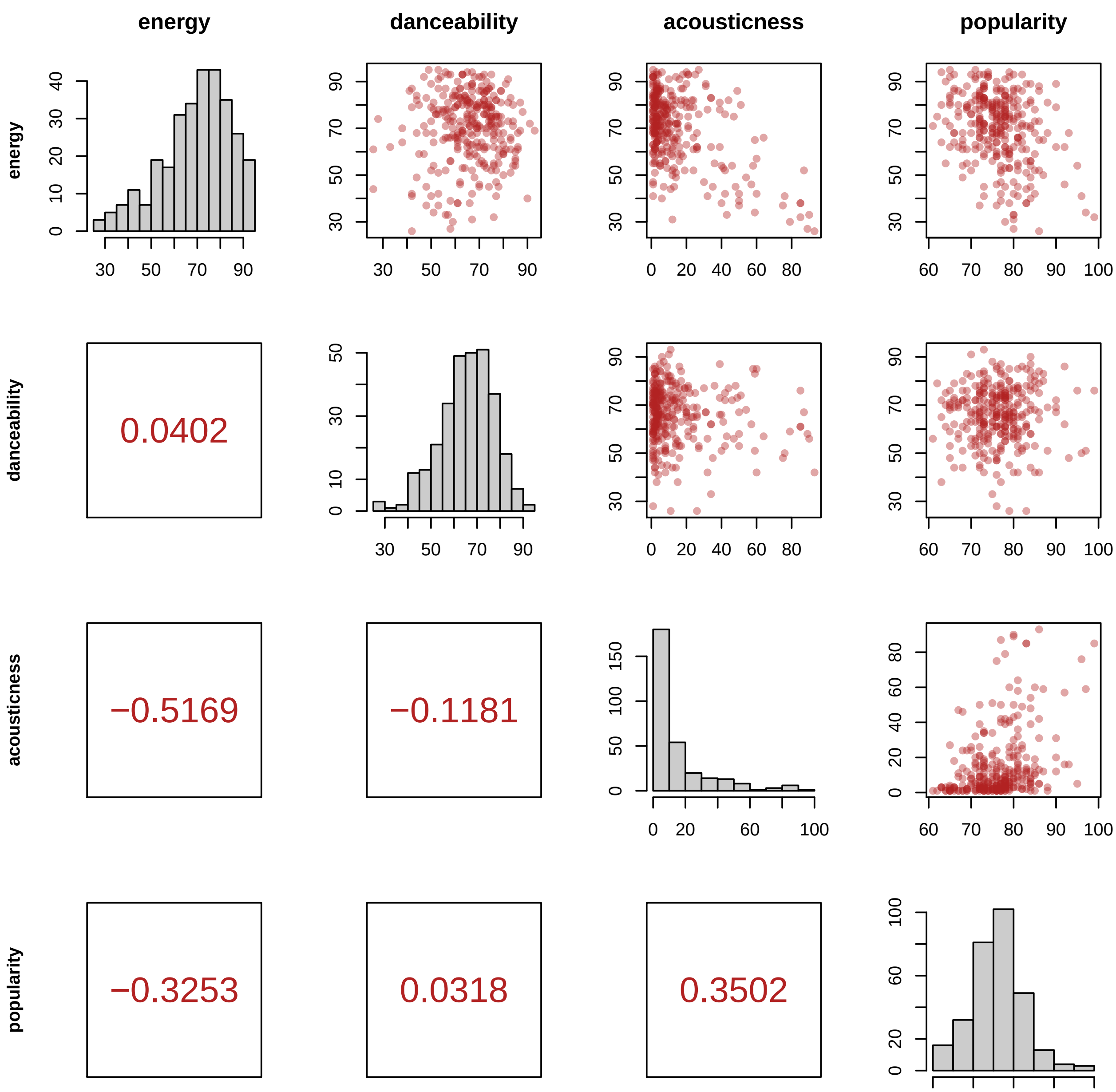
Variate	Description
`genre`	the genre of the track
`year`	the release year of the recording (note that due to vagaries of releases, re-releases, re-issues and general madness, sometimes the release years are not what you'd expect)
`bpm`	beats per minute - the tempo of the song
`energy`	the higher the value the more energetic the song
`danceability`	the higher the value the easier it is to dance to the song
`loudness`	the higher the value the louder the song
`liveness`	the higher the value the more likely the song is a live recording
`valence`	the higher the value the more positive the mood of the song
`duration`	the duration of the song (in seconds)
`acousticness`	the higher the value the more acoustic the song is
`speechiness`	the higher the value the more spoken words the song contains
`popularity`	the higher the value the more popular the song is

Available for us to study is the population of $N = 300$ Billboard Top 30 songs from 2010 - 2019 (inclusive). In addition to the song's title and artist, measurements on each of the 12 variates listed in the table above have also been recorded for each of these songs. This data is available in the `spotify.csv` file.

(a) [2 points] Using the `matrix.plot()` function you developed in Question 2, produce the summary graphic for `energy`, `danceability`, `acousticness`, and `popularity`.

Solution:

```
songs <- read.csv(file = "/Users/nstevens/Dropbox/Teaching/STAT_341/Assignments/Assignment1/spotify.csv
                  header = TRUE)
matrix.plot(df = songs[,c(6,7,12,14)])
```



(b) [3 points] Considering all variates (except `genre` and `year`), which three are most strongly correlated with popularity? For each variate, explain the nature of its linear relationship with popularity.

Solution:

```
cor(songs[,5:14])[1:9,10][order(abs(cor(songs[,5:14])[1:9,10]), decreasing = TRUE)]
```

```
## acousticness       energy     duration          bpm      valence     loudness 
##   0.35020635  -0.32533264  -0.19717674  -0.13737400  -0.12370367  -0.11021924 
##     liveness danceability  speechiness 
##  -0.06759539   0.03179367  -0.01871555
```

Based on the magnitude of their correlation coefficients `acousticness`, `energy`, and `duration` are most strongly correlated with `popularity`. Sonds that are more acoustic, less energetic and shorter tend to be more popular.

(c) [1 point] Using R, determine which are the Top 10 most popular songs.

Solution:

```
songs[order(songs$popularity, decreasing = TRUE),][1:10,1:2]
```

```
##                                                title         artist
## 271                                         Memories       Maroon 5
## 272                                Lose You To Love Me   Selena Gomez
## 273                                 Someone You Loved Lewis Capaldi
## 274                                       Se\xf1orita   Shawn Mendes
## 275                                 How Do You Sleep?      Sam Smith
## 276 South of the Border (feat. Camila Cabello & Cardi B)    Ed Sheeran
## 277                               Trampoline (with ZAYN)          SHAED
## 278                                          Happier     Marshmello
## 279                                      Truth Hurts          Lizzo
## 280           Good as Hell (feat. Ariana Grande) - Remix          Lizzo
```

(d) [2 points] Using R, determine which song is the shortest and which is the longest.

Solution:

```
songs[which.min(songs$duration),][1:2]
```

```
##                                  title artist
## 290 All Around The World (La La La)  R3HAB
```

```
songs[which.max(songs$duration),][1:2]
```

```
##      title     artist
## 43 Monster Kanye West
```

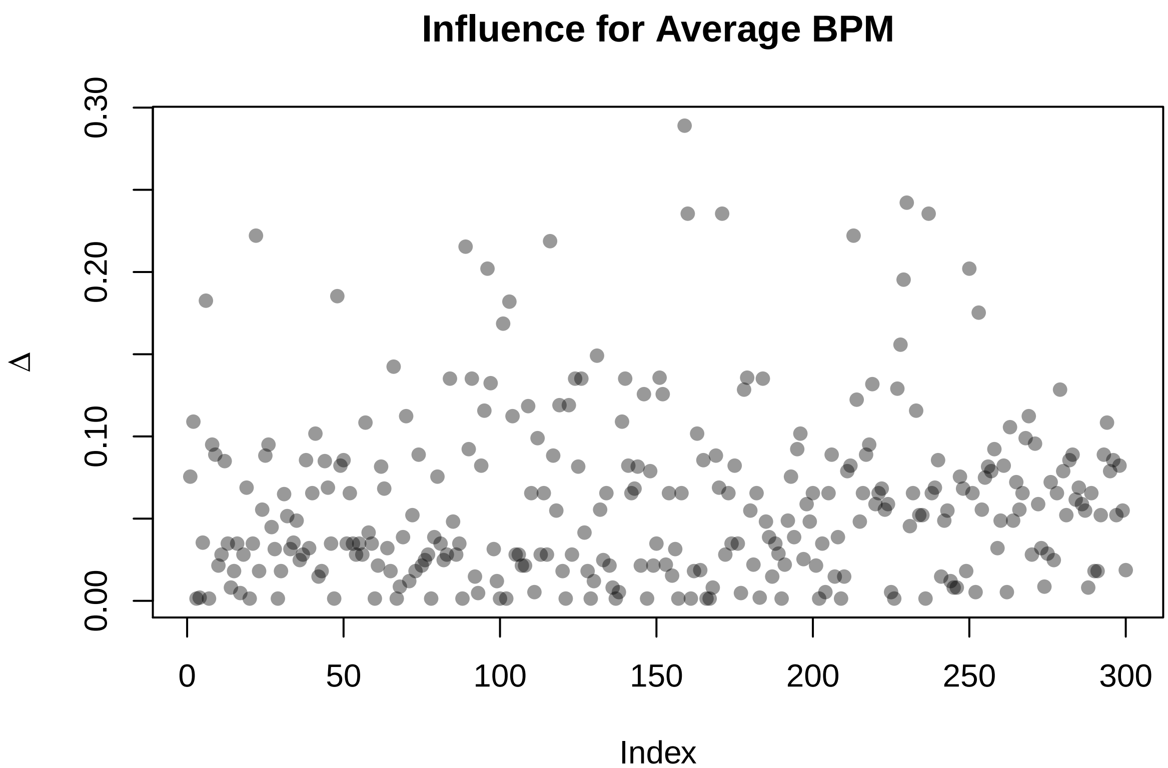
(e) [4 points] Let y denote the beats per minute (`bpm`) of a song, and let $a(\mathcal{P}) = \bar{y}$ be the attribute of interest. Define the influence of song u on $a(\mathcal{P})$ to be:

$$\Delta(a, u) = |a(y_1, \dots, y_{u-1}, y_u, y_{u+1}, \dots, y_N) - a(y_1, \dots, y_{u-1}, y_{u+1}, \dots, y_N)|$$

Construct an influence plot of Δ vs. observation number and identify the song with the largest influence on the average `bpm` attribute. Why is this song in particular more influential than all of the others?

Solution:

```
y <- songs$bpm
delta <- abs((y-mean(y))/(length(y)-1))
par(mfrow = c(1,1))
plot(delta, main = "Influence for Average BPM", xlab = "Index", ylab = bquote(Delta), pch = 16, col = ad
```



The song with the largest influence is:

```
songs[which.max(delta), 1:2]
```

```
##                 title  artist
## 159 FourFiveSeconds Rihanna
```

Note that this is the same song with the largest `bpm` value, which is why it has more influence on the mean than the other songs.

```
songs[which.max(songs$bpm), 1:2]
```

```
##                 title  artist
## 159 FourFiveSeconds Rihanna
```

(f) [3 points] Using R, determine which artists have appeared in the Billboard Top 30 five or more times. For each of these artists state the number of times they have appeared and calculate the average popularity score of their songs.

Solution:

```
top_artists <- names(which(table(songs$artist)>=5))
num_appears <- rep(0, length(top_artists))
artist_popularity <- rep(0, length(top_artists))
```

```r
for(i in 1:length(top_artists)){
  num_appears[i] <- length(songs[songs$artist == top_artists[i],]$popularity)
  artist_popularity[i] <- mean(songs[songs$artist == top_artists[i],]$popularity)
}
data.frame(artist = top_artists, num.hits = num_appears, avg.pop = artist_popularity)
```

```
##              artist num.hits  avg.pop
## 1     Ariana Grande        7 78.28571
## 2        Bruno Mars       11 75.36364
## 3     Calvin Harris       10 78.20000
## 4        Ed Sheeran        9 82.11111
## 5     Justin Bieber        8 77.87500
## 6        Katy Perry       11 69.45455
## 7             Kesha        7 71.85714
## 8         Lady Gaga        7 74.42857
## 9          Maroon 5       11 78.09091
## 10    One Direction        5 76.40000
## 11          Rihanna        8 73.87500
## 12     Shawn Mendes        7 83.28571
## 13     Taylor Swift        5 76.00000
## 14 The Chainsmokers        8 79.37500
## 15       The Weeknd        5 82.60000
```

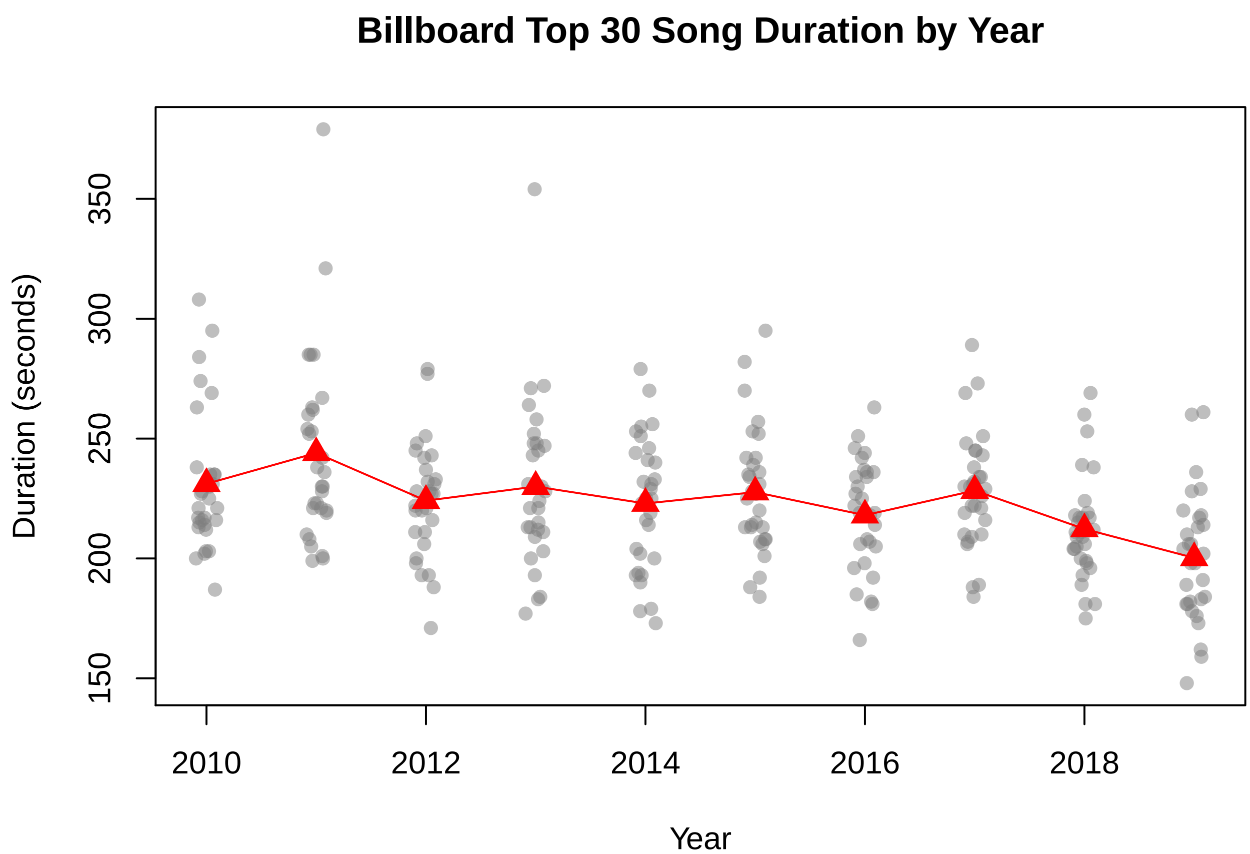
(g) [5 points] Construct the following plot:

- Make a scatter plot of `duration` vs. `year`, but where `year` has been jittered slightly
- Add to this plot 10 red triangles indicating the average song duration in each year.
- Connect these 10 red triangles with red line segments.
- Add appropriate labels and a title to the plot.

Does the duration of popular songs seem to have changed over time? If so, in which direction?

Solution:

```r
plot(x = jitter(songs$year, factor = 0.5), y = songs$duration,
     xlab = "Year", ylab = "Duration (seconds)", main = "Billboard Top 30 Song Duration by Year",
     pch = 16, col = adjustcolor("gray50", alpha.f = 0.5))
points(x = unique(songs$year),
       y = apply(X = matrix(data = songs$duration, nrow = 30, ncol = 10, byrow = FALSE), MARGIN = 2, FUN
       pch = 17, col = "red", cex = 1.5)
lines(x = unique(songs$year),
      y = apply(X = matrix(data = songs$duration, nrow = 30, ncol = 10, byrow = FALSE), MARGIN = 2, FUN
      col = "red")
```

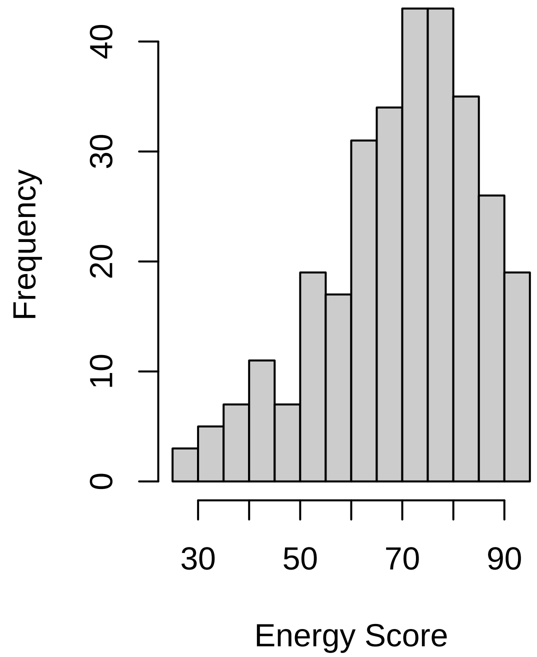


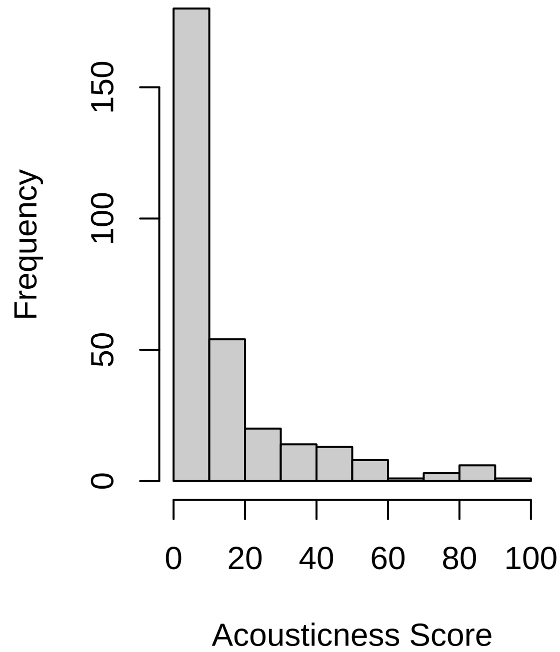
Yes, popular songs appear to be getting shorter over time.

(h) [5 points] Construct a 1×2 plot which contains histograms of `energy` and `acousticness`. Using the `powerfun()` function from class determine a range of powers (values of α) for each variate which make its distribution more symmetric. Plot another 1×2 plot containing histograms of the transformed `energy` and `acousticness` variates using what you feel is the *best* value of α in each case. Make sure all of your plots are appropriately titled and labelled.

Solution:

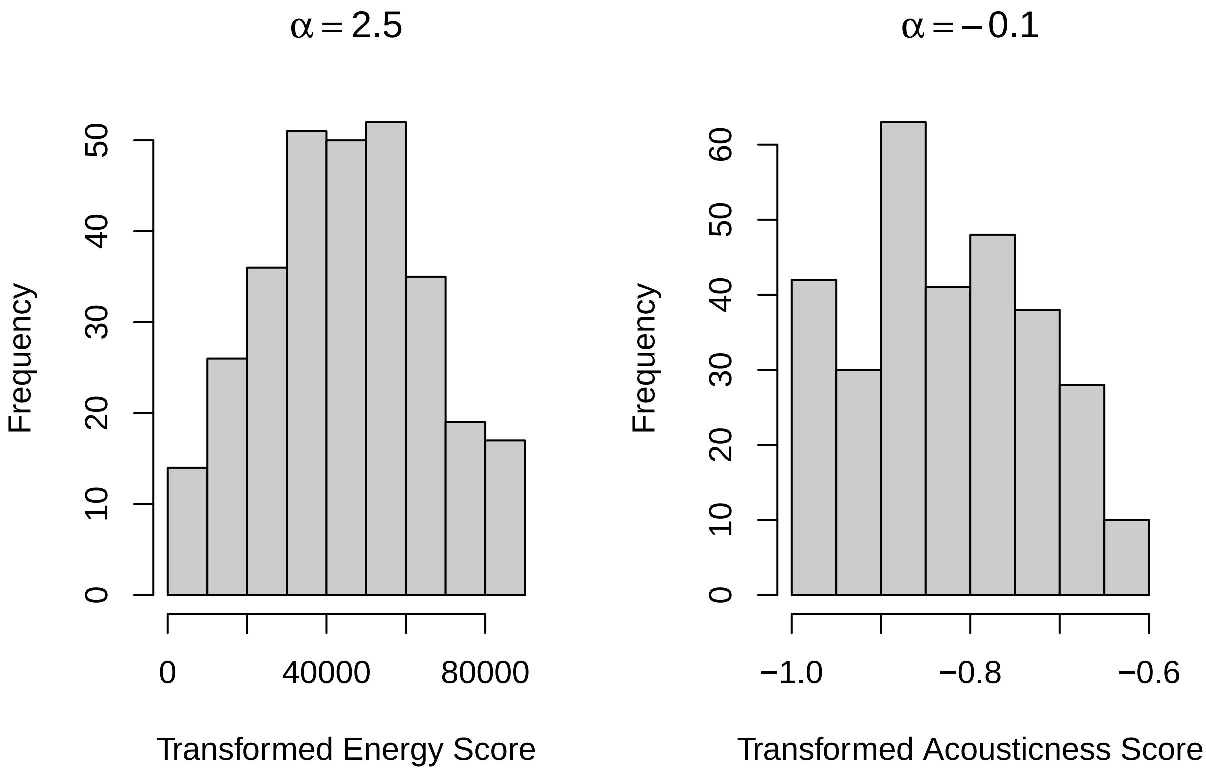
```r
par(mfrow = c(1,2))
hist(songs$energy, xlab = "Energy Score", col = "gray80", main = "")
hist(songs$acousticness, xlab = "Acousticness Score", col = "gray80", main = "")
```





Transformations in the region of $\alpha \in [2, 3]$ for `energy` and $\alpha \in [-0.1, 0]$ for `acousticness` seem to do pretty well. Below are plots with $\alpha_{\text{energy}} = 2.5$ and $\alpha_{\text{acousticness}} = -0.1$

```
powerfun <- function(x, alpha) {
  if(sum(x <= 0) > 0) stop("x must be positive")
  if (alpha == 0)
    log(x)
  else if (alpha > 0) {
    x^alpha
  } else -x^alpha
}
par(mfrow = c(1,2))
hist(powerfun(songs$energy, alpha = 2.5), xlab = "Transformed Energy Score", main = bquote(alpha == 2.5)
hist(powerfun(songs$acousticness, alpha = -0.1), xlab = "Transformed Acousticness Score", main = bquote
```


α = 2.5
Frequency
0
10
20
30
40
50
0
40000
80000
Transformed Energy Score
α = −0.1
Frequency
0
10
20
30
40
50
60
−1.0
−0.8
−0.6
Transformed Acousticness Score