MSCI446 - Assignment 1 - Question 1

H. Gomma, K. Morris, M. Harper

05/02/2021

Question 1: Linear Regression

The first thing we will do is import and read the dataset and display the top 5 rows using the head() function to better understand the starting point in which we are working with.

```
data <- read.csv('spotify_songs.csv')
names(data)
```

```
##  [1] "track_id"                "track_name"
##  [3] "track_artist"            "track_popularity"
##  [5] "track_album_id"          "track_album_name"
##  [7] "track_album_release_date" "playlist_name"
##  [9] "playlist_id"             "playlist_genre"
## [11] "playlist_subgenre"       "danceability"
## [13] "energy"                  "key"
## [15] "loudness"                "mode"
## [17] "speechiness"             "acousticness"
## [19] "instrumentalness"        "liveness"
## [21] "valence"                 "tempo"
## [23] "duration_ms"
```

For this question, we will take the output variable to be **Y = track_popularity (numeric)** with inputs **X:**

- **danceability (numeric)**
- **tempo (numeric)**
- **energy (numeric)**
- **playlist_genre (categorical)**

1.1 Basic Insights

Determining the min/max values and median of the output variable $\mathbf{Y}$ = **'track_popularity'**

```
cols <- c('track_popularity','danceability', 'energy', 'tempo', 'playlist_genre')
data.short <- data[,cols]

cat('Min Track Popularity: ', min(data.short$track_popularity), '\n')
```

```
## Min Track Popularity:  0
```

```
cat('Min Track Popularity: ',max(data.short$track_popularity), '\n')
```

```
## Min Track Popularity:  100
```

```
cat('Median track_popularity: ', median(data.short$track_popularity))
```

```
## Median track_popularity:  45
```

Summarizing the dataset characteristics based on the various genre classes:

```
library('tidyverse')
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.0.4     v dplyr   1.0.2
## v tidyr   1.1.2     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.0

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
data.short %>%
  group_by(playlist_genre) %>%
  summarize(Avg_Popularity = mean(track_popularity),
            Median_Popularity = median(track_popularity))
```

```
## `summarise()` ungrouping output (override with `.groups` argument)

## # A tibble: 6 x 3
##   playlist_genre Avg_Popularity Median_Popularity
##   <chr>                   <dbl>             <dbl>
## 1 edm                      34.8                36
## 2 latin                    47.0                50
## 3 pop                      47.7                52
## 4 r&b                      41.2                44
## 5 rap                      43.2                47
## 6 rock                     41.7                46
```

1.2: Visualizing the Data

Selected Numeric Input Variables:

- energy(numeric/continuous)
- tempo (numeric/continuous)
- duration_ms(numeric/continuous)

```
library('gridExtra')
```

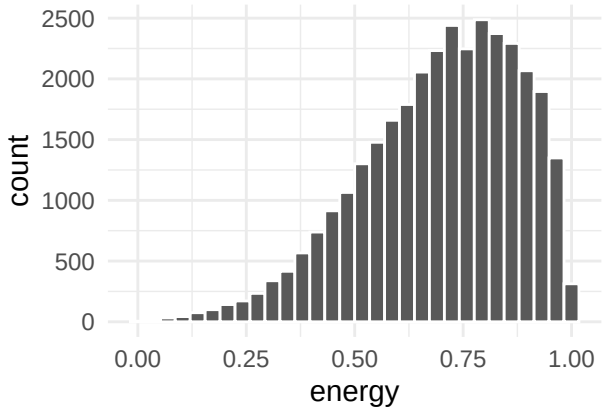
```
##
## Attaching package: 'gridExtra'
```

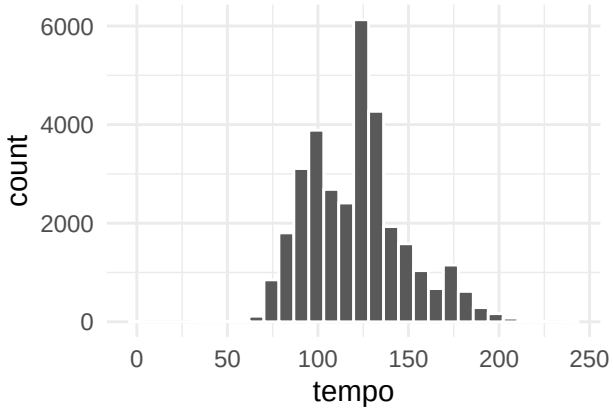
```
## The following object is masked from 'package:dplyr':
##
##     combine
```

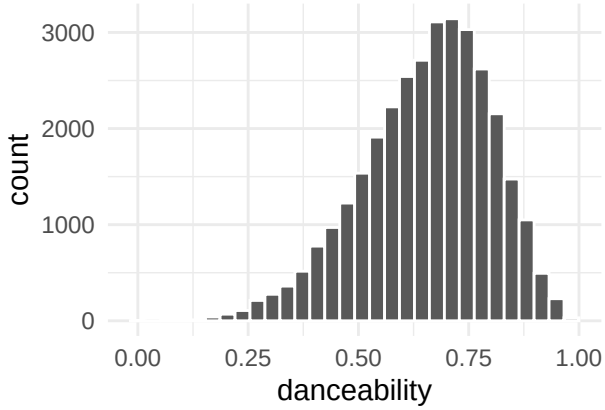
```
theme_set(theme_minimal())

g1 <- ggplot(data.short, aes(x=energy)) +
  geom_histogram(colour='white')
g2 <- ggplot(data.short, aes(x=tempo)) +
  geom_histogram(colour='white')
g3 <- ggplot(data.short, aes(x=danceability)) +
  geom_histogram(colour='white')

grid.arrange(g1,g2,g3, ncol=2)
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



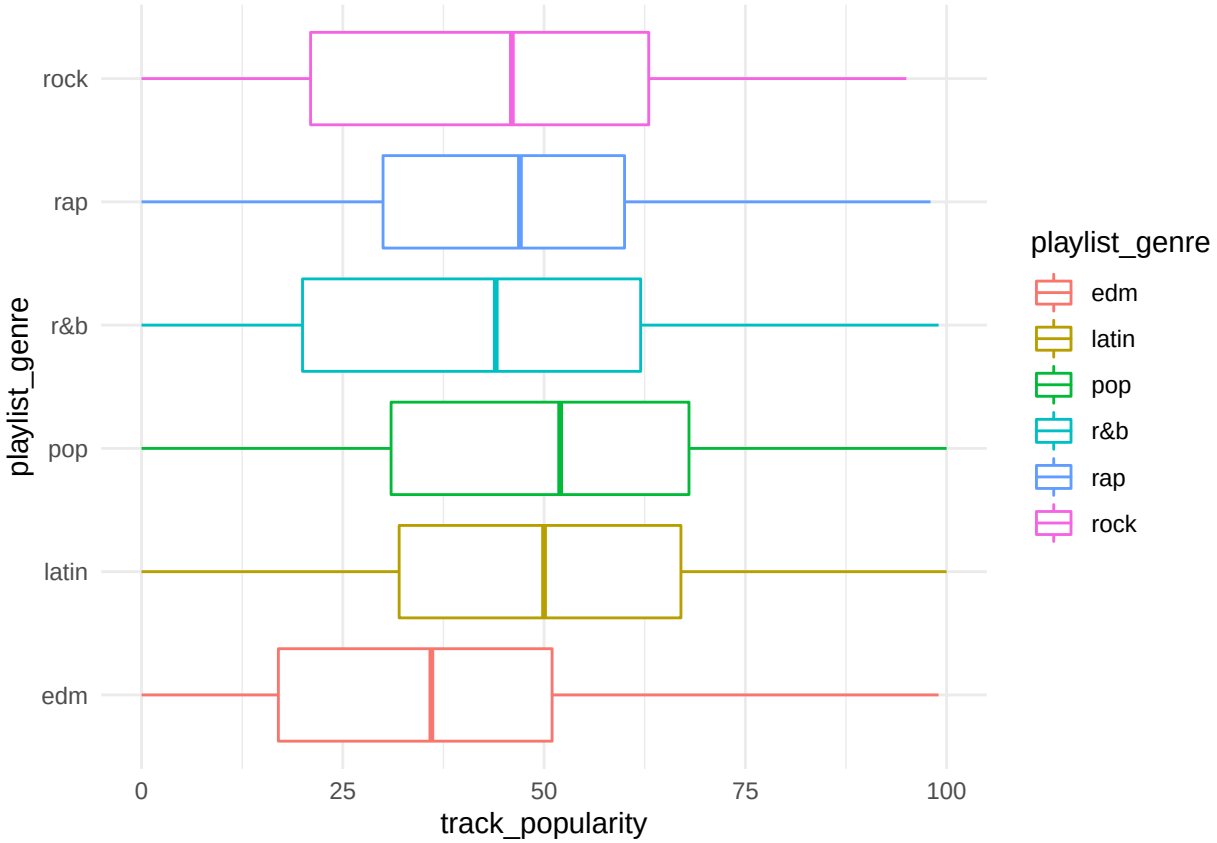




From the plots above, it appears as though the *energy* and *danceability* distributions are left skewed, *energy* being slightly more so than *danceability*. All of the above distributions appear to be normal in that they all follow a *bell curve* shape indicating that they are roughly centered around their data means.

We will now group each of the tracks in the dataset by it's *playlist_genre* categorical value, and create a boxplot to show the inter-quartile range (IQR) and means of the *track_popularity* for each genre.

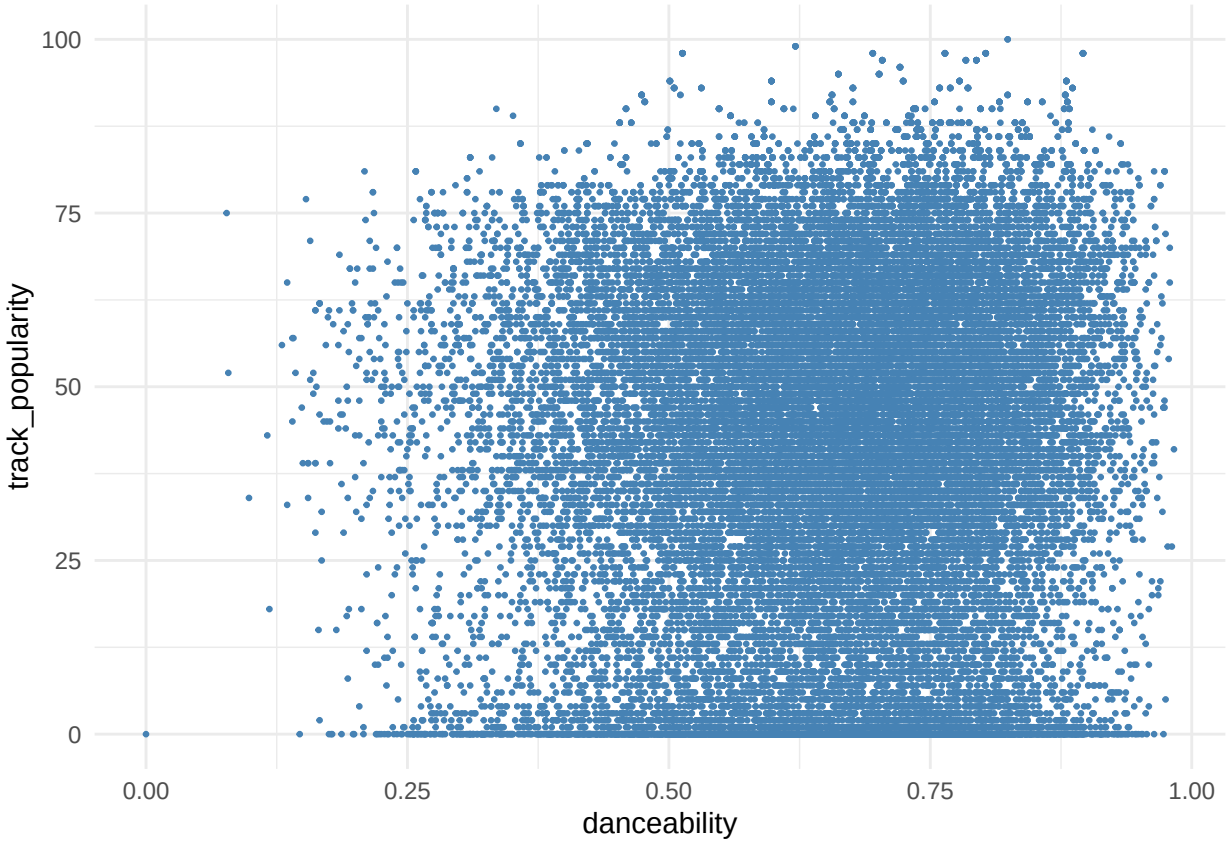
```
data.short$playlist_genre <- as.factor(data.short$playlist_genre)

ggplot(data = data.short)+
  geom_boxplot(aes(x=track_popularity, y=playlist_genre, colour=playlist_genre))
```



At first glance, it is observed that the **pop** genre has the highest mean track popularity rating of approximately 52-55. Conversely, the **edm** genre has the with the lowest mean track popularity rating of approximately 30-33.

We will now investigate the relationship between one possible numeric output variable, $\mathbf{Y} =$ **track_popularity**, and one of the chosen numeric input variables $\mathbf{X} =$ **danceability**

```
theme_set(theme_minimal())
ggplot(data=data.short)+
  geom_point(aes(x=danceability, y=track_popularity), size=0.5,
             colour='steelblue')
```



There does not seem to be a noticeable relationship (i.e. linear) between the data. With the quantity of data it is difficult to make out any visible relationships as the scatter plot appears as a large blob with multiple outputs for every input. There may not be any relationship between *danceability* and *track_popularity*. In the next part of the assignment, we will quantitatively assess the strength of the relationship between the two variables listed above.

1.3 Regression Using Entire Dataset

1. Using one input and one numeric output

Input Variable: X = danceability **Output Variable:** Y = track_popularity

Simple linear reqression will be used. Because there is one input and one output, the regression line will follow the form:

$$Y_i \sim \beta_0 + \beta_1 X_i$$

Where Y_i is the dependent / output variable (track_popularity), and X_i is the independent variable (danceability). The Code below will fit the line above and approximate values for β_0 and β_1.

```
fit1 = lm(data = data.short, track_popularity ~ danceability)
coef(fit1)
```

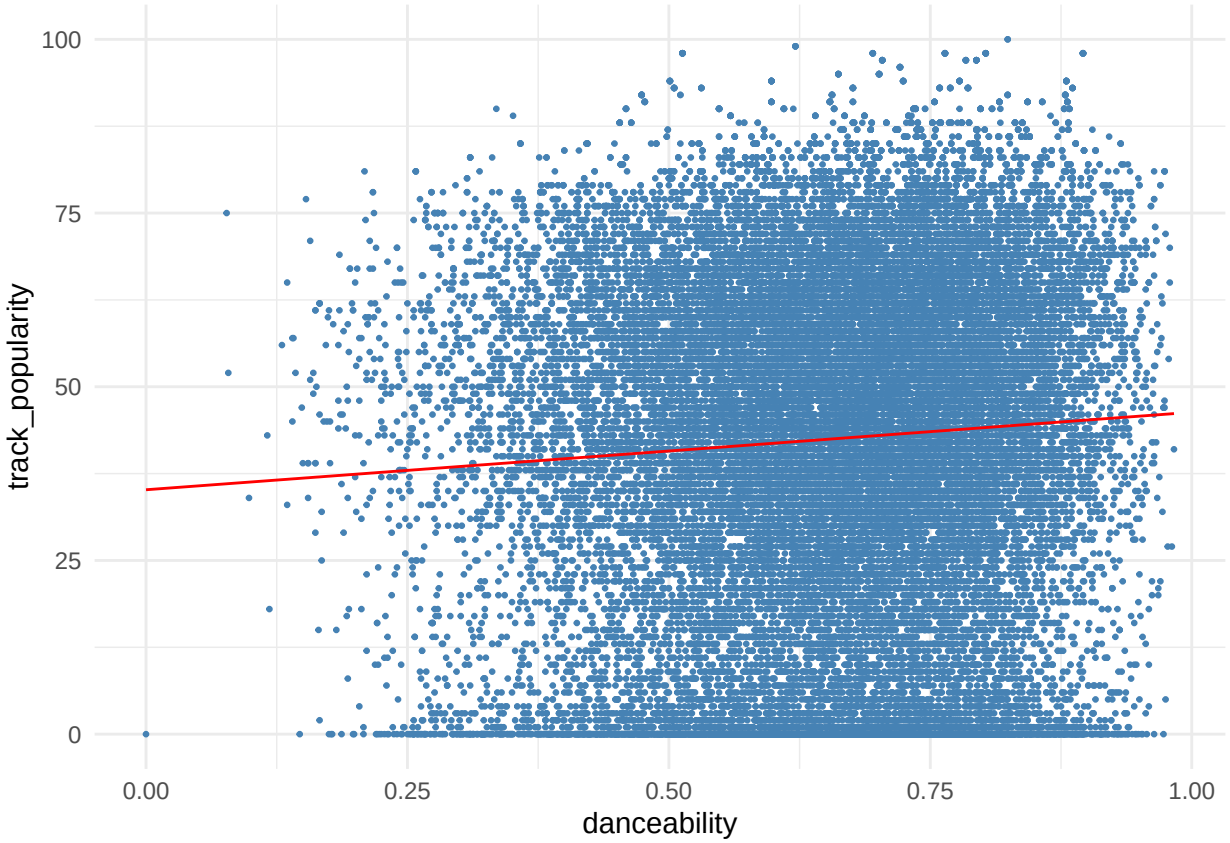
```
##  (Intercept) danceability
##     35.17569     11.14972
```

From the model summary of **fit1** above, the intercept, β_0, is approximately 35.18, and the coeffienct β_1 is approximated to be 11.15. Therefore the linear equation is predicted to be:

$$Y_i \sim 35.18 + 11.15 X_i$$

Plotting the data and the predicted linear model:

```
pred1 = predict(fit1)

ggplot(data=data.short, aes(x=danceability))+
  geom_point(aes(y=track_popularity), size=0.5, colour='steelblue')+
  geom_line(aes(y=pred1), colour='red', size=0.5)
```


100
75
50
25
0
track_popularity
0.00
0.25
0.50
0.75
1.00
danceability

```
summary(fit1)
```

```
##
## Call:
## lm(formula = track_popularity ~ danceability, data = data.short)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -46.024 -18.415   2.934  19.480  57.105
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   35.1757     0.6361   55.30   <2e-16 ***
## danceability  11.1497     0.9484   11.76   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.93 on 32831 degrees of freedom
## Multiple R-squared:  0.004192,   Adjusted R-squared:  0.004162
## F-statistic: 138.2 on 1 and 32831 DF,  p-value: < 2.2e-16
```

The model summary *fit1* suggest that it is *statistically significant*. This is observed by *p-value* that is basically zero, indicating that the null hypothesis, intercept only model, $Y_0 : \beta_0 = 0$, can be confidently rejected. However, the *p-value's* for the slope and intercepts are insignificant and suggest that the probability that the coefficient of the *danceability* term is zero is also negligible. Therefore *danceability* plays an important role in this model.

For this model, $R^2 = 0.0042$, which is significantly low. R^2 is known as the *coefficient of determination* and is an indication of how useful the predictor variable, in this case *danceability*, is at predicting the output variable, *track_popularity*. Therefore, for this model, with significantly low R^2 indicates that our input variable is not good at explaining the output variable.

Root Mean Squared Error (RMSE) of fit1

```
rmse1 <- sigma(fit1)
rmse1
```

```
## [1] 24.93203
```

2. Using all numeric inputs (danceability, energy, tempo)

Fitting the linear model to three numeric input values will yield the form: $Y_i \sim \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_3$ where $X_1 = danceability_i$, $X_2 = energy_i$, and $X_3 = tempo_i$

```
fit2 = lm(data = data.short, track_popularity ~ danceability + energy + tempo)
coef(fit2)
```

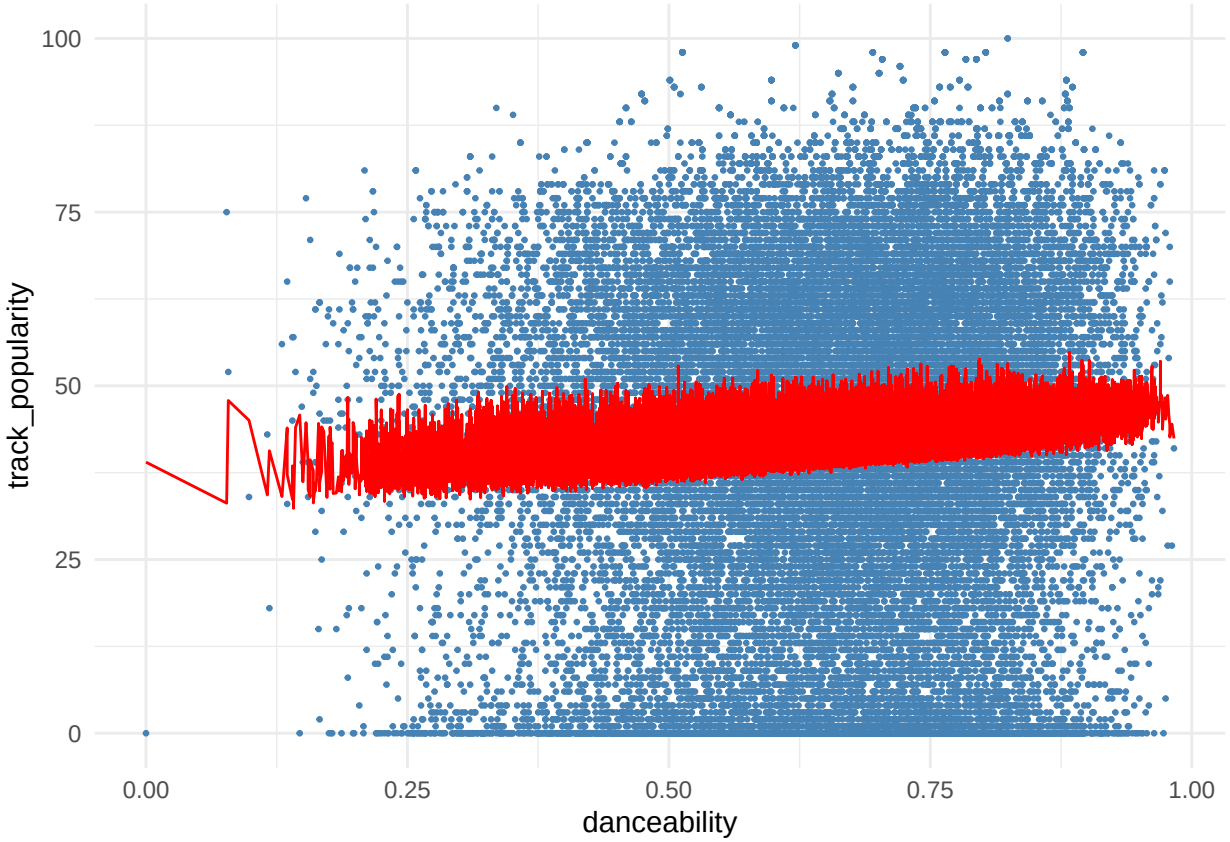
```
##  (Intercept) danceability       energy        tempo
##  43.68231834  10.24673920 -14.80969134   0.02011088
```

From **fit2**, the linear model is represented as:

$$Y_i \sim 43.68 + 10.25 danceability_i - 14.81 energy_i + 0.02 tempo_i$$

Plotting the data and the predicted linear model:

```
pred2 = predict(fit2)

ggplot(data=data.short, aes(x=danceability))+
  geom_point(aes(y=track_popularity), size=0.5, colour='steelblue')+
  geom_line(aes(y=pred2), colour='red', size=0.5)
```



```
summary(fit2)
```

```
##
## Call:
## lm(formula = track_popularity ~ danceability + energy + tempo,
##     data = data.short)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.799 -18.338   2.803  19.408  57.410
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   43.682318   1.089166   40.11  < 2e-16 ***
## danceability  10.246739   0.961183   10.66  < 2e-16 ***
## energy       -14.809691   0.766336  -19.32  < 2e-16 ***
## tempo          0.020111   0.005223    3.85 0.000118 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.79 on 32829 degrees of freedom
## Multiple R-squared:  0.01544,    Adjusted R-squared:  0.01535
## F-statistic: 171.6 on 3 and 32829 DF,  p-value: < 2.2e-16
```

The model summary of *fit2* suggest that it is *statistically significant*. This is observed by *p-value* that is basically zero, indicating that the null hypothesis, intercept only model, $Y_0 : \beta_0 = 0$, can be confidently rejected. Similar to the previous model, the *p_value's* for danceability and energy suggest they are important to the model. Tempo is not as impactful for this model as the other two variables, although it's p-value is still negligibly small suggesting that it does contribute to the models accuracy.

With model *fit2*, the R^2 value increase from 0.0042 to 0.015 which is still significantly low. This suggests that increasing the number of input variables has had a fairly significant effect on predicting the output variable *track_popularity*, when compared to the single input variable model, *fit1*. However, here the input variables themselves are still poor predictor variables nonetheless.

Root Mean Squared Error (RMSE) of fit2

```
rmse2 <- sigma(fit2)
rmse2
```

```
## [1] 24.79162
```

3. Adding and interaction between a categorical input (playlist_genre) and one numeric input (energy)

```
fit3 = lm(data = data.short, track_popularity ~ danceability + energy + tempo + playlist_genre + energy
coef(fit3)
```

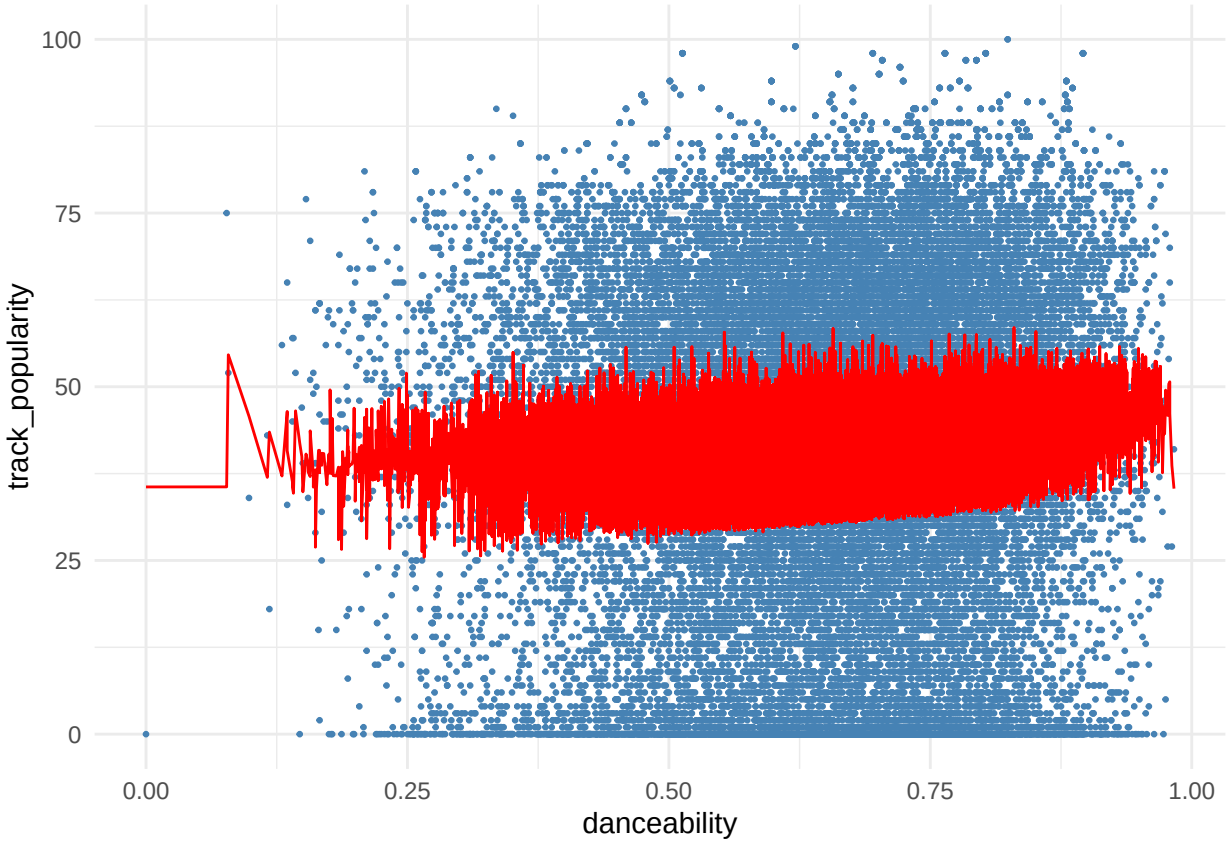
```
##                (Intercept)               danceability
##                43.62369435                11.19568119
##                     energy                      tempo
##               -23.77510439                 0.02349685
##        playlist_genrelatin          playlist_genrepop
##                 6.12655358                -0.28618218
##          playlist_genrer&b          playlist_genrerap
##                -4.99450176                -1.94192699
##         playlist_genrerock energy:playlist_genrelatin
##                -6.06643664                 4.72043509
##   energy:playlist_genrepop   energy:playlist_genrer&b
##                15.80517254                10.92657739
##   energy:playlist_genrerap  energy:playlist_genrerock
##                 9.41572073                17.50658860
```

From **fit3**, the linear model is represented as:

$$Y_i \sim 56.14 + 11.93 danceability_i - 35.12 energy_i + 0.02 tempo_i - 3.74 genre_i + 5.73 energy : genre_i$$

Plotting the data and the predicted linear model:

```
pred3 = predict(fit3)

ggplot(data=data.short, aes(x=danceability))+
  geom_point(aes(y=track_popularity), size=0.5, colour='steelblue')+
  geom_line(aes(y=pred3), colour='red', size=0.5)
```


100
75
50
25
0
track_popularity
0.00
0.25
0.50
0.75
1.00
danceability

```
summary(fit3)
```

```
##
## Call:
## lm(formula = track_popularity ~ danceability + energy + tempo +
##     playlist_genre + energy:playlist_genre, data = data.short)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -56.01 -17.76   3.21  19.13  63.54
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  43.623694   2.118236  20.594  < 2e-16 ***
## danceability                 11.195681   1.058416  10.578  < 2e-16 ***
## energy                      -23.775104   2.261476 -10.513  < 2e-16 ***
## tempo                         0.023497   0.005176   4.539 5.66e-06 ***
## playlist_genrelatin           6.126554   2.451720   2.499 0.012463 *
## playlist_genrepop            -0.286182   2.306087  -0.124 0.901238
## playlist_genrer&b            -4.994502   2.169831  -2.302 0.021353 *
## playlist_genrerap            -1.941927   2.235473  -0.869 0.385025
## playlist_genrerock           -6.066437   2.288900  -2.650 0.008044 **
## energy:playlist_genrelatin    4.720435   3.184223   1.482 0.138232
## energy:playlist_genrepop     15.805173   2.967662   5.326 1.01e-07 ***
## energy:playlist_genrer&b     10.926577   2.928571   3.731 0.000191 ***
## energy:playlist_genrerap      9.415721   2.944687   3.198 0.001387 **
## energy:playlist_genrerock    17.506589   2.877327   6.084 1.18e-09 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.44 on 32819 degrees of freedom
## Multiple R-squared:  0.04337,    Adjusted R-squared:  0.04299
## F-statistic: 114.4 on 13 and 32819 DF,  p-value: < 2.2e-16
```

The model summary *fit3* suggest that it is *statistically significant*. This is observed by *p-value* that is basically zero, indicating that the null hypothesis, intercept only model, $Y_0 : \beta_0 = 0$, can be confidently rejected. We can see from the p-value column that there is a statistically significant relationship between the tracks energy rating and its genre. With the exception of *energy:playlist_genrelatin*, with a p-value of approximately 0.138. This suggests that there is a 13.8% probability that the coefficient for this relationship is the null hypothesis, equal to 0.

We notice that adding the interaction between the *playlist_genre* and *energy*, the R^2 value increases from 0.015 to 0.021. This small value suggests that the predictor variables and the interaction are still poor predictors of the output variable *track_popularity*. However, The increase from the previous models *fit1* and *fit2* indicate that the combination of additional variables and variable interaction has contributed an improvement to the existing model.

Root Mean Squared Error (RMSE) of fit3

```
rmse3 <- sigma(fit3)
rmse3
```

```
## [1] 24.44119
```

4. Comparing fit1, fit2 and fit3

```
models <- c('fit1', 'fit2', 'fit3')
rmse <- c(rmse1, rmse2, rmse3)
r2 <- c(0.0042, 0.015, 0.021)

df <- data.frame(models, rmse, r2)
df
```

```
##   models     rmse     r2
## 1   fit1 24.93203 0.0042
## 2   fit2 24.79162 0.0150
## 3   fit3 24.44119 0.0210
```

From the above dataframe, the rmse value decreases and the R^2 value increases as we move from *fit1* to *fit3*. This would suggest that although all three models are poor at predicting the output variable *track_popularity*, **fit3** is the best of the three. Following this logic, fit2 would be the second best model and fit1 would be the worst of the three.

1.4 Repeat with test/train Datasets

The first step is to split the data in two. We will use a 20/80 test-train ratio as indicated in the assingment. The code below is mostly taken from week 3 tutorial.

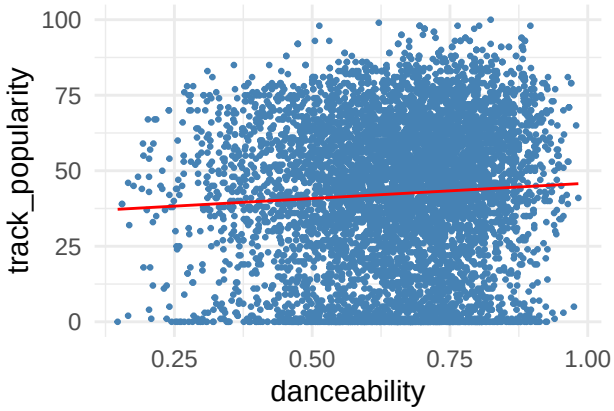
```
set.seed(156)
train_size <- floor(0.8 * nrow(data.short))
train_inds <- sample(1:nrow(data.short), size=train_size)
test_inds <- setdiff(1:nrow(data.short), train_inds)

train <- data.short[train_inds, ]
test <- data.short[test_inds, ]

cat('train size: ', nrow(train), 'test size: ', nrow(test))
```

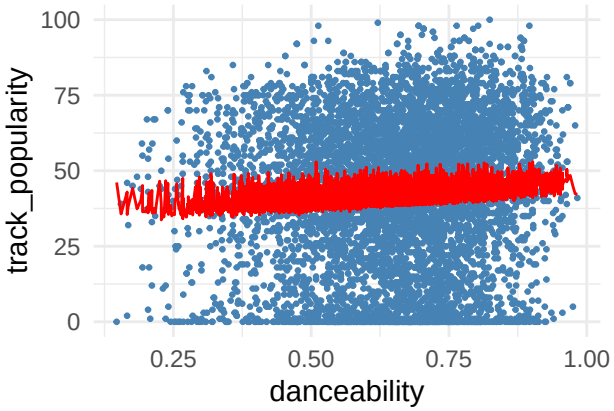
```
## train size:  26266 test size:  6567
```

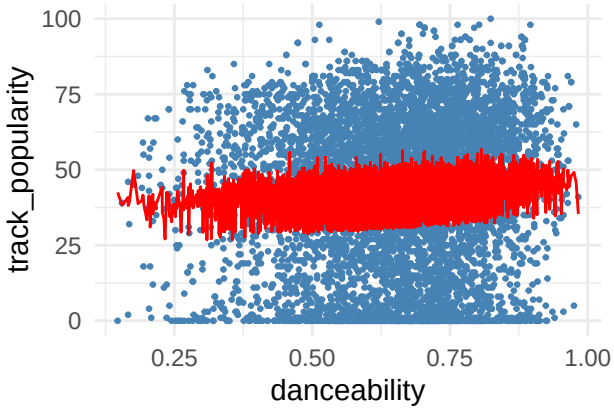
Below we will determine 3 models that are identical to the ones determined in 1.3. The models will be fitted using the training data set and tested on the test data set.

- fit4: track_popularity vs. danceability
- fit5: tracK_popularity vs. danceability, energy, tempo
- fit6: track_popularity vs. *danceability, energy, tempo* & an interaction between *energy* and *playlist_genre*

```
library('gridExtra')
theme_set(theme_minimal())

#Fit the models using the training data set
fit4 <- lm(data=train, track_popularity ~ danceability)
fit5 <- lm(data=train, track_popularity ~ danceability + energy + tempo)
fit6 <- lm(data=train, track_popularity ~ danceability + energy + tempo
           + playlist_genre + energy:playlist_genre)

#Use the models above to predict the track_popularity from the test set
pred4 <- predict(fit4, newdata = test)
pred5 <- predict(fit5, newdata = test)
pred6 <- predict(fit6, newdata = test)

#Create Plots and Display the Models
g4<-ggplot(test, aes(x=danceability)) +
      geom_point(aes(y=track_popularity), colour='steelblue', size=0.5)+
      geom_line(aes(y=pred4), colour='red', size=0.5)
g5<-ggplot(test, aes(x=danceability)) +
      geom_point(aes(y=track_popularity), colour='steelblue', size=0.5)+
      geom_line(aes(y=pred5), colour='red', size=0.5)
g6<-ggplot(test, aes(x=danceability)) +
      geom_point(aes(y=track_popularity), colour='steelblue', size=0.5)+
      geom_line(aes(y=pred6), colour='red', size=0.5)


grid.arrange(g4, g5, g6, ncol=2)
```


track_popularity
100
75
50
25
0
0.25
0.50
0.75
1.00
danceability


track_popularity
100
75
50
25
0
0.25
0.50
0.75
1.00
danceability


track_popularity
100
75
50
25
0
0.25
0.50
0.75
1.00
danceability

Summary of Model Fit4:

```
summary(fit4)
```

```
##
## Call:
## lm(formula = track_popularity ~ danceability, data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -45.613 -18.460   2.906  19.538  57.050
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   35.7503     0.7109  50.286   <2e-16 ***
## danceability  10.1368     1.0600   9.563   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.95 on 26264 degrees of freedom
## Multiple R-squared:  0.00347,    Adjusted R-squared:  0.003432
## F-statistic: 91.46 on 1 and 26264 DF,  p-value: < 2.2e-16
```

```
rmse4 <- sigma(fit4)
```

Fit4 shows a p-value of insignificance and therefore the null hypotheses can be rejected, suggesting that this model is statistically significant. However, the R^2 value is equal to 0.0034 which is significantly low. This would suggest that the predictor variable, *danceability* is not great a predicting the output variable *track_popularity*.

Summary of Model Fit5:

```
summary(fit5)
```

```
##
## Call:
## lm(formula = track_popularity ~ danceability + energy + tempo,
##     data = train)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -54.353 -18.396   2.767  19.495  57.179
##
## Coefficients:
##                Estimate Std. Error t value Pr(>|t|)
## (Intercept)   43.437447   1.218442  35.650  < 2e-16 ***
## danceability   9.445581   1.074677   8.789  < 2e-16 ***
## energy       -14.405809   0.858648 -16.777  < 2e-16 ***
## tempo          0.023417   0.005867   3.991 6.59e-05 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.82 on 26262 degrees of freedom
## Multiple R-squared:  0.01414,    Adjusted R-squared:  0.01403
## F-statistic: 125.6 on 3 and 26262 DF,  p-value: < 2.2e-16
```

```
rmse5 <- sigma(fit5)
```

Fit5 also shows an insignificant p-value and therefore, like *Fit4*, suggests that the model is statistically significant. For this model that uses three numeric inputs, *danceability*, *energy* and *tempo*, the R^2 value is approximately 0.0141. This is a significant improvement from the model *Fit4*, however still suggests that the use of these three variables is not a good predictor of *track_popularity*.

Summary of Model Fit6:

```
summary(fit6)
```

```
##
## Call:
## lm(formula = track_popularity ~ danceability + energy + tempo +
##     playlist_genre + energy:playlist_genre, data = train)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -55.89 -17.85   3.22  19.15  61.92
##
## Coefficients:
##                               Estimate Std. Error t value Pr(>|t|)
## (Intercept)                  42.693819   2.358407  18.103  < 2e-16 ***
## danceability                 10.631547   1.182237   8.993  < 2e-16 ***
## energy                      -22.493421   2.524504  -8.910  < 2e-16 ***
## tempo                         0.026136   0.005814   4.496 6.97e-06 ***
## playlist_genrelatin           7.035415   2.722977   2.584  0.00978 **
## playlist_genrepop             0.954658   2.580780   0.370  0.71145
## playlist_genrer&b            -3.635743   2.420501  -1.502  0.13309
## playlist_genrerap            -1.537083   2.495152  -0.616  0.53788
## playlist_genrerock           -5.912339   2.552226  -2.317  0.02054 *
## energy:playlist_genrelatin    3.316600   3.541136   0.937  0.34898
## energy:playlist_genrepop     14.132251   3.325714   4.249 2.15e-05 ***
## energy:playlist_genrer&b      8.595878   3.270595   2.628  0.00859 **
## energy:playlist_genrerap      8.820983   3.290697   2.681  0.00735 **
## energy:playlist_genrerock    17.194154   3.212274   5.353 8.74e-08 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 24.47 on 26252 degrees of freedom
## Multiple R-squared:  0.04172,    Adjusted R-squared:  0.04125
## F-statistic: 87.93 on 13 and 26252 DF,  p-value: < 2.2e-16
```

```
rmse6 <- sigma(fit6)
```

The last model tested, *Fit6* uses the same 3 input variables as *Fit5*, and includes an interaction between a fourth categorical input, *playlist_genre*. This model improves the R^2 value compared to the previous models, however is still negligible and suggests that this combination of inputs still has a poor effect on predicting *track_popularity*. However, the model is still statistically significant and that the null, intercept only hypothesis can be rejected as evidence of the negligibly small p-value.

Model Comparison (Fit4, Fit5, Fit6)

```
models <- c('fit4', 'fit5', 'fit6')
rmse <- c(rmse4, rmse5, rmse6)
r2 <- c(0.0034, 0.0141, 0.0195)

df <- data.frame(models, rmse, r2)
df
```

```
##   models     rmse     r2
## 1   fit4 24.95203 0.0034
## 2   fit5 24.81903 0.0141
## 3   fit6 24.47402 0.0195
```

The dataframe above shows essentially identical RMSE values for the three models, however there are small improvements from fit4 through to fit6. The R^2 value also follows this same trend. This result would indicate that fit6 is the best of the three at predicting the output variable *track_popularity*. Followed by fit5 and finally fit4. This ranking order matches that of *question 1.3*.