# Project Draft

Dancing Queens: Karianna Klassen, Madeline Waterfield, Mark Yukelis, Shrikrishna Sriram

```
library(tidyverse)
```

```
## Warning in system("timedatectl", intern = TRUE): running command 'timedatectl'
## had status 1
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v purrr   0.3.4
## v tibble  3.1.6     v dplyr   1.0.7
## v tidyr   1.1.4     v stringr 1.4.0
## v readr   2.1.1     v forcats 0.5.1
```

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

```
library(tidytuesdayR)
library(tidymodels)
```

```
## Registered S3 method overwritten by 'tune':
##   method                   from
##   required_pkgs.model_spec parsnip
```

```
## -- Attaching packages --------------------------------------- tidymodels 0.1.4 --
```

```
## v broom        0.7.10     v rsample      0.1.1
## v dials        0.0.10     v tune         0.1.6
## v infer        1.0.0      v workflows    0.2.4
## v modeldata    0.1.1      v workflowsets 0.1.0
## v parsnip      0.1.7      v yardstick    0.0.9
## v recipes      0.1.17
```

```
## -- Conflicts ------------------------------------------ tidymodels_conflicts() --
## x scales::discard() masks purrr::discard()
## x dplyr::filter()   masks stats::filter()
## x recipes::fixed()  masks stringr::fixed()
## x dplyr::lag()      masks stats::lag()
## x yardstick::spec() masks readr::spec()
## x recipes::step()   masks stats::step()
## * Search for functions across packages at https://www.tidymodels.org/find/
```

```
library(car)
```

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

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

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

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

```
library(plot3D)
```

```
## Warning in fun(libname, pkgname): couldn't connect to display ":0"
```

```
spotify_data <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/da
```

```
## Rows: 32833 Columns: 23
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr (10): track_id, track_name, track_artist, track_album_id, track_album_na...
## dbl (13): track_popularity, danceability, energy, key, loudness, mode, speec...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
spotify_data_clean <- spotify_data %>%
  select(-playlist_id, -playlist_name, -playlist_genre, -playlist_subgenre) %>%
  distinct()
```

## Introduction and Data

In the following project, we will investigate a dataset of 28, 356 Spotify songs in order to determine whether songs with certain characteristics are more likely to be popular than others. Specifically, we will be looking at two characteristics drawn from Spotify's Web API: tempo and duration. Additionally, we will be also be investigating whether these characteristics are independent from each other.

Research Question:

Are the factors song tempo and song duration predictors of a song's popularity, and further, are those factors independent of each other?

Hypotheses:

(1) Song Genre:

Null Hypothesis: Song tempo is not a predictor of a song's popularity.

Alternative Hypothesis: Song tempo is a significant predictor of a song's popularity.

(2) Song Length:

Null Hypothesis: Song duration is not a predictor of a song's popularity.

Alternative Hypothesis: Song duration is a significant predictor of a song's popularity.

(3) Independence:

Null Hypothesis: Song tempo and duration are independent from each other.

Alternative Hypothesis: Song tempo and duration are not independent from each other.

About the Data Set:

The dataset spotify_songs.csv is from a TidyTuesday launch on January 21, 2020. The data was gathered by spotifyr, an R wrapper for pulling track audio features and other information from Spotify's Web API in bulk. There are 32833 observations in this data set, but because some songs are put on multiple playlists, those

32833 observations include some duplicate songs. After cleaning the data, our dataset has 28356 observations, and each observation is an individual song. There are 23 variables in this data set, which means that for each song there are 23 variables to identify it, like song name, artist, tempo, and release date. In order to clean the data, we had to remove variables associated with playlist information, so in our clean dataset therre are 19 variables. Thus, the dataset we will be working with has 28356 observations and 19 variables. We will be focusing on two variables: song tempo and song duration. Song tempo refers to the average beats per minute (BPM) of a song. Song duration refers to the length of the genre in minutes. Duration and tempo were calculated by Spotify's Web API.

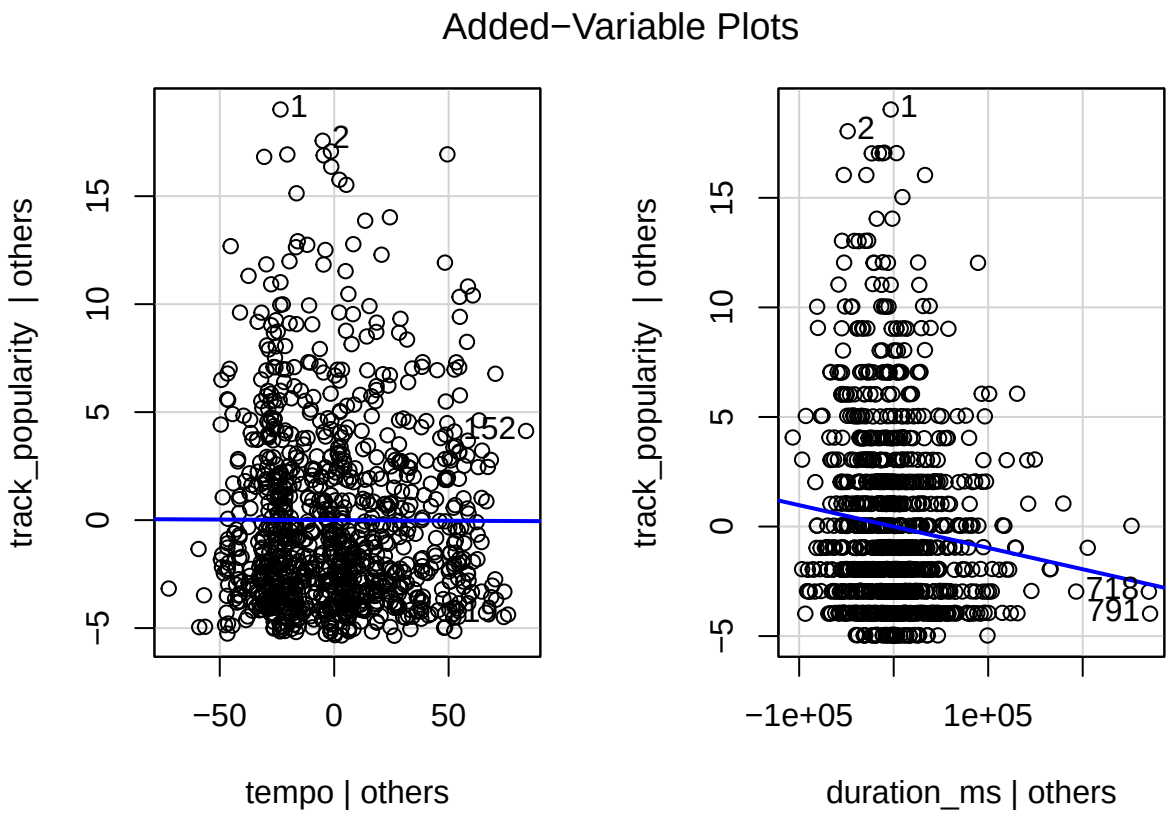
## Methodology

Here, to find out if tempo and duration are accurate predictors of popularity and whether they are independent of each other, we will be using Multiple Regression and a P-value test. In the case of checking whether multiple factors attribute to the popularity of a song, we utilize a multiple regression such that their correlation to one another becomes apparent by way of statistical analysis. Here, we will take the factors, conduct a regression analysis and visualize it using various linear regression models and a 3D multi-regression model of all three factors using the plot3D library. Finally, to prove independence, we will be using a P-value test such that we can check whether the variables are affected by change in each other.

## Results

```
top_1000 = spotify_data_clean %>% arrange(desc(track_popularity)) %>% slice(1:1000)

lm(track_popularity ~ tempo + duration_ms,
   data = top_1000) %>%
  avPlots()
```



```
linear_reg() %>%
  set_engine("lm") %>%
  fit(track_popularity ~ tempo + duration_ms, data = spotify_data_clean) %>% tidy()
```

```
## # A tibble: 3 x 5
##   term          estimate  std.error statistic   p.value
##   <chr>            <dbl>      <dbl>     <dbl>     <dbl>
## 1 (Intercept) 51.2       0.824         62.1   0
## 2 tempo        0.00370   0.00517        0.716 4.74e-  1
## 3 duration_ms -0.0000542 0.00000228   -23.8   1.69e-123
```

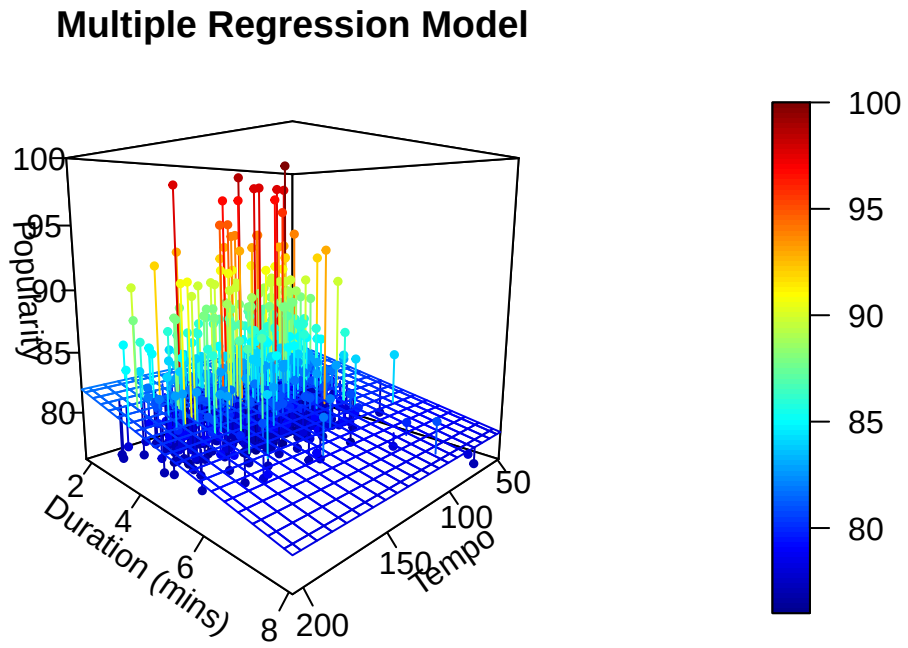
```
x <- top_1000$tempo
y <- top_1000$duration_ms/60000
z <- top_1000$track_popularity

fit <- lm(z ~ x + y)

grid.lines = 20
x.pred <- seq(min(x), max(x), length.out = grid.lines)
y.pred <- seq(min(y), max(y), length.out = grid.lines)
xy <- expand.grid(x = x.pred, y = y.pred)
z.pred <- matrix(predict(fit, newdata = xy),
                 nrow = grid.lines, ncol = grid.lines)


fitpoints <- predict(fit)

scatter3D(x, y, z, pch = 19, cex = 0.5,
    theta = 135, phi = 10, ticktype = "detailed", xlab = "Tempo",
    ylab = "Duration (mins)", zlab = "Popularity" ,
    surf = list(x = x.pred, y = y.pred, z = z.pred,
    facets = NA, fit = fitpoints), main = "Multiple Regression Model")
```

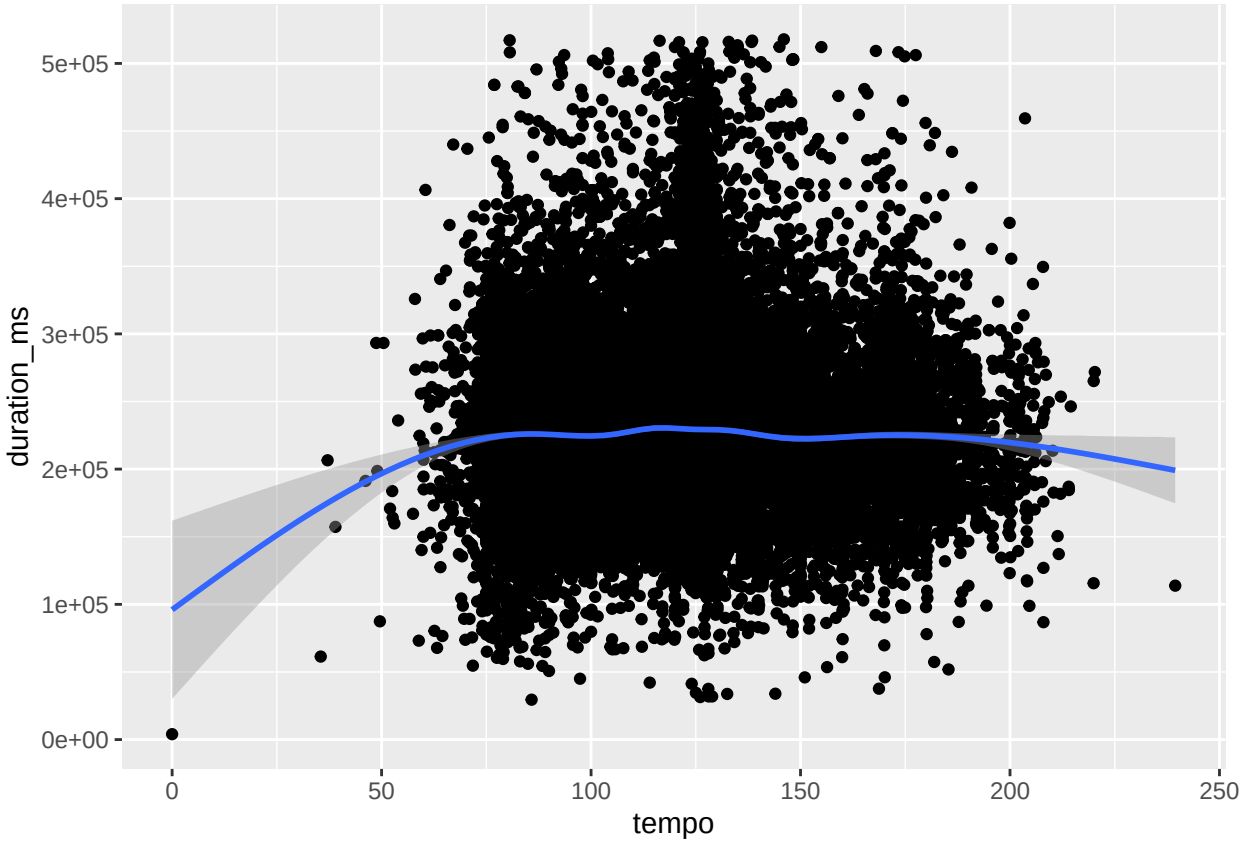



Independence:

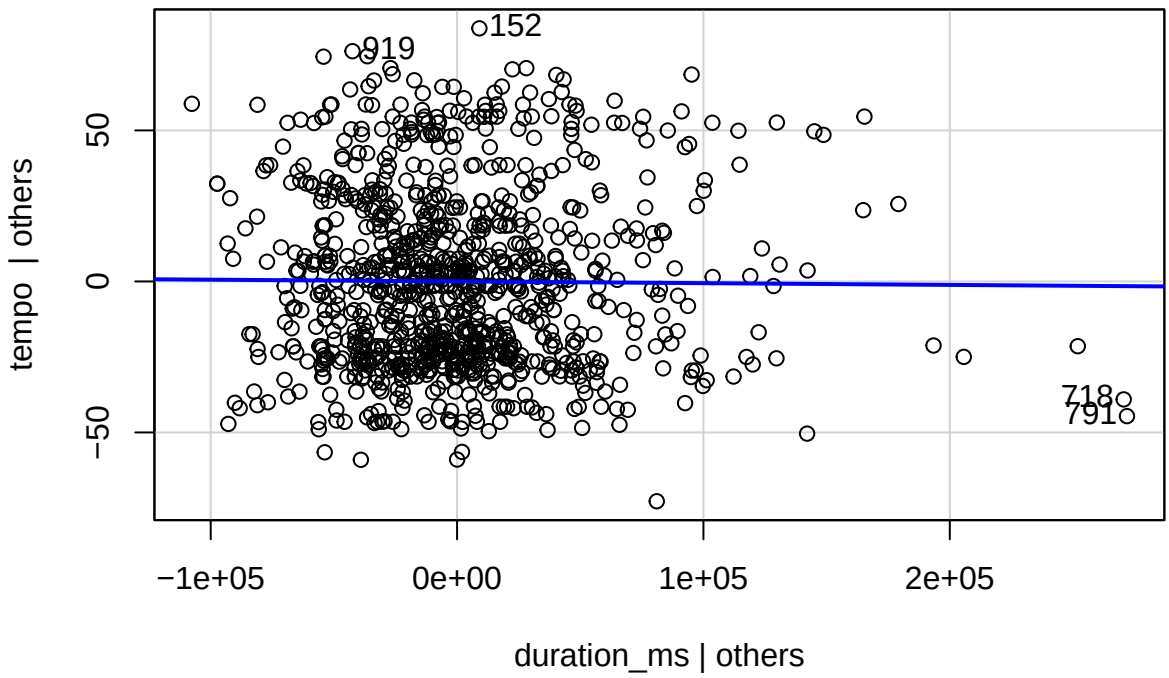
We reject the null for duration_ms, meaning that length does have a significant affect on popularity. The opposite goes for tempo.

```
spotify_data_clean %>%
  ggplot(mapping = aes(x = tempo, y = duration_ms)) +
  geom_point() + geom_smooth()
```

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




```
lm(tempo ~ duration_ms,
   data = top_1000) %>%
  avPlots()
```




```
linear_reg() %>%
  set_engine("lm") %>%
  fit(tempo ~ duration_ms, data = spotify_data_clean) %>% tidy()
```

```
## # A tibble: 2 x 5
##   term             estimate  std.error statistic p.value
##   <chr>               <dbl>      <dbl>     <dbl>   <dbl>
## 1 (Intercept) 121.          0.615        197.      0
## 2 duration_ms  -0.000000748 0.00000262    -0.286   0.775
```

## Discussion