report

Introduction

[space for text]

Exploratory Data Analysis

Steve's Section

```
# Insert code here
```

[space for text]

Zack's Section

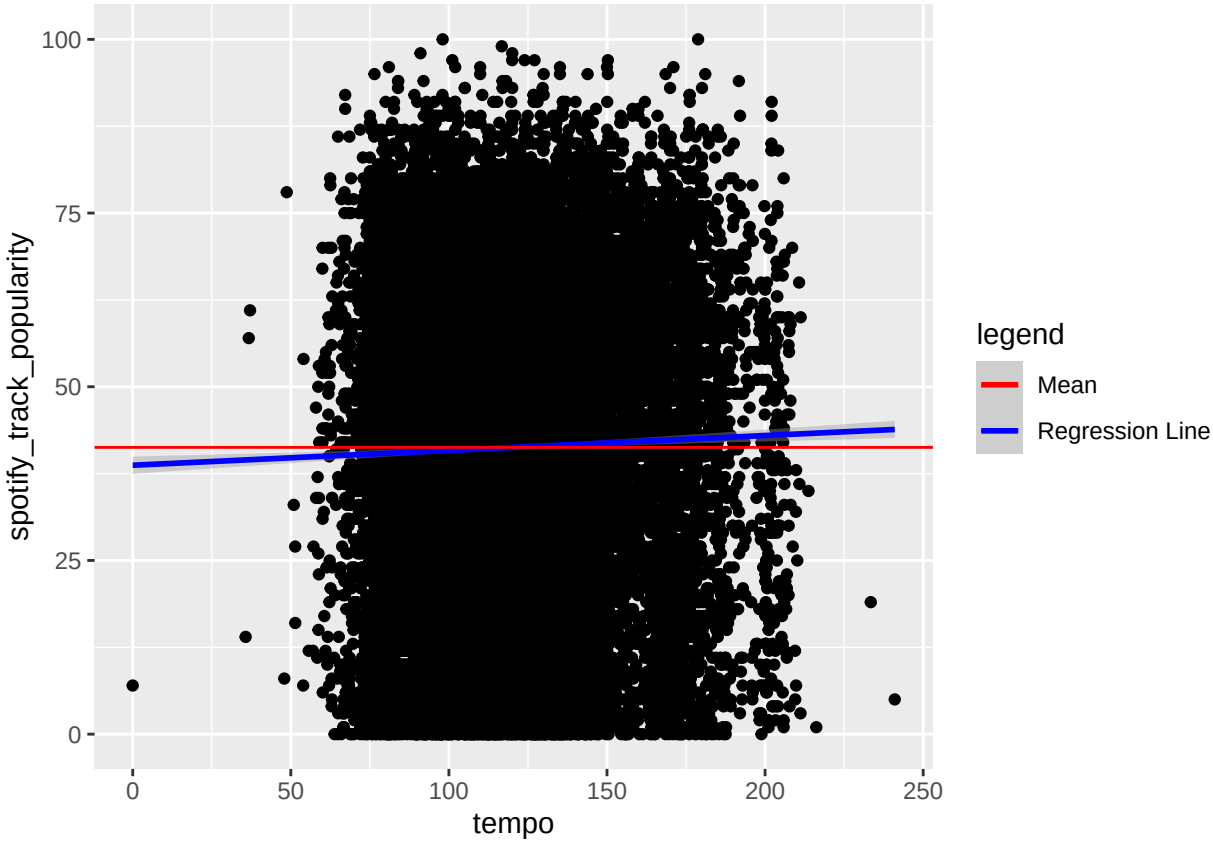
As we have seen above, `peak_performance` isn't the best response variable for popularity in our data set since it's not correlated with many of the audio features. In this section we will look at another measure of popularity of a song which our data set contains, this is `spotify_track_popularity`.

As we can see in the correlation plot (figure ??) in the above section, `spotify_track_popularity` has a correlation with almost all parameters in our data set. This means we are much more likely to be able to actually predict it's popularity based on the values of the other parameters, which will preform better than `peak_performance` did.

Here we see the two least correlated variables with respect to `spotify_track_popularity`:

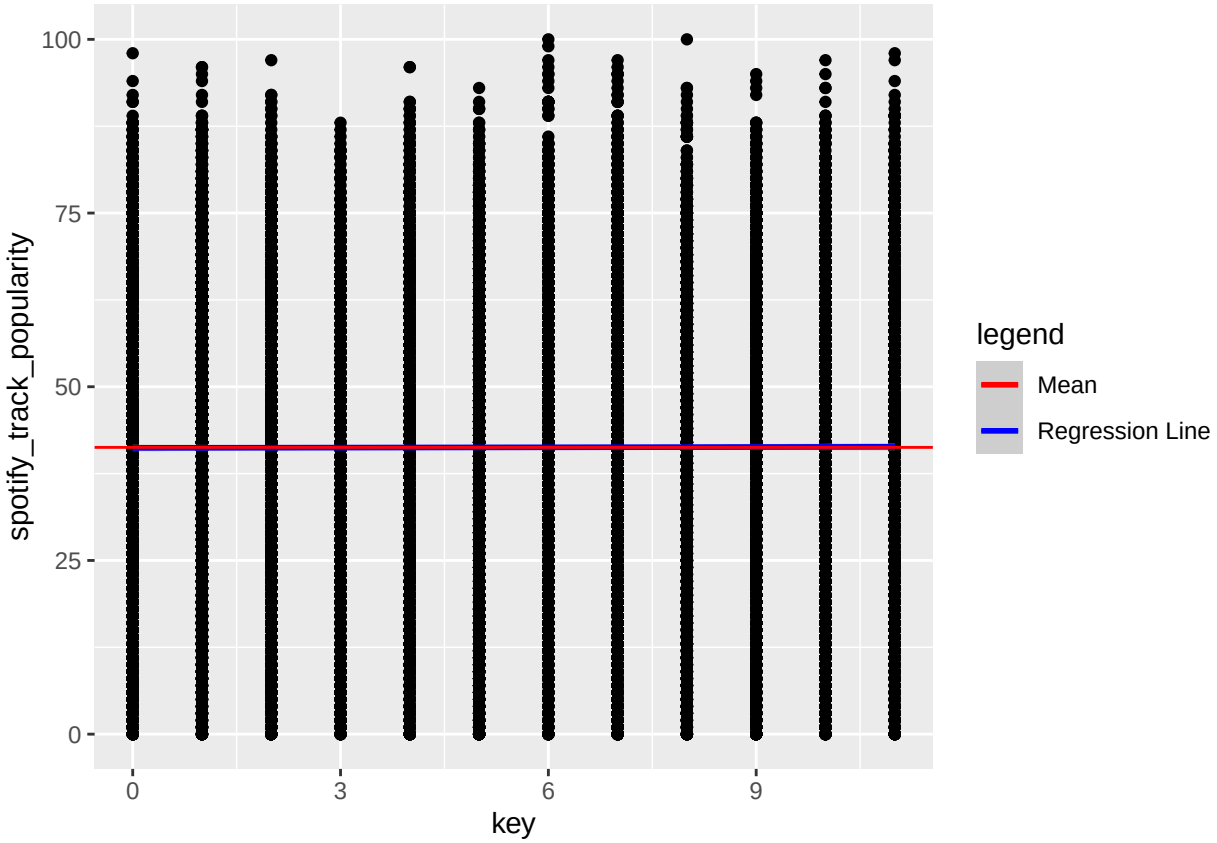
```
mean_popularity <- mean(songs$spotify_track_popularity)
ggplot(data = songs, aes(tempo, spotify_track_popularity) ) +
  geom_point() + geom_smooth(method="lm", aes(color = "Regression Line")) + geom_hline(aes(yintercept =
  scale_color_manual(name = "legend", values = c("red", "blue"))
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
ggplot(data = songs, aes(key, spotify_track_popularity) ) +
  geom_point() + geom_smooth(method="lm", aes(color = "Regression Line")) + geom_hline(aes(yintercept =
  scale_color_manual(name = "legend", values = c("red", "blue"))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



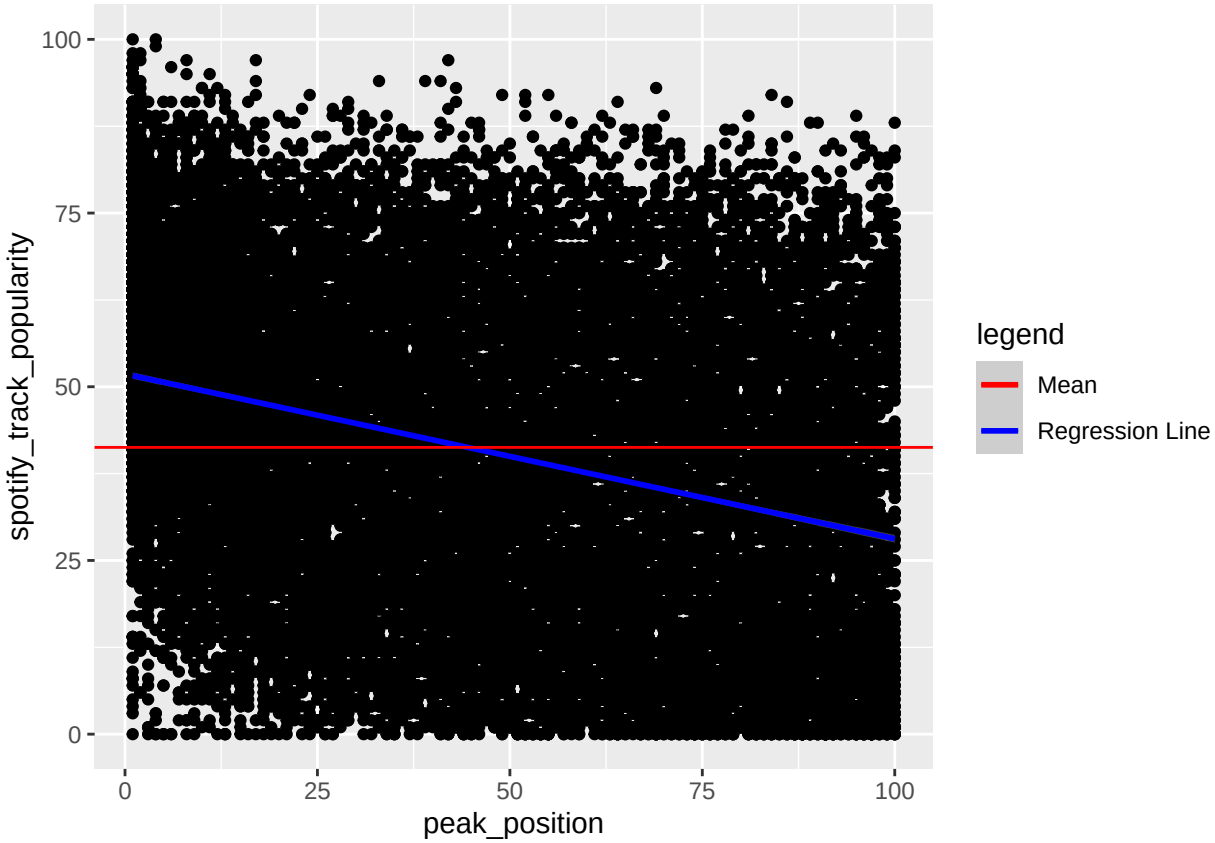
As we can see `key` is quite a unique variable, since it's discrete on a scale of 1-11. But on the plot here we see that the values are roughly equally distributed across the different keys, and all the keys have nearly full coverage of popularity. We see that the mean of the track popularity almost exactly matches the univariate regression line of the two variables, which means that there's almost no information gained about the popularity through looking at the `key`. We expect that using LASSO, this would be thrown out of the model quite quickly, or minimized to be near zero using Ridge, so I'd expect this to have very little weight in any final model we pick. The `key` is an integer mapping to the pitch of the song (ex. A, B, C, A# ...) which explains the lack of information we get from that variable, as popular songs often come in many different pitches.

With respect to popularity `tempo` is also a weird parameter. It's clear from looking at this plot that the vast majority of tempos fall between roughly 60-200 bpm. When looking at the data here it's easy to tell there's not a lot of information to gain here. We see that reinforced from the regression line that nearly matches the mean here too. Logically there's no magic tempo that makes songs into hits and almost all songs fall between this set range, so there's no way of determining which song of a specific tempo is a hit versus one that is unpopular, without using some other parameter as well, so intuitively tempo will not play a large role in predicting the popularity. If we use this for anything, it may make more sense to include tempo as a categorical variable to represent whether a song falls into this range of tempos or not.

The following plots are some of the higest correlation with `spotify_track_popularity`:

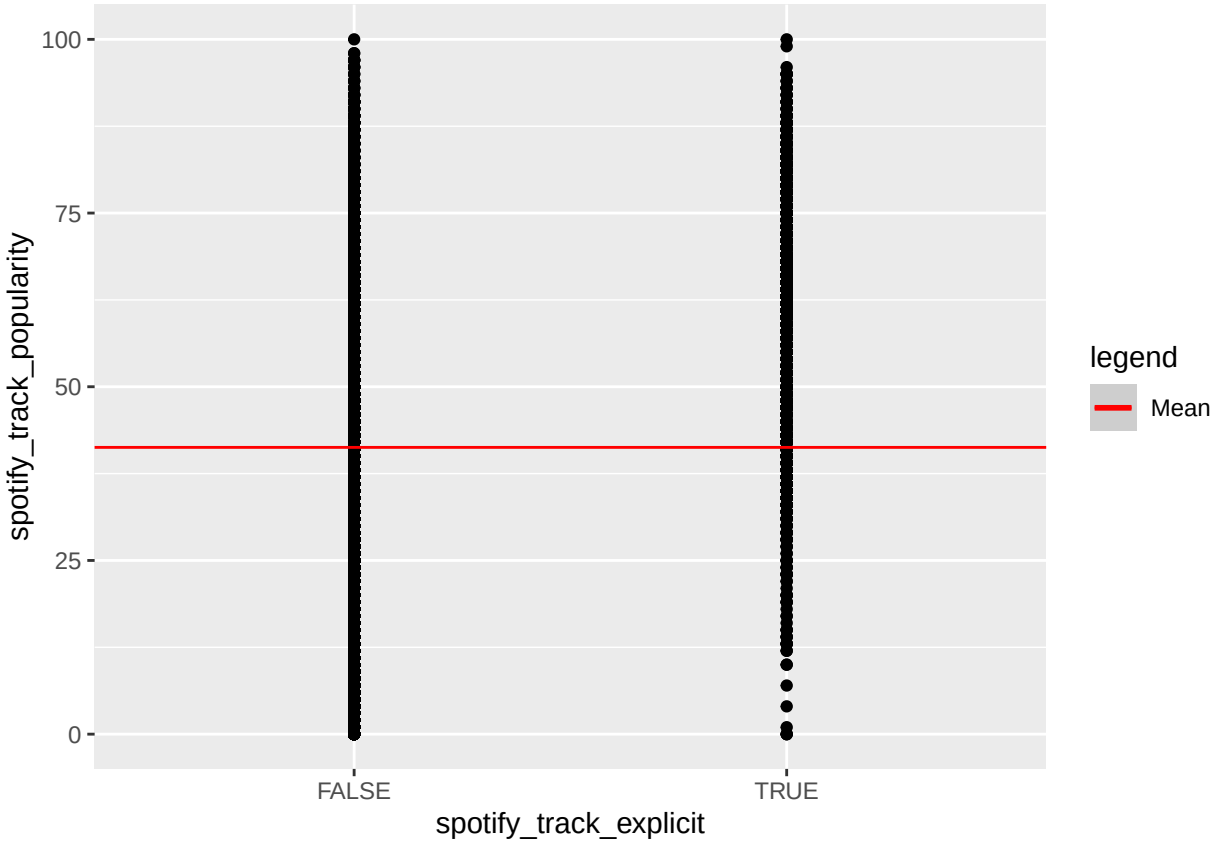
```
ggplot(data = songs, aes(peak_position, spotify_track_popularity) ) +
  geom_point() + geom_smooth(method="lm", aes(color = "Regression Line")) + geom_hline(aes(yintercept =
  scale_color_manual(name = "legend", values = c("red", "blue"))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



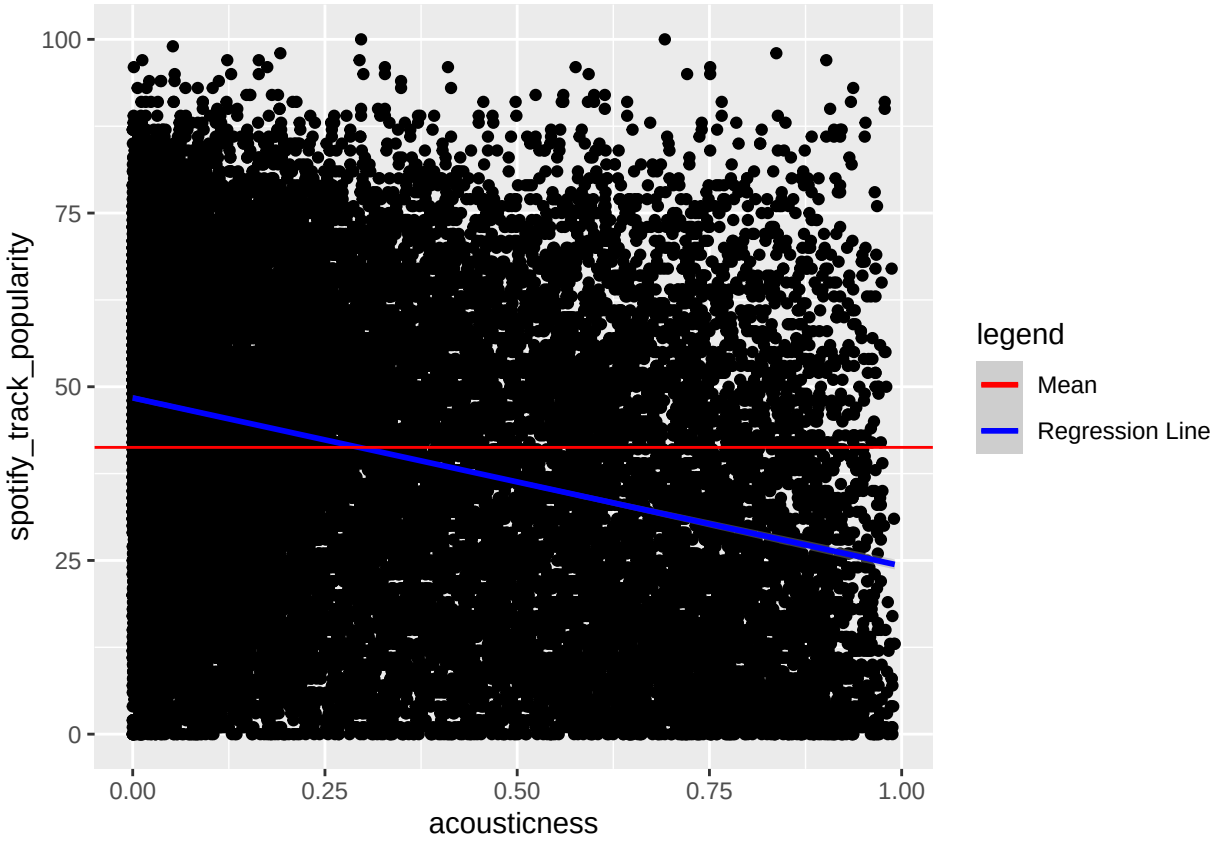

```
ggplot(data = songs, aes(spotify_track_explicit, spotify_track_popularity) ) +
  geom_point() + geom_smooth(method="lm", aes(color = "Regression Line")) + geom_hline(aes(yintercept =
  scale_color_manual(name = "legend", values = c("red", "blue"))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



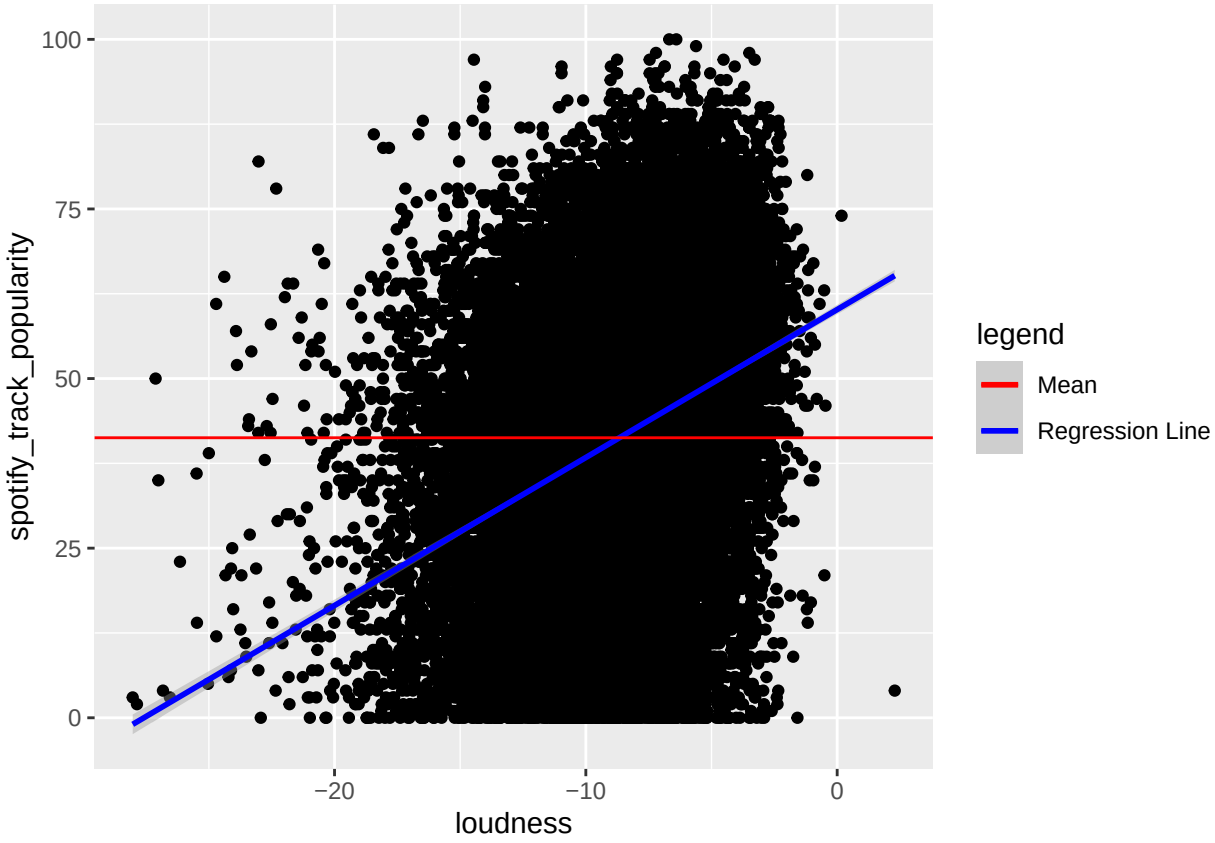

```
ggplot(data = songs, aes(acousticness, spotify_track_popularity) ) +
  geom_point() + geom_smooth(method="lm", aes(color = "Regression Line")) + geom_hline(aes(yintercept =
  scale_color_manual(name = "legend", values = c("red", "blue"))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



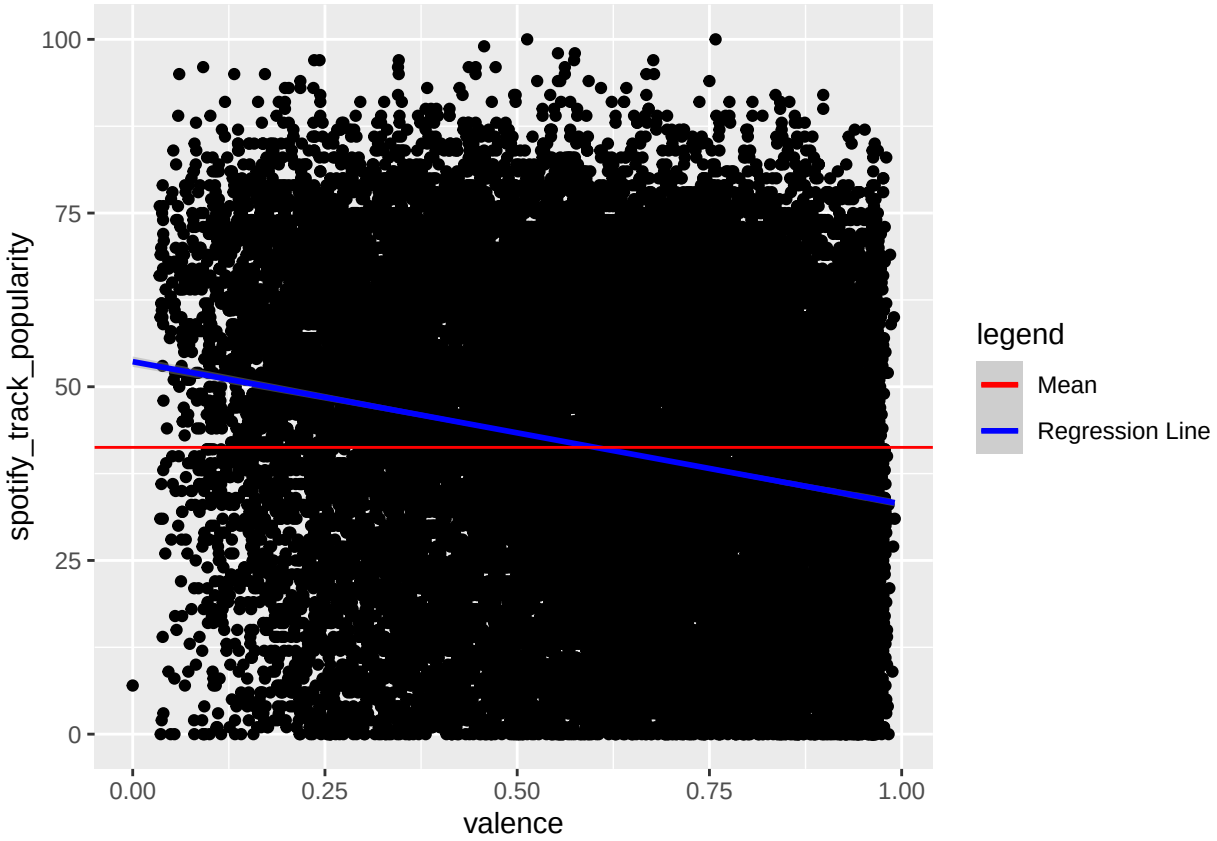

```
ggplot(data = songs, aes(loudness, spotify_track_popularity) ) +
  geom_point() + geom_smooth(method="lm", aes(color = "Regression Line")) + geom_hline(aes(yintercept =
  scale_color_manual(name = "legend", values = c("red", "blue"))
```

```
## `geom_smooth()` using formula 'y ~ x'
```




```
ggplot(data = songs, aes(valence, spotify_track_popularity) ) +
  geom_point() + geom_smooth(method="lm", aes(color = "Regression Line")) + geom_hline(aes(yintercept =
  scale_color_manual(name = "legend", values = c("red", "blue"))
```

```
## `geom_smooth()` using formula 'y ~ x'
```



From the plots above, we see that while these are the "strongest" correlated parameters to our response variable, they are all quite weakly correlated with them. While `peak_position` which is the song's peak on the billboard charts (with 1 being the best) does show a negative correlation with the track popularity, it is quite interestingly not a fantastic predictor, we see a number of number one songs on the billboard charts with low popularity on Spotify. The binary variables `spotify_track_explicit` also shows that there's popular songs that are both explicit and not explicit, so it's strong correlation must come from the fact that there are fewer unpopular (less that 25 popularity) explicit songs. `acousticness` and `valence` are also both audio features that are negatively correlated to the popularity, showing that people must not like highly acoustic or valeant (highly positive) music. While `loudness` is quite a dramatic regression line, in some ways it is similar to `tempo` where most values are centered in a certain range, but this likely does have more effect on the popularity. As we've seen with all these variables we will likely need them all in tandem in order to create a model to predict the popularity of the songs, but we do have correlation in some form across many variables, so `spotify_track_popularity` should be a good response variable.

Anirudh's Section

```
# Insert code here
```

[space for text]