Music

Jason Flores UT EID: *jf36995*

This is the dataset I will be using:

```
audio <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/data/202
spotify_songs <- readr::read_csv('https://raw.githubusercontent.com/rfordatascience/tidytuesday/master/
```

More information about the data set can be found at https://github.com/rfordatascience/tidytuesday/blob/master/data/2021/2021-09-14/readme.md and https://github.com/rfordatascience/tidytuesday/blob/master/data/2020/2020-01-21/readme.md

Data Prep:

```
#The dataset used are already tidy, the functions pivot_wider/longer are used in data analysis
spotify <- audio %>%
  left_join(spotify_songs) %>%
#Removing repeating observations
  distinct(song_id, .keep_all= TRUE)
```

```
## Joining, by = c("danceability", "energy", "key", "loudness", "mode", "speechiness", "acousticness",
```

Joining/Merging

The two datasets above were merged into one using the `left_join` function. The dataset `audio` had 29,503 observations with 22 variables describing the audio quality of each song and time signature. The dataset `spotify_songs` had 32,833 observations with 34 where the variables of audio quality is seen but it includes the tempo of each song. After merging the dataset with `left_join` the total observation was 29,386 observations with 34 variables.

Part 1

Introduction:

The two datasets `audio` and `spotify_songs` are taken from the github thread `rfordatascience`. Here the dataset `audio` is about the audio quality of the songs from the billboard top 100 from the year 2021. It describes the loudness, key, energy and time signature. The dataset `spotify_songs` is about 5000 songs picked from a blogpost that covers EDM, Latin, Pop, R&B, Rap, & Rock. It describes the sub genre of each song and the duration of each song in ms including the tempo.

These two datasets interest me because I want to analyze the analytics of rock songs as that is my favorite type of genre. Generally at rock concerts the crowd goes wild forming mosh pits. I want to see the distribution of time signatures done for each sub genre of rock. Additionally, I want to observe which type of genre of music has the most dance ability.

Approach

The question I will be answering for part 1 is: What is the distribution of time signature within the sub genre of rock?

I will answer this question by using a summary table displaying the sub genre of rock with each column showing the time signature. To achieve the data, first we filter the data set `spotify` using `filter` to focus on

the genre of rock. From there we can count the number of observations of each sub genre of rock along with the time signature using `count`. Next, we can use `pivot_wider` to show the distribution of time signatures of each sub genre of rock. To remove the NA's values we use `mutate(across(everthing..))` to change the NA's value to zero. Then we can find the sum of each category using `rowSums` where we can find the percentage of distribution of time signatures overall.

After obtaining the summary table `rock_sub` we change the data set back to a long format using `pivot_longer` where the number values of the time signature are changed to categorical using `mutate` with `case_when`. From there, a ggplot of a piechart using `geom_arc_bar` is used to view the time signature done for each sub genre of rock.

```
rock_sub <- spotify %>%
#Filtering the data to just rock genre
  filter(playlist_genre == "rock") %>%
#count
   count(playlist_subgenre,time_signature) %>%
#pivot_wider
  pivot_wider(names_from = time_signature, values_from = n)%>%
#changing NA's to 0
  mutate(
    across(everything(), ~replace_na(.x, 0))
  ) %>%
#Finding total for each row
  mutate(
    Total = rowSums(across(where(is.numeric)))
  ) %>%
#Finding percentage of sub genre within the rock genre
  mutate(
    percentage = 100*Total/sum(Total)
  )%>%
#Arranging data
  arrange(desc(Total))
print(rock_sub)
```

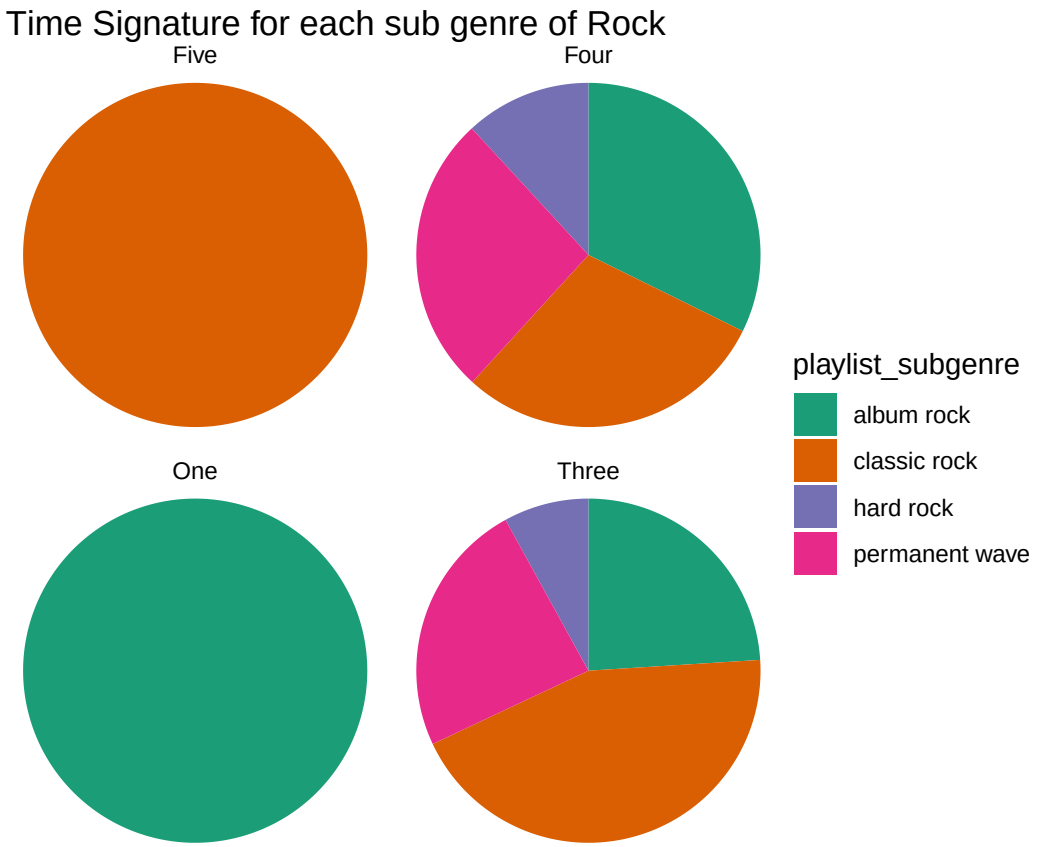
```
## # A tibble: 4 x 7
##   playlist_subgenre   `1`   `3`   `4`   `5` Total percentage
##   <chr>             <dbl> <dbl> <dbl> <dbl> <dbl>      <dbl>
## 1 album rock            1     6   207     0   214       32.0
## 2 classic rock          0    11   189     1   201       30.1
## 3 permanent wave        0     6   169     0   175       26.2
## 4 hard rock             0     2    76     0    78       11.7
```

```
#Changing dataframe to longer format
rockpie <- rock_sub %>%
#Selecting columns 1 to 5
   select(1:5) %>%
#changin wide to long format
      pivot_longer(cols = c(2:5), names_to = "Time_Signature", values_to = "Count") %>%
#changing the time signature value to a categorical
  mutate(
    Time_Sig = case_when(
      Time_Signature == 1 ~ "One",
      Time_Signature == 3 ~ "Three",
      Time_Signature == 4 ~ "Four",
      Time_Signature == 5 ~ "Five",
      TRUE ~ NA_character_
```

```
    )
  ) %>%
#using select to view appropriate variables
  select(playlist_subgenre,Time_Sig,Count)
  print(rockpie)
```

```
## # A tibble: 16 x 3
##    playlist_subgenre Time_Sig Count
##    <chr>             <chr>    <dbl>
##  1 album rock        One          1
##  2 album rock        Three        6
##  3 album rock        Four       207
##  4 album rock        Five         0
##  5 classic rock      One          0
##  6 classic rock      Three       11
##  7 classic rock      Four       189
##  8 classic rock      Five         1
##  9 permanent wave    One          0
## 10 permanent wave    Three        6
## 11 permanent wave    Four       169
## 12 permanent wave    Five         0
## 13 hard rock         One          0
## 14 hard rock         Three        2
## 15 hard rock         Four        76
## 16 hard rock         Five         0
```

```
#graph of rockpie as a pie chart
   ggplot(rockpie) +
  aes(
    x0 = 0, y0 = 0,
    r0 = 0, r = 1,
    amount = Count,
    fill = playlist_subgenre
  ) +
  geom_arc_bar(stat = "pie", color = NA) +
     scale_fill_brewer(palette="Dark2") +
     facet_wrap(~(Time_Sig)) +
  coord_fixed() +
  ggtitle("Time Signature for each sub genre of Rock")+
theme_void()
```



Discussion

As observed in the summary table `rock_sub` the sub genre of rock that was viewed the most in the `spotify` dataset is album rock at 214 times, classic rock at 201 times, permanent wave at 175 times and hard rock at 78 times. My favorite sub genre of rock is hard rock so I am satisfied that it made it on the list. Furthermore, observing the pie chart from `rock_pie` it can be concluded that the majority of sub genre of rock is done in time signature of 3 or 4, with 4 residing more. Additionally, the only time signature done in 5 was classic rock and for 1 it was album rock.

Part 2

Introduction:

As stated previously, the dataset `spotify` is used and the variables in focus is the genre of music and dance ability.

Approach

The question I will be answering for part 2 is: What is the average dance ability for each genre of music?

To answer this question. The method is similar where a summary table of dancerock is viewed showing the average dance ability of each genre of music using `summarize` and `group_by`.

From there, since the date set `dancerock` is already in a long format, a bar plot is is used using `geom_col` to view the distribution.

```
genre <- spotify %>%
#grouping
  group_by(playlist_genre)%>%
```

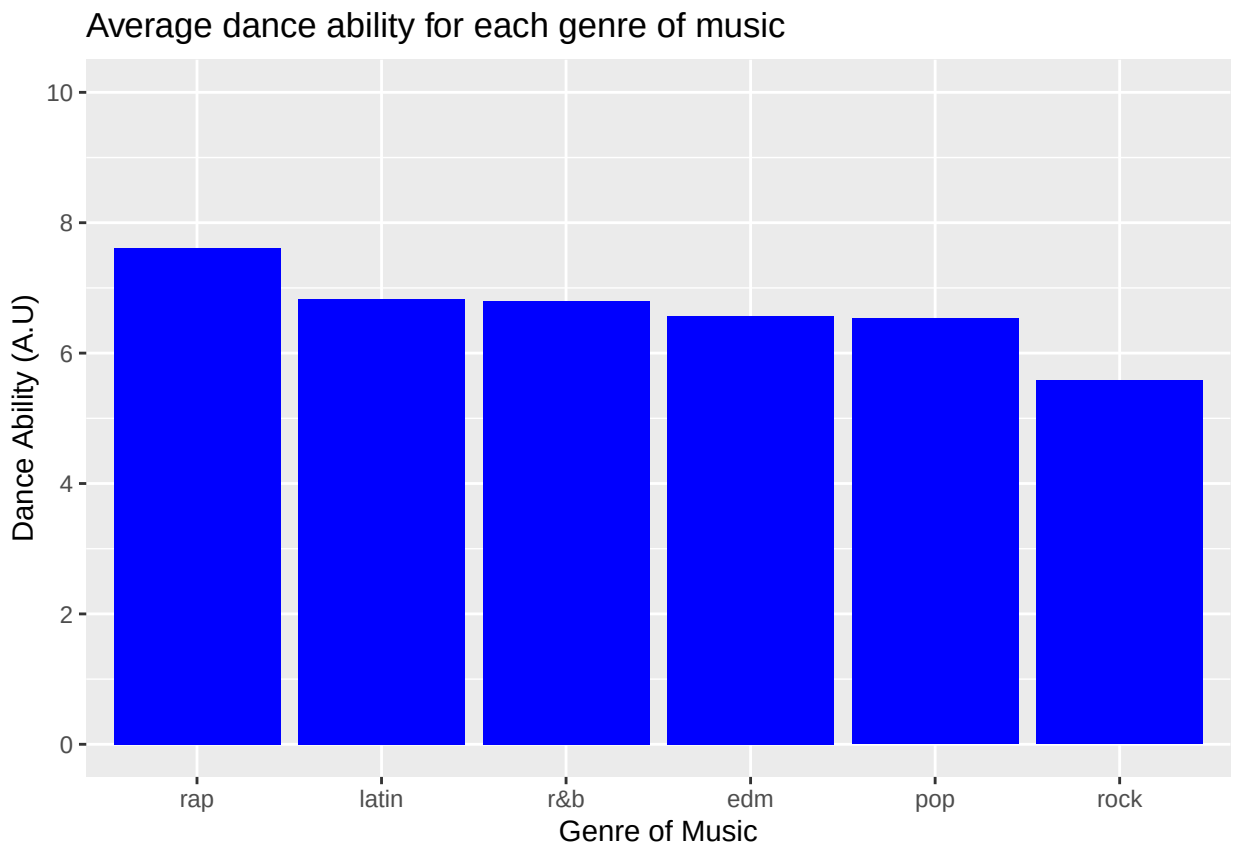
```
#summary
  summarize(
    mean_danceability = mean(danceability)
  )%>%
#changing the scale to 0 to 10  for dance ability
  mutate(
    dance = 10*mean_danceability
  )%>%
#select
  select(playlist_genre,dance)%>%
#removing last row
  slice(1:6)%>%
#arranging data
  arrange(desc(dance))
```

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

```
print(genre)
```

```
## # A tibble: 6 x 2
##   playlist_genre dance
##   <chr>          <dbl>
## 1 rap             7.61
## 2 latin           6.83
## 3 r&b             6.80
## 4 edm             6.57
## 5 pop             6.53
## 6 rock            5.58
```

```
ggplot(genre,
       aes(fct_reorder(playlist_genre,-dance),dance)) +
  geom_col(fill="blue") +
  ggtitle( "Average dance ability for each genre of music") +
  scale_x_discrete(
    name = "Genre of Music"
    )+
  scale_y_continuous(
    name = "Dance Ability (A.U)",
    limits = c(0,10),
    breaks = c(0,2,4,6,8,10),
    labels = c("0","2","4","6","8","10")
  )
```



Discussion

As observed in the bar plot `genre`, the genre of music that has the highest average of dance ability is rap at 7.61, latin at 6.83, r&b at 6.80, edm at 6.57, pop at 6.53 and rock at 5.58. I did not expect rock to be the lowest dance ability considering the songs are done in a faster tempo with drums and guitars. Additionally, I am surprised that rap has the highest average, beating latin and edm. I expected edm to be the highest since it is essentially party music, upbeat and fun.