Jaqueline Ma SDS Exploratory Project

Introduction

For my exploratory project, I decided to choose two datasets on topics that I love, music and books. For both datasets, the data was pulled from the kaggle website, where the *books* dataset was data on more than 20,000 books on Goodreads' most popular book lists (Goodreads.com is the world's largest site for readers and book recommendations - this description was found through a Google search). The *music* dataset was based on Spotify data and was made up of the top 100 songs on Spotify in 2018. The *books* dataset includes variables such as author average rating, author gender, author genres, author id, author name, author page url, author rating count, author review count, birthplace, book average rating, book full url, book id, book title, genre1, genre2, number of ratings, number of reviews, pages, publish date and score. The *music* dataset includes variables such as id, name, artists, genre, dancability, energy, key, loudness, mode, speechiness, acousticness, instrumentalness, liveness, valence, tempo, duration ms (in milliseconds), and time signature.

For the *books* dataset, the variables of book's url and author's page url were discarded for simplicity. In addition, the two genre variables were united to create one genre variable.

I chose these variables since I love reading books and often use Goodreads as a website when looking up book descriptions or when trying to find new books to read. I also do a fair amount of listening to music and wanted to find out what the top songs on Spotify were (Spotify is the main way I listen to my music).

It is possible that there are some genres that might be the same between the *music* and the *books* dataset. However, since genres of music and genres of books often differ, I wouldn't be surprised if there were no associations between the two datasets even though they share a variable name.

```
library(tidyverse)
music <- read.csv("top2018.csv")
glimpse(music)
```

```
## Observations: 100
## Variables: 17
## $ id               <fct> 6DCZcSspjsKoFjzjrWoCd, 3ee8Jmje8o58CHK66QrVC,...
## $ name             <fct> "God's Plan", "SAD!", "rockstar (feat. 21 Sav...
## $ artists          <fct> Drake, XXXTENTACION, Post Malone, Post Malone...
## $ genre            <fct> Hip-Hop/Rap, Hip-Hop/Rap, Hip-Hop/Rap, Hip-Ho...
## $ danceability     <dbl> 0.754, 0.740, 0.587, 0.739, 0.835, 0.680, 0.8...
## $ energy           <dbl> 0.449, 0.613, 0.535, 0.559, 0.626, 0.563, 0.7...
## $ key              <int> 7, 8, 5, 8, 1, 10, 5, 9, 7, 9, 2, 6, 8, 0, 7,...
## $ loudness         <dbl> -9.211, -4.880, -6.090, -8.011, -5.833, -5.84...
## $ mode             <int> 1, 1, 0, 1, 1, 1, 0, 0, 1, 0, 1, 0, 1, 1, 1, ...
## $ speechiness      <dbl> 0.1090, 0.1450, 0.0898, 0.1170, 0.1250, 0.045...
## $ acousticness     <dbl> 0.03320, 0.25800, 0.11700, 0.58000, 0.05890, ...
## $ instrumentalness <dbl> 8.29e-05, 3.72e-03, 6.56e-05, 0.00e+00, 6.00e...
## $ liveness         <dbl> 0.5520, 0.1230, 0.1310, 0.1120, 0.3960, 0.136...
## $ valence          <dbl> 0.357, 0.473, 0.140, 0.439, 0.350, 0.374, 0.6...
## $ tempo            <dbl> 77.169, 75.023, 159.847, 140.124, 91.030, 145...
## $ duration_ms      <int> 198973, 166606, 218147, 221440, 217925, 23126...
## $ time_signature   <int> 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, ...
```

```
books <- read.csv("good_reads_final.csv")
glimpse(books)
```

```
## Observations: 22,891
```

```
## Variables: 20
## $ author_average_rating <dbl> 4.01, 4.15, 4.00, 3.88, 4.10, 3.77, 4.16...
## $ author_gender         <fct> female, male, female, male, female, male...
## $ author_genres         <fct> "historical-fiction,", "literature-ficti...
## $ author_id             <int> 74489, 706255, 5618190, 37871, 36122, 58...
## $ author_name           <fct> Victoria Thompson
## , Stieg Larsson
## , Mimi...
## $ author_page_url       <fct> /author/show/74489.Victoria_Thompson, /a...
## $ author_rating_count   <int> 74399, 3726435, 76496, 5522, 291013, 479...
## $ author_review_count   <int> 6268, 142704, 7975, 489, 13453, 3240, 94...
## $ birthplace            <fct>  United States
##     , Sweden
##     ,  Unite...
## $ book_average_rating   <dbl> 4.02, 4.13, 3.99, 4.14, 4.01, 3.80, 3.95...
## $ book_fullurl          <fct> https://www.goodreads.com/book/show/6867...
## $ book_id               <fct> 686717, 2429135, 27833684, 382975, 64207...
## $ book_title            <fct> "\n      Murder on St. Mark's Place\n", ...
## $ genre_1               <fct> Mystery, Fiction, Romance, Fiction, Fant...
## $ genre_2               <fct> Historical, Mystery, Contemporary, Magic...
## $ num_ratings           <int> 5260, 2229163, 2151, 1844, 17051, 17122,...
## $ num_reviews           <int> 375, 65227, 391, 173, 1890, 561, 1107, 4...
## $ pages                 <fct> 277, 465, 354, 438, 326, 104, 327, 319, ...
## $ publish_date          <fct> 2000, August 2005, 2016, 1970, April 15t...
## $ score                 <int> 3230, 3062, 4585, 1533, 2105, 4372, 2396...
```

```
books<- books%>%select(-book_fullurl, -author_page_url)
books<- books%>%unite(genre_1, genre_2, col = "genre")

mus1<-music%>%pivot_wider(names_from="genre", values_from = "genre")
mus1%>%pivot_longer(col = c("Hip-Hop/Rap":"Alternative/Indie"), names_to = "genre",
                    values_to = "value")%>%na.omit()%>%select(-value)
```

```
## # A tibble: 100 x 17
##    id    name  artists danceability energy   key loudness  mode speechiness
##    <fct> <fct> <fct>          <dbl>  <dbl> <int>    <dbl> <int>       <dbl>
##  1 6DCZ~ God'~ Drake          0.754  0.449     7    -9.21     1      0.109
##  2 3ee8~ SAD!  XXXTEN~        0.74   0.613     8    -4.88     1      0.145
##  3 0e7i~ rock~ Post M~        0.587  0.535     5    -6.09     0      0.0898
##  4 3swc~ Psyc~ Post M~        0.739  0.559     8    -8.01     1      0.117
##  5 2G7V~ In M~ Drake          0.835  0.626     1    -5.83     1      0.125
##  6 7dt6~ Bett~ Post M~        0.68   0.563    10    -5.84     1      0.0454
##  7 58q2~ I Li~ Cardi B        0.816  0.726     5    -4.00     0      0.129
##  8 7ef4~ One ~ Calvin~        0.791  0.862     9    -3.24     0      0.11
##  9 76cy~ IDGAF Dua Li~        0.836  0.544     7    -5.98     1      0.0943
## 10 08bN~ FRIE~ Marshm~        0.626  0.88      9    -2.38     0      0.0504
## # ... with 90 more rows, and 8 more variables: acousticness <dbl>,
## #   instrumentalness <dbl>, liveness <dbl>, valence <dbl>, tempo <dbl>,
## #   duration_ms <int>, time_signature <int>, genre <chr>
```

```
#mus1 after na.omit becomes the same as music

b1<-books%>%pivot_wider(names_from = "genre", values_from = "genre")
b1%>%pivot_longer(col = c("Mystery_Historical":"Self Help_Religion",
                          "Nonfiction_Inspirational", "Christian_Inspirational",
```

```
                          "Christian_Spirituality"), names_to = "genre",
                 values_to = "value")%>%select(-value)
```

```
## # A tibble: 27,263,181 x 17
##    author_average_~ author_gender author_genres author_id author_name
##               <dbl> <fct>         <fct>             <int> <fct>
##  1             4.01 female        historical-f~     74489 "Victoria ~
##  2             4.01 female        historical-f~     74489 "Victoria ~
##  3             4.01 female        historical-f~     74489 "Victoria ~
##  4             4.01 female        historical-f~     74489 "Victoria ~
##  5             4.01 female        historical-f~     74489 "Victoria ~
##  6             4.01 female        historical-f~     74489 "Victoria ~
##  7             4.01 female        historical-f~     74489 "Victoria ~
##  8             4.01 female        historical-f~     74489 "Victoria ~
##  9             4.01 female        historical-f~     74489 "Victoria ~
## 10             4.01 female        historical-f~     74489 "Victoria ~
## # ... with 27,263,171 more rows, and 12 more variables:
## #   author_rating_count <int>, author_review_count <int>,
## #   birthplace <fct>, book_average_rating <dbl>, book_id <fct>,
## #   book_title <fct>, num_ratings <int>, num_reviews <int>, pages <fct>,
## #   publish_date <fct>, score <int>, genre <chr>
```

```
#b1 after selecting becomes the same as books
```

I used the pivot_longer and pivot_wider functions to demonstrate my use of these functions (the datasets were already tidy).

Below, I used full_join to join together the *books* and *music* datasets. I used this join since I did not think that any of the observations would match up (this was proven with running inner_join; there were no rows in common between the two datasets). To preserve all of my data, I used a full_join so that none of the observations were dropped (all of the data was kept) and the blanks were filled with NAs. A potential problem with this join is that I have to make sure to use na.omit when calculating my summary statistics to avoid getting an error in taking the mean of something that does not have a value for each observation (such as tempo).

```
music%>%inner_join(books)
```

```
##  [1] id                   name                 artists
##  [4] genre                danceability         energy
##  [7] key                  loudness             mode
## [10] speechiness          acousticness         instrumentalness
## [13] liveness             valence              tempo
## [16] duration_ms          time_signature       author_average_rating
## [19] author_gender        author_genres        author_id
## [22] author_name          author_rating_count  author_review_count
## [25] birthplace           book_average_rating  book_id
## [28] book_title           num_ratings          num_reviews
## [31] pages                publish_date         score
## <0 rows> (or 0-length row.names)
```

```
fullset<-music%>%full_join(books)
```

Summary Statistics

In the code below, I used filter, select, arrange, group_by, mutate and summarize to manipulate and explore the fully joined dataset. For the new variable that was created, I made a total author rating, which multiplied the average author rating by the total number of author reviews. This variable allowed you to see the total value of all the reviews that an author received.

```
fullset%>%filter(loudness>-5)%>%arrange(loudness)%>%select(loudness, id, name)%>%glimpse()
```

```
## Observations: 41
## Variables: 3
## $ loudness <dbl> -4.985, -4.979, -4.946, -4.929, -4.880, -4.877, -4.85...
## $ id       <fct> 3Ga6eKrUFf12ouh9Yw3v2, 77UjLW8j5UAGAGVGhR5oU, 3GCdLUS...
## $ name     <fct> "Downtown", "Pray For Me (with Kendrick Lamar)", "All...
```

```
fullset%>%select(author_gender)%>%count(author_gender)%>%distinct%>%na.omit
```

```
## # A tibble: 2 x 2
##   author_gender     n
##   <fct>         <int>
## 1 female        10690
## 2 male          12201
```

```
fullset%>%group_by(genre)%>%summarize(n=n())%>%arrange(desc(n))
```

```
## # A tibble: 1,197 x 2
##    genre                       n
##    <chr>                   <int>
##  1 Fantasy_Young Adult      1164
##  2 Romance_Romance          1124
##  3 Historical_Fiction        680
##  4 Fantasy_Fantasy           678
##  5 Young Adult_Fantasy       639
##  6 Young Adult_Contemporary  482
##  7 Classics_Fiction          453
##  8 Romance_Contemporary      409
##  9 Fiction_Historical        399
## 10 Fantasy_Fiction           381
## # ... with 1,187 more rows
```

```
fullset%>%mutate(tot_author_rate=author_average_rating*author_rating_count) %>%
  arrange(desc(tot_author_rate))%>%select(tot_author_rate, author_average_rating,
                                          author_rating_count, author_name)%>%glimpse()
```

```
## Observations: 22,991
## Variables: 4
## $ tot_author_rate       <dbl> 93972065, 93925345, 93925345, 93925345, ...
## $ author_average_rating <dbl> 4.45, 4.45, 4.45, 4.45, 4.45, 4.45, 4.03...
## $ author_rating_count   <int> 21117318, 21106819, 21106819, 21106819, ...
## $ author_name           <fct> J.K. Rowling
## , J.K. Rowling
## , J.K. Rowli...
```

In the first group of codes below, I used summary statistics on the numeric variables of danceability, tempo, and duration using mean, sd, count (n) and variance. Based on this code, duration seemed to have the most amount of variance. Next, I found the median, min and max number of pages (I had to make pages into a numeric vector) where I found out that the minimum number of pages was one! In addition, I found the IQR

of the acousticness variable as well as the quantile of liveness which helped describe the variance found in acousticness and told me that a liveness of 0.350 was in the 10th quantile. Lastly, I found the correlation between danceability, tempo, and duration and based on this matrix, loudness and energy had the highest correlation. I could only use a correlation matrix between these variables due to having NAs if I included pages (since pages was originally from a different database).

For the second group of codes, I used summary statistics on categorical variables of author id to count the number of distinct authors there were, which turned out to be 12,155. I also grouped by author gender to figure out the mean of the number of pages for each gender and discovered that male authors had a higher number of pages, on average. Lastly, I used summary statistics to group by the categorical variables of author genre and author name to then find the first in the numerical ratings of books, which was under the fiction genre with the author name of V.W. Singer.

```
#First group of code
fullset%>%select(danceability, tempo, duration_ms)%>%na.omit%>%
  summarize(meandance = mean(danceability), meantempo = mean(tempo),
            meandur = mean(duration_ms), sddance = sd(danceability), sdtempo = sd(tempo),
            sddur = sd(duration_ms), vardance = var(danceability), vartempo = var(tempo),
            vardur = var(duration_ms), count= n())
```

```
##   meandance meantempo  meandur   sddance  sdtempo    sddur   vardance
## 1   0.71646  119.9042 205206.8 0.1310701 28.79598 40007.89 0.01717938
##   vartempo     vardur count
## 1 829.2087 1600631535   100
```

```
fullset$pages<-as.numeric(as.factor(fullset$pages))
fullset%>%select(pages)%>%na.omit%>%summarize(medpage = median(fullset$pages, na.rm=T))
```

```
##   medpage
## 1     353
```

```
fullset%>%select(pages)%>%na.omit%>%min
```

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

```
fullset%>%select(pages)%>%na.omit%>%max
```

```
## [1] 964
```

```
fullset%>%group_by(genre)%>%select(acousticness)%>%na.omit%>%
  summarize(IQRacoustic = IQR(acousticness))%>%arrange(desc(IQRacoustic))
```

```
## # A tibble: 6 x 2
##   genre             IQRacoustic
##   <chr>                   <dbl>
## 1 Alternative/Indie       0.446
## 2 R&B                     0.353
## 3 Reggaeton               0.268
## 4 Hip-Hop/Rap             0.182
## 5 Dance/Electronic        0.175
## 6 Pop                     0.142
```

```
fullset%>%select(liveness)%>%na.omit%>%mutate(quantile = ntile(liveness, seq(10)))%>%
  arrange(desc(quantile))%>%glimpse
```

```
## Observations: 100
## Variables: 2
## $ liveness <dbl> 0.350, 0.334, 0.297, 0.255, 0.372, 0.183, 0.173, 0.12...
## $ quantile <int> 10, 8, 8, 8, 7, 7, 7, 6, 6, 6, 6, 6, 6, 5, 5, 5, 5, 5...
```

```
numbers <- fullset%>%select_if(is.numeric)%>%select(danceability, energy, key, loudness)%>%
  na.omit
cor(numbers)
```

```
##              danceability      energy         key    loudness
## danceability   1.00000000 -0.07258179 -0.05175895  0.01551749
## energy        -0.07258179  1.00000000 -0.13634473  0.73271905
## key           -0.05175895 -0.13634473  1.00000000 -0.10530884
## loudness       0.01551749  0.73271905 -0.10530884  1.00000000
```

```
#Second group of code
fullset%>%select(author_id)%>%na.omit%>%distinct%>%count
```

```
## # A tibble: 1 x 1
##       n
##   <int>
## 1 12155
```

```
fullset%>%select(pages, author_gender)%>%group_by(author_gender)%>%
  summarize(mean(pages))%>%na.omit
```

```
## # A tibble: 2 x 2
##   author_gender `mean(pages)`
##   <fct>                 <dbl>
## 1 female                 363.
## 2 male                   374.
```

```
fullset%>%group_by(author_genres, author_name) %>%summarize(first=first(num_ratings))%>%
  arrange(first)
```

```
## # A tibble: 12,157 x 3
## # Groups:   author_genres [1,804]
##    author_genres            author_name        first
##    <fct>                    <fct>              <int>
##  1 fiction,                 "V.W. Singer\n"        1
##  2 memoir,                  "Aysha Akhtar\n"       2
##  3 spirituality,            "Mimi Novic\n"         2
##  4 children-s,              "Richard Denney\n"     3
##  5 children-s,              "T.M. McLean\n"        3
##  6 fantasy,                 "S.D. Grimm\n"         3
##  7 fantasy,science-fiction, "C.D. John\n"          3
##  8 fiction,                 "Rick Copper\n"        3
##  9 romance,                 "Rachel  Harris\n"     4
## 10 young-adult,             "Nandini Bajpai\n"     4
## # ... with 12,147 more rows
```

Visualizations

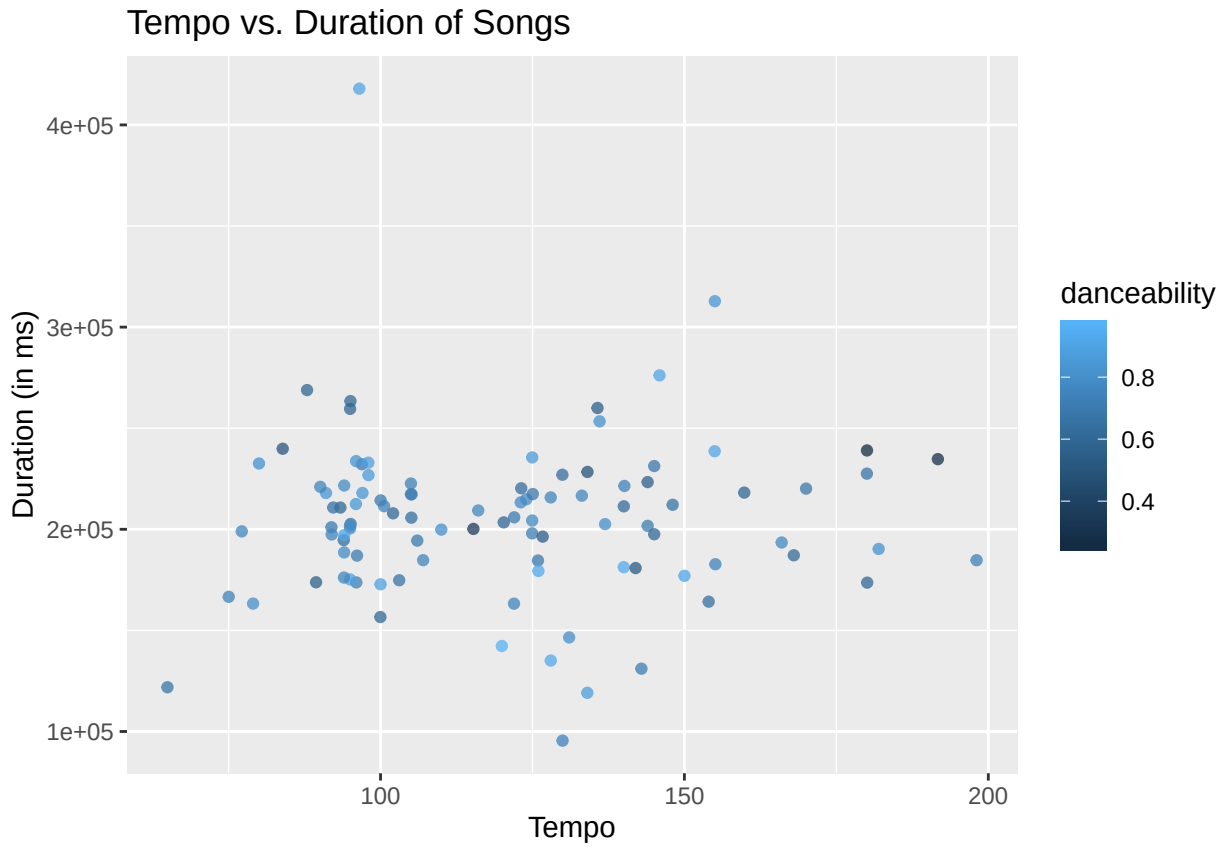
For the visualizations, I created ggplots to compare the variables that I ran summary statistics on. For the first ggplot, I used geom_point to create a scatterplot of tempo vs duration and colored the points with danceability. Based on the graph made, there appears to be higher danceability when tempo is high and a somewhat lower danceability when duration is high. However, most of the points appear to be clustered around the same duration so it may be difficult to find any apparent trends between these three variables.

For the second plot, I made a correlation heatmap of my numeric variables of danceability, energy, key,

and loudness with geom_tile. Based on this correlation plot, energy and loudness were the most strongly correlated and the other variables were not very strongly correlated with one another.

For the last plot, the number of pages were compared with the author's gender and the bar plots were colored by gender. Based on this graph, there is a difference between the mean number of pages for male authors and the mean number of pages for female authors (male authors have a higher number of average pages in their books).

```
ggplot(fullset, aes(tempo, duration_ms, color = danceability))+geom_point(alpha=0.75)+
  ggtitle("Tempo vs. Duration of Songs")+ xlab("Tempo")+
  ylab("Duration (in ms)")
```

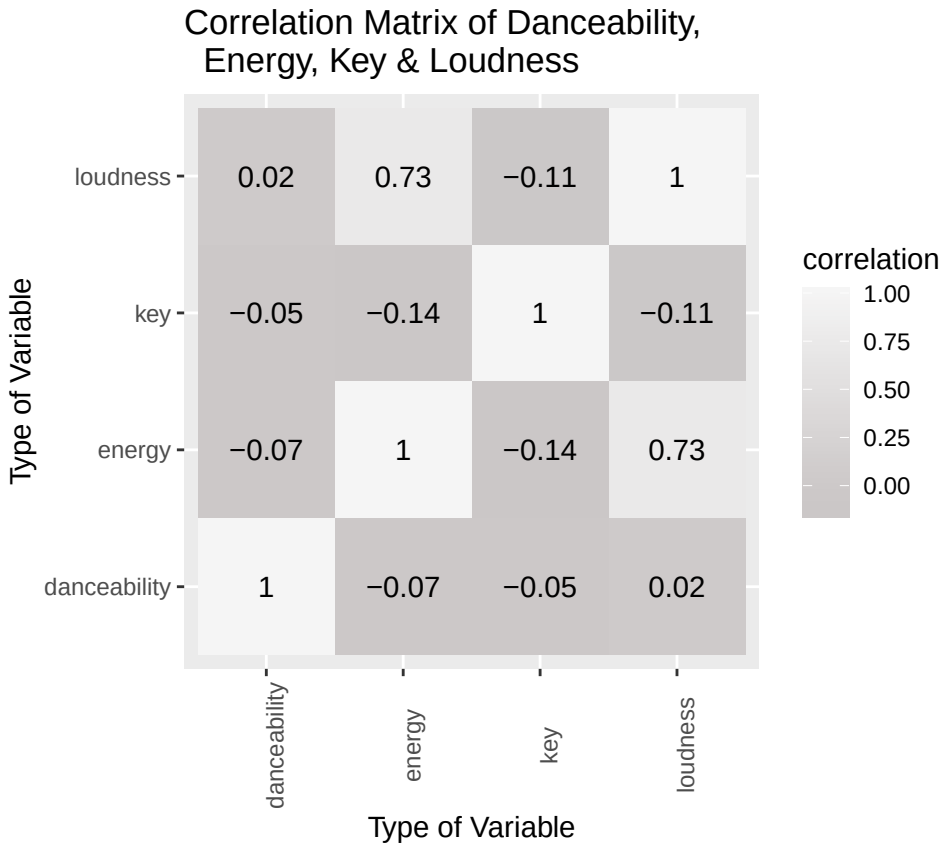



```
tidycor <- cor(numbers)%>%as.data.frame%>%rownames_to_column%>%
  pivot_longer(-1, names_to = "name", values_to = "correlation")
head(tidycor)
```

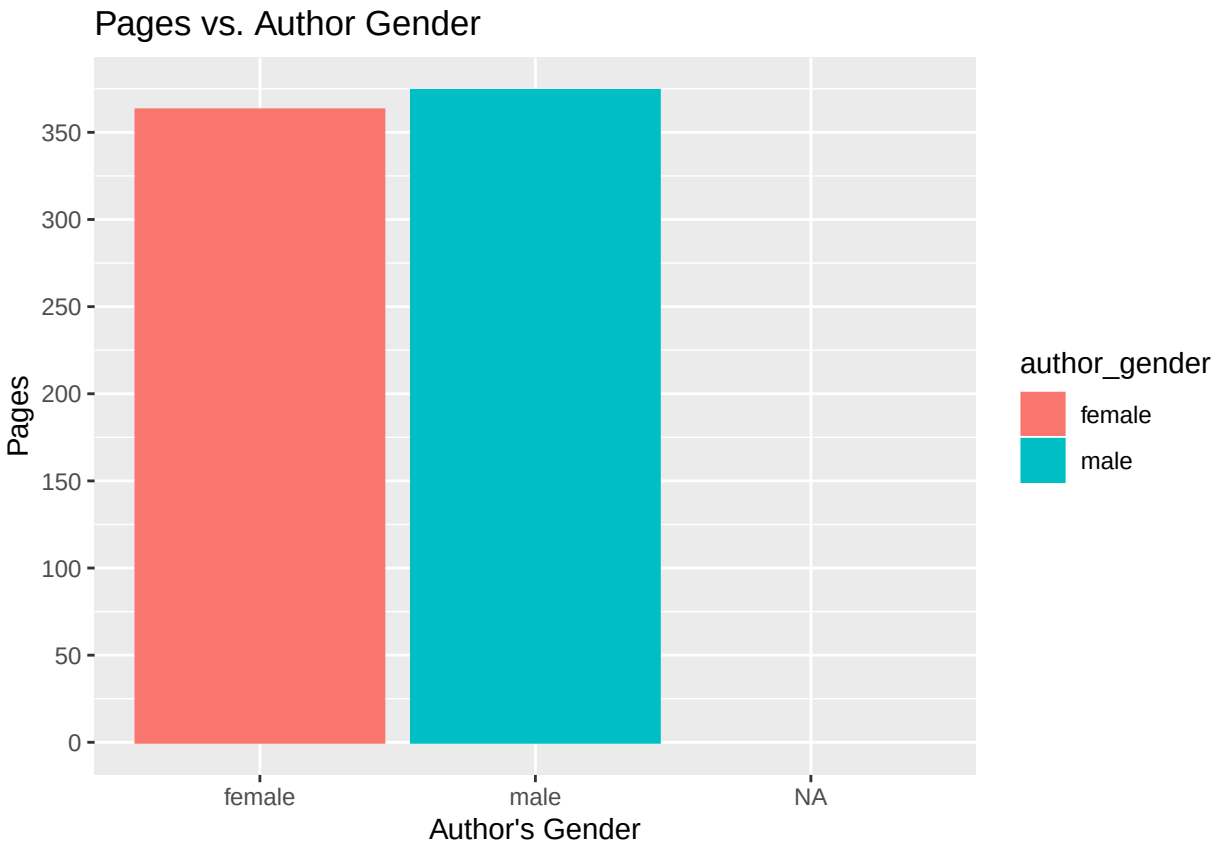
```
## # A tibble: 6 x 3
##   rowname      name         correlation
##   <chr>        <chr>              <dbl>
## 1 danceability danceability      1
## 2 danceability energy           -0.0726
## 3 danceability key              -0.0518
## 4 danceability loudness          0.0155
## 5 energy       danceability     -0.0726
## 6 energy       energy            1
```

```
tidycor%>%ggplot(aes(rowname, name, fill = correlation))+geom_tile()+
  scale_fill_gradient2(low="gray", mid = "snow3", high = "whitesmoke")+
```

```
  geom_text(aes(label=round(correlation,2)),color = "black", size = 4)+
  theme(axis.text.x = element_text(angle = 90))+ coord_fixed()+
  ggtitle("Correlation Matrix of Danceability,
  Energy, Key & Loudness")+xlab("Type of Variable")+ylab("Type of Variable")
```



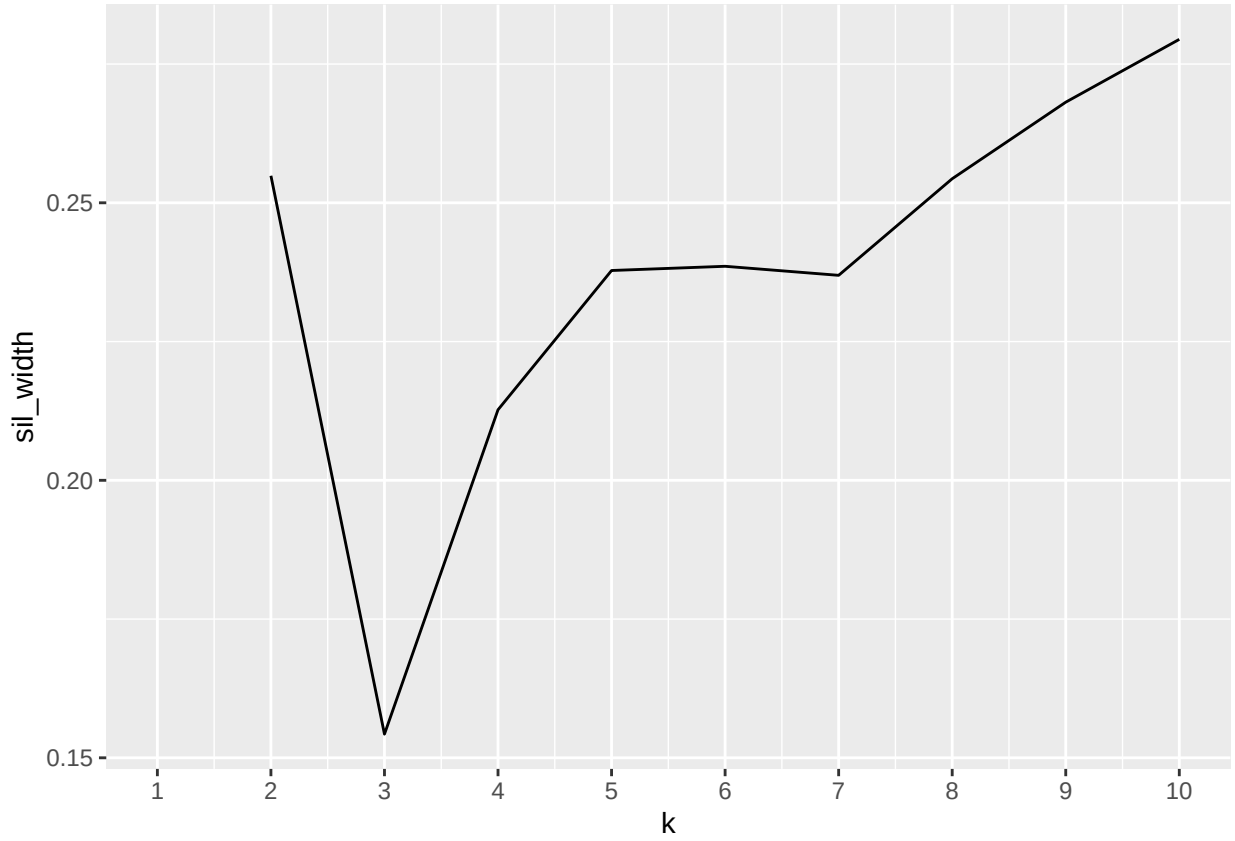

```
ggplot(fullset, aes(author_gender, pages, color = author_gender))+
  geom_bar(stat = "summary", aes(fill = author_gender))+ggtitle("Pages vs. Author Gender")+
  xlab("Author's Gender")+ylab("Pages")+
  scale_y_continuous(breaks=seq(0, 400, 50))
```



K-Means Clustering

For this part of the project, I used PAM to cluster my variables of tempo, duration_ms and danceability. To start, I processed the data by scaling my numeric variables and renaming this data pam_dat. The number of clusters was determined by creating an empty vector to hold mean silhouette widths and then used a for loop where i was equal to values from 1 to 10. The clusters were then created for values 1 to 10 and the silhouette widths were found. After this was created, a ggplot was created to make a line graph to find the lowest value on the graph (this value would be what I choose for the number of clusters). This value was determined to be three clusters since the lowest silhouette width was found at this cluster value.

```
library(cluster)
pam_dat <- fullset%>%select(tempo, duration_ms, danceability)%>%na.omit%>%scale
sil_width<-vector()
for(i in 1:10){
  pam_fit <- pam(pam_dat, k = i)
  sil_width[i]<-pam_fit$silinfo$avg.width
}
ggplot()+geom_line(aes(x=1:10, y=sil_width))+scale_x_continuous(name = "k", breaks = 1:10)
```



After the number of clusters was determined, I clustered the three variables into three clusters. I ran the cluster analysis with PAM and then created a new variable called "cluster" that included the cluster number that the data was assigned to. I then plotted the variables on a ggplot with geom_point and colored the points by cluster. With this graph, I observed that there was some overlap in the clusters and that the clusters were based on the different ranges of tempo, since duration was somewhat constant as tempo increased.

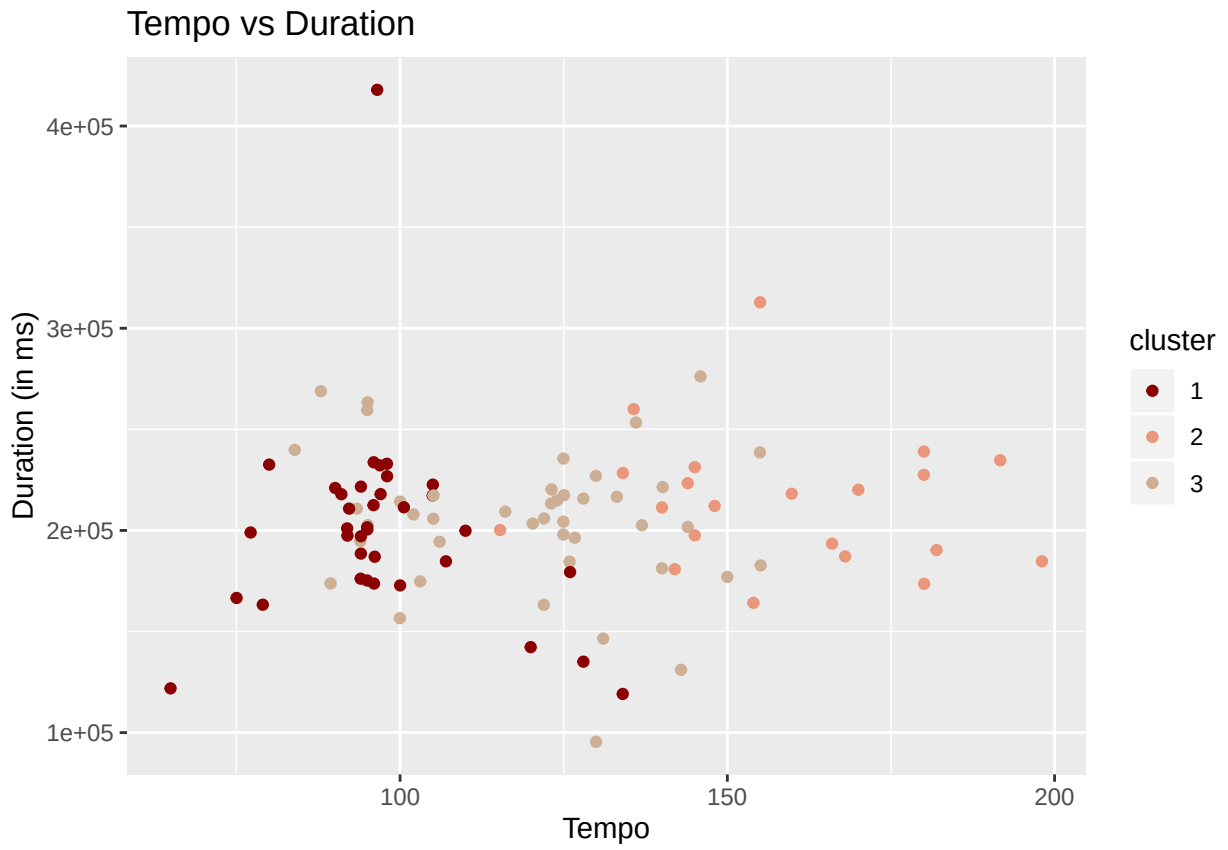
Lastly, I visualized all pairwise combinations of the three variables with ggpairs. Based on this visualization, the highest correlation was between danceability and tempo and the genre of Reggaeton had the highest peak in tempo. In addition, danceability was somewhat constant when comparing across genres, which can be seen with the danceability graph.

```
pam1 <- fullset%>%select(tempo, duration_ms, danceability)%>% na.omit%>%scale%>%pam(3)
pam1
```

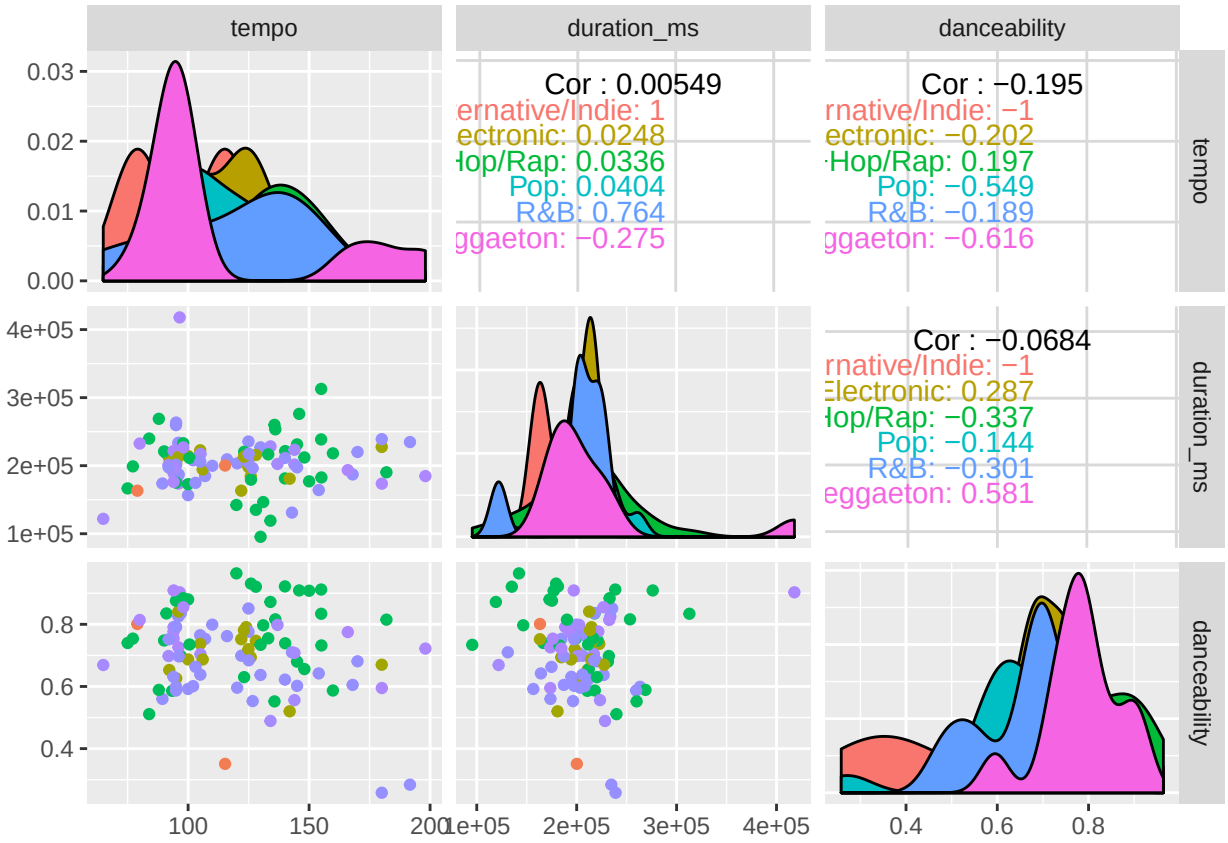
```
## Medoids:
##    ID       tempo duration_ms danceability
## 78 78 -0.86596035 -0.09200134    0.5763326
## 3   3  1.38709691  0.32344167   -0.9877154
## 17 17  0.07253859  0.01782698   -0.1332111
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   1   1   2   3   1   2   3   3   1   3   1   3   3   3   1   1   3   2
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   1   3   1   1   3   3   3   1   3   2   1   3   1   1   2   3   1   3
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   3   3   1   3   1   1   2   2   1   3   3   3   2   2   1   3   1   3
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
```

```
##   3   1   3   3   3   1   1   3   3   2   2   2   2   3   1   1   3   1
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   1   3   3   2   3   1   1   1   2   2   3   1   3   2   3   1   2   1
##  91  92  93  94  95  96  97  98  99 100
##   3   1   1   2   3   3   3   2   2   3
## Objective function:
##   build    swap
## 1.17573 1.17573
##
## Available components:
##  [1] "medoids"    "id.med"     "clustering" "objective"  "isolation"
##  [6] "clusinfo"   "silinfo"    "diss"       "call"       "data"
```

```
pamclust<-fullset%>%select(tempo, duration_ms, danceability, genre)%>%na.omit%>%
  mutate(cluster = as.factor(pam1$clustering))
pamclust%>%ggplot(aes(x=tempo, y=duration_ms, color = cluster))+ geom_point()+
  scale_color_manual(values=c("dark red", "dark salmon", "peachpuff3")) +xlab("Tempo")+
  ylab("Duration (in ms)")+
  ggtitle("Tempo vs Duration ")
```




```
library(GGally)
ggpairs(fullset, columns = c(15,16,5), aes(color = genre))
```



This concludes my exploratory project and summarizes my work researching Goodreads *books* and Spotify *music* data.