Assignment 1

By Simran Talawdekar
400265677

January 31, 2022

Question 1

a)The Variable "visib" is the mean visibility for the day in miles to tenths. The variable "gust" is the maximum wind gust reported for the day in knots to tenths.

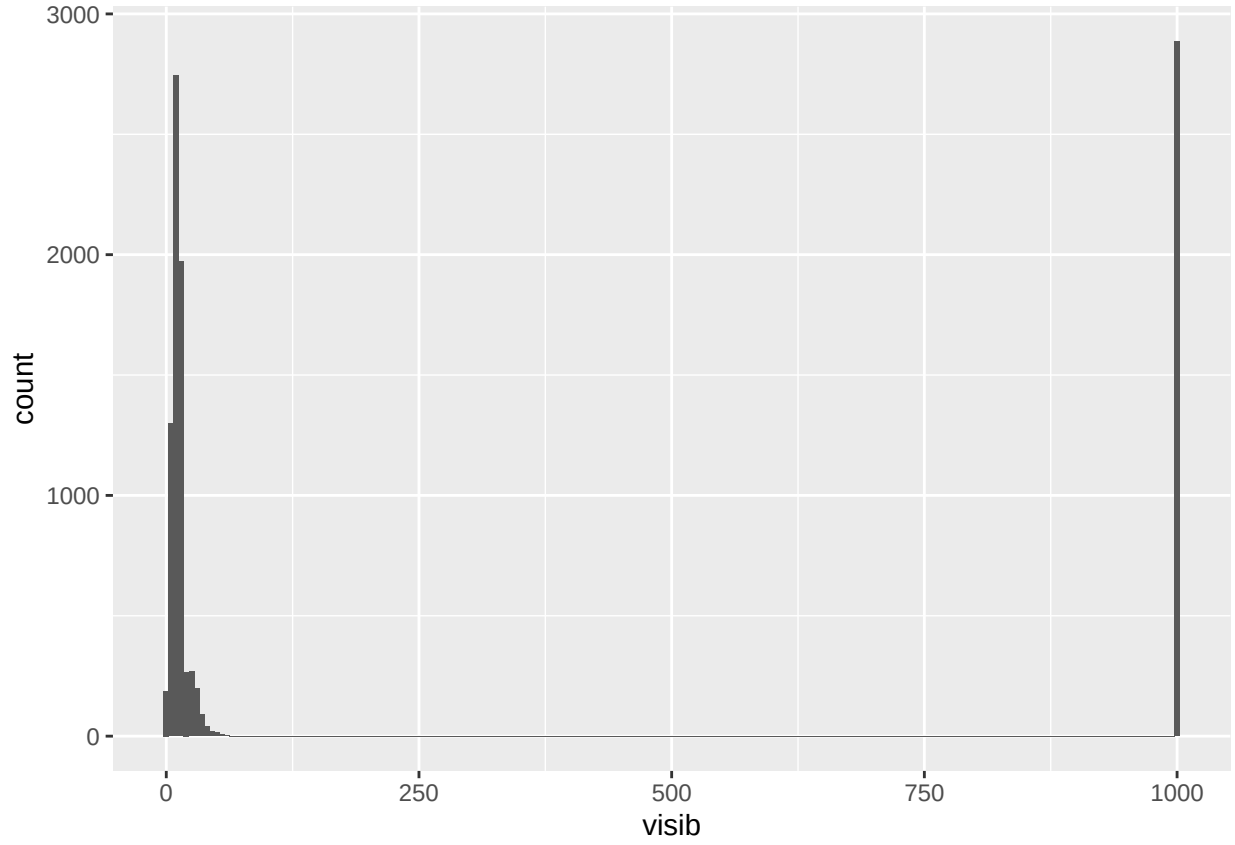
```
library(dplyr)
url <- "https://gist.githubusercontent.com/krisrs1128/3845514e2d5eef57ec3271ea20fdcdb1
noaagsod  <- read.table(url, header = T, sep = ",")
head(noaagsod$visib)
```

```
## [1]   3.0 999.9 999.9 999.9 999.9 999.9
```

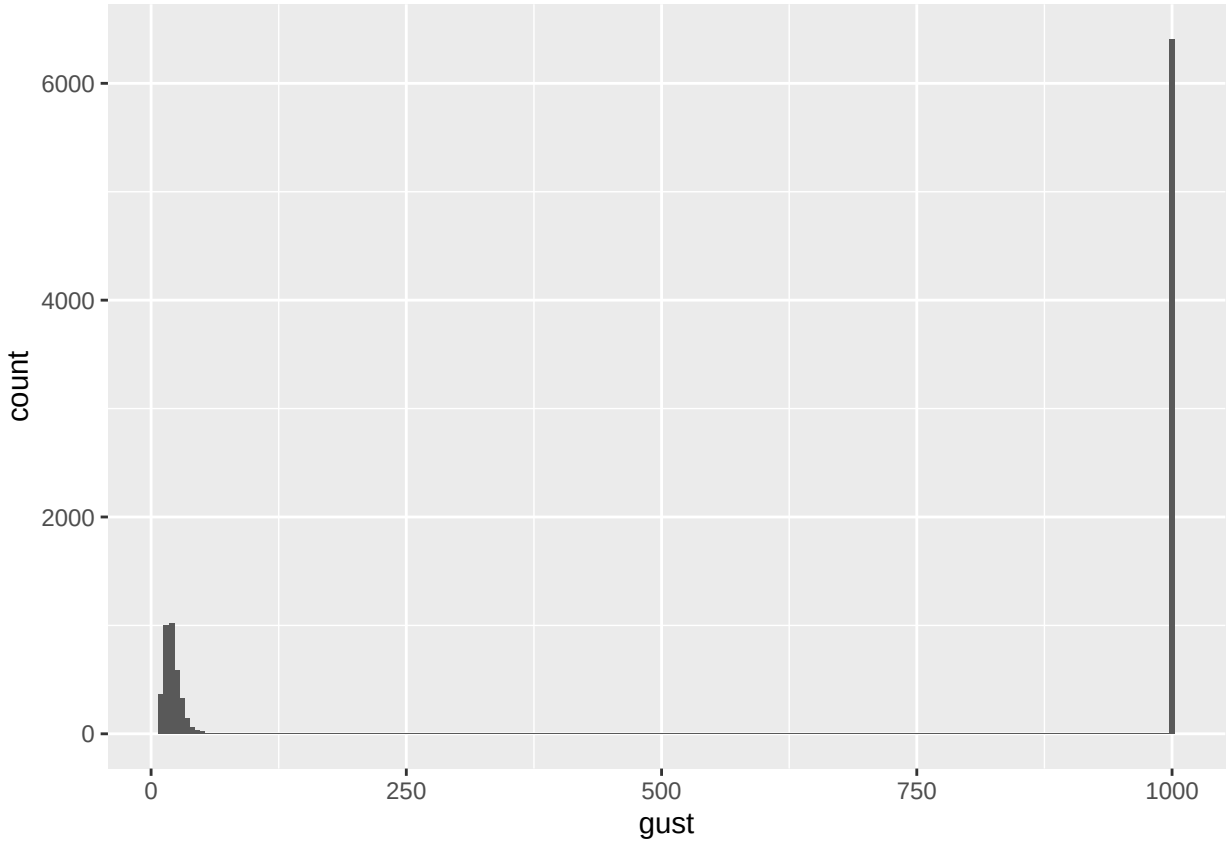
```
head(noaagsod$gust)
```

```
## [1]  17.1 999.9 999.9  18.1  19.6 999.9
```

```
library(ggplot2)
ggplot(data=noaagsod, aes(visib)) +
  geom_histogram(binwidth = 5)
```



```
ggplot(data=noaagsod, aes(gust)) +
  geom_histogram(binwidth=5)
```



~

From these two histograms, we can learn that both histograms are not ideal. Ideal histograms have the graph spread from edge to edge without any edge peaks. For both we see that the graph is stuck at both edges with nothing in the middle. Also we see that the edges have peaks.

b) This is the code used to change any missingness in the data file. Any of the missingness is corrected with NA.Both values are encoded to be the number 999.9. Once we decode it, we can see that both visib and gust don't have values 999.9 but have small decimal numbers.

```
noaagsod$gust[noaagsod$gust == 999.9] <- NA
noaagsod$visib[noaagsod$visib == 999.9] <- NA
head(na.omit(noaagsod$visib))
```

```
## [1] 3.0 8.9 6.2 2.8 8.6 8.8
```

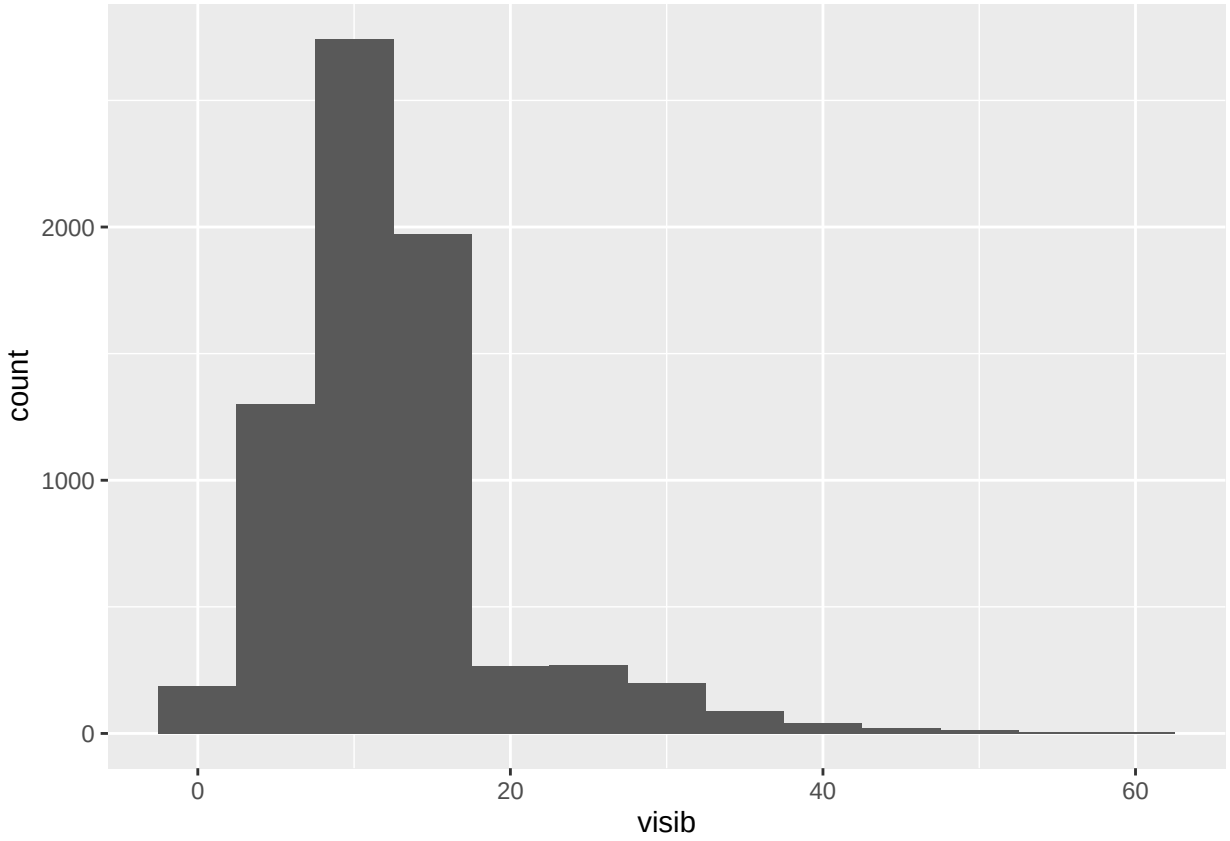
```
head(na.omit(noaagsod$gust))
```

```
## [1] 17.1 18.1 19.6 22.1 22.7 26.0
```

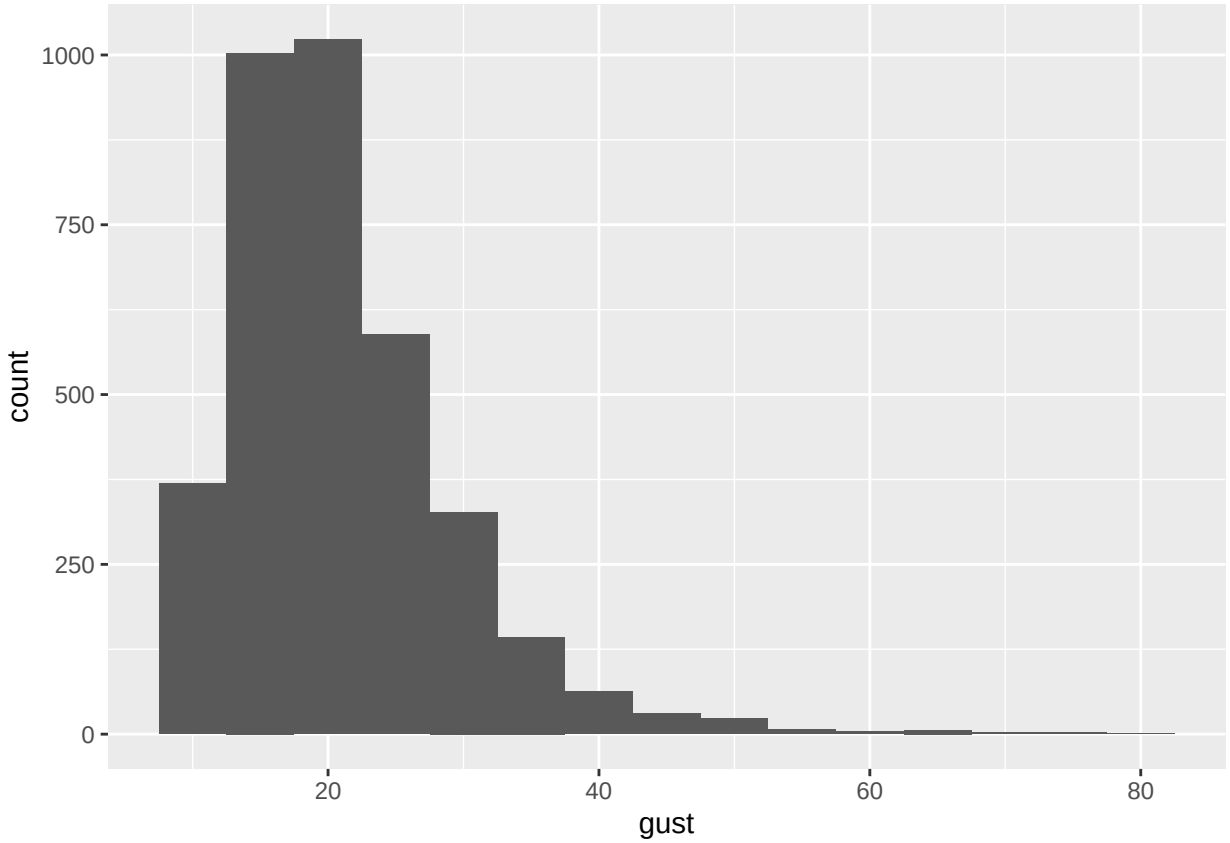
~

c)Now that the numbers are decoded, we see that the graphs are spread out better from edge to edge. Also, we see that there are no high peaks at the edges.

```
library(ggplot2)
ggplot(data=noaagsod, aes(visib)) +
  geom_histogram(binwidth = 5)
```




```
ggplot(data=noaagsod, aes(gust)) +
  geom_histogram(binwidth=5)
```


1000
750
500
250
0
count
20
40
60
80
gust

Question 2

a)The number of observations is 232725. The number of variables is 18.

```
spotify <- read.csv("SpotifyFeatures.csv")
nrow(spotify)
```

```
## [1] 232725
```

```
ncol(spotify)
```

```
## [1] 18
```

b) There are 55951 duplicates in the data set. By dropping all the duplicated tracks, we see that our total number of tracks is now 176774.

```
spotify <-read.csv("SpotifyFeatures.csv")
library(dplyr)
head(dplyr :: select(spotify,track_id))
```

```
##                 track_id
## 1 0BRjO6ga9RKCKjfDqeFgWV
## 2 0BjC1NfoEOOusryehmNudP
## 3 0CoSDzoNIKCRs124s9uTVy
## 4 0Gc6TVm52BwZD07Ki6tIvf
## 5 0IuslXpMROHdEPvSl1fTQK
## 6 0Mf1jKa8eNAf1a4PwTbizj
```

```
sum(duplicated(spotify$track_id))
```

```
## [1] 55951
```

```
head(spotify[!duplicated(spotify$track_id),])
```

```
##   ï..genre        artist_name                       track_name
## 1    Movie     Henri Salvador     C'est beau de faire un Show
## 2    Movie Martin & les fÃ©es Perdu d'avance (par Gad Elmaleh)
## 3    Movie    Joseph Williams   Don't Let Me Be Lonely Tonight
## 4    Movie     Henri Salvador   Dis-moi Monsieur Gordon Cooper
## 5    Movie       Fabien Nataf                         Ouverture
## 6    Movie     Henri Salvador   Le petit souper aux chandelles
##                 track_id popularity acousticness danceability duration_ms
```

```
## 1 0BRjO6ga9RKCKjfDqeFgWV          0        0.611        0.389       99373
## 2 0BjC1NfoEOOusryehmNudP          1        0.246        0.590      137373
## 3 0CoSDzoNIKCRs124s9uTVy          3        0.952        0.663      170267
## 4 0Gc6TVm52BwZD07Ki6tIvf          0        0.703        0.240      152427
## 5 0IuslXpMROHdEPvSl1fTQK          4        0.950        0.331       82625
## 6 0Mf1jKa8eNAf1a4PwTbizj          0        0.749        0.578      160627
##   energy instrumentalness key liveness loudness  mode speechiness   tempo
## 1 0.9100            0.000  C#   0.3460   -1.828 Major      0.0525 166.969
## 2 0.7370            0.000  F#   0.1510   -5.559 Minor      0.0868 174.003
## 3 0.1310            0.000   C   0.1030  -13.879 Minor      0.0362  99.488
## 4 0.3260            0.000  C#   0.0985  -12.178 Major      0.0395 171.758
## 5 0.2250            0.123   F   0.2020  -21.150 Major      0.0456 140.576
## 6 0.0948            0.000  C#   0.1070  -14.970 Major      0.1430  87.479
##   time_signature valence
## 1            4/4   0.814
## 2            4/4   0.816
## 3            5/4   0.368
## 4            4/4   0.227
## 5            4/4   0.390
## 6            4/4   0.358
```

```
sum(!duplicated(spotify$track_id))
```

```
## [1] 176774
```

c) We want to look at each of the types of variables:

- **genre, artist_name, track_name, track_id, key, mode and time_signature are characters -popularity and duration_ms are integers**
- **acousticness, danceability, energy, instrumentalness, liveness, loudness, speechiness, tempo and valence are numeric variables.**

```
str(spotify)
```

```
## 'data.frame':    232725 obs. of  18 variables:
##  $ ï..genre       : chr  "Movie" "Movie" "Movie" "Movie" ...
##  $ artist_name    : chr  "Henri Salvador" "Martin & les fÃ©es" "Joseph Williams" "He
##  $ track_name     : chr  "C'est beau de faire un Show" "Perdu d'avance (par Gad Elma
##  $ track_id       : chr  "0BRjO6ga9RKCKjfDqeFgWV" "0BjC1NfoEOOusryehmNudP" "0CoSDzoN
##  $ popularity     : int  0 1 3 0 4 0 2 15 0 10 ...
##  $ acousticness   : num  0.611 0.246 0.952 0.703 0.95 0.749 0.344 0.939 0.00104 0.31
##  $ danceability   : num  0.389 0.59 0.663 0.24 0.331 0.578 0.703 0.416 0.734 0.598 .
##  $ duration_ms    : int  99373 137373 170267 152427 82625 160627 212293 240067 22620
##  $ energy         : num  0.91 0.737 0.131 0.326 0.225 0.0948 0.27 0.269 0.481 0.705
```

```
##  $ instrumentalness: num  0 0 0 0 0.123 0 0 0 0.00086 0.00125 ...
##  $ key             : chr  "C#" "F#" "C" "C#" ...
##  $ liveness        : num  0.346 0.151 0.103 0.0985 0.202 0.107 0.105 0.113 0.0765 0.3
##  $ loudness        : num  -1.83 -5.56 -13.88 -12.18 -21.15 ...
##  $ mode            : chr  "Major" "Minor" "Minor" "Major" ...
##  $ speechiness     : num  0.0525 0.0868 0.0362 0.0395 0.0456 0.143 0.953 0.0286 0.046
##  $ tempo           : num  167 174 99.5 171.8 140.6 ...
##  $ time_signature  : chr  "4/4" "4/4" "5/4" "4/4" ...
##  $ valence         : num  0.814 0.816 0.368 0.227 0.39 0.358 0.533 0.274 0.765 0.718
```

d) There are 27 different genres in the data.

```
spotify <-read.csv("SpotifyFeatures.csv")
unique(spotify$ï..genre)
```

```
##  [1] "Movie"              "R&B"                "A Capella"
##  [4] "Alternative"        "Country"            "Dance"
##  [7] "Electronic"         "Anime"              "Folk"
## [10] "Blues"              "Opera"              "Hip-Hop"
## [13] "Children's Music"   "Childrenâ\200\231s Music" "Rap"
## [16] "Indie"              "Classical"          "Pop"
## [19] "Reggae"             "Reggaeton"          "Jazz"
## [22] "Rock"               "Ska"                "Comedy"
## [25] "Soul"               "Soundtrack"         "World"
```

```
length(unique(spotify$ï..genre))
```

```
## [1] 27
```

e) The five most popular genres are: Pop, Rap, Rock, Hip-Hop and Dance.

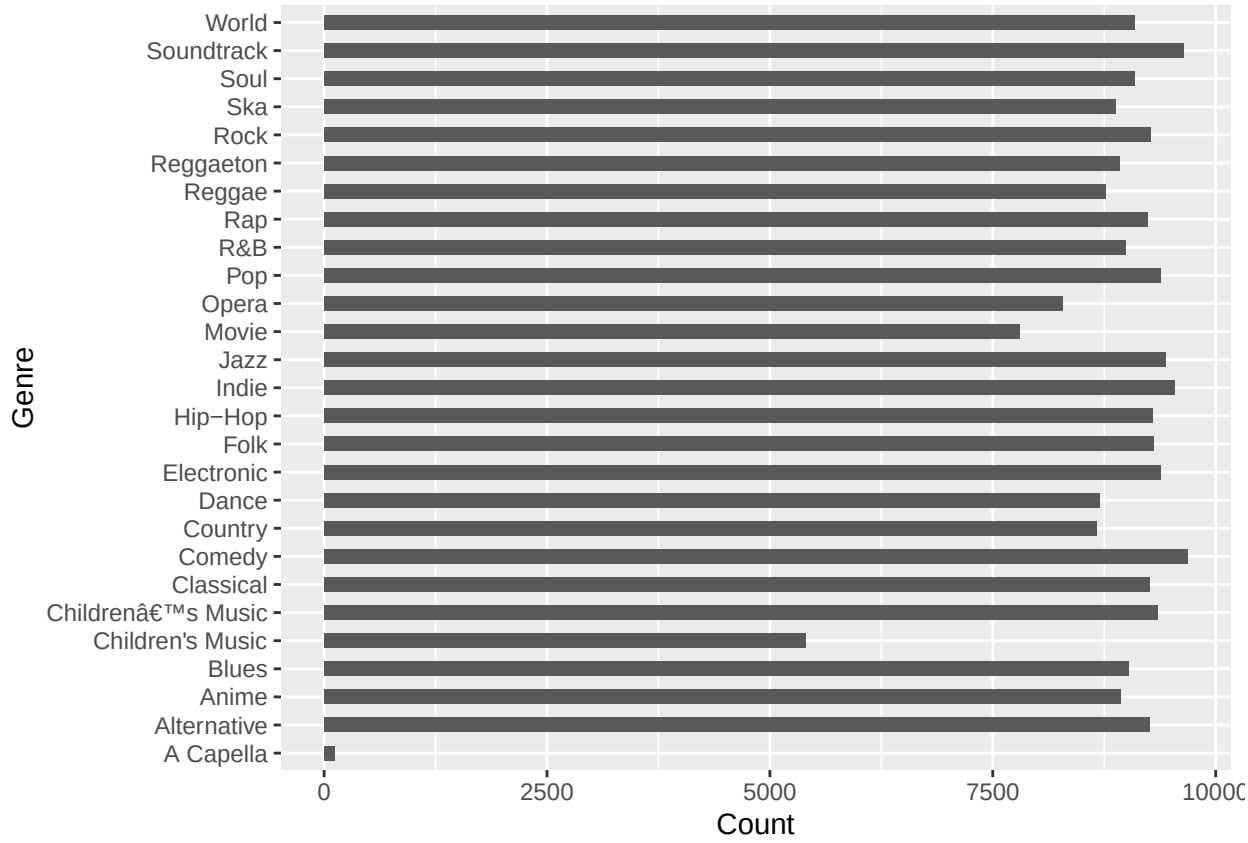
```
#most popular genres
five_most_popular_genre <- dplyr::group_by(spotify,ï..genre)%>%
  summarise(avg_pop=mean(popularity,na.rm=TRUE))%>%
  dplyr::arrange(desc(avg_pop))%>%
  top_n(n=5)%>%
  dplyr::pull(ï..genre)
five_most_popular_genre
```

```
## [1] "Pop"     "Rap"     "Rock"    "Hip-Hop" "Dance"
```

```
spotify1<- dplyr::filter(spotify,ï..genre%in%five_most_popular_genre)
```

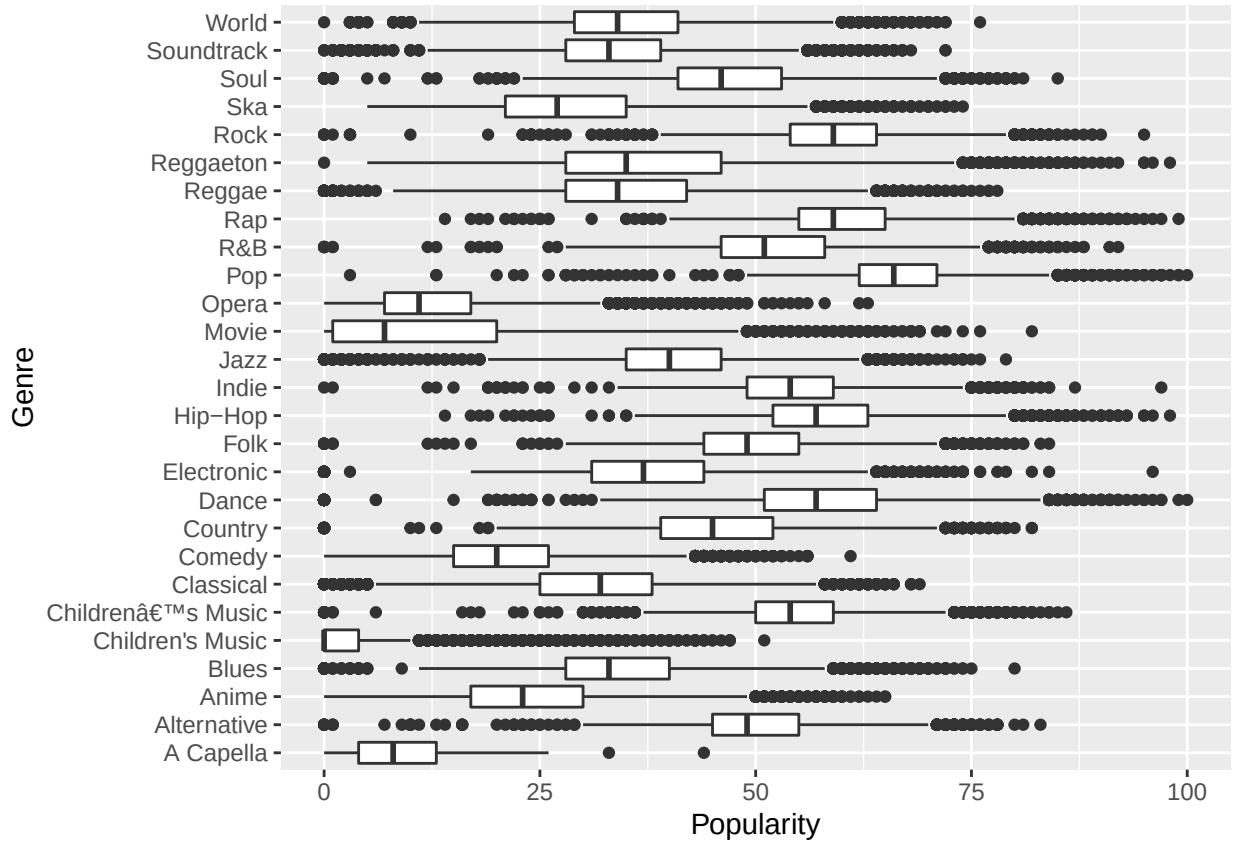
f) First, we see that the data type genre is a character. For this, we need to choose the geom_bar method. This creates a bar graph of the amount of times each genre is used.I decided to swap the x and y variables for the chart to help with interpretation. Here, we see that Soundtrack and Comedy are the most popular genres and A Capella is the lowest. We can also see that the range for most of he genres is 7500-10000.

```
library(ggplot2)
plot1 <- ggplot(data=spotify, aes(y=ï..genre))+
  geom_bar(width = 0.5)+
  ylab("Genre") +
  xlab("Count")
print(plot1)
```



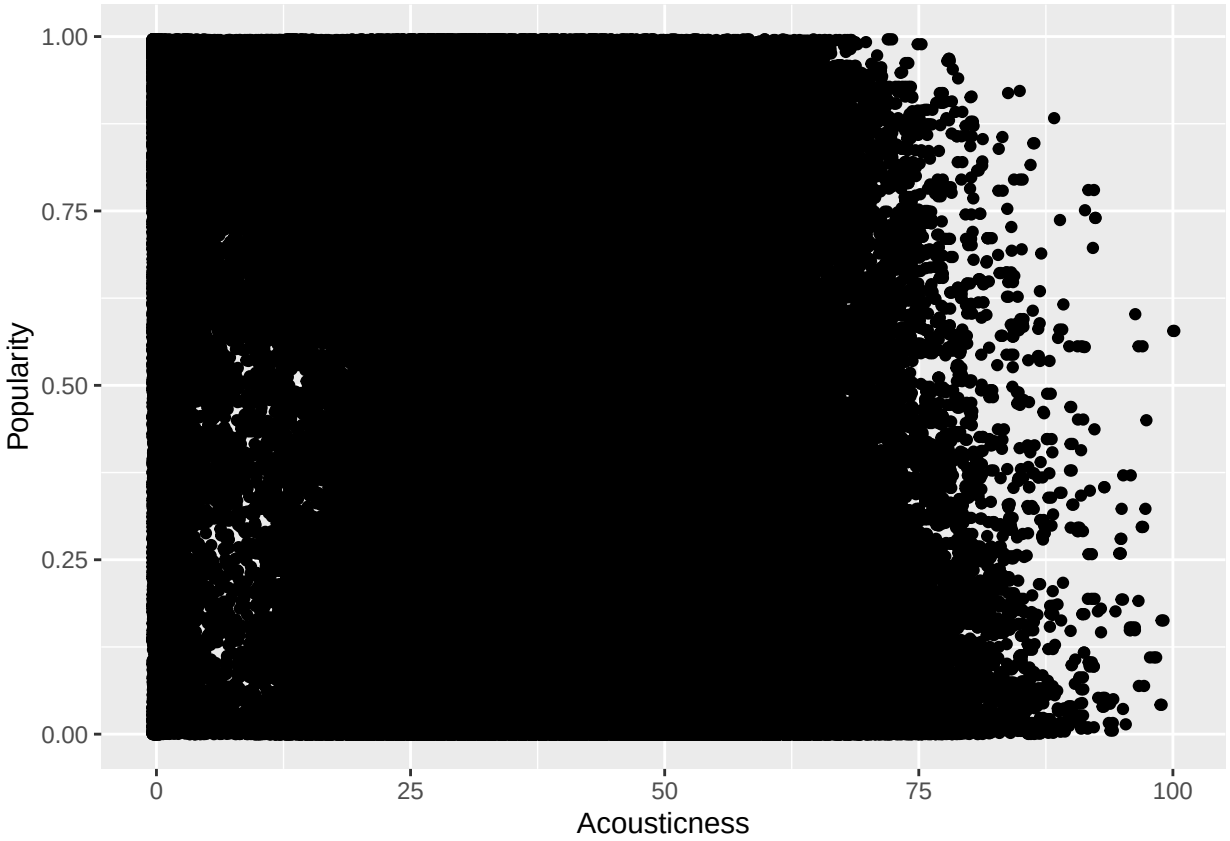
g) For this association, we are using the boxplot graph. We can see there is a mix of normal distribution, positive skew and negative skew. Larger range of scores indicate wider distribution and therefore a more scattered data. We see that most of the genres (aside from Children's Music and A Capella) have large range of scores and so more scattered data. We can also see that there are plenty of outliers (the black dots that are outside of the whiskers). Most of the genres are also in the middle part of the popularity, meaning that popularity is around 50 or less.

```
# plot
library(ggplot2)
ggplot(data = spotify,
       aes(y = ï..genre, x = popularity)) +
  geom_boxplot() +
  ylab("Genre") +
  xlab("Popularity")
```



h)For one integer and one numeric variable, we use scatterplots. This is a very large amount of data, so we have to make sure we do not have overplotting. SO, we use the funtion "jitter" to make the data look less overplotting. Now that we see the graph, we can look at the graph. The plot shows a high correlation between the two variables. This is because the points are condensed. As acousticnes increases (after 70), there is a negative correlation between the two.

```
# plot
library(ggplot2)
ggplot(data = spotify,
       aes(x = popularity, y = acousticness)) +
  geom_point(position = "jitter") +
  xlab("Acousticness") +
  ylab("Popularity")
```



i)Using the same code as part e, we can see the top most popular artists based on the maximum popularity. The top 5 are: Ariana Grande, Post Malone, Daddy Yankee, Ava Mac and Halsey.

```
five_most_popular_artist <- dplyr::group_by(spotify,artist_name)%>%
  summarise(max_pop=max(popularity,na.rm=TRUE))%>%
  dplyr::arrange(desc(max_pop))%>%
  top_n(n=5)%>%
  dplyr::pull(artist_name)
```

```
## Selecting by max_pop
```

```
five_most_popular_artist
```

```
## [1] "Ariana Grande" "Post Malone"   "Daddy Yankee"  "Ava Max"
## [5] "Halsey"        "Marshmello"    "Pedro CapÃ³"   "Sam Smith"
```

```
spotify1<- dplyr::filter(spotify,artist_name%in%five_most_popular_artist)
```

Question 3

```
library(readtext)
library(tidytext)
library(dplyr)
library(data.table)
```

a) I used the journal Domain Generalization by Marginal Transfer Learning (Blanchard et al. 2021) I chose the abstract section of the journal. By using strsplit, we find that the number of words in the file is 1333.

```
library(tm)
library(SnowballC)
library(wordcloud)
library(RColorBrewer)
library(knitrdata)
library(stringr)
```

```
filePath <- "abstractjournal.txt"
text <- readLines(filePath, warn = FALSE)
sapply(strsplit(text, ""), length)
```

```
## [1] 1333
```

```
docs<-VCorpus(VectorSource(text))
inspect(docs)
```

```
## <<VCorpus>>
## Metadata:  corpus specific: 0, document level (indexed): 0
## Content:  documents: 1
##
## [[1]]
## <<PlainTextDocument>>
## Metadata:  7
## Content:  chars: 1333
```

b) Next, we want to prepare the document. To do this, we need to take out any punctuation, hash-tags and clean out unnecessary things.

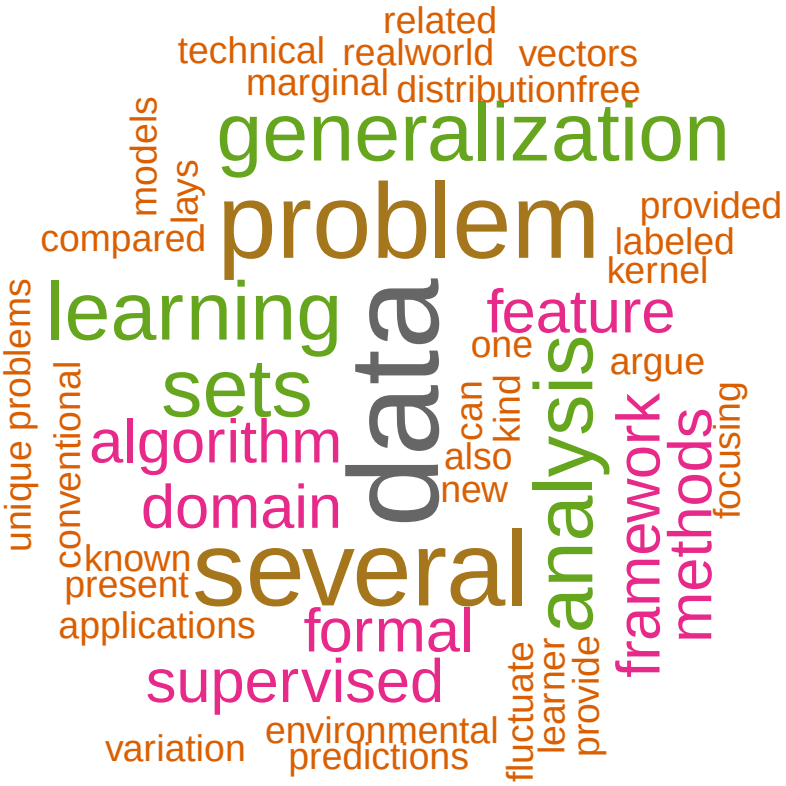
```
docs <- tm_map(docs, content_transformer(tolower))
docs <- tm_map(docs, removePunctuation)
docs <- tm_map(docs, removeWords, stopwords("english"))
docs <- tm_map(docs, stripWhitespace)
```

c) Next, we want to create a word document with the top 10 most frequent words.

```
dtm <- TermDocumentMatrix(docs)
m <- as.matrix(dtm)
v <- sort(rowSums(m),decreasing=TRUE)
d <- data.frame(word = names(v),freq=v)
head(d, 10)
```

```
##                          word freq
## data                     data    5
## problem               problem    4
## several               several    4
## analysis             analysis    3
## generalization generalization    3
## learning             learning    3
## sets                     sets    3
## algorithm           algorithm    2
## domain                 domain    2
## feature               feature    2
```

```
set.seed(100)
wordcloud(words = d$word, freq = d$freq, min.freq = 1,
          max.words=45, random.order=FALSE, rot.per=0.35,
          colors=brewer.pal(8, "Dark2"))
```



d)Using the word cloud, we can see that the word data is the most frequently used word in the file. Some other frequent words are problem, several, generalization, learning and sets. The Most used words are black. As the amount of frequency is reduced, we see that the colour of the words get brighter (from brown to orange to green to pink). Knowing that data is the most common used words, we can predict that the focus of the journal uses data and manipulates it in some type of way. It also makes sense that a frequent word is generalization, since the journal title is called Domain Generalization.

Question 4

My Helpers name is Larissa Padayachee.

References

Blanchard, Gilles, Aniket Anand Deshmukh, Urun Dogan, Gyemin Lee, and Clayton Scott. 2021. "Domain Generalization by Marginal Transfer Learning." *Journal of Machine Learning Research* 22 (2): 1–55. http://jmlr.org/papers/v22/17-679.html.