MA 354: Data Analysis I – Fall 2019
Homework 3:

Complete the following opportunities to use what we've talked about in class. These questions will be graded for correctness, communication and succinctness. Ensure you show your work and explain your logic in a legible and refined submission.

0. **Complete weekly diagnostics.**

1. The goal of this question is to ensure you can simply explain it to yourself in preparation for the next exam and final. Think of this as an opportunity to make something quick you can read that summarizes all of our discussions about this topic that you can study from later.

   (Part A) Map out a decision tree for the hypothesis testing methodologies we've discussed so far this semester. You might want to create this chart in a program other than LaTeX.

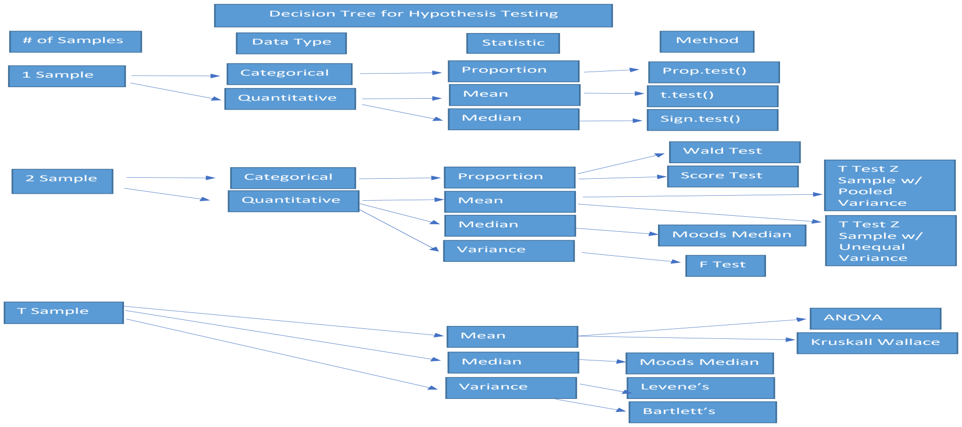


Figure 1: Hypothesis Testing Methodologies Mapped

(Part B) Succinctly describe the difference between ANOVA, Mood's Median Test, and the Kruskal Wallis test.
All three tests aim to tell us if there's evidence that there's a difference across treatments/groups. ANOVA specifically tells us if there's a difference using means. It tests population variances within and across groups/treatments. It assumes normality of the population distribution, equal variance, and independent random sample. Mood's Median and Kruskal Wallis are the nonparametric versions of the ANOVA. Mood's Median test uses medians and does not require normality of the underlying data or equal variances. Kruskal Wallis uses the mean population ranks. It also does not require Gaussian distributed data or equal variances, and can be applied to discrete or continuous data.

(Part C) What is the purpose of post hoc testing in relation to the tests described in Part B.
The purpose of post-hoc testing after omnibus testing is to identify specifically which treatments/subgroups are different after omnibus testing identifies whether or not there is any difference. They require adjustments for multiple comparisons. Tukey HSD is used for ANOVA, Pairwise Mood's Median Test and Pairwise bootstrapping are used for Mood's Median Test, and Dunn's Test is used for Kruskal Wallis. This is where correction for multiple comparisons occurs.

(Part D) Succinctly describe the difference between Pearson correlation, Spearman's rank correlation, and Kendall's Tau b correlation.
All of these correlation methods aim to identify if there is a relationship between 2 quantitative variables. Pearson's correlation identifies the linear relationship and assumes a continuous distribution with no outliers, both variables are Gaussian distributed, the pairs of observations are independent, and the sample is generalized. Kendall's Tau and Spearman's Rank identify if there is a monotone bivariate relationship. Kendall's Tau is always lower than Spearman's rank and is the rank-based correlation. Spearman's rank is robust to outliers.

2. **(Twitter)** The data consist of a subset of 248,915 tweets, in English, excluding retweets and quotes, that contain "self-injurious behavior", "self injurious behaviour", "non suicidal self injury", or "self harm" from June 1, 2018, through May 31, 2019 obtained from the premium Twitter API.

   We noticed a significant spike in Twitter activity in and around the news that Instagram would ban graphic images of self-harm (February 7, 2019). By far, the greatest volume of Twitter activity occurred in the few days after the imposition of the ban.

   Of interest to us is how this event changed the discourse about self-harm and Instagram on Twitter. Below, we provide 5,875 tweets that mention Instagram, but not Facebook or Twitter, from the full dataset. You'll need to download the Excel file from Moodle and place Q2Tweets.csv into your working folder, and erase `eval=FALSE`.

```
> tweets<-read.csv(file="/Users/canaan/Q2Tweets.csv",header = TRUE, sep = ","
+                  ,stringsAsFactors = FALSE)
```

   (a) Create columns that contain the count of sentiment providing words. You can have `R` automatically count sentiment provided words using the "syuzhet" package (Jockers, 2015) for `R` using the code below. You might find toggling `eval=FALSE` helpful as you complete the rest of the assignment. The `get_nrc_sentiment()` function takes a couple of minutes to run as `R` "reads" the 5,875 tweets.

   (b) Create a new column called "sentwords" that counts the number of sentiment providing words.

   (c) Remove tweets that contain no sentiment; e.g., remove rows where "sentwords=0".

   (d) Calculate the percent-sentiments by dividing all the sentiment counts in each row by the "sentwords" value for that row.

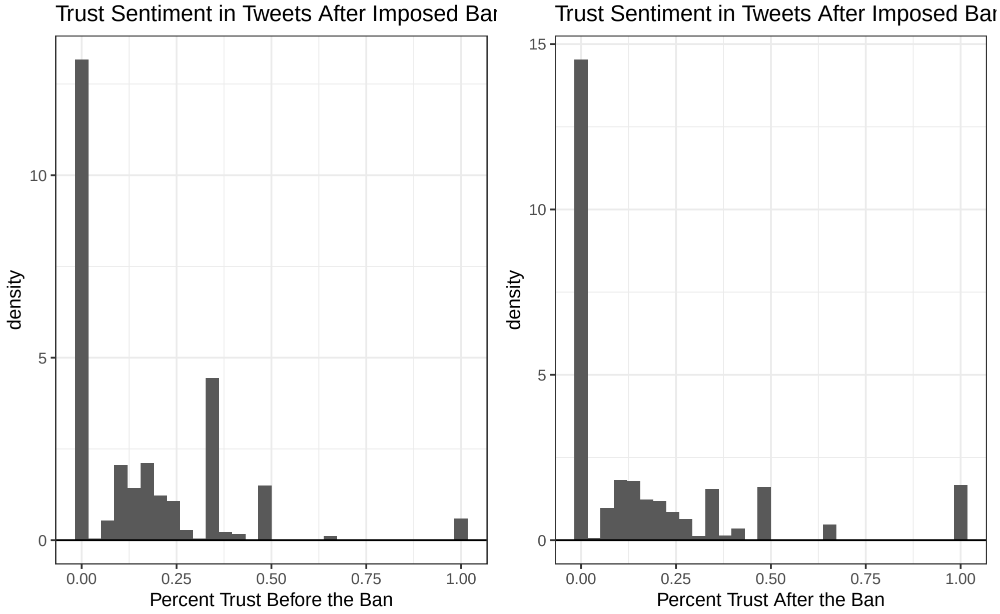
   (e) Is there a difference in trust-sentiment content of tweets that mention Instagram before and after the imposed ban? You can use the "group" to see whether each tweet occurred before, after or the day of the ban. Your answer should include both graphical exploration and numerical test.

   Remark: Interestingly, in the full data set (which takes hours and hours and hours to run), we see a large increase in anticipation for tweets mentioning social media more generally, e.g., tweets mentioning Twitter, Facebook, or Instagram. We take this to mean that users are showing anticipation that other platforms would follow Instagram's lead.
   Solution:

```
> library(syuzhet)
> library(tm)
> #remove keywords that have negative sentiments that are just topical
> sentText<-removeWords(tweets$text,c("ban","harm","injury","injurious",
+                                     "suicide","suicidal", "hurt","remove"))
> #get sentiment of all other left over words for each tweet
> sentText=iconv(sentText, from="UTF-8", to="ASCII", sub="")
> sentiment<-get_nrc_sentiment(as.character(sentText))#it's reading 5,875 tweets!
> #add sentiment columns to tweets
> tweets<-cbind(tweets,sentiment)
> #create new column "sentwords"
> tweets$sentwords<-tweets$anger+ tweets$anticipation+ tweets$disgust+ tweets$fear+ tweets$joy+ twe
> #remove tweets that contain no sentiment
> tweets<- subset(tweets, sentwords!=0)
> #calculate percent sentiments
> tweets$percent.anger<-tweets$anger/tweets$sentwords
> tweets$percent.anticipation<-tweets$anticipation/tweets$sentwords
> tweets$percent.disgust<-tweets$disgust/tweets$sentwords
> tweets$percent.fear<-tweets$fear/tweets$sentwords
> tweets$percent.joy<-tweets$joy/tweets$sentwords
> tweets$percent.sadness<-tweets$sadness/tweets$sentwords
> tweets$percent.surprise<-tweets$surprise/tweets$sentwords
> tweets$percent.trust<-tweets$trust/tweets$sentwords
> tweets$percent.negative<-tweets$negative/tweets$sentwords
> tweets$percent.positive<-tweets$positive/tweets$sentwords
> #diff in trust sentiment
> before<- subset(tweets, group=="Before")
> after<- subset(tweets, group=="After")
```

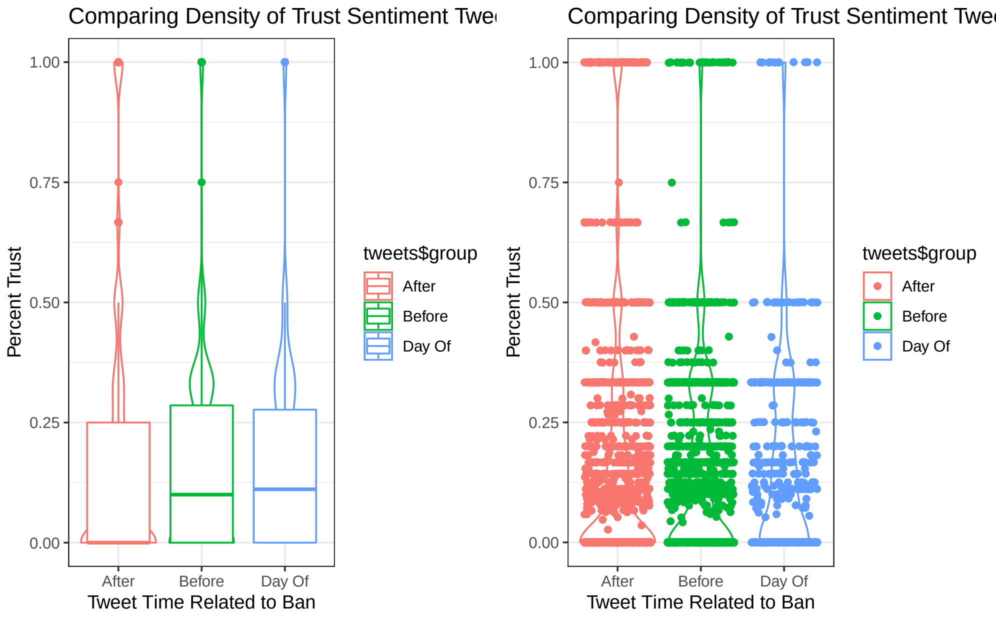
Wickham (2016) Auguie (2017)

```
> library(ggplot2)
> library(gridExtra)
> before.ggdat<- data.frame(before$trust)
> after.ggdat<- data.frame(after$trust)
> g1<-ggplot(data=before.ggdat,aes(x=before$percent.trust))+
+   geom_histogram(aes(y=..density..))+
+   theme_bw()+
+   xlab("Percent Trust Before the Ban")+
+   ggtitle("Trust Sentiment in Tweets After Imposed Ban")+
+   geom_hline(yintercept = 0)
> g2<-ggplot(data=after.ggdat,aes(x=after$percent.trust))+
+   geom_histogram(aes(y=..density..))+
+   theme_bw()+
+   xlab("Percent Trust After the Ban")+
+   ggtitle("Trust Sentiment in Tweets After Imposed Ban")+
+   geom_hline(yintercept = 0)
> grid.arrange(g1,g2,ncol=2)
```



We plot the data to determine whether or not the trust-sentiment content of tweets that mention Instagram before and after the imposed ban. We see that the data is not normal for either sentiment, although trust sentiment after the ban is slightly more spread than before the ban. Since the ggplot isn't particularly telling about the trust sentiment before and after the ban, we can use further statistical analysis.

```
> #violin plot comparing after and before
> ggdat<-data.frame(tweets$percent.trust, tweets$group)
> g1<-ggplot(data=ggdat,aes(x=tweets$group, y=tweets$percent.trust, color=tweets$group))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Tweet Time Related to Ban")+
+   ylab("Percent Trust")+
+   ggtitle("Comparing Density of Trust Sentiment Tweets Across Time")+
+   theme_bw()
> g2<-ggplot(data=ggdat,aes(x=tweets$group, y=tweets$percent.trust, color=tweets$group))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Tweet Time Related to Ban")+
+   ylab("Percent Trust")+
+   ggtitle("Comparing Density of Trust Sentiment Tweets Across Time")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



This pair of plots identifies that there is a higher median of trust sentiment in tweets before the ban compared to after the ban. We see this information specifically from the boxplot. The boxplot also points us to the fact that the median trust sentiment for data after the ban is different from the before and day of median, which are very similar. The jitter plot tells us that there are many observations of trust sentiment at 0.

```
> #install.packages('qqplotr')
> library("qqplotr")
> ggdat<-data.frame(Trust=tweets$percent.trust, Group=tweets$group)
> ggplot(data=ggdat,aes(sample="Trust"))+
+   stat_qq_band(alpha=0.25) +
+   stat_qq_line() +
+   stat_qq_point() +
+   theme_bw()+
+   xlab("Gaussian Quantiles")+
+   ylab("Sample Quantiles")+
+   facet_grid(. ~ Group)
```

Hervé (2019) Mangiafico (2019)

```
> #difference in medians, perform mood test
> library(RVAideMemoire)
> mood.medtest(percent.trust~group,data=tweets)

        Mood's median test

data:  percent.trust by group
X-squared = 20.122, df = 2, p-value = 4.27e-05

> #pairwise to find particular difference
> library(rcompanion)

> #benjamini hochberg
> PTBH<-pairwiseMedianTest(percent.trust~group,
+                         data   = tweets,
+                         method = "BH")
>

> #compact Letter display for grouping where no significant differences
> cldList(p.adjust ~ Comparison,
+         data = PTBH,
+         threshold = 0.05)

   Group Letter MonoLetter
1  After      a          a
2 Before      b           b
3  DayOf      b           b

> #bonferroni
> PTB<-pairwiseMedianTest(percent.trust~group,
+                         data   = tweets,
+                         method = "bonferroni")

> #CLD
> cldList(p.adjust ~ Comparison,
+         data = PTB,
+         threshold = 0.05)

   Group Letter MonoLetter
1  After      a          a
2 Before      b           b
3  DayOf      b           b
```

Our null hypothesis is that there is not a significant difference in median trust sentiment content of tweets that mention Instagram before and after the ban. Our alternative hypothesis is that there is a significant difference in median trust sentiment content of tweets that mention Instagram before and after the ban. Since the plots of the data do not point towards normality, we will use a mood's median test which is a nonparametric alternative to the ANOVA.

The Mood's median test is a chi-squared test that tests for differences across medians. Our chi-squared variable is 20.122 and our p-value is essentially zero. Since our p value is less than 0.05, we have sufficient evidence to reject the null.

I performed a post-hoc test, specifically the pairwise Median test to adjust the P value using Benjamini Hochberg and Bonferroni methods. Benjamini Hochberg adjusts the p-value to 0.001675 and the Bonferroni method adjusts the p-value to 0.001675 as well.

3. Adams Upper Orchard in Grove City, UT is the location of a study on the Velvet Longhorned Beetle throughout the summer of 2018 (June 4th-August 6th). It is currently hypothesized that the type of lure, specifically the presence or amount of a pheromone, will affect the number of beetles captured in traps and that more beetles will be captured with lures that contain a larger amount of the pheromone. The orchard is broken into 15 replicates (labeled R1-R15) each with four traps evenly spaced 10 meters apart. Each trap (labeled T1-T4, per replicate) is a black paneled intercept trap with a collection cup filled with antifreeze.

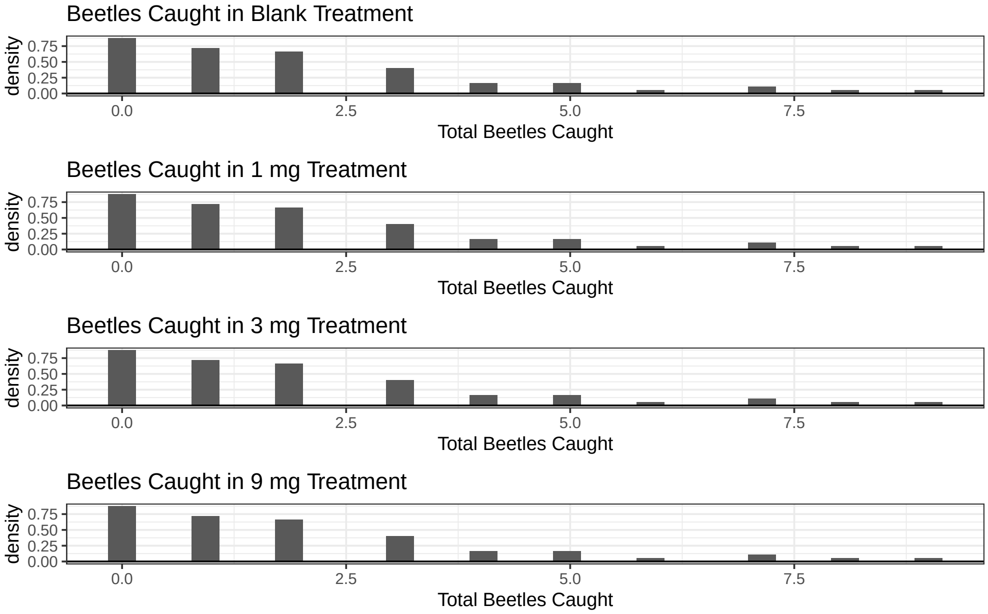
   Research is based on three pheromone treatments (1 mg, 3 mg, 9 mg) and one control or blank treatment. Every replicate has one trap/lure with one of the four treatments. The lures are randomly assigned to trap locations within a replicate and are reassigned randomly each week. Every week the number of beetles in each trap, its treatment, and replicate are recorded and entered into a spreadsheet. After the collection and the quantities are recorded the specimens were sent to the Otis lab for confirmation of sex, expected to be completed in Winter 2018.

   Data regarding the number of beetles captured that was used for the following analysis was collected twice in June, on the 12th and 19th, and formatted in an Excel spreadsheet. The dataset includes columns listing the recording date, replicate and trap number, lure type, and the number of beetles collected. Overall the research seeks to ask if there is a difference between the mean number of beetles caught per trap, per week across the four different treatment types.

   Provide the researcher information on the how the number of beetles captured varies by lure type.

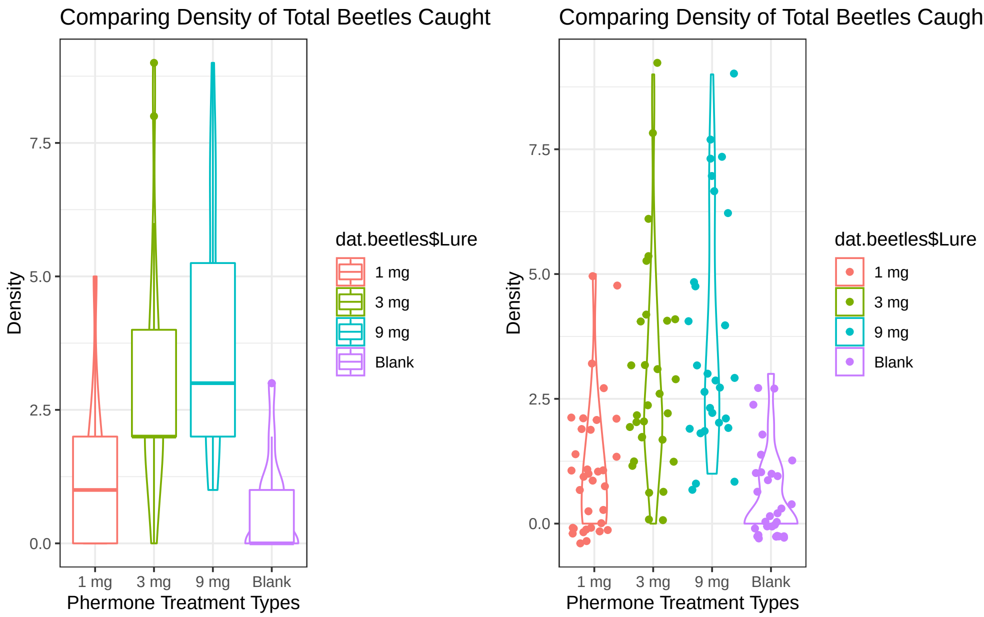
```
> #load data
> dat.beetles<-read.csv(file = "https://cipolli.com/students/data/beetles.txt",
+                        header = TRUE,sep = ",")
> #want to identify if there is a difference between the mean number of beetles caught per trap per

> lureblank<-subset(dat.beetles, Lure="Blank")
> onemgblank<-subset(dat.beetles, Lure="1 mg")
> threemgblank<-subset(dat.beetles, Lure="3 mg")
> ninemgblank<-subset(dat.beetles, Lure="9 mg")
> lureblank.ggdat<-data.frame(lureblank$Total)
> onemgblank.ggdat<-data.frame(onemgblank$Total)
> threemgblank.ggdat<-data.frame(threemgblank$Total)
> ninemgblank.ggdat<-data.frame(ninemgblank$Total)
> g1<-ggplot(data=lureblank.ggdat,aes(x=lureblank$Total))+
+   geom_histogram(aes(y=..density..))+
+   theme_bw()+
+   xlab("Total Beetles Caught")+
+   ggtitle("Beetles Caught in Blank Treatment")+
+   geom_hline(yintercept = 0)
> g2<-ggplot(data=onemgblank.ggdat,aes(x=onemgblank$Total))+
+   geom_histogram(aes(y=..density..))+
+   theme_bw()+
+   xlab("Total Beetles Caught")+
+   ggtitle("Beetles Caught in 1 mg Treatment")+
+   geom_hline(yintercept = 0)
> g3<-ggplot(data=threemgblank.ggdat,aes(x=threemgblank$Total))+
+   geom_histogram(aes(y=..density..))+
+   theme_bw()+
+   xlab("Total Beetles Caught")+
+   ggtitle("Beetles Caught in 3 mg Treatment")+
+   geom_hline(yintercept = 0)
> g4<-ggplot(data=ninemgblank.ggdat,aes(x=ninemgblank$Total))+
+   geom_histogram(aes(y=..density..))+
+   theme_bw()+
+   xlab("Total Beetles Caught")+
+   ggtitle("Beetles Caught in 9 mg Treatment")+
+   geom_hline(yintercept = 0)
> grid.arrange(g1,g2,g3,g4,nrow=4)
```



After taking a preliminary look at the data, we see that the data for all treatments is skewed. Therefore, to proceed with our omnibus test we will use a mood's median test since we are specifically interested in averages and ANOVA requires normality.

```
> #violin plot comparing treatment types
> ggdat<-data.frame(dat.beetles$Lure, dat.beetles$Total)
> g1<-ggplot(data=ggdat,aes(dat.beetles$Lure, y=dat.beetles$Total, color=dat.beetles$Lure))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Phermone Treatment Types")+
+   ylab("Density")+
+   ggtitle("Comparing Density of Total Beetles Caught Across Phermone Treatment")+
+   theme_bw()
> g2<-ggplot(data=ggdat,aes(dat.beetles$Lure, y=dat.beetles$Total, color=dat.beetles$Lure))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Phermone Treatment Types")+
+   ylab("Density")+
+   ggtitle("Comparing Density of Total Beetles Caught Across Phermone Treatment")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



This pair of plots gives a more specific picture about the data. We see that the control pheromone and the 1 mg pheromone have a smaller median of total beetles caught compared to the 3 mg and 9 mg pheromone levels. The jitter plot shows how the observations vary.

To proceed with our omnibus test, we need to check some assumptions. We assume that these are independent random samples that are representative.

```
> ggdat<-data.frame(Total=dat.beetles$Total, Group=dat.beetles$Lure)
> ggplot(data=ggdat,aes(sample="Total"))+
+   stat_qq_band(alpha=0.25) +
+   stat_qq_line() +
+   stat_qq_point() +
+   theme_bw()+
+   xlab("Gaussian Quantiles")+
+   ylab("Sample Quantiles")+
+   facet_grid(. ~ Group)
```

```
> bartlett.test(Total~Lure, data=dat.beetles)

        Bartlett test of homogeneity of variances

data:  Total by Lure
Bartlett's K-squared = 28.975, df = 3, p-value = 2.267e-06
```

For good measure and practice, I run a Bartlett test to see if the population variances are significantly different. Since the p-value is essentially zero, we can conclude that the population variances are not homogenous.

```
> library(RVAideMemoire)
> mood.medtest(Total~Lure,data=dat.beetles)
```

```
        Mood's median test

data:  Total by Lure
p-value = 8.557e-07

> #pairwise to find particular difference
> library(rcompanion)

> #benjamini hochberg
> PTBH<-pairwiseMedianTest(Total~Lure,
+                         data   = dat.beetles,
+                         method = "BH")

> #compact Letter display for grouping where no significant differences
> cldList(p.adjust ~ Comparison,
+         data = PTBH,
+         threshold = 0.05)

  Group Letter MonoLetter
1   1mg      a         a
2   3mg      b          b
3   9mg      b          b
4 Blank      a         a

> #bonferroni
> PTB<- pairwiseMedianTest(Total~Lure,
+                         data   = dat.beetles,
+                         method = "bonferroni")

> #CLD
> cldList(p.adjust ~ Comparison,
+         data = PTB,
+         threshold = 0.05)

  Group Letter MonoLetter
1   1mg      a         a
2   3mg      b          b
3   9mg      b          b
4 Blank      a         a
```

Our null hypothesis is that there is not a difference between the median number of beetles caught per trap, per week across the four different treatment types. Our alternative hypothesis is that there is a difference between the median number of beetles caught per trap, per week across the four different treatment types. We checked our assumptions before performing the test. The p value of the Mood's median test is essentially zero, meaning that we reject our null hypothesis. We proceed by performing our post-hoc test for Mood's median test - a Pairwise Median Test. When we run our pairwise median test using the Benjamini Hochberg as well as the Bonferroni approach, we find that the 1 mg pheromone treatment and the blank pheromone treatment are not significantly different, but they are both significantly different from the 3 mg and 9 mg pheromone treatments.

4. **(Spotify)** Below you will load and summarize a dataset containing the albums of a band called The Front Bottoms. The data includes the following measurements provided by spotify:

    - artist
    - track
    - danceability

- energy
- key
- loudness
- mode
- speechiness
- acousticness
- instrumentalness
- liveness
- valence
- tempo
- duration_ms
- time_signature

My wife, who is taking me to see The Front Bottoms for my birthday for the second time in Albany this year (we've seen them a couple other times), has learned to enjoy the most recent albumn Ann but still doesn't love the older albumns like their self-titled album. I'm curious as to why she might not like the earlier albums.

Create a spotify application by following the instructions here.

- You can use whatever "Application name" you'd like
- You don't need to specify a "Website"
- Specify your "Redirect URIs" as http://localhost:1410/

Once you've created an application you should see the Client ID and Client Secret on the application home page.

Once you've created the application, you can collect data about The Front Bottom albums as follows. Use this data to explore the differences between the more recent and earlier albums from The Front Bottoms.

Below is an outline of the code you'll need to use to access the playlist I created containing the discography of The Front Bottoms using the "Rspotify" package (Dantas, 2019) for `R`. You should only need to change the `spotifyOAuth()` arguments.

Remark: When you compile your pdf, you may be prompted to copy and paste a url to authenticate your Spotify user credentials. Cheng et al. (2019)

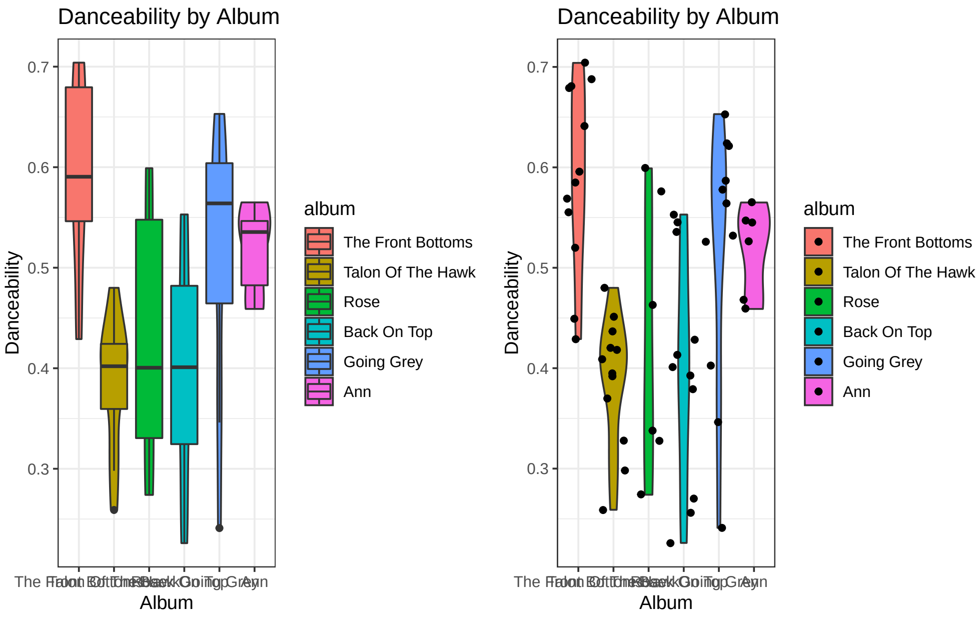
```
> library("Rspotify")
> #install.packages("httpuv")
> library(httpuv)
> #place your application information below.
> keys <- spotifyOAuth("R-Homework",  # change
+                      "722ef6d26b0b4a8485eefd472746b350",
+                      "76f67392a65f45dd92d5e6d984b2523d")
> # grab songs from by front bottoms playlist
> songs<-getPlaylistSongs("spotify","6r0vro3m8iOCJHypfzI4gM",token=keys)
> ##Download features about the tracks
> features<-data.frame()
> for(i in 1:nrow(songs)){#for each song
+  features<-rbind(features,getFeatures(songs$id[i],token=keys))
+ }
> ##Data frame for saving features
> df<-data.frame(artist=songs$artist,track=songs$tracks,album=songs$album,
+               danceability=features$danceability,
+               energy=features$energy,
+               key=features$key,
+               loudness=features$loudness,
+               mode=features$mode,
+               speechiness=features$speechiness,
+               acousticness=features$acousticness,
+               instrumentalness=features$instrumentalness,
+               liveness=features$liveness,
+               valence=features$valence,
+               tempo=features$tempo,
+               duration_ms=features$duration_ms,
+               time_signature=features$time_signature
+             )
>
> # ann<-subset(df, album=="Ann")
> # not.ann<-subset(df, album!="Ann")
```

I created the Spotify API and authorized myself to use the data. Since we are interested in examining the different albums over time, it is important to sort the albums by release date. Wickham (2017) Wickham (2018)

```
> #order albums by order of release date
> library(tidyverse)
> library(scales)
> df$album<-factor(df$album,levels = c("The Front Bottoms",
+                                      "Talon Of The Hawk",
+                                      "Rose",
+                                      "Back On Top",
+                                      "Going Grey", "Ann"))
> # df$album<-as.factor(df$album)
> # str(df)
```

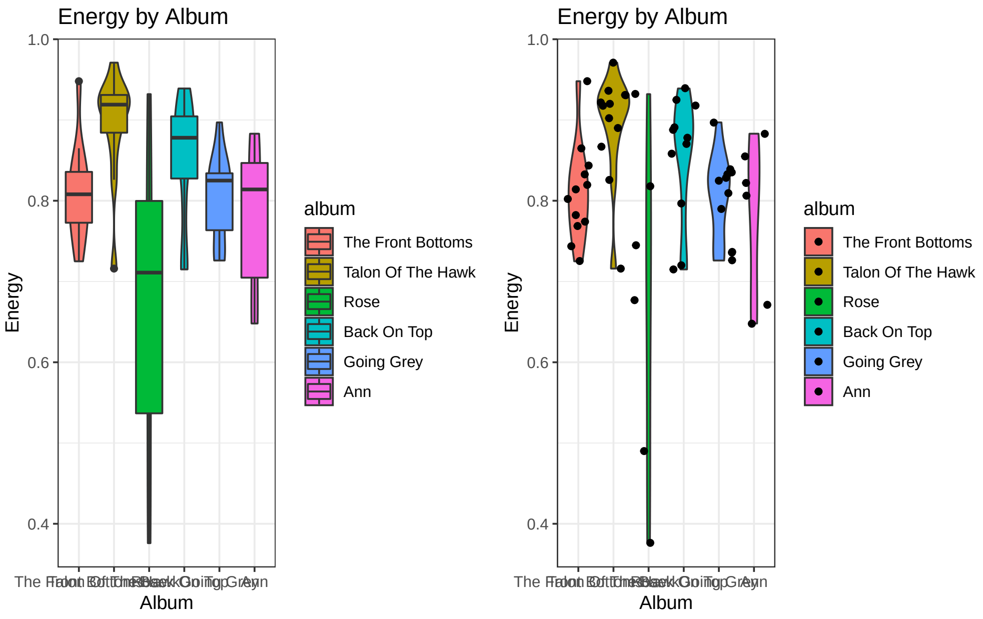
In sorting the albums chronologically by release date, this allows us to produce figures that display change over time/change over album.

```
> ggdat<-data.frame(album=df$album, y=df$danceability)
> g1<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Album")+
+   ylab("Danceability")+
+   ggtitle("Danceability by Album")+
+   theme_bw()
> g2<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Album")+
+   ylab("Danceability")+
+   ggtitle("Danceability by Album")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



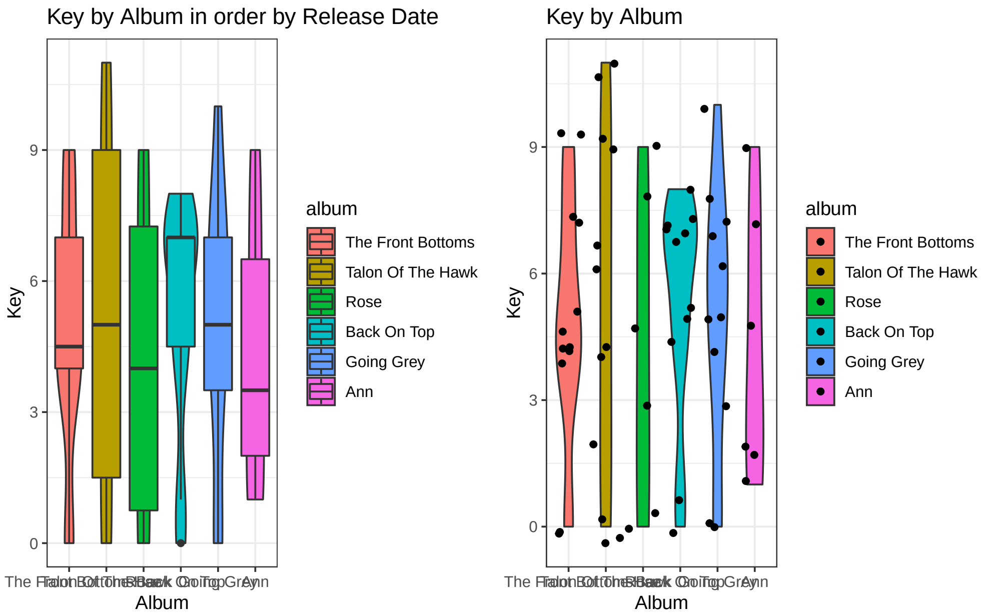
At first glance, the medians of danceability by album seem to be different. In particular, the medians of Ann, Going Grey, and The Front Bottoms seem to be different from the 3 other albums.

```
> ggdat<-data.frame(album=df$album, y=df$energy)
> g1<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Album")+
+   ylab("Energy")+
+   ggtitle("Energy by Album")+
+   theme_bw()
> g2<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Album")+
+   ylab("Energy")+
+   ggtitle("Energy by Album")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



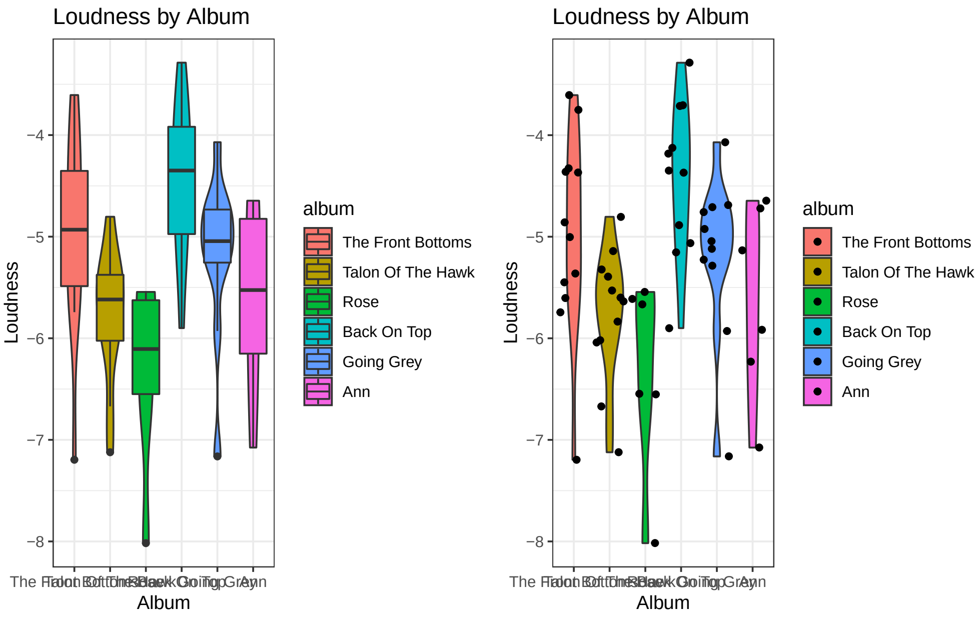
Energy levels seem to be similar, yet differ across albums. Rose in particular also has a very large spread, which lends us to want to further investigate energy.

```
> ggdat<-data.frame(album=df$album, y=df$key)
> g1<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Album")+
+   ylab("Key")+
+   ggtitle("Key by Album in order by Release Date")+
+   theme_bw()
> g2<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Album")+
+   ylab("Key")+
+   ggtitle("Key by Album")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



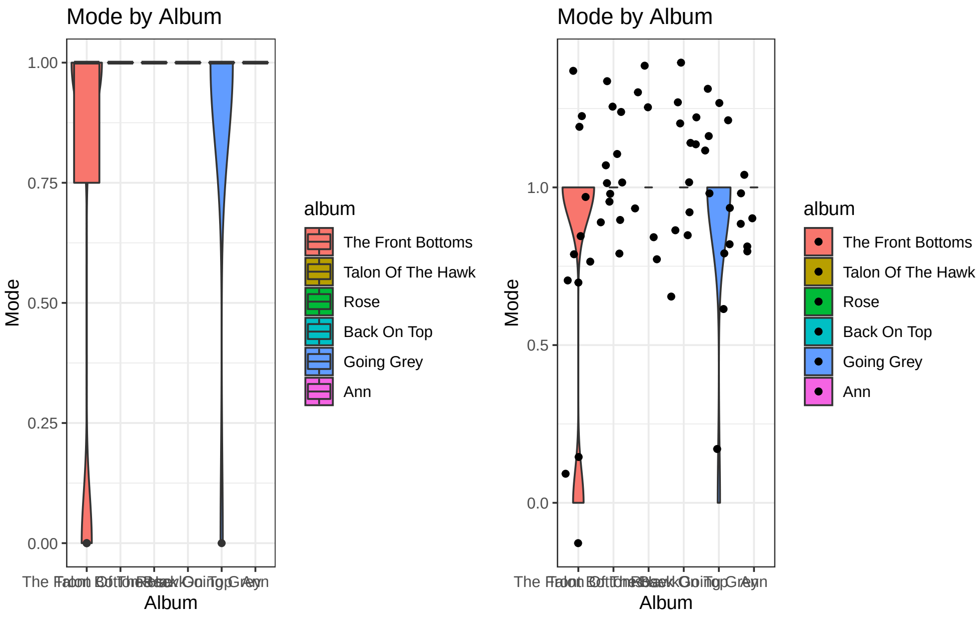
Key levels seem to be similar across albums, except for Back On Top, which has a higher median than all of the other albums. This is enough to motivate us to further investigate this variable.

```
> ggdat<-data.frame(album=df$album, y=df$loudness)
> g1<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Album")+
+   ylab("Loudness")+
+   ggtitle("Loudness by Album")+
+   theme_bw()
> g2<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Album")+
+   ylab("Loudness")+
+   ggtitle("Loudness by Album")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



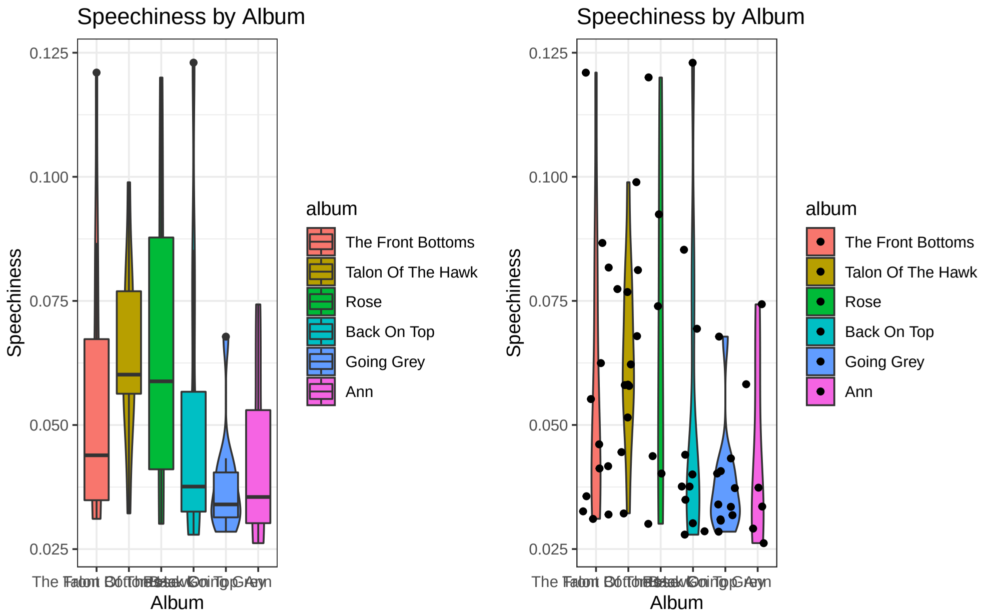
Loudness medians seem to differ across all albums, with Ann having a lower relative median. Loudness will therefore be further interpretted.

```
> ggdat<-data.frame(album=df$album, y=df$mode)
> g1<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Album")+
+   ylab("Mode")+
+   ggtitle("Mode by Album")+
+   theme_bw()
> g2<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Album")+
+   ylab("Mode")+
+   ggtitle("Mode by Album")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



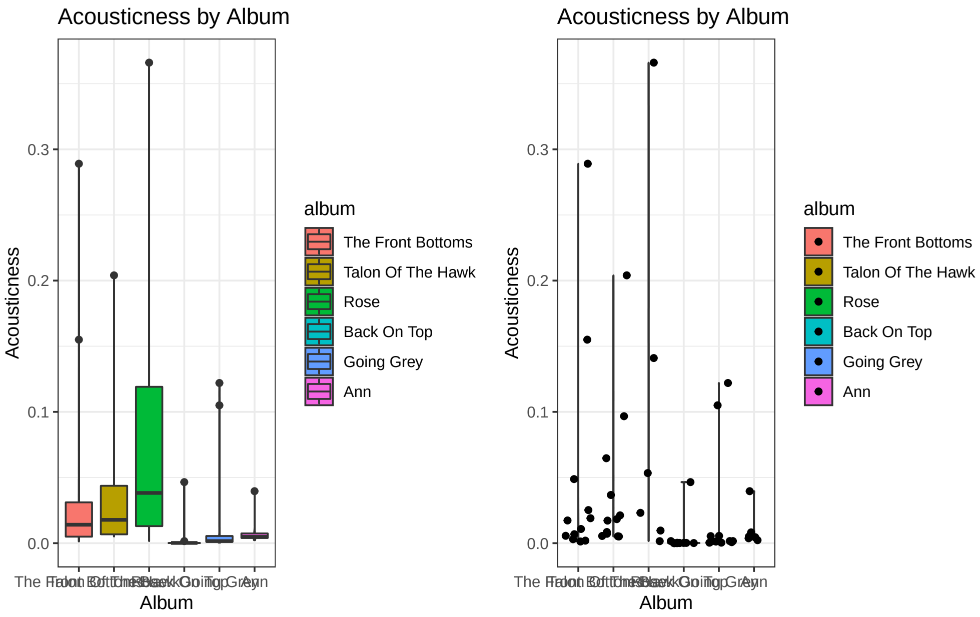
Mode is a variable with only 0 and 1 as the correct values, meaning that this plot isn't informative. Regardless, we see that the median for all albums is 1, meaning we will not be further investigating this variable.

```
> ggdat<-data.frame(album=df$album, y=df$speechiness)
> g1<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Album")+
+   ylab("Speechiness")+
+   ggtitle("Speechiness by Album")+
+   theme_bw()
> g2<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Album")+
+   ylab("Speechiness")+
+   ggtitle("Speechiness by Album")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



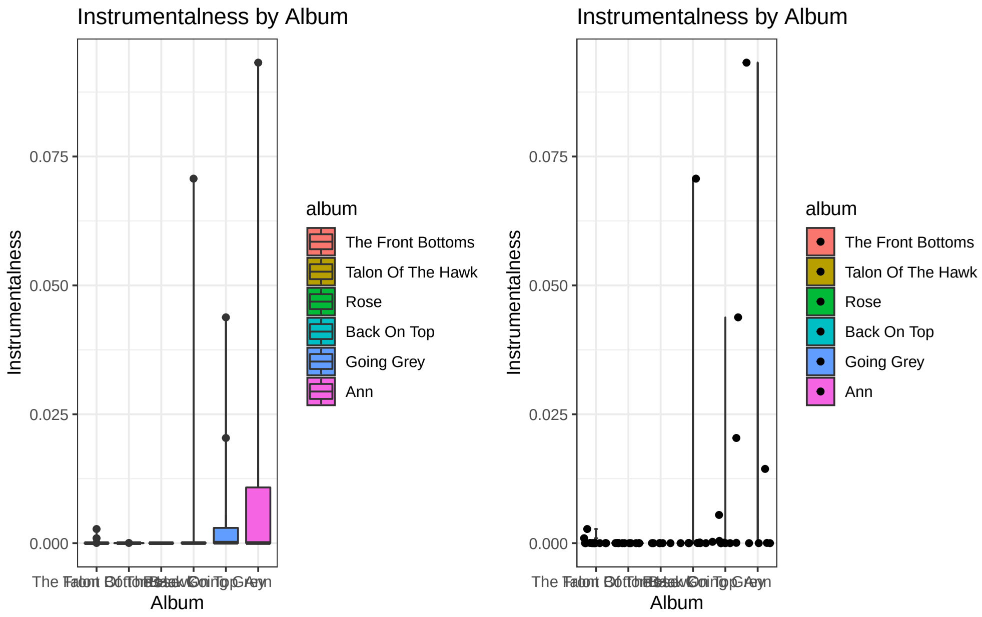
Generally, speechiness seems to decrease as the albums become newer. This could point to a reason why Ann is the preferred album.

```
> ggdat<-data.frame(album=df$album, y=df$acousticness)
> g1<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Album")+
+   ylab("Acousticness")+
+   ggtitle("Acousticness by Album")+
+   theme_bw()
> g2<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Album")+
+   ylab("Acousticness")+
+   ggtitle("Acousticness by Album")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



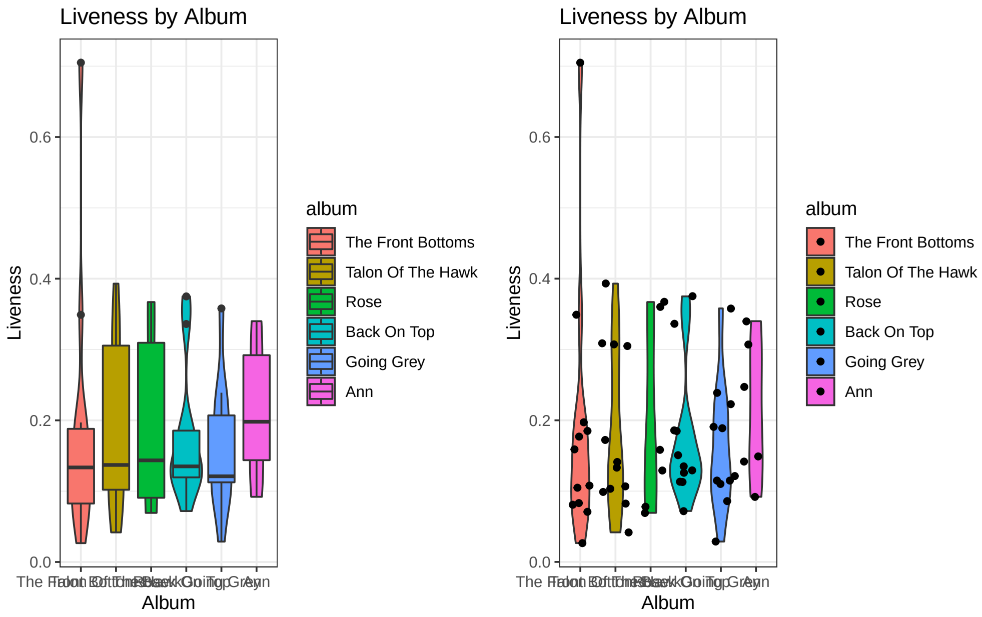
Acousticness, while having similar medians across albums, has very different spreads depending on the album, leading to further speculation.

```
> ggdat<-data.frame(album=df$album, y=df$instrumentalness)
> g1<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Album")+
+   ylab("Instrumentalness")+
+   ggtitle("Instrumentalness by Album")+
+   theme_bw()
> g2<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Album")+
+   ylab("Instrumentalness")+
+   ggtitle("Instrumentalness by Album")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



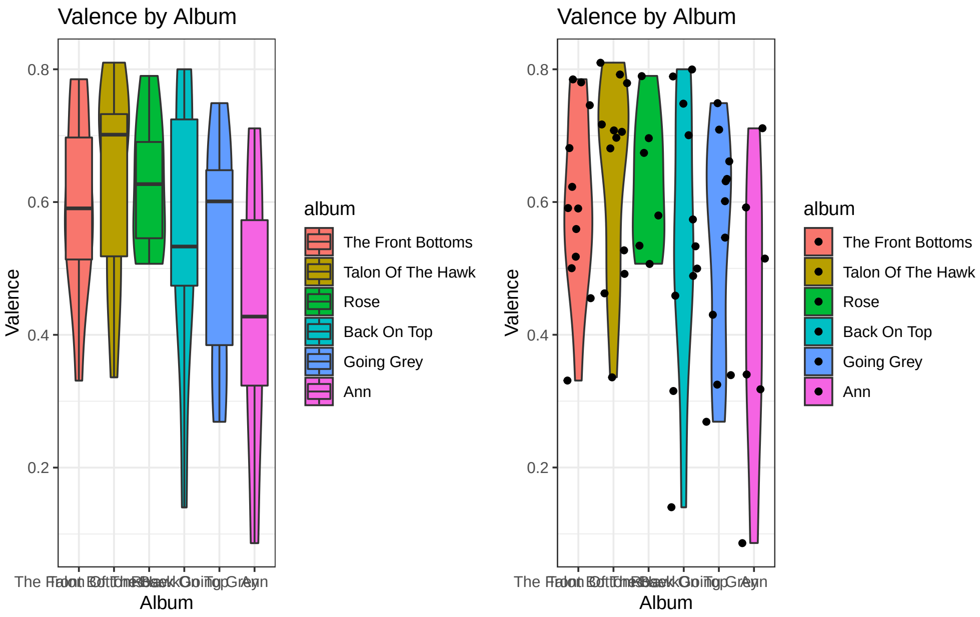
Instrumentalness, while having a mean close to zero for all albums, seems to have a particularly larger spread for Ann, lending to further analyses.

```
> ggdat<-data.frame(album=df$album, y=df$liveness)
> g1<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Album")+
+   ylab("Liveness")+
+   ggtitle("Liveness by Album")+
+   theme_bw()
> g2<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Album")+
+   ylab("Liveness")+
+   ggtitle("Liveness by Album")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



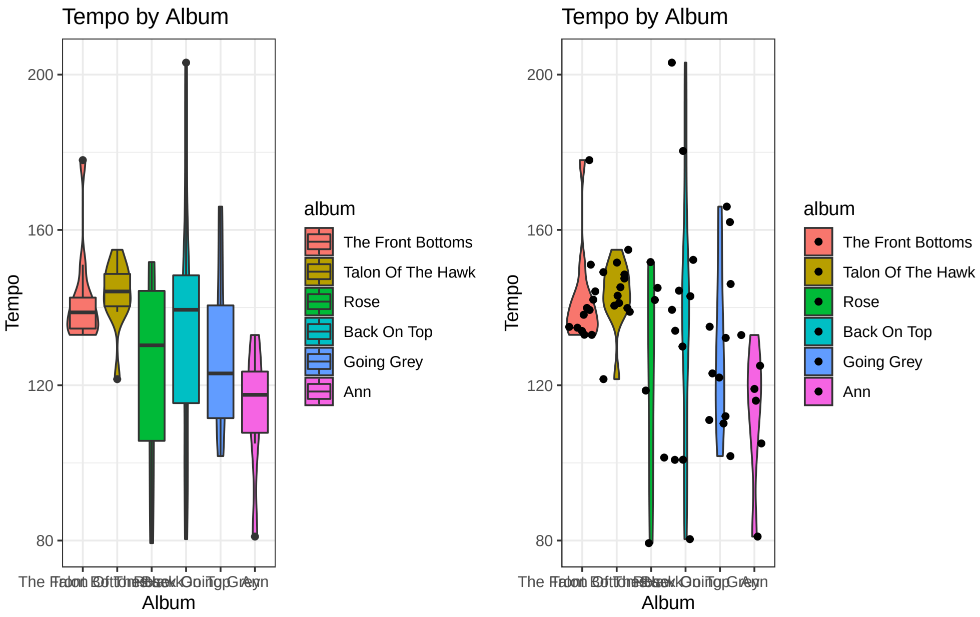
Ann seems to have a higher median liveness than the other albums, but whether or not this median is statistically significant will have to be further investigated.

```
> ggdat<-data.frame(album=df$album, y=df$valence)
> g1<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Album")+
+   ylab("Valence")+
+   ggtitle("Valence by Album")+
+   theme_bw()
> g2<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Album")+
+   ylab("Valence")+
+   ggtitle("Valence by Album")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



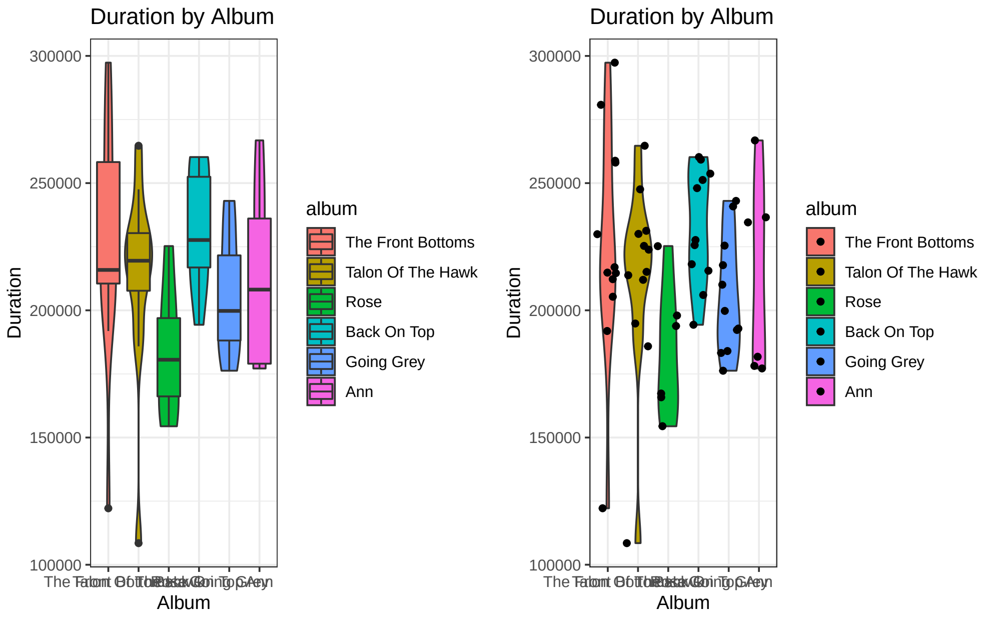
Valence seems to decrease as the albums are released, pointing to a trend that may explain why Ann, the most recent album, is preferred.

```
> ggdat<-data.frame(album=df$album, y=df$tempo)
> g1<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Album")+
+   ylab("Tempo")+
+   ggtitle("Tempo by Album")+
+   theme_bw()
> g2<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Album")+
+   ylab("Tempo")+
+   ggtitle("Tempo by Album")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



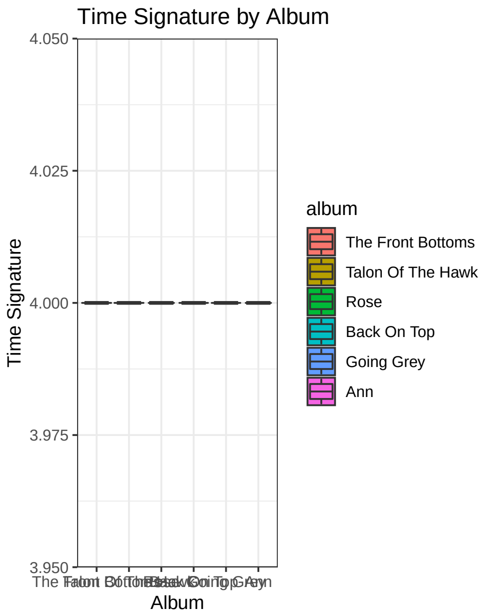
Similarly to Valence, median tempo by album seems to decrease as the albums are released, pointing to a trend that may explain why Ann, the most recent album, is preferred.

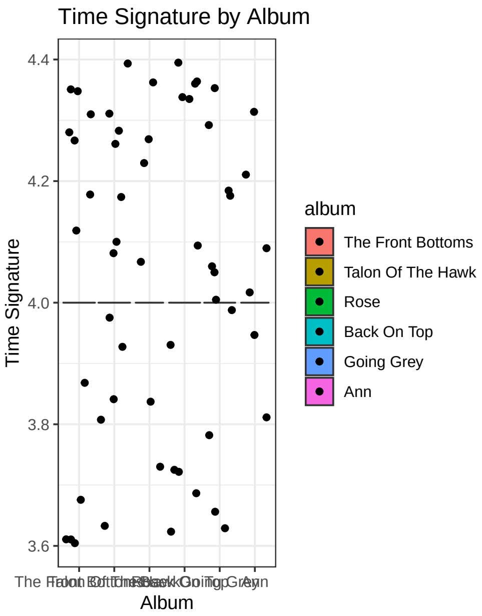
```
> ggdat<-data.frame(album=df$album, y=df$duration_ms)
> g1<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Album")+
+   ylab("Duration")+
+   ggtitle("Duration by Album")+
+   theme_bw()
> g2<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Album")+
+   ylab("Duration")+
+   ggtitle("Duration by Album")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```



Duration medians seem to be similar across albums except for Rose, pointing to further statistical investigation.

```
> ggdat<-data.frame(album=df$album, y=df$time_signature)
> g1<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   #geom_jitter()+
+   geom_boxplot()+
+   xlab("Album")+
+   ylab("Time Signature")+
+   ggtitle("Time Signature by Album")+
+   theme_bw()
> g2<-ggplot(data = ggdat, aes(x=album, y=y, fill=album))+
+   geom_violin()+
+   geom_jitter()+
+   #geom_boxplot()+
+   xlab("Album")+
+   ylab("Time Signature")+
+   ggtitle("Time Signature by Album")+
+   theme_bw()
> grid.arrange(g1,g2,ncol=2)
```





Time Signature (4) is the same across all albums. Since all observations are the same value across all albums, there is no need for further statistical analysis.

To ensure robustness of our results, I default to a Mood's Median Test. The assumptions for this test are that the observations are independent and representative. Clearly, the observations of the band's albums are representative because they are some of the band's albums varied over time. For the purposes of our test, we assume independence. However, it can clearly be argued that the observations, or songs/albums are not independent since artists don't necessarily produce songs independently of each other.

```
> mood.medtest(danceability~album,data=df) #significant

        Mood's median test

data:  danceability by album
p-value = 0.0003028

> mood.medtest(energy~album,data=df) #significant

        Mood's median test

data:  energy by album
p-value = 0.02076

> mood.medtest(key~album,data=df) #not significant

        Mood's median test

data:  key by album
p-value = 0.9066

> mood.medtest(loudness~album,data=df) #sig

        Mood's median test

data:  loudness by album
p-value = 0.0005943
```

```
> mood.medtest(acousticness~album,data=df) #sig

        Mood's median test

data:  acousticness by album
p-value = 0.001224

> mood.medtest(speechiness~album,data=df) #sig

        Mood's median test

data:  speechiness by album
p-value = 0.006577

> mood.medtest(instrumentalness~album,data=df) #sig

        Mood's median test

data:  instrumentalness by album
p-value = 0.04229

> mood.medtest(liveness~album,data=df) #not

        Mood's median test

data:  liveness by album
p-value = 0.7005

> mood.medtest(valence~album,data=df) #not

        Mood's median test

data:  valence by album
p-value = 0.7325

> mood.medtest(tempo~album,data=df) #sig

        Mood's median test

data:  tempo by album
p-value = 0.002634

> mood.medtest(duration_ms~album,data=df) #not

        Mood's median test

data:  duration_ms by album
p-value = 0.1596
```

We run the Mood's median test for all variables spotify provides besides Time Signature and Mode. Our null hypothesis for each test is that the median of the variable is the same across all albums. Our alternative hypothesis for each test is that at least one album has a different median for the variable. I find that danceability, energy, loudness, acousticness, speechiness, instrumentalness, and tempo all have a group median that differs from the other group medians. To see if these differences are specifically related to the album Ann, I can run pairwise median tests on each of the 7 variables.

```
> PTBH1<-pairwiseMedianTest(danceability˜album,
+                         data   = df,
+                         method = "BH")

> cldList(p.adjust ˜ Comparison,
+         data = PTBH1,
+         threshold = 0.05)

            Group Letter MonoLetter
1 TheFrontBottoms      a        a
2  TalonOfTheHawk      b         b
3            Rose    abc        abc
4       BackOnTop     bc         bc
5       GoingGrey    abc        abc
6             Ann     ac        a c

> #Ann is sig diff from Talon of the Hawk

> PTBH2<-pairwiseMedianTest(energy˜album,
+                         data   = df,
+                         method = "BH")

> cldList(p.adjust ˜ Comparison,
+         data = PTBH2,
+         threshold = 0.05)

            Group Letter MonoLetter
1 TheFrontBottoms      a         a
2  TalonOfTheHawk      b          b
3            Rose     ab         ab
4       BackOnTop     ab         ab
5       GoingGrey      a         a
6             Ann      a         a

> #Ann is sig diff from Talon of the Hawk

> PTBH3<-pairwiseMedianTest(loudness˜album,
+                         data   = df,
+                         method = "BH")

> cldList(p.adjust ˜ Comparison,
+         data = PTBH3,
+         threshold = 0.05)

            Group Letter MonoLetter
1 TheFrontBottoms     ab        ab
2  TalonOfTheHawk     ac        a c
3            Rose      c          c
4       BackOnTop      b         b
5       GoingGrey      b         b
6             Ann    abc        abc

> #Ann isn't sig diff in terms of loudness

> PTBH4<-pairwiseMedianTest(acousticness˜album,
+                         data   = df,
+                         method = "BH")
```

```
> cldList(p.adjust ~ Comparison,
+         data = PTBH4,
+         threshold = 0.05)

            Group Letter MonoLetter
1 TheFrontBottoms      a        a
2  TalonOfTheHawk      a        a
3            Rose     ab        ab
4       BackOnTop      c          c
5       GoingGrey      b         b
6             Ann     ab        ab

> #Ann is sig dif from Back On Top

> PTBH5<-pairwiseMedianTest(speechiness~album,
+                        data   = df,
+                        method = "BH")

> cldList(p.adjust ~ Comparison,
+         data = PTBH5,
+         threshold = 0.05)

            Group Letter MonoLetter
1 TheFrontBottoms     ab         ab
2  TalonOfTheHawk      a         a
3            Rose     ab         ab
4       BackOnTop     ab         ab
5       GoingGrey      b          b
6             Ann     ab         ab

> #Ann isn't sig dif in terms of speechiness

> PTBH6<-pairwiseMedianTest(instrumentalness~album,
+                        data   = df,
+                        method = "BH")

> cldList(p.adjust ~ Comparison,
+         data = PTBH6,
+         threshold = 0.05)

            Group Letter MonoLetter
1 TheFrontBottoms     ab         ab
2  TalonOfTheHawk     ab         ab
3            Rose      a         a
4       BackOnTop      b          b
5       GoingGrey      b          b
6             Ann     ab         ab

>
> #Ann isn't sig dif in terms of instrumentalness

> PTBH7<-pairwiseMedianTest(tempo~album,
+                        data   = df,
+                        method = "BH")
```

```
> cldList(p.adjust ~ Comparison,
+         data = PTBH7,
+         threshold = 0.05)

            Group Letter MonoLetter
1 TheFrontBottoms      a          a
2  TalonOfTheHawk      a          a
3            Rose     ab         ab
4       BackOnTop     ab         ab
5       GoingGrey     ab         ab
6             Ann      b          b

> #Ann is sig diff from The Front Bottoms and Talon of the Hawk
```

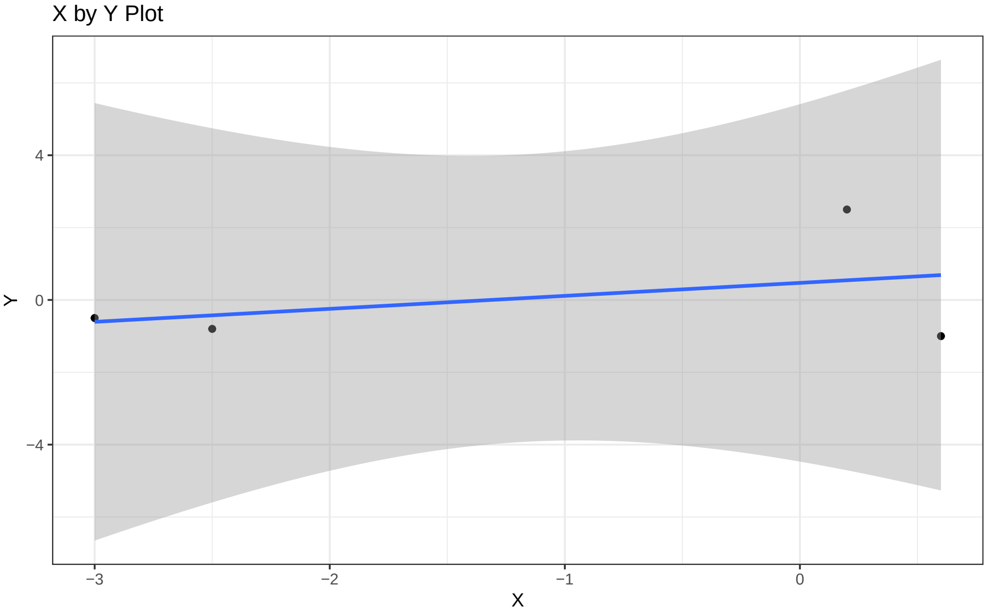
In terms of danceability and energy, we find that Ann is significantly different from Talon of the Hawk. In terms of acousticness, Ann is significantly different from Back On Top. In terms of Tempo, Ann is significantly different from The Front Bottoms and Talon of the Hawk, two of the Front Bottom's early albums. The other variables did not show a significant difference between the album Ann and any other albums. From our analysis, we can inference that Ann may be your wife's preferred album due to its danceability, energy, acousticness, and especially tempo. Maybe this research will prompt a new playlist to be made with only Front Bottom songs that meet specific criteria.

5. Consider the following data.

```
> y = c(-0.5,-0.8,2.5,-1)
> x = c(-3,-2.5,0.2,0.6)
```

(a) Plot the data.

```
> ggdat<-data.frame(x=x, y=y)
> ggplot(data=ggdat, aes(x=x, y=y))+
+   geom_point()+
+   geom_smooth(method="lm")+
+   theme_bw()+
+   xlab("X")+
+   ylab("Y")+
+   ggtitle("X by Y Plot")
```



This plot shows that there is slightly positive relationship between the X and Y values. The large confidence interval leads us to believe though that this is not a strong relationship.

(b) Calculate the Pearson correlation – compare this to the plot.

```
> cor(x,y,method="pearson")
[1] 0.4005973
```

The Pearson correlation is about 0.40. This means that there is a weak to moderate linear relationship between the X and Y values.

(c) Calculate the Spearman's rank and Kendall's tau-b correlation. Compare these values to the Pearson correlation and the plot.

```
> cor(x,y,method = "spearman")
[1] -0.4
> cor(x,y,method = "kendall")
[1] -0.3333333
```

Spearman's Rank and Kendall's tau-b give us correlation values of -0.4 and -0.333, respectively. This tells us that the x and y values have a weak to moderate negative relationship. If we compare these possible slopes to the plot, we see that they would probably intersect, but the lines would be going in opposite directions.

References

Auguie, B. (2017). *gridExtra: Miscellaneous Functions for "Grid" Graphics.* R package version 2.3.

Cheng, J., Bravo, H. C., Ooms, J., and Chang, W. (2019). *httpuv: HTTP and WebSocket Server Library.* R package version 1.5.2.

Dantas, T. (2019). *Rspotify: Access to Spotify API via R.* R package version 0.1.0.

Hervé, M. (2019). *RVAideMemoire: Testing and Plotting Procedures for Biostatistics.* R package version 0.9-73.

Jockers, M. L. (2015). *Syuzhet: Extract Sentiment and Plot Arcs from Text.*

Mangiafico, S. (2019). *rcompanion: Functions to Support Extension Education Program Evaluation.* R package version 2.3.7.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis.* Springer-Verlag New York.

Wickham, H. (2017). *tidyverse: Easily Install and Load the 'Tidyverse'.* R package version 1.2.1.

Wickham, H. (2018). *scales: Scale Functions for Visualization.* R package version 1.0.0.