Final Project

Daniel Yan, Ira Bradie, Nathan Zhang

2024-03-09

```
music <- read.csv ("https://corgis-edu.github.io/corgis/datasets/csv/music/music.csv")

# Testing correlation between duration and popularity
music_clean <- music%>% filter(song.duration != 0, song.hotttnesss > 0) # Removes null values for song
count(music_clean) # gives us number of cases without null values
```

```
##      n
## 1 4214
```

```
min(music_clean$song.duration) # verifies that the minimum duration isn't 0, which would be a null valu
```

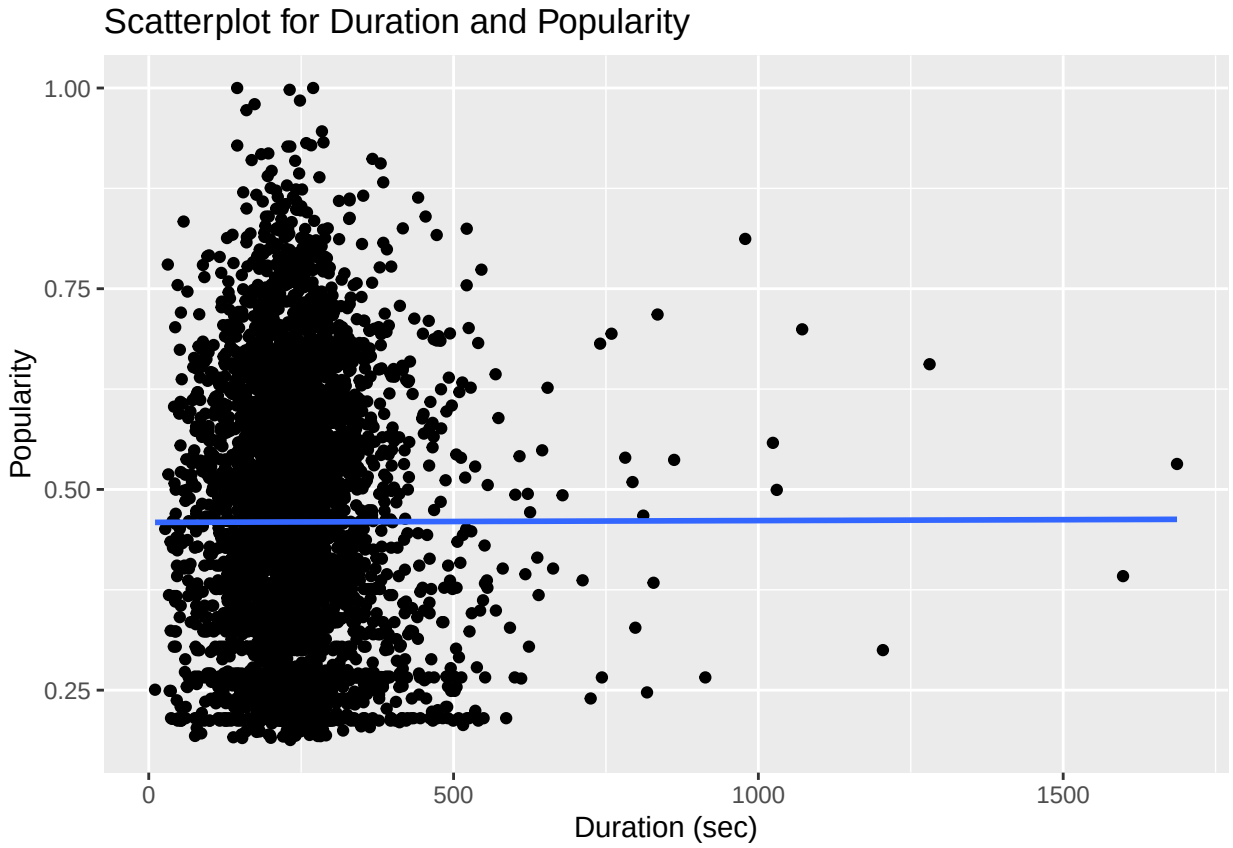
```
## [1] 10.34404
```

```
cor(music_clean$song.hotttnesss, music_clean$song.duration)# gives correlation coefficient
```

```
## [1] 0.001382954
```

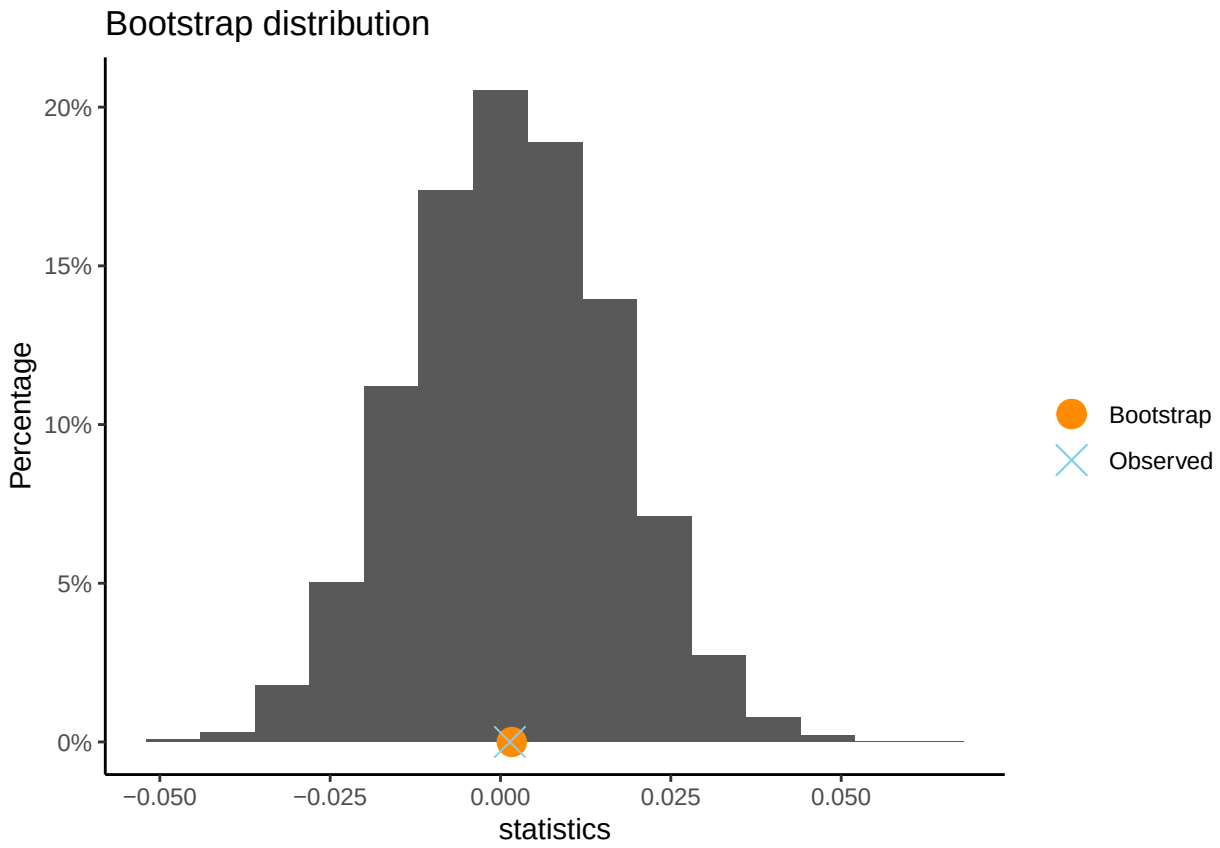
```
ggplot(music_clean, aes(x = song.duration, y = song.hotttnesss)) +  labs(title = "Scatterplot for Durati
```

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




```
bootCor(music_clean$song.hotttnesss ~ music_clean$song.duration, data = music_clean) # bootstraps corre
```

```
##
##  ** Bootstrap interval of correlation
##
##  Observed correlation between music_clean$song.duration and music_clean$song.hotttnesss : 0.00138
##  Mean of bootstrap distribution: 0.00165
##  Standard error of bootstrap distribution: 0.01486
##
##  Bootstrap percentile interval
##        2.5%       97.5%
## -0.02705368  0.03043021
##
##      *--------------*
```

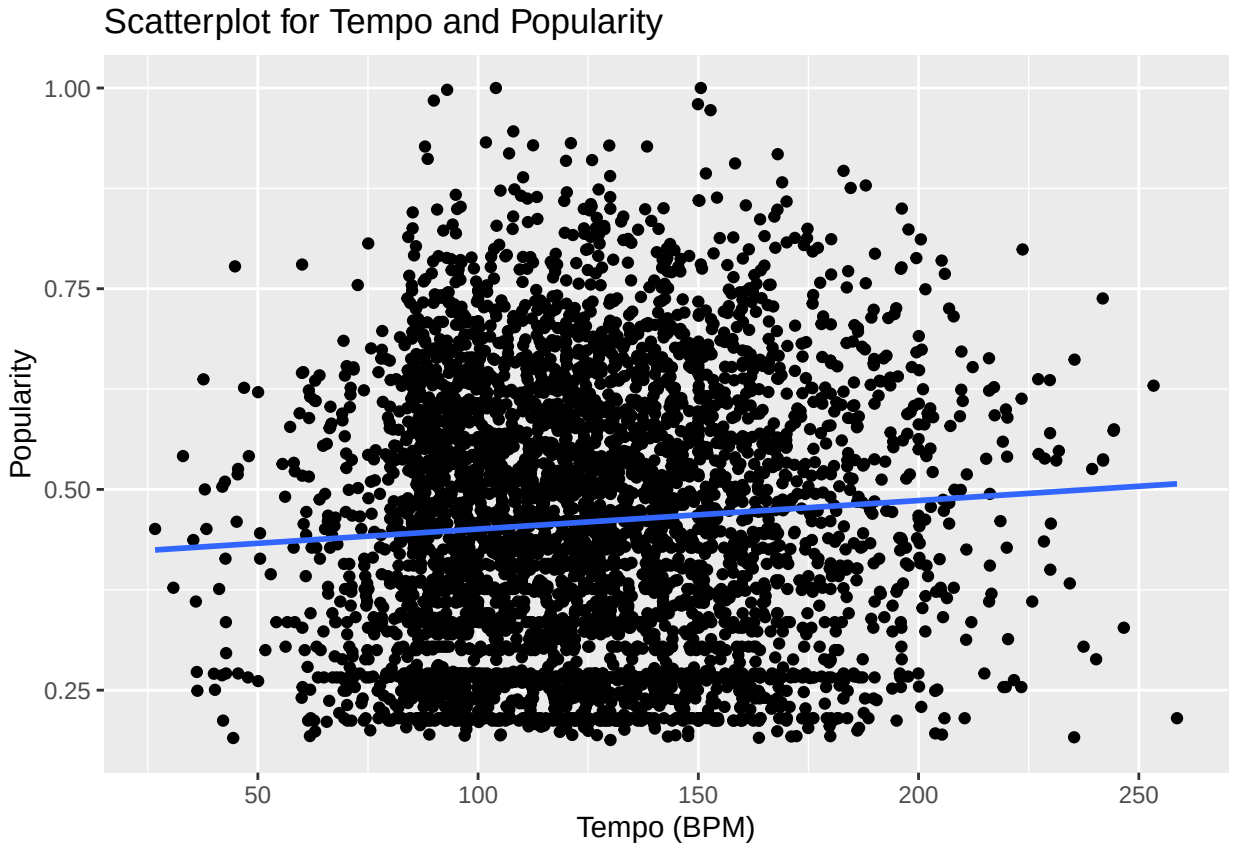



```
#Is there association between tempo and popularity?
# read data
# create filtered data frame without value 0 or less for hotttnesss and song.tempo
# print the counts for the filtered data
# create scatterplot
# create residual plot
# make the bootstrap interval for the correlation
mus <- music%>% filter(song.tempo > 0, song.hotttnesss>0)
count(mus)
```

```
##      n
## 1 4208
```

```
ggplot(mus, aes(x = song.tempo, y = song.hotttnesss)) +
  geom_point() +
  stat_smooth(method = "lm", se = FALSE) +
  labs(title = "Scatterplot for Tempo and Popularity", x = "Tempo (BPM)", y = "Popularity")
```

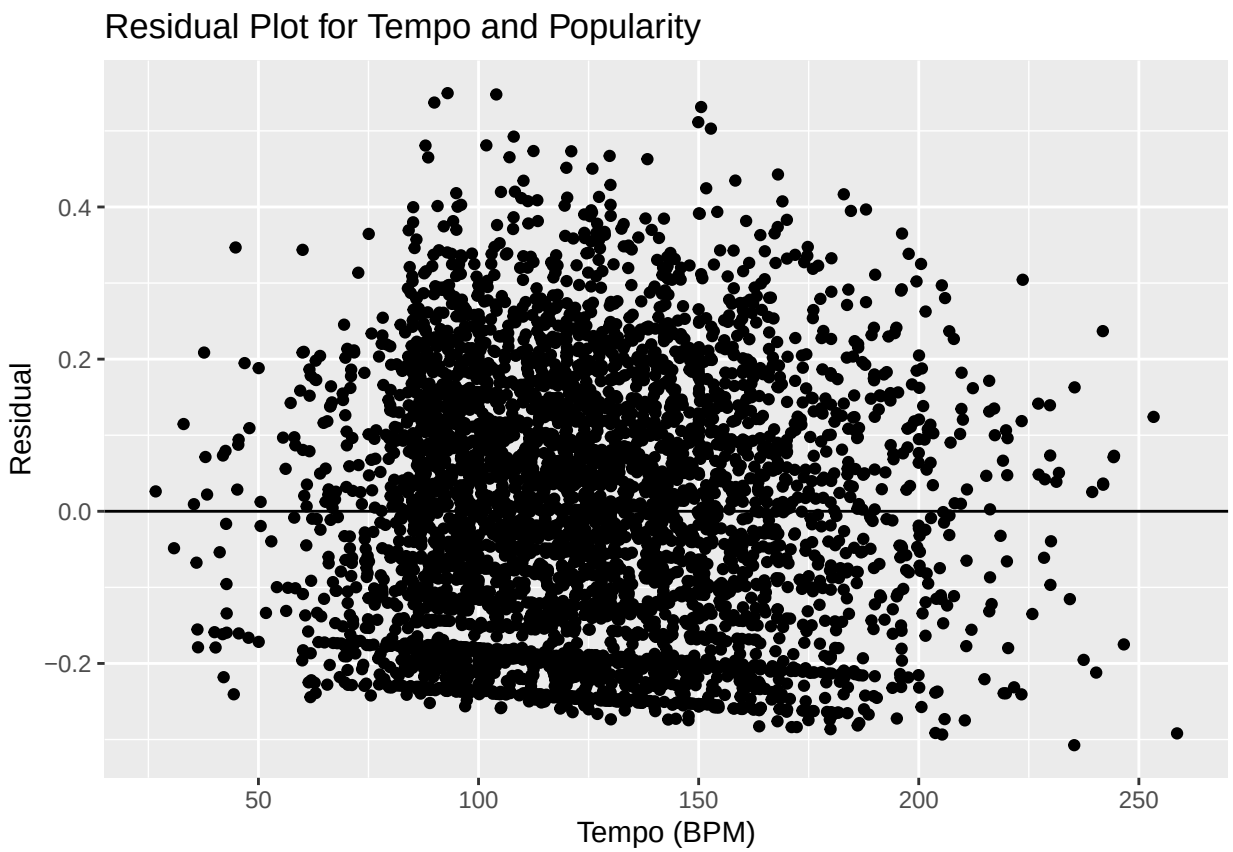
```
## `geom_smooth()` using formula = 'y ~ x'
```




```
library(broom)
mus.lm <- lm(song.hotttnesss ~ song.tempo, data = mus)
summary(mus.lm)
```

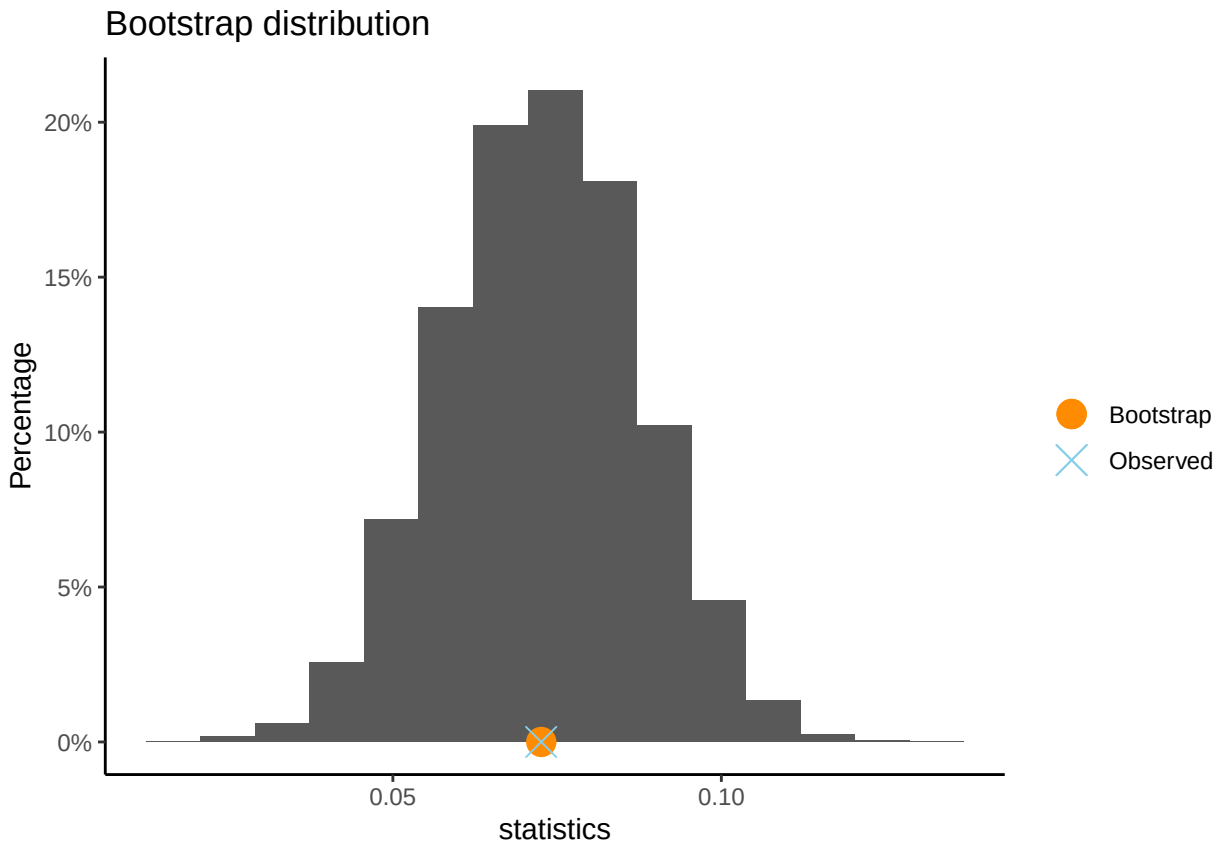
```
##
## Call:
## lm(formula = song.hotttnesss ~ song.tempo, data = mus)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.3074 -0.1439 -0.0088  0.1245  0.5496
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) 4.152e-01  9.734e-03  42.650  < 2e-16 ***
## song.tempo  3.551e-04  7.525e-05   4.719 2.45e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1678 on 4206 degrees of freedom
## Multiple R-squared:  0.005267,   Adjusted R-squared:  0.00503
## F-statistic: 22.27 on 1 and 4206 DF,  p-value: 2.446e-06
```

```
mus.aug <- augment(mus.lm)
ggplot(mus.aug, aes(x = song.tempo, y = .resid)) +
  geom_point() +
  geom_hline(yintercept = 0) +
  labs(title = "Residual Plot for Tempo and Popularity", x = "Tempo (BPM)", y = "Residual")
```

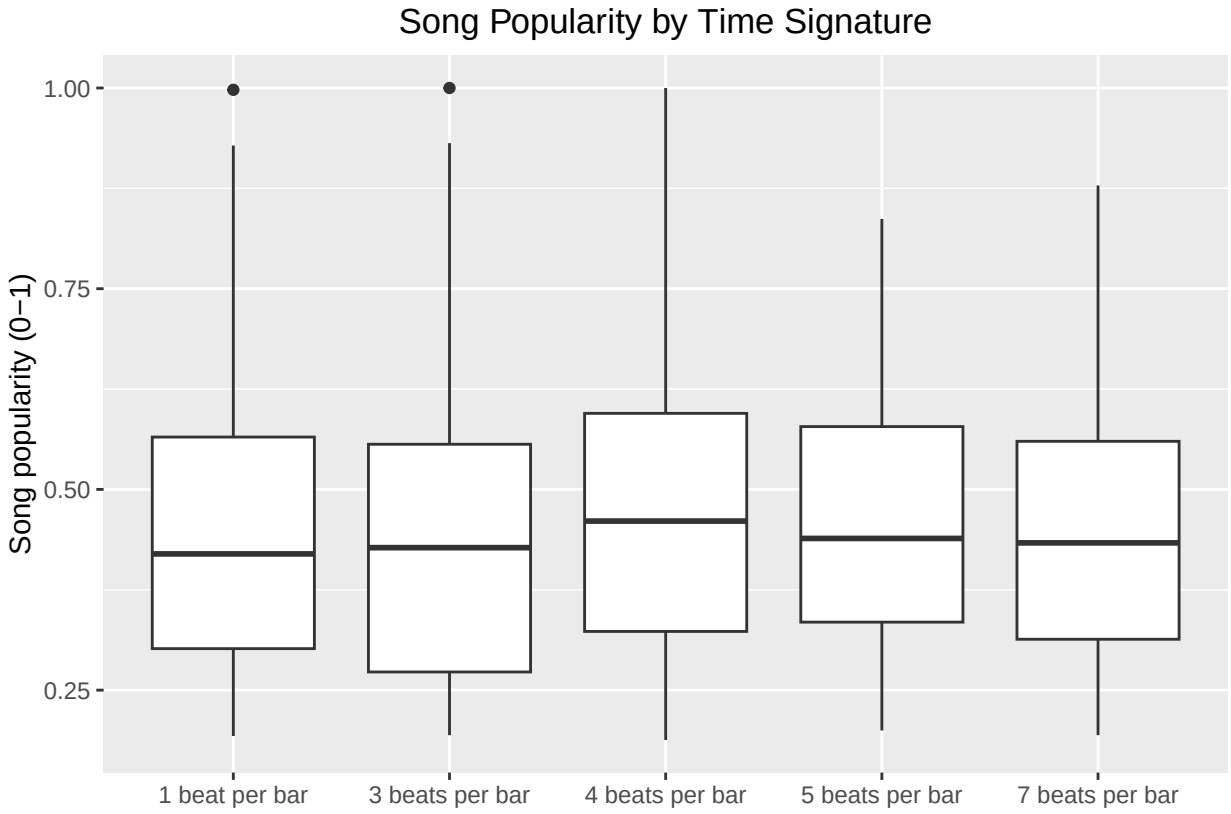



```
bootCor(song.hotttnesss ~ song.tempo, data = mus)
```

```
##
##  ** Bootstrap interval of correlation
##
##  Observed correlation between song.tempo and song.hotttnesss : 0.07257
##  Mean of bootstrap distribution: 0.07259
##  Standard error of bootstrap distribution: 0.01486
##
##  Bootstrap percentile interval
##       2.5%      97.5%
## 0.04380278 0.10149818
##
##      *--------------*
```




```
#filters the variables song.time_signature and song.hotttnesss for null values
mus <- music%>% filter(song.time_signature != 0, song.hotttnesss > 0)
#makes variables for song.hotttnesss of each time signature
speed_1 = mus$song.hotttnesss[mus$song.time_signature == 1]
speed_3 = mus$song.hotttnesss[mus$song.time_signature == 3]
speed_4 = mus$song.hotttnesss[mus$song.time_signature == 4]
speed_5 = mus$song.hotttnesss[mus$song.time_signature == 5]
speed_7 = mus$song.hotttnesss[mus$song.time_signature == 7]
#creates side-by-side boxplots for each time signature
p1 <- ggplot() +
  geom_boxplot(mapping = aes(x = "1 beat per bar", y = speed_1)) +
  geom_boxplot(mapping = aes(x = "3 beats per bar", y = speed_3)) +
  geom_boxplot(mapping = aes(x = "4 beats per bar", y = speed_4)) +
  geom_boxplot(mapping = aes(x = "5 beats per bar", y = speed_5)) +
  geom_boxplot(mapping = aes(x = "7 beats per bar", y = speed_7))
#labels boxplots
p1 + xlab("") + ylab("Song popularity (0-1)") +
  ggtitle("Song Popularity by Time Signature") +
  theme(plot.title = element_text(hjust = .5))
```



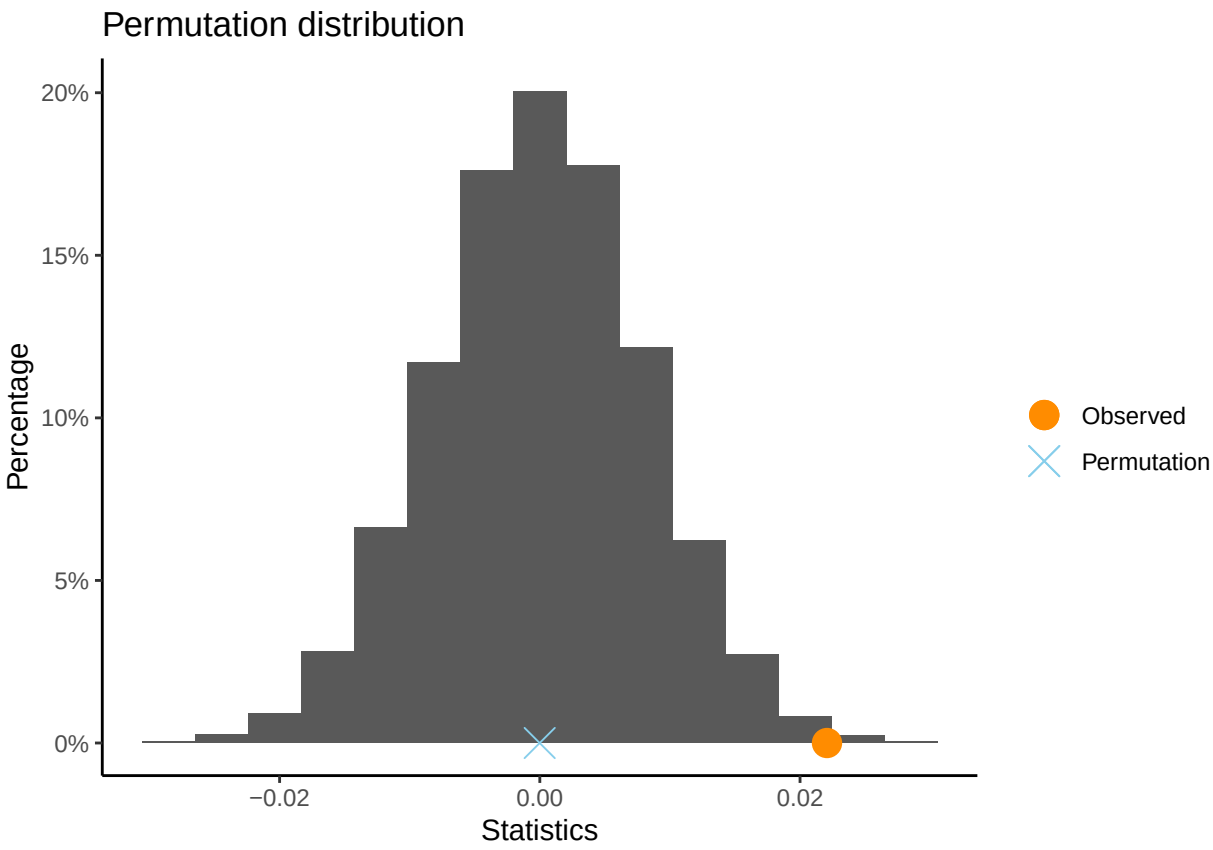

```
#makes song.time_signature categorical
song.time2 <- as.factor(mus$song.time_signature)
#anova test for if one of mean song.hotttnesss' is different for each song.time_signature
mus <- music%>% filter(song.time_signature != 0, song.hotttnesss > 0)
speeds_anova = aov(song.hotttnesss ~ song.time2, data = mus)
summary(speeds_anova)
```

```
##               Df Sum Sq Mean Sq F value   Pr(>F)
## song.time2     4   0.55  0.1365   4.841 0.000679 ***
## Residuals   4207 118.63  0.0282
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
#separates between time signature 4 and time signature 3
mus$time_3_ind <- ifelse(mus$song.time_signature == 3, 1, 0)
#permutation test for difference in means between popularity for ts4 and ts3
permTest(song.hotttnesss ~ mus$time_3_ind, data = mus)
```

```
##
##  ** Permutation test **
##
##  Permutation test with alternative: two.sided
##  Observed statistic
##   0 :  0.461986   1 :  0.4399144
##  Observed difference: 0.02207
##
##  Mean of permutation distribution: -1e-05
##  Standard error of permutation distribution: 0.00808
##  P-value:  0.0064
```

```
##
##  *------------*
```


Permutation distribution
20%
15%
10%
5%
0%
Percentage
−0.02
0.00
0.02
Statistics
Observed
Permutation