

# IST687 - Music Classification Project

*Team 2 - Sebastian Castro, John Fields, Courtney Smith, Jeremy Wallner*

*5/13/2019*

## Executive Summary

The purpose of this project is to analyze the Million Song Database to predict “Hot” artists and songs based on the attributes such as familiarity, artist location, loudness, terms used, etc. The analysis was done using R software on a 10,000 track subset of the data and our model was able to predict “Hot” songs with ~80% accuracy.

## Table of Contents

Executive Summary Data Analysis Conclusion Final proofing

## Introduction

## Related Work

Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011), 2011.

## Dataset

```
#New code from Courtney to change from 3 to 5 categories of artist hotness
#music <- read.csv("~/Intro data science/Music project/newmusic.csv")
music <- read.csv("/Users/johnfields/Library/Mobile Documents/com~apple~CloudDocs/Syracuse/IST687/Project 1/music.csv")
str(music)
```

```
## 'data.frame': 10000 obs. of 35 variables:
## $ artist.hotttnesss : num 0.402 0.417 0.343 0.454 0.402 ...
## $ artist.id : Factor w/ 3888 levels "AR009211187B989185",...: 1269 2354 2168 715 3609 ...
## $ artist.name : Factor w/ 4412 levels ":Blacks On :Blondes",...: 682 3798 3562 67 1569 ...
## $ artist_mbtags : Factor w/ 277 levels "", "0.333", "60s",...: 1 52 1 262 1 1 1 1 1 ...
## $ artist_mbtags_count : num 0 1 0 1 0 0 0 0 0 0 ...
## $ bars_confidence : num 0.643 0.007 0.98 0.017 0.175 0.121 0.709 0.142 0.806 0.047 ...
## $ bars_start : num 0.585 0.711 0.732 1.306 1.064 ...
## $ beats_confidence : num 0.834 1 0.98 0.809 0.883 0.438 0.709 0.234 0.44 1 ...
## $ beats_start : num 0.585 0.206 0.732 0.81 0.136 ...
## $ duration : num 219 148 177 233 210 ...
## $ end_of_fade_in : num 0.247 0.148 0.282 0 0.066 ...
## $ familiarity : num 0.582 0.631 0.487 0.63 0.651 ...
## $ key : num 1 6 8 0 2 5 1 4 4 7 ...
## $ key_confidence : num 0.736 0.169 0.643 0.751 0.092 0.635 0 0 0.717 0.053 ...
## $ latitude : num 37.2 35.1 37.2 37.2 37.2 ...
## $ location : Factor w/ 1046 levels " ", " NC", " UbA!", " Minas Gerais",...: 157 584 705 ...
## $ longitude : num -63.9 -90 -63.9 -63.9 -63.9 ...
```

```
## $ loudness           : num  -11.2 -9.84 -9.69 -9.01 -4.5 ...
## $ mode               : int    0 0 1 1 1 1 1 0 1 0 ...
## $ mode_confidence    : num   0.636 0.43 0.565 0.749 0.371 0.557 0 0.16 0.652 0.473 ...
## $ release.id         : int   300848 300822 514953 287650 611336 41838 25824 8876 358182 692313
## $ release.name       : Factor w/ 7833 levels " Lazy Afternoon En Anglais",...: 2192 1746 3536 1
## $ similar            : Factor w/ 2839 levels "AR00K8N11C8A41687B",...: 2410 2227 1145 304 2333
## $ song.hottnesss     : num    0.602 NA NA NA 0.605 ...
## $ song.id           : Factor w/ 10000 levels " Polovtsian Dances / Rimsky-Korsakov: Russian 1
## $ start_of_fade_out  : num    219 138 172 217 199 ...
## $ tatums_confidence  : num    0.779 0.969 0.482 0.601 1 0.136 0.467 0.292 0.121 1 ...
## $ tatums_start      : num    0.285 0.206 0.421 0.563 0.136 ...
## $ tempo             : num    92.2 121.3 100.1 119.3 129.7 ...
## $ terms             : Factor w/ 459 levels "", "8-bit", "acid jazz",...: 216 34 372 327 325 396
## $ terms_freq        : num    1 1 1 0.989 0.887 ...
## $ time_signature     : num    4 4 1 4 4 3 1 3 4 4 ...
## $ time_signature_confidence: num    0.778 0.384 0 0 0.562 0.454 0 0.408 0.487 0.878 ...
## $ title             : Factor w/ 9709 levels "", " -start ID-",...: 3574 7529 482 7477 2532 828
## $ year              : int    0 1969 0 1982 2007 0 0 0 1984 0 ...
```

```
colnames(music)[1] <- "artist.hottnesss"
```

```
#Plot of the variables
```

```
library(ggplot2)
```

```
## Registered S3 methods overwritten by 'ggplot2':
```

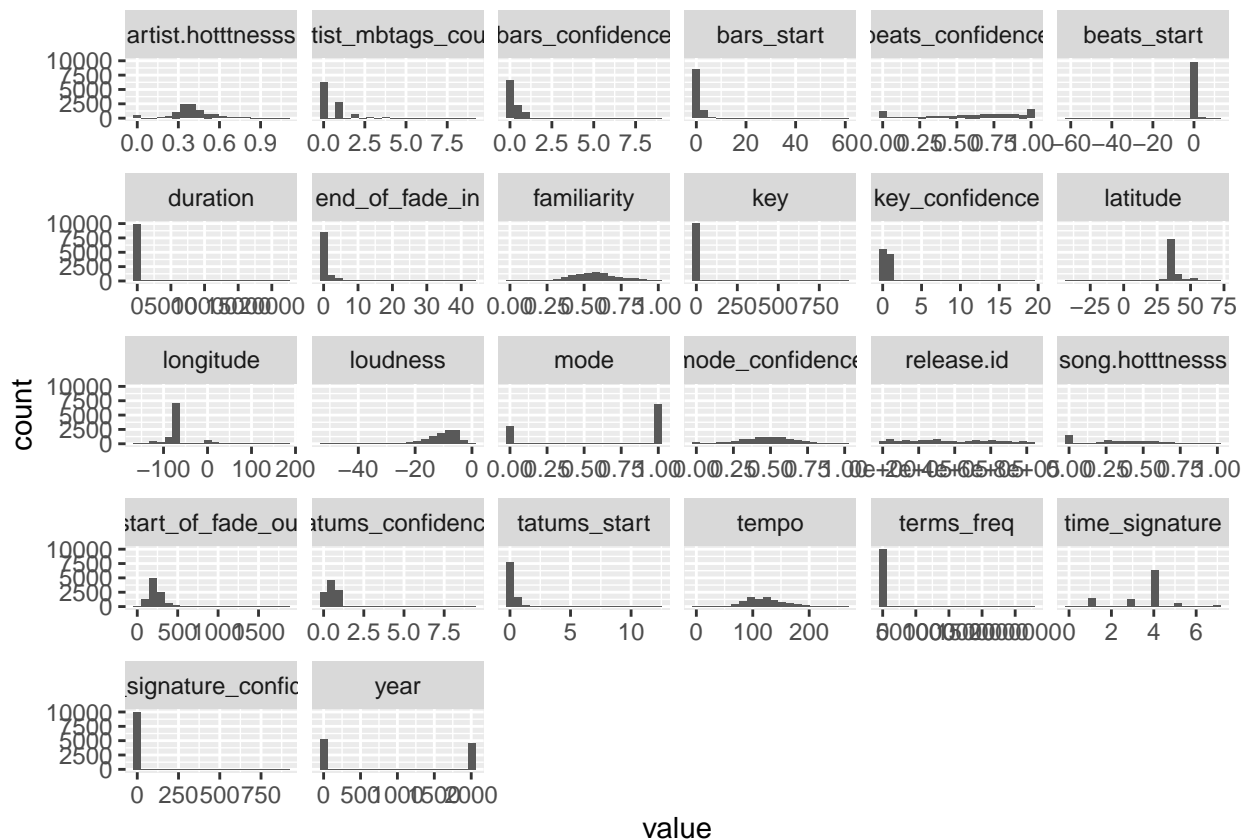
```
##   method      from
## [.quosures    rlang
## c.quosures    rlang
## print.quosures rlang
```

```
library(reshape2)
```

```
ggplot(data = melt(music), mapping = aes(x = value)) + geom_histogram(bins = 20) + facet_wrap(~variable
```

```
## Using artist.id, artist.name, artist_mbtags, location, release.name, similar, song.id, terms, title :
```

```
## Warning: Removed 4355 rows containing non-finite values (stat_bin).
```



```
#New code from Jeremy importing of song list
#newmusic <- read.csv("~/Intro data science/Music project/newmusic3.csv")
newmusic <- read.csv("/Users/johnfields/Library/Mobile Documents/com~apple~CloudDocs/Syracuse/IST687/Pr

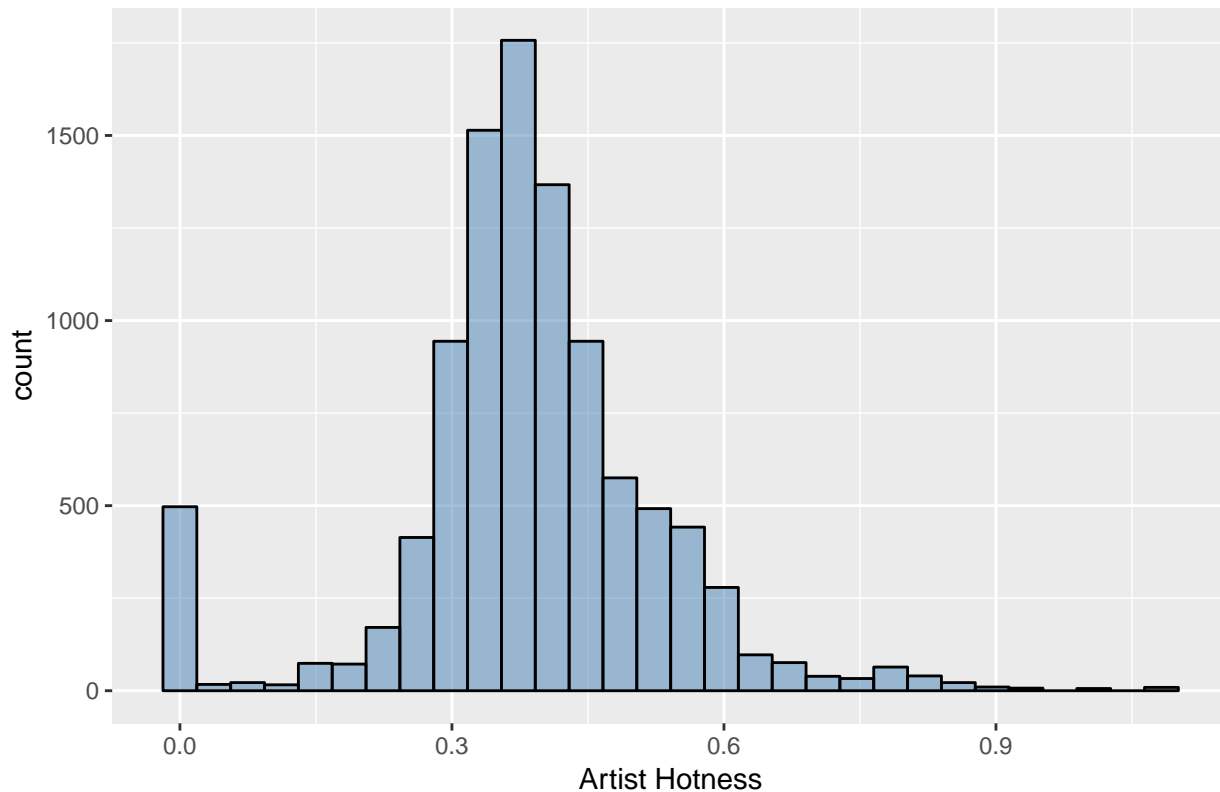
#head(newmusic)
newmusic2 <- newmusic
newmusic3 <- newmusic2[-c(1:2,4:9,11,13:14,19:20,23,36)]

newmusic3 <- na.omit(newmusic3)
cmbomusic <- newmusic3

##Artist Hotness Histogram
library(ggplot2)
ggplot(music, aes(x=artist.hotttness)) + geom_histogram(color="black", fill="steelblue", alpha=0.5) +

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

# Histogram: Artist Hotness



*##Function to create descriptive statistics for artist hotness*

```
descriptive_stats <- function(vector) { library(moments)
  result <- c(Mean=mean(vector),
              Median=median(vector),
              Min = min(vector),
              Max = max(vector),
              SD = sd(vector),
              Quantile = quantile(vector, probs = c(0.25,.50,0.75, 0.95)),
              Skewness = skewness(vector) )
  print(result)
}
descriptive_stats(music$artist.hotttnesss)
```

##	Mean	Median	Min	Max	SD
##	0.3855522	0.3807423	0.0000000	1.0825026	0.1436473
##	Quantile.25%	Quantile.50%	Quantile.75%	Quantile.95%	Skewness
##	0.3252656	0.3807423	0.4538581	0.6011861	-0.1522617

*##Methodology for assigning artist hotness levels - uses quantiles from descriptive\_statistics functi*

*#95% Quantile: 0.6011861 - Hot*

*#75% Quantile: 0.453858 - Warm*

*#50% Quantile: 0.3807423 - Tepid*

*#25% Quantile: 0.3252656 - Cool*

*##Code for assigning labels based on above quantiles*

```
music$artist.hottness.label <- ifelse(music$artist.hotttnesss >=0.6011861, "Hot",
                                     ifelse(music$artist.hotttnesss >=0.453858 & music$artist.hotttnesss < 0.6011861, "Warm",
                                             ifelse(music$artist.hotttnesss >=0.3807423 & music$artist.hotttnesss < 0.453858, "Tepid",
                                                    ifelse(music$artist.hotttnesss >=0.3252656 & music$artist.hotttnesss < 0.3807423, "Cool",
                                                           "Other"))))
```

```

                                                    ifelse(music$artist.hottnesss < 0.3252656, "I
unique(music$artist.hottness.label)

## [1] "Tepid" "Cool" "Warm" "Frigid" "Hot"

#End of new code from Courtney
#Prior to importing, a new column artist.hottnesss.label was adding with
#Hot(>.4590), Warm(<.4590 and >.3357), Cold(<.3357). Four rows with blanks in
#familiarity were also deleted.

music <- na.omit(music)
#Copy original data to a new dataframe music1 and exclude unneeded data
music <- music[-c(2:5,7,16,19,21:25,30,34)]
music$artist.hottness.label <- as.factor(music$artist.hottness.label)
str(music)

## 'data.frame': 5648 obs. of 22 variables:
## $ artist.hottnesss : num 0.402 0.402 0.332 0.296 0.352 ...
## $ bars_confidence : num 0.643 0.175 0.806 0.873 0.018 0.013 1 0.507 0.125 0.03 ...
## $ beats_confidence : num 0.834 0.883 0.44 0.873 1 0.699 1 0 0.768 1 ...
## $ beats_start : num 0.585 0.136 1.226 0.112 0.429 ...
## $ duration : num 219 210 270 219 245 ...
## $ end_of_fade_in : num 0.247 0.066 5.3 2.125 0.357 ...
## $ familiarity : num 0.582 0.651 0.427 0.36 0.545 ...
## $ key : num 1 2 4 5 7 9 10 7 8 7 ...
## $ key_confidence : num 0.736 0.092 0.717 0.354 0.07 0.205 0 1 0.041 0.725 ...
## $ latitude : num 37.2 37.2 37.2 35.2 37.2 ...
## $ longitude : num -63.9 -63.9 -63.9 -80 -63.9 ...
## $ loudness : num -11.2 -4.5 -13.5 -10.02 -7.54 ...
## $ mode_confidence : num 0.636 0.371 0.652 0.485 0.686 0.305 0.198 0.829 0.516 0.756 ...
## $ start_of_fade_out : num 219 199 259 207 227 ...
## $ tatums_confidence : num 0.779 1 0.121 0.229 0.728 1 0.774 0.377 0.767 0.238 ...
## $ tatums_start : num 0.285 0.136 1.226 0.112 0.173 ...
## $ tempo : num 92.2 129.7 86.6 146.8 118 ...
## $ terms_freq : num 1 0.887 0.96 0.956 1 ...
## $ time_signature : num 4 4 4 1 4 4 1 4 5 4 ...
## $ time_signature_confidence: num 0.778 0.562 0.487 0 0.835 0 0.319 0.756 0.579 0.931 ...
## $ year : int 0 2007 1984 0 0 0 0 1987 0 2004 ...
## $ artist.hottness.label : Factor w/ 5 levels "Cool","Frigid",...: 4 4 1 2 1 1 2 4 1 5 ...

##SONG HOTNESS HISTOGRAM From Jeremy
cmbomusic[cmbomusic==0]<- NA
#cmbomusic2 <- cmbomusic[-c(5,6)]
cmbomusic3 <- na.omit(cmbomusic)
cmbomusic3$song.hottnesss.label <- ifelse( cmbomusic3$song.hottnesss >=0.6011861, "Hot",ifelse(cmbomu
unique(cmbomusic3$song.hottnesss.label)

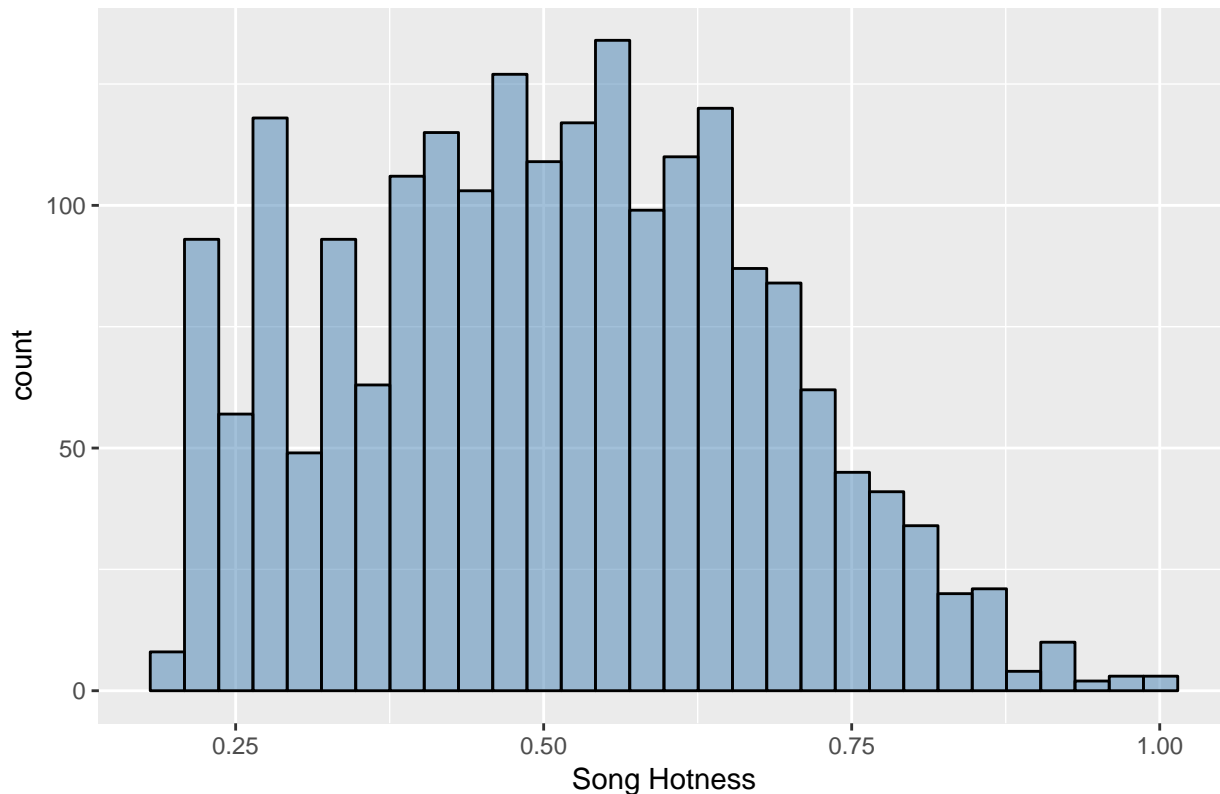
## [1] "Hot" "Tepid" "Cool" "Warm" "Frigid"

cmbomusic3 <- cmbomusic3[-c(2:3,12)]
ggplot(cmbomusic3, aes(x=song.hottnesss)) + geom_histogram(color="black", fill="steelblue", alpha=0.5

## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.

```

# Histogram: Song Hotness



*##Function to create descriptive statistics for song hotness*

```
descriptive_stats2 <- function(vector) { library(moments)
  result <- c(Mean=mean(vector),
              Median=median(vector),
              Min = min(vector),
              Max = max(vector),
              SD = sd(vector),
              Quantile = quantile(vector, probs = c(0.25,.50,0.75, 0.95)),
              Skewness = skewness(vector) )
  print(result)
}
descriptive_stats2(cmbomusic3$song.hotttnesss)
```

```
##      Mean      Median      Min      Max      SD
##  0.5073226  0.5096410  0.1938578  1.0000000  0.1686679
## Quantile.25% Quantile.50% Quantile.75% Quantile.95%  Skewness
##  0.3827233  0.5096410  0.6301876  0.7900643  0.1304601
```

```
cmbomusic3$song.hottness.label <- ifelse( cmbomusic3$song.hotttnesss >=0.64787976, "Hot",ifelse(cmbomusic3$song.hotttnesss < 0.64787976, "Cold", "Warm"))
unique(cmbomusic3$song.hottness.label)
```

```
## [1] "Hot" "Cold" "Warm"
```

```
cmbomusic3$song.hotttnesss.label <- as.factor(cmbomusic3$song.hotttnesss.label)
str(cmbomusic3)
```

```
## 'data.frame': 2037 obs. of 20 variables:
## $ artist.name : Factor w/ 4408 levels ":Blacks On :Blondes",...: 3571 3380 1641 2281 3201 ...
## $ latitude : num 47.6 37.2 53.5 37.2 37.2 ...
```

```
## $ location          : Factor w/ 1043 levels "", " UbA!, Minas Gerais",...: 856 703 557 283 703
## $ longitude         : num  -122.33 -63.93 -2.25 -63.93 -63.93 ...
## $ loudness          : num  -9.31 -6.08 -9.62 -10.54 -14.01 ...
## $ release.id        : int   15964 114401 186364 171807 512792 583091 192588 92902 15316 77794
## $ release.name      : Factor w/ 7829 levels ". . . Till Then",...: 715 5751 1083 3597 921 909
## $ song.hottnesss    : num   0.654 0.43 0.346 1 0.694 ...
## $ song.id           : Factor w/ 9995 levels "SOAAAQN12AB01856D3",...: 3 6 7 11 15 16 19 24 29
## $ tatums_confidence : num   0.898 1 0.445 0.388 0.484 0.873 0.408 0.284 0.992 1 ...
## $ tatums_start      : num   0.1569 0.0346 0.089 0.1008 0.2263 ...
## $ tempo             : num   131 114 102 151 123 ...
## $ terms             : Factor w/ 458 levels "", "8-bit", "acid jazz",...: 10 216 8 37 301 198 10
## $ terms_freq        : num    1 1 1 0.998 0.82 ...
## $ time_signature    : int    4 5 4 3 4 4 4 4 4 3 ...
## $ time_signature_confidence: num   0.59 0.583 0.097 1 0.369 1 1 0.866 0.919 0.741 ...
## $ title             : Factor w/ 9704 levels "", "-start ID-",...: 7342 6931 9501 3916 539 4665
## $ year              : int   1991 2005 1988 1970 1977 2009 2008 2007 1998 2010 ...
## $ song.hottnesss.label : Factor w/ 5 levels "Cool","Frigid",...: 3 4 1 3 3 3 1 3 3 ...
## $ song.hotness.label  : chr   "Hot" "Cold" "Cold" "Hot" ...
```

```
cmbomusic3$song.hottnesss.label <- ifelse( cmbomusic3$song.hottnesss >=0.6011861, "Hot",ifelse(cmbomusic3$song.hottnesss < 0.6011861,
unique(cmbomusic3$song.hottnesss.label)
```

```
## [1] "Hot" "Tepid" "Cool" "Warm" "Frigid"
```

```
cmbomusic3$song.hottnesss.label <- as.factor(cmbomusic3$song.hottnesss.label)
str(cmbomusic3)
```

```
## 'data.frame': 2037 obs. of 20 variables:
## $ artist.name       : Factor w/ 4408 levels ":Blacks On :Blondes",...: 3571 3380 1641 2281 32
## $ latitude          : num   47.6 37.2 53.5 37.2 37.2 ...
## $ location          : Factor w/ 1043 levels "", " UbA!, Minas Gerais",...: 856 703 557 283 703
## $ longitude         : num  -122.33 -63.93 -2.25 -63.93 -63.93 ...
## $ loudness          : num  -9.31 -6.08 -9.62 -10.54 -14.01 ...
## $ release.id        : int   15964 114401 186364 171807 512792 583091 192588 92902 15316 77794
## $ release.name      : Factor w/ 7829 levels ". . . Till Then",...: 715 5751 1083 3597 921 909
## $ song.hottnesss    : num   0.654 0.43 0.346 1 0.694 ...
## $ song.id           : Factor w/ 9995 levels "SOAAAQN12AB01856D3",...: 3 6 7 11 15 16 19 24 29
## $ tatums_confidence : num   0.898 1 0.445 0.388 0.484 0.873 0.408 0.284 0.992 1 ...
## $ tatums_start      : num   0.1569 0.0346 0.089 0.1008 0.2263 ...
## $ tempo             : num   131 114 102 151 123 ...
## $ terms             : Factor w/ 458 levels "", "8-bit", "acid jazz",...: 10 216 8 37 301 198 10
## $ terms_freq        : num    1 1 1 0.998 0.82 ...
## $ time_signature    : int    4 5 4 3 4 4 4 4 4 3 ...
## $ time_signature_confidence: num   0.59 0.583 0.097 1 0.369 1 1 0.866 0.919 0.741 ...
## $ title             : Factor w/ 9704 levels "", "-start ID-",...: 7342 6931 9501 3916 539 4665
## $ year              : int   1991 2005 1988 1970 1977 2009 2008 2007 1998 2010 ...
## $ song.hottnesss.label : Factor w/ 5 levels "Cool","Frigid",...: 3 4 1 3 3 3 1 3 3 ...
## $ song.hotness.label  : chr   "Hot" "Cold" "Cold" "Hot" ...
```

```
cmbomusic3$song.hottnesss.label <- ifelse( cmbomusic3$song.hottnesss >=0.6011861, "Hot",ifelse(cmbomusic3$song.hottnesss < 0.6011861,
unique(cmbomusic3$song.hottnesss.label)
```

```
## [1] "Hot" "Tepid" "Cool" "Warm" "Frigid"
```

```
str(cmbomusic3)
```

```
## 'data.frame': 2037 obs. of 20 variables:
```

```
## $ artist.name      : Factor w/ 4408 levels ":Blacks On :Blondes",...: 3571 3380 1641 2281 32
## $ latitude         : num  47.6 37.2 53.5 37.2 37.2 ...
## $ location         : Factor w/ 1043 levels "", " UbA!, Minas Gerais",...: 856 703 557 283 703
## $ longitude        : num  -122.33 -63.93 -2.25 -63.93 -63.93 ...
## $ loudness         : num  -9.31 -6.08 -9.62 -10.54 -14.01 ...
## $ release.id       : int   15964 114401 186364 171807 512792 583091 192588 92902 15316 77794
## $ release.name     : Factor w/ 7829 levels ". . . Till Then",...: 715 5751 1083 3597 921 909
## $ song.hottnesss   : num   0.654 0.43 0.346 1 0.694 ...
## $ song.id          : Factor w/ 9995 levels "SOAAAQN12AB01856D3",...: 3 6 7 11 15 16 19 24 29
## $ tatums_confidence : num   0.898 1 0.445 0.388 0.484 0.873 0.408 0.284 0.992 1 ...
## $ tatums_start     : num   0.1569 0.0346 0.089 0.1008 0.2263 ...
## $ tempo            : num   131 114 102 151 123 ...
## $ terms            : Factor w/ 458 levels "", "8-bit", "acid jazz",...: 10 216 8 37 301 198 10
## $ terms_freq       : num   1 1 1 0.998 0.82 ...
## $ time_signature    : int    4 5 4 3 4 4 4 4 3 ...
## $ time_signature_confidence: num   0.59 0.583 0.097 1 0.369 1 1 0.866 0.919 0.741 ...
## $ title            : Factor w/ 9704 levels "", "-start ID-",...: 7342 6931 9501 3916 539 4665
## $ year             : int   1991 2005 1988 1970 1977 2009 2008 2007 1998 2010 ...
## $ song.hottnesss.label : chr   "Hot" "Tepid" "Cool" "Hot" ...
## $ song.hotness.label  : chr   "Hot" "Cold" "Cold" "Hot" ...
```

```
cmbomusic3$song.hottnesss.label <- as.factor(cmbomusic3$song.hottnesss.label)
cmbomusic3$song.hottnesss.label <- as.factor(cmbomusic3$song.hottnesss.label)
str(cmbomusic3)
```

```
## 'data.frame':    2037 obs. of  20 variables:
## $ artist.name      : Factor w/ 4408 levels ":Blacks On :Blondes",...: 3571 3380 1641 2281 32
## $ latitude         : num  47.6 37.2 53.5 37.2 37.2 ...
## $ location         : Factor w/ 1043 levels "", " UbA!, Minas Gerais",...: 856 703 557 283 703
## $ longitude        : num  -122.33 -63.93 -2.25 -63.93 -63.93 ...
## $ loudness         : num  -9.31 -6.08 -9.62 -10.54 -14.01 ...
## $ release.id       : int   15964 114401 186364 171807 512792 583091 192588 92902 15316 77794
## $ release.name     : Factor w/ 7829 levels ". . . Till Then",...: 715 5751 1083 3597 921 909
## $ song.hottnesss   : num   0.654 0.43 0.346 1 0.694 ...
## $ song.id          : Factor w/ 9995 levels "SOAAAQN12AB01856D3",...: 3 6 7 11 15 16 19 24 29
## $ tatums_confidence : num   0.898 1 0.445 0.388 0.484 0.873 0.408 0.284 0.992 1 ...
## $ tatums_start     : num   0.1569 0.0346 0.089 0.1008 0.2263 ...
## $ tempo            : num   131 114 102 151 123 ...
## $ terms            : Factor w/ 458 levels "", "8-bit", "acid jazz",...: 10 216 8 37 301 198 10
## $ terms_freq       : num   1 1 1 0.998 0.82 ...
## $ time_signature    : int    4 5 4 3 4 4 4 4 3 ...
## $ time_signature_confidence: num   0.59 0.583 0.097 1 0.369 1 1 0.866 0.919 0.741 ...
## $ title            : Factor w/ 9704 levels "", "-start ID-",...: 7342 6931 9501 3916 539 4665
## $ year             : int   1991 2005 1988 1970 1977 2009 2008 2007 1998 2010 ...
## $ song.hottnesss.label : Factor w/ 5 levels "Cool","Frigid",...: 3 4 1 3 3 3 1 3 3 ...
## $ song.hotness.label  : chr   "Hot" "Cold" "Cold" "Hot" ...
```

*#View the number of Cold/Warm/Hot labels*

```
table(cmbomusic3$song.hottnesss.label)
```

```
##
##   Cool Frigid   Hot  Tepid   Warm
##   171   337   629   278   622
```

```
cmbomusic3$song.hotness.label <- ifelse( cmbomusic3$song.hottnesss >=0.64787976, "Hot",ifelse(cmbomusic3$song.hottnesss < 0.64787976, "Cold", "Warm"))
unique(cmbomusic3$song.hotness.label)
```



```
## [1] "Hot" "Cold" "Warm"
```

## Features

```
#View the number of Cold/Warm/Hot labels
```

```
table(music$artist.hotttnesss.label)
```

```
## < table of extent 0 >
```

```
#View the number of Frigid/Cool/Tepid/Warm/Hot labels
```

```
table(music$artist.hottness.label)
```

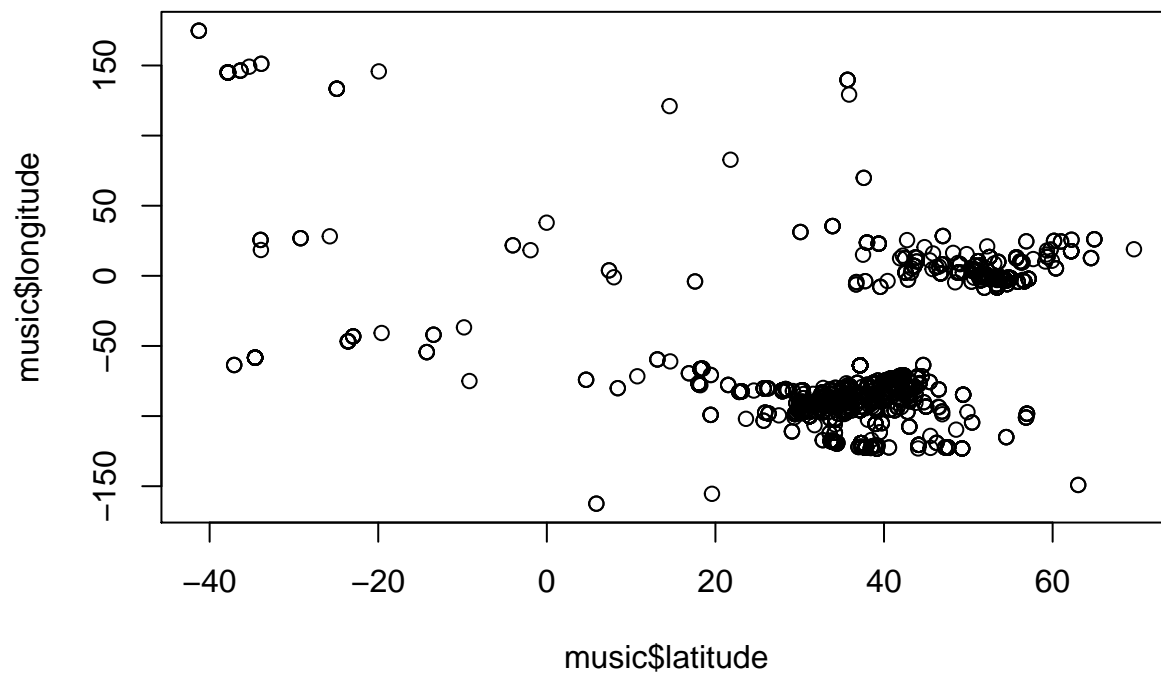
```
##
```

```
##   Cool Frigid   Hot  Tepid   Warm
```

```
##  1444   973   278  1566  1387
```

```
#Plot artists latitude and longitude
```

```
plot(music$latitude,music$longitude)
```



```
cmbomusic3$song.hottness.label <- as.factor(cmbomusic3$song.hottness.label)
```

```
cmbomusic3$song.hottness.label <- as.factor(cmbomusic3$song.hottness.label)
```

```
str(cmbomusic3)
```

```
## 'data.frame':   2037 obs. of  20 variables:
```

```
##  $ artist.name      : Factor w/ 4408 levels ":Blacks On :Blondes",...: 3571 3380 1641 2281 32
```

```
##  $ latitude         : num  47.6 37.2 53.5 37.2 37.2 ...
```

```
##  $ location         : Factor w/ 1043 levels ""," UbA!, Minas Gerais",...: 856 703 557 283 703
```

```
##  $ longitude        : num  -122.33 -63.93 -2.25 -63.93 -63.93 ...
```

```
##  $ loudness         : num  -9.31 -6.08 -9.62 -10.54 -14.01 ...
```

```
##  $ release.id       : int   15964 114401 186364 171807 512792 583091 192588 92902 15316 77794
```

```
##  $ release.name     : Factor w/ 7829 levels ". . . Till Then",...: 715 5751 1083 3597 921 909
```

```
##  $ song.hotttnesss  : num   0.654 0.43 0.346 1 0.694 ...
```

```
##  $ song.id          : Factor w/ 9995 levels "SOAAAQN12AB01856D3",...: 3 6 7 11 15 16 19 24 29
```

```
##  $ tatums_confidence : num   0.898 1 0.445 0.388 0.484 0.873 0.408 0.284 0.992 1 ...
```

```
## $ tatums_start      : num  0.1569 0.0346 0.089 0.1008 0.2263 ...
## $ tempo             : num  131 114 102 151 123 ...
## $ terms             : Factor w/ 458 levels "", "8-bit", "acid jazz", ...: 10 216 8 37 301 198 107
## $ terms_freq       : num  1 1 1 0.998 0.82 ...
## $ time_signature    : int   4 5 4 3 4 4 4 4 4 3 ...
## $ time_signature_confidence: num  0.59 0.583 0.097 1 0.369 1 1 0.866 0.919 0.741 ...
## $ title            : Factor w/ 9704 levels "", "-start ID-", ...: 7342 6931 9501 3916 539 4665
## $ year             : int   1991 2005 1988 1970 1977 2009 2008 2007 1998 2010 ...
## $ song.hotttnesss.label : Factor w/ 5 levels "Cool","Frigid",...: 3 4 1 3 3 3 1 3 3 ...
## $ song.hotness.label  : Factor w/ 3 levels "Cold","Hot","Warm": 2 1 1 2 2 2 2 1 3 2 ...
```

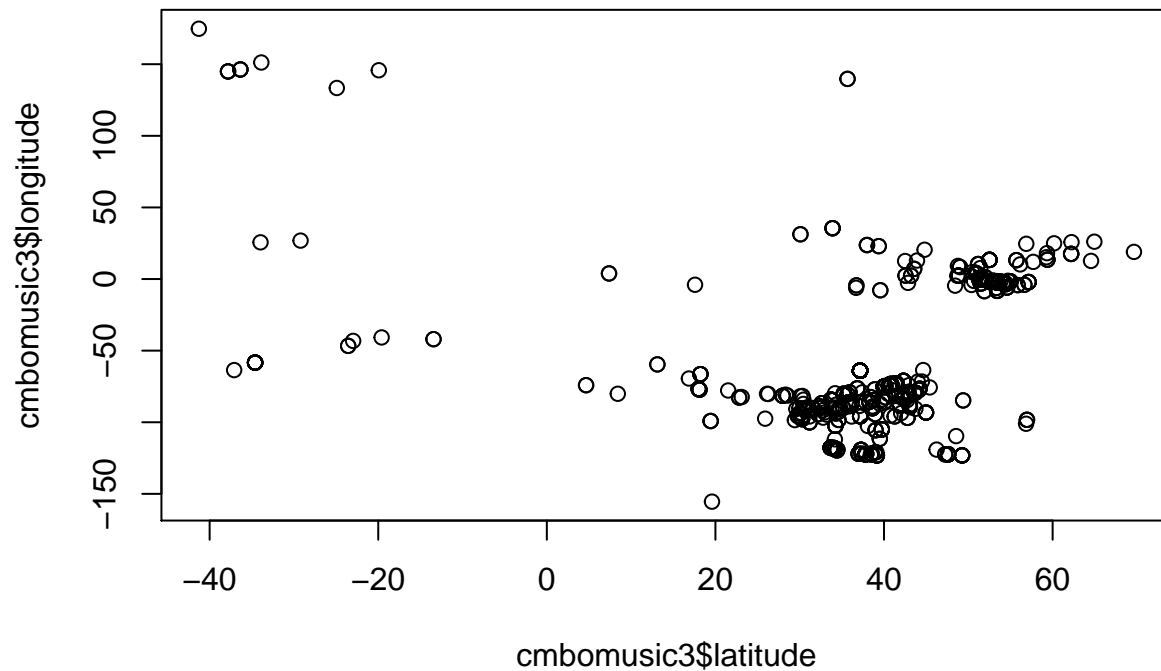
```
#View the number of Cold/Warm/Hot labels
table(cmbomusic3$song.hotness.label)
```

```
##
## Cold Hot Warm
## 707 440 890
```

```
#View the number of Frigid/Cool/Tepid/Warm/Hot labels
table(cmbomusic3$song.hotttnesss.label)
```

```
##
## Cool Frigid Hot Tepid Warm
## 171 337 629 278 622
```

```
#Plot artists latitude and longitude
plot(cmbomusic3$latitude, cmbomusic3$longitude)
```



```
#Plot artist hotttnesss
#hist(music$artist.hotttnesss, breaks=20)
#hist(music$artist.hottness, breaks=20)
```

```
#Create a map of the world
World <- borders("world", colour="gray50", fill="white")
```

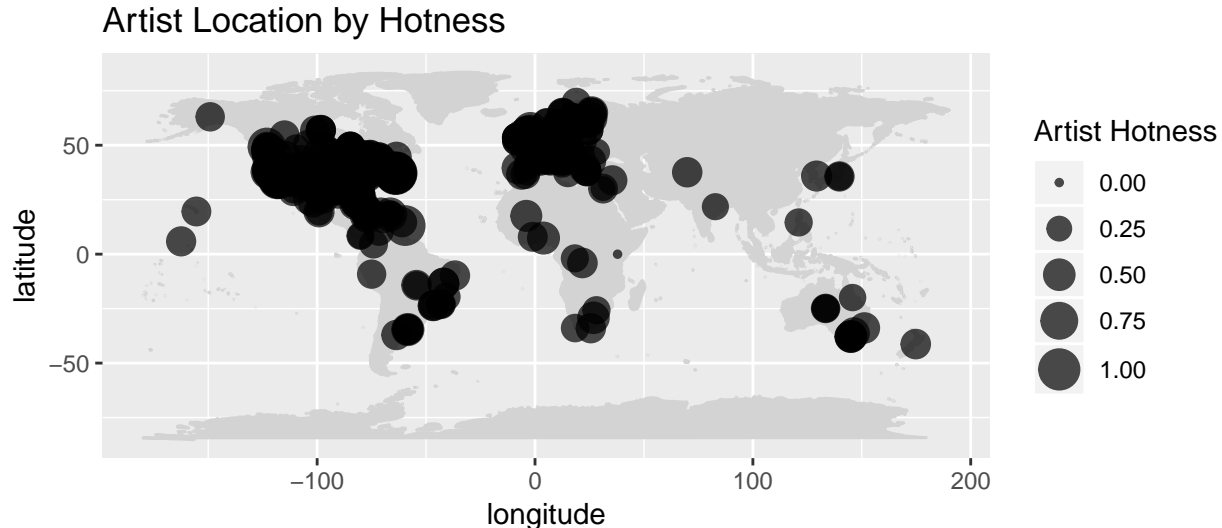
```
#New code from John for creating a map of the world showing latitude/longitude and artist hotness
#Code based on info from https://rpubs.com/spoonerf/global_map
library(dplyr)
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
##
##   filter, lag

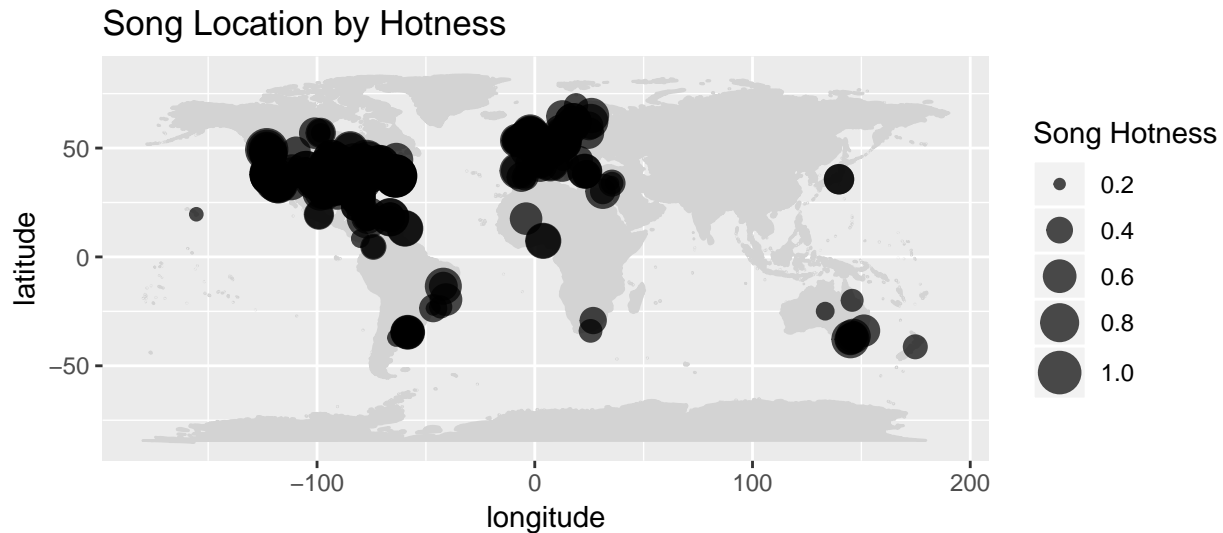
## The following objects are masked from 'package:base':
##
##   intersect, setdiff, setequal, union
```

```
loc<-data.frame(music$longitude,music$latitude,music$artist.hottness)
loc<-unique(loc)
colnames(loc)<-c("longitude", "latitude","artist hotness")
loc_df<-data.frame(loc)
library(maps)
library(mapdata)
library(ggplot2)
ahworld <- ggplot(data=loc_df, aes(longitude, latitude, group=NULL,fill=NULL,size=artist.hotness))+#, f
  borders(fill="light grey",colour="light grey")+
  geom_point(color="black",alpha=I(7/10))+
  scale_size(range=c(1,7), guide = "legend",labs(size="Artist Hotness"))+
  coord_equal()+ ggtitle("Artist Location by Hotness")
ahworld
```



```
#New code from John for creating a map of the world showing latitude/longitude and artist hotness
#Code based on info from https://rpubs.com/spoonerf/global_map
library(dplyr)
songlc<-data.frame(cmbomusic3$longitude,cmbomusic3$latitude,cmbomusic3$song.hottness)
songlc<-unique(songlc)
colnames(songlc)<-c("longitude", "latitude","song hotness")
songlc_df<-data.frame(songlc)
library(maps)
library(mapdata)
```

```
library(ggplot2)
songlc_dfwrld <- ggplot(data=songlc_df, aes(longitude, latitude, group=NULL, fill=NULL, size=song.hotness,
  borders(fill="light grey", colour="light grey")+
  geom_point(color="black", alpha=I(7/10))+
  scale_size(range=c(1,7), guide = "legend", labs(size="Song Hotness"))+
  coord_equal()+ ggtitle("Song Location by Hotness")
songlc_dfwrld
```



## Methods - Linear Regression

```
# code from Juan
lm(formula = music$artist.hottnesss ~ music$year + music$bars_confidence +
  + music$tempo + music$duration + music$start_of_fade_out +
  music$atums_start + music$familiarity + music$latitude +
  music$tempo + music$longitude + music$beats_start + music$beats_confidence +
  music$end_of_fade_in)

##
## Call:
## lm(formula = music$artist.hottnesss ~ music$year + music$bars_confidence +
##   +music$tempo + music$duration + music$start_of_fade_out +
##   music$atums_start + music$familiarity + music$latitude +
##   music$tempo + music$longitude + music$beats_start + music$beats_confidence +
##   music$end_of_fade_in)
##
## Coefficients:
##              (Intercept)              music$year      music$bars_confidence
##              1.500e-02              6.911e-06              -4.242e-04
##              music$tempo              music$duration      music$start_of_fade_out
##              -3.122e-05              1.842e-05              -2.842e-05
##              music$atums_start      music$familiarity              music$latitude
##              -5.004e-03              6.625e-01              -1.039e-04
##              music$longitude      music$beats_start      music$beats_confidence
##              -5.606e-05              5.494e-03              -2.277e-03
##              music$end_of_fade_in
```

```
## 9.355e-05
#removed music$bars_start which was causing an error
```

## Methods - Random Forest

```
#Do analysis to determine hot/warm/cold artists based on hotttnesss
#The random forest analysis is from a training video by Bharatendra Rai
#at https://www.youtube.com/watch?v=dJclNIN-TPo
#Data Partition - ind = independent samples
#The code below runs in console but not R Markdown
set.seed(123)
ind<- sample(2,nrow(music), replace=TRUE,prob=c(0.7,0.3))
train <- music[ind==1,]
test <- music[ind==2,]
#Run randomForest on 3 levels
library(randomForest)
```

```
## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
##
## The following object is masked from 'package:dplyr':
##
##   combine
##
## The following object is masked from 'package:ggplot2':
##
##   margin
set.seed(222)
rf <- randomForest(music[, -21:-22], music[, 21])
print(rf)
```

```
##
## Call:
## randomForest(x = music[, -21:-22], y = music[, 21])
##           Type of random forest: regression
##           Number of trees: 500
## No. of variables tried at each split: 6
##
##           Mean of squared residuals: 826953
##           % Var explained: 16.59
```

```
attributes(rf)
```

```
## $names
## [1] "call"           "type"           "predicted"
## [4] "mse"            "rsq"            "oob.times"
## [7] "importance"     "importanceSD"   "localImportance"
## [10] "proximity"      "ntree"          "mtry"
## [13] "forest"         "coefs"          "y"
## [16] "test"          "inbag"
##
```

```
## $class
## [1] "randomForest"

rf$confusion

## NULL

#Run randomForest on 5 levels
library(randomForest)
set.seed(222)
rf2 <- randomForest(music[, -21:-22], music[, 22])
print(rf2)

##
## Call:
## randomForest(x = music[, -21:-22], y = music[, 22])
##           Type of random forest: classification
##           Number of trees: 500
## No. of variables tried at each split: 4
##
##           OOB estimate of  error rate: 0.18%
## Confusion matrix:
##           Cool Frigid Hot Tepid Warm class.error
## Cool      1444      0  0      0      0 0.000000000
## Frigid      0      973  0      0      0 0.000000000
## Hot          0      0 270      0      8 0.028776978
## Tepid        0      0  0 1566      0 0.000000000
## Warm         0      0  0      2 1385 0.001441961

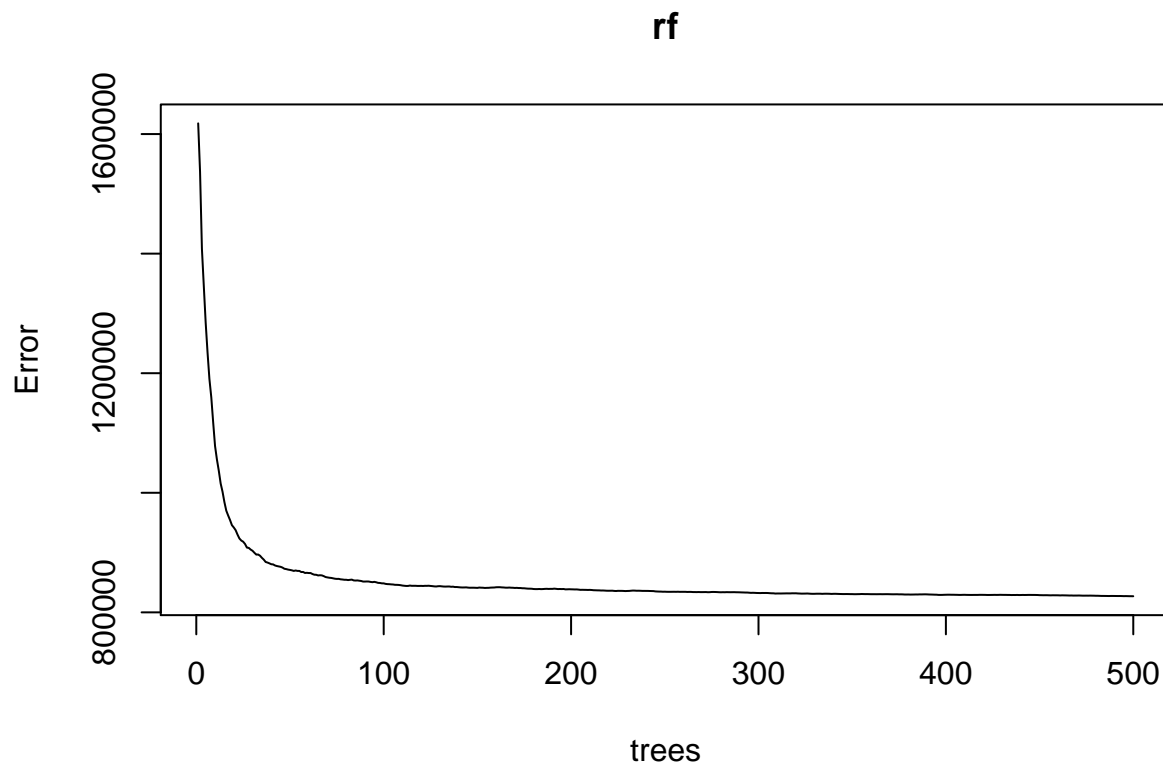
attributes(rf2)

## $names
## [1] "call"           "type"           "predicted"
## [4] "err.rate"       "confusion"      "votes"
## [7] "oob.times"      "classes"        "importance"
## [10] "importanceSD"   "localImportance" "proximity"
## [13] "ntree"          "mtry"           "forest"
## [16] "y"              "test"           "inbag"
##
## $class
## [1] "randomForest"

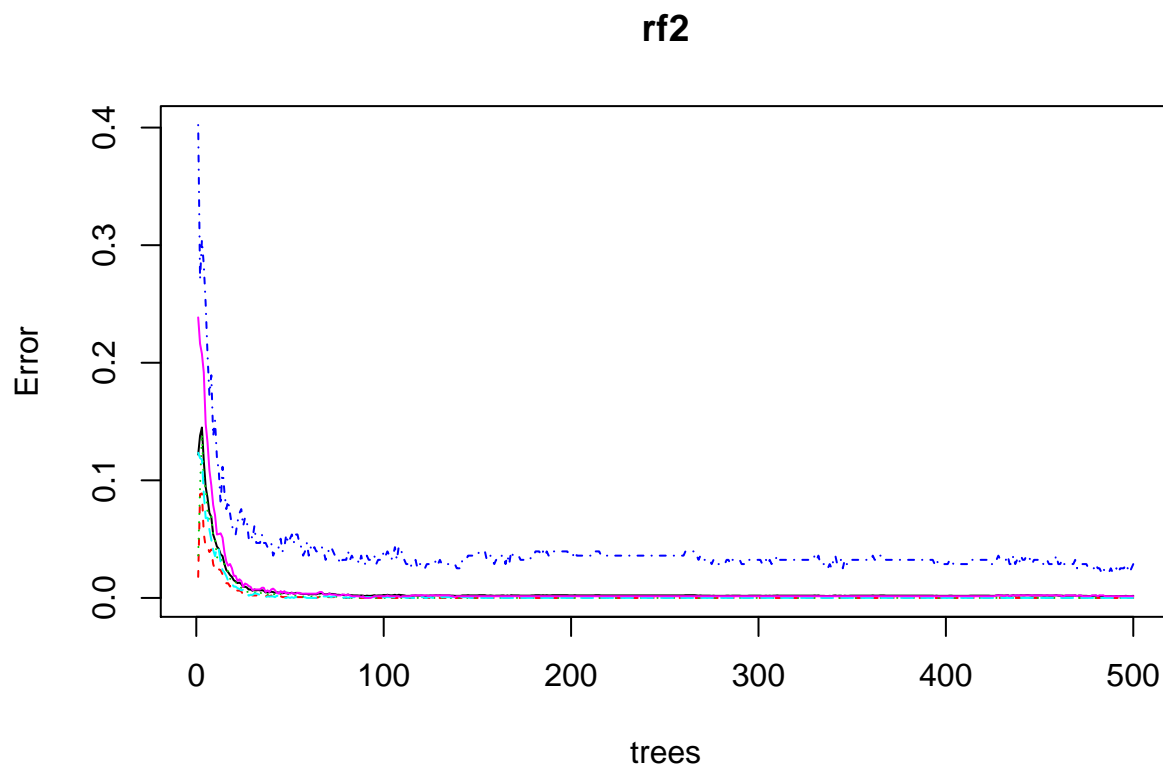
rf2$confusion

##           Cool Frigid Hot Tepid Warm class.error
## Cool      1444      0  0      0      0 0.000000000
## Frigid      0      973  0      0      0 0.000000000
## Hot          0      0 270      0      8 0.028776978
## Tepid        0      0  0 1566      0 0.000000000
## Warm         0      0  0      2 1385 0.001441961

#Error rate of Random Forest
plot(rf)
```



```
plot(rf2)
```



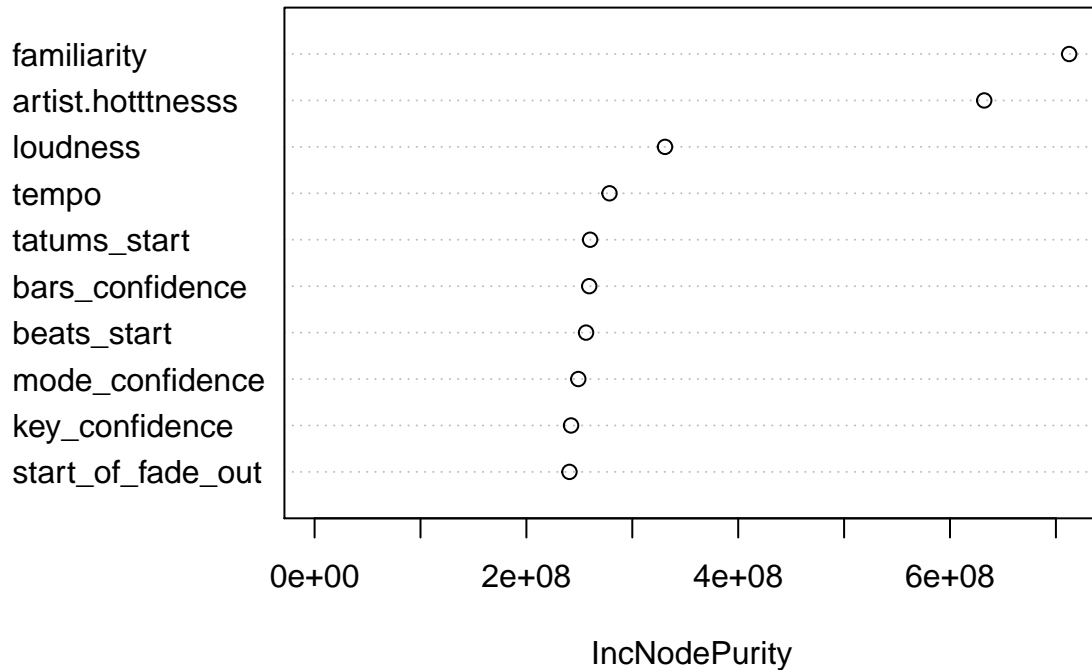
*#The error rate is not improving after ~100 trees*

*# Variable Importance*

*# Familiarity is much more important than the other variables.*

```
varImpPlot(rf,
           sort=T,
           n.var=10,
           main="Top 10 - Variable Importance")
```

## Top 10 – Variable Importance



```
importance(rf)
```

##	IncNodePurity
## artist.hotttnesss	632328644
## bars_confidence	259319249
## beats_confidence	206820915
## beats_start	256324242
## duration	240237155
## end_of_fade_in	192120434
## familiarity	712524089
## key	143890690
## key_confidence	242165770
## latitude	153094783
## longitude	149771653
## loudness	330880900
## mode_confidence	248999376
## start_of_fade_out	240653115
## tatums_confidence	231190858
## tatums_start	260177689
## tempo	278464617
## terms_freq	214067856
## time_signature	64162932
## time_signature_confidence	180372321

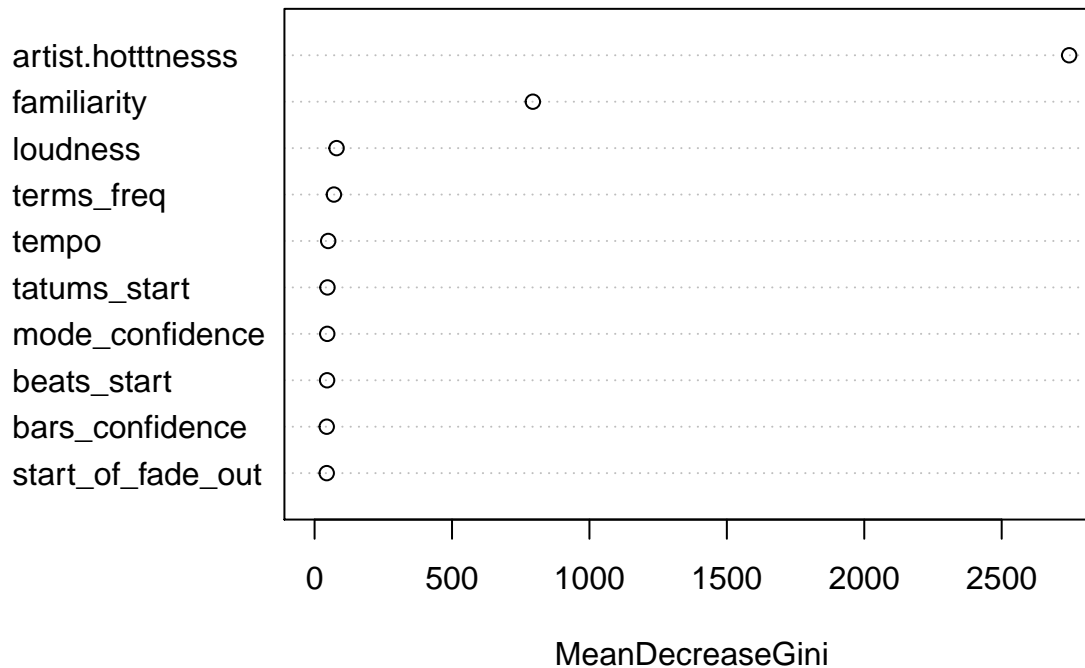


```
varUsed(rf)
```

```
## [1] 53861 51368 44324 50446 48698 41054 55164 37123 48813 29847 28240  
## [12] 55430 49848 47951 47006 50713 52172 34675 20506 40779
```

```
varImpPlot(rf2,  
            sort=T,  
            n.var=10,  
            main="Top 10 - Variable Importance")
```

## Top 10 – Variable Importance



```
importance(rf2)
```

```
##               MeanDecreaseGini  
## artist.hotttnesss      2745.57471  
## bars_confidence       44.14672  
## beats_confidence      40.36867  
## beats_start           45.50021  
## duration              43.16086  
## end_of_fade_in        39.30387  
## familiarity           794.13225  
## key                   27.13917  
## key_confidence        41.77856  
## latitude              38.36968  
## longitude             39.90725  
## loudness              80.02057  
## mode_confidence       46.08476  
## start_of_fade_out     44.13825  
## tatums_confidence     39.80128  
## tatums_start          46.82259  
## tempo                 49.53280  
## terms_freq            70.28740
```

```

## time_signature          12.79758
## time_signature_confidence 33.38332

varUsed(rf2)

## [1] 29551 8173 7342 8318 8064 6548 16205 5780 7942 5273 5429
## [12] 9726 8386 8120 7533 8555 8940 7161 2773 6443

cmbomusic4 <- na.omit(cmbomusic3)
cmbomusic5 <- cmbomusic4[-c(1,3,7:9,13,17,20)]
str(cmbomusic5)

## 'data.frame': 2037 obs. of 12 variables:
## $ latitude : num 47.6 37.2 53.5 37.2 37.2 ...
## $ longitude : num -122.33 -63.93 -2.25 -63.93 -63.93 ...
## $ loudness : num -9.31 -6.08 -9.62 -10.54 -14.01 ...
## $ release.id : int 15964 114401 186364 171807 512792 583091 192588 92902 15316 77794
## $ tatums_confidence : num 0.898 1 0.445 0.388 0.484 0.873 0.408 0.284 0.992 1 ...
## $ tatums_start : num 0.1569 0.0346 0.089 0.1008 0.2263 ...
## $ tempo : num 131 114 102 151 123 ...
## $ terms_freq : num 1 1 1 0.998 0.82 ...
## $ time_signature : int 4 5 4 3 4 4 4 4 4 3 ...
## $ time_signature_confidence : num 0.59 0.583 0.097 1 0.369 1 1 0.866 0.919 0.741 ...
## $ year : int 1991 2005 1988 1970 1977 2009 2008 2007 1998 2010 ...
## $ song.hottnesss.label : Factor w/ 5 levels "Cool","Frigid",...: 3 4 1 3 3 3 3 1 3 3 ...

cmbomusic5$song.hottnesss.label <- as.factor(cmbomusic4$song.hottnesss.label)
rf3 <- randomForest(cmbomusic5[, -12:-13], cmbomusic5[, 13])
rf3

##
## Call:
## randomForest(x = cmbomusic5[, -12:-13], y = cmbomusic5[, 13])
## Type of random forest: classification
## Number of trees: 500
## No. of variables tried at each split: 3
##
## OOB estimate of error rate: 51.1%
## Confusion matrix:
## Cold Hot Warm class.error
## Cold 303 35 369 0.5714286
## Hot 63 119 258 0.7295455
## Warm 243 73 574 0.3550562

print(rf3)

##
## Call:
## randomForest(x = cmbomusic5[, -12:-13], y = cmbomusic5[, 13])
## Type of random forest: classification
## Number of trees: 500
## No. of variables tried at each split: 3
##
## OOB estimate of error rate: 51.1%
## Confusion matrix:
## Cold Hot Warm class.error
## Cold 303 35 369 0.5714286

```

```
## Hot      63 119 258 0.7295455
## Warm    243  73 574 0.3550562
```

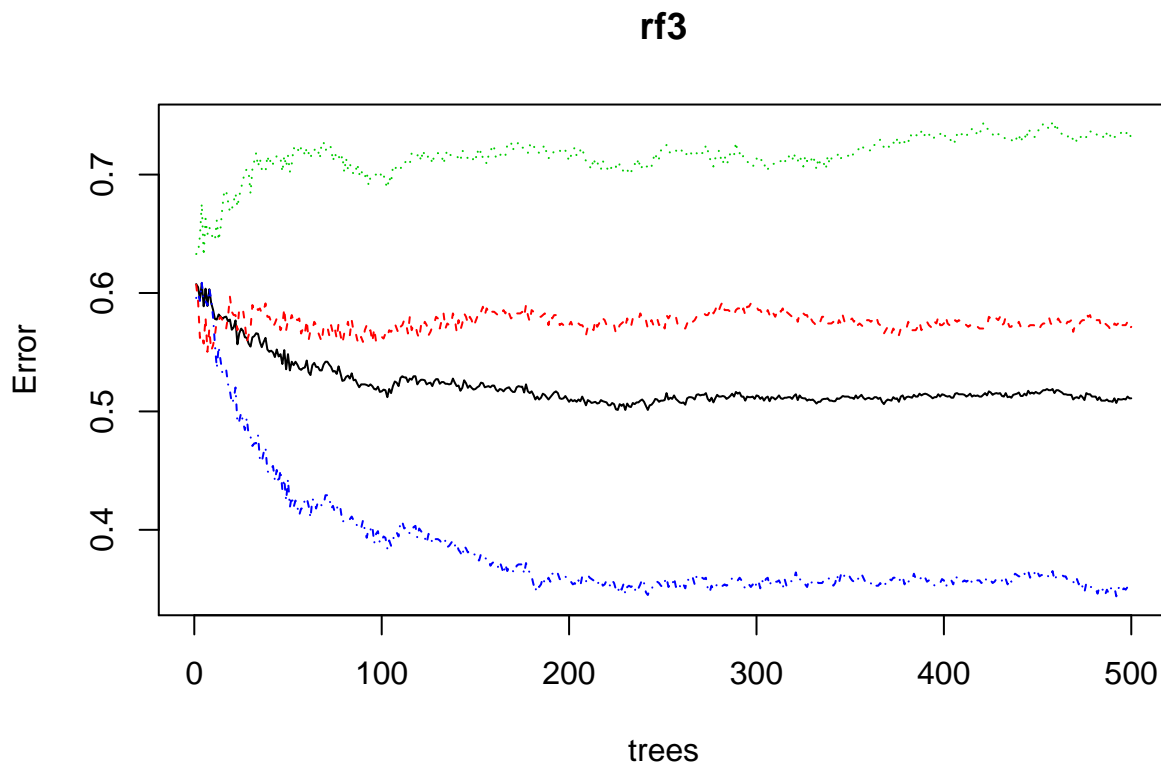
```
attributes(rf3)
```

```
## $names
## [1] "call"          "type"          "predicted"
## [4] "err.rate"      "confusion"     "votes"
## [7] "oob.times"     "classes"       "importance"
## [10] "importanceSD"  "localImportance" "proximity"
## [13] "ntree"         "mtry"          "forest"
## [16] "y"            "test"          "inbag"
##
## $class
## [1] "randomForest"
```

```
rf3$confusion
```

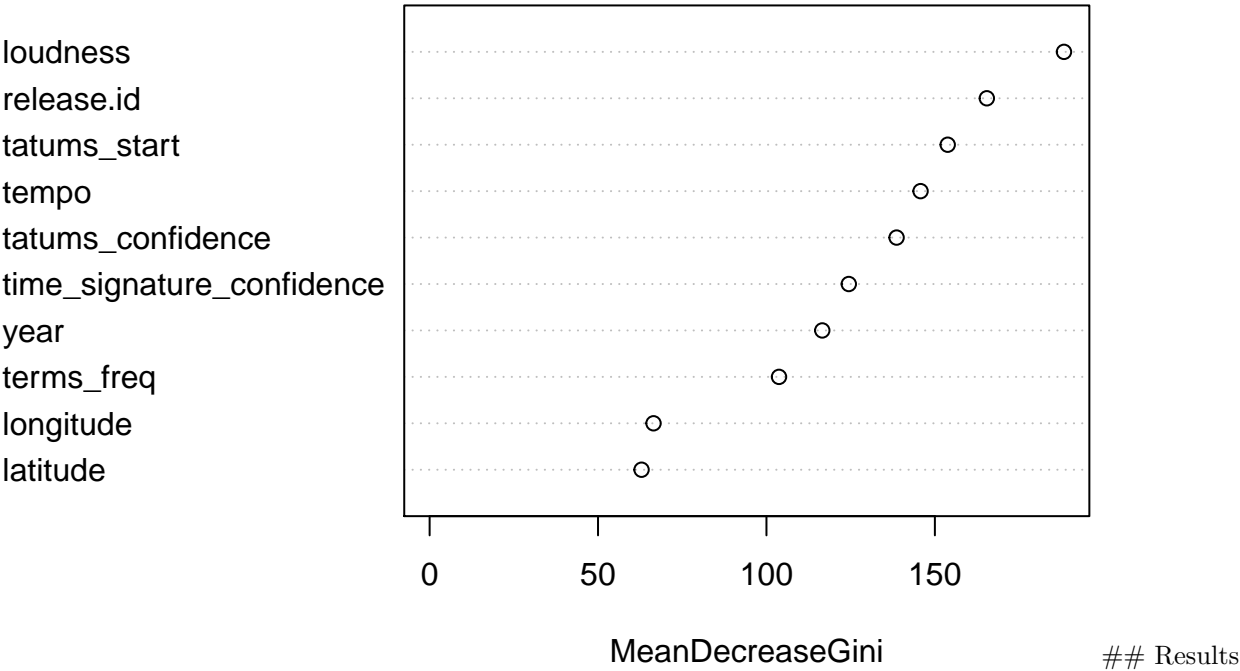
```
##      Cold Hot Warm class.error
## Cold  303  35 369  0.5714286
## Hot   63 119 258  0.7295455
## Warm  243  73 574  0.3550562
```

```
plot(rf3)
```



```
varImpPlot(rf3,
  sort=T,
  n.var=10,
  main="Top 10 - Variable Importance")
```

Top 10 – Variable Importance



Conclusion

Appendices