

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("/Users/johnfields/Library/Mobile Documents/com~apple~CloudDocs/Syracuse/IST687/GitHub/Intro data science/Music project/newmusic.csv")
#music <- read.csv("~/Intro data science/Music project/newmusic.csv")
#setwd("X:/Users/Courtney/Downloads")
#music <- read.csv("music.csv")
str(music)
```

```
## 'data.frame': 9996 obs. of 36 variables:
## $ artist.hotttnesss : num 0.402 0.417 0.343 0.454 0.402 ...
## $ artist.id : Factor w/ 3885 levels "AR009211187B989185",...: 1269 2353 2168 715 3606 ...
## $ artist.name : Factor w/ 4409 levels ":Blacks On :Blondes",...: 682 3796 3560 67 1569 ...
## $ artist_mbtags : Factor w/ 277 levels "", "0.333", "60s",...: 1 52 1 262 1 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/ 7830 levels " Lazy Afternoon En Anglais",...: 2191 1746 3535 ...
```

```
## $ similar : Factor w/ 2837 levels "AR00K8N11C8A41687B",...: 2408 2225 1145 304 2331
## $ song.hottnesss : num 0.602 NA NA NA 0.605 ...
## $ song.id : Factor w/ 9996 levels " Polovtsian Dances / Rimsky-Korsakov: Russian E
## $ 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/ 9705 levels "", " -start ID-",...: 3572 7526 481 7474 2531 828
## $ year : int 0 1969 0 1982 2007 0 0 0 1984 0 ...
## $ artist.hottnesss.label : Factor w/ 3 levels "Cold", "Hot", "Warm": 3 3 3 3 3 3 1 2 1 3 ...
```

```
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)
#understand the structure of the data
#install.packages("psych")
library(psych)
```

```
##
## Attaching package: 'psych'

## The following objects are masked from 'package:ggplot2':
##
## %+%, alpha
```

```
describeBy(music,)
```

```
## Warning in describeBy(music, ): no grouping variable requested
```

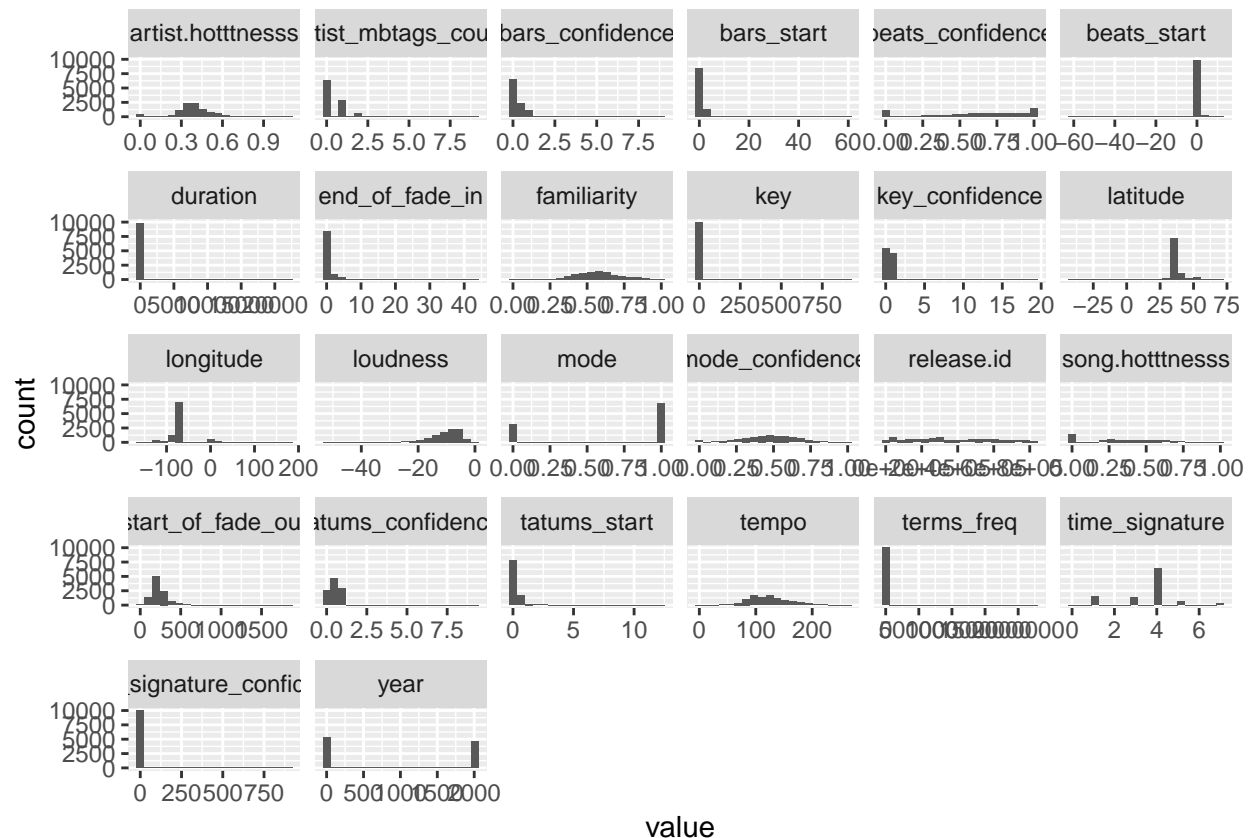
##		vars	n	mean	sd	median	trimmed	mad	min	
##	artist.hottnesss	1	9996	0.39	0.14	0.38	0.39	0.09	0.00	
##	artist.id*	2	9996	1905.96	1122.20	1881.50	1900.72	1429.23	1.00	388
##	artist.name*	3	9996	2205.03	1269.94	2194.00	2206.58	1623.45	1.00	440
##	artist_mbtags*	4	9996	50.06	79.27	1.00	33.38	0.00	1.00	27
##	artist_mbtags_count	5	9996	0.52	0.88	0.00	0.34	0.00	0.00	9
##	bars_confidence	6	9996	0.24	0.29	0.12	0.19	0.15	0.00	8
##	bars_start	7	9996	1.07	1.72	0.79	0.84	0.57	0.00	55
##	beats_confidence	8	9996	0.61	0.32	0.69	0.64	0.33	0.00	7
##	beats_start	9	9996	0.43	0.81	0.33	0.35	0.22	-60.00	11
##	duration	10	9996	240.63	246.13	223.06	226.88	73.82	1.04	2205
##	end_of_fade_in	11	9996	0.76	1.86	0.20	0.33	0.30	0.00	41
##	familiarity	12	9996	0.57	0.16	0.56	0.57	0.15	0.00	1
##	key	13	9996	5.37	9.67	5.00	5.25	4.45	0.00	90
##	key_confidence	14	9996	0.45	0.33	0.47	0.45	0.31	0.00	11
##	latitude	15	9996	37.16	9.54	37.16	37.45	0.00	-41.28	69

## location*	16	9996	596.91	238.96	705.00	616.26	65.23	1.00	104
## longitude	17	9996	-63.93	30.90	-63.93	-67.56	0.00	-162.44	17
## loudness	18	9996	-10.49	5.40	-9.38	-9.84	4.75	-51.64	0
## mode	19	9996	0.69	0.46	1.00	0.74	0.00	0.00	1
## mode_confidence	20	9996	0.48	0.19	0.49	0.48	0.18	0.00	1
## release.id	21	9996	370953.83	236766.22	333100.50	364626.38	294894.33	0.00	82359
## release.name*	22	9996	3921.32	2257.12	3902.50	3923.69	2899.22	1.00	783
## similar*	23	9996	1416.79	822.48	1402.00	1418.57	1077.85	1.00	283
## song.hottnesss	24	5648	0.34	0.25	0.36	0.34	0.27	0.00	1
## song.id*	25	9996	4998.50	2885.74	4998.50	4998.50	3705.02	1.00	999
## start_of_fade_out	26	9996	229.89	112.04	213.88	218.25	71.55	-21.39	181
## tatums_confidence	27	9996	0.51	0.33	0.50	0.51	0.40	0.00	9
## tatums_start	28	9996	0.30	0.51	0.19	0.21	0.13	0.00	11
## tempo	29	9996	122.90	35.20	120.16	121.09	34.87	0.00	26
## terms*	30	9996	215.27	129.18	214.00	212.02	169.02	1.00	45
## terms_freq	31	9996	224.98	22396.64	1.00	0.98	0.00	0.00	223921
## time_signature	32	9996	3.56	1.27	4.00	3.65	0.00	0.00	7
## time_signature_confidence	33	9996	0.60	8.99	0.55	0.51	0.53	0.00	89
## title*	34	9996	4863.28	2799.10	4859.50	4864.67	3586.41	1.00	970
## year	35	9996	935.08	996.67	0.00	917.89	0.00	0.00	201
## artist.hottnesss.label*	36	9996	2.19	0.85	2.00	2.24	1.48	1.00	3

```
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 4348 rows containing non-finite values (stat_bin).
```



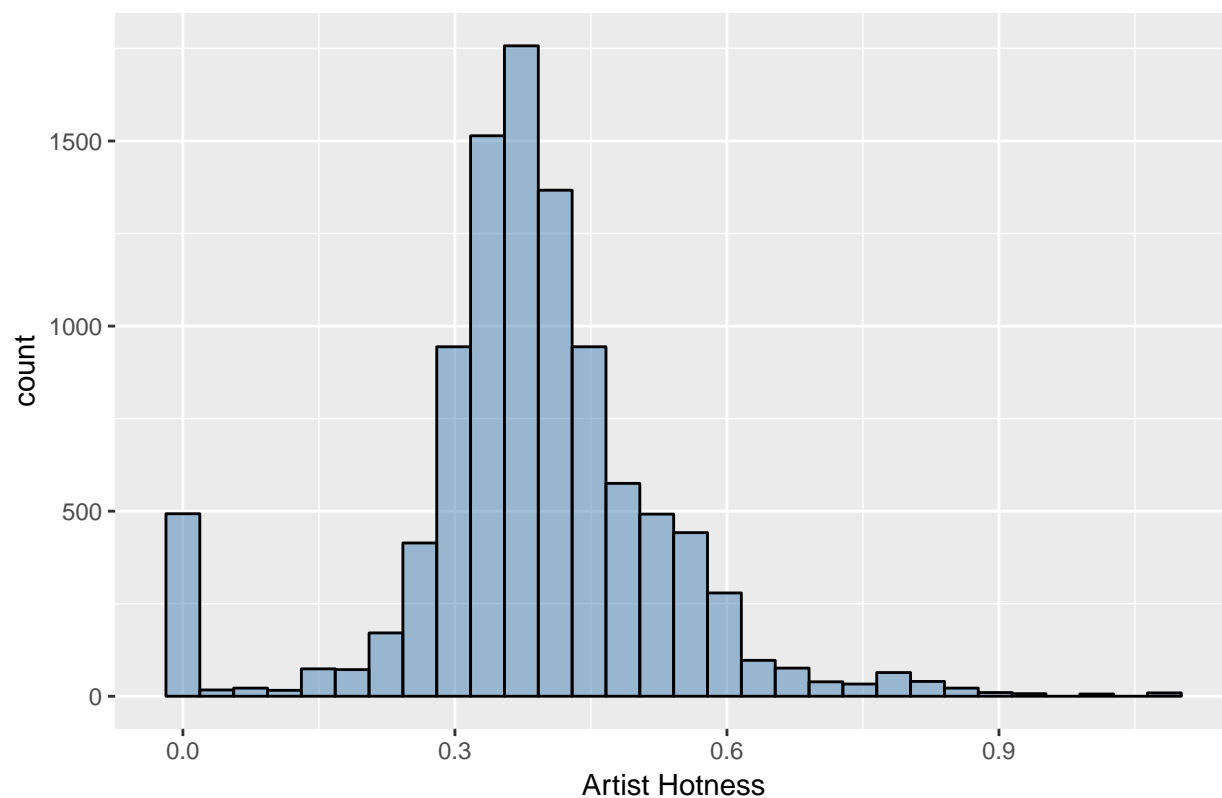
```

#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/Gi
#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.hotttnesss)) + 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 Quantile.25% Quantile.50% Quantile.75%
##    0.3857065    0.3807564    0.0000000    1.0825026    0.1434688    0.3255062    0.3807564    0.4539062

##Methodology for assigning artist hotness levels - uses quantiles from descriptive_statistics function
#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$artist.hotness.label <- ifelse(music$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",
                                                           ifelse(music$artist.hotttnesss < 0.3252656, "Frigid", "Cold")),
                                     unique(music$artist.hotness.label)

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

#End of new code from Courtney
#Prior to importing, a new column artist.hotttnesss.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$artist.hotness.label <- as.factor(music$artist.hotness.label)
str(music)

## 'data.frame': 5648 obs. of 23 variables:
## $ artist.hotttnesss : 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.hotttnesss.label : Factor w/ 3 levels "Cold","Hot","Warm": 3 3 1 1 3 3 1 3 1 2 ...
## $ artist.hotness.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
```

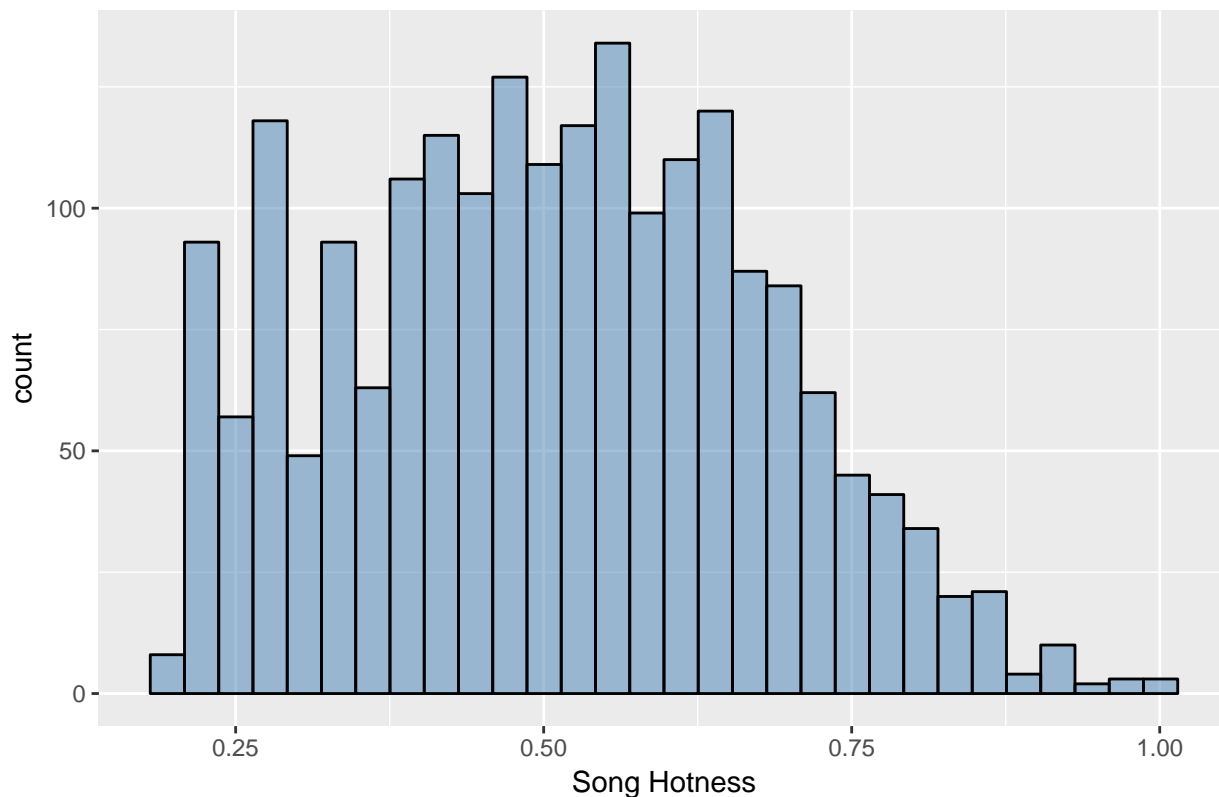
```
#cmbomusic2 <- cmbomusic[-c(5,6)]
cmbomusic3 <- na.omit(cmbomusic)
cmbomusic3$song.hottnesss.label <- ifelse( cmbomusic3$song.hottnesss >=0.6011861, "Hot",ifelse(cmbomusic3$song.hottnesss < 0.6011861, "Tepid",ifelse(cmbomusic3$song.hottnesss < 0.4, "Cool", ifelse(cmbomusic3$song.hottnesss < 0.3, "Warm", "Frigid"))))
unique(cmbomusic3$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.hottnesss)
```

```
##          Mean          Median          Min          Max          SD Quantile.25% Quantile.50% Quantile.75%
##    0.5073226    0.5096410    0.1938578    1.0000000    0.1686679    0.3827233    0.5096410    0.6301000
```

```
cmbomusic3$song.hottness.label <- ifelse( cmbomusic3$song.hotttnesss >=0.64787976, "Hot",ifelse(cmbomusic3$song.hotttnesss < 0.64787976, "Cold", "Warm"))
unique(cmbomusic3$song.hottnesss.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 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 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.hotttnesss.label : Factor w/ 5 levels "Cool","Frigid",...: 3 4 1 3 3 3 1 3 3 ...
## $ song.hottness.label : chr "Hot" "Cold" "Cold" "Hot" ...
```

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

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

```
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 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 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.hotttnesss.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$song.hotttnesss.label <- ifelse( cmbomusic3$song.hotttnesss >=0.6011861, "Hot",ifelse(cmbomusic3$song.hotttnesss < 0.6011861, "Cold", "Hot"))
unique(cmbomusic3$song.hotttnesss.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 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.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 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.hotttnesss.label : chr "Hot" "Tepid" "Cool" "Hot" ...
## $ song.hotness.label : chr "Hot" "Cold" "Cold" "Hot" ...
```

```
cmbomusic3$song.hotttnesss.label <- as.factor(cmbomusic3$song.hotttnesss.label)
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.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 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.hotttnesss.label : Factor w/ 5 levels "Cool","Frigid",...: 3 4 1 3 3 3 3 1 3 3 ...
## $ song.hottness.label : chr "Hot" "Cold" "Cold" "Hot" ...

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

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

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"
```

Features

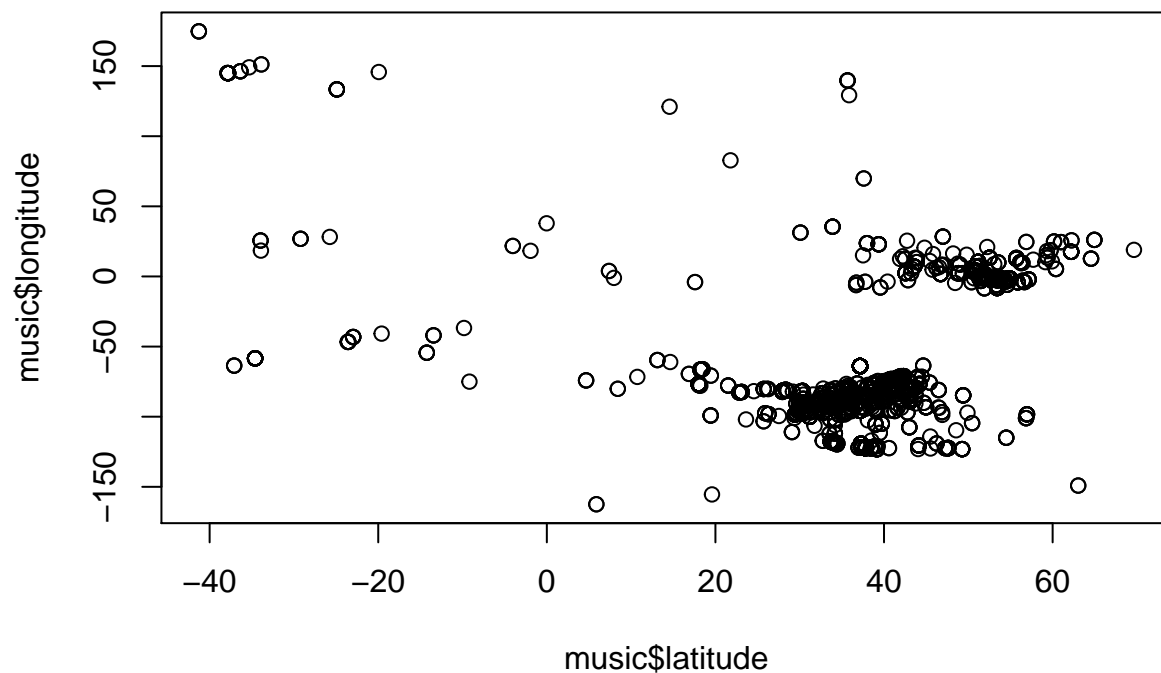
```
#View the number of Cold/Warm/Hot labels
table(music$artist.hotttnesss.label)

##
## Cold Hot Warm
## 1180 1579 2889

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

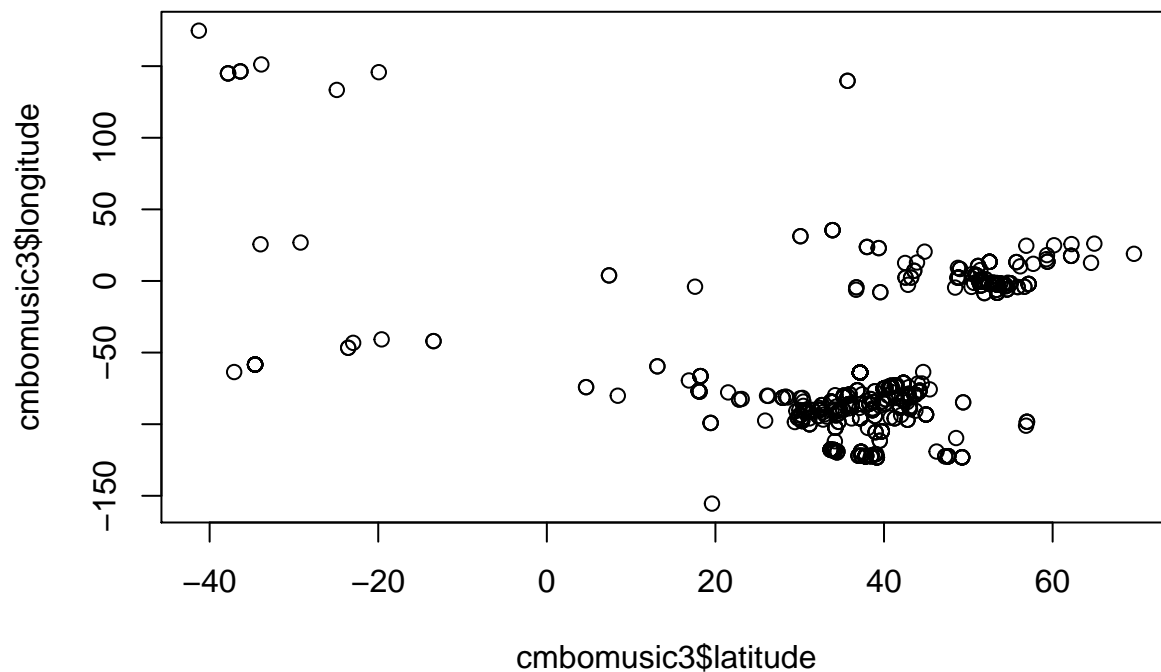
```
#View the number of Cold/Warm/Hot labels
table(cmbomusic3$song.hotness.label)
```

```
##
## 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 mapWorld <- borders("world", colour="gray50", fill="white")

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

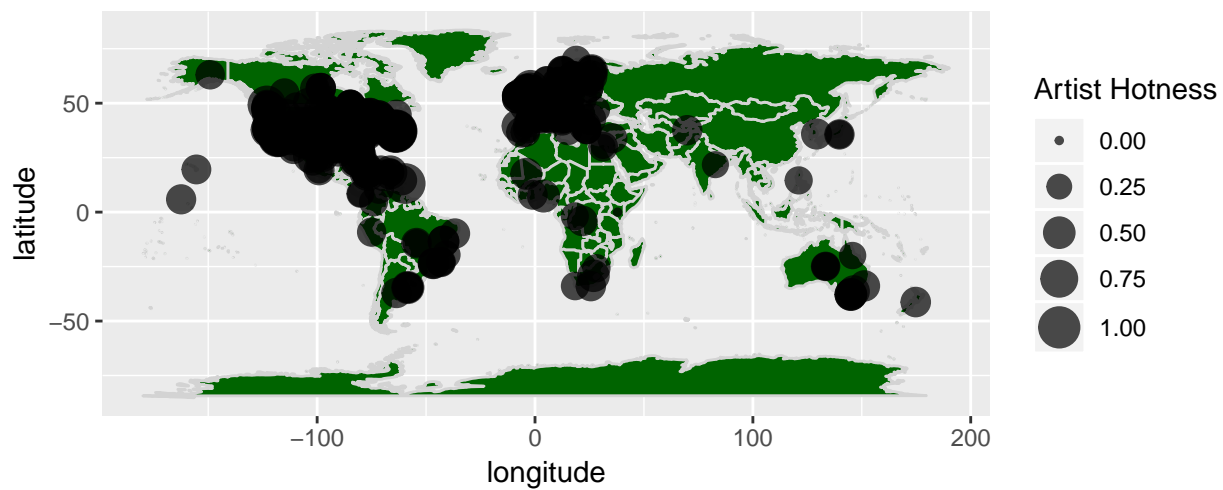
##
## Attaching package: 'dplyr'

## 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.hotttnesss)
loc<-unique(loc)
colnames(loc)<-c("longitude", "latitude","artist hottness")
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.hottness))+#, f
  borders(fill="dark green",colour="light grey")+
  geom_point(color="black",alpha=I(7/10))+
  scale_size(range=c(1,7), guide = "legend",labs(size="Artist Hottness"))+
  coord_equal()+ ggtitle("Artist Location by Hottness")
ahworld
```

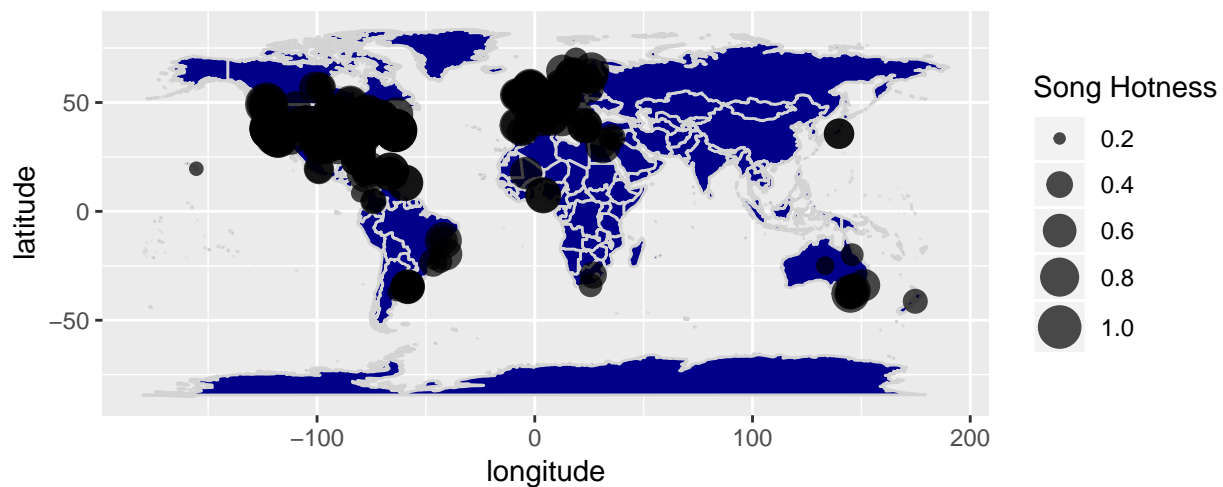
Artist Location by Hotness



```
#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.hottnesss)
songlc<-unique(songlc)
colnames(songlc)<-c("longitude", "latitude","song hottness")
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.hottnesss))
borders(fill="dark blue",colour="light grey")+
  geom_point(color="black",alpha=I(7/10))+
  scale_size(range=c(1,7), guide = "legend",labs(size="Song Hottness"))+
  coord_equal()+ ggtitle("Song Location by Hottness")

songlc_dfwrld
```

Song Location by Hotness



Methods - Linear Regression

```
library("PerformanceAnalytics")

## Loading required package: xts
## Loading required package: zoo
##
## Attaching package: 'zoo'
##
## The following objects are masked from 'package:base':
##
##   as.Date, as.Date.numeric
##
## Registered S3 method overwritten by 'xts':
##   method      from
##   as.zoo.xts zoo
##
## Attaching package: 'xts'
##
## The following objects are masked from 'package:dplyr':
##
##   first, last
##
## Attaching package: 'PerformanceAnalytics'
##
## The following objects are masked from 'package:moments':
##
##   kurtosis, skewness
##
## The following object is masked from 'package:graphics':
##
##   legend
#code from Juan
#Artist prediction
lm1 <- lm(formula = music$artist.hotttnesss ~ music$year + music$bars_confidence + music$tempo + music$duration + music$statums_start + music$statums_end + music$statums_confidence + music$statums_familiarity + music$statums_latitude + music$statums_longitude + music$statums_beats_start + music$statums_beats_confidence + music$statums_end_of_fade_in)
#removed music$bars_start which was causing an error

#Songs with labels
lm2 <- lm(cmbomusic3$song.hotttnesss ~ cmbomusic3$year + cmbomusic3$loudness + cmbomusic3$statums_confidence + cmbomusic3$statums_familiarity + cmbomusic3$statums_latitude + cmbomusic3$statums_longitude + cmbomusic3$statums_beats_start + cmbomusic3$statums_beats_confidence + cmbomusic3$statums_end_of_fade_in)

#Songs no labels
lm3 <- lm(cmbomusic3$song.hotttnesss ~ cmbomusic3$year + cmbomusic3$loudness + cmbomusic3$statums_confidence + cmbomusic3$statums_familiarity + cmbomusic3$statums_latitude + cmbomusic3$statums_longitude + cmbomusic3$statums_beats_start + cmbomusic3$statums_beats_confidence + cmbomusic3$statums_end_of_fade_in)

summary(lm1)

##
## Call:
## lm(formula = music$artist.hotttnesss ~ music$year + music$bars_confidence +
##   music$tempo + music$duration + music$statums_start + music$statums_end + music$statums_confidence +
##   music$statums_familiarity + music$statums_latitude +
##   music$tempo + music$longitude + music$beats_start + music$beats_confidence +
##   music$end_of_fade_in)
```

```
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.41865 -0.03239 -0.00136  0.03219  0.50014
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      1.500e-02  8.045e-03   1.865  0.0622 .
## music$year        6.911e-06  1.081e-06   6.392 1.77e-10 ***
## music$bars_confidence -4.242e-04  3.754e-03  -0.113  0.9100
## music$tempo       -3.122e-05  3.082e-05  -1.013  0.3111
## music$duration     1.842e-05  1.881e-05   0.979  0.3276
## music$start_of_fade_out -2.842e-05  2.121e-05  -1.340  0.1803
## music$statums_start -5.004e-03  7.041e-03  -0.711  0.4773
## music$familiarity    6.625e-01  7.156e-03  92.582 < 2e-16 ***
## music$latitude     -1.039e-04  1.006e-04  -1.033  0.3015
## music$longitude     -5.606e-05  3.190e-05  -1.758  0.0789 .
## music$beats_start    5.494e-03  6.748e-03   0.814  0.4155
## music$beats_confidence -2.277e-03  3.227e-03  -0.706  0.4804
## music$end_of_fade_in  9.355e-05  6.367e-04   0.147  0.8832
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.07602 on 5635 degrees of freedom
## Multiple R-squared:  0.6443, Adjusted R-squared:  0.6436
## F-statistic: 850.6 on 12 and 5635 DF, p-value: < 2.2e-16
```

```
summary(lm2)
```

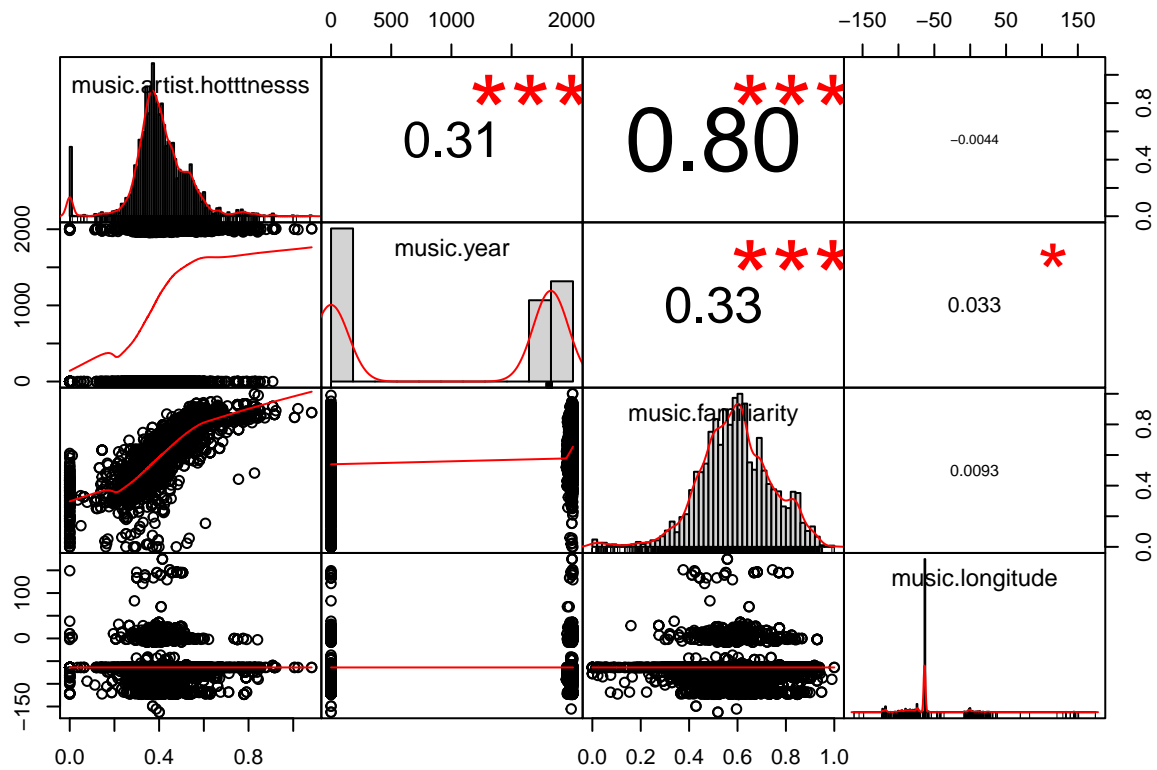
```
##
## Call:
## lm(formula = cmbomusic3$song.hottnesss ~ cmbomusic3$year + cmbomusic3$loudness +
##      cmbomusic3$statums_confidence + cmbomusic3$statums_start +
##      cmbomusic3$tempo + cmbomusic3$terms_freq + cmbomusic3$time_signature_confidence +
##      cmbomusic3$year + factor(cmbomusic3$song.hottnesss.label) +
##      factor(cmbomusic3$song.hottness.label))
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.091973 -0.025431 -0.000346  0.019615  0.262749
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)      7.378e-01  2.254e-01   3.273  0.00108 **
## cmbomusic3$year    -1.955e-04  1.119e-04  -1.747  0.08071 .
## cmbomusic3$loudness  4.688e-04  2.289e-04   2.048  0.04070 *
## cmbomusic3$statums_confidence  5.998e-04  3.520e-03   0.170  0.86473
## cmbomusic3$statums_start -3.365e-04  1.940e-03  -0.174  0.86226
## cmbomusic3$tempo     3.435e-05  3.042e-05   1.129  0.25891
## cmbomusic3$terms_freq  2.783e-03  2.177e-02   0.128  0.89827
## cmbomusic3$time_signature_confidence  4.067e-03  3.476e-03   1.170  0.24217
## factor(cmbomusic3$song.hottnesss.label)Frigid -9.418e-02  4.209e-03 -22.373 < 2e-16 ***
## factor(cmbomusic3$song.hottnesss.label)Hot    2.351e-01  7.613e-03  30.877 < 2e-16 ***
## factor(cmbomusic3$song.hottnesss.label)Tepid  5.401e-02  4.676e-03  11.550 < 2e-16 ***
## factor(cmbomusic3$song.hottnesss.label)Warm    1.374e-01  7.112e-03  19.321 < 2e-16 ***
```

```
## factor(cmbomusic3$song.hotness.label)Hot      1.504e-01  7.127e-03  21.103 < 2e-16 ***
## factor(cmbomusic3$song.hotness.label)Warm      3.659e-02  5.961e-03   6.139 9.96e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.04478 on 2023 degrees of freedom
## Multiple R-squared:  0.93, Adjusted R-squared:  0.9295
## F-statistic: 2067 on 13 and 2023 DF, p-value: < 2.2e-16
summary(lm3)
```

```
##
## Call:
## lm(formula = cmbomusic3$song.hotttnesss ~ cmbomusic3$year + cmbomusic3$loudness +
##      cmbomusic3$statums_confidence + cmbomusic3$statums_start +
##      cmbomusic3$tempo + cmbomusic3$terms_freq + cmbomusic3$time_signature_confidence +
##      cmbomusic3$year)
##
## Residuals:
##      Min        1Q    Median        3Q        Max
## -0.34868 -0.12344 -0.00192  0.11879  0.50615
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)   -0.0030165   0.8243585  -0.004   0.997
## cmbomusic3$year    0.0002593   0.0004092   0.634   0.526
## cmbomusic3$loudness  0.0082323   0.0008177  10.068 <2e-16 ***
## cmbomusic3$statums_confidence -0.0057153   0.0128559  -0.445   0.657
## cmbomusic3$statums_start -0.0010096   0.0070970  -0.142   0.887
## cmbomusic3$tempo    0.0001436   0.0001112   1.291   0.197
## cmbomusic3$terms_freq  0.0515568   0.0794760   0.649   0.517
## cmbomusic3$time_signature_confidence 0.0021321   0.0127176   0.168   0.867
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.1639 on 2029 degrees of freedom
## Multiple R-squared:  0.05922, Adjusted R-squared:  0.05597
## F-statistic: 18.24 on 7 and 2029 DF, p-value: < 2.2e-16
```

```
#Artist hotness correlation
cor1 <- data.frame(music$artist.hotttnesss, music$year, music$familiarity, music$longitude)
cor(cor1)
```

```
##              music.artist.hotttnesss music.year music.familiarity music.longitude
## music.artist.hotttnesss      1.000000000 0.31443416      0.800773449  -0.004446176
## music.year                   0.314434159 1.000000000      0.333867762   0.032639949
## music.familiarity            0.800773449 0.33386776      1.000000000   0.009304099
## music.longitude             -0.004446176 0.03263995      0.009304099   1.000000000
chart.Correlation(cor1, histogram=TRUE, pch=10, cex.labels=2.9)
```

#Song hotness correlation

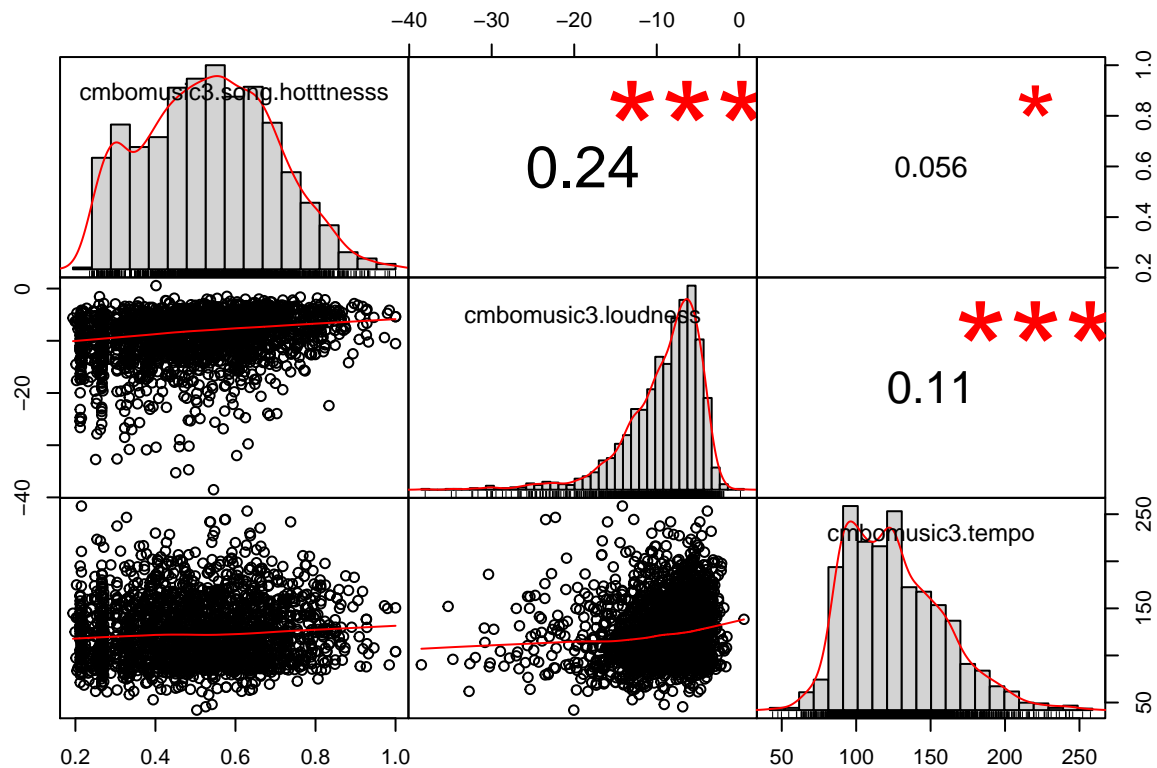
```
cor2 <- data.frame(cmbomusic3$song.hottnesss, cmbomusic3$loudness, cmbomusic3$tempo)
cor(cor2)
```

```
##                                cmbomusic3.song.hottnesss  cmbomusic3.loudness  cmbomusic3.temp
## cmbomusic3.song.hottnesss      1.00000000      0.2406421      0.05570821
## cmbomusic3.loudness             0.24064215      1.0000000      0.11259809
## cmbomusic3.temp                0.05570821      0.1125981      1.00000000
```

#install.packages("PerformanceAnalytics")

```
library(PerformanceAnalytics)
```

```
chart.Correlation(cor2, histogram=TRUE, pch=10, cex.labels=2.9)
```



```
#sum
library(kernlab)
```

```
##
## Attaching package: 'kernlab'
## The following object is masked from 'package:psych':
##
##   alpha
## The following object is masked from 'package:ggplot2':
##
##   alpha
```

```
library(e1071)
```

```
##
## Attaching package: 'e1071'
## The following objects are masked from 'package:PerformanceAnalytics':
##
##   kurtosis, skewness
## The following objects are masked from 'package:moments':
##
##   kurtosis, moment, skewness
```

```
rndm <- sample(1:dim(music)[1])
summary(rndm)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##         1    1413    2824    2824    4236    5648
```

```
length(rndm)

## [1] 5648
cut <- floor(2 * dim(music)[1]/3)
cut

## [1] 3765
train <- music[rndm[1:cut],]
test <- music[rndm[(cut + 1):dim(music)[1]],]
svm0 <- ksvm(artist.hotttnesss.label ~ ., data=train, kernel= "rbfdot", kpar="automatic", C=5, cross=3,
svm0

## Support Vector Machine object of class "ksvm"
##
## SV type: C-svc (classification)
## parameter : cost C = 5
##
## Gaussian Radial Basis kernel function.
## Hyperparameter : sigma = 0.0467503138947049
##
## Number of Support Vectors : 888
##
## Objective Function Value : -47.7061 -1158.314 -510.3978
## Training error : 0.028951
## Cross validation error : 0.05923
## Probability model included.

svmP <- predict(svm0, test, type= "votes")
comp <- data.frame(test[,22], svmP[1,])
table(comp)

##          svmP.1...
## test...22.    0    1    2
##      Cold    0   61  318
##      Hot   506    0    0
##      Warm  295  699    4
```

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:psych':
##
##      outlier

## The following object is masked from 'package:ggplot2':
##
##      margin

#John commented out rf because it is running against the same file (music) as rf2
#set.seed(222)
#rf <- randomForest(music[,c(-1,-21,-22)],music[,21])
#print(rf)
#attributes(rf)
#rf$confusion
#Run randomForest on 5 levels
#John added -1 to remove the hotness variable from the rf
library(randomForest)
set.seed(222)
rf2 <- randomForest(music[,c(-1,-22,-23)],music[,22])
print(rf2)

##
## Call:
## randomForest(x = music[, c(-1, -22, -23)], 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: 20.18%
## Confusion matrix:
##      Cold  Hot  Warm class.error
## Cold  728   4  448   0.3830508
## Hot    6 1289  284   0.1836605
## Warm  200  198 2491   0.1377639

attributes(rf2)

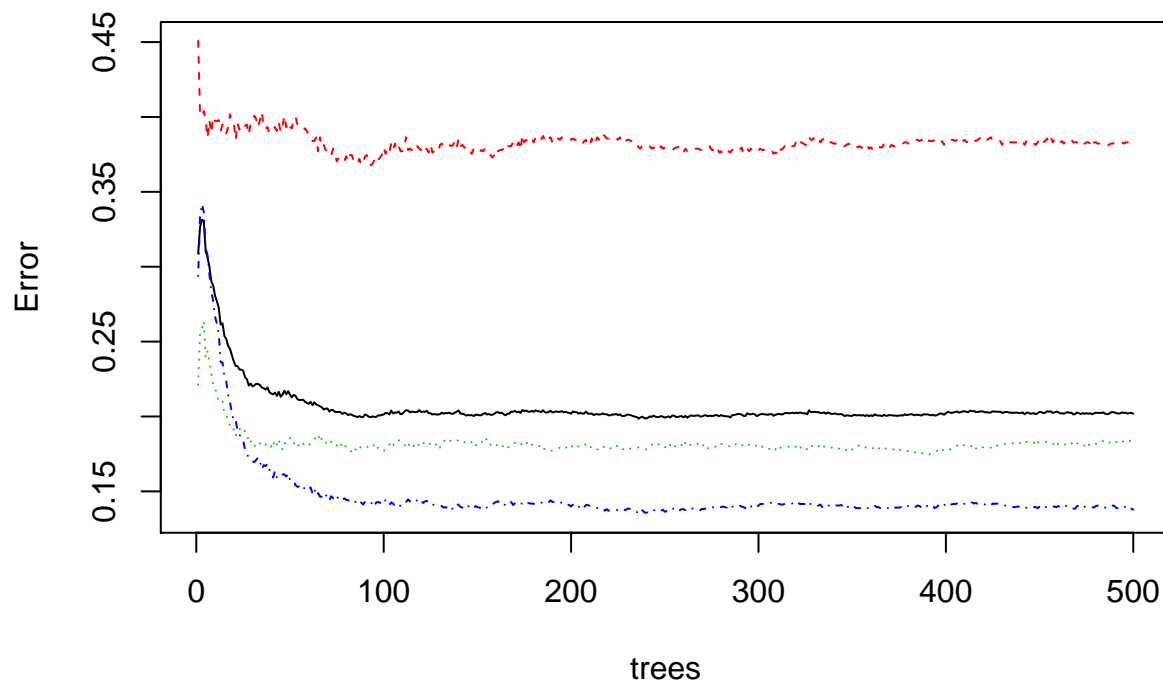
## $names
## [1] "call"          "type"          "predicted"     "err.rate"     "confusion"    "votes"
##
## $class
## [1] "randomForest"

rf2$confusion

##      Cold  Hot  Warm class.error
## Cold  728   4  448   0.3830508
## Hot    6 1289  284   0.1836605
## Warm  200  198 2491   0.1377639
```

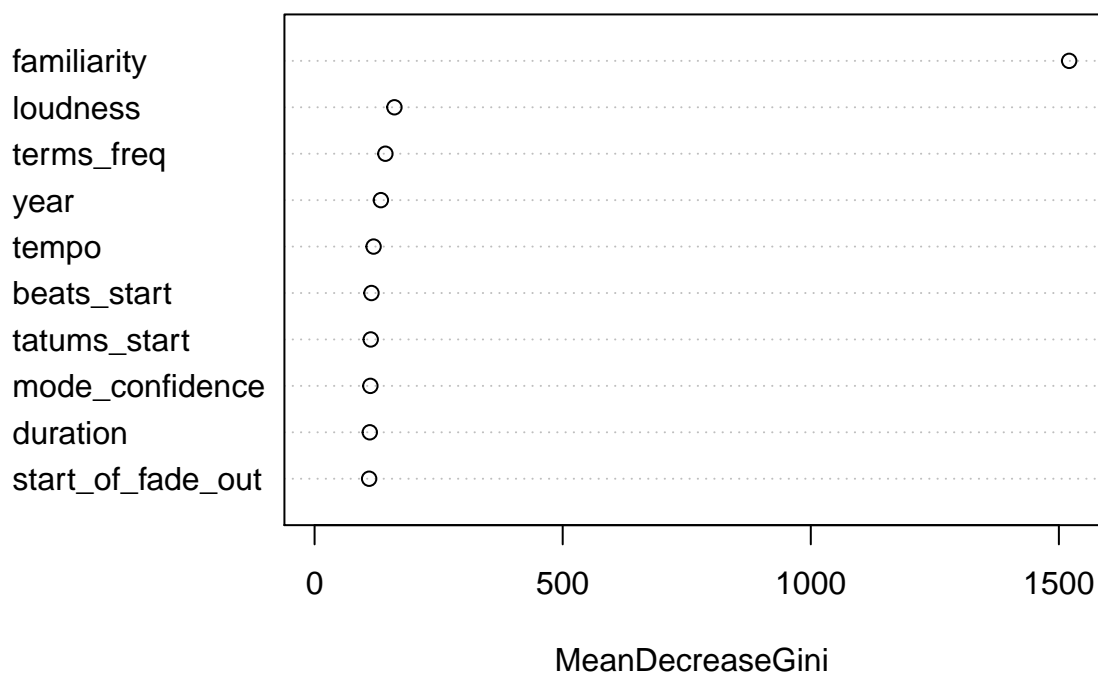
```
#Error rate of Random Forest
#plot(rf)
plot(rf2)
```

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

Top 10 – Variable Importance



```
importance(rf2)
```

```
##               MeanDecreaseGini
## bars_confidence      107.58378
## beats_confidence      94.47263
## beats_start          114.60290
## duration             111.20674
## end_of_fade_in        88.18414
## familiarity          1520.73102
## key                   70.97697
## key_confidence       107.84287
## latitude              80.31500
## longitude             80.00661
## loudness              160.85558
## mode_confidence      112.40895
## start_of_fade_out    109.94686
## tatums_confidence    100.29904
## tatums_start         113.12138
## tempo                 118.86929
## terms_freq           142.67373
## time_signature        31.82993
## time_signature_confidence 81.87304
## year                 133.56141
```

```
varUsed(rf2)
```

```
## [1] 22981 20248 24173 23234 18362 42545 16709 23076 13678 13543 26260 23733 23098 21580 23800 24889
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 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$song.hottness.label <- as.factor(cmbomusic4$song.hottness.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: 51.74%
## Confusion matrix:
## Cold Hot Warm class.error
## Cold 297 29 381 0.5799151
## Hot 76 118 246 0.7318182
## Warm 244 78 568 0.3617978

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: 51.74%
## Confusion matrix:
## Cold Hot Warm class.error
## Cold 297 29 381 0.5799151
## Hot 76 118 246 0.7318182
## Warm 244 78 568 0.3617978

attributes(rf3)

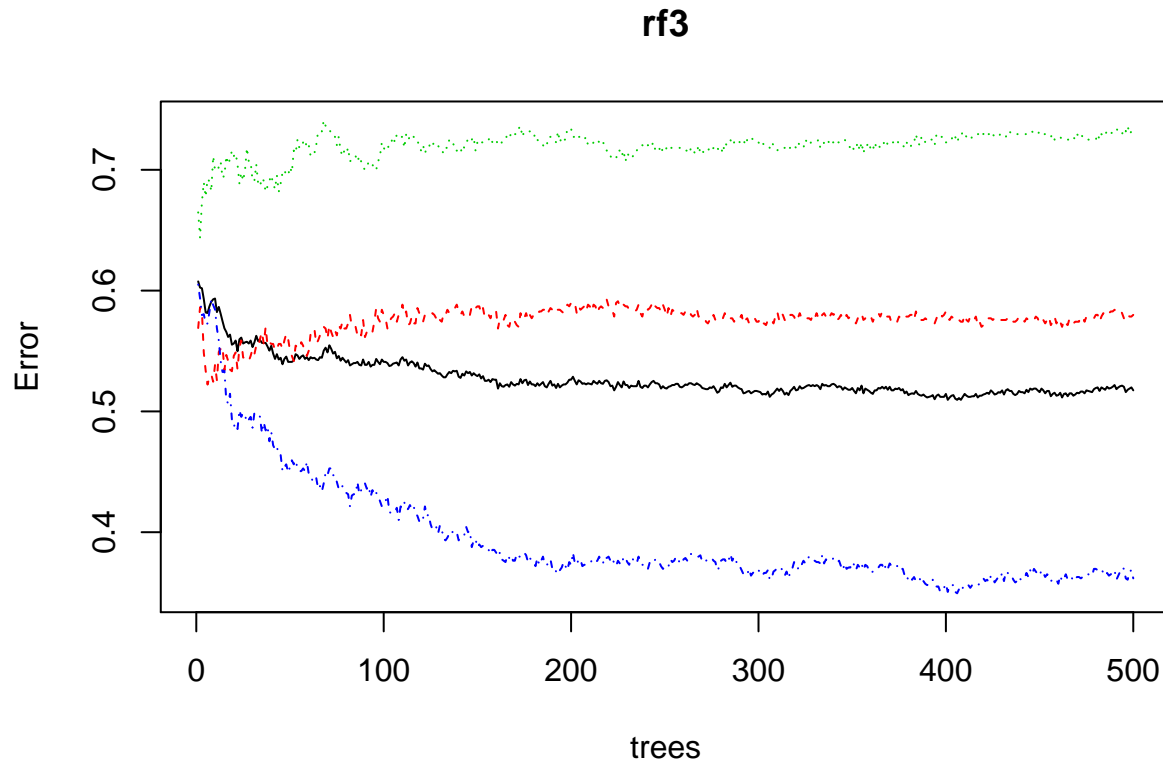
## $names
## [1] "call" "type" "predicted" "err.rate" "confusion" "votes"
##
## $class
## [1] "randomForest"
```



```
rf3$confusion
```

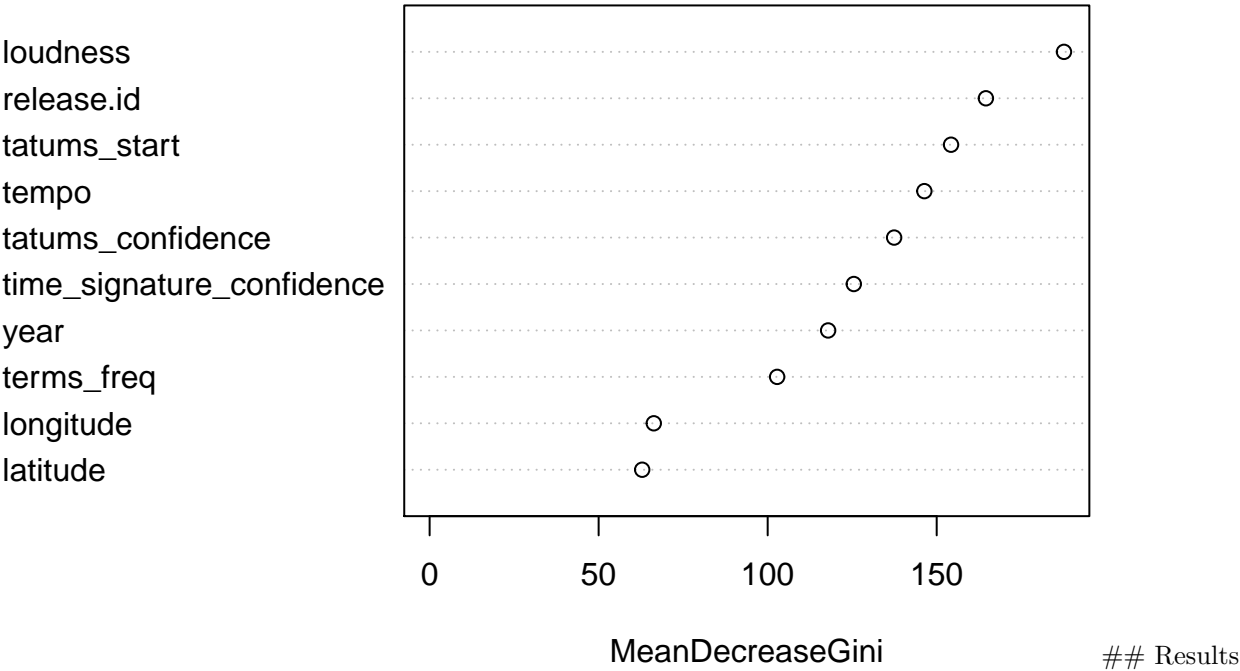
```
##      Cold Hot Warm class.error
## Cold  297  29  381  0.5799151
## Hot   76 118  246  0.7318182
## Warm  244  78  568  0.3617978
```

```
plot(rf3)
```



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

Top 10 – Variable Importance



Conclusion

Appendices