

ANUJATESTASS2

Anuja Saira Abraham

2023-06-02

```
library(boot)
library(tidyverse)
```

```
## -- Attaching packages ----- tidyverse 1.3.2 --
## v ggplot2 3.4.0      v purrr  1.0.1
## v tibble  3.2.1      v dplyr  1.1.2
## v tidyr   1.3.0      v stringr 1.5.0
## v readr   2.1.3      v forcats 0.5.2
## -- Conflicts ----- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(reshape2)
```

```
##
## Attaching package: 'reshape2'
##
## The following object is masked from 'package:tidyr':
##
##      smiths
```

```
library(combinat)
```

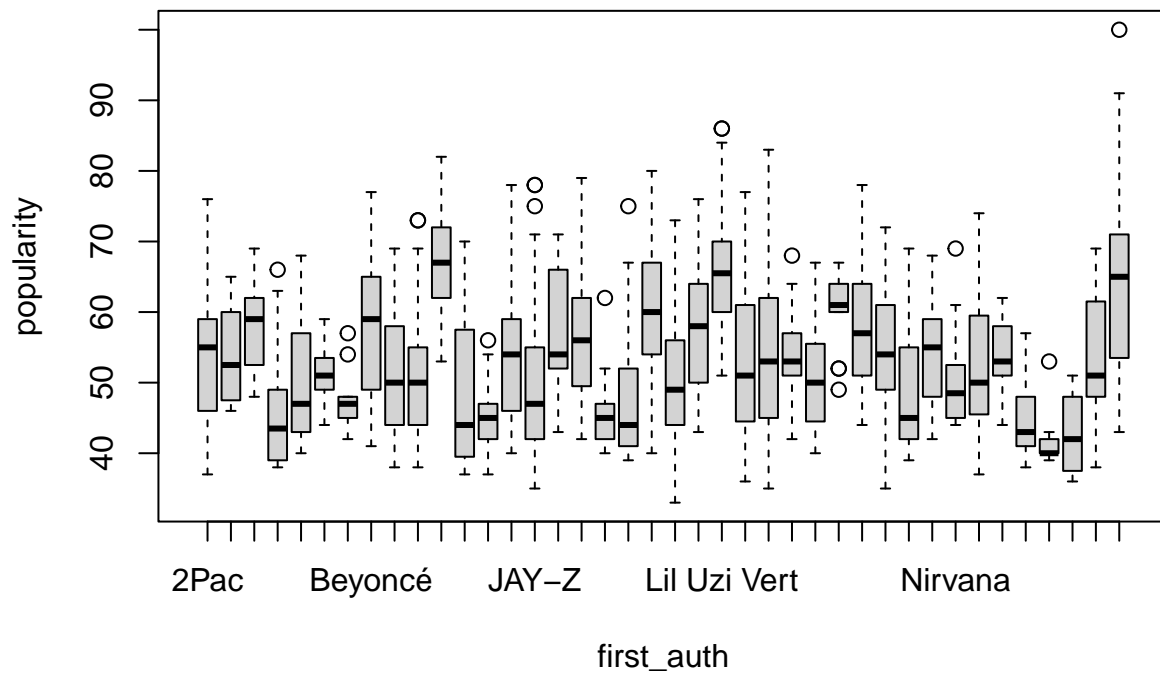
```
##
## Attaching package: 'combinat'
##
## The following object is masked from 'package:utils':
##
##      combn
```

```
songs <- readRDS("spot (1).RDS")
head(songs)
```

```
## # A tibble: 6 x 18
##   acousticness danceability duration_ms energy explicit id      instrumentalness
##   <dbl>         <dbl>         <int> <dbl>   <int> <chr>          <dbl>
## 1     0.116         0.908       203520 0.573     1 3iNzFu~      0.000547
## 2     0.299         0.703       247600 0.639     1 3eBBbA~       0
## 3     0.283         0.874       230227 0.64      1 1SgENh~      0.000002
```

```
## 4      0.0672      0.748      230493  0.614      1 36UjTT~      0.0000781
## 5      0.0634      0.536      267813  0.836      1 0lqKrm~      0
## 6      0.114      0.824      286893  0.847      1 06jYyA~      0.00000284
## # i 11 more variables: liveness <dbl>, loudness <dbl>, name <chr>,
## #   popularity <int>, release_date <chr>, speechiness <dbl>, tempo <dbl>,
## #   valence <dbl>, first_auth <chr>, n <int>, pop <dbl>
```

```
boxplot(popularity~first_auth,data=songs)
```



```
u_art<- unique(songs$first_auth)

#punto B

X <- songs$pop[songs$first_auth == "AC/DC"]

# Extract popularity measurements for Artist Y
Y <- songs$pop[songs$first_auth == "Kanye West"]

med_logit_X <-median( logit(X))
med_logit_Y <-median( logit(Y))

T_XY <- abs(med_logit_X-med_logit_Y)

pop_diff_T <- function(){
  res<-sapply(1:40, function(x) {
```

```

sapply(1:40, function(y) {
  abs(median(logit(songs$pop[songs$first_auth == u_art[x]]))-median(logit(songs$pop[songs$first_auth == u_art[y]]))
})
})
return(res)
}

```

```
pop_diff<- pop_diff_T()
```

```

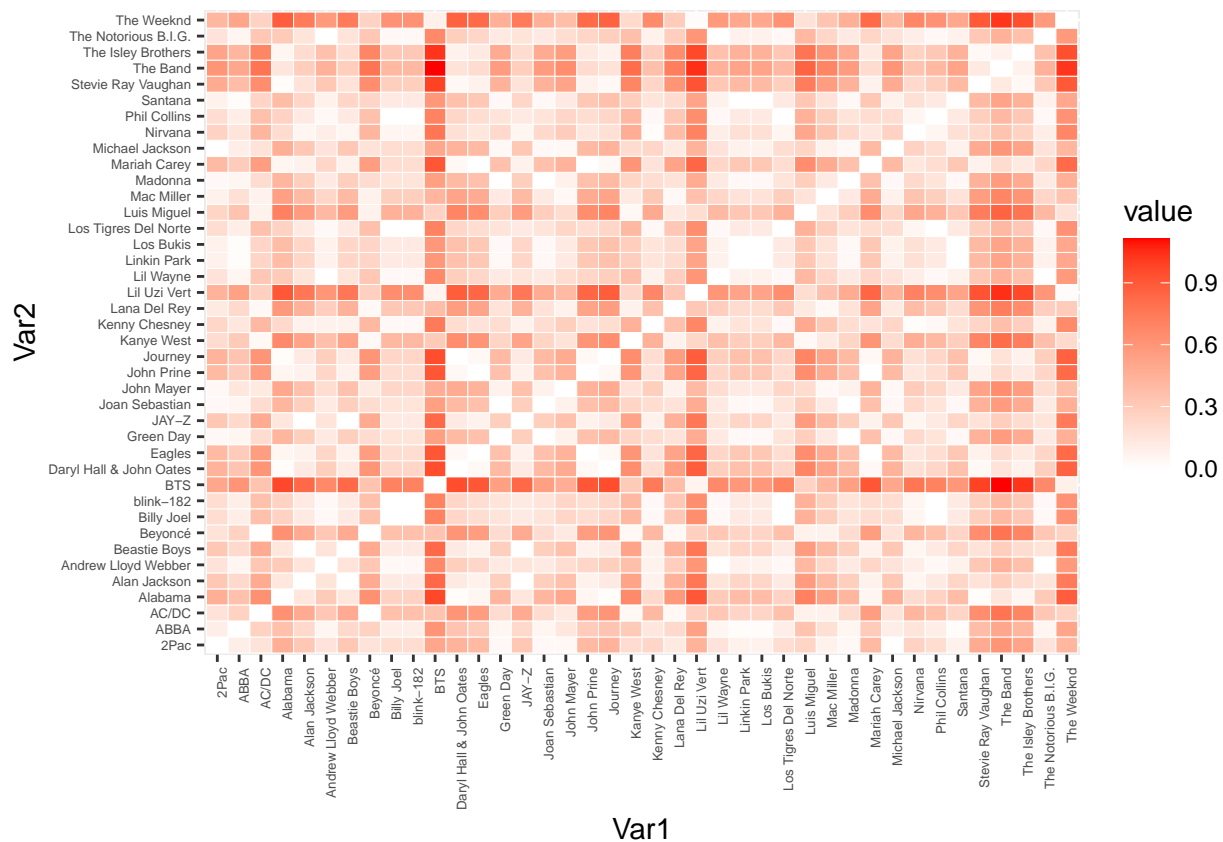
rownames(pop_diff) <- u_art
colnames(pop_diff) <- u_art

```

```

ggplot(melt(pop_diff),aes(Var1,Var2))+
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient(low = "white", high = "red")+
  theme(axis.text.x = element_text(angle=90,hjust = 1,siz=5),
        axis.text.y = element_text(hjust = 1,siz=5))

```



```

#punto c
data<-songs
p_values <- matrix(0,nrow = length(u_art),ncol = length(u_art))
for (i in 1:length(u_art)){
  for(j in 1:length(u_art)){
    if(i>=j){

```

```

X <- data[data$first_auth == u_art[i], "pop"]
Y <- data[data$first_auth == u_art[j], "pop"]
observed_stat=abs(median(logit(songs$pop[songs$first_auth == u_art[i]]))-median(logit(songs$pop[s

num_perm<- 1000
permutation_stat<- numeric(num_perm)
for (k in 1:num_perm){
  combined<- union_all(X,Y)
  combined$pop <- sample(combined$pop,nrow(combined),replace = F)
  perm_X<- combined$pop[1:length(X$pop)]
  perm_Y<-combined$pop[(length(X$pop)+1):(length(X$pop)+length(Y$pop))]
  permutation_stat[k]<- abs(median(logit(perm_X))-median(logit(perm_Y)))
}

p_value <- mean(permutation_stat>=observed_stat)

p_values[i,j]<- p_value
p_values[j,i]<- p_value

}
}
}

```

p_values

```

##      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12]
## [1,] 1.000 0.745 0.421 0.000 0.060 0.516 0.159 0.046 0.484 0.050 0.000 0.014
## [2,] 0.745 1.000 0.349 0.000 0.138 0.368 0.145 0.097 0.613 0.254 0.000 0.189
## [3,] 0.421 0.349 1.000 0.000 0.011 0.042 0.004 1.000 0.160 0.025 0.007 0.051
## [4,] 0.000 0.000 0.000 1.000 0.094 0.003 0.293 0.000 0.116 0.010 0.000 0.996
## [5,] 0.060 0.138 0.011 0.094 1.000 0.193 1.000 0.000 0.510 0.078 0.000 0.298
## [6,] 0.516 0.368 0.042 0.003 0.193 1.000 0.039 0.031 1.000 0.726 0.000 0.061
## [7,] 0.159 0.145 0.004 0.293 1.000 0.039 1.000 0.002 0.373 0.202 0.000 0.571
## [8,] 0.046 0.097 1.000 0.000 0.000 0.031 0.002 1.000 0.046 0.002 0.000 0.000
## [9,] 0.484 0.613 0.160 0.116 0.510 1.000 0.373 0.046 1.000 1.000 0.000 0.471
## [10,] 0.050 0.254 0.025 0.010 0.078 0.726 0.202 0.002 1.000 1.000 0.000 0.080
## [11,] 0.000 0.000 0.007 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000
## [12,] 0.014 0.189 0.051 0.996 0.298 0.061 0.571 0.000 0.471 0.080 0.000 1.000
## [13,] 0.001 0.002 0.001 0.324 0.234 0.030 0.329 0.000 0.073 0.022 0.000 1.000
## [14,] 0.834 0.616 0.137 0.000 0.040 0.393 0.096 0.012 0.283 0.119 0.000 0.014
## [15,] 0.001 0.093 0.004 0.138 1.000 0.226 1.000 0.000 0.464 0.087 0.000 0.320
## [16,] 0.858 0.831 0.411 0.000 0.045 0.209 0.014 0.179 0.320 0.031 0.000 0.026
## [17,] 0.744 0.376 0.372 0.000 0.000 0.121 0.017 0.079 0.112 0.001 0.000 0.000
## [18,] 0.014 0.005 0.002 0.654 0.229 0.009 0.321 0.000 0.185 0.071 0.000 0.731
## [19,] 0.000 0.001 0.005 0.691 0.110 0.005 0.214 0.000 0.212 0.022 0.000 1.000
## [20,] 0.004 0.018 0.864 0.000 0.000 0.010 0.000 0.788 0.005 0.000 0.000 0.000
## [21,] 0.012 0.148 0.019 0.015 0.288 0.476 0.493 0.000 0.681 0.463 0.000 0.102
## [22,] 0.297 0.243 0.879 0.000 0.001 0.096 0.005 0.893 0.065 0.004 0.000 0.000

```

```

## [23,] 0.000 0.000 0.035 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.145 0.000
## [24,] 0.221 0.717 0.159 0.055 0.200 1.000 0.403 0.012 0.869 0.815 0.000 0.000 0.192
## [25,] 0.508 0.956 0.240 0.016 0.053 0.669 0.256 0.023 0.528 0.176 0.000 0.000 0.067
## [26,] 0.586 0.836 0.089 0.002 0.133 0.718 0.032 0.045 0.164 0.087 0.000 0.000 0.050
## [27,] 0.105 0.425 0.016 0.004 0.292 0.800 0.402 0.001 1.000 1.000 0.000 0.000 0.107
## [28,] 0.126 0.065 0.319 0.002 0.001 0.000 0.001 0.517 0.019 0.003 0.033 0.033 0.011
## [29,] 0.335 0.170 0.548 0.000 0.000 0.091 0.003 0.424 0.032 0.000 0.000 0.000 0.000
## [30,] 0.863 0.790 0.272 0.000 0.074 0.286 0.050 0.116 0.270 0.062 0.000 0.000 0.037
## [31,] 0.015 0.043 0.013 0.531 0.162 0.058 0.492 0.000 0.229 0.031 0.000 0.000 0.813
## [32,] 1.000 0.429 0.247 0.000 0.030 0.286 0.030 0.104 0.189 0.035 0.000 0.000 0.002
## [33,] 0.126 0.120 0.005 0.001 0.621 0.156 0.461 0.001 0.668 0.350 0.000 0.000 0.063
## [34,] 0.618 0.803 0.232 0.071 0.421 1.000 0.166 0.092 1.000 1.000 0.000 0.000 0.388
## [35,] 0.674 0.936 0.311 0.001 0.227 0.271 0.094 0.124 0.303 0.173 0.000 0.000 0.121
## [36,] 0.000 0.000 0.001 1.000 0.105 0.002 0.095 0.000 0.074 0.010 0.000 0.000 0.759
## [37,] 0.000 0.002 0.001 0.037 0.034 0.002 0.003 0.000 0.059 0.000 0.000 0.000 0.043
## [38,] 0.000 0.002 0.003 0.620 0.078 0.000 0.035 0.000 0.074 0.013 0.000 0.000 0.537
## [39,] 0.499 0.774 0.141 0.001 0.099 1.000 0.152 0.016 1.000 0.686 0.000 0.000 0.093
## [40,] 0.001 0.001 0.266 0.000 0.000 0.002 0.001 0.004 0.001 0.000 0.000 0.191 0.000
##      [,13] [,14] [,15] [,16] [,17] [,18] [,19] [,20] [,21] [,22] [,23] [,24]
## [1,] 0.001 0.834 0.001 0.858 0.744 0.014 0.000 0.004 0.012 0.297 0.000 0.000 0.221
## [2,] 0.002 0.616 0.093 0.831 0.376 0.005 0.001 0.018 0.148 0.243 0.000 0.000 0.717
## [3,] 0.001 0.137 0.004 0.411 0.372 0.002 0.005 0.864 0.019 0.879 0.035 0.000 0.159
## [4,] 0.324 0.000 0.138 0.000 0.000 0.654 0.691 0.000 0.015 0.000 0.000 0.000 0.055
## [5,] 0.234 0.040 1.000 0.045 0.000 0.229 0.110 0.000 0.288 0.001 0.000 0.000 0.200
## [6,] 0.030 0.393 0.226 0.209 0.121 0.009 0.005 0.010 0.476 0.096 0.000 0.000 1.000
## [7,] 0.329 0.096 1.000 0.014 0.017 0.321 0.214 0.000 0.493 0.005 0.000 0.000 0.403
## [8,] 0.000 0.012 0.000 0.179 0.079 0.000 0.000 0.788 0.000 0.893 0.000 0.000 0.012
## [9,] 0.073 0.283 0.464 0.320 0.112 0.185 0.212 0.005 0.681 0.065 0.000 0.000 0.869
## [10,] 0.022 0.119 0.087 0.031 0.001 0.071 0.022 0.000 0.463 0.004 0.000 0.000 0.815
## [11,] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.145 0.000
## [12,] 1.000 0.014 0.320 0.026 0.000 0.731 1.000 0.000 0.102 0.000 0.000 0.000 0.192
## [13,] 1.000 0.003 0.449 0.005 0.001 1.000 0.802 0.000 0.149 0.000 0.000 0.000 0.052
## [14,] 0.003 1.000 0.008 1.000 0.326 0.010 0.001 0.000 0.023 0.096 0.000 0.000 0.274
## [15,] 0.449 0.008 1.000 0.009 0.000 0.457 0.172 0.000 0.381 0.000 0.000 0.000 0.061
## [16,] 0.005 1.000 0.009 1.000 0.649 0.001 0.001 0.050 0.031 0.410 0.000 0.000 0.300
## [17,] 0.001 0.326 0.000 0.649 1.000 0.002 0.000 0.025 0.000 0.389 0.000 0.000 0.104
## [18,] 1.000 0.010 0.457 0.001 0.002 1.000 1.000 0.000 0.173 0.000 0.000 0.000 0.153
## [19,] 0.802 0.001 0.172 0.001 0.000 1.000 1.000 0.000 0.034 0.000 0.000 0.000 0.030
## [20,] 0.000 0.000 0.000 0.050 0.025 0.000 0.000 1.000 0.000 0.297 0.000 0.000 0.000
## [21,] 0.149 0.023 0.381 0.031 0.000 0.173 0.034 0.000 1.000 0.000 0.000 0.000 0.362
## [22,] 0.000 0.096 0.000 0.410 0.389 0.000 0.000 0.297 0.000 1.000 0.000 0.000 0.012
## [23,] 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 1.000 0.000
## [24,] 0.052 0.274 0.061 0.300 0.104 0.153 0.030 0.000 0.362 0.012 0.000 0.000 1.000
## [25,] 0.009 0.869 0.006 0.854 0.260 0.072 0.004 0.000 0.047 0.112 0.000 0.000 0.433
## [26,] 0.007 0.843 0.056 0.788 0.277 0.010 0.008 0.019 0.130 0.191 0.000 0.000 0.512
## [27,] 0.059 0.127 0.252 0.033 0.003 0.098 0.047 0.000 0.512 0.002 0.000 0.000 0.869
## [28,] 0.000 0.036 0.000 0.168 0.163 0.001 0.001 0.638 0.004 0.577 0.094 0.032 0.032
## [29,] 0.000 0.142 0.000 0.294 0.775 0.000 0.000 0.060 0.000 0.887 0.000 0.000 0.010
## [30,] 0.001 1.000 0.010 1.000 0.475 0.009 0.000 0.018 0.042 0.320 0.000 0.000 0.347
## [31,] 1.000 0.012 0.375 0.016 0.000 1.000 0.797 0.000 0.080 0.000 0.000 0.000 0.086
## [32,] 0.000 0.830 0.001 1.000 0.858 0.002 0.000 0.019 0.020 0.304 0.000 0.000 0.288
## [33,] 0.047 0.075 0.624 0.014 0.005 0.078 0.048 0.000 0.623 0.007 0.000 0.000 0.512
## [34,] 0.060 0.367 0.493 0.390 0.185 0.090 0.149 0.015 0.731 0.100 0.000 0.000 0.890
## [35,] 0.003 0.798 0.126 0.671 0.456 0.007 0.005 0.059 0.259 0.409 0.000 0.000 0.653

```

```

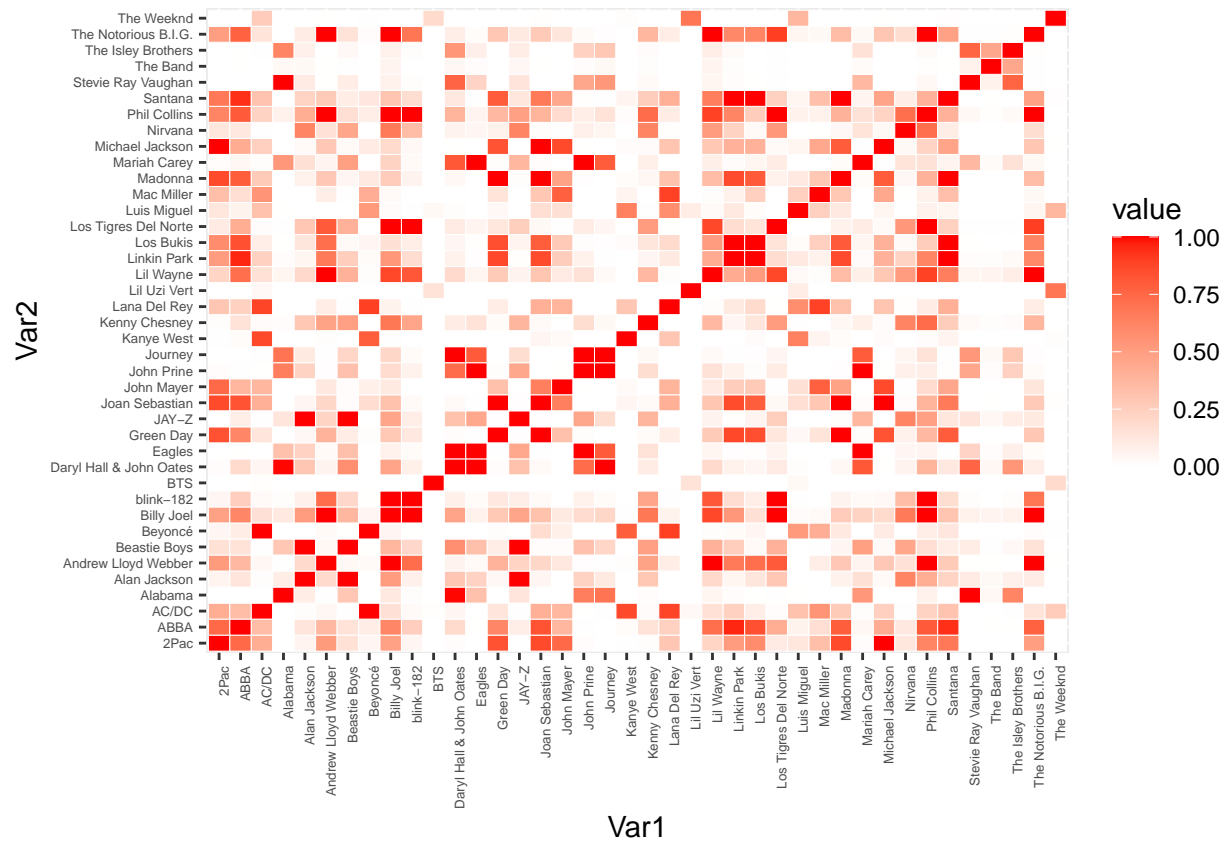
## [36,] 0.223 0.001 0.139 0.001 0.000 0.455 0.527 0.000 0.027 0.000 0.000 0.055
## [37,] 0.015 0.000 0.042 0.002 0.000 0.015 0.034 0.000 0.011 0.000 0.000 0.057
## [38,] 0.082 0.001 0.097 0.001 0.000 0.238 0.288 0.000 0.020 0.000 0.000 0.089
## [39,] 0.018 0.292 0.112 0.275 0.133 0.022 0.008 0.001 0.374 0.095 0.000 1.000
## [40,] 0.000 0.000 0.000 0.006 0.000 0.000 0.000 0.012 0.000 0.001 0.688 0.000
##      [,25] [,26] [,27] [,28] [,29] [,30] [,31] [,32] [,33] [,34] [,35] [,36]
## [1,] 0.508 0.586 0.105 0.126 0.335 0.863 0.015 1.000 0.126 0.618 0.674 0.000
## [2,] 0.956 0.836 0.425 0.065 0.170 0.790 0.043 0.429 0.120 0.803 0.936 0.000
## [3,] 0.240 0.089 0.016 0.319 0.548 0.272 0.013 0.247 0.005 0.232 0.311 0.001
## [4,] 0.016 0.002 0.004 0.002 0.000 0.000 0.531 0.000 0.001 0.071 0.001 1.000
## [5,] 0.053 0.133 0.292 0.001 0.000 0.074 0.162 0.030 0.621 0.421 0.227 0.105
## [6,] 0.669 0.718 0.800 0.000 0.091 0.286 0.058 0.286 0.156 1.000 0.271 0.002
## [7,] 0.256 0.032 0.402 0.001 0.003 0.050 0.492 0.030 0.461 0.166 0.094 0.095
## [8,] 0.023 0.045 0.001 0.517 0.424 0.116 0.000 0.104 0.001 0.092 0.124 0.000
## [9,] 0.528 0.164 1.000 0.019 0.032 0.270 0.229 0.189 0.668 1.000 0.303 0.074
## [10,] 0.176 0.087 1.000 0.003 0.000 0.062 0.031 0.035 0.350 1.000 0.173 0.010
## [11,] 0.000 0.000 0.000 0.033 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [12,] 0.067 0.050 0.107 0.011 0.000 0.037 0.813 0.002 0.063 0.388 0.121 0.759
## [13,] 0.009 0.007 0.059 0.000 0.000 0.001 1.000 0.000 0.047 0.060 0.003 0.223
## [14,] 0.869 0.843 0.127 0.036 0.142 1.000 0.012 0.830 0.075 0.367 0.798 0.001
## [15,] 0.006 0.056 0.252 0.000 0.000 0.010 0.375 0.001 0.624 0.493 0.126 0.139
## [16,] 0.854 0.788 0.033 0.168 0.294 1.000 0.016 1.000 0.014 0.390 0.671 0.001
## [17,] 0.260 0.277 0.003 0.163 0.775 0.475 0.000 0.858 0.005 0.185 0.456 0.000
## [18,] 0.072 0.010 0.098 0.001 0.000 0.009 1.000 0.002 0.078 0.090 0.007 0.455
## [19,] 0.004 0.008 0.047 0.001 0.000 0.000 0.797 0.000 0.048 0.149 0.005 0.527
## [20,] 0.000 0.019 0.000 0.638 0.060 0.018 0.000 0.019 0.000 0.015 0.059 0.000
## [21,] 0.047 0.130 0.512 0.004 0.000 0.042 0.080 0.020 0.623 0.731 0.259 0.027
## [22,] 0.112 0.191 0.002 0.577 0.887 0.320 0.000 0.304 0.007 0.100 0.409 0.000
## [23,] 0.000 0.000 0.000 0.094 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000
## [24,] 0.433 0.512 0.869 0.032 0.010 0.347 0.086 0.288 0.512 0.890 0.653 0.055
## [25,] 1.000 1.000 0.179 0.117 0.075 0.857 0.020 0.403 0.232 0.619 1.000 0.014
## [26,] 1.000 1.000 0.143 0.018 0.244 0.799 0.052 0.402 0.031 0.259 1.000 0.002
## [27,] 0.179 0.143 1.000 0.000 0.000 0.070 0.100 0.028 0.529 1.000 0.240 0.003
## [28,] 0.117 0.018 0.000 1.000 0.250 0.116 0.000 0.054 0.001 0.069 0.055 0.000
## [29,] 0.075 0.244 0.000 0.250 1.000 0.289 0.000 0.448 0.002 0.067 0.332 0.000
## [30,] 0.857 0.799 0.070 0.116 0.289 1.000 0.018 0.798 0.040 0.398 1.000 0.001
## [31,] 0.020 0.052 0.100 0.000 0.000 0.018 1.000 0.005 0.143 0.153 0.056 0.365
## [32,] 0.403 0.402 0.028 0.054 0.448 0.798 0.005 1.000 0.034 0.226 0.470 0.000
## [33,] 0.232 0.031 0.529 0.001 0.002 0.040 0.143 0.034 1.000 0.719 0.096 0.001
## [34,] 0.619 0.259 1.000 0.069 0.067 0.398 0.153 0.226 0.719 1.000 0.403 0.038
## [35,] 1.000 1.000 0.240 0.055 0.332 1.000 0.056 0.470 0.096 0.403 1.000 0.001
## [36,] 0.014 0.002 0.003 0.000 0.000 0.001 0.365 0.000 0.001 0.038 0.001 1.000
## [37,] 0.010 0.001 0.003 0.000 0.000 0.000 0.021 0.000 0.000 0.007 0.002 0.069
## [38,] 0.030 0.000 0.003 0.000 0.000 0.000 0.154 0.000 0.000 0.027 0.000 0.762
## [39,] 0.608 0.620 0.899 0.049 0.034 0.348 0.033 0.297 0.175 1.000 0.485 0.004
## [40,] 0.000 0.000 0.000 0.375 0.000 0.000 0.000 0.002 0.000 0.000 0.008 0.000
##      [,37] [,38] [,39] [,40]
## [1,] 0.000 0.000 0.499 0.001
## [2,] 0.002 0.002 0.774 0.001
## [3,] 0.001 0.003 0.141 0.266
## [4,] 0.037 0.620 0.001 0.000
## [5,] 0.034 0.078 0.099 0.000
## [6,] 0.002 0.000 1.000 0.002
## [7,] 0.003 0.035 0.152 0.001

```

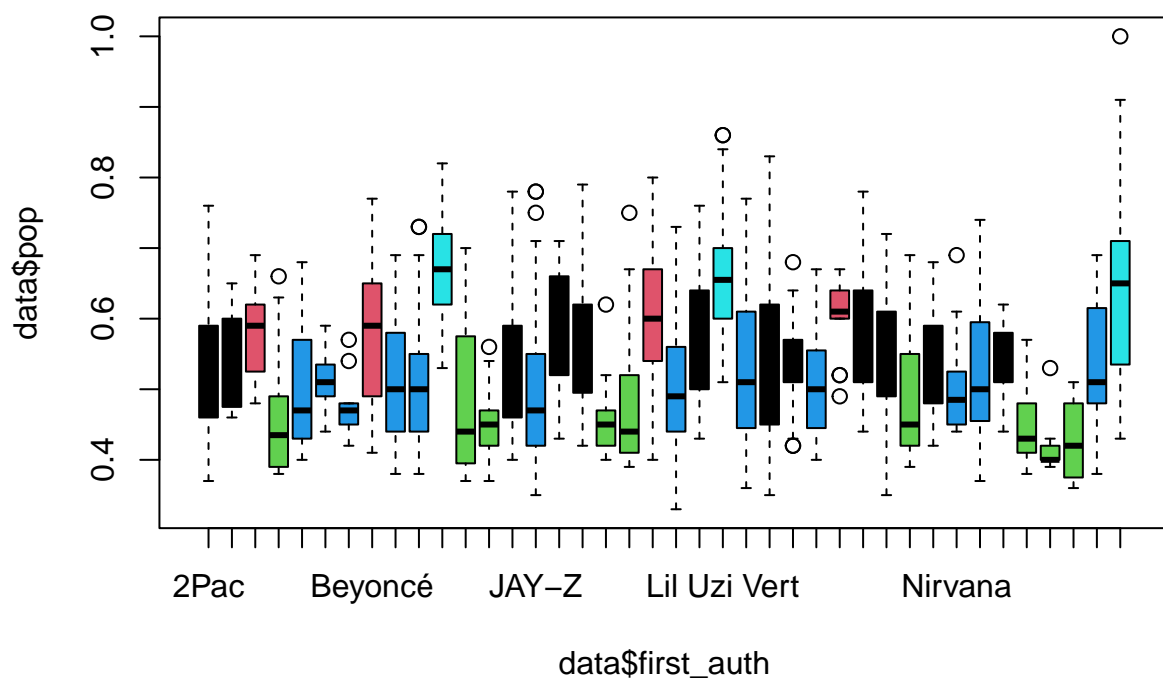
```
## [8,] 0.000 0.000 0.016 0.004
## [9,] 0.059 0.074 1.000 0.001
## [10,] 0.000 0.013 0.686 0.000
## [11,] 0.000 0.000 0.000 0.191
## [12,] 0.043 0.537 0.093 0.000
## [13,] 0.015 0.082 0.018 0.000
## [14,] 0.000 0.001 0.292 0.000
## [15,] 0.042 0.097 0.112 0.000
## [16,] 0.002 0.001 0.275 0.006
## [17,] 0.000 0.000 0.133 0.000
## [18,] 0.015 0.238 0.022 0.000
## [19,] 0.034 0.288 0.008 0.000
## [20,] 0.000 0.000 0.001 0.012
## [21,] 0.011 0.020 0.374 0.000
## [22,] 0.000 0.000 0.095 0.001
## [23,] 0.000 0.000 0.000 0.688
## [24,] 0.057 0.089 1.000 0.000
## [25,] 0.010 0.030 0.608 0.000
## [26,] 0.001 0.000 0.620 0.000
## [27,] 0.003 0.003 0.899 0.000
## [28,] 0.000 0.000 0.049 0.375
## [29,] 0.000 0.000 0.034 0.000
## [30,] 0.000 0.000 0.348 0.000
## [31,] 0.021 0.154 0.033 0.000
## [32,] 0.000 0.000 0.297 0.002
## [33,] 0.000 0.000 0.175 0.000
## [34,] 0.007 0.027 1.000 0.000
## [35,] 0.002 0.000 0.485 0.008
## [36,] 0.069 0.762 0.004 0.000
## [37,] 1.000 0.453 0.005 0.000
## [38,] 0.453 1.000 0.006 0.000
## [39,] 0.005 0.006 1.000 0.000
## [40,] 0.000 0.000 0.000 1.000
```

```
rownames(p_values) <- u_art
colnames(p_values) <- u_art

ggplot(melt(p_values), aes(Var1, Var2)) +
  geom_tile(aes(fill = value), colour = "white") +
  scale_fill_gradient(low = "white", high = "red") +
  theme(axis.text.x = element_text(angle=90, hjust = 1, siz=5),
        axis.text.y = element_text(hjust = 1, siz=5))
```



```
#punto d
boxplot( data$pop ~ data$first_auth,
         col= cutree( hclust(as.dist(1-p_values)),5))
```

```
#punto e
pop_scaled <- scale(data$pop)
df<- data.frame(Artist = data$first_auth,pop_scaled)

t_test<- function(df,artist){
  x<- df[df$Artist==artist,"pop_scaled"]
  t_test<- t.test(x,mu=0,alternative = "greater")
  return(t_test$p.value)
}

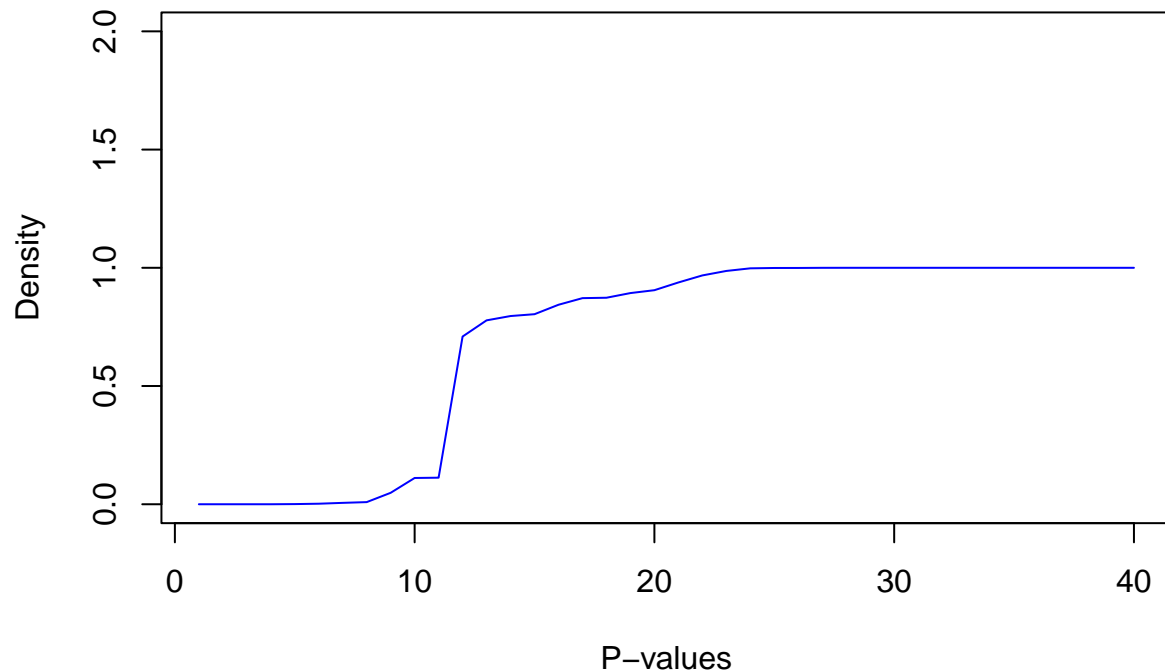
p_values<- sapply(unique(df$Artist),t_test,df=df)

prob_violation <- sum(p_values<0.10)/length(p_values)
prob_violation
```

```
## [1] 0.225
```

```
plot(sort(p_values), type="l",col = "blue", main = "Density Plot of P-values", xlab = "P-values", ylab = "Density")
```

Density Plot of P-values



```
#punto f
adjusted_bh <- p.adjust(p_values, method = "BH")
adjusted_bonferroni <- p.adjust(p_values, method = "bonferroni")
adjusted_holm <- p.adjust(p_values, method = "holm")
results <- data.frame(p_values, adjusted_bh, adjusted_bonferroni, adjusted_holm)

# Plotting the p-values and adjusted p-values
plot(sort(p_values), type="l", col = "blue", xlab = "Observation", ylab = "P-value",
     main = "Comparison of P-values and Adjusted P-values")
lines(sort(adjusted_bh), col = "red")
lines(sort(adjusted_bonferroni), col = "yellow")
lines(sort(adjusted_holm), col = "green")
abline(h = 0.05, lwd = 1, lty = 2)
legend("bottomright", legend = c("P-values", "Adjusted (BH)", "Adjusted (Bonferroni)", "Adjusted (Holm)",
                                col = c("blue", "red", "yellow", "green"), lwd=1,)
```

Comparison of P-values and Adjusted P-values

