B563\_hw3

## 1

x = runif(n=1000,min=-4,max=4)  
y = -x  
z = rep.int(1,1000)  
X = data.frame(x,y,z)  
  
cov = cov(X)  
svd = svd(cov)  
U = svd$u  
D = diag(svd$d)  
U

## [,1] [,2] [,3]  
## [1,] -0.7071068 0.7071068 0  
## [2,] 0.7071068 0.7071068 0  
## [3,] 0.0000000 0.0000000 1

D

## [,1] [,2] [,3]  
## [1,] 10.67632 0.00000e+00 0  
## [2,] 0.00000 6.28037e-16 0  
## [3,] 0.00000 0.00000e+00 0

## 2(a)

library("mosaic")

## Loading required package: 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

## Loading required package: lattice

## Loading required package: ggformula

## Loading required package: ggplot2

## Loading required package: ggstance

##   
## Attaching package: 'ggstance'

## The following objects are masked from 'package:ggplot2':  
##   
## geom\_errorbarh, GeomErrorbarh

##   
## New to ggformula? Try the tutorials:   
## learnr::run\_tutorial("introduction", package = "ggformula")  
## learnr::run\_tutorial("refining", package = "ggformula")

## Loading required package: mosaicData

## Loading required package: Matrix

## Registered S3 method overwritten by 'mosaic':  
## method from   
## fortify.SpatialPolygonsDataFrame ggplot2

##   
## The 'mosaic' package masks several functions from core packages in order to add   
## additional features. The original behavior of these functions should not be affected by this.  
##   
## Note: If you use the Matrix package, be sure to load it BEFORE loading mosaic.

##   
## Attaching package: 'mosaic'

## The following object is masked from 'package:Matrix':  
##   
## mean

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

## The following objects are masked from 'package:dplyr':  
##   
## count, do, tally

## The following objects are masked from 'package:stats':  
##   
## binom.test, cor, cor.test, cov, fivenum, IQR, median,  
## prop.test, quantile, sd, t.test, var

## The following objects are masked from 'package:base':  
##   
## max, mean, min, prod, range, sample, sum

df = read.csv("D:/Data Mining/hw3/Arrests.csv")  
head(df)

## X released colour year age sex employed citizen checks  
## 1 1 Yes White 2002 21 Male Yes Yes 3  
## 2 2 No Black 1999 17 Male Yes Yes 3  
## 3 3 Yes White 2000 24 Male Yes Yes 3  
## 4 4 No Black 2000 46 Male Yes Yes 1  
## 5 5 Yes Black 1999 27 Female Yes Yes 1  
## 6 6 Yes Black 1998 16 Female Yes Yes 0

t = table(df[,2],df[,3])  
t

##   
## Black White  
## No 333 559  
## Yes 955 3379

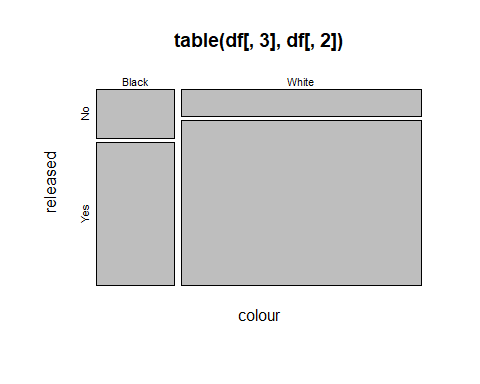
p\_br = t[2,1]/sum(t[2,1]+t[2,2])  
p\_br

## [1] 0.2203507

p\_wr = t[2,2]/sum(t[2,1]+t[2,2])  
p\_wr

## [1] 0.7796493

mosaicplot(table(df[,3],df[,2]),xlab="colour",ylab="released")



## 2(b)

* It is evident that given employment status, colour and released are not independent. Heights of blacks and white given employment status and released status is not same.

df1 = df[df$employed == 'Yes',]  
df2 = df[df$employed == 'No',]  
head(df1)

## X released colour year age sex employed citizen checks  
## 1 1 Yes White 2002 21 Male Yes Yes 3  
## 2 2 No Black 1999 17 Male Yes Yes 3  
## 3 3 Yes White 2000 24 Male Yes Yes 3  
## 4 4 No Black 2000 46 Male Yes Yes 1  
## 5 5 Yes Black 1999 27 Female Yes Yes 1  
## 6 6 Yes Black 1998 16 Female Yes Yes 0

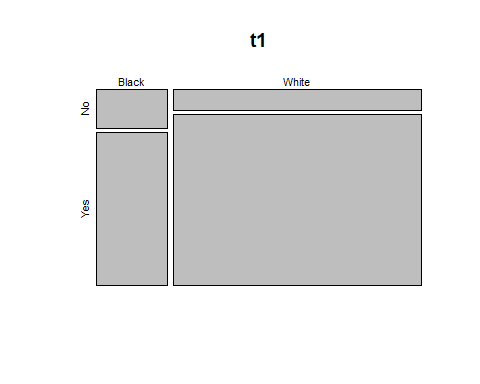
t1 = table(df1[,3],df1[,2])  
t1

##   
## No Yes  
## Black 185 726  
## White 358 2842

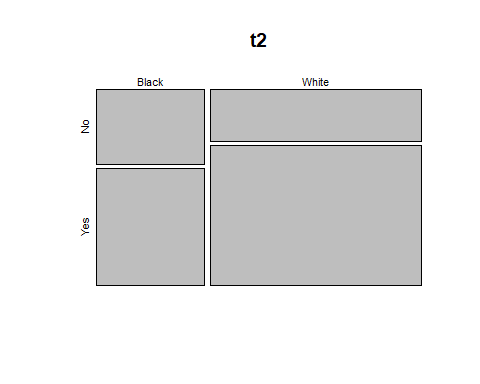
t2 = table(df2[,3],df2[,2])  
t2

##   
## No Yes  
## Black 148 229  
## White 201 537

tab1 = mosaicplot(t1)



tab2 = mosaicplot(t2)



## 2(c)

* It is evident that given checks, colour and released are not independent. Heights of blacks and white given check status and released status is not same.

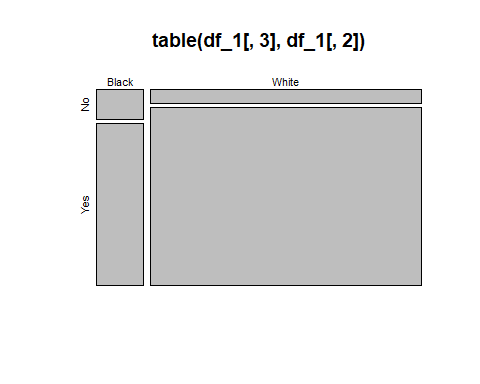
df\_1 = df[df$checks == 0,]  
df\_2 = df[df$checks == 1,]  
df\_3 = df[df$checks == 2,]  
df\_4 = df[df$checks == 3,]  
df\_5 = df[df$checks == 4,]  
df\_6 = df[df$checks == 5,]  
df\_7 = df[df$checks == 6,]  
  
t1 = table(df1[,3],df1[,2])  
t1

##   
## No Yes  
## Black 185 726  
## White 358 2842

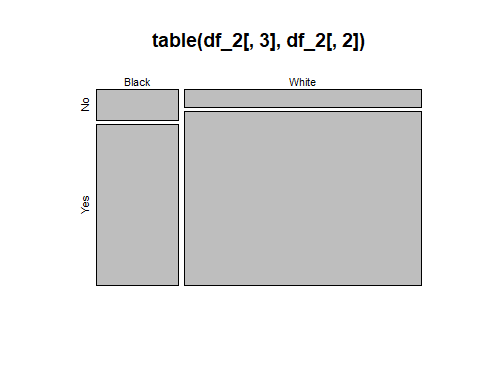
t2 = table(df2[,3],df2[,2])  
t2

##   
## No Yes  
## Black 148 229  
## White 201 537

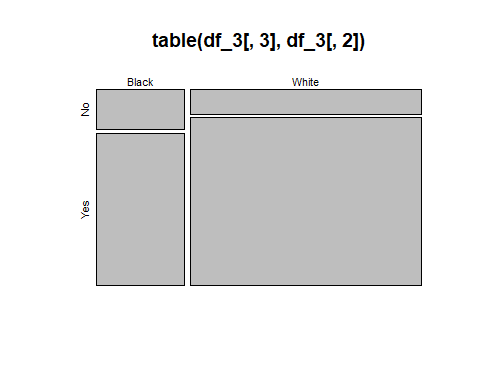
tab1 = mosaicplot(table(df\_1[,3],df\_1[,2]))



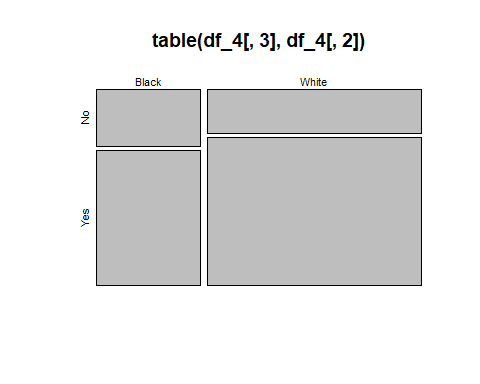
tab2 = mosaicplot(table(df\_2[,3],df\_2[,2]))



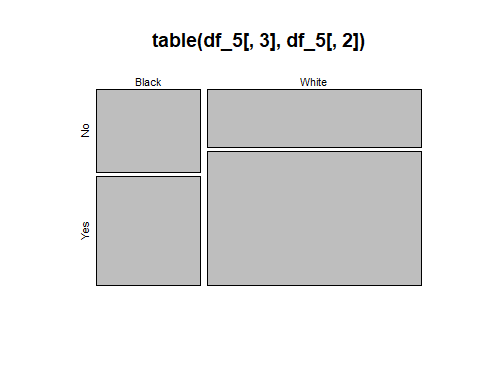
tab3 = mosaicplot(table(df\_3[,3],df\_3[,2]))



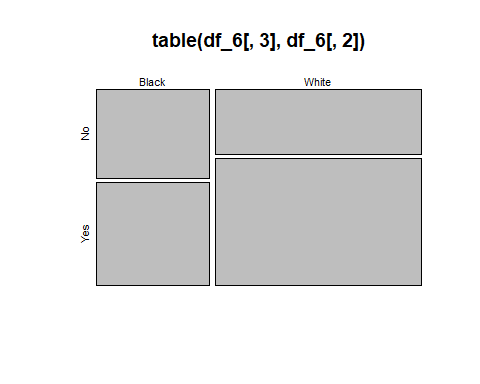
tab4 = mosaicplot(table(df\_4[,3],df\_4[,2]))



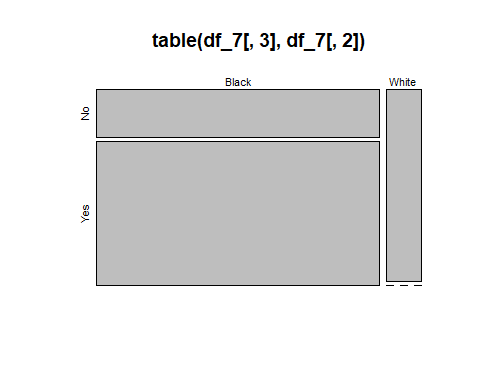
tab5 = mosaicplot(table(df\_5[,3],df\_5[,2]))



tab6 = mosaicplot(table(df\_6[,3],df\_6[,2]))



tab7 = mosaicplot(table(df\_7[,3],df\_7[,2]))

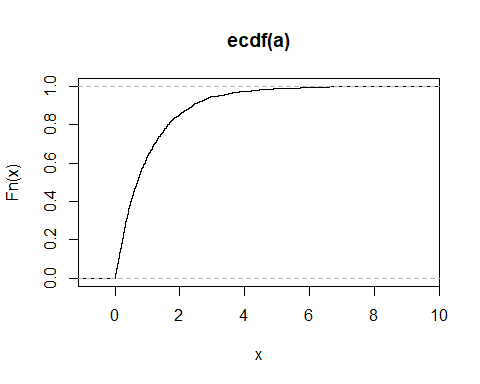


# 2(d)

* Given employment, we can see that whites are more probable to get released than blacks
* For checks also, we can see that whites are more probable to get released than blacks except when check = 7
* It is evident that higher arrest rate is there among blacks and data appears to be consistent with racial bias
* Also, there could be external factors which may affect the higher arrest rate of blacks but it cannot be explained through mosaic plot

# 3(a)

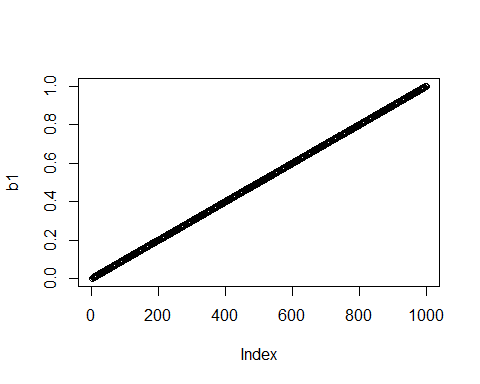
a = rexp(1000,1)  
plot(ecdf(a))



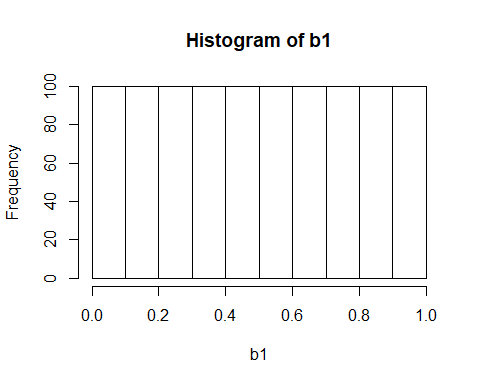
# 3(b)

* The distribution is uniform

b = ecdf(a)  
b1 = b(sort(a))  
plot(b1)



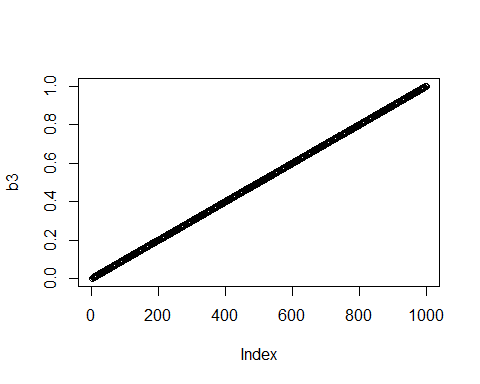
hist(b1)



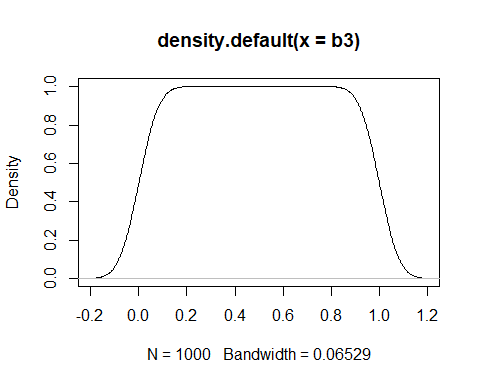
# 3(c)

* Assumed two distribution, one of which is exponentially distributed and other random distributed with each has 10 datapoints
* F(X<x(i)) = (number of X less than x(i))/1000 where i is the index
* For any given index i , F(X<x(i)) will have same probablity irrespective of value of i
* Thus, P(x) will have same distribution irrespective of any values of i

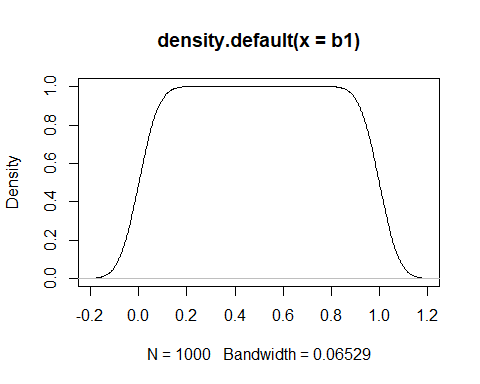
a1 = runif(1000)  
b2 = ecdf(a1)  
b3 = b2(sort(a1))  
plot(b3)



d = density(b3)  
plot(d)



d1 = density(b1)  
plot(d1)



## 4(a)

* The different models of time series has different amplitude and frequency.
* Model 1: It has low frequency and high amplitude
* Model 2: It has high frequency and low amplitude
* Model 3: It has high frequency and high amplitude
* Model 4 : It has low frequency and low amplitude

library(reshape2)  
library(ggplot2)  
library(forecast)

## Registered S3 method overwritten by 'xts':  
## method from  
## as.zoo.xts zoo

## Registered S3 method overwritten by 'quantmod':  
## method from  
## as.zoo.data.frame zoo

## Registered S3 methods overwritten by 'forecast':  
## method from   
## fitted.fracdiff fracdiff  
## residuals.fracdiff fracdiff

library(fpp2)

## Loading required package: fma

## Loading required package: expsmooth

library(ggplot2)  
library(tidyr)

##   
## Attaching package: 'tidyr'

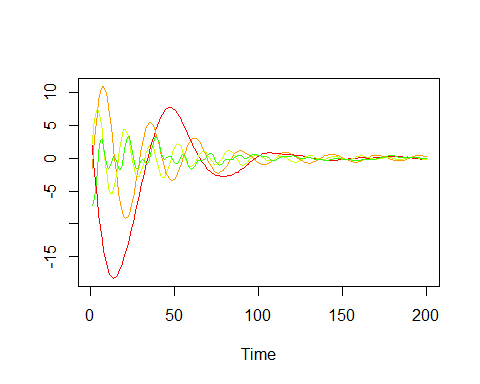
## The following object is masked from 'package:reshape2':  
##   
## smiths

## The following objects are masked from 'package:Matrix':  
##   
## expand, pack, unpack

df = read.csv("D:/Data Mining/hw3/time\_series.csv")  
head(df)

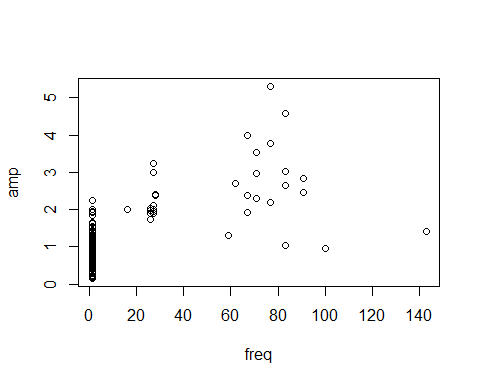
## V1 V2 V3 V4 V5 V6  
## 1 0.2414646 0.60220621 0.9371337 1.2520707 1.5484549 1.82128456  
## 2 0.8630342 0.04853502 -0.7290897 -1.4727296 -2.1711401 -2.81546451  
## 3 0.7192325 0.85314926 0.8597248 0.7453704 0.5173139 0.22531290  
## 4 0.6255325 0.12929581 -0.3419876 -0.7872321 -1.2011452 -1.57286333  
## 5 0.9544028 0.76850889 0.5976743 0.4297294 0.2582170 0.07674176  
## 6 0.9634639 0.53510341 0.0496688 -0.4357841 -0.8214009 -1.06860632  
## V7 V8 V9 V10 V11 V12  
## 1 2.0603278 2.2646880 2.4412774 2.5843382 2.691047235 2.7691792  
## 2 -3.3976259 -3.9351596 -4.4061131 -4.8157436 -5.164241808 -5.4391567  
## 3 -0.0928033 -0.3776043 -0.5856962 -0.6898406 -0.691000782 -0.5932595  
## 4 -1.8899962 -2.1796259 -2.4338976 -2.6597164 -2.868081642 -3.0301274  
## 5 -0.1003107 -0.2541000 -0.3833087 -0.5054255 -0.617276170 -0.7108297  
## 6 -1.1503108 -1.0456532 -0.7824648 -0.4079417 0.009840117 0.4276631  
## V13 V14 V15 V16 V17 V18  
## 1 2.8191881 2.8371034 2.83727274 2.8165010 2.7812759 2.6964495  
## 2 -5.6510528 -5.8096949 -5.90368007 -5.9329882 -5.8985044 -5.7895127  
## 3 -0.4039354 -0.1668375 0.05469909 0.2624700 0.4325709 0.5371693  
## 4 -3.1485551 -3.2242821 -3.26554096 -3.2736672 -3.2460081 -3.1861077  
## 5 -0.7838835 -0.8350688 -0.86679695 -0.8814907 -0.8974502 -0.9135124  
## 6 0.7482577 0.9553605 1.01639135 0.9244964 0.6901158 0.3685096  
## V19 V20 V21 V22 V23 V24  
## 1 2.6112727713 2.5226273 2.3950375 2.2717183 2.13125012 1.9764004  
## 2 -5.6335503732 -5.4220269 -5.1729175 -4.8918238 -4.56844963 -4.2159064  
## 3 0.5486620806 0.4750514 0.3386260 0.1389494 -0.06976423 -0.2535864  
## 4 -3.0845926360 -2.9599647 -2.7935015 -2.6086674 -2.41413354 -2.1953351  
## 5 -0.9152423367 -0.9027580 -0.8650264 -0.8326907 -0.78469305 -0.7430910  
## 6 0.0005068763 -0.3671456 -0.6526604 -0.8176531 -0.84770402 -0.7491650  
## V25 V26 V27 V28 V29 V30  
## 1 1.8068201 1.6445092 1.47634669 1.3054582 1.1245928 0.9426177  
## 2 -3.8282345 -3.4120616 -3.01057632 -2.6017898 -2.1995985 -1.7972349  
## 3 -0.4004984 -0.4809765 -0.47183117 -0.4002317 -0.2758444 -0.1092177  
## 4 -1.9589976 -1.7207068 -1.48812582 -1.2749105 -1.0771467 -0.8868897  
## 5 -0.6856495 -0.6287056 -0.57211488 -0.5321724 -0.4960174 -0.4611027  
## 6 -0.5676546 -0.3141285 -0.01851084 0.2710970 0.5082670 0.6461080  
## V31 V32 V33 V34 V35 V36  
## 1 0.77009638 0.5872326 0.4090238 0.2274396 0.05201333 -0.1113150  
## 2 -1.38290334 -0.9658190 -0.5589029 -0.1662976 0.20733668 0.5576234  
## 3 0.06175975 0.2159373 0.3344798 0.3912843 0.38012519 0.2960233  
## 4 -0.70765112 -0.5271186 -0.3398669 -0.1537926 0.03225267 0.1971532  
## 5 -0.41119487 -0.3785129 -0.3436862 -0.2972555 -0.25487199 -0.2202506  
## 6 0.67723047 0.6026500 0.4311881 0.2020576 -0.04737648 -0.2610710  
## V37 V38 V39 V40 V41 V42  
## 1 -0.2699646 -0.44265384 -0.60279748 -0.7419618 -0.8463262 -0.93446224  
## 2 0.8810721 1.18472015 1.46492261 1.7177754 1.9618097 2.15613518  
## 3 0.1899110 0.05858319 -0.06142357 -0.1697193 -0.2302467 -0.25740342  
## 4 0.3602271 0.49163244 0.61008778 0.7213486 0.8054279 0.86526369  
## 5 -0.1799220 -0.15727192 -0.13225865 -0.1172291 -0.1117786 -0.10165283  
## 6 -0.4247557 -0.50584310 -0.50304869 -0.4015672 -0.2382623 -0.05364887  
## V43 V44 V45 V46 V47 V48  
## 1 -1.02031635 -1.10785253 -1.17510439 -1.239525406 -1.28215440 -1.30461647  
## 2 2.30917011 2.43059773 2.53014853 2.602519173 2.64037239 2.64556046  
## 3 -0.23911213 -0.19617745 -0.12605863 -0.048320290 0.04093191 0.12953763  
## 4 0.91621578 0.93444810 0.92539786 0.894245136 0.82904171 0.76954376  
## 5 -0.08011641 -0.05534951 -0.02495447 0.001966358 0.02355569 0.03767469  
## 6 0.11100267 0.25834646 0.37495976 0.434071166 0.45628552 0.39952483  
## V49 V50 V51 V52 V53 V54  
## 1 -1.28494396 -1.23939846 -1.17327389 -1.0919580 -1.0007830 -0.90121247  
## 2 2.62486159 2.58071210 2.53236750 2.4667720 2.3706054 2.25171506  
## 3 0.21373832 0.24855486 0.26699162 0.2357333 0.1623774 0.05374744  
## 4 0.70851800 0.64509503 0.56885592 0.4877386 0.3960431 0.30942870  
## 5 0.05945271 0.08110275 0.10376762 0.1306466 0.1355377 0.13548528  
## 6 0.27708076 0.10717790 -0.06981289 -0.2160178 -0.3367857 -0.41912197  
## V55 V56 V57 V58 V59 V60  
## 1 -0.80550240 -0.71906931 -0.63241102 -0.53600445 -0.43745258 -0.36249098  
## 2 2.10994707 1.94424303 1.75180847 1.55839376 1.36000684 1.17735595  
## 3 -0.05943616 -0.15994585 -0.23687583 -0.26879823 -0.25432882 -0.21330116  
## 4 0.24159065 0.16961968 0.10611009 0.05343752 -0.01177671 -0.05974624  
## 5 0.12007024 0.09414092 0.08132866 0.07205238 0.06432573 0.06322110  
## 6 -0.43325280 -0.37192246 -0.23804605 -0.09938686 0.04035648 0.15742017  
## V61 V62 V63 V64 V65 V66  
## 1 -0.29362327 -0.22537246 -0.16635395 -0.1153266 -0.04406867 0.0184068  
## 2 1.00009474 0.82604254 0.66095652 0.5079303 0.36530158 0.2487957  
## 3 -0.12857577 -0.03178126 0.06215976 0.1424287 0.20197253 0.2231431  
## 4 -0.11508831 -0.16663249 -0.20201610 -0.2368849 -0.27001993 -0.2908967  
## 5 0.06851892 0.07523296 0.08358937 0.1081316 0.13126003 0.1549042  
## 6 0.27229774 0.34030517 0.36276983 0.3373260 0.27070248 0.1527084  
## V67 V68 V69 V70 V71 V72  
## 1 0.07004234 0.13370809 0.18852024 0.21622982 0.2310884 0.24070234  
## 2 0.14954271 0.04879158 -0.06301515 -0.17051978 -0.2785833 -0.39157133  
## 3 0.20738372 0.16376111 0.09917002 0.02110582 -0.0481482 -0.09441728  
## 4 -0.29961341 -0.30306388 -0.28674697 -0.26545383 -0.2429924 -0.22015859  
## 5 0.18219230 0.22704643 0.26917148 0.31749846 0.3574896 0.39254399  
## 6 0.01175244 -0.12632762 -0.23737322 -0.30136595 -0.3132747 -0.27233222  
## V73 V74 V75 V76 V77 V78  
## 1 0.2634417 0.27290979 0.2831534 0.28521155 0.28929092 0.3075297661  
## 2 -0.5021553 -0.59720383 -0.6754017 -0.74451874 -0.79935255 -0.8290338824  
## 3 -0.1279007 -0.13895499 -0.1180559 -0.08221841 -0.03437604 0.0009670538  
## 4 -0.1819007 -0.15880101 -0.1324383 -0.09458187 -0.07295101 -0.0649832711  
## 5 0.4228181 0.45486436 0.4692644 0.47212246 0.46557770 0.4457333384  
## 6 -0.1910519 -0.08651462 0.0217224 0.12245943 0.20398180 0.2613501262  
## V79 V80 V81 V82 V83 V84  
## 1 0.3228056 0.34252602 0.35848231 0.38107499 0.390159809 0.3953239  
## 2 -0.8655793 -0.88334739 -0.87244281 -0.86367636 -0.839746806 -0.8070081  
## 3 0.0433184 0.08881268 0.12934463 0.14245687 0.132575697 0.1187048  
## 4 -0.0700344 -0.06524920 -0.03176835 -0.01304836 0.012843904 0.0481796  
## 5 0.4064732 0.35485953 0.30688777 0.25949946 0.212109755 0.1597200  
## 6 0.2791966 0.26131633 0.19512075 0.10249348 -0.003305397 -0.1213538  
## V85 V86 V87 V88 V89 V90  
## 1 0.39835659 0.40484695 0.41420052 0.41762884 0.39911232 0.36853046  
## 2 -0.76784753 -0.71196221 -0.64007313 -0.58756620 -0.53747196 -0.49030436  
## 3 0.10009641 0.06629498 0.01455024 -0.04337916 -0.08614549 -0.10972369  
## 4 0.07728801 0.11274583 0.15071262 0.16567306 0.17044525 0.18073589  
## 5 0.10544015 0.07068675 0.03161395 -0.01534381 -0.05867301 -0.08050836  
## 6 -0.21789513 -0.27028644 -0.26849309 -0.20174797 -0.11801716 -0.03088701  
## V91 V92 V93 V94 V95 V96  
## 1 0.33557994 0.29738545 0.27002426 0.24402582 0.22019124 0.21464755  
## 2 -0.42647708 -0.36381230 -0.29274330 -0.22397028 -0.16398973 -0.11102170  
## 3 -0.11873338 -0.09348177 -0.05431855 -0.01609557 0.03101490 0.06261653  
## 4 0.17139884 0.15002189 0.12073691 0.08034810 0.01679041 -0.04649784  
## 5 -0.09885300 -0.13768968 -0.17376775 -0.21130891 -0.23353154 -0.26009921  
## 6 0.04811032 0.11279868 0.15331187 0.16379440 0.14495738 0.10872142  
## V97 V98 V99 V100 V101  
## 1 0.19117315 0.1720470021 0.16351654 0.14829964 0.13378613  
## 2 -0.05191743 0.0258400779 0.08909994 0.14786537 0.20577763  
## 3 0.07716939 0.0924273278 0.09876680 0.11108846 0.09992552  
## 4 -0.11732571 -0.1749265809 -0.23522088 -0.27989487 -0.31806981  
## 5 -0.30381068 -0.3548165543 -0.40961930 -0.47413520 -0.53253331  
## 6 0.06001288 -0.0007185779 -0.05104991 -0.09542442 -0.13347865  
## V102 V103 V104 V105 V106 V107  
## 1 0.12296125 0.10216694 0.07889526 0.05130411 0.02861647 0.01521957  
## 2 0.25234261 0.29728789 0.32150367 0.33794827 0.32789731 0.31498164  
## 3 0.08787254 0.04332161 -0.01189442 -0.07780434 -0.12634138 -0.14907640  
## 4 -0.34883390 -0.37372927 -0.40517745 -0.43039875 -0.47313720 -0.52085744  
## 5 -0.59060426 -0.63728893 -0.68144617 -0.70162397 -0.70831993 -0.72875524  
## 6 -0.15220043 -0.14289355 -0.10540236 -0.04719949 0.01480630 0.07706286  
## V108 V109 V110 V111 V112 V113  
## 1 -0.008799514 -0.04433452 -0.08670757 -0.12700173 -0.17853411 -0.23461663  
## 2 0.304893638 0.29681346 0.28541343 0.28532604 0.28409044 0.29102450  
## 3 -0.160524910 -0.13774701 -0.08402399 -0.03964849 0.01550363 0.06513209  
## 4 -0.559424780 -0.59136697 -0.60626017 -0.61102500 -0.62107638 -0.60964816  
## 5 -0.740037518 -0.75120633 -0.74575998 -0.72581999 -0.70699409 -0.67287243  
## 6 0.122616148 0.16043855 0.16566919 0.12619220 0.07676744 0.03039772  
## V114 V115 V116 V117 V118 V119  
## 1 -0.27157680 -0.32013993 -0.36979200 -0.40110510 -0.43108739 -0.44008476  
## 2 0.28595067 0.28129816 0.28443125 0.30208557 0.30740187 0.29448277  
## 3 0.09789064 0.12375535 0.13674957 0.13574363 0.10225401 0.04507294  
## 4 -0.58896286 -0.55684710 -0.50558263 -0.44589139 -0.37629386 -0.32461883  
## 5 -0.63887088 -0.61548212 -0.58657655 -0.54194687 -0.48734577 -0.43620772  
## 6 -0.00549277 -0.04226254 -0.06263307 -0.06620646 -0.04967723 -0.04414254  
## V120 V121 V122 V123 V124 V125  
## 1 -0.45819636 -0.47216927 -0.46841416 -0.47091834 -0.44572955 -0.41652835  
## 2 0.28092826 0.26525857 0.24593374 0.24720074 0.24100551 0.22468343  
## 3 -0.01521323 -0.06291792 -0.11260190 -0.15011958 -0.14819291 -0.10334693  
## 4 -0.27836231 -0.23340815 -0.18181067 -0.14487094 -0.10700323 -0.07850615  
## 5 -0.38015505 -0.32615471 -0.25087786 -0.18396558 -0.09763846 -0.01428864  
## 6 -0.04229395 -0.04217284 -0.02924245 -0.01025661 0.01250276 0.02350656  
## V126 V127 V128 V129 V130  
## 1 -0.373950629 -0.331309243 -0.2857801054 -0.231759990 -0.180764518  
## 2 0.207935569 0.193642165 0.1690436454 0.131194338 0.080514257  
## 3 -0.043233338 0.015269901 0.0778084622 0.124652909 0.151514187  
## 4 -0.050275392 -0.025761741 -0.0009509242 0.035917680 0.071064033  
## 5 0.079347503 0.164682036 0.2363995241 0.290615931 0.342345081  
## 6 0.002694321 -0.006727442 -0.0042617316 0.001347502 0.009301228  
## V131 V132 V133 V134 V135  
## 1 -0.12039295 -0.068179773 -0.02708585 -0.008391707 0.01387078  
## 2 0.03239408 -0.009734038 -0.05920859 -0.102215721 -0.13562819  
## 3 0.16458535 0.141797298 0.09258248 0.027220435 -0.03841397  
## 4 0.12041537 0.167387688 0.20617964 0.248695584 0.29347256  
## 5 0.39601993 0.459060678 0.51142994 0.564498231 0.60945686  
## 6 0.03590714 0.060727767 0.09035670 0.103954387 0.10270564  
## V136 V137 V138 V139 V140 V141  
## 1 0.04309927 0.07768229 0.11845544 0.140361082 0.18041332 0.21704704  
## 2 -0.17223622 -0.21333457 -0.27573938 -0.344503316 -0.39257774 -0.41398584  
## 3 -0.08797781 -0.14205948 -0.16723468 -0.170269337 -0.15492919 -0.13158584  
## 4 0.33506243 0.37006483 0.39575755 0.427502146 0.44673526 0.46873665  
## 5 0.65029504 0.69096676 0.72813278 0.749604900 0.75284312 0.72021165  
## 6 0.07326926 0.04837042 0.02843912 0.002858809 -0.01514941 -0.02610902  
## V142 V143 V144 V145 V146  
## 1 0.24212479 0.25633241 0.27056503 0.289212263 0.311086424  
## 2 -0.41762756 -0.42334214 -0.42000173 -0.414407138 -0.399402036  
## 3 -0.09815243 -0.05297910 -0.01537393 0.024151105 0.054832416  
## 4 0.48434484 0.49595465 0.50586830 0.498740818 0.483976066  
## 5 0.66835922 0.61387607 0.56650795 0.518731437 0.468599092  
## 6 -0.03200760 -0.04503494 -0.03068152 -0.009295474 0.005684713  
## V147 V148 V149 V150 V151  
## 1 0.35227712 0.36221922 0.38049036 0.39985609 0.404733143  
## 2 -0.39336257 -0.40162241 -0.40069154 -0.39359576 -0.384916447  
## 3 0.06269694 0.05750117 0.04800130 0.03454529 0.002147495  
## 4 0.47494453 0.47765050 0.48622018 0.47219628 0.444702049  
## 5 0.41427490 0.36527715 0.31289872 0.26779784 0.199459870  
## 6 0.01307103 0.01790596 0.02961501 0.02895766 0.012193265  
## V152 V153 V154 V155 V156 V157  
## 1 0.390814582 0.37517857 0.37010667 0.35140409 0.32628138 0.29259215  
## 2 -0.381795078 -0.37304280 -0.36715310 -0.34862102 -0.32144879 -0.29083561  
## 3 -0.021157340 -0.02468809 -0.03013951 -0.02626365 -0.02346302 -0.01458925  
## 4 0.429227559 0.40507947 0.37289304 0.33239801 0.29176091 0.26763070  
## 5 0.147517972 0.10391241 0.07110462 0.02787692 -0.02092002 -0.06663946  
## 6 -0.006296453 -0.01170980 -0.02010374 -0.01240551 0.01080517 0.03443003  
## V158 V159 V160 V161 V162  
## 1 0.259764174 0.22211898 0.19021862 0.14890937 0.113121409  
## 2 -0.262759673 -0.23755983 -0.20547366 -0.16060706 -0.125579052  
## 3 -0.005085083 0.01514454 0.05283970 0.08850326 0.112400590  
## 4 0.226113329 0.18080075 0.13547649 0.09299258 0.038712849  
## 5 -0.099238422 -0.12521954 -0.14850702 -0.17746346 -0.205401061  
## 6 0.041487007 0.03500508 0.01885598 0.01275539 -0.009572669  
## V163 V164 V165 V166 V167  
## 1 0.075611373 0.043067550 -0.007421412 -0.05274594 -0.09878942  
## 2 -0.101434988 -0.084778472 -0.055631283 -0.04621775 -0.02574370  
## 3 0.130089193 0.123143792 0.098324149 0.04117120 -0.01899870  
## 4 -0.012808583 -0.053911626 -0.090481900 -0.12705899 -0.15253370  
## 5 -0.209582462 -0.213243192 -0.201970119 -0.19088771 -0.16842615  
## 6 -0.009327035 0.002680478 0.012923842 0.02079036 0.02686607  
## V168 V169 V170 V171 V172  
## 1 -0.136385457 -0.17001994 -0.18246211 -0.19857641 -0.193704159  
## 2 -0.009036198 0.01459270 0.03635203 0.05768524 0.072276753  
## 3 -0.064280850 -0.09690267 -0.09472589 -0.06931249 -0.038048315  
## 4 -0.179729158 -0.21105651 -0.26884780 -0.30550655 -0.335505488  
## 5 -0.143492183 -0.10842653 -0.06842098 -0.03388086 -0.005975241  
## 6 0.046136054 0.07717613 0.10032795 0.11358196 0.106139571  
## V173 V174 V175 V176 V177 V178  
## 1 -0.19300574 -0.21361930 -0.23485167 -0.27075744 -0.30886142 -0.33631697  
## 2 0.08631618 0.11010480 0.13895436 0.16207014 0.18963125 0.22232358  
## 3 -0.01101382 0.01177768 0.03497744 0.03870726 0.05532633 0.07308608  
## 4 -0.37307301 -0.40792120 -0.44632195 -0.47499327 -0.49605787 -0.52613454  
## 5 0.02842068 0.05138828 0.05954828 0.07963562 0.08996278 0.10962632  
## 6 0.06104721 0.01502074 -0.02000512 -0.05801580 -0.10161535 -0.13526686  
## V179 V180 V181 V182 V183 V184  
## 1 -0.35365424 -0.36225809 -0.38210175 -0.39798896 -0.38918897 -0.380833538  
## 2 0.25810729 0.28887371 0.32604596 0.35724169 0.37447546 0.387096071  
## 3 0.07983301 0.07889331 0.06830907 0.05502149 0.02573839 0.007321899  
## 4 -0.56048964 -0.60279682 -0.62869173 -0.65766678 -0.67876042 -0.691801383  
## 5 0.13263818 0.16098635 0.20259190 0.22894989 0.24215531 0.244804938  
## 6 -0.15025091 -0.13767461 -0.08844604 -0.03121101 0.02631496 0.074731739  
## V185 V186 V187 V188 V189 V190  
## 1 -0.33929272 -0.29590699 -0.24712469 -0.21155989 -0.17308156 -0.140467555  
## 2 0.39154206 0.40445021 0.42134155 0.43773560 0.45253411 0.453748251  
## 3 -0.02391210 -0.04634356 -0.05017604 -0.04137622 -0.01500944 0.002016862  
## 4 -0.70301539 -0.71487517 -0.72317024 -0.73215530 -0.72736968 -0.703863093  
## 5 0.24218903 0.24365241 0.23374323 0.21611359 0.18008997 0.131336900  
## 6 0.09866032 0.11693955 0.12896767 0.11990367 0.10584746 0.072285003  
## V191 V192 V193 V194 V195 V196  
## 1 -0.11914984 -0.09476114 -0.05923107 -0.01857949 0.02455808 0.06638725  
## 2 0.46557004 0.46868578 0.46566829 0.44555326 0.41290440 0.36580215  
## 3 0.01161533 0.01928440 0.02218878 0.03097357 0.03412825 0.04096053  
## 4 -0.67392401 -0.63760356 -0.59004882 -0.52848563 -0.46816472 -0.41320700  
## 5 0.07814523 0.01585358 -0.04790319 -0.10264015 -0.15434664 -0.20427522  
## 6 0.02740579 -0.02011726 -0.06388656 -0.08999478 -0.07135949 -0.03751333  
## V197 V198 V199 V200  
## 1 0.094032668 0.124261160 0.16145333 0.20609795  
## 2 0.305389418 0.252972876 0.18162990 0.11529921  
## 3 0.024761551 0.006815464 -0.00501051 -0.01445861  
## 4 -0.349877160 -0.286045156 -0.22699702 -0.17914945  
## 5 -0.251573352 -0.320685918 -0.37272698 -0.42463959  
## 6 -0.007566694 0.022941656 0.06085636 0.08134216

df\_t = t(df)  
cov = cov(df\_t)  
svd = svd(cov)  
U = svd$u  
  
X = df\_t%\*%U[,1:4]  
  
ts.plot(X,gpars=list(col=rainbow(10)))



# 4(b)

library("forecast")  
amp = apply(df, 1, max)  
  
df\_t = t(df)  
  
freq = rep(NA,200)  
for (i in 1:ncol(df\_t))  
 freq[i] = findfrequency(df\_t[,i])  
  
plt = plot(freq,amp)



plt

## NULL