D3

2025-06-03

```r
#setwd("/dades/eric.diez/6Q/ADEI")
setwd("C:/Users/Ericd/Pictures/ADEI")
dd <- read.csv("adult_def.csv", stringsAsFactors = TRUE);
names(dd)
```

```
##  [1] "age"            "workclass"      "fnlwgt"         "edu_num"
##  [5] "marital"        "occupation"     "relationship"   "race"
##  [9] "sex"            "cap_gain"       "cap_loss"       "hours_week"
## [13] "native_country" "income"         "income_integer"
```

```r
dim(dd)
```

```
## [1] 48842    15
```

```r
summary(dd)
```

```
##       age        workclass         fnlwgt           edu_num
##  Min.   :17.00   Fed  : 1432   Min.   :  12285   Min.   : 1.00
##  1st Qu.:28.00   Loc  : 3136   1st Qu.: 117551   1st Qu.: 9.00
##  Median :37.00   NoPay:   31   Median : 178145   Median :10.00
##  Mean   :38.64   Priv :36705   Mean   : 189664   Mean   :10.08
##  3rd Qu.:48.00   SelfI: 1695   3rd Qu.: 237642   3rd Qu.:12.00
##  Max.   :90.00   SelfN: 3862   Max.   :1490400   Max.   :16.00
##                  State: 1981
##     marital          occupation            relationship
##  Div    : 6633   Prof     : 8981   Husband       :19716
##  Married:22416   CraftRep : 6112   Not-in-family :12583
##  NevMarr:16117   ExecMan  : 6086   Other-relative: 1506
##  Sep    : 2158   AdminCler: 5611   Own-child     : 7581
##  Widow  : 1518   Sales    : 5504   Unmarried     : 5125
##                  Other    : 4923   Wife          : 2331
##                  (Other)  :11625
##                  race           sex           cap_gain        cap_loss
##  Amer-Indian-Eskimo:  470   Female:16192   Min.   :    0   Min.   :   0.0
##  Asian-Pac-Islander: 1519   Male  :32650   1st Qu.:    0   1st Qu.:   0.0
##  Black             : 4685                  Median :    0   Median :   0.0
##  Other             :  406                  Mean   : 1079   Mean   :  87.5
##  White             :41762                  3rd Qu.:    0   3rd Qu.:   0.0
##                                            Max.   :99999   Max.   :4356.0
##
##    hours_week    native_country   income      income_integer
##  Min.   : 1.00   Other: 5010    <=50K:37155   Min.   :27850
##  1st Qu.:40.00   USA  :43832    >50K :11687   1st Qu.:40820
##  Median :40.00                                Median :44870
##  Mean   :40.42                                Mean   :45684
##  3rd Qu.:45.00                                3rd Qu.:49835
##  Max.   :99.00                                Max.   :80040
```

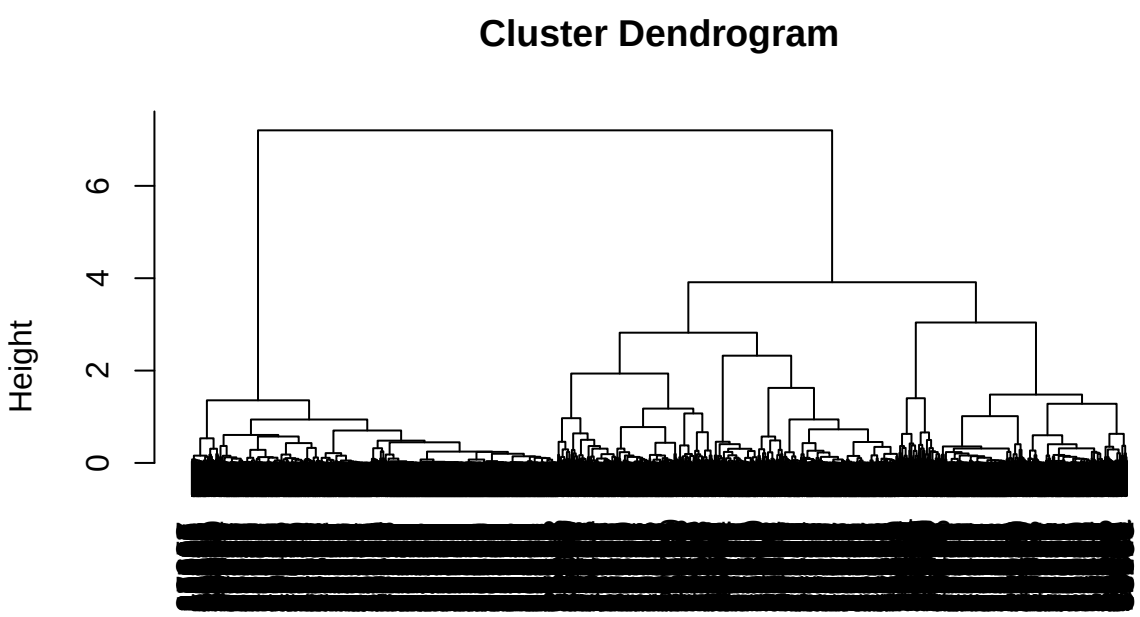
##

```
set.seed(123)
samp<-sample(48842,5000)
par(mar = c(5, 4, 4, 2))  # aumenta espacio inferior


dd<-dd[samp,]
#set a list of numerical variables
names(dd)
```

```
##  [1] "age"            "workclass"      "fnlwgt"         "edu_num"
##  [5] "marital"        "occupation"     "relationship"   "race"
##  [9] "sex"            "cap_gain"       "cap_loss"       "hours_week"
## [13] "native_country" "income"         "income_integer"
```

```
attach(dd)
```

```
#euclidean distance si totes son numeriques
dcon<-data.frame (age,edu_num,cap_gain,cap_loss,hours_week)

d  <- dist(dcon[1:6,])

#move to Gower mixed distance to deal
#simoultaneously with numerical and qualitative data

library(cluster)

#dissimilarity matrix
#do not include in actives the identifier variables nor the potential response variable

actives <- c("age", "workclass", "edu_num",
             "marital", "occupation", "relationship", "race", "sex",
             "cap_gain", "cap_loss", "hours_week", "native_country")
actives <- c(1:2,4:13)
dissimMatrix <- daisy(dd[,actives], metric = "gower", stand=TRUE)
distMatrix <- as.dist(dissimMatrix^2)

h1 <- hclust(distMatrix,method="ward.D2")  # NOTICE THE COST
#versions noves "ward.D" i abans de plot: par(mar=rep(2,4)) si se quejara de los margenes del plot

plot(h1)
```



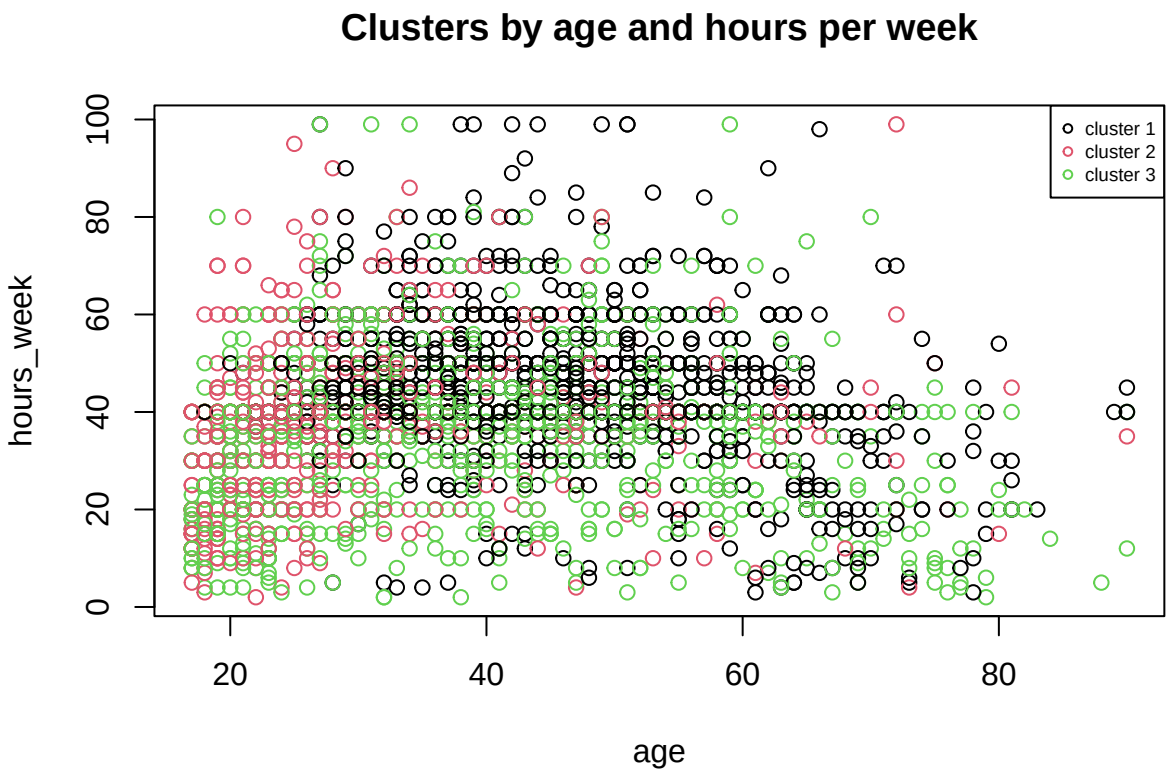

```
k<-3

c2 <- cutree(h1, k=k)
```

Añadir cluster al dataset

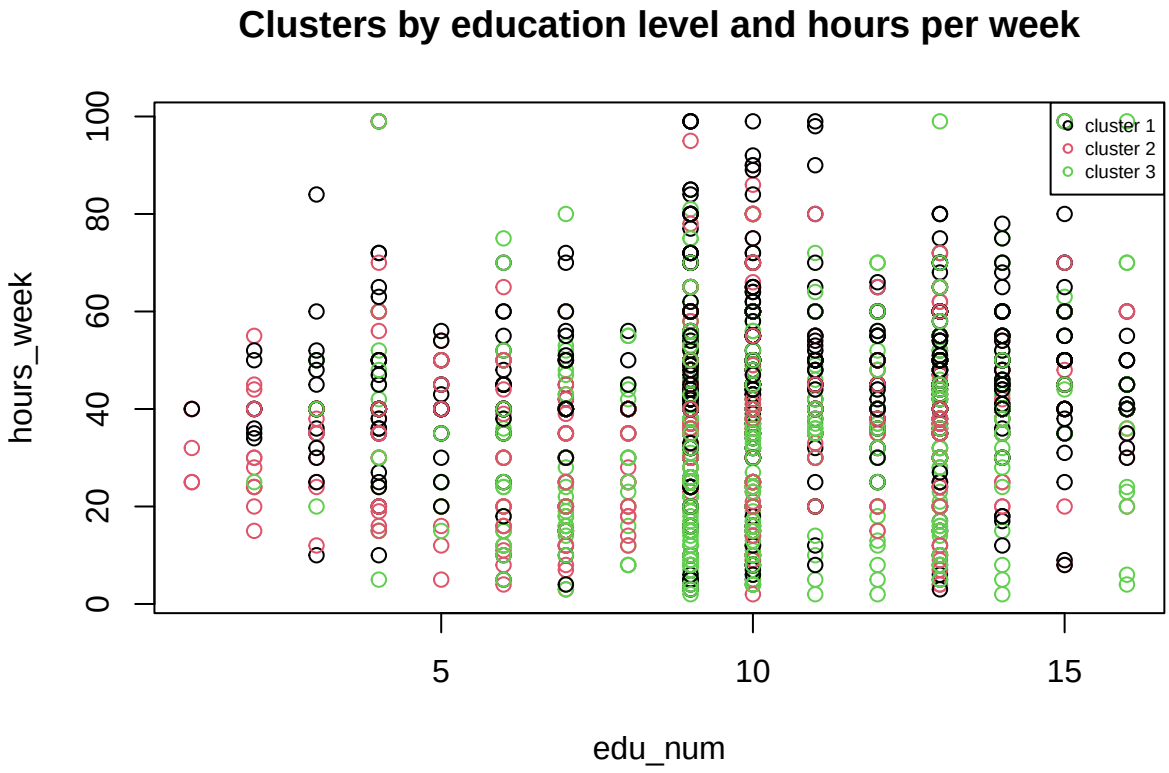
```
dd$cluster <- as.factor(c2)

# Análisis descriptivo de los clusters
table(dd$cluster)
```

```
##
##    1    2    3
## 1957 1221 1822
```

```
# LETS SEE THE PARTITION VISUALLY

c1<-c2
# Basic scatter plots for three key numeric variables (modified version)
plot(age, hours_week, col=c1, main="Clusters by age and hours per week")
legend("topright", paste("cluster", 1:k), pch=1, col=1:k, cex=0.6)
```

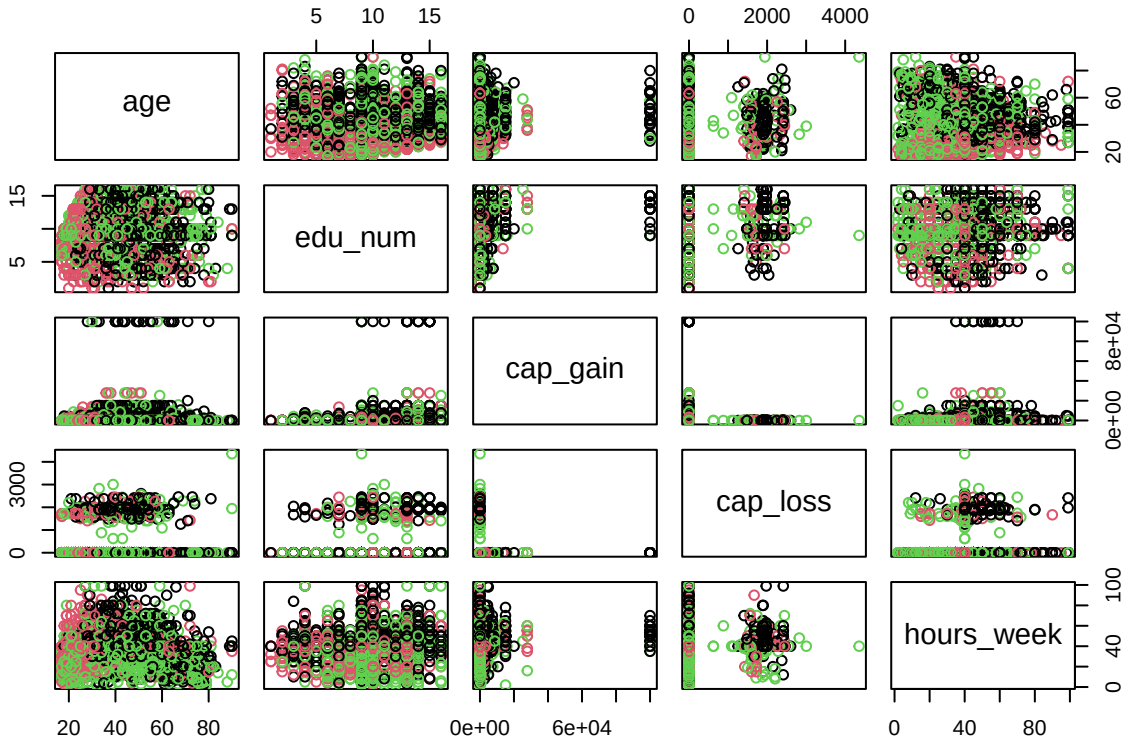



```
plot(edu_num, hours_week, col=c1, main="Clusters by education level and hours per week")
legend("topright", paste("cluster", 1:k), pch=1, col=1:k, cex=0.6)
```




```
plot(age, edu_num, col=c1, main="Clusters by age and education level")
legend("topright", paste("cluster", 1:k), pch=1, col=1:k, cex=0.6)
```


```
pairs(dcon[,1:5], col=c1)
```

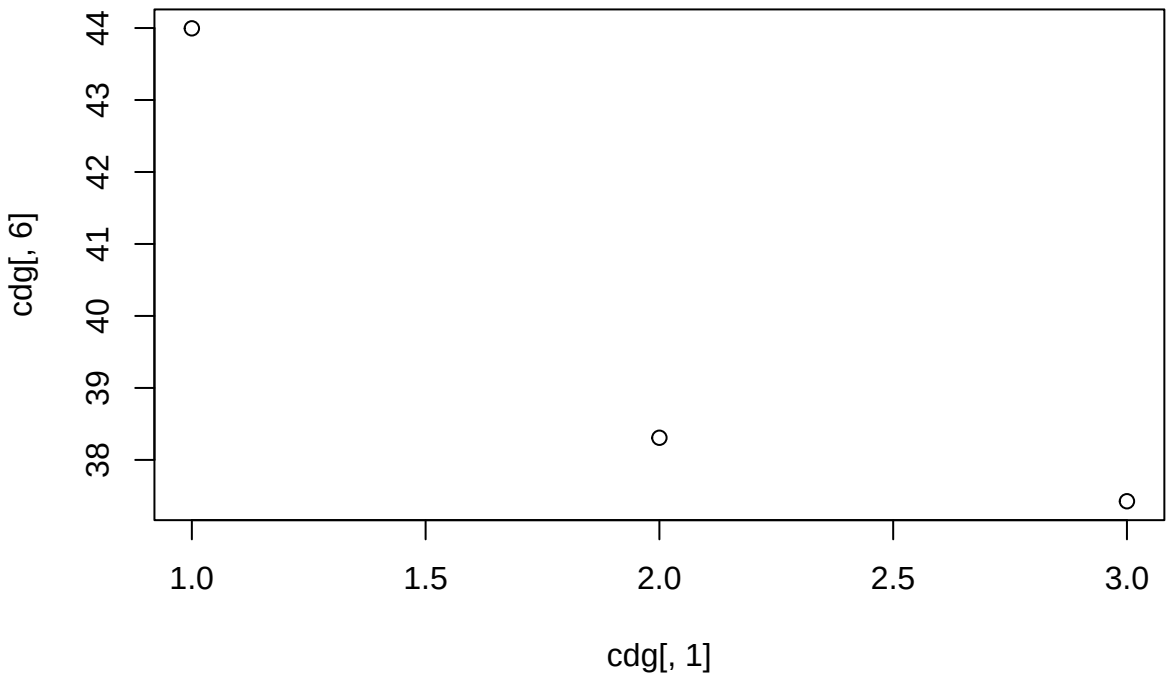


LETS SEE THE QUALITY OF THE HIERARCHICAL PARTITION

```
cdg <- aggregate(as.data.frame(dcon), list(c1), mean)
cdg
```

```
##   Group.1      age   edu_num  cap_gain  cap_loss hours_week
## 1       1 44.17527 10.273889 1922.9918 138.15330   43.99540
## 2       2 30.63718  9.594595  297.0614  54.32514   38.30794
## 3       3 37.61690 10.135565  544.7629  65.65642   37.42591
```

```
plot(cdg[,1], cdg[,6])
```




```r
#Profiling plots

#Calcula els valor test de la variable Xnum per totes les modalitats del factor P
ValorTestXnum <- function(Xnum,P){
  #freq dis of fac
  nk <- as.vector(table(P));
  n <- sum(nk);
  #mitjanes x grups
  xk <- tapply(Xnum,P,mean);
  #valors test
  txk <- (xk-mean(Xnum))/(sd(Xnum)*sqrt((n-nk)/(n*nk)));
  #p-values
  pxk <- pt(txk,n-1,lower.tail=F);
  for(c in 1:length(levels(as.factor(P)))){if (pxk[c]>0.5){pxk[c]<-1-pxk[c]}}
  return (pxk)
}



ValorTestXquali <- function(P,Xquali){
  taula <- table(P,Xquali);
  n <- sum(taula);
  pk <- apply(taula,1,sum)/n;
  pj <- apply(taula,2,sum)/n;
  pf <- taula/(n*pk);
```

```
  pjm <- matrix(data=pj,nrow=dim(pf)[1],ncol=dim(pf)[2], byrow=TRUE);
  dpf <- pf - pjm;
  dvt <- sqrt(((1-pk)/(n*pk))%*%t(pj*(1-pj)));
  #i hi ha divisions iguals a 0 dona NA i no funciona
  zkj <- dpf
  zkj[dpf!=0]<-dpf[dpf!=0]/dvt[dpf!=0];
  pzkj <- pnorm(zkj,lower.tail=F);
  for(c in 1:length(levels(as.factor(P)))){for (s in 1:length(levels(Xquali))){if (pzkj[c,s]> 0.5){pzkj
  return (list(rowpf=pf,vtest=zkj,pval=pzkj))
}
```

```
#source("file")
#dades contain the dataset
dades <- dd[, setdiff(names(dd),
                      c("fnlwgt", "income", "cluster", "income_integer"))]
#dades<-dd[filtro,]
#dades<-df
K<-dim(dades)[2]
par(ask=TRUE)


#P must contain the class variable
#P<-dd[,3]
P<-c2
#P<-dd[,18]
nameP<-"classe"
#P<-df[,33]

nc<-length(levels(factor(P)))
nc
```

```
## [1] 3
```

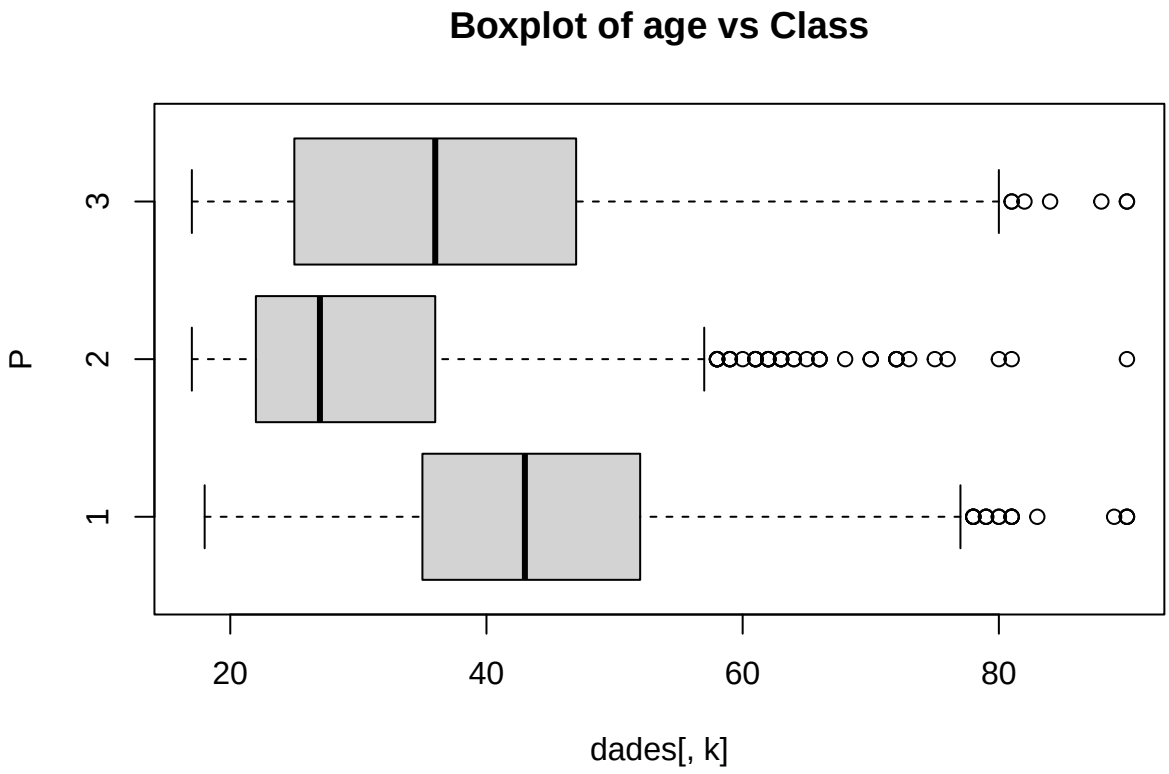
```
pvalk <- matrix(data=0,nrow=nc,ncol=K, dimnames=list(levels(P),names(dades)))
nameP<-"Class"
n<-dim(dades)[1]

for(k in 1:K){
  if (is.numeric(dades[,k])){
    print(paste("Anàlisi per classes de la Variable:", names(dades)[k]))

    boxplot(dades[,k]~P, main=paste("Boxplot of", names(dades)[k], "vs", nameP ), horizontal=TRUE)

    barplot(tapply(dades[[k]], P, mean),main=paste("Means of", names(dades)[k], "by", nameP ))
    abline(h=mean(dades[[k]]))
    legend(0,mean(dades[[k]]),"global mean",bty="n")
    print("Estadístics per groups:")
    for(s in levels(as.factor(P))) {print(summary(dades[P==s,k]))}
    o<-oneway.test(dades[,k]~P)
    print(paste("p-valueANOVA:", o$p.value))
    kw<-kruskal.test(dades[,k]~P)
    print(paste("p-value Kruskal-Wallis:", kw$p.value))
    pvalk[,k]<-ValorTestXnum(dades[,k], P)
    print("p-values ValorsTest: ")
    print(pvalk[,k])
```

```
}else{
  if(class(dd[,k])=="Date"){
    print(summary(dd[,k]))
    print(sd(dd[,k]))
    #decide breaks: weeks, months, quarters...
    hist(dd[,k],breaks="weeks")
  }else{
    #qualitatives
    print(paste("Variable", names(dades)[k]))
    table<-table(P,dades[,k])
    #   print("Cross-table")
    #   print(table)
    rowperc<-prop.table(table,1)

    colperc<-prop.table(table,2)
    #  print("Distribucions condicionades a files")
    # print(rowperc)

    #ojo porque si la variable es true o false la identifica amb el tipus Logical i
    #aquest no te levels, por tanto, coertion preventiva

    dades[,k]<-as.factor(dades[,k])


    marg <- table(as.factor(P))/n
    print(append("Categories=",levels(as.factor(dades[,k]))))
    par(mar = c(8, 4, 4, 2))  # más espacio abajo


    #from next plots, select one of them according to your practical case
    plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of classes by",names(dades)[k]))
    paleta<-rainbow(length(levels(dades[,k])))
    for(c in 1:length(levels(dades[,k]))){lines(colperc[,c],col=paleta[c]) }

    #with legend
    par(mar = c(5, 4, 4, 6))
    plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of classes by",names(dades)[k]))
    paleta<-rainbow(length(levels(dades[,k])))
    for(c in 1:length(levels(dades[,k]))){lines(colperc[,c],col=paleta[c]) }
    legend("topright",inset = c(-0.19, 0), levels(dades[,k]), col=paleta, lty=2, cex=0.6, bg = "transp
    par(mar = c(8, 4, 4, 2))

    #condicionades a classes
    print(append("Categories=",levels(dades[,k])))
    plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of classes by",names(dades)[k]))
    paleta<-rainbow(length(levels(dades[,k])))
    for(c in 1:length(levels(dades[,k]))){lines(rowperc[,c],col=paleta[c]) }

    #with legend
    plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of classes by",names(dades)[k]))
    paleta<-rainbow(length(levels(dades[,k])))
    for(c in 1:length(levels(dades[,k]))){lines(rowperc[,c],col=paleta[c]) }
    legend("topright", levels(dades[,k]), col=paleta, lty=2, cex=0.6)
```

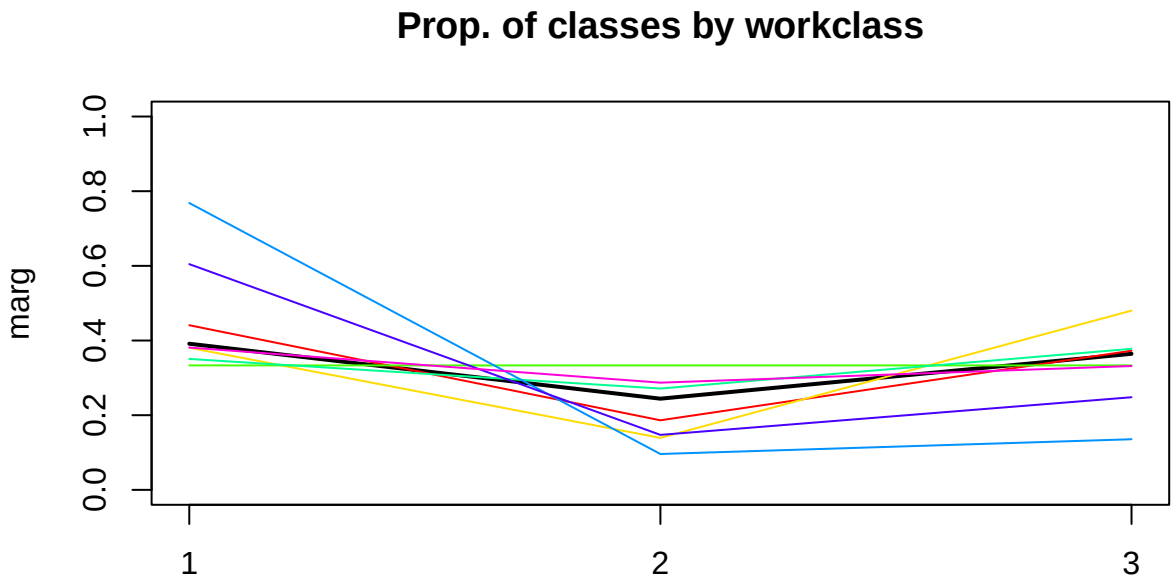
```r
#amb variable en eix d'abcisses
marg <-table(dades[,k])/n
print(append("Categories=",levels(dades[,k])))
plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of classes by",names(dades)[k]), las=3)
#x<-plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of classes by",names(dades)[k]), xaxt="n")
#text(x=x+.25, y=-1, adj=1, levels(CountryName), xpd=TRUE, srt=25, cex=0.7)
paleta<-rainbow(length(levels(as.factor(P))))
for(c in 1:length(levels(as.factor(P)))){lines(rowperc[c,],col=paleta[c]) }

#with legend
plot(marg,type="l",ylim=c(0,1),main=paste("Prop. of classes by",names(dades)[k]), las=3)
for(c in 1:length(levels(as.factor(P)))){lines(rowperc[c,],col=paleta[c])}
legend("topright", levels(as.factor(P)), col=paleta, lty=2, cex=0.6)

#condicionades a columna
plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of classes by",names(dades)[k]), las=3)
paleta<-rainbow(length(levels(as.factor(P))))
for(c in 1:length(levels(as.factor(P)))){lines(colperc[c,],col=paleta[c]) }

#with legend
par(mar = c(8, 4, 4, 6))
plot(marg,type="n",ylim=c(0,1),main=paste("Prop. of classes by",names(dades)[k]), las=3)
for(c in 1:length(levels(as.factor(P)))){lines(colperc[c,],col=paleta[c])}
legend("topright", inset = c(-0.10, 0),levels(as.factor(P)), col=paleta, lty=2, cex=0.6, bg = "tra
par(mar = c(8, 4, 4, 2))

table<-table(dades[,k],P)
print("Cross Table:")
print(table)
print("Distribucions condicionades a columnes:")
print(colperc)

#diagrames de barres apilades

paleta<-rainbow(length(levels(dades[,k])))
barplot(table(dades[,k], as.factor(P)), beside=FALSE,col=paleta )

barplot(table(dades[,k], as.factor(P)), beside=FALSE,col=paleta )
legend("topright",levels(as.factor(dades[,k])),pch=1,cex=0.5, col=paleta)

#diagrames de barres adosades
barplot(table(dades[,k], as.factor(P)), beside=TRUE,col=paleta )

par(mar = c(5, 4, 4, 6))
barplot(table(dades[,k], as.factor(P)), beside=TRUE,col=paleta)
legend("topright", inset = c(-0.17, 0),levels(as.factor(dades[,k])),pch=1,cex=0.5, col=paleta, bg
par(mar = c(8, 4, 4, 2))

print("Test Chi quadrat: ")
print(chisq.test(dades[,k], as.factor(P)))

print("valorsTest:")
print( ValorTestXquali(P,dades[,k]))
```

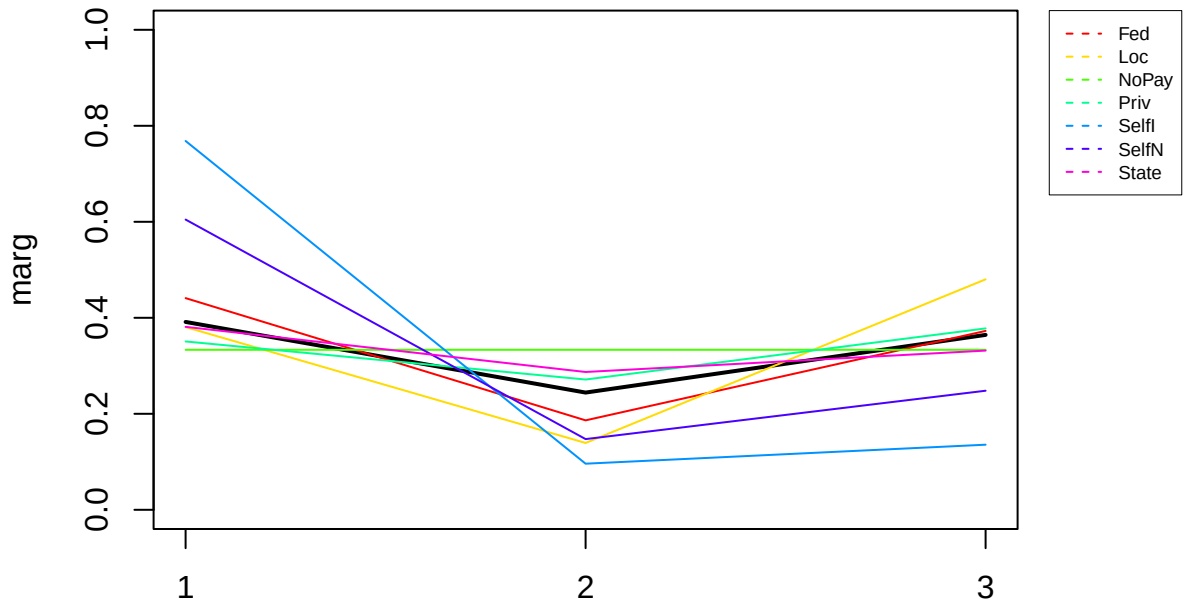
```
      #calcular els pvalues de les quali
    }
  }
}#endfor
```

```
## [1] "Anàlisi per classes de la Variable: age"
```



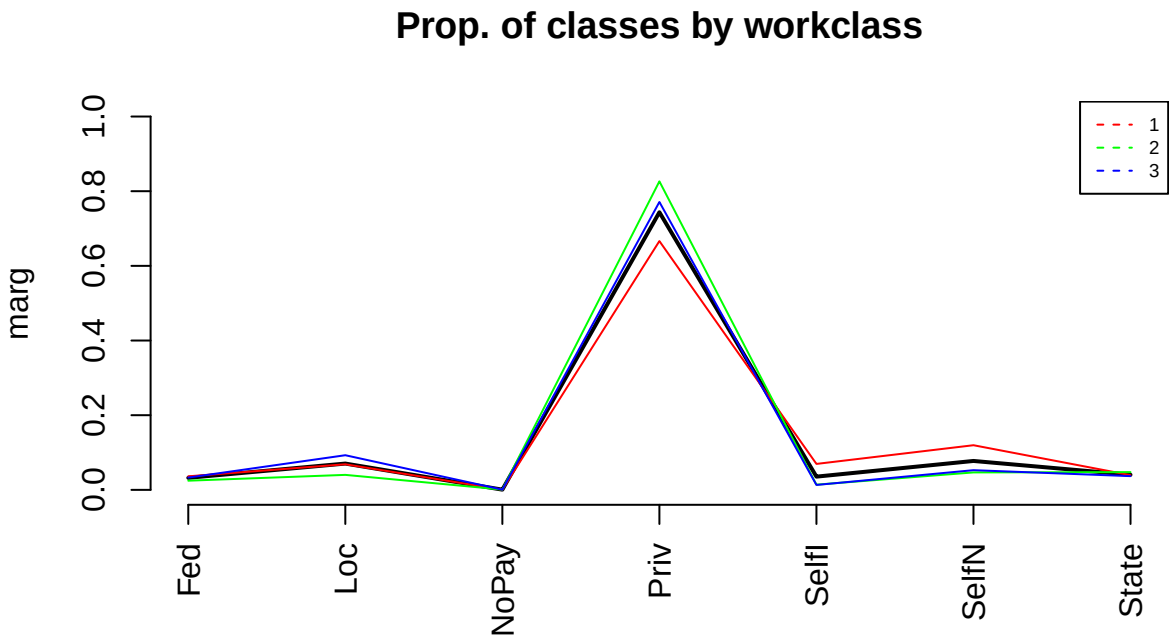

```
## [1] "Estadístics per groups:"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   18.00   35.00   43.00   44.18   52.00   90.00
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.00   22.00   27.00   30.64   36.00   90.00
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   17.00   25.00   36.00   37.62   47.00   90.00
## [1] "p-valueANOVA: 3.37964367791504e-190"
## [1] "p-value Kruskal-Wallis: 9.67398091244748e-189"
## [1] "p-values ValorsTest: "
## [1] 9.361532e-115  0.000000e+00  4.265215e-04
## [1] "Variable workclass"
## [1] "Categories=" "Fed"         "Loc"         "NoPay"       "Priv"
## [6] "SelfI"       "SelfN"       "State"
```

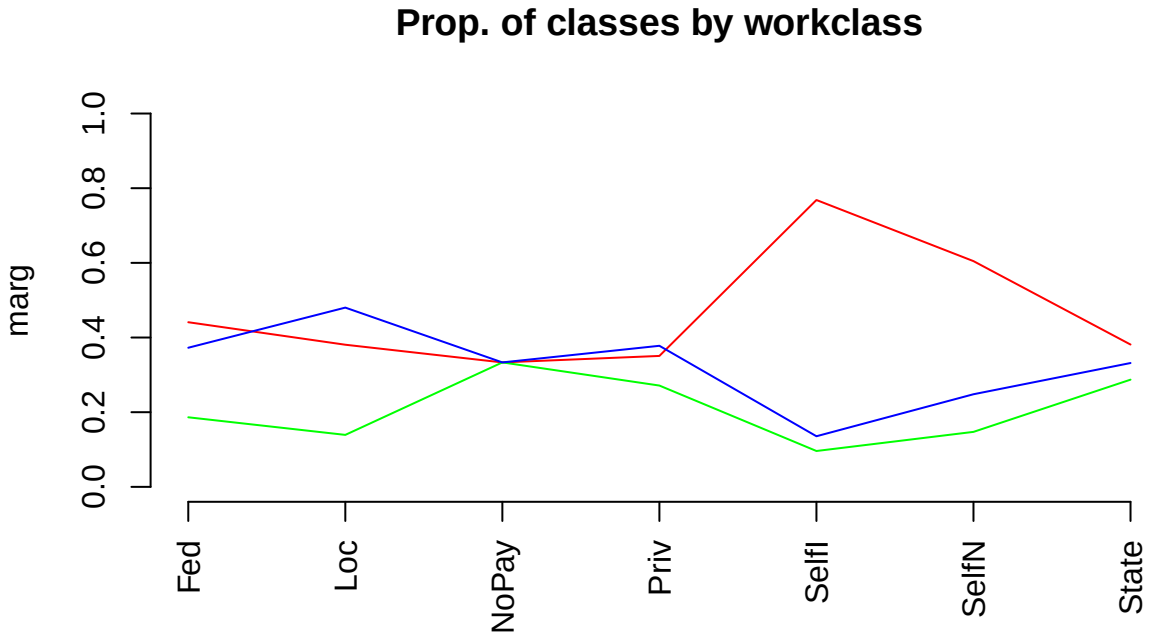

Prop. of classes by workclass
marg
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3

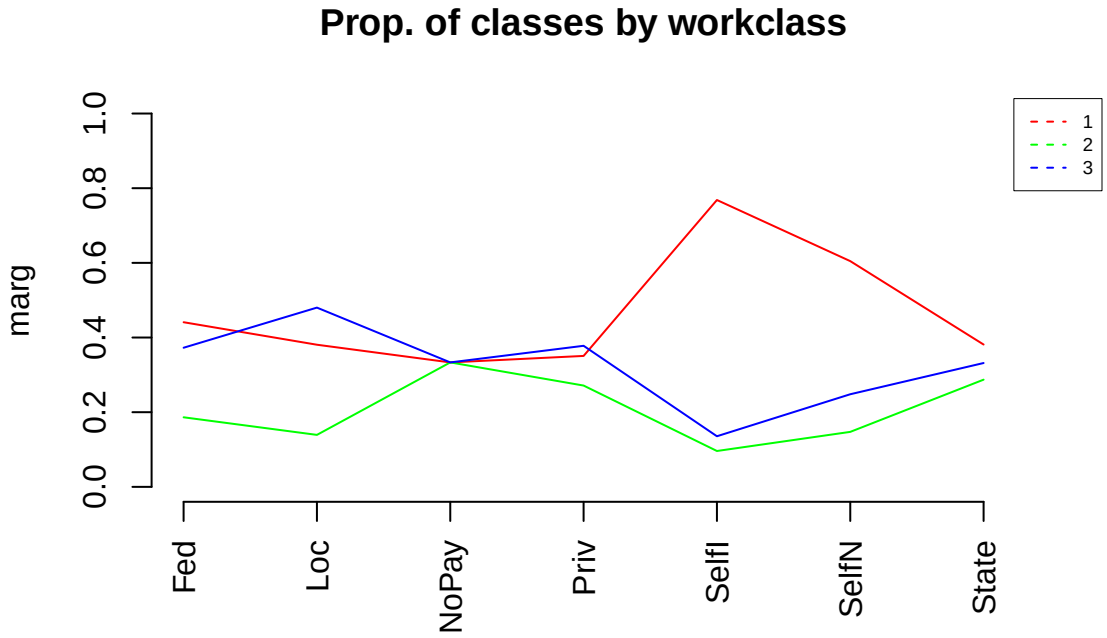



```
## [1] "Categories=" "Fed"         "Loc"         "NoPay"       "Priv"
## [6] "SelfI"       "SelfN"       "State"
```

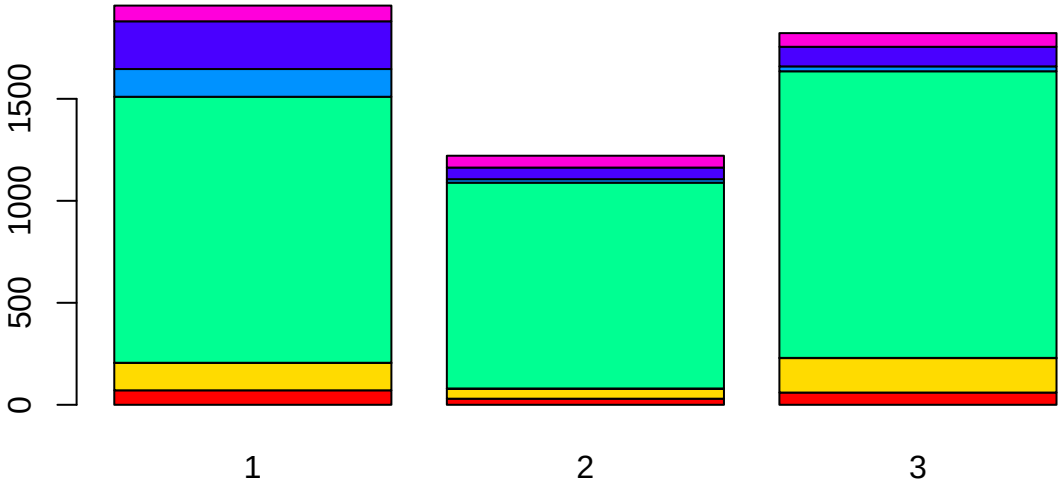

```
## [1] "Categories=" "Fed"         "Loc"         "NoPay"       "Priv"
## [6] "SelfI"       "SelfN"       "State"
```

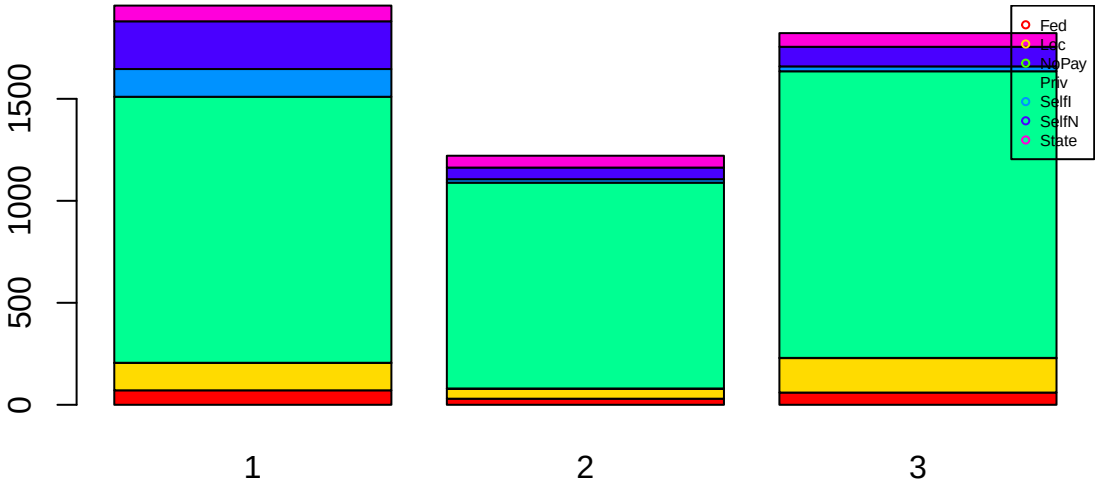

Prop. of classes by workclass
marg
0.0
0.2
0.4
0.6
0.8
1.0
Fed
Loc
NoPay
Priv
SelfI
SelfN
State
1
2
3

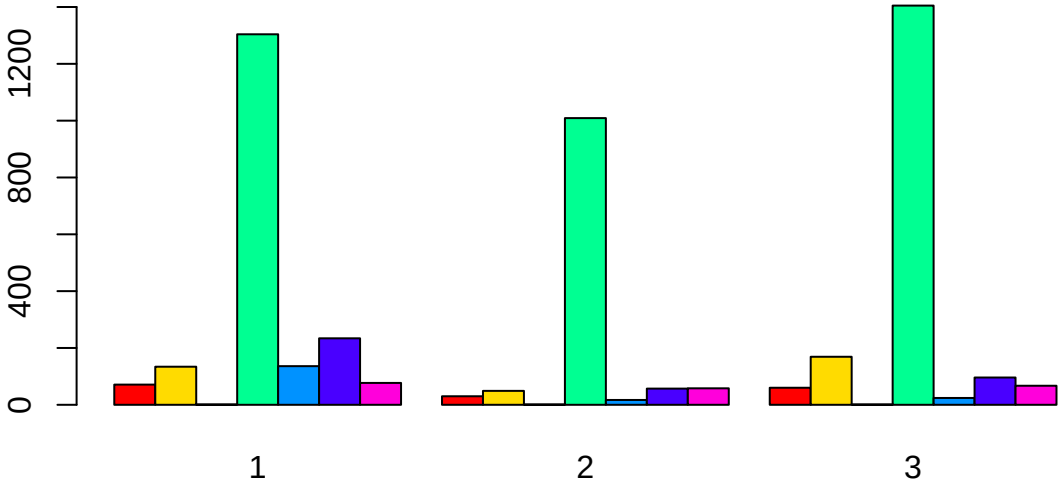

Prop. of classes by workclass
marg
0.0
0.2
0.4
0.6
0.8
1.0
Fed
Loc
NoPay
Priv
SelfI
SelfN
State

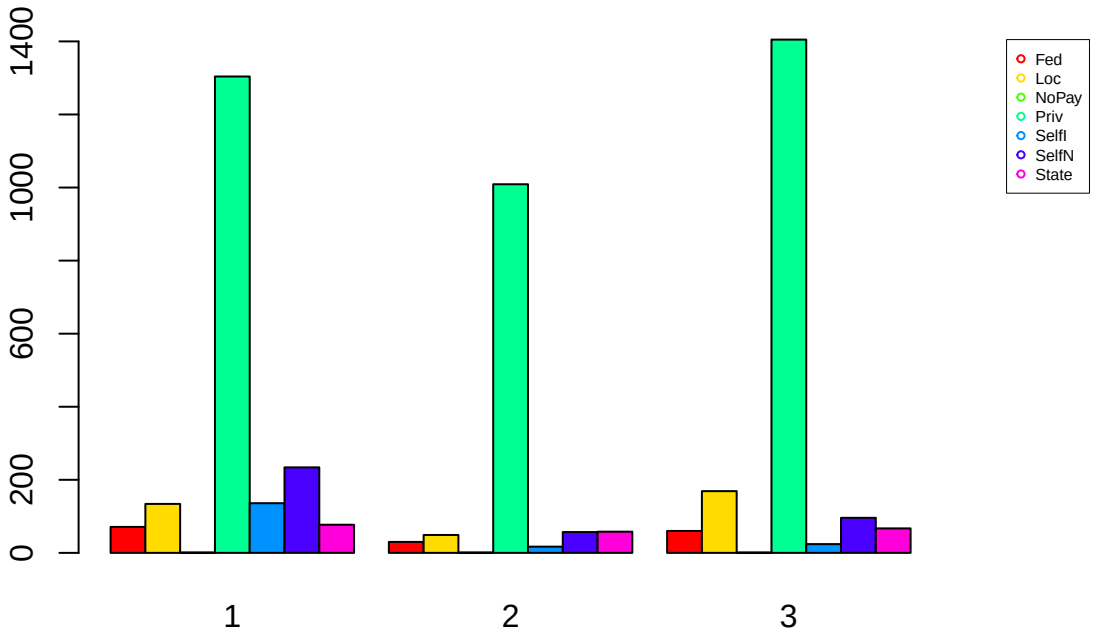



```
## [1] "Cross Table:"
##        P
##            1    2    3
##   Fed     71   30   60
##   Loc    134   49  169
##   NoPay    1    1    1
##   Priv  1304 1009 1405
##   SelfI  136   17   24
##   SelfN  234   57   96
##   State   77   58   67
## [1] "Distribucions condicionades a columnes:"
##
## P         Fed       Loc     NoPay      Priv     SelfI     SelfN     State
##   1 0.4409938 0.3806818 0.3333333 0.3507262 0.7683616 0.6046512 0.3811881
##   2 0.1863354 0.1392045 0.3333333 0.2713825 0.0960452 0.1472868 0.2871287
##   3 0.3726708 0.4801136 0.3333333 0.3778913 0.1355932 0.2480620 0.3316832
```


0
500
1000
1500
1
2
3

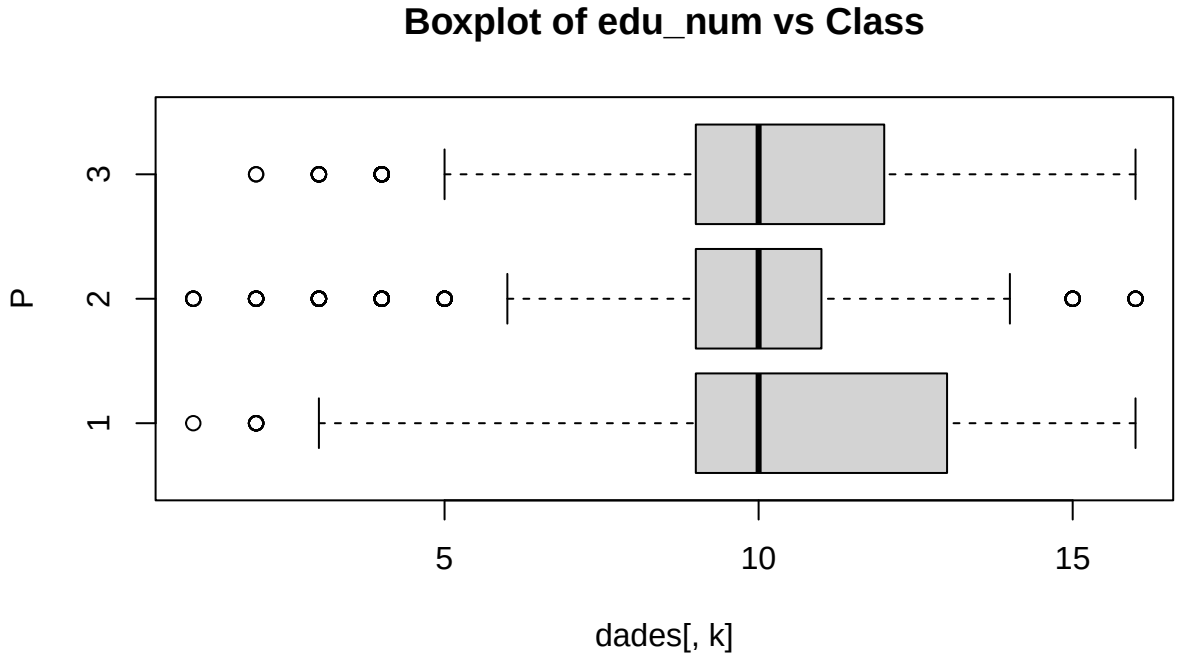

0
500
1000
1500
1
2
3
Fed
Loc
NoPay
Priv
SelfI
SelfN
State

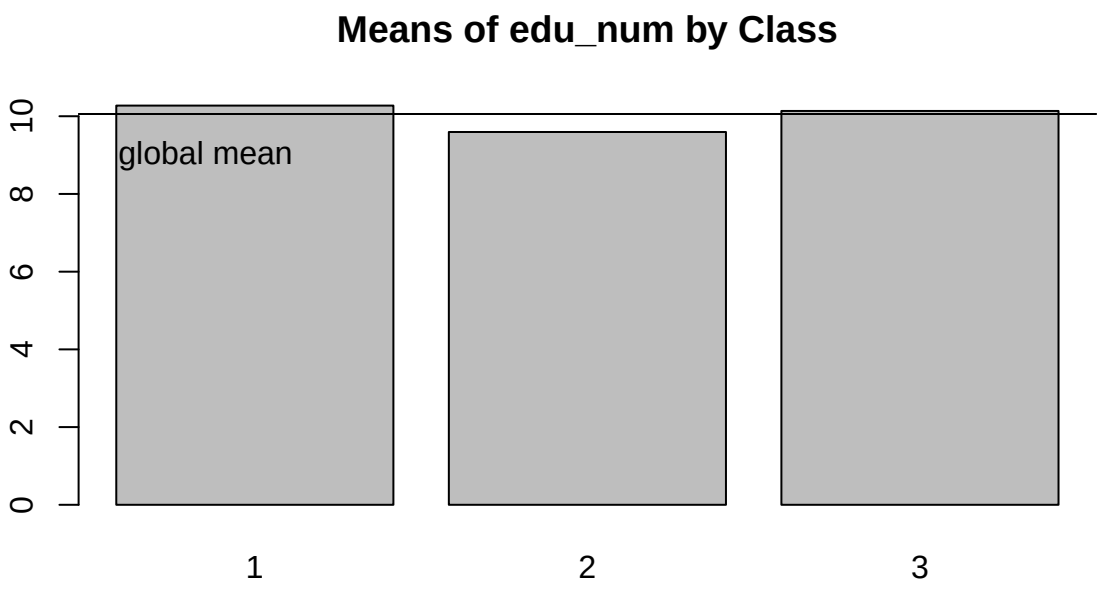

0
400
800
1200
1
2
3



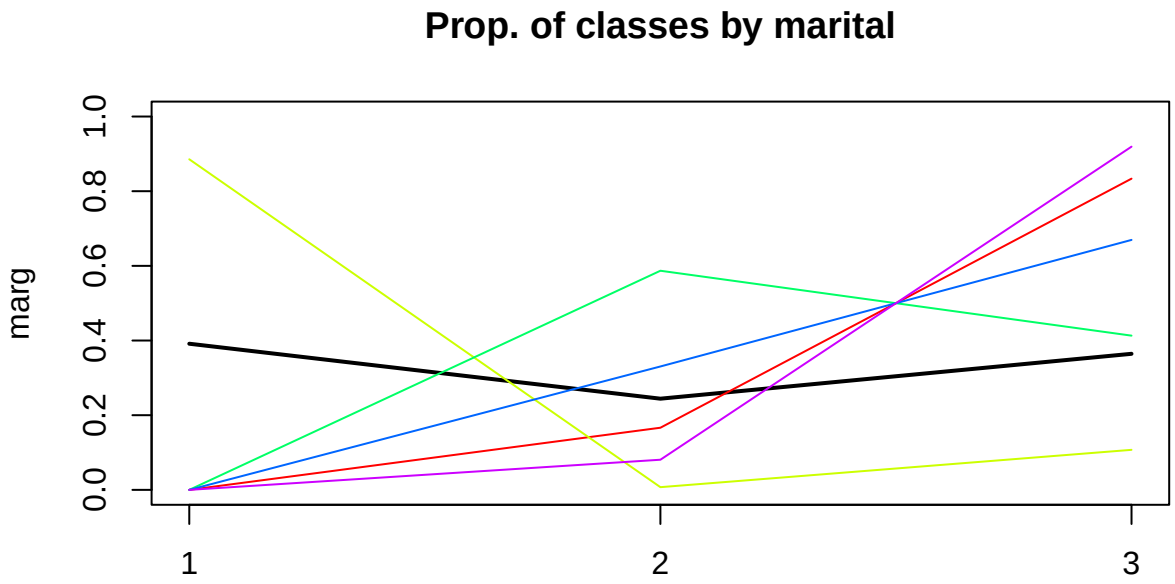

```
## [1] "Test Chi quadrat: "
## Warning in chisq.test(dades[, k], as.factor(P)): Chi-squared approximation may
## be incorrect
##
##  Pearson's Chi-squared test
##
## data:  dades[, k] and as.factor(P)
## X-squared = 243.13, df = 12, p-value < 2.2e-16
##
## [1] "valorsTest:"
## $rowpf
##    Xquali
## P            Fed          Loc        NoPay         Priv        SelfI
##   1 0.0362800204 0.0684721513 0.0005109862 0.6663260092 0.0694941237
##   2 0.0245700246 0.0401310401 0.0008190008 0.8263718264 0.0139230139
##   3 0.0329308452 0.0927552141 0.0005488474 0.7711306257 0.0131723381
##    Xquali
## P          SelfN        State
##   1 0.1195707716 0.0393459377
##   2 0.0466830467 0.0475020475
##   3 0.0526893524 0.0367727772
##
## $vtest
##    Xquali
## P           Fed         Loc       NoPay        Priv       SelfI       SelfN
```

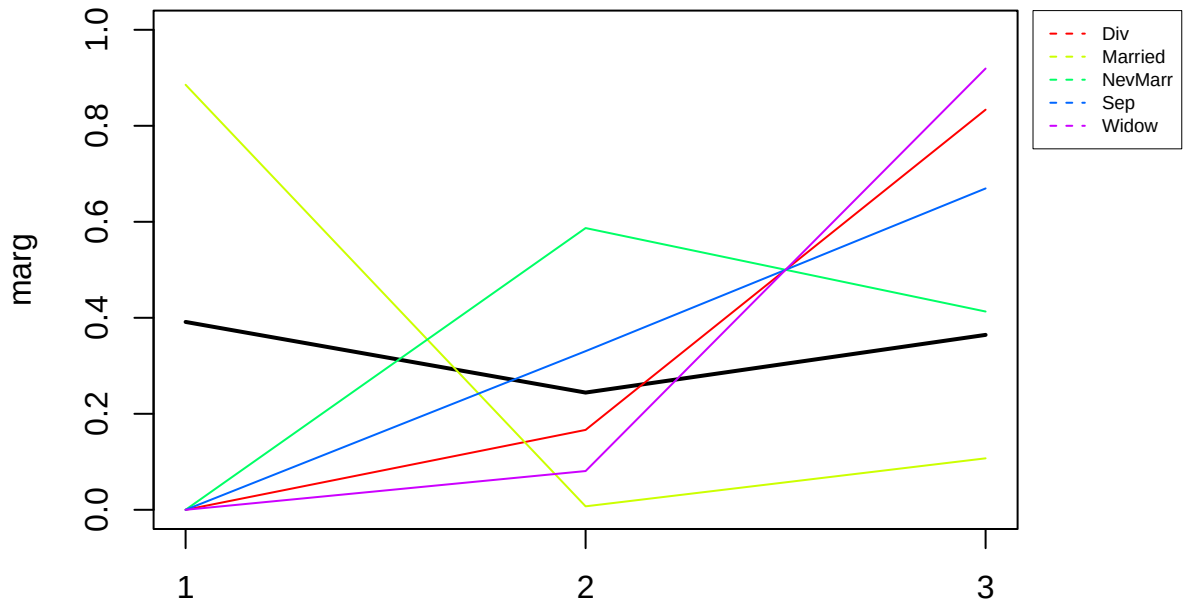
```
##   1   1.3106026  -0.4273346  -0.2061301 -10.0353860  10.4624576   8.9487790
##   2  -1.7372278  -4.7557334   0.3594635   7.6191799  -4.6714582  -4.6201420
##   3   0.2216598   4.6787160  -0.1118419   3.3757449  -6.4402209  -4.9509571
##    Xquali
## P         State
##   1  -0.3035708
##   2   1.4497802
##   3  -0.9863264
##
## $pval
##    Xquali
## P            Fed          Loc        NoPay         Priv        SelfI
##   1 9.499603e-02 3.345678e-01 4.183446e-01 0.000000e+00 6.423963e-26
##   2 4.117348e-02 9.886364e-07 3.596242e-01 1.276463e-14 1.495345e-06
##   3 4.122894e-01 1.443385e-06 4.554744e-01 3.680807e-04 5.964984e-11
##    Xquali
## P          SelfN        State
##   1 1.797181e-19 3.807275e-01
##   2 1.917388e-06 7.355991e-02
##   3 3.692469e-07 1.619865e-01
##
## [1] "Anàlisi per classes de la Variable: edu_num"
```



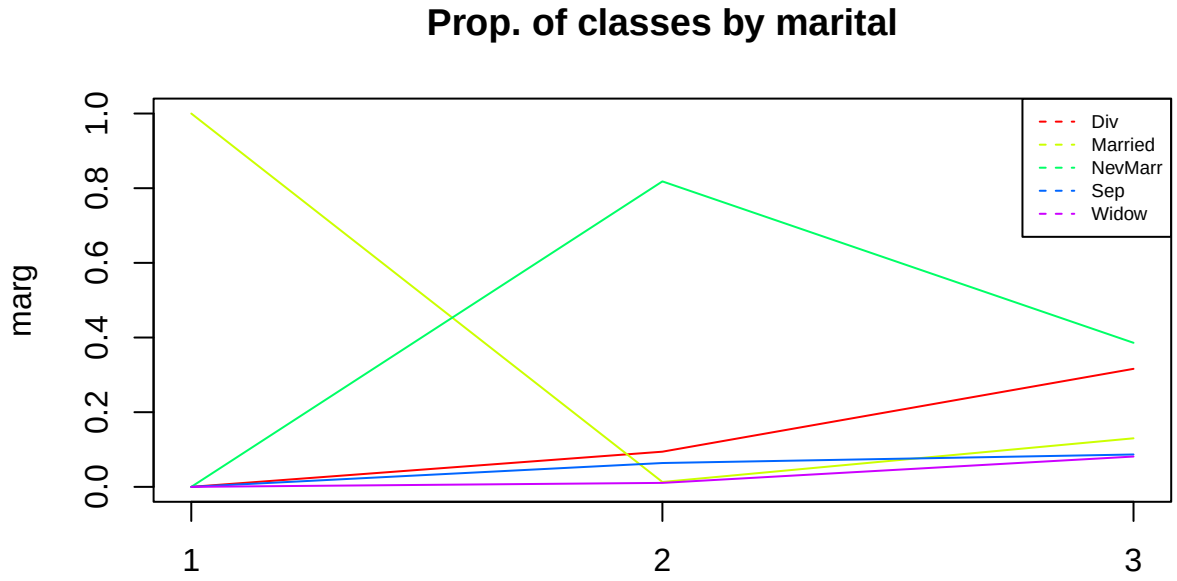



```
## [1] "Estadístics per groups:"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    1.00    9.00   10.00   10.27   13.00   16.00
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   1.000   9.000  10.000   9.595  11.000  16.000
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00    9.00   10.00   10.14   12.00   16.00
## [1] "p-valueANOVA: 1.231187841333e-11"
## [1] "p-value Kruskal-Wallis: 1.0090777949887e-08"
## [1] "p-values ValorsTest: "
## [1] 8.904730e-07 2.289280e-13 5.182494e-02
## [1] "Variable marital"
## [1] "Categories=" "Div"         "Married"     "NevMarr"     "Sep"
## [6] "Widow"
```

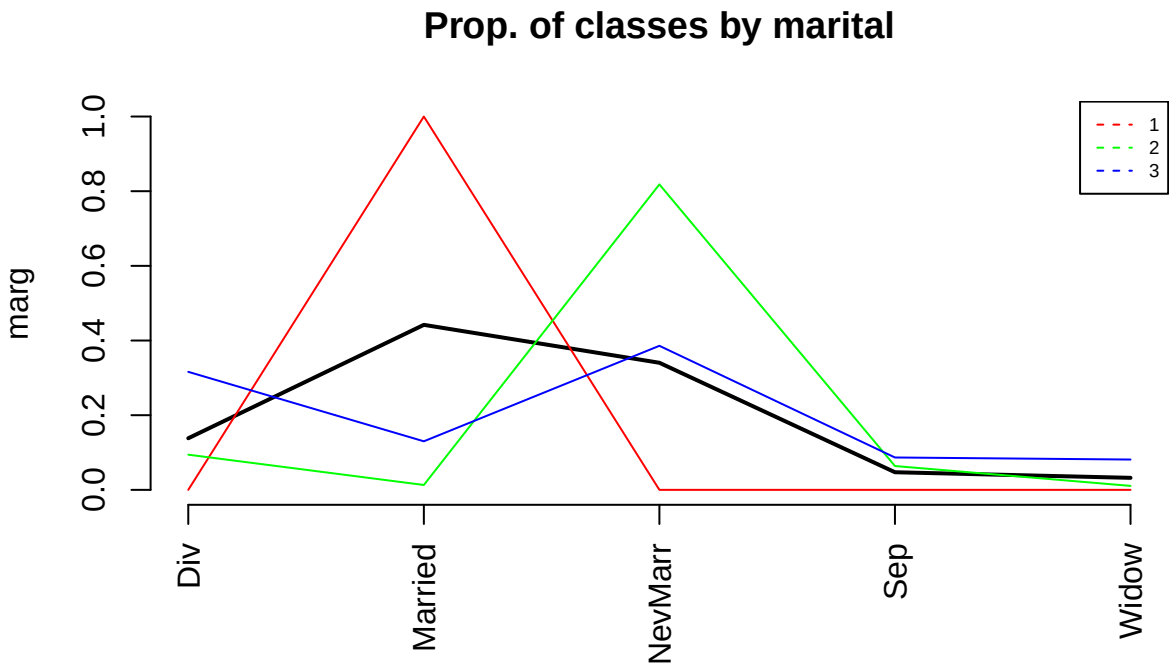

Prop. of classes by marital
marg
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3

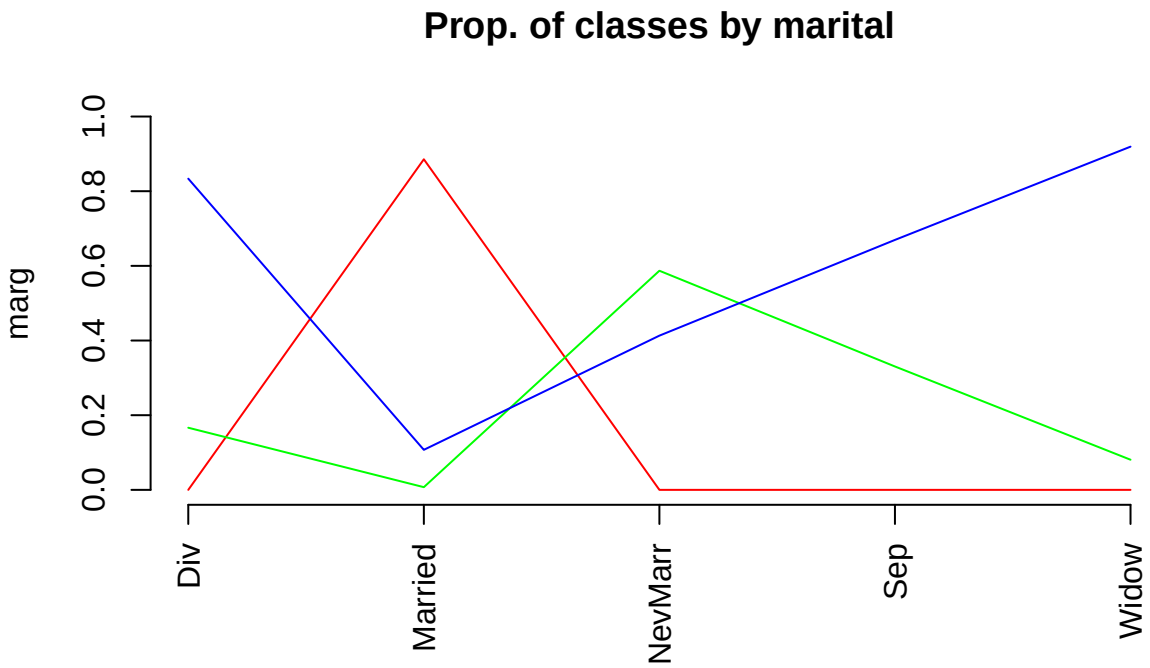


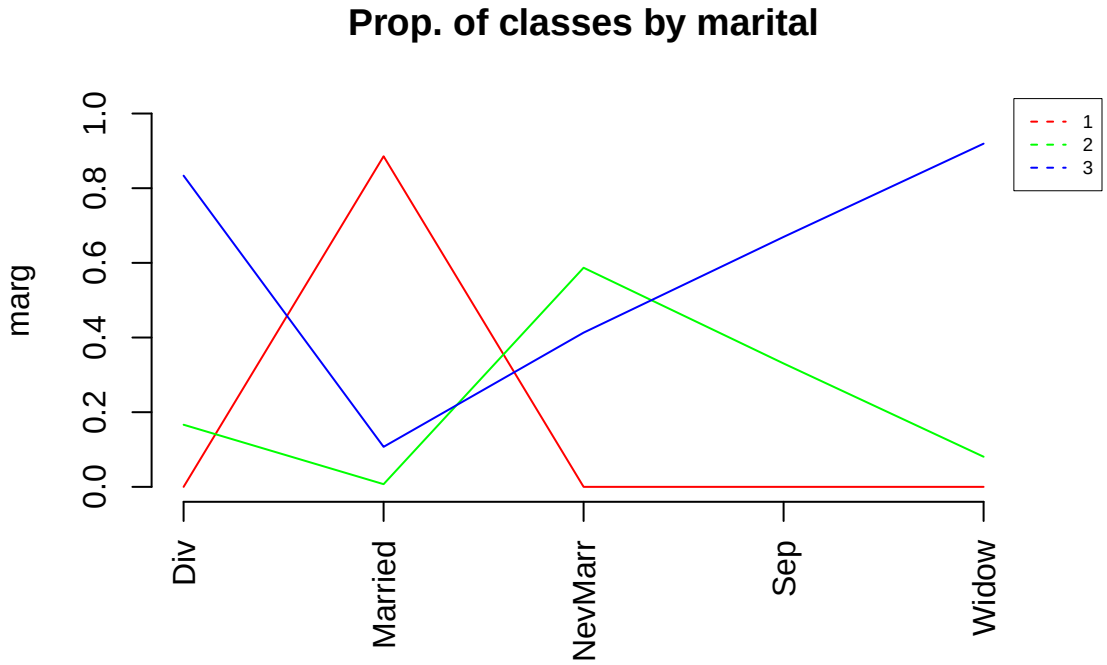

```
## [1] "Categories=" "Div"         "Married"     "NevMarr"     "Sep"
## [6] "Widow"
```



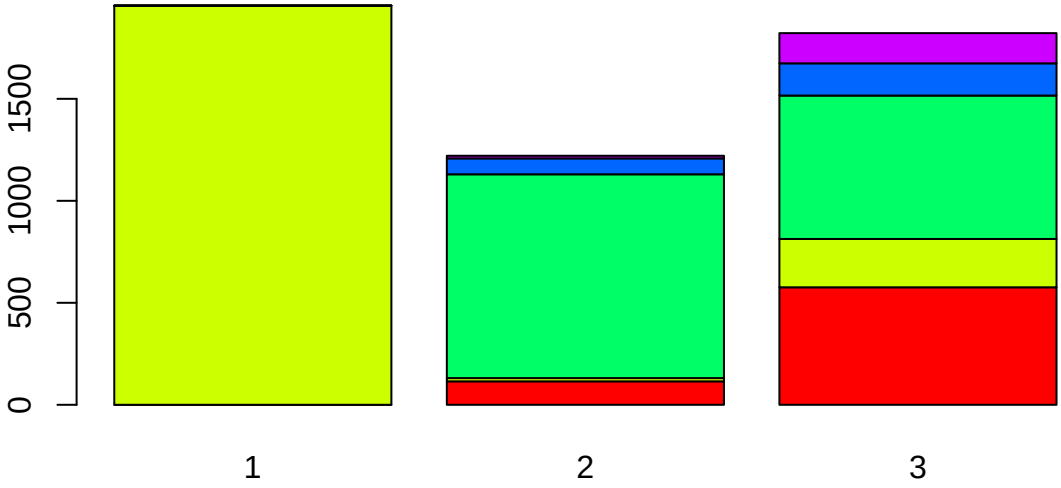

```
## [1] "Categories=" "Div"         "Married"     "NevMarr"     "Sep"
## [6] "Widow"
```

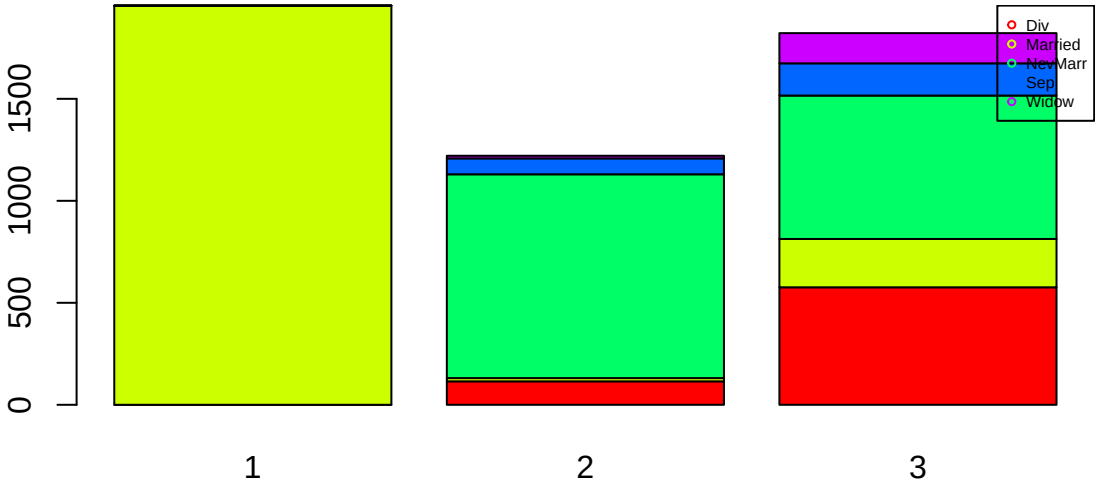

Prop. of classes by marital
marg
0.0
0.2
0.4
0.6
0.8
1.0
Div
Married
NevMarr
Sep
Widow
1
2
3

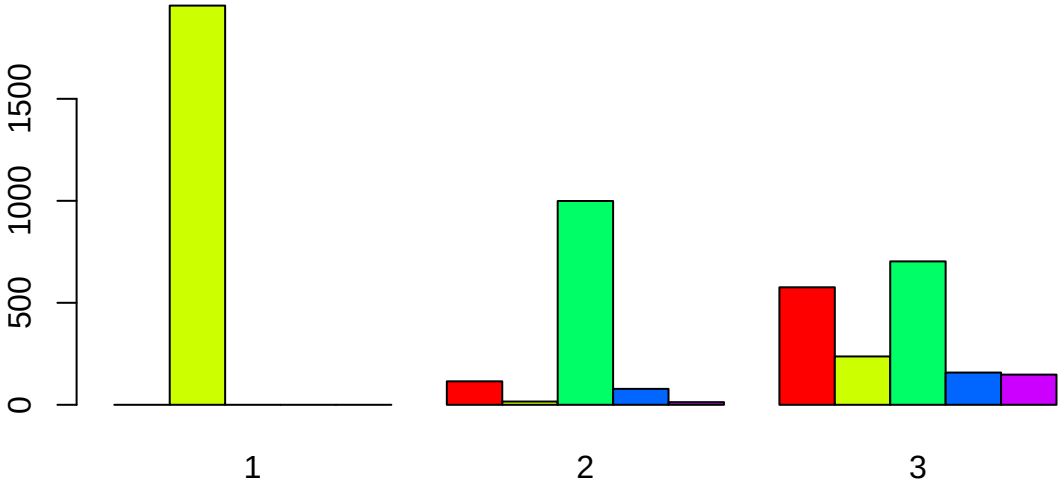

Prop. of classes by marital
marg
0.0
0.2
0.4
0.6
0.8
1.0
Div
Married
NevMarr
Sep
Widow

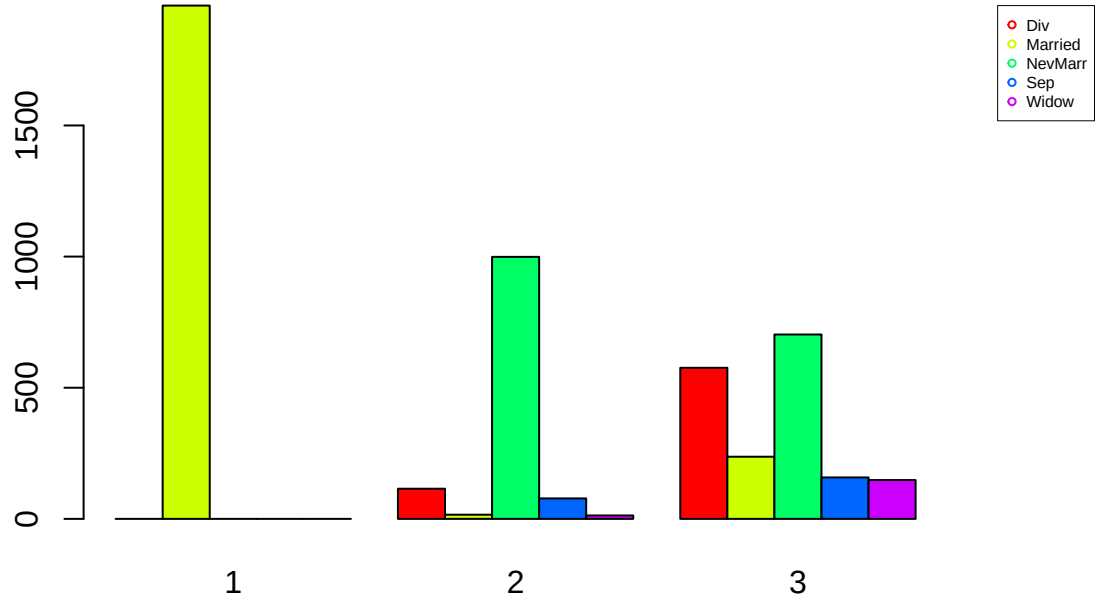



```
## [1] "Cross Table:"
##          P
##              1    2    3
##   Div        0  115  576
##   Married 1957   16  237
##   NevMarr    0  999  703
##   Sep        0   78  158
##   Widow      0   13  148
## [1] "Distribucions condicionades a columnes:"
##
## P           Div     Married     NevMarr         Sep       Widow
##   1 0.000000000 0.885520362 0.000000000 0.000000000 0.000000000
##   2 0.166425470 0.007239819 0.586956522 0.330508475 0.080745342
##   3 0.833574530 0.107239819 0.413043478 0.669491525 0.919254658
```


0
500
1000
1500
1
2
3

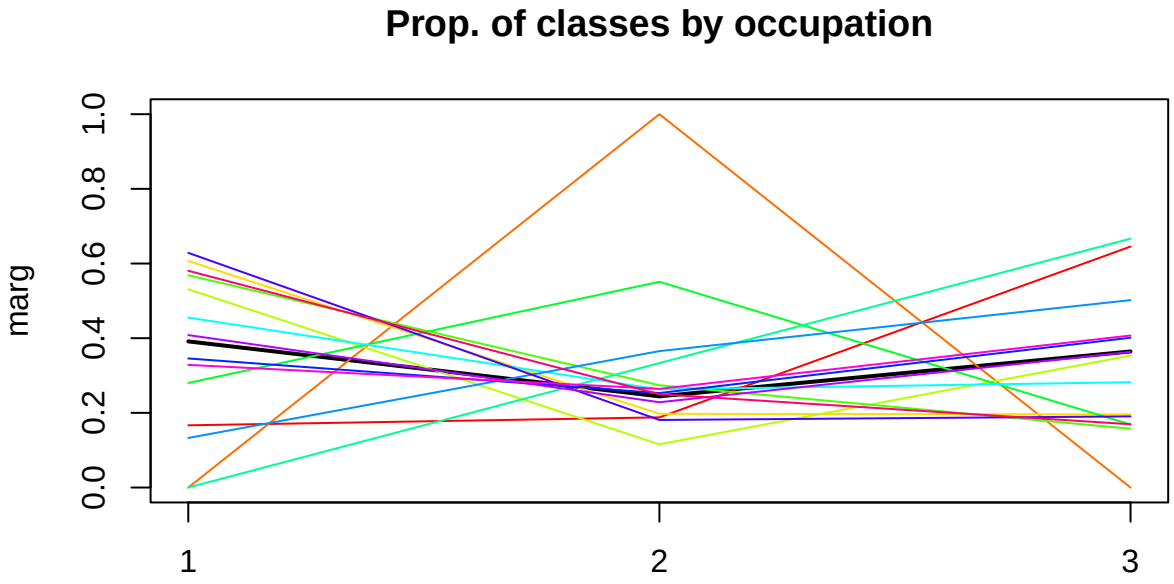

0
500
1000
1500
1
2
3
Div
Married
NewMarr
Sep
Widow

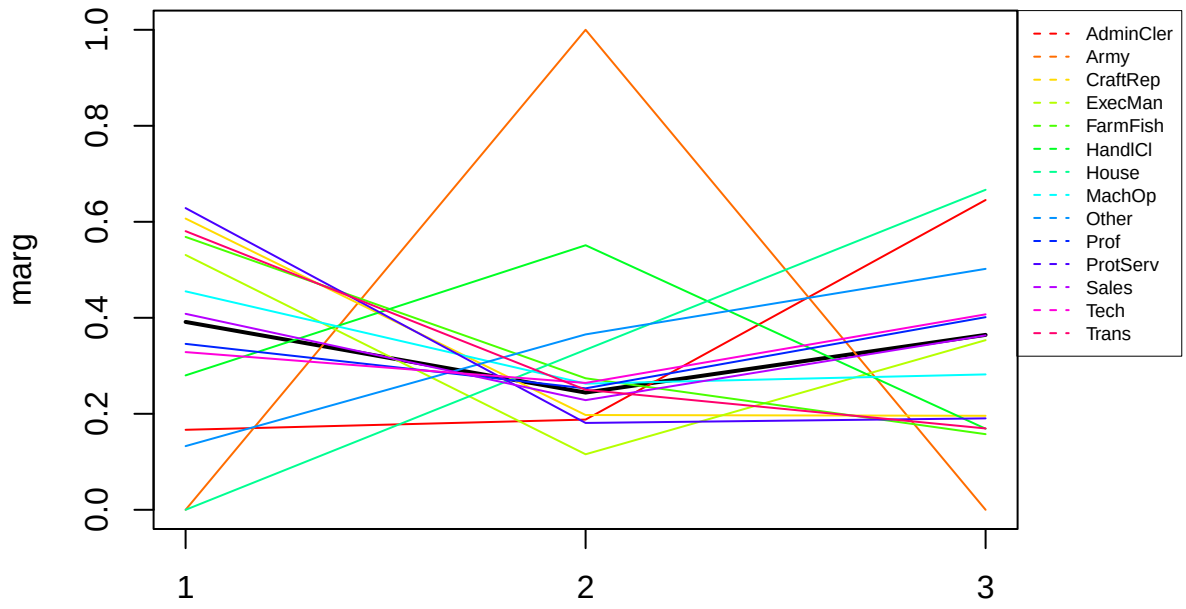

0
500
1000
1500
1
2
3



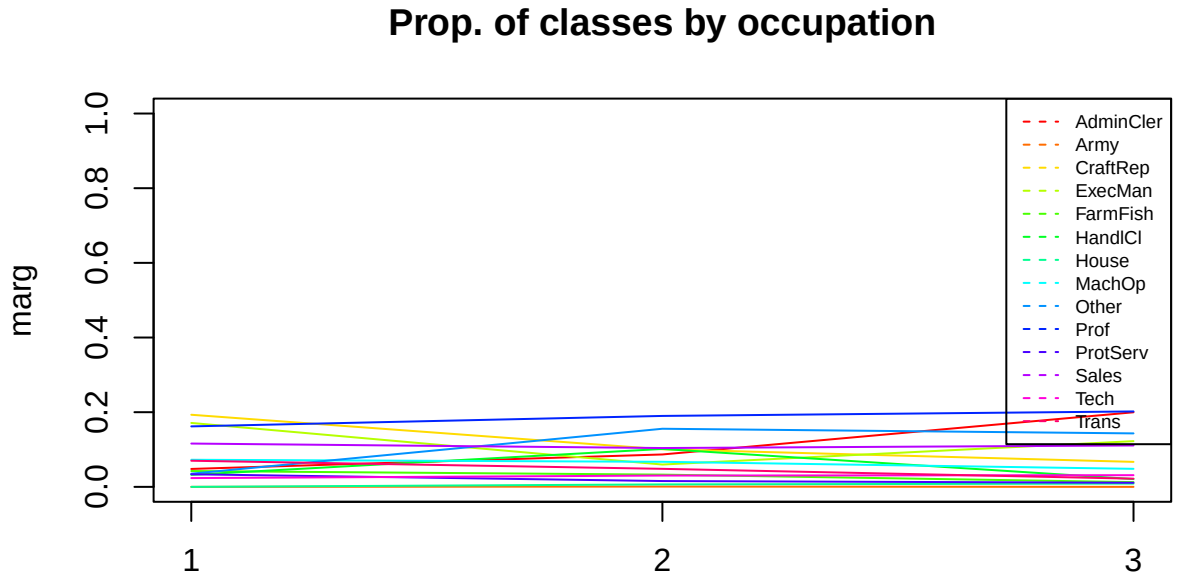

```
## [1] "Test Chi quadrat: "
##
##  Pearson's Chi-squared test
##
## data:  dades[, k] and as.factor(P)
## X-squared = 4865.3, df = 8, p-value < 2.2e-16
##
## [1] "valorsTest:"
## $rowpf
##    Xquali
## P          Div    Married    NevMarr        Sep      Widow
##   1 0.00000000 1.00000000 0.00000000 0.00000000 0.00000000
##   2 0.09418509 0.01310401 0.81818182 0.06388206 0.01064701
##   3 0.31613611 0.13007684 0.38583974 0.08671789 0.08122942
##
## $vtest
##    Xquali
## P          Div    Married    NevMarr        Sep      Widow
##   1 -22.708070  63.714078 -40.736517 -12.621168 -10.343430
##   2  -5.126216 -34.711880  40.527415   3.161785  -4.907284
##   3  27.605084 -33.628046   5.134320   9.977096  14.870249
##
## $pval
##    Xquali
## P             Div       Married       NevMarr           Sep         Widow
##   1  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00  0.000000e+00
```

```
##   2  1.478117e-07  0.000000e+00  0.000000e+00  7.840272e-04  4.617323e-07
##   3 4.834171e-168  0.000000e+00  1.415828e-07  9.600218e-24  2.571091e-50
##
## [1] "Variable occupation"
##  [1] "Categories=" "AdminCler"   "Army"        "CraftRep"    "ExecMan"
##  [6] "FarmFish"    "HandlCl"     "House"       "MachOp"      "Other"
## [11] "Prof"        "ProtServ"    "Sales"       "Tech"        "Trans"
```

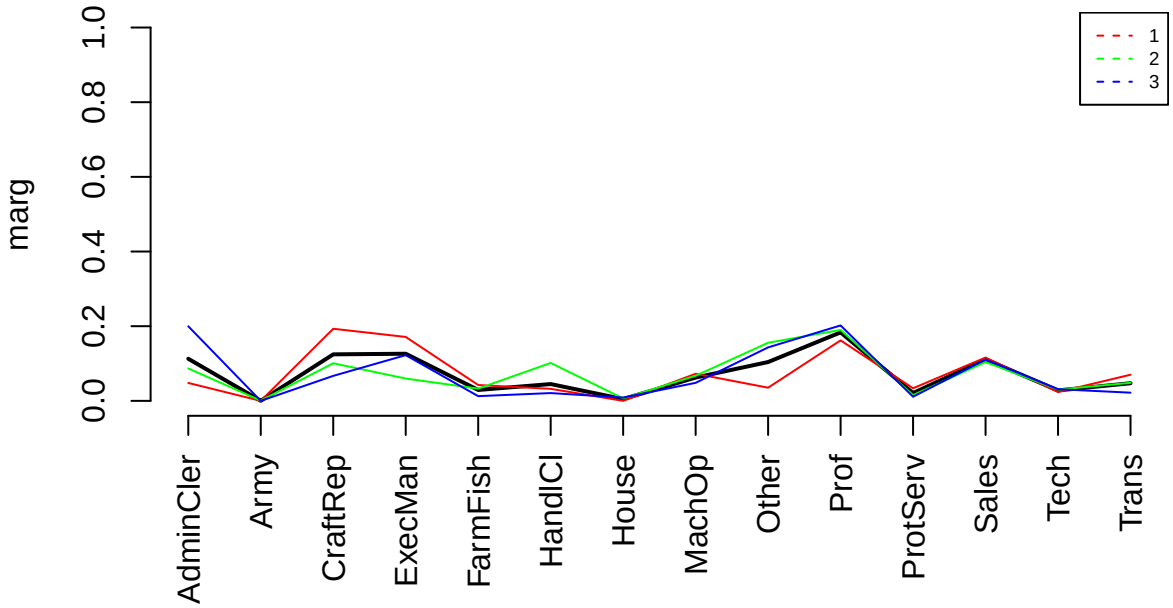


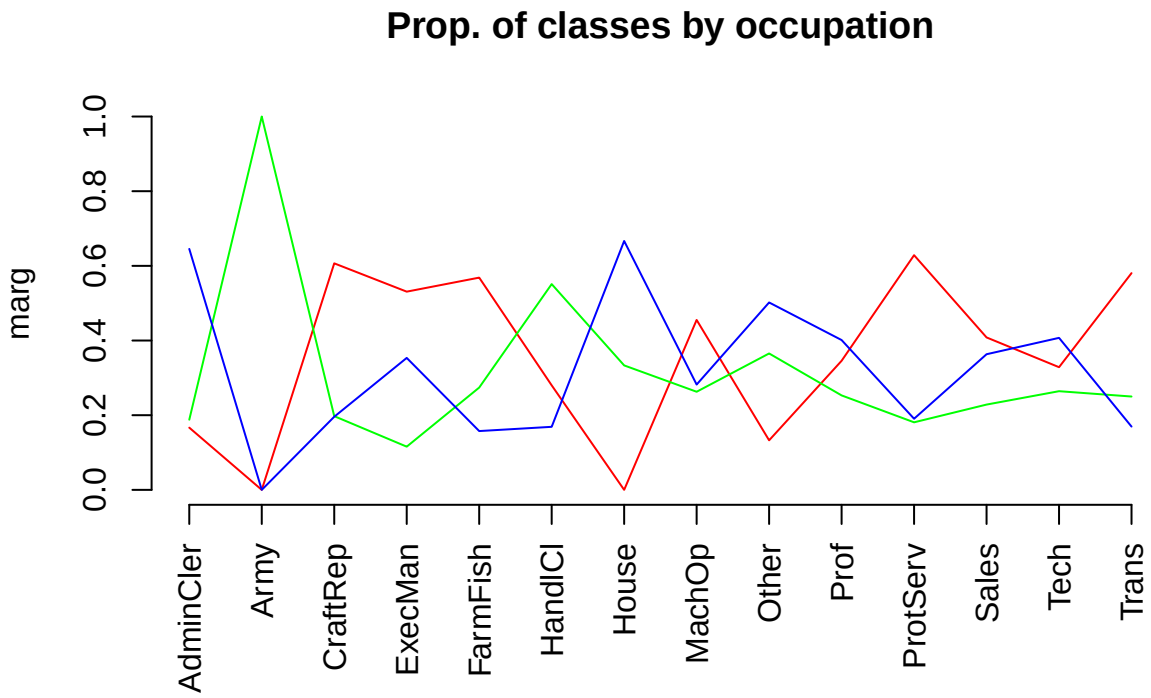


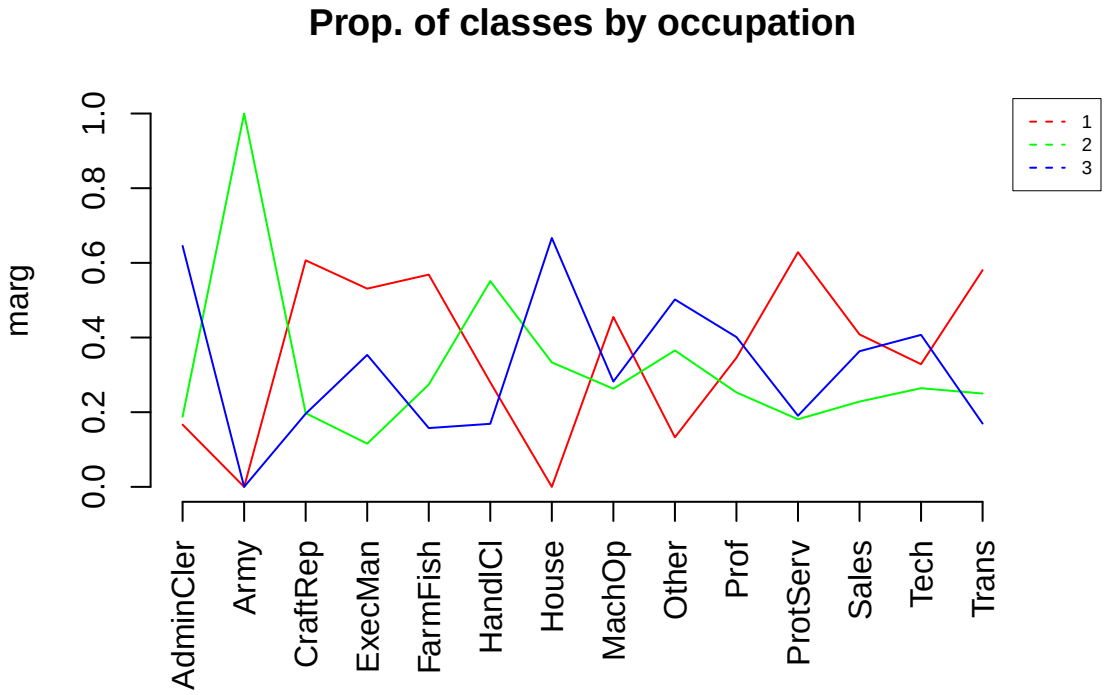

```
##  [1] "Categories=" "AdminCler"   "Army"       "CraftRep"   "ExecMan"
##  [6] "FarmFish"    "HandlCl"     "House"      "MachOp"     "Other"
## [11] "Prof"        "ProtServ"    "Sales"      "Tech"       "Trans"
```




```
##  [1] "Categories=" "AdminCler"   "Army"       "CraftRep"   "ExecMan"
##  [6] "FarmFish"    "HandlCl"     "House"      "MachOp"     "Other"
## [11] "Prof"        "ProtServ"    "Sales"      "Tech"       "Trans"
```

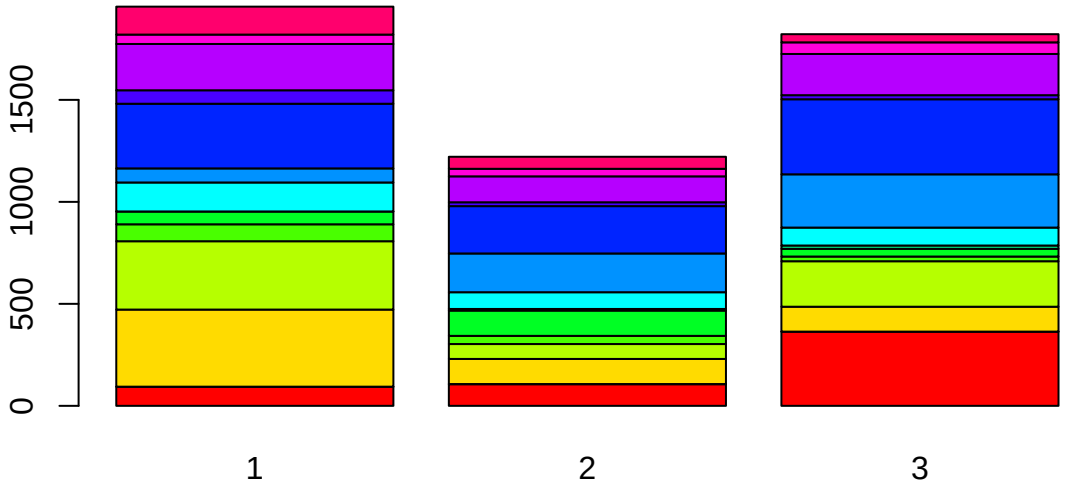

Prop. of classes by occupation
marg
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
AdminCler
Army
CraftRep
ExecMan
FarmFish
HandlCl
House
MachOp
Other
Prof
ProtServ
Sales
Tech
Trans

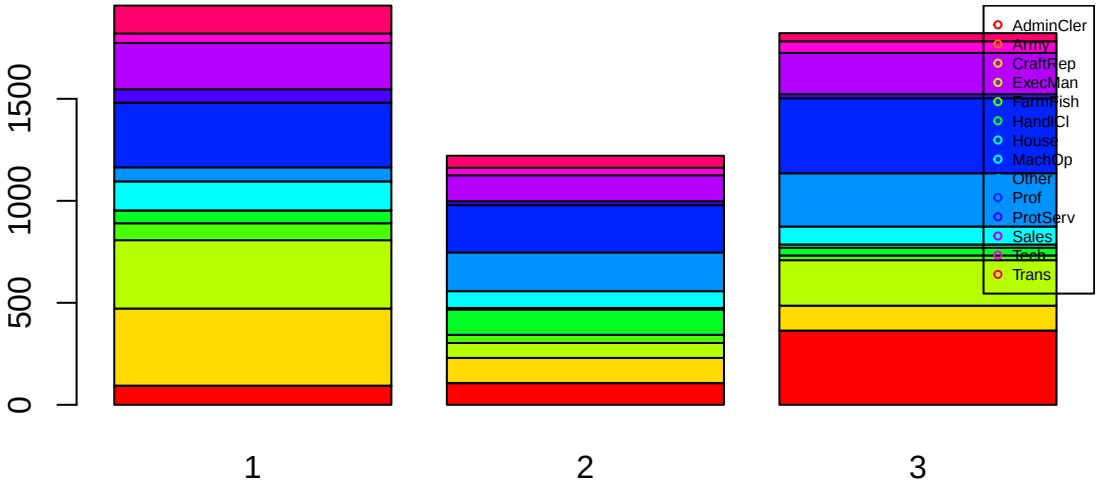

Prop. of classes by occupation
marg
0.0
0.2
0.4
0.6
0.8
1.0
AdminCler
Army
CraftRep
ExecMan
FarmFish
HandlCl
House
MachOp
Other
Prof
ProtServ
Sales
Tech
Trans

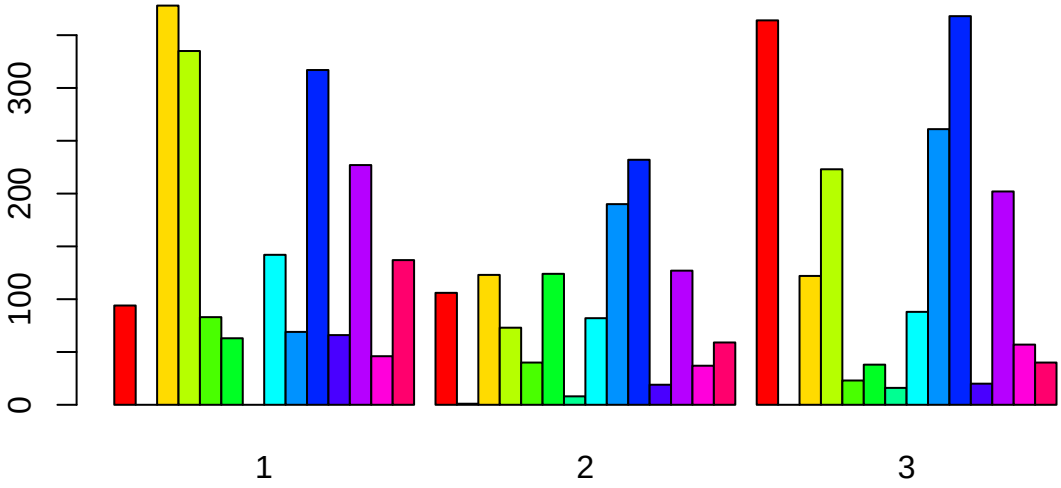


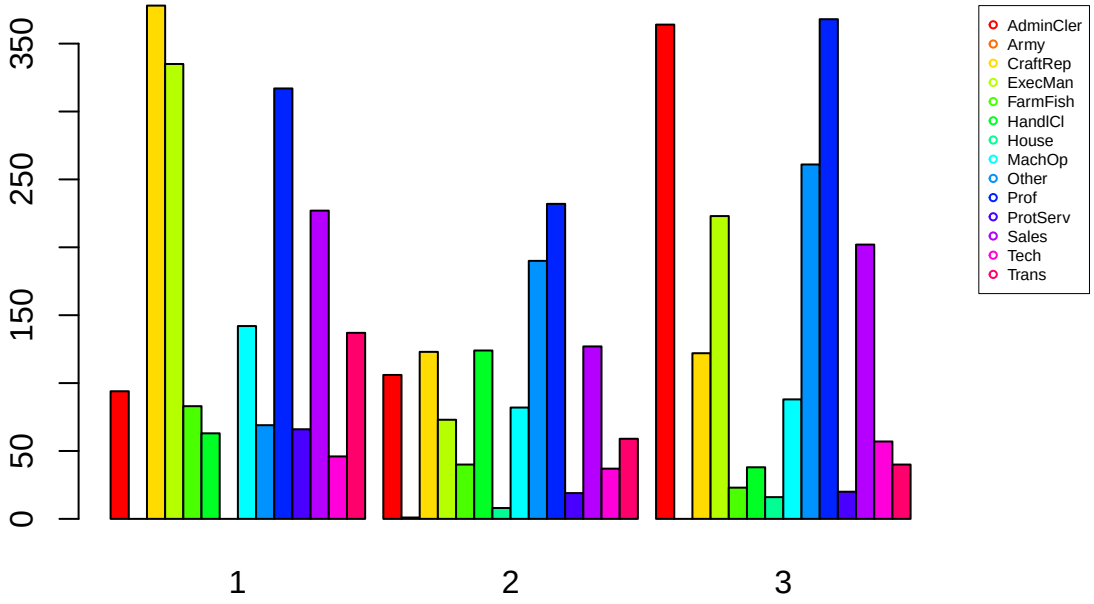

```
## [1] "Cross Table:"
##             P
##                1   2   3
##   AdminCler   94 106 364
##   Army         0   1   0
##   CraftRep   378 123 122
##   ExecMan    335  73 223
##   FarmFish    83  40  23
##   HandlCl     63 124  38
##   House        0   8  16
##   MachOp     142  82  88
##   Other       69 190 261
##   Prof       317 232 368
##   ProtServ    66  19  20
##   Sales      227 127 202
##   Tech        46  37  57
##   Trans      137  59  40
## [1] "Distribucions condicionades a columnes:"
##
## P   AdminCler      Army  CraftRep   ExecMan  FarmFish   HandlCl     House
##   1 0.1666667 0.0000000 0.6067416 0.5309033 0.5684932 0.2800000 0.0000000
##   2 0.1879433 1.0000000 0.1974318 0.1156894 0.2739726 0.5511111 0.3333333
##   3 0.6453901 0.0000000 0.1958266 0.3534073 0.1575342 0.1688889 0.6666667
##
## P      MachOp     Other      Prof  ProtServ     Sales      Tech     Trans
##   1 0.4551282 0.1326923 0.3456925 0.6285714 0.4082734 0.3285714 0.5805085
```

```
##    2 0.2628205 0.3653846 0.2529989 0.1809524 0.2284173 0.2642857 0.2500000
##    3 0.2820513 0.5019231 0.4013086 0.1904762 0.3633094 0.4071429 0.1694915
```



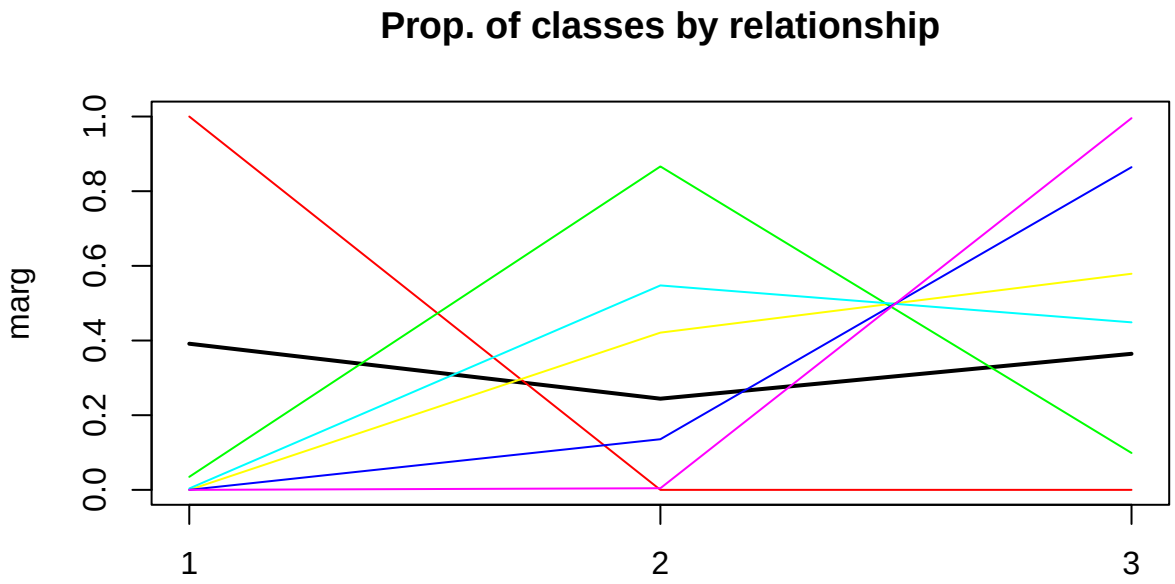

0
500
1000
1500
1
2
3
AdminCler
Army
CraftRep
ExecMan
FarmFish
HandCl
House
MachOp
Other
Prof
ProtServ
Sales
Tech
Trans

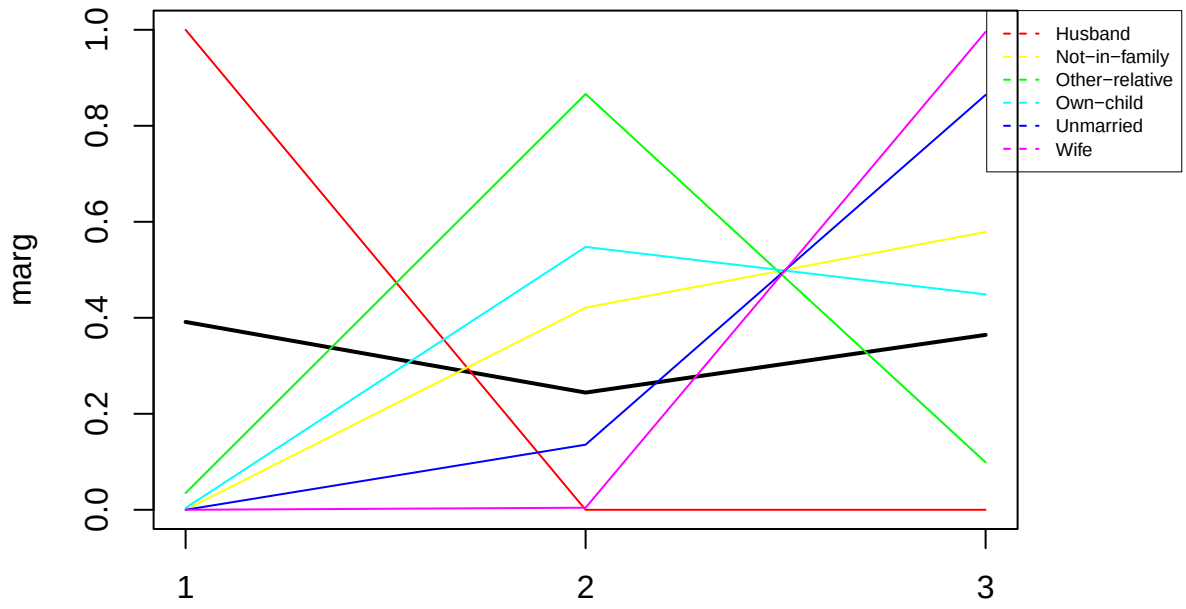

0
100
200
300
1
2
3



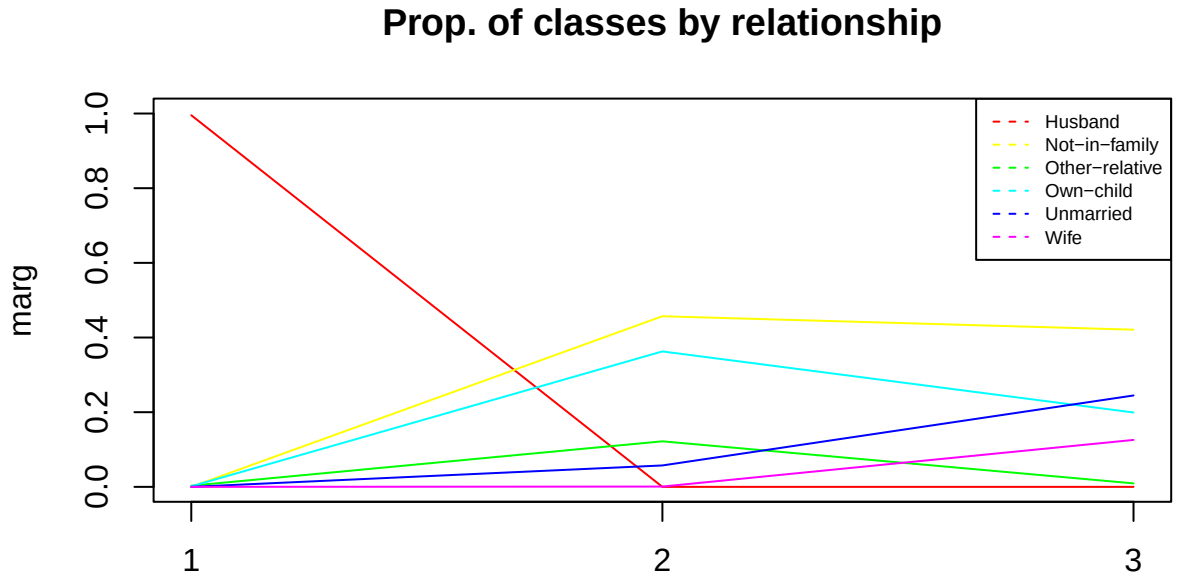

```
## [1] "Test Chi quadrat: "
## Warning in chisq.test(dades[, k], as.factor(P)): Chi-squared approximation may
## be incorrect
##
##  Pearson's Chi-squared test
##
## data:  dades[, k] and as.factor(P)
## X-squared = 811.03, df = 26, p-value < 2.2e-16
##
## [1] "valorsTest:"
## $rowpf
##    Xquali
## P       AdminCler          Army     CraftRep      ExecMan     FarmFish
##   1 0.0480327031 0.0000000000 0.1931527849 0.1711803781 0.0424118549
##   2 0.0868140868 0.0008190008 0.1007371007 0.0597870598 0.0327600328
##   3 0.1997804610 0.0000000000 0.0669593853 0.1223929748 0.0126234907
##    Xquali
## P         HandlCl        House       MachOp        Other         Prof
##   1 0.0321921308 0.0000000000 0.0725600409 0.0352580480 0.1619826265
##   2 0.1015561016 0.0065520066 0.0671580672 0.1556101556 0.1900081900
##   3 0.0208562020 0.0087815587 0.0482985730 0.1432491767 0.2019758507
##    Xquali
## P        ProtServ        Sales         Tech        Trans
##   1 0.0337250894 0.1159938682 0.0235053654 0.0700051099
##   2 0.0155610156 0.1040131040 0.0303030303 0.0483210483
```

```
##   3 0.0109769484 0.1108671789 0.0312843030 0.0219538968
##
## $vtest
##    Xquali
## P      AdminCler         Army     CraftRep      ExecMan     FarmFish
##   1 -11.60967051  -0.80202493  11.77042783   7.68098370   4.44976345
##   2  -3.30161976   1.75943833  -2.90412499  -8.03843303   0.84986917
##   3  14.72104468  -0.75725243  -9.34433831  -0.61380766  -5.27131555
##    Xquali
## P        HandlCl        House       MachOp        Other         Prof
##   1  -3.50346981  -3.93817376   2.38190151 -12.76969794  -3.13827716
##   2  10.96543617   1.01885953   0.79064925   6.79546034   0.68632933
##   3  -6.23562355   3.08432096  -3.12136252   6.88401993   2.56996140
##    Xquali
## P       ProtServ        Sales         Tech        Trans
##   1   5.03256379   0.86469134  -1.54494024   6.09803249
##   2  -1.52465370  -0.91884304   0.56110210   0.21247452
##   3  -3.74266723  -0.05668115   1.06589225  -6.37389231
##
## $pval
##    Xquali
## P      AdminCler         Army     CraftRep      ExecMan     FarmFish
##   1 0.000000e+00 2.112693e-01 2.772131e-32 7.893579e-15 4.298246e-06
##   2 4.806414e-04 3.925154e-02 1.841406e-03 4.440892e-16 1.976989e-01
##   3 2.361536e-49 2.244493e-01 0.000000e+00 2.696712e-01 6.772468e-08
##    Xquali
## P        HandlCl        House       MachOp        Other         Prof
##   1 2.296194e-04 4.105207e-05 8.611752e-03 0.000000e+00 8.497205e-04
##   2 2.801487e-28 1.541348e-01 2.145744e-01 5.398361e-12 2.462527e-01
##   3 2.249910e-10 1.020087e-03 9.000814e-04 2.909336e-12 5.085492e-03
##    Xquali
## P       ProtServ        Sales         Tech        Trans
##   1 2.419817e-07 1.936041e-01 6.118036e-02 5.369096e-10
##   2 6.367275e-02 1.790888e-01 2.873640e-01 4.158684e-01
##   3 9.103864e-05 4.773996e-01 1.432362e-01 9.214496e-11
##
## [1] "Variable relationship"
## [1] "Categories="    "Husband"        "Not-in-family"  "Other-relative"
## [5] "Own-child"      "Unmarried"      "Wife"
```

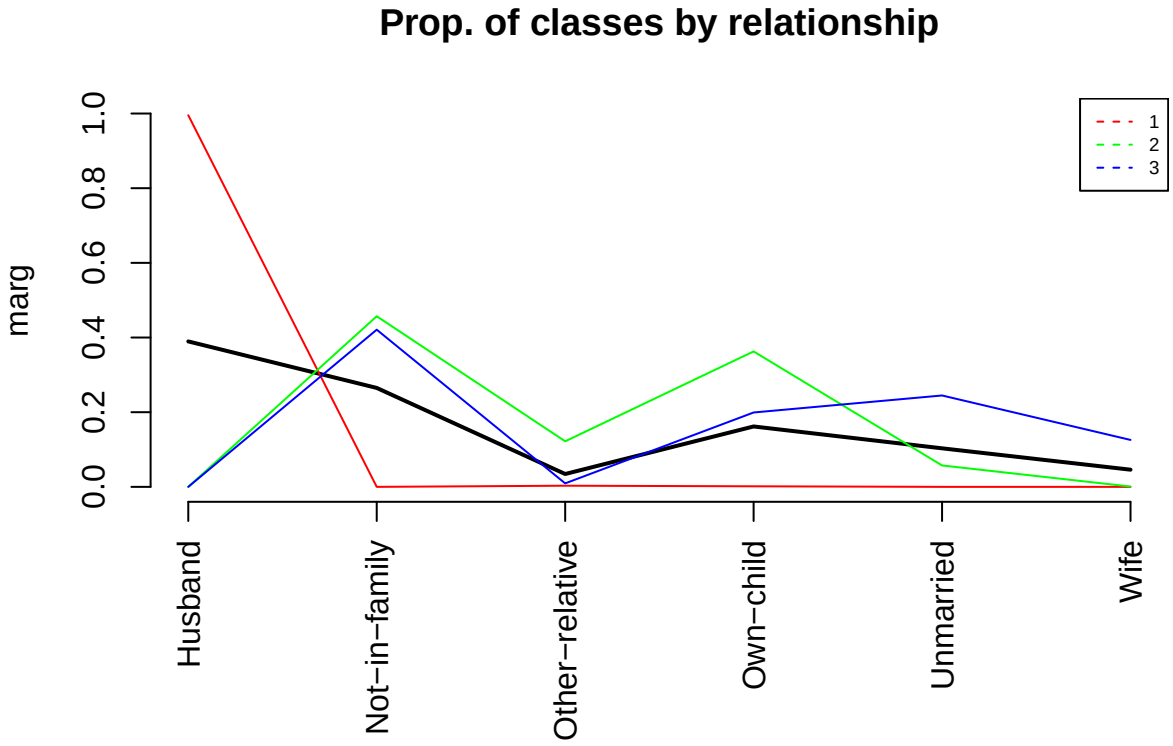

Prop. of classes by relationship
marg
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3

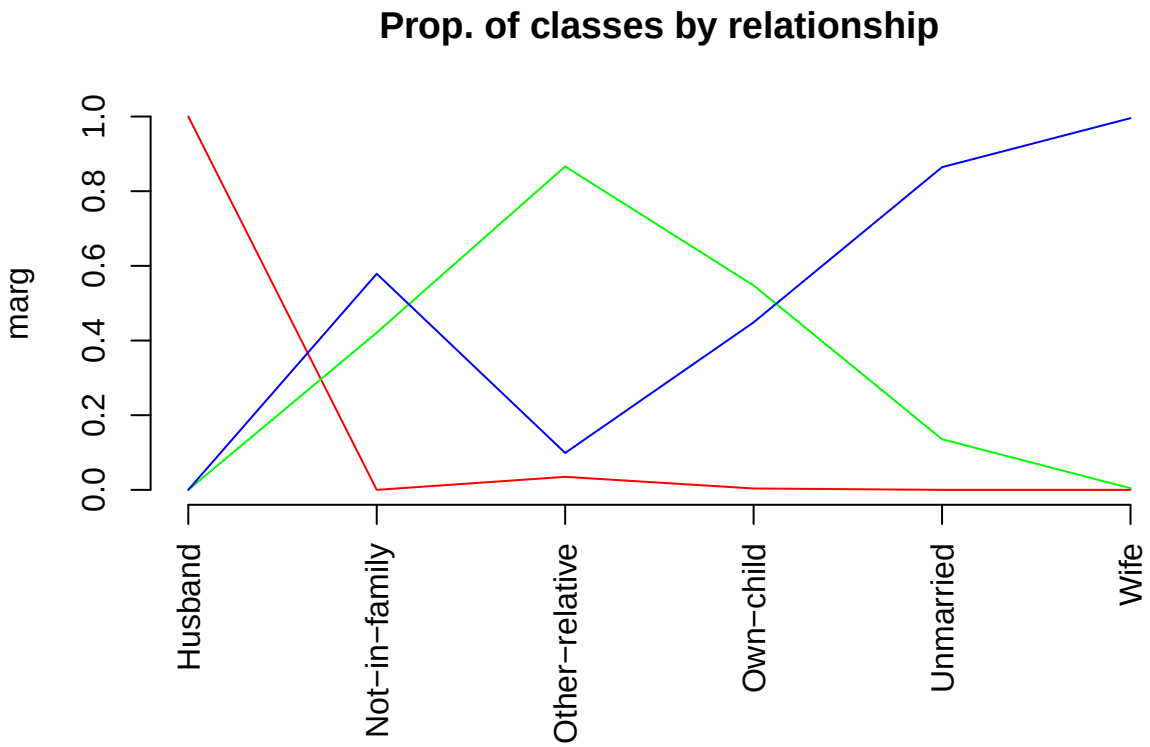


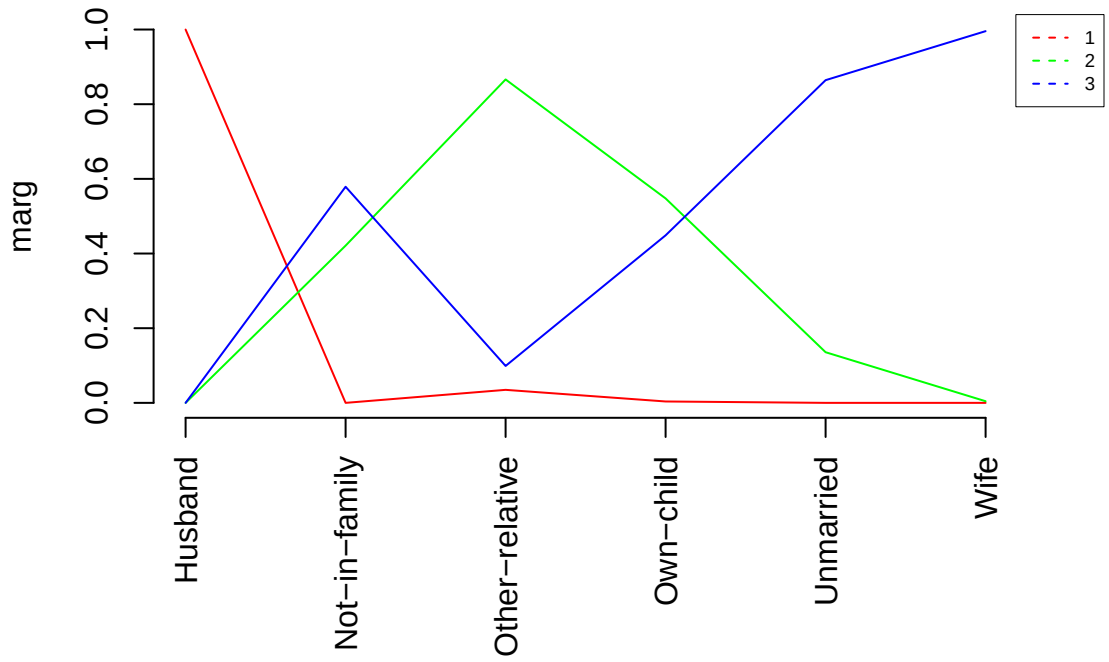

```
## [1] "Categories="    "Husband"        "Not-in-family"  "Other-relative"
## [5] "Own-child"      "Unmarried"      "Wife"
```



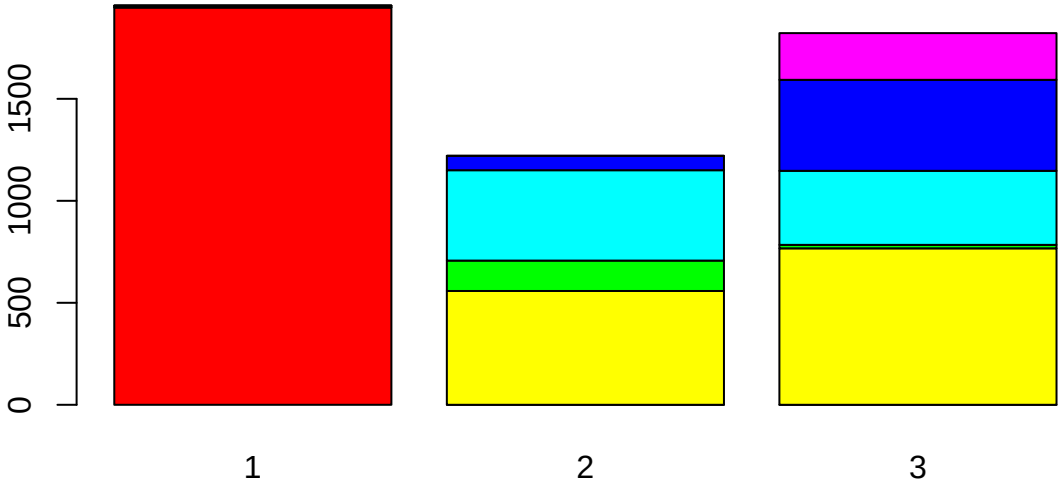

```
## [1] "Categories="    "Husband"        "Not-in-family"  "Other-relative"
## [5] "Own-child"      "Unmarried"      "Wife"
```

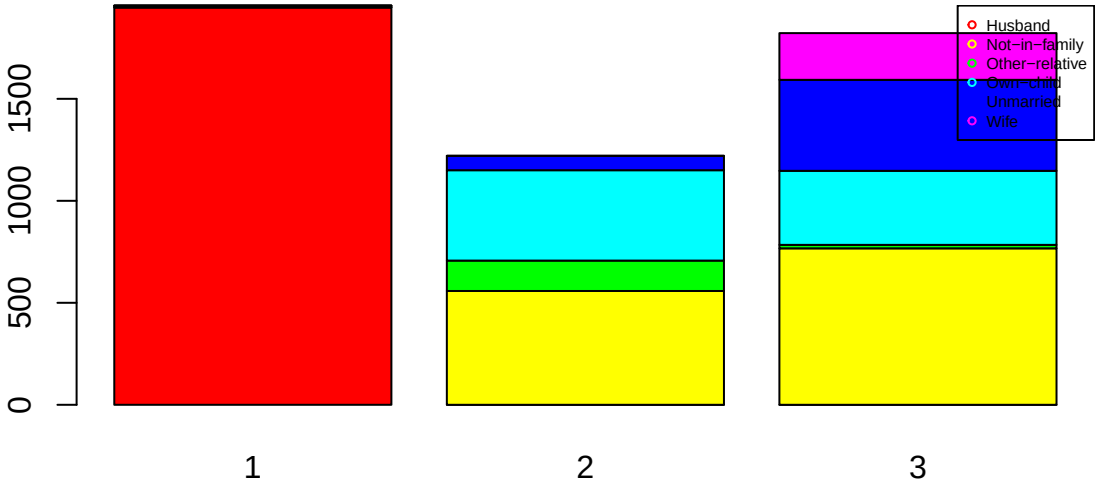

Prop. of classes by relationship
marg
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
Husband
Not−in−family
Other−relative
Own−child
Unmarried
Wife

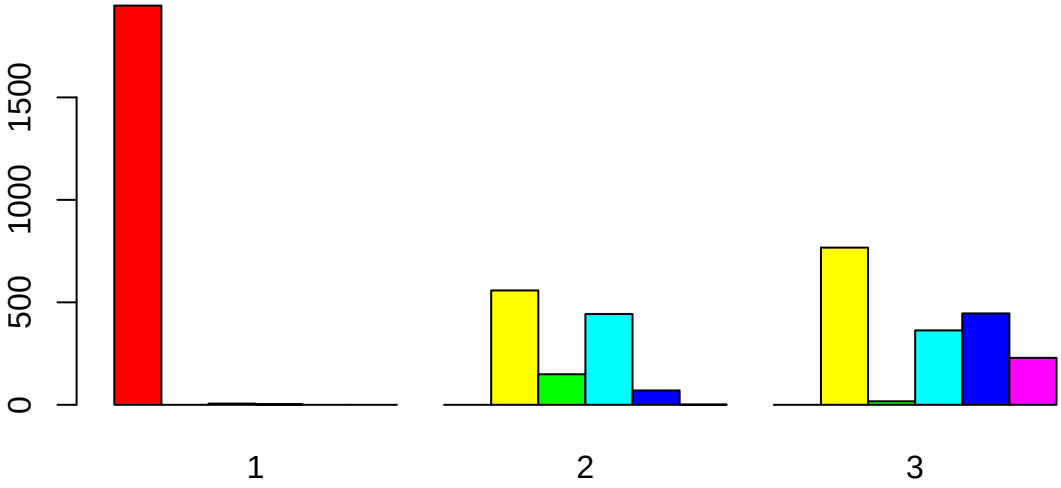

Prop. of classes by relationship
marg
0.0
0.2
0.4
0.6
0.8
1.0
Husband
Not−in−family
Other−relative
Own−child
Unmarried
Wife

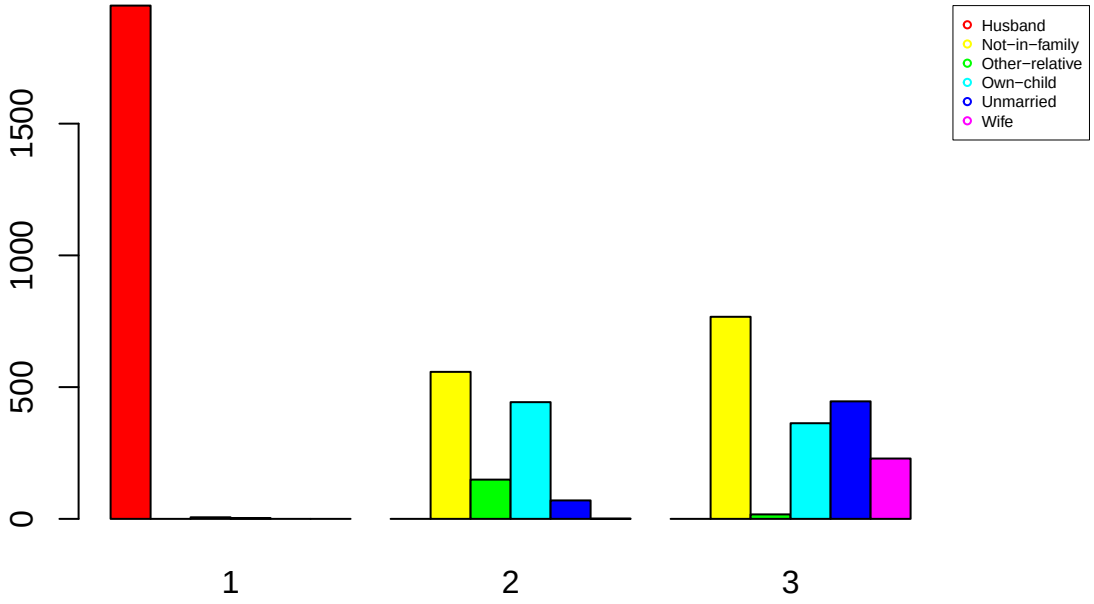



```
## [1] "Cross Table:"
##                  P
##                     1    2    3
##   Husband        1948    0    0
##   Not-in-family     0  558  767
##   Other-relative    6  149   17
##   Own-child         3  443  363
##   Unmarried         0   70  446
##   Wife              0    1  229
## [1] "Distribucions condicionades a columnes:"
##
## P       Husband Not-in-family Other-relative   Own-child   Unmarried
##   1 1.000000000   0.000000000    0.034883721 0.003708282 0.000000000
##   2 0.000000000   0.421132075    0.866279070 0.547589617 0.135658915
##   3 0.000000000   0.578867925    0.098837209 0.448702101 0.864341085
##
## P          Wife
##   1 0.000000000
##   2 0.004347826
##   3 0.995652174
```


0
500
1000
1500
1
2
3

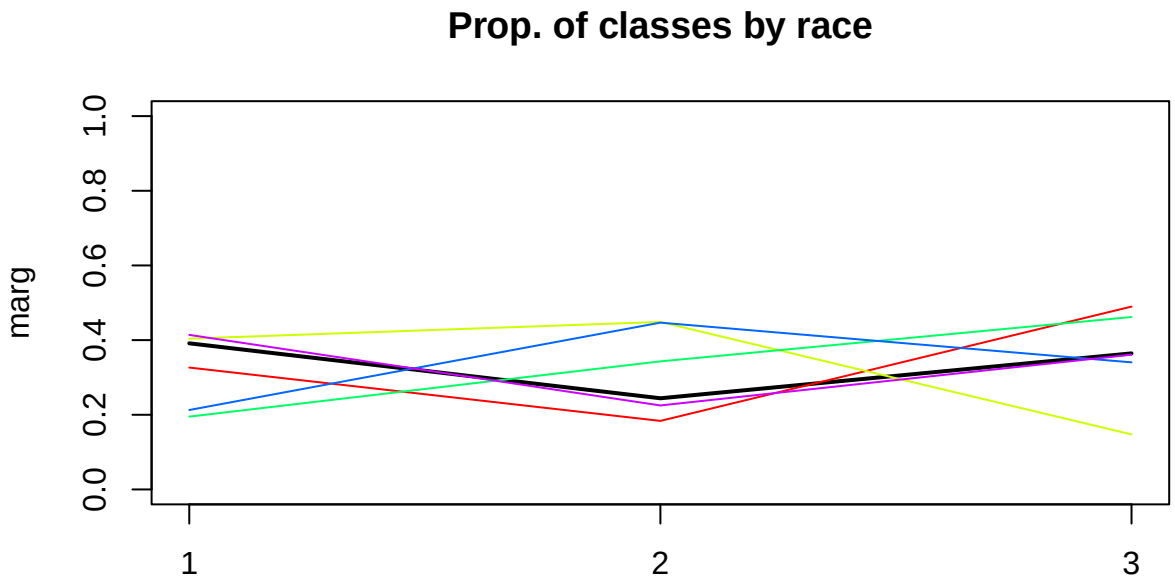

0
500
1000
1500
1
2
3
Husband
Not−in−family
Other−relative
Own−child
Unmarried
Wife

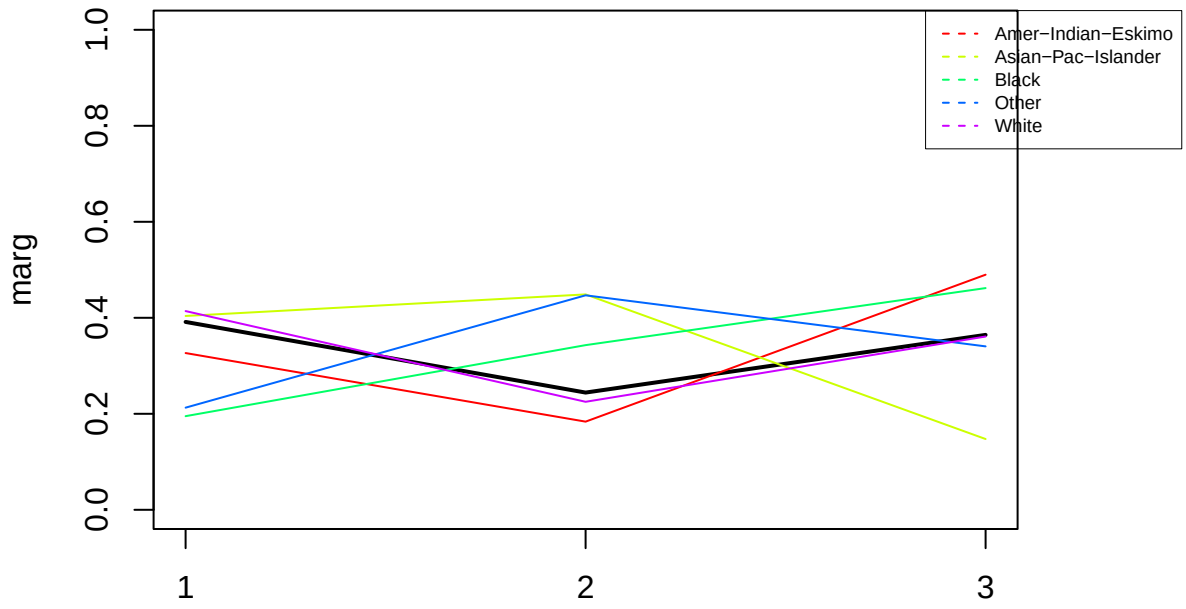

0
500
1000
1500
1
2
3



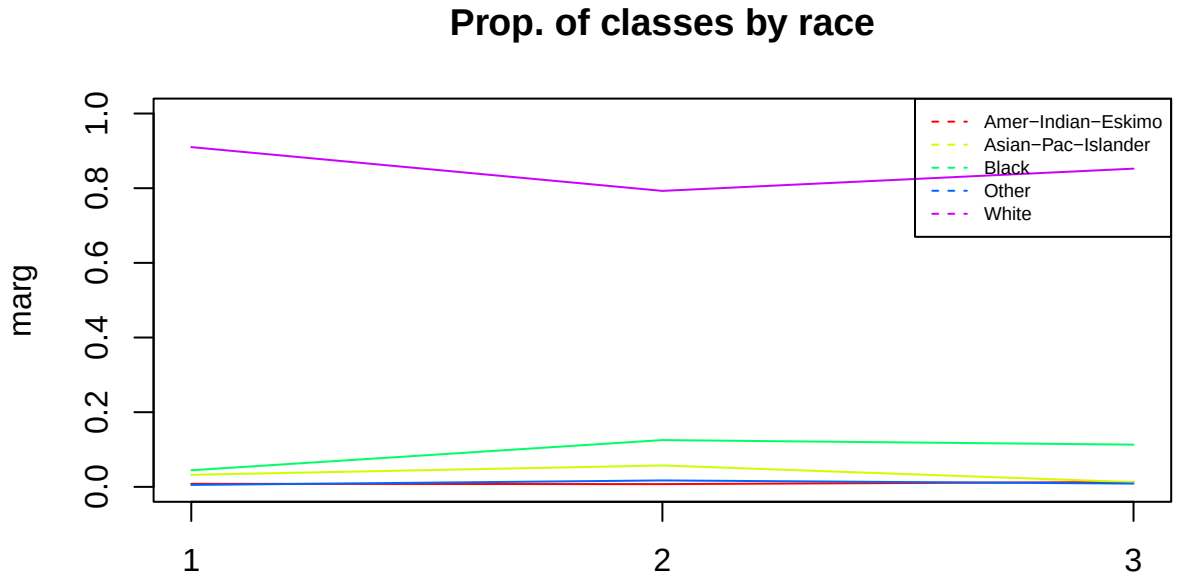

```
## [1] "Test Chi quadrat: "
##
##  Pearson's Chi-squared test
##
## data:  dades[, k] and as.factor(P)
## X-squared = 5854.3, df = 10, p-value < 2.2e-16
##
## [1] "valorsTest:"
## $rowpf
##    Xquali
## P        Husband Not-in-family Other-relative    Own-child    Unmarried
##   1 0.9954011242  0.0000000000   0.0030659172 0.0015329586 0.0000000000
##   2 0.0000000000  0.4570024570   0.1220311220 0.3628173628 0.0573300573
##   3 0.0000000000  0.4209659715   0.0093304061 0.1992316136 0.2447859495
##    Xquali
## P           Wife
##   1 0.0000000000
##   2 0.0008190008
##   3 0.1256860593
##
## $vtest
##    Xquali
## P      Husband Not-in-family Other-relative  Own-child  Unmarried       Wife
##   1  70.443800    -34.049341      -9.749191 -24.678035 -19.236298 -12.451858
##   2 -32.111205     17.486165      19.325686  21.939406  -6.060311  -8.668763
##   3 -42.774452     18.921061      -7.364643   5.441990  24.918090  20.366270
```

```
##
## $pval
##    Xquali
## P         Husband Not-in-family Other-relative     Own-child     Unmarried
##   1  0.000000e+00  0.000000e+00   0.000000e+00  0.000000e+00  0.000000e+00
##   2  0.000000e+00  9.132125e-69   1.633078e-83 5.465957e-107  6.792915e-10
##   3  0.000000e+00  3.824933e-80   8.881784e-14  2.634438e-08 2.368874e-137
##    Xquali
## P            Wife
##   1  0.000000e+00
##   2  0.000000e+00
##   3  1.665531e-92
##
## [1] "Variable race"
## [1] "Categories="        "Amer-Indian-Eskimo" "Asian-Pac-Islander"
## [4] "Black"              "Other"              "White"
```

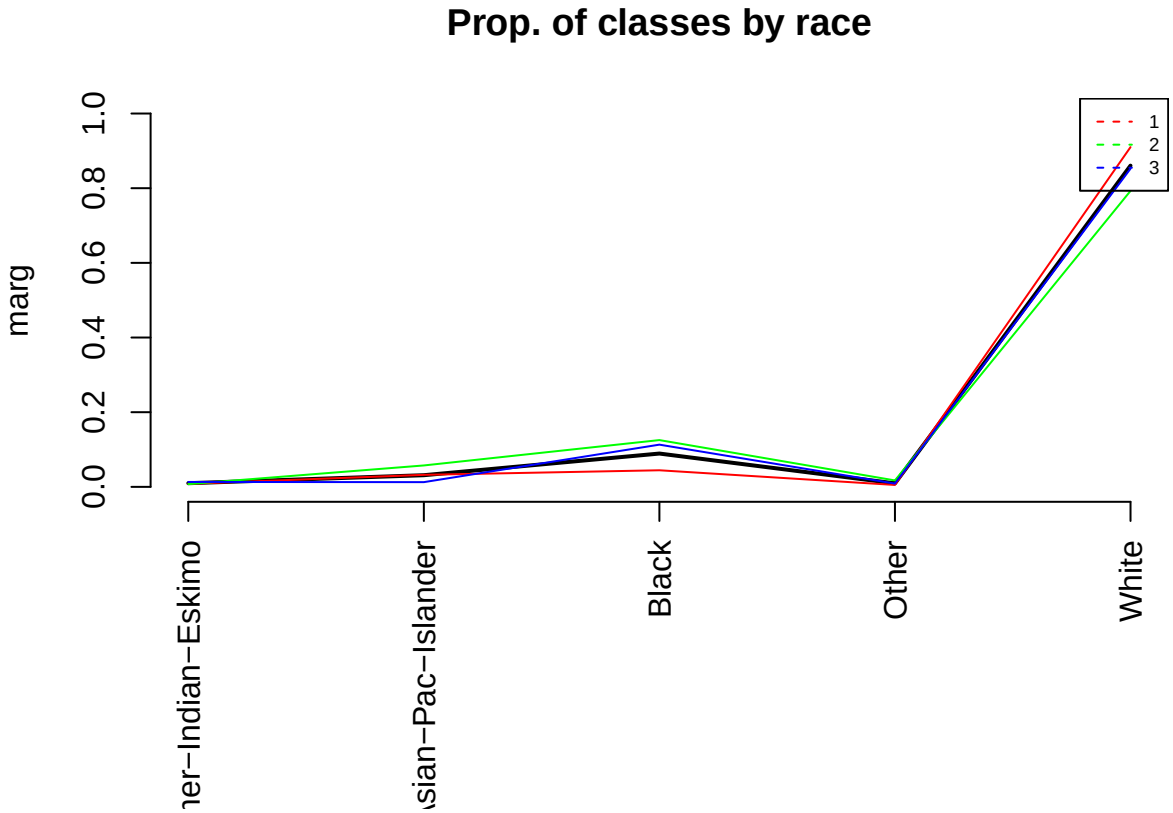


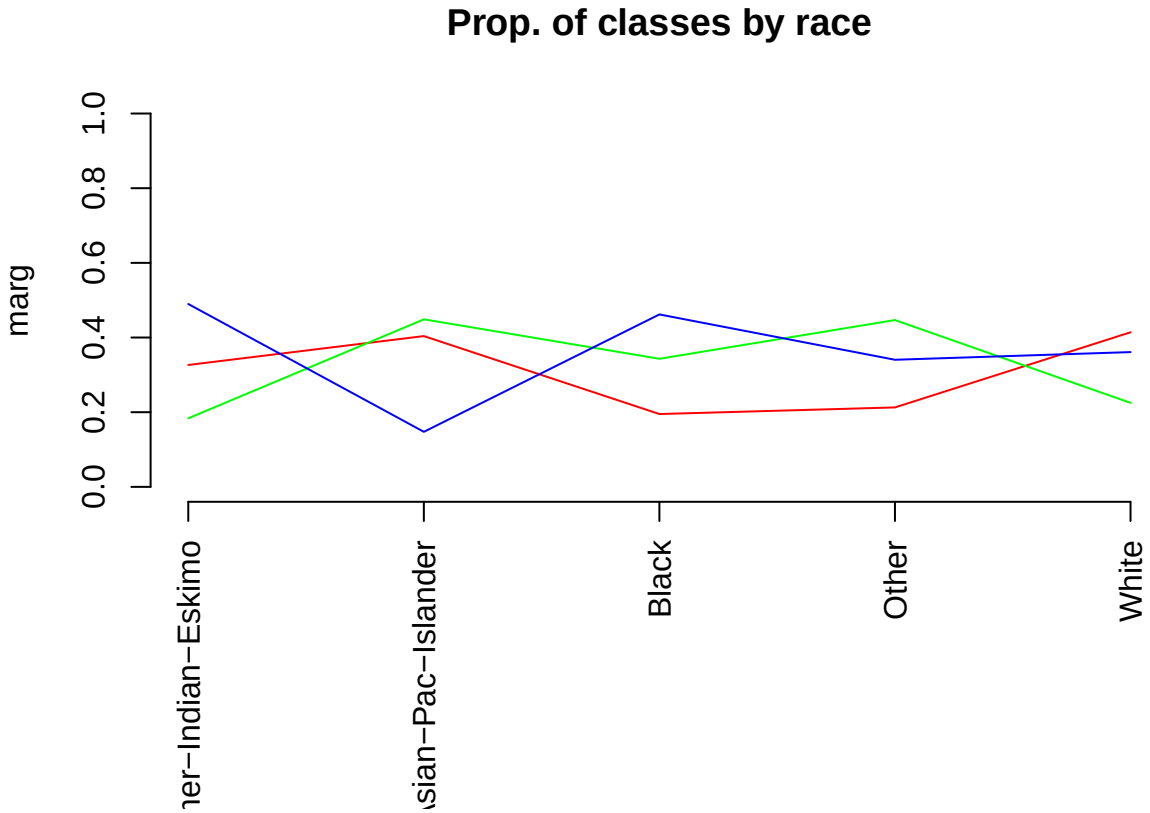


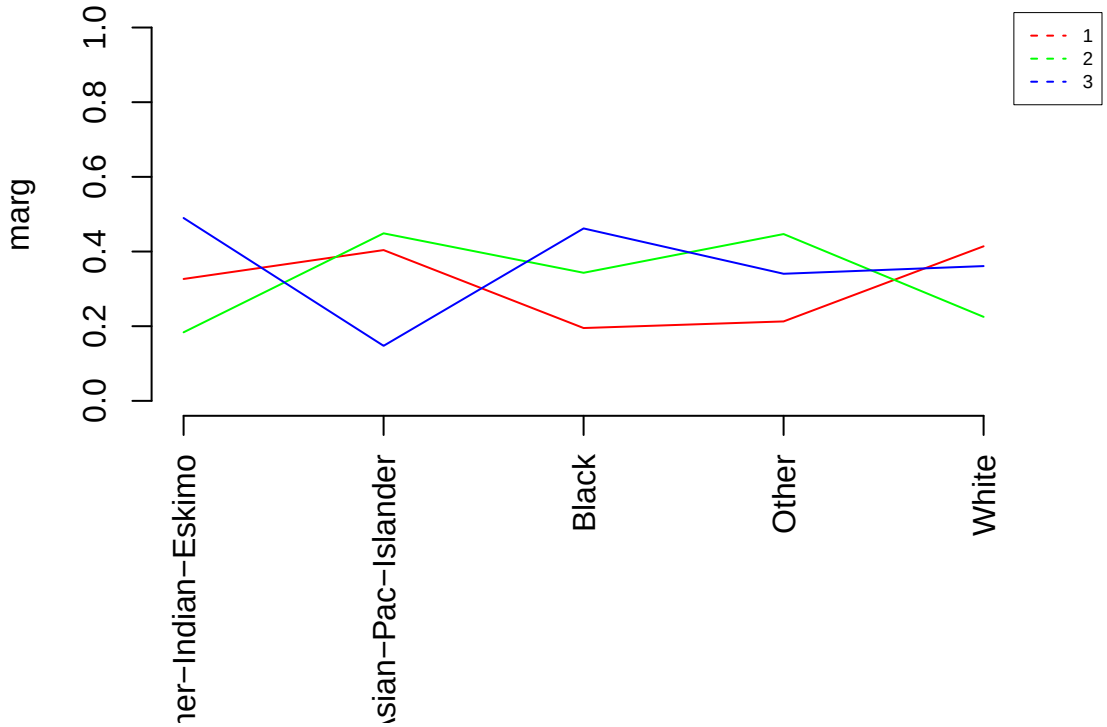

```
## [1] "Categories="        "Amer-Indian-Eskimo" "Asian-Pac-Islander"
## [4] "Black"              "Other"              "White"
```



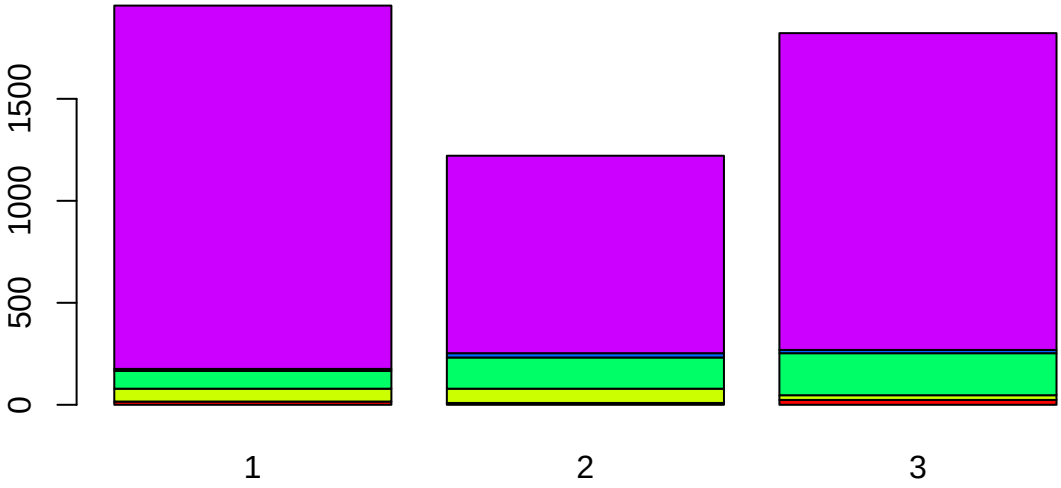

```
## [1] "Categories="        "Amer-Indian-Eskimo" "Asian-Pac-Islander"
## [4] "Black"              "Other"              "White"
```

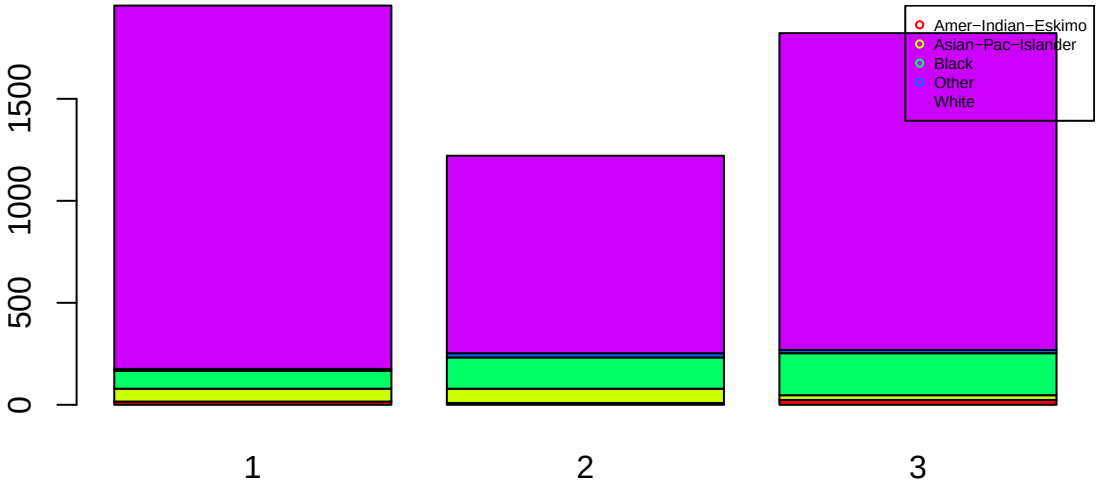

Prop. of classes by race
marg
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3
ner−Indian−Eskimo
sian−Pac−Islander
Black
Other
White

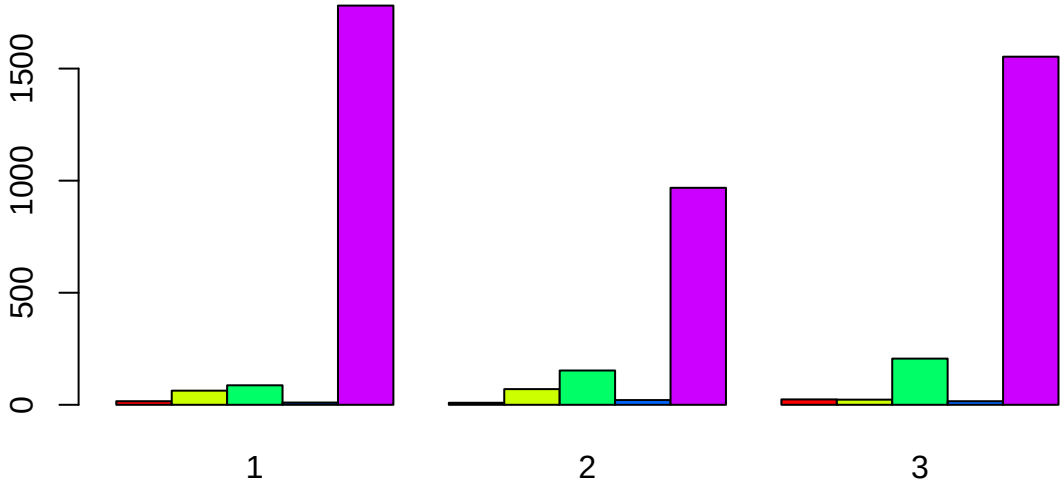

Prop. of classes by race
marg
0.0
0.2
0.4
0.6
0.8
1.0
ner−Indian−Eskimo
sian−Pac−Islander
Black
Other
White

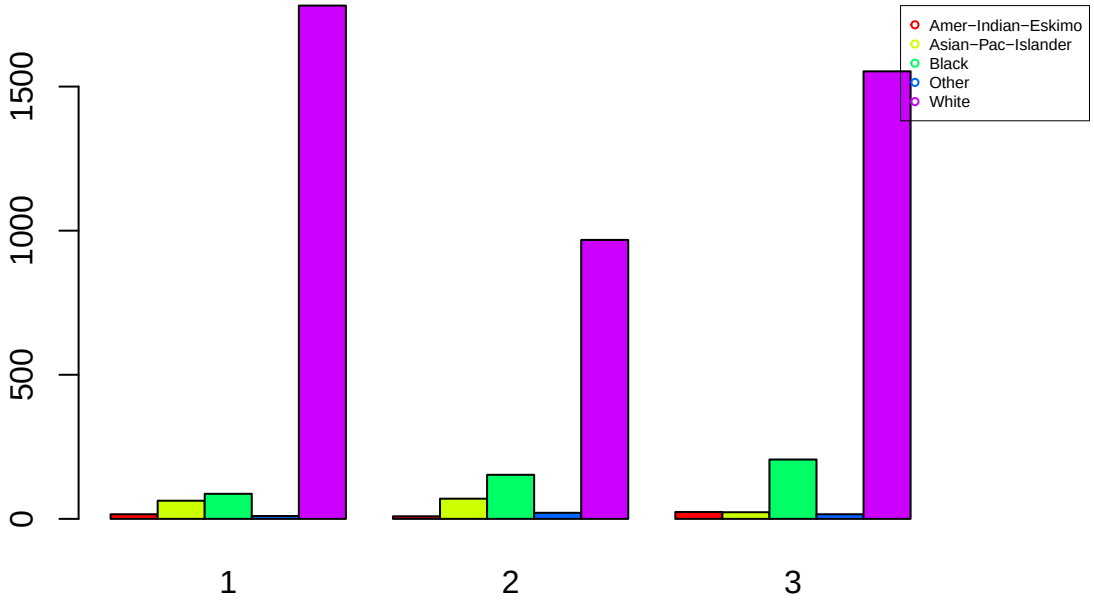



```
## [1] "Cross Table:"
##                       P
##                         1    2    3
##   Amer-Indian-Eskimo   16    9   24
##   Asian-Pac-Islander   63   70   23
##   Black                87  153  206
##   Other                10   21   16
##   White              1781  968 1553
## [1] "Distribucions condicionades a columnes:"
##
## P   Amer-Indian-Eskimo Asian-Pac-Islander     Black     Other     White
##   1          0.3265306          0.4038462 0.1950673 0.2127660 0.4139935
##   2          0.1836735          0.4487179 0.3430493 0.4468085 0.2250116
##   3          0.4897959          0.1474359 0.4618834 0.3404255 0.3609949
```


0
500
1000
1500
1
2
3

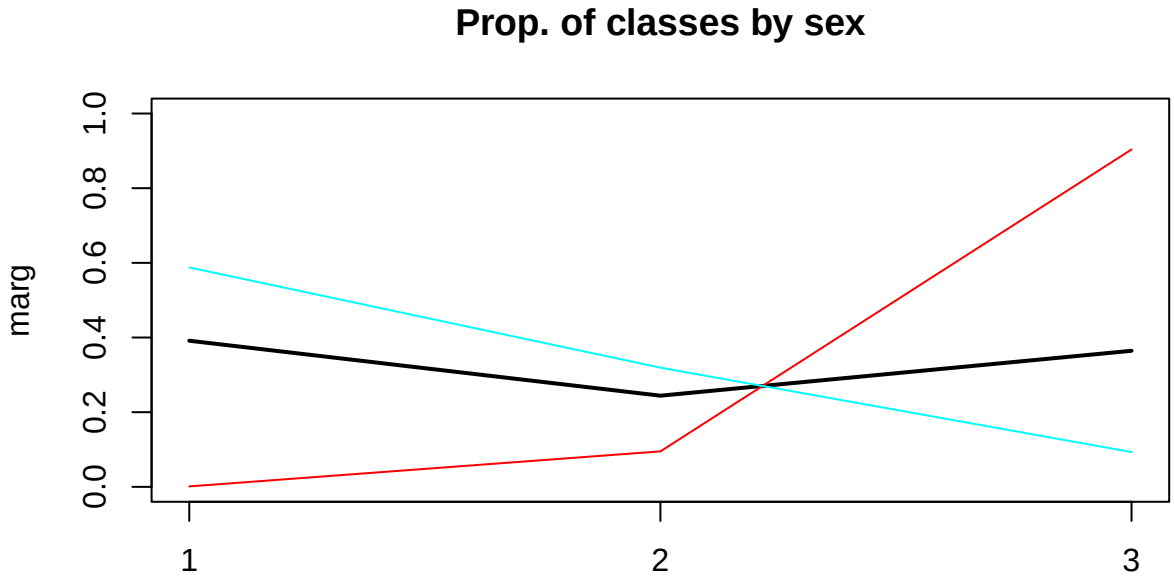

Amer−Indian−Eskimo
Asian−Pac−Islander
Black
Other
White
0
500
1000
1500
1
2
3

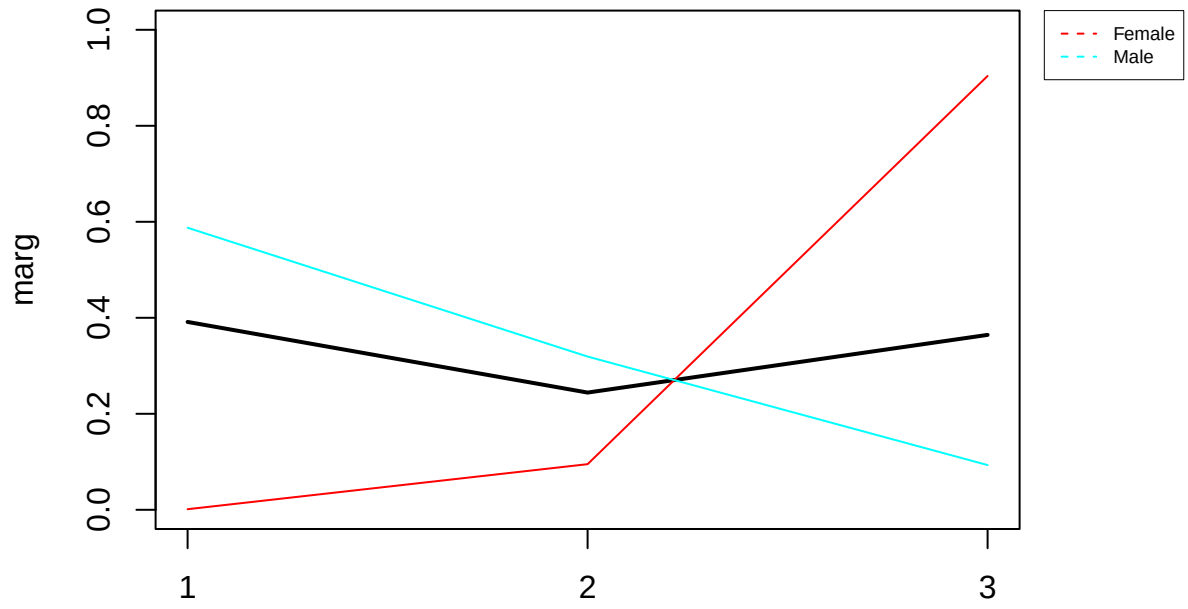

0
500
1000
1500
1
2
3



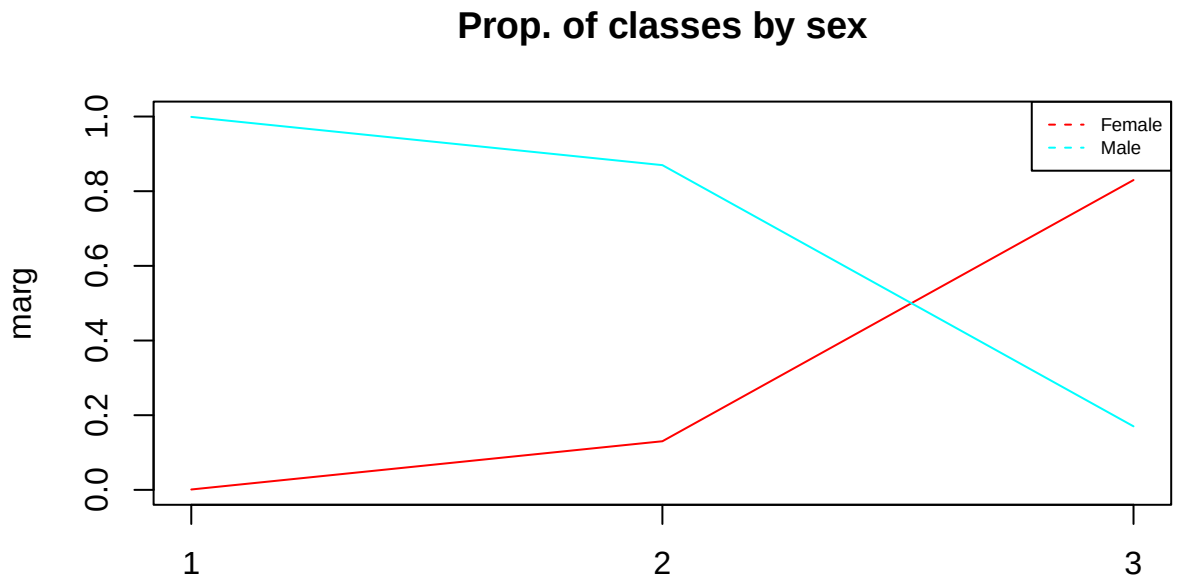

```
## [1] "Test Chi quadrat: "
##
##  Pearson's Chi-squared test
##
## data:  dades[, k] and as.factor(P)
## X-squared = 147.75, df = 8, p-value < 2.2e-16
##
## [1] "valorsTest:"
## $rowpf
##    Xquali
## P   Amer-Indian-Eskimo Asian-Pac-Islander       Black       Other       White
##   1        0.008175779        0.032192131 0.044455800 0.005109862 0.910066428
##   2        0.007371007        0.057330057 0.125307125 0.017199017 0.792792793
##   3        0.013172338        0.012623491 0.113062569 0.008781559 0.852360044
##
## $vtest
##    Xquali
## P   Amer-Indian-Eskimo Asian-Pac-Islander      Black      Other      White
##   1         -0.9349750          0.3235966 -8.9016856 -2.5210857  8.1264236
##   2         -0.9910740          6.0408825  5.0915893  3.2484873 -7.8406934
##   3          1.8329003         -5.7207371  4.4823472 -0.3431371 -1.2420612
##
## $pval
##    Xquali
## P   Amer-Indian-Eskimo Asian-Pac-Islander        Black        Other
##   1       1.749006e-01       3.731217e-01 0.000000e+00 5.849668e-03
```

```
##   2       1.608247e-01       7.663678e-10 1.775373e-07 5.801019e-04
##   3       3.340870e-02       5.303145e-09 3.691325e-06 3.657477e-01
##    Xquali
## P          White
##   1 2.210710e-16
##   2 2.220446e-15
##   3 1.071070e-01
##
## [1] "Variable sex"
## [1] "Categories=" "Female"      "Male"
```

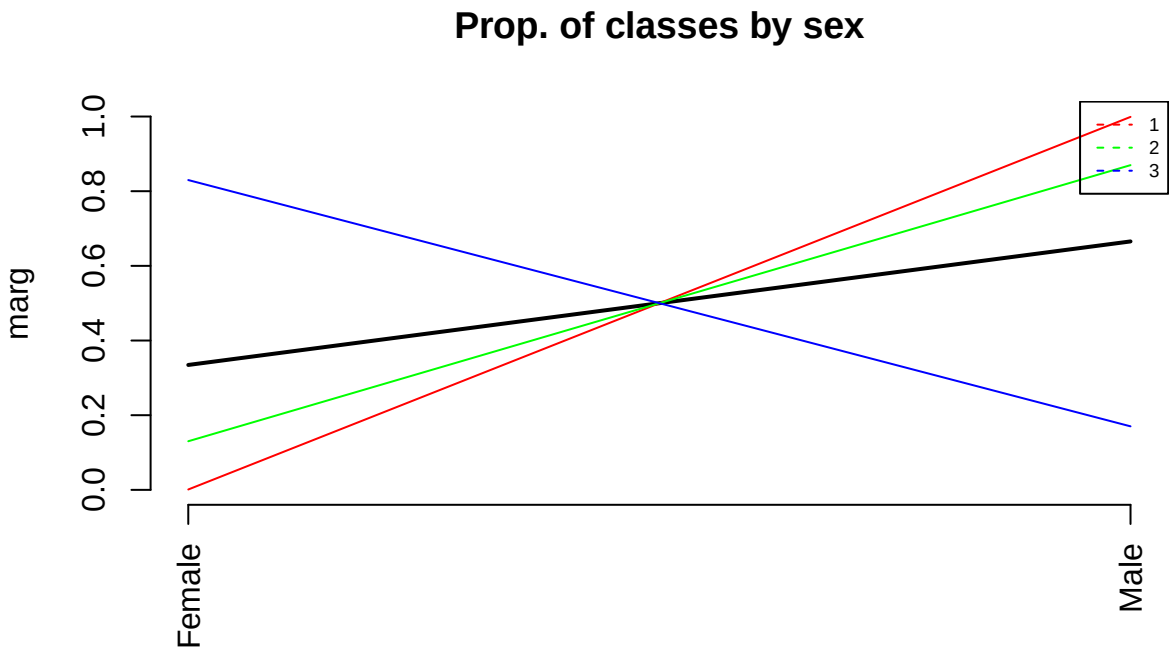


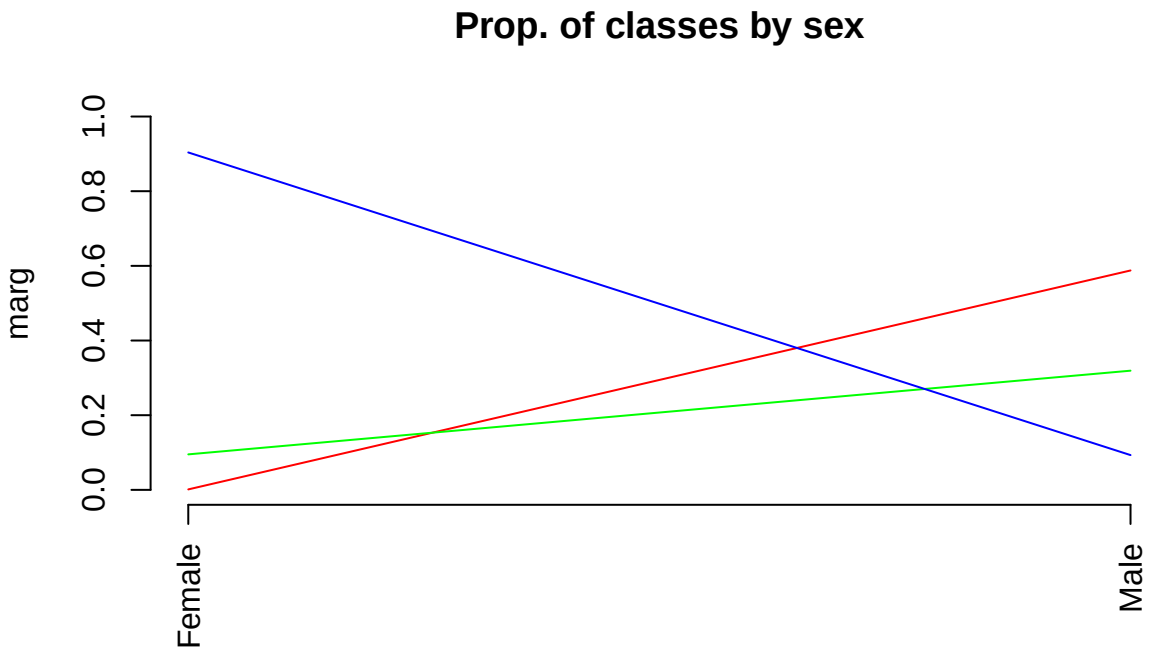


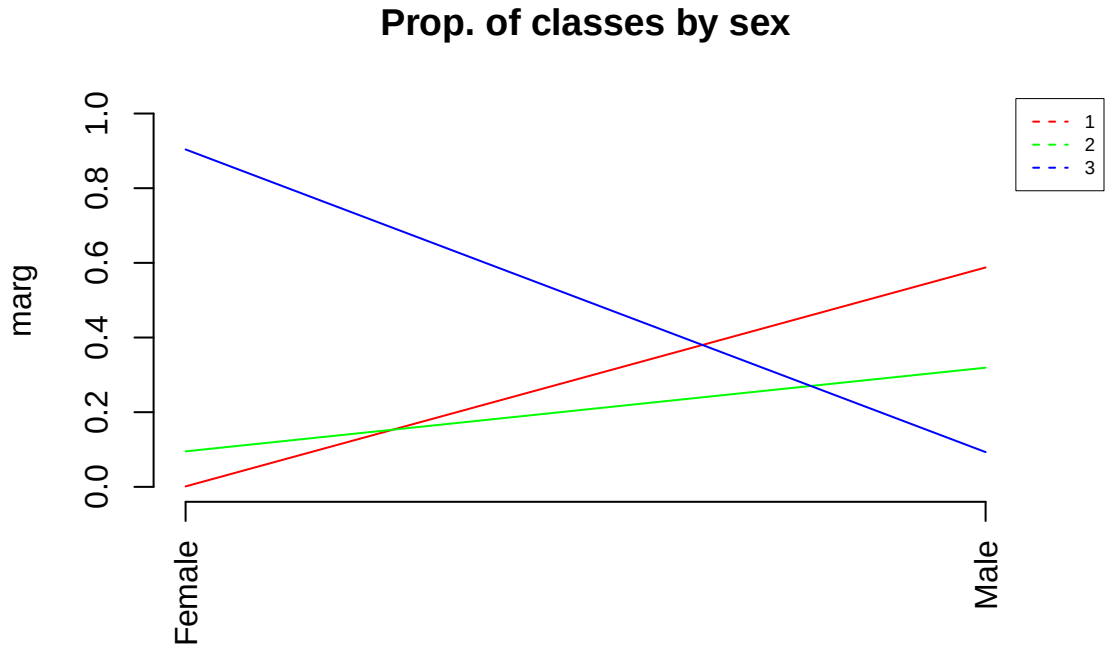

```
## [1] "Categories=" "Female"      "Male"
```



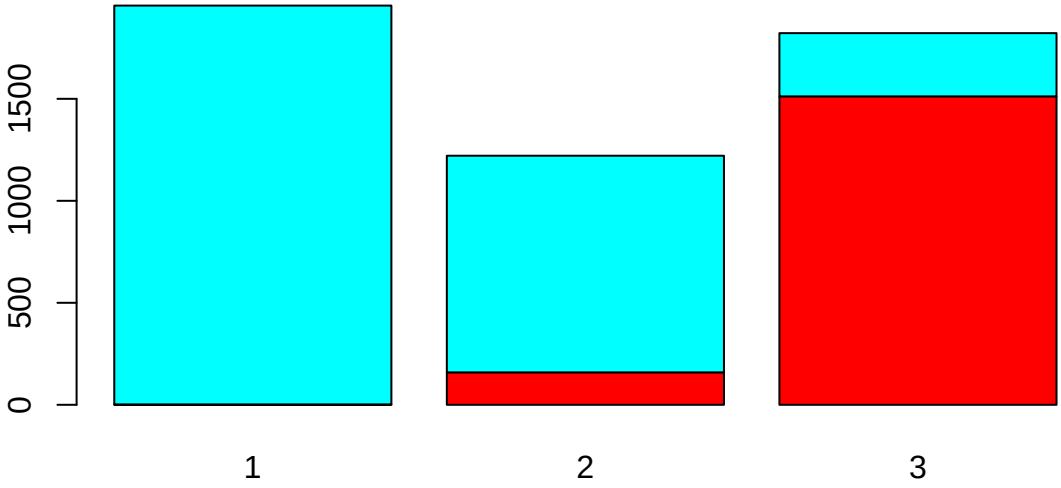

```
## [1] "Categories=" "Female"      "Male"
```

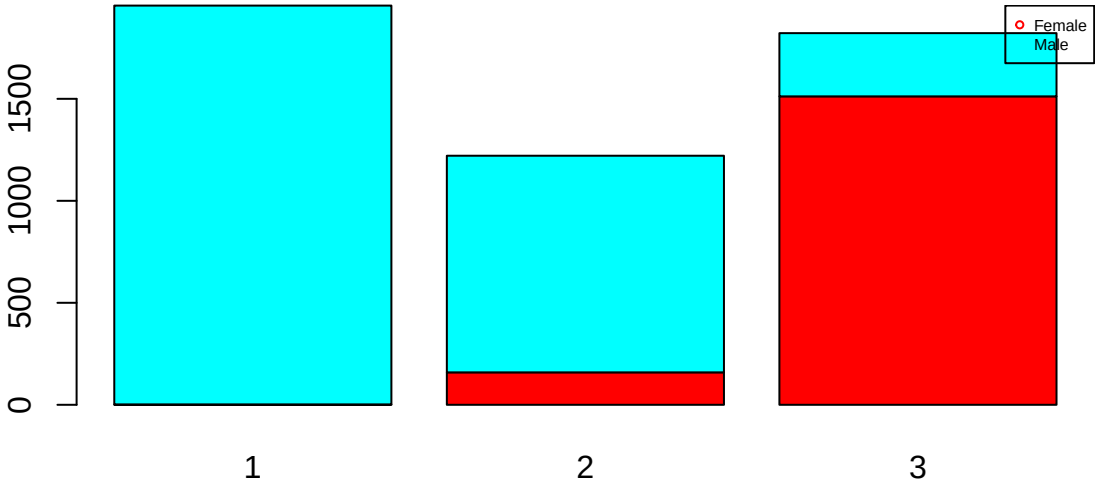

Prop. of classes by sex
marg
0.0
0.2
0.4
0.6
0.8
1.0
Female
Male
1
2
3

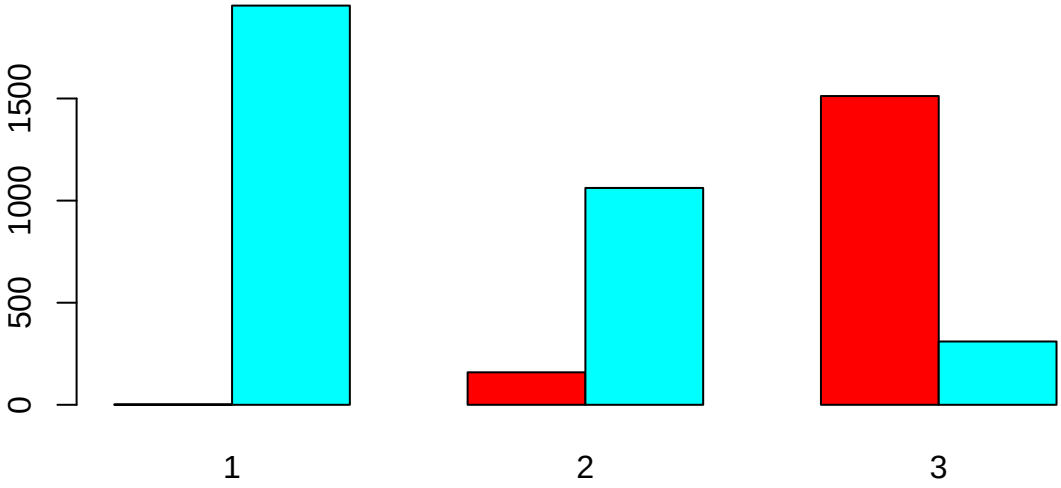

Prop. of classes by sex
marg
0.0
0.2
0.4
0.6
0.8
1.0
Female
Male

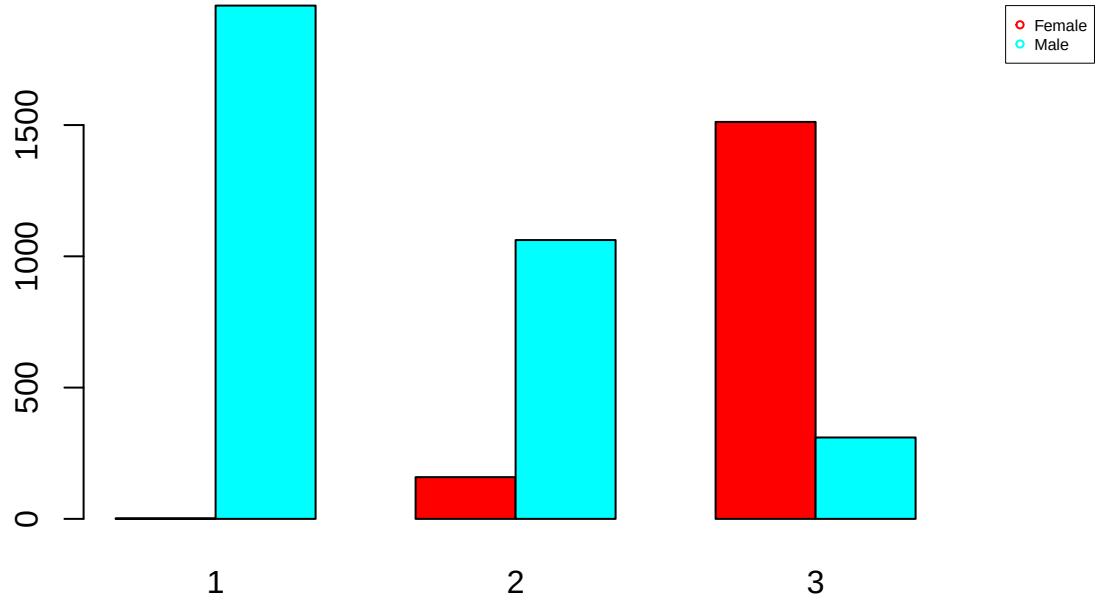



```
## [1] "Cross Table:"
##         P
##             1    2    3
##   Female    2  159 1512
##   Male   1955 1062  310
## [1] "Distribucions condicionades a columnes:"
##
## P        Female        Male
##   1 0.001195457 0.587616471
##   2 0.095038852 0.319206492
##   3 0.903765690 0.093177036
```


0
500
1000
1500
1
2
3

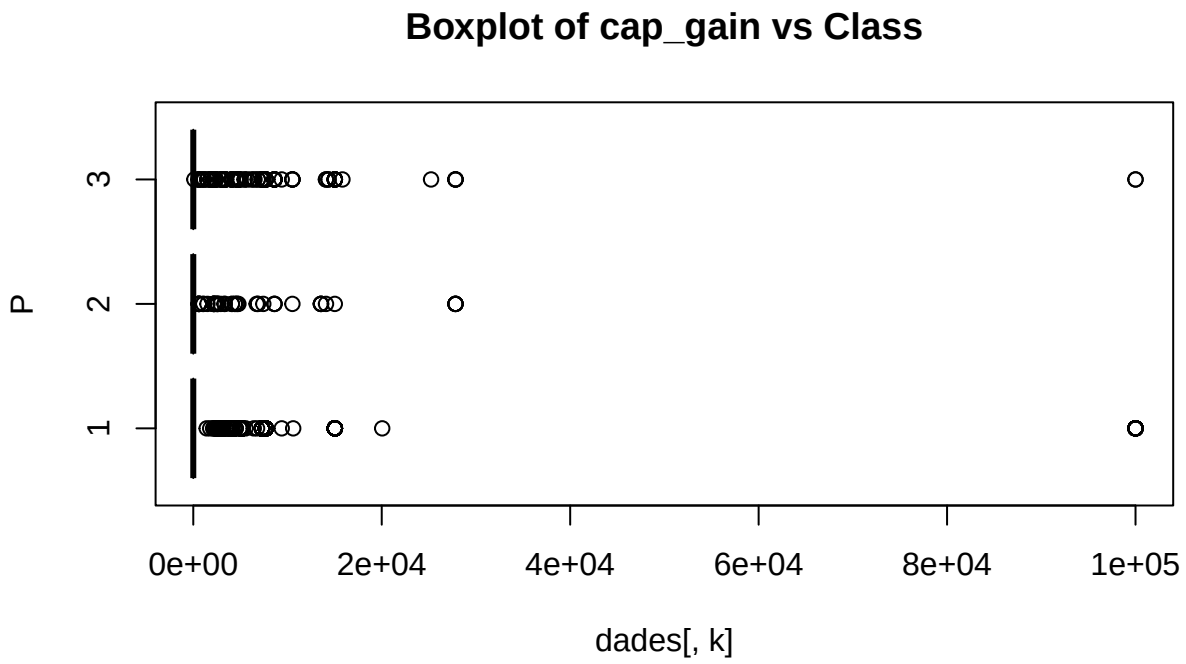

Female
Male
0
500
1000
1500
1
2
3

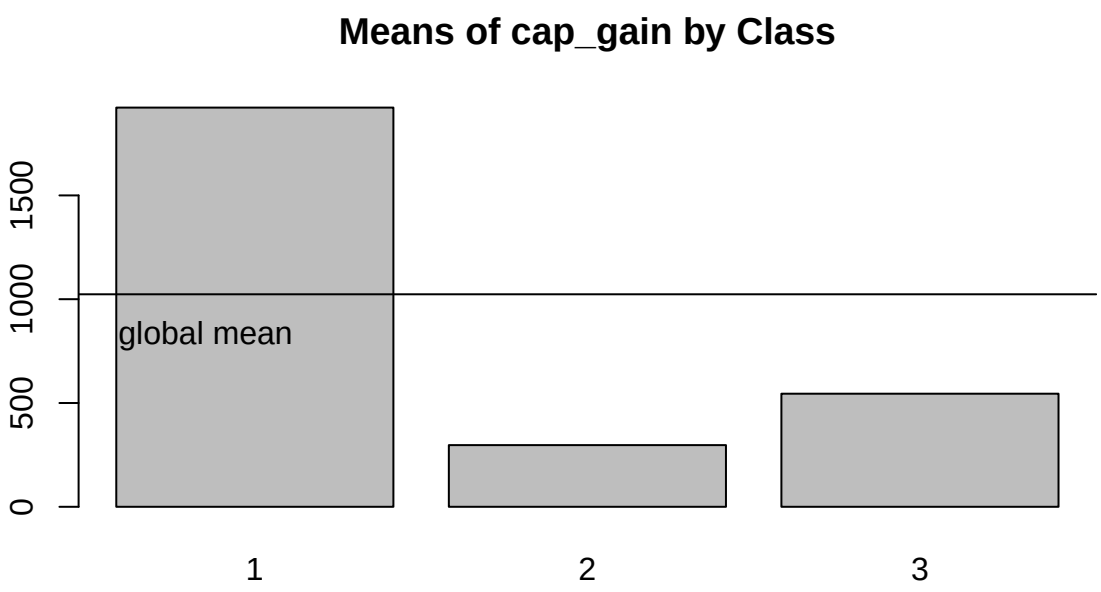

0
500
1000
1500
1
2
3



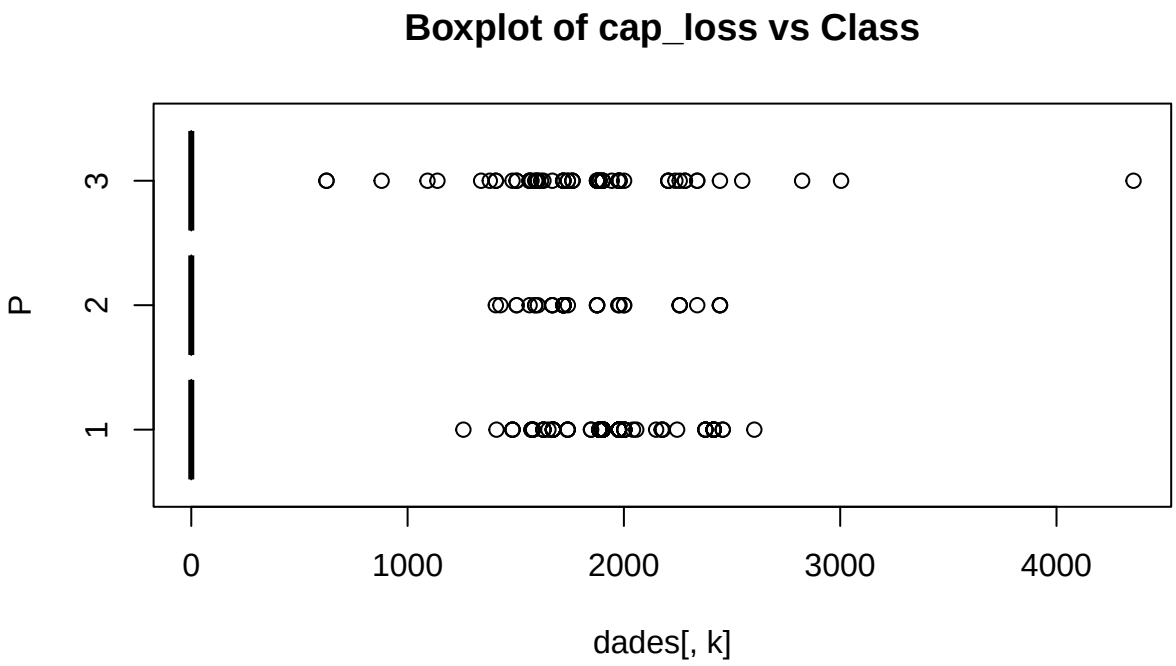

```
## [1] "Test Chi quadrat: "
##
##  Pearson's Chi-squared test
##
## data:  dades[, k] and as.factor(P)
## X-squared = 3214.4, df = 2, p-value < 2.2e-16
##
## [1] "valorsTest:"
## $rowpf
##    Xquali
## P        Female        Male
##   1 0.001021972 0.998978028
##   2 0.130221130 0.869778870
##   3 0.829857300 0.170142700
##
## $vtest
##    Xquali
## P      Female      Male
##   1 -40.08875  40.08875
##   2 -17.40949  17.40949
##   3  56.19643 -56.19643
##
## $pval
##    Xquali
## P         Female         Male
##   1 0.000000e+00 0.000000e+00
```

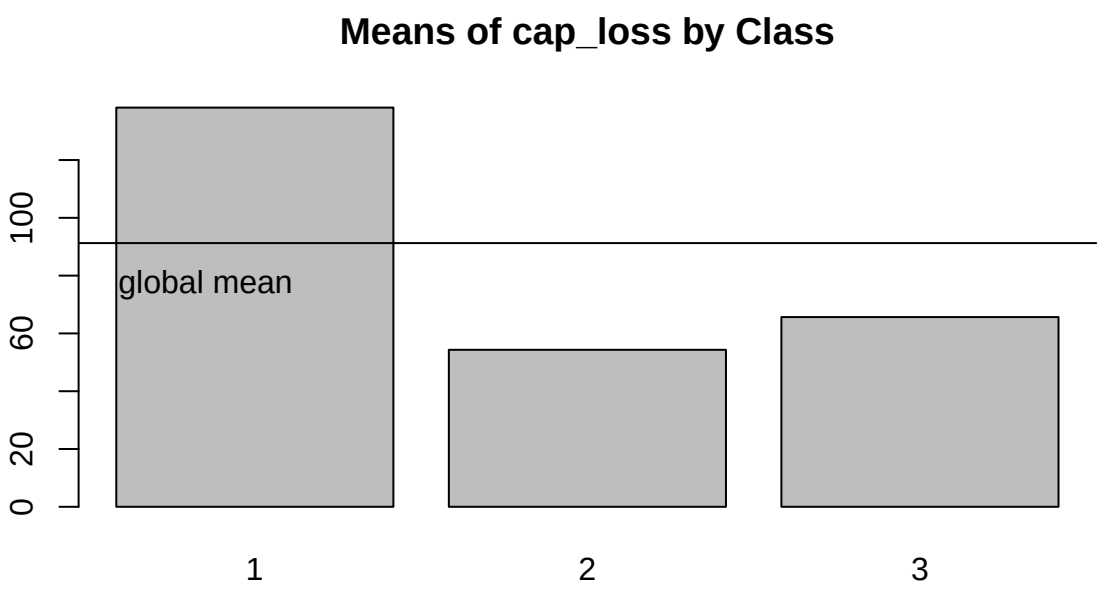
```
##   2 0.000000e+00 3.495277e-68
##   3 0.000000e+00 0.000000e+00
##
## [1] "Anàlisi per classes de la Variable: cap_gain"
```



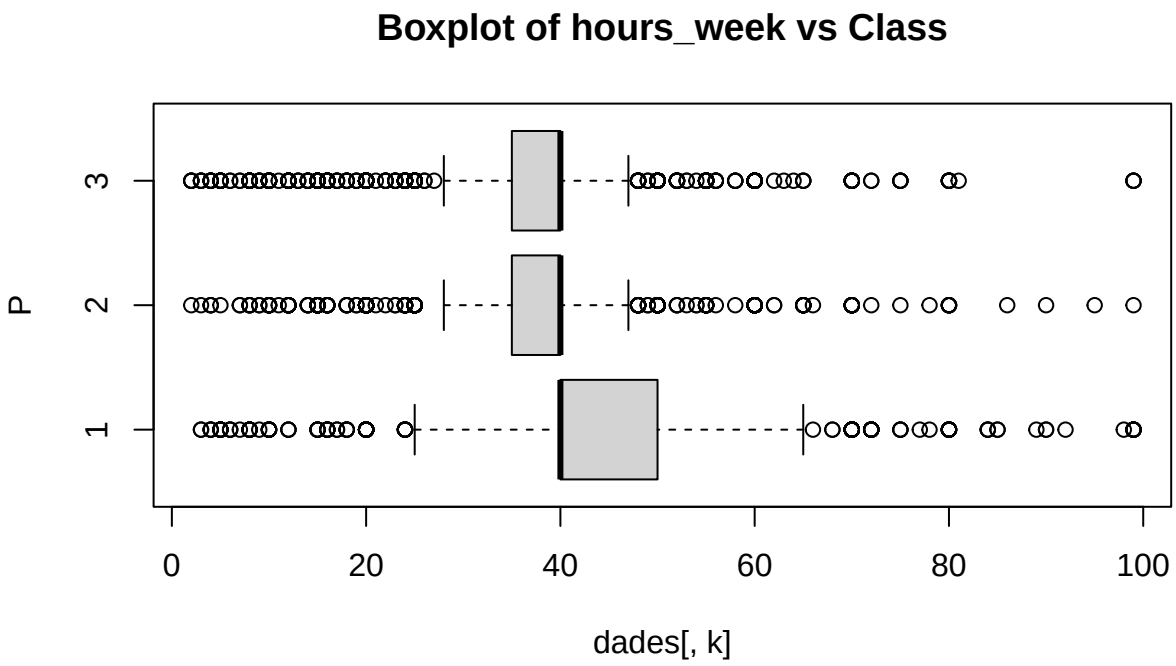


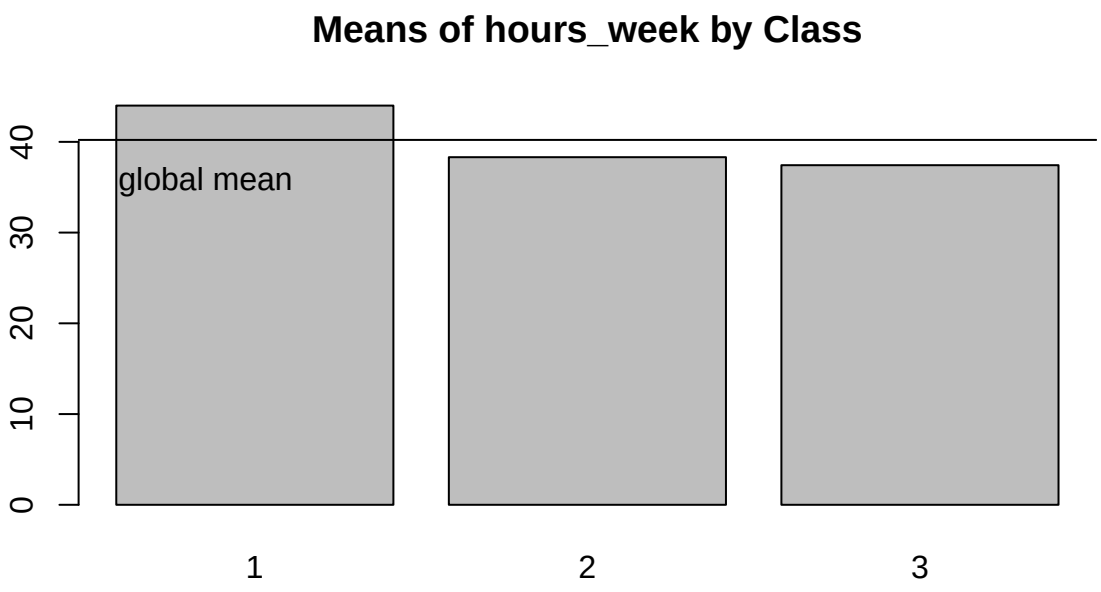

```
## [1] "Estadístics per groups:"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       0       0    1923       0   99999
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0   297.1     0.0 27828.0
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0   544.8     0.0 99999.0
## [1] "p-valueANOVA: 2.01817862825246e-10"
## [1] "p-value Kruskal-Wallis: 1.64878128351474e-19"
## [1] "p-values ValorsTest: "
## [1] 6.363026e-13 2.332985e-05 1.746724e-04
## [1] "Anàlisi per classes de la Variable: cap_loss"
```


Boxplot of cap_loss vs Class
P
3
2
1
0
1000
2000
3000
4000
dades[, k]

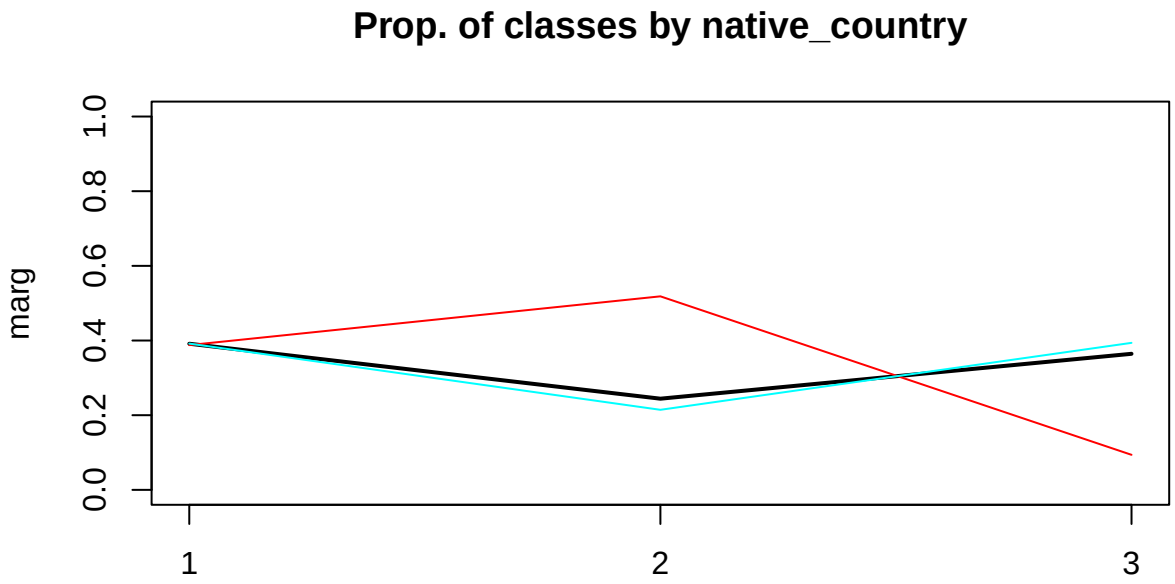


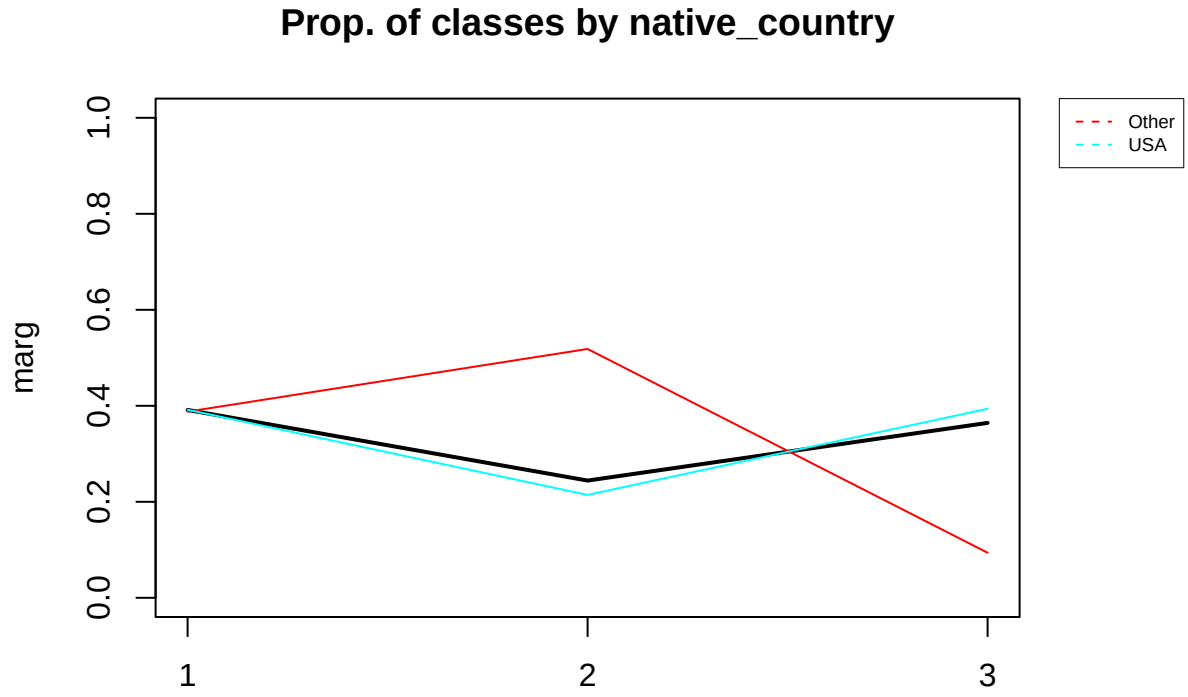

```
## [1] "Estadístics per groups:"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##     0.0     0.0     0.0   138.2     0.0  2603.0
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   54.33    0.00 2444.00
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    0.00    0.00    0.00   65.66    0.00 4356.00
## [1] "p-valueANOVA: 9.61265012137871e-09"
## [1] "p-value Kruskal-Wallis: 2.10298545751866e-09"
## [1] "p-values ValorsTest: "
## [1] 6.588531e-11 1.635098e-04 4.530013e-04
## [1] "Anàlisi per classes de la Variable: hours_week"
```


Boxplot of hours_week vs Class
P
3
2
1
0
20
40
60
80
100
dades[, k]

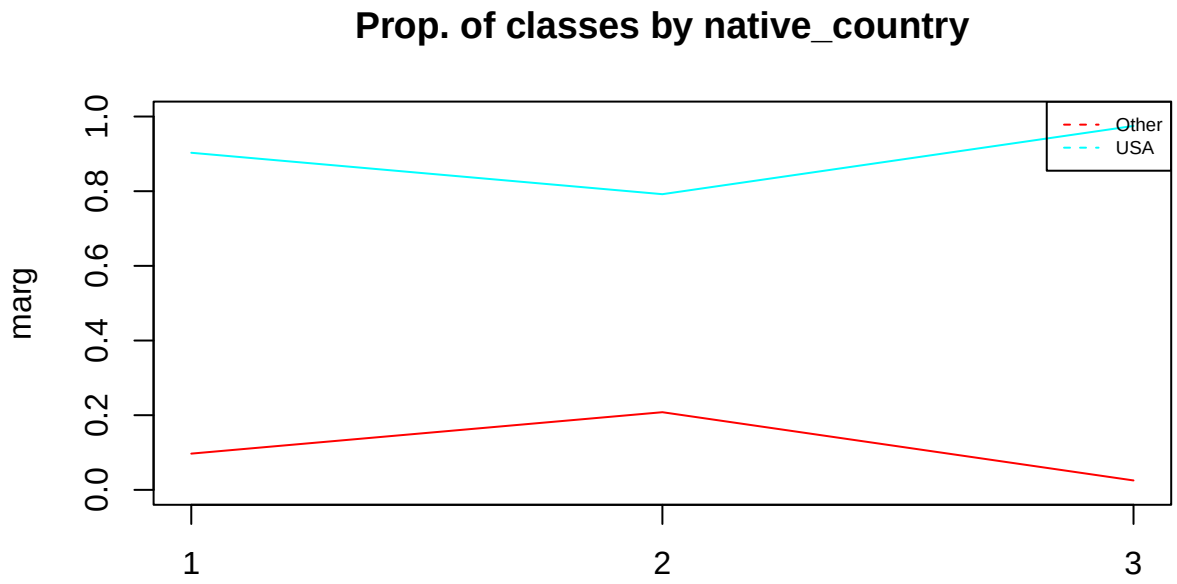



```
## [1] "Estadístics per groups:"
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       3      40      40      44      50      99
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00   35.00   40.00   38.31   40.00   99.00
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##    2.00   35.00   40.00   37.43   40.00   99.00
## [1] "p-valueANOVA: 1.14181725805632e-66"
## [1] "p-value Kruskal-Wallis: 6.19411685035705e-85"
## [1] "p-values ValorsTest: "
## [1] 6.124643e-64 5.816605e-10 0.000000e+00
## [1] "Variable native_country"
## [1] "Categories=" "Other"       "USA"
```

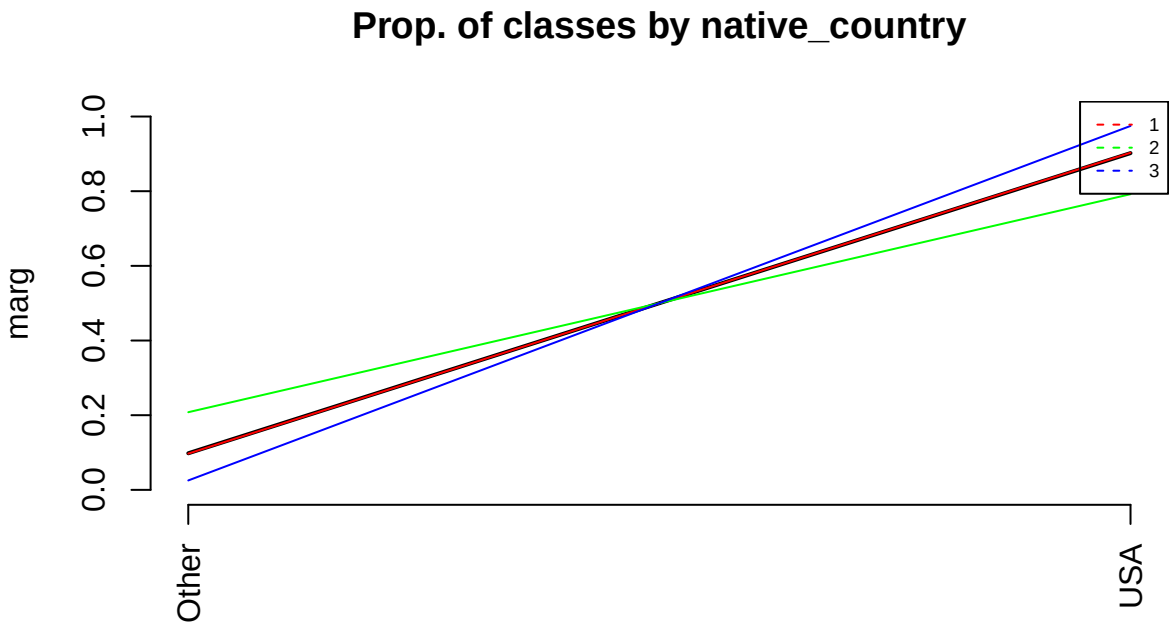

Prop. of classes by native_country
marg
0.0
0.2
0.4
0.6
0.8
1.0
1
2
3

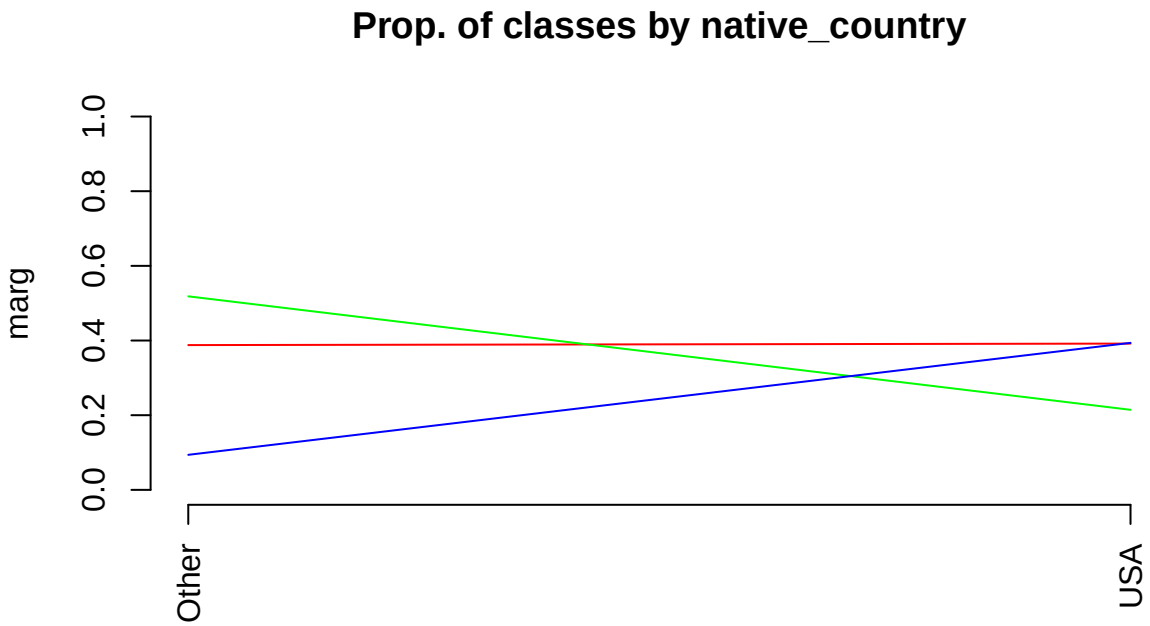


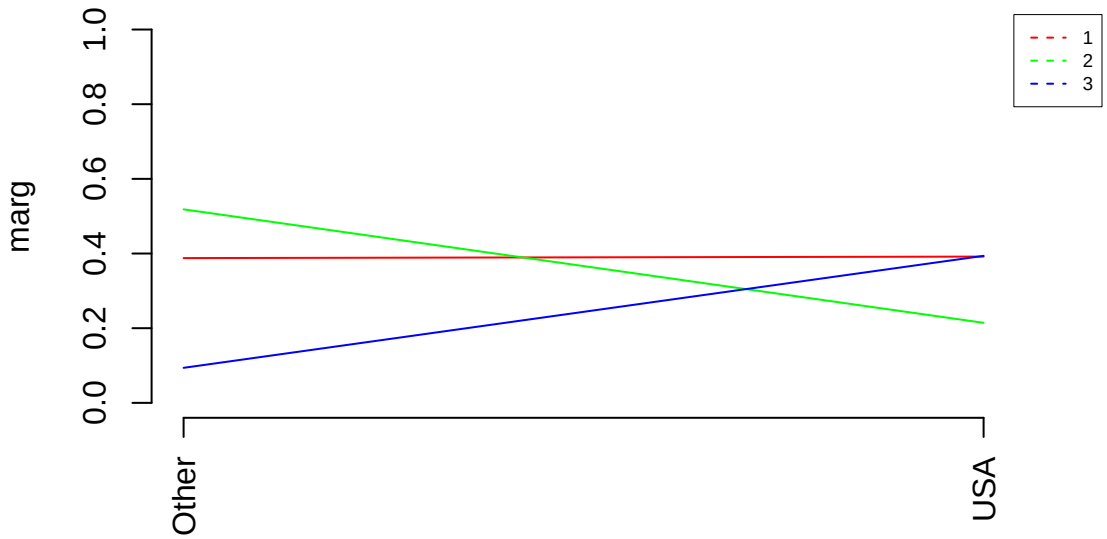

```
## [1] "Categories=" "Other"       "USA"
```



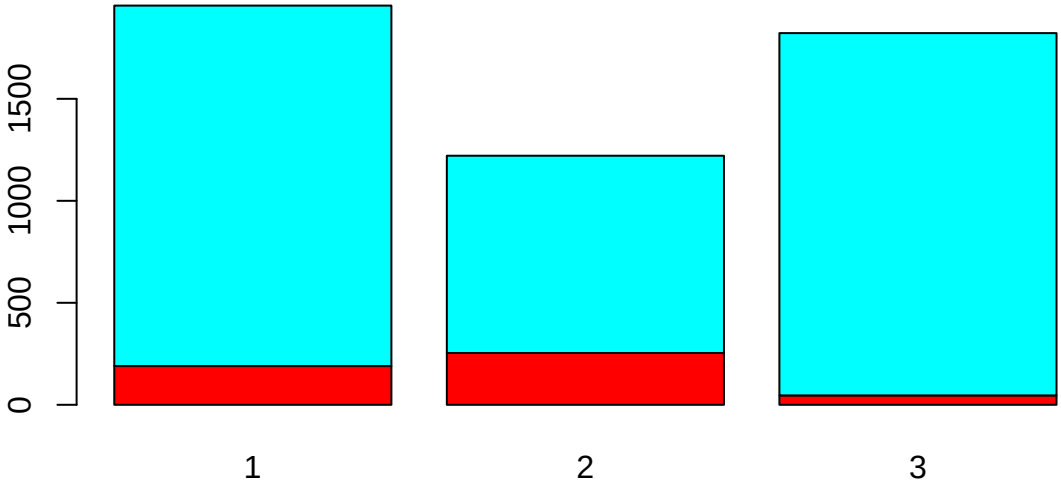

```
## [1] "Categories=" "Other"       "USA"
```

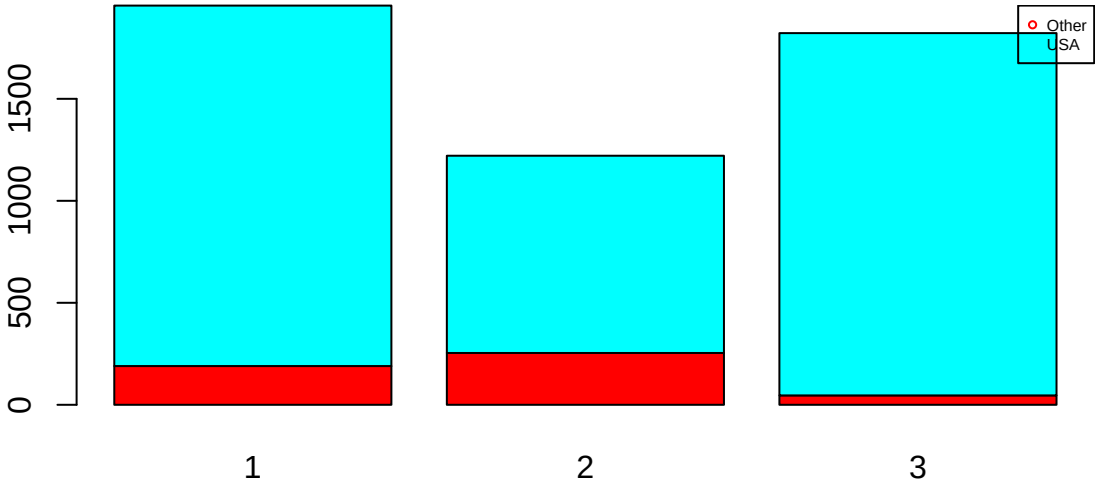

Prop. of classes by native_country
marg
0.0
0.2
0.4
0.6
0.8
1.0
Other
USA
1
2
3

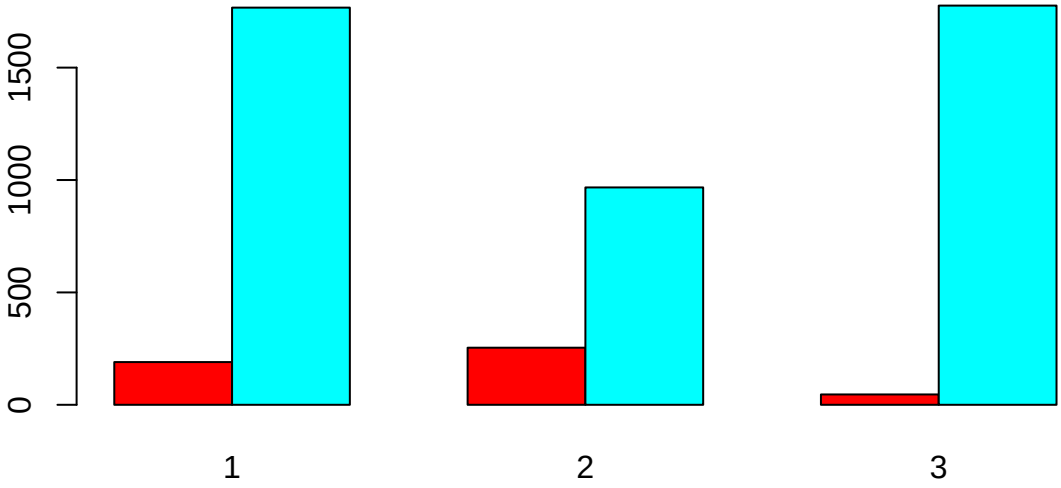

Prop. of classes by native_country
marg
0.0
0.2
0.4
0.6
0.8
1.0
Other
USA

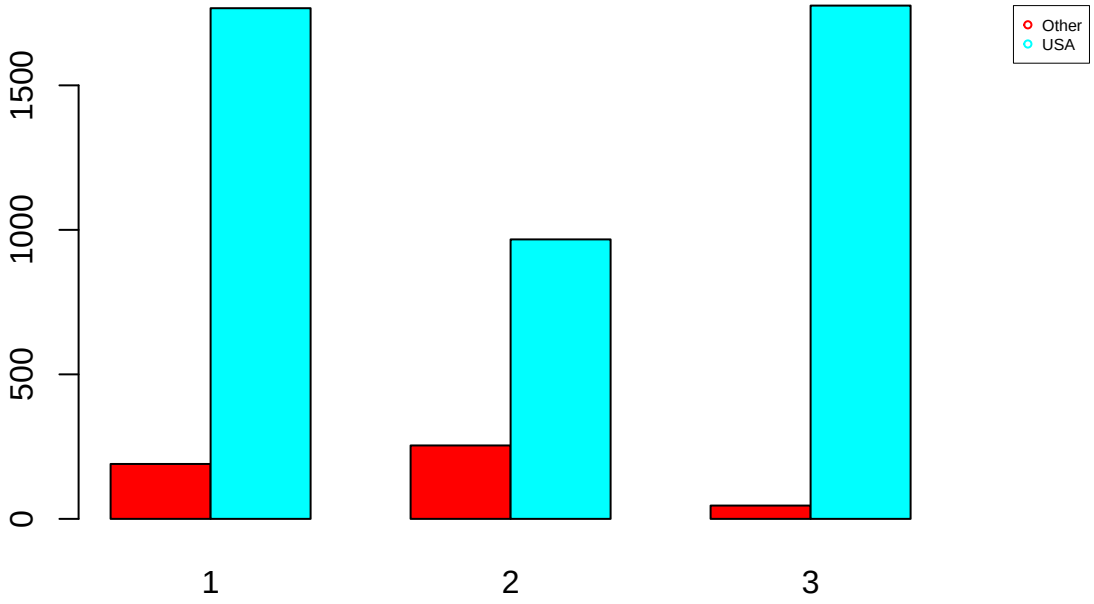



```
## [1] "Cross Table:"
##        P
##            1    2    3
##   Other  190  254   46
##   USA   1767  967 1776
## [1] "Distribucions condicionades a columnes:"
##
## P        Other        USA
##   1 0.38775510 0.39179601
##   2 0.51836735 0.21441242
##   3 0.09387755 0.39379157
```


0
500
1000
1500
1
2
3


0
500
1000
1500
1
2
3
Other
USA


0
500
1000
1500
1
2
3




```
## [1] "Test Chi quadrat: "
##
##  Pearson's Chi-squared test
##
## data:  dades[, k] and as.factor(P)
## X-squared = 276.33, df = 2, p-value < 2.2e-16
##
## [1] "valorsTest:"
## $rowpf
##    Xquali
## P        Other        USA
##   1 0.09708738 0.90291262
##   2 0.20802621 0.79197379
##   3 0.02524698 0.97475302
##
## $vtest
##    Xquali
## P         Other         USA
##   1  -0.1740619   0.1740619
##   2  14.8742209 -14.8742209
##   3 -13.1013773  13.1013773
##
## $pval
##    Xquali
## P          Other          USA
##   1 4.309084e-01 4.309084e-01
```

```
##   2 2.422967e-50 0.000000e+00
##   3 0.000000e+00 1.616682e-39
```

```
#descriptors de les classes més significatius. Afegir info qualits
for (c in 1:length(levels(as.factor(P)))) {
  if(!is.na(levels(as.factor(P))[c])){
    print(paste("P.values per class:",levels(as.factor(P))[c]));
    print(sort(pvalk[c,]), digits=3)
  }
}
```

```
## [1] "P.values per class: 1"
##      workclass        marital     occupation   relationship           race
##       0.00e+00       0.00e+00       0.00e+00       0.00e+00       0.00e+00
##            sex native_country            age     hours_week       cap_gain
##       0.00e+00       0.00e+00      9.36e-115       6.12e-64       6.36e-13
##       cap_loss        edu_num
##       6.59e-11       8.90e-07
## [1] "P.values per class: 2"
##            age      workclass        marital     occupation   relationship
##       0.00e+00       0.00e+00       0.00e+00       0.00e+00       0.00e+00
##           race            sex native_country        edu_num     hours_week
##       0.00e+00       0.00e+00       0.00e+00       2.29e-13       5.82e-10
##       cap_gain       cap_loss
##       2.33e-05       1.64e-04
## [1] "P.values per class: 3"
##      workclass        marital     occupation   relationship           race
##       0.000000       0.000000       0.000000       0.000000       0.000000
##            sex     hours_week native_country       cap_gain            age
##       0.000000       0.000000       0.000000       0.000175       0.000427
##       cap_loss        edu_num
##       0.000453       0.051825
```

#PCA # 1. Load Libraries and Data

```
library(dplyr)
```

```
##
## Adjuntando el paquete: 'dplyr'
```

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

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

```
setwd("C:/Users/Ericd/Pictures/ADEI")
dd <- read.csv("adult_def.csv", stringsAsFactors = TRUE);
set.seed(123)
samp <- sample(48842, 5000)
dd <- dd[samp,]
objects()
```

```
##  [1] "actives"         "c"               "c1"              "c2"
##  [5] "cdg"             "colperc"         "d"               "dades"
##  [9] "dcon"            "dd"              "dissimMatrix"    "distMatrix"
```

```
## [13] "h1"              "k"               "K"               "kw"
## [17] "marg"            "n"               "nameP"           "nc"
## [21] "o"               "P"               "paleta"          "pvalk"
## [25] "rowperc"         "s"               "samp"            "table"
## [29] "ValorTestXnum"   "ValorTestXquali"
```

```
attributes(dd)
```

```
## $names
##  [1] "age"            "workclass"      "fnlwgt"         "edu_num"
##  [5] "marital"        "occupation"     "relationship"   "race"
##  [9] "sex"            "cap_gain"       "cap_loss"       "hours_week"
## [13] "native_country" "income"         "income_integer"
##
## $row.names
##    [1]  2986 29925 29710 37529  2757 45404 46435 38938  9642 31313 14183 15180
##   [13] 27168 24173  9097 30538 28981  7989 13536 24541  6216 17983 29394 28825
##   [25] 41237    41 14426 40159  7284 43782 28502 11473 42802 43042 12301  6134
##   [37] 33523 21812 39895  9640 19742  9326 26510 20960 14403  5967 45221 48306
##   [49] 43244 28799 26836 12049 15150  5027 47982 32606 16152 25559 14215 14287
##   [61] 23194 24558 34976 14491 17413 12048 40503 34724 38126 20477 17369 31542
##   [73] 37001 21069 40632 30575 43204 37544 33753 18891 11284 16579 44196 45932
##   [85] 26801 44983 25902 36717 28078  3833 26028 43406 41488 47485 47626 14536
##   [97] 17533 26503 37783 22763 32953   413 10762 48182 30571 42521  8986 14745
##  [109] 25946 14804 43928  6601  6790   618 47802 31517   539 32263 30453  3625
##  [121] 15582 19392 46137  2211  2286 18762 32281 42039  7826 14751 24263 25449
##  [133] 40849 10687 27455 12637 20193  1165 31665 22467 29826 13689 18496 13795
##  [145] 19419 15703  6623  8983 37769 29479  5407 44617 32634 12585 21085  1835
##  [157] 42221 21099 41043 25442 36940 40153 31919 10563 38265 10848 46438 48803
##  [169] 44128 22925  3814  8176 28583 10833 17746 45299 29133    31 46195 45072
##  [181]  7588 44449  4620 45980  3247 24016  5242 11667 33027 22369  5409 25931
##  [193] 11256 23900  2450 44500  1673 19851 45064 48434 17993  6007 18144 25243
##  [205] 13431 38467 34728  1060 13179 42846 27753 46468  1204 32117 24111  9919
##  [217] 43694 38191  3984  7880 22119  2096 44990 20873 24522  6333  2894 24696
##  [229] 23883 19955 47818 14417 27380 25244 41009  5596 40276 37167  3696 22974
##  [241] 46707  4913 10301 11372 34165 18547 38488 13238 29370 46843 36086 11187
##  [253] 16635 41603  9182 38210  6312 23316 17049 41405 24671 28112 34134 47800
##  [265] 25196   242  7071 47048 40160  1498 20698    79  9723 39149 34599  4518
##  [277] 16346  4695 32102 13353 15489 31606  9448 27020 42824 16436 26691 42887
##  [289] 30237  1230 35556 32076 26715 18388 16990 23977  3080 31315 43755 45555
##  [301]  9123  8263 36810 20400  4972  3509 48101 47241 37492 43072  6373 48257
##  [313]  7860 17220   602  8523 47606 46867 28577  7877  2777 30712 17749  7488
##  [325] 35916 11126 34564 44143 45430 41409  9739 19512 14607 35817 24963 35560
##  [337] 38245 33510  4682  3421 37765 34827 20943  2748 44859 36352 13376 39084
##  [349] 10400  1553 32128 27365 10605 22402 19927 10216  2882  4015 27913 29465
##  [361]  9174 37320  7657 31351  3017 17656 42155   459 27463 46642 41099 10044
##  [373] 28390 14666 26226 25658  7132 38920 31319 19360 28871 22344 33227 48715
##  [385] 18300 25871 20878 29851 36385 35602  8539 35914 36671 29343 47159 34162
##  [397] 48538   860 35351 18117 15954 46910 11341 24243 46738  8905 36829 41218
##  [409]  1963 23294 11269 27103 17819  1923 31466  5178  8482 40931 13102 47695
##  [421] 34696 43289 25640 16011 37262 12636 20547  3804 37247 37015 37208 18876
##  [433] 17017 26566 26584 41184  9104 39264 24378 13763 13590 15828  4750 18807
##  [445] 26334 48451 26034 41609 13797  8848 38947 36378 12791  7196 25245 47061
##  [457] 10333 35533 33285 34214  2647 14816 34796 18191 38582  2826 46122 20393
##  [469] 21086 28852 18304  2761 47651 33471    76 16029 37853 40302  9597  4297
```

```
##  [481] 47529 37106 36344 40838 25703 18082 35998 33749 34490 19879 48068 37294
##  [493] 31859 36380  8946  6529 23985 15524  7053 16719  2222 22135 23404 25191
##  [505] 32164 42040 20995 40459 11166 35046 29919 16322 42611 39155   318  1627
##  [517] 31167 32035 10402 28613 19586 38376 11735 12053 40211 40133 23248  4968
##  [529] 12336 15395 44447 24728 30541 15546 15117 32580 10550 47854 20479 22040
##  [541] 38046  6511 28535 19685 25008 33838 48815 32661 28999 17711 48224  7635
##  [553] 47505  7789 12447  9020 30341  4298 25948 47534  7348  3167 27481   756
##  [565] 13157 28676 15114 36480 28419 39116 14600 27525 24934 12546   887 28360
##  [577]   914 27422 14405 45533 27148 16385 15198 31218 36234 18176 27903  5799
##  [589] 25932 20874 18653 20474 14478 10943  1789 23506 28434 32638  5483  6847
##  [601] 41125 25618 22849 38854  4961 11688 19260  8181  3004  8390 30774 47868
##  [613] 25775 22642 15017  1565 12248 33037 42886 16876  6534 47332 25082 34174
##  [625]  2437 42079 26219 25593 18771 36623 43025 41513 18671 35326 21556  2315
##  [637] 39113  5459 42775 12305  5576 46937 38983 40332  4176 33598 48389 24730
##  [649] 20570 42965 12921 29278 39404   352   384 24163 40818 32763 26283  1191
##  [661] 32496 22660 41885 39396 30006 16707 30933 17832 40197  4588 24135  2304
##  [673] 21771 33442 43901  3277 38179 33979 39150 45174 30536 29444   303 26638
##  [685] 32817 13637 20696 20624 31566  1003 15390 19373  4676 37984 21973 45332
##  [697] 14632 44090 24226  9008 22784  9348 16090 25681 39460 36573 35113 17955
##  [709] 36091 47698 40269 37039 41194  5392 45030  9563 31880 32710 24790  4869
##  [721] 28650  2473 10549 18937 43801 42984 16886 17562 31072 10174 16021 13939
##  [733] 23569  2910 37272 35544 42459 21079 12334 30092 13448  9046 29573 16461
##  [745] 10847 10260 29397  9471 33714 40393  3649 14539 17521 39886 12530 15886
##  [757]   232 28709 12893 16433 37831 47788 37856 14610 11455 13092 31389 48617
##  [769] 10666  8374 44078   674 17674 26234 25855  3211 45669  7103 42083 20665
##  [781] 19743  1762 27251  4425 23162 47086 13344 18420 32385 14407  1607 43391
##  [793] 14303  1611 34337 24825  9813 36839 33778 11855  3271  6506 25392 46339
##  [805] 10557 19807 26367  8155 46129 28623 29417 27955  3302 43315 30526  1847
##  [817] 11276 38756  8412 46971  1201 39234 46355 18535 44088  7111  6764 41939
##  [829] 20024 41452  3366  1448 21542 12083 42481 30549 16087 42868 18970 15483
##  [841] 22312  8260  1308 16951 20439  2002 29708  3288 32984 11159  7242 35716
##  [853] 27644 15611 40703 12467 44010 21336   328 41788 43474 38461 34622 45776
##  [865] 15127  5190 36313 13835 46988  4798 10247  4809 23948 30457 47840  4343
##  [877] 15571 27682 46776 17226 43536 37450 26818 30856 16225 30552   528 24743
##  [889]  9150 46577   663 30867 35730 39959 20475 43097 30481 44237   793 48731
##  [901] 28476 22232 25036 32489 25605 34867 44288 16007  3802  3632 38026 22300
##  [913] 16648 29532 39728 13054 22253  6502  3103 48599 38557 32411 11080 20992
##  [925] 20316 16680 48570 41032 46346 17194 19310  9291 43648 20627 30402 34565
##  [937] 17670 15569 41100 41570 21634 41615 11722  5083 36008 32798  6025 10075
##  [949] 12781 10185 27056 36904 12606 15238 28438 20260 34888 29782  2038  5081
##  [961]  6403  9386 37023 25197  6813 19460 40656 10355 12199 33557 22054 26365
##  [973] 38436 10021 45764 33415 10870  2946 30603  6440 40315 30213 11621 17089
##  [985]  4738   743 39798 46479 43069 17890 11683 44850 35532  8328 10857  9332
##  [997]  1873 29127 23053 47797 20876 38288 47461 41311 27336 32727 22510 19508
## [1009] 37135  6877 10956  7753 27949 30696 11978   970  6668 34648  7748 36194
## [1021] 28865 30247 18907 24581 27049 18421 37679 23770 26438  4288 20303 38269
## [1033]   613 30529 30504 26286 42842 36517  4824 39739 25361  8561 46600 25356
## [1045] 24056 14130 24755 36997 24247 33596 26602 47647 45876 17161 13073 17770
## [1057]  2405 26557 13596 21727 24799  6493  9737 17477 45512 39907  2873 11181
## [1069] 25554 42976  5291 36719 18575 48818 25961 28528 20029 41277 46999 46078
## [1081] 39196 15492  6045 43740  9042  6145 24008 25102 35080  4159 44198 30699
## [1093] 40635 12837  5251 12387 28283 12285  3203 17158 25295  5087 29587  9315
## [1105]  1936 14987 12492 20318 43319 22549 18559  8980  4399 44574 41042 43202
## [1117] 32485 43703  2959  2136 27818  9940 47351 21650 27655 13800 29169  5456
```

```
## [1129]  2021  7432 47155 41580 11274  3527 47022  9293  2173 28362 21587 42279
## [1141] 45408  6207 15786 18608 26198  5301 11918 40281  3493 11860 46697 34454
## [1153]  8314 16123 10729 44396 37223 22340 27772 44587 14174 10597 15656 43972
## [1165] 21811 47918  8062 32864 15850  9013  6140 44349 25882 14964 33497 48808
## [1177]  2539  1055  5751 22262 41892 20390 45075 34250 47784 35546 21846 27771
## [1189] 47915 39750  2859 41063 26024 47782   474  4846 38407 46804 44140 37425
## [1201] 22781 34845 33445 38942 41094 37175 39374   316 26454  8807 33190 16402
## [1213] 43367 18669 22345 19935 18585 14892 30677  9546 39794 27695  6531 17072
## [1225] 47866  6431 12724  1004 20294 36840 33878 40492 33338 16491  4392 39077
## [1237] 29649 13741 35153 48219  7234 40937  8259 39842 38052 27825 27351  4645
## [1249] 31845 40493 42752 37296 27268 31978 30398 44320 48204 39546 17235  9063
## [1261] 11496 12716 45371 18751 28178 39808 33058 14257 15835 14669 28338 16766
## [1273] 36871 40522 18892  9248 37983 34963 11608 42632 25641 40816 16151  2970
## [1285] 47296 34503  5247 23979 15588 20308 16444 24256 37225 10728 41019 11857
## [1297] 48294 10865 25944 41301 23947 40663 26711 18582 46415  4193 48211  4427
## [1309] 26921 24848 37389   854  3831 25288  9098 20517 12366 37625 28509 30463
## [1321] 44780 26380 12426  7868 35421 28866  1158 31322 29049 44106 42434  6239
## [1333] 12614 32414 18217 46313  5932 40322 25788 36684 40859 31410 37807 19126
## [1345] 40645 15396 11204 38297 43968 23631 12976 31805 21512 44818 30735 32674
## [1357] 45081  7354 27649 14664  9901 40528 43544 32940 22170 22094 27294 40116
## [1369] 48684  9349  3059 19416 13217 14686  6078  5525  9321 10610 46152 21502
## [1381] 32677 17458 44404 18468 33439 18186 33602 10358 11117 45098  7807 11989
## [1393]  4356  3855 10491 46624  1746 20497 30922  5103 37695   287  3066 18818
## [1405]  8313  2963 46191  1042  3214  7902 15069 37771 27837 22342 33020 29010
## [1417] 30827 14806 46494   964 35184 47689 23797 38791 33495 13160 10118 31519
## [1429]  2936 37150 19711 15215 48579 26539 14292  2665  7395 48814 35846 39371
## [1441] 21904 12539 46226 10909 15971 26094 20002 22363   260 39285 11521 17394
## [1453]  8864 29515 25306 45311  1274 40830  1723 35255  7188 32757 37491     5
## [1465]  1994  3957 27134 37971 18015  5072 22158 16438 11390 45324 45219   718
## [1477] 11004  3009 45204 33967 41074  8931 27353 18253 20587  9552 44511 27743
## [1489]  9247   543 16907 26802 32668 39661 43979 28601 40564 44334 39182 32463
## [1501]  4547 12411 38190 20026 10241 38048 13239 17427 45580 17988 30318 40811
## [1513]  8509 40441  2421 45585 44730 19833  3178 38577 37699  8024 22928 46741
## [1525] 23642 39214 23331 33857 21719  2581 31513    55 19463 44601 12143 46118
## [1537] 14834  9659   910 39014 39723  7831 19193 13700 29677 11934  9130 11058
## [1549] 24738  1962 47424 42461 47377 39962  3180 32893 35850  3386 33101 40884
## [1561] 14566 23380 26260 16245 28332 26808 18419 36204 28638  2403 20997  5858
## [1573] 11430  2422 24530 38088 26194 28260 45750 30455 31460  3798 24482  4905
## [1585] 40040 40665 33258 23522 12453   706 33382 34148 13584  6402 10478 41330
## [1597] 16294 22248 37465 19442 34224 20256 38603 25479 23939 11602 25113 37857
## [1609] 42623 19076 28947 20644 22132  5872 34792 44653 17156  1278 23897 23568
## [1621] 35301 26439 45599 41721  5264 37476 47710 35321 45789 20230 13452  9411
## [1633] 35576 46082  1841 24038  1272 10330 46781  1714  8464 46977 11464 47429
## [1645] 36854 18466 46139 46871 26753  1604 38443 48109 39956 45898 32614 27470
## [1657] 19698 18059 14759 17138 36933 43962   812 13472 10454 40289 26625 26379
## [1669] 47601 22413 22382  3642 48810 12044  8448 32900 10267 29937 46345 28508
## [1681] 10317 22999  1548 42772 35873 15041  5008 13071 40744  9745 27877  3753
## [1693] 18157 44074  7429 24453 42246 34890  6187 12762  2435 38717 47966  4359
## [1705] 12440 19128 24603 22597 34868 36522 25458 27328  3728 42749 12131  4002
## [1717] 26299  6383 36290 10191 28521 41674 37098  6951 10641 43941 17490   701
## [1729] 46125 22506  7120 24568 43643 30435 36231 13201 40171 45699   113 22558
## [1741]   206 21259 24027 25012 15792 36387 43824 29927 45374 45207  3339 35953
## [1753] 37978 30723 36456  5723 42074 40127  3540 27920 30334 29720 16925 20222
## [1765] 31451 33930   948 37986 28414 26793  7874 18060  1202 29306  6875 12100
```

```
## [1777]  7200 47278 35304  4992 36558 17242 10026  3838 46218 38258 11799 39574
## [1789] 41687 34462  8935  7751 42650 29308  1275 16162 19149 45816   894 29745
## [1801] 16856 47131 34256 33859 19349  3486 19131 20170 40280  7083  3118 24198
## [1813] 27327 30880 26351 45269 30160 48600 13122  4049 32929 32826 12641 26687
## [1825]   132 10391 38698 34912  8385 42579 45148 18135  3018 27366 36182 20921
## [1837]  6397 22430 13893  2960 44157 32307 31772 20786  2961 44756  8870 15179
## [1849]   532 42712  5689 35526 17816 20578 42045 25680 16858 42046 14848  1682
## [1861]  7828 28789 15211  5821 33127 15199 26603 34652 33281  2356  4307 38777
## [1873] 29249 26889 31234 31422 32109 32834  3796  2413 17599 27849 39587  4638
## [1885]  5918 28784 25431 22193 32371 18454 47038 46752 15564 22542 21302  4574
## [1897]   274 18356  7690 28786 20816  8620 34421 23704 28841  3355 43482 32762
## [1909] 24458 16710 30777 13762 15983 46098  8832  1970 14546 47856 21234  9821
## [1921] 30886 47661 39008 29435 29284 38703 17849 39644 11706 24364 37561  5859
## [1933]  9542 29245 40102 33870 31779 31400 17542 35318 37899 10388  2391 11268
## [1945] 23172 23360 31617 46942 10578 12020 12857 21645 45247 27200 13603 31564
## [1957] 11138 34245 17384 48782 21439 34167 32691 25110 17248 43651 20199 28918
## [1969] 26795 33012 25906 36499 44714 43650 20384 48527 33484 16786 18311 28109
## [1981]  5112 14037 20746 10625  2486 21410 46405 35518 20772 45665 17940 27663
## [1993] 28978 47039 18968 34628 41845   309 41567 33671 26044 24892 24302 21040
## [2005] 10250 39987 43390 19061 24466   609 27995 48148 37202  8800 34742 27524
## [2017]  5224 17438  2954  6384 11701 43993 38931 42368 29874 42735 42963 43820
## [2029] 13548 37227 37998 32635 15762 44470  9073 32085 29907 19025 32800 30287
## [2041] 40633 33554 29870 43872 22723 29576  3948 10036 24950 41756 27058 12929
## [2053] 20030 28692 15861  7882 45061 34397 32862 46185 19274 38617 17351 28511
## [2065] 25649 24447 20093 10790  5376 45223 45229 42209  8146 42533 25985  6366
## [2077]  8624 25986 28409 20589 12647 16567 47284 31277   923  2324 26232 29006
## [2089]  1668  8149 29626 17783 36461  3108 10471 38760 47664  8546 21969 12871
## [2101] 47697 39605 13581 28429  2172 35240 43353 25057 14000 29552 36502 34367
## [2113] 32759 33002 32345 33563  4982 39002 37683 16599 26724 22823  1989 21985
## [2125]  1489  3348 26476 32259 28506 24006 44142 46702 43606 28276 10459 43018
## [2137] 16292  6499 40651 13032 26851  2850 33372 19767 37044 22610  6063 19554
## [2149]  7334 15195 27081 19954 25045 34911 19136 24876 11679 30976 38353   108
## [2161] 33138 47127 34232 17707 28035 29466 36209 29461 16149 14936 38408 16936
## [2173] 32112 28842 46304 47809 40243 28378 33378 26892 33004 48011 16729 48788
## [2185] 40451 44360 34617 28292  6919 24160 28427 32178 37190  3887 14018  9265
## [2197] 24674 18978 29410 46364  8527  9311 43091 33096 14851 47508  8845  5543
## [2209]  3847 32781 46842 34656  6882 44807 22107 14948 22721 44219 17443 24024
## [2221] 47251 32882 15398 46696 29493 47496 13888 38181 16375 14318 21619 18119
## [2233] 14946 22584 13313 46891   366  7759 33114 32468 33212 35895 35273 16199
## [2245]  6694  6316  9468 27224 38385 41396 45534 18384 27848 28130 42567 15271
## [2257] 19649 30413  3934 30981 39553 34436 40388 24404  5750 13752 16360  3610
## [2269] 17374 14798 36403 18318 29981 33377 30193 26174 46102 19244 32723 37685
## [2281] 33943  2017 41386   382 41597 13867 43471 37814 18707 47908 13283 27702
## [2293] 41233 31364 21477 25742  3474 12545 28652  4713 27441 30335  6073 34352
## [2305] 33777 34171 21318 42615 46229 22229 17468 24475 14821  7758 45370 47280
## [2317] 34681 32274 10483 19222 31181 27967 48092 36782 40432 34052  1850 39880
## [2329]  7045 30565 32336 25328  1466 22434 37006  4044 17618 42512  7535 21430
## [2341] 11678 27711 26988 42951 10117 33329   178 25122 32410 14736 19772 17344
## [2353]  6337 47081 17016 12151  8762 17863 33288 12603  6874 39489  5869  1333
## [2365] 16340 22525 21329  3643 20870 40805 42553 18926 39642 33985 25501 41061
## [2377]  5096  2469 47785 14775  3446 45851 29335 44413 35152  1334 17043 27864
## [2389] 41213 34081 44796 25873  1674 44847 13998 44599 27595 26258 44681 35408
## [2401] 39031  4098 46777 31132 37632  8830 12018 17354  3523 11111 38326 46302
## [2413] 45742 15632 47264 45035  1452 45357 22855 31280 41684 34894 32328 40041
```

```
## [2425] 10248 18515 33317 44977 18533 37738 23753  4401 32712 15159 27323 34613
## [2437] 31636 42092 15710 48609 39208 24990 35425  3886 12458 46619 29563 24803
## [2449] 46127 13459 18497 36358 33560 43299  8193 16369 28749 15145 34092 36128
## [2461] 34444 24250 24986 24853  1993 32728  9067 13695 30362 20506 17092  5927
## [2473] 14316 34292 10689 13119  5951  5944 13144 43105 34532 20724 18638 43240
## [2485] 41299 13971 36345 17744  4671 40939 41812  6052 10307 44683  9913 29203
## [2497]  5119 14752 48418 21459 16015 47096 28969 36792 36953 21461 44186   354
## [2509] 17465 31639 23179 31999 17796 13723 34453  5583 11277 31247 25817  1245
## [2521] 24845  7153 14830 22907 31939 35616 27227 28585 17471  8373 33937 10726
## [2533] 27707 40730  1803  2933 29443 37063  4134 47897 12528  4224 45919 46733
## [2545]  3960 21481    57 21670 25118  1216 36015 33685 33437  3200  3420  8094
## [2557]  8050 32363 28152 18469 18353  6004 16668  9765 46710 43396 20733  5879
## [2569]  7920  7272 25366 12133 44761 38735  9399 40540 23078  2139 44158 33135
## [2581] 29604 36870 45711 24684  1660 41538 32500 40623  9921  2747 16049 11379
## [2593] 32194 47867  3677 25997 26245 34109 10946 24758 19672 30351 31375 11924
## [2605] 41750 34107 11515 40899 15674  5013 37157 18048 33686 28022 25349  8760
## [2617] 19705 48081 20349 44783  7918 26500 30019 20945 47189 34072 24897 38284
## [2629]  6236  8847 17945 24409  8797 11922 16177  7599 29312 26155   126 23652
## [2641] 18769 11422 10722 27861 30164 27035  6774 44301 26707 39584 10151 32941
## [2653] 46670 36543 19488 10163 37186  2178 43130  9910 35426 28168 27860   810
## [2665]  7370 42908 21429 41116 43090  5360 13473 21008  9508 35615 17981 15096
## [2677] 10879 41060 30764 28932 28844 19650 17683 47445  3374  9453 18021 11374
## [2689] 41652 44611 47004 35537 23185 10884 30958 47036 16897 34263 12144 11085
## [2701] 16801 20988 24634 30091 28251 43907  5516  7167 47414 39590 34084 38536
## [2713] 38427 48209 28727 12832 21607  4606 25941 45079 12297 27487 37861  3433
## [2725] 18767 23812  1755 11974 23699  7382 37101 47311 41441 10691 44209 18714
## [2737] 15349 15562 45203 45000 38234  4167 27181 34775 31465 27221 17360 45963
## [2749] 43805 23228  3758 10747 33330 26375   486 32624 45853 36986 45264 36743
## [2761]  2370 47885  2458 38828 22898 42365 22900  9452 16016  4631  8809 32160
## [2773]  7384 32302 11338 38377 12615 10634 38487 42220 34869 37791 18188 31797
## [2785]  2881 48425 26148 42129 36629  9041  8275 13607 44322 27205 41235 46855
## [2797] 37565 45278 11698 30262 38496 40836  9636 42841 47777 48098 26770 31936
## [2809] 17735 10911 32012 13994  6634 26838  1573 48223 28273 25148 46517 11306
## [2821] 27276  2362 24597 32971 13199 29084 19230  1872 37468 44160 11132 19929
## [2833]  8010  5690  8487 30876 28622 46530 20641 31022 43951 27245 22163 10745
## [2845] 44475 15109  5634 48811 36256 44340 21115 17698  4680 12118 22259 26520
## [2857] 30059 20324   947  8395  2943  7944 31378 15742 44794 10362 47065 27583
## [2869] 13300 14615 26029   563 40559 19494 16457  2288 35987 11492 40284 11120
## [2881] 43772 34519  9968 42219 12120 37171  9782 18107  2026 40193 40776 21962
## [2893] 14983 24310 40143 24325 40659 21612  1616 18229  4991 32813 32301  7953
## [2905]  6771 25695  3526 21845  5608 23338 42799 43407  9286 44862 23539  2255
## [2917]  3221 19190 31846 15001 23984 15837 12793 26780 27880 19371  6663  2623
## [2929] 25327 30172 31695 21942 12438 24545 32037 14786 18080 39076 22502 48218
## [2941] 33855  6723 27957 28224 36421 48453  9605  7241   932 48606 15481  1532
## [2953] 37929 43163  4815   995 39822  4227  8794 19295  5273 11487  4319 24275
## [2965] 13052 36633 23575 29217 19028 19492  6230 37612  1151  1107 37367 28641
## [2977] 18257 36215 27726 40266 24231 12480  3756 21486 30870 44811 32795 33923
## [2989] 43150 27313  9613 15839 21395 26567 16804 39175 40385 32582 47312 40717
## [3001] 23980 48489 34407  3231  3817 29162 27573 22350  4489 38489 11843  4749
## [3013]  1434 27728 15226 47237 28628 40774 20754 10141 43538  9018 31256 16896
## [3025] 16691 22047 28615 37236 20862 21642 21893 25674  2808 28127 41164 18819
## [3037] 19551  3268 14609 17289 10282  6970 35429 24075 29247 22867  1783  5323
## [3049]  4457 22848 31761  9544 38426 14005 46965  3686 11982 14603 42923 48226
## [3061] 11945 31329 23541 33843 18118 20274 19958 28574 33622 38018 43735 11239
```

```
## [3073] 31049 36691  7446 14765 20361 33450   804 41022 44851 38347   369 33980
## [3085] 31437  2302  4591 23862  3641 20001 31710 24425 24606 14993 45032 44721
## [3097]  8885 32454 44445 13370  6351  1907 31554 31731 21393 30779  4074 37123
## [3109] 17888 46955 23208 14975  9270 32711 10825 35190 31084 39288 23940 24789
## [3121] 37756 48778 38833  5305 48622 31012 33206 41023 18779 10812 39079 37110
## [3133] 37951  2885 27846 10200 24547 21513   637 10582  5817 48562 46188 40136
## [3145]  3864  8625 19179 33787 19584 40770   188 26855 27950 27872 36978 23620
## [3157] 35252 21471 44809 23121 18859 46398 17586 26321 42392 16912  5604 10210
## [3169]  7888 32188  2183 31819 41248 16086 48529 45228  8978 38859 40473 20244
## [3181] 29468 36279 27963 34730 21399 28370 46967 33773 11594 47538 23118 47192
## [3193] 35229 42127 43294 40449 17383 46719 43297 11453 27334 30693 12310 42740
## [3205] 41983 20639 15213 17363 30546 34035 14516 23385 45547  6433 43793   744
## [3217] 13664  9985 13720 27263  3562 33551 40039 43613  9345 16666 43978 37419
## [3229] 36523 36767 42063 43702  8334 36428 47268 45710 32352 22398  7914 10892
## [3241]  1108 39944 28700 39952 16627 18349  6339 14401  7078   883 17722 45006
## [3253] 40118 17498 16213 40005 46446 25555 43278 16913  6447 36414  1729 26961
## [3265]  9352  7435 35908  2585 13345 43016 32055 18397 12352 37496 38435 43669
## [3277] 38293  7205 46094 32549 47372 31655 21260 16774 20833  8447 37605 43975
## [3289]  5127 39274 22069  2353 19617  4372 36670   538 17029  4659 20642 39896
## [3301] 34591  5029 14645  1152  5331 46666 14097     7 27983 14766  5060 18522
## [3313] 23233 11466  9506 19610 41594 39473 43890 20414 36510 36495 15743 25677
## [3325] 37583 40837   506  7186 10279 39695  5334 12058  7572 17173 29747 24479
## [3337] 14641 40336 17787 10538 24426 10558 39355  1099  8563 16487 31146  7932
## [3349]  3357 27169 10768 36317 48060 30458 30280 37873 16967 14332 36703  8595
## [3361] 40104 42188 27586 20745 15679  7151 29599  7450 12748 11428 44184 25646
## [3373] 32343 20053 19323 12427 31154 13135 31367 43184  2398 21629 40772 32007
## [3385]  7408 43030 31911 42457 12650 23057 24749  2825   250  2401 35901   324
## [3397]   279 41104 39736 31189 26590 35474 18325 48203  2496 30819  3719 20757
## [3409]  7773 35365 12631  1098 27435 46210 19106 40967 30548 33365 10096 27036
## [3421]  5871 17629 32576 28411 19214  7525 33708 43610 29918 16065  7022 40586
## [3433] 46089 11954  9354 27574 30913  2610 10783 24058 33129 32568 29109 10702
## [3445] 41879 40916 20483 41335 45472 44503 26046 41601 13338 38612 16972 32357
## [3457] 29300 13258 36099 26005 17200 41365 34376 40177 34330 39385 40764 44559
## [3469]  3919 11600 38562 38806 16347 47931 31390 45707 32123 30633 27098 42659
## [3481] 30929 20909 20778 18957 18900 19801 16216 12196 39615 34522 10561 17027
## [3493] 41680  2112 13585 34660 30407 19642 46541 15823 17965 27859 42100 38899
## [3505] 24164 15472 46128 46618 27085 16884  3233  8985 10049 33485 12302 48316
## [3517]  9954 48464 35399 23100  1593 21894 48575 14302 43195 18376  7800  4782
## [3529] 10395 41534  2010 45953  6331 29024  2040 48325 28253 26021 17573 12866
## [3541] 43666 12822 28379 47993 12094 17705 43276 13158  5850 17941 18032   825
## [3553] 32964 33809 23405  4617 30061 27396  1223 34608 14950  6408 36441 35788
## [3565] 25316 36322 25989 47704 38773 37833  3856 47444 32356 21729 40959 24710
## [3577] 23698 45501 40094 13151 14613 30065 19748 38961 12727 27536 14429 26993
## [3589] 14472 42348 44802 34393 23520  1897 26593 15299 26609 25104 12663 40873
## [3601] 31213 11899 35585 22754   100 35493 26763 15154 21594 14552 34445 36104
## [3613] 13753 41487 14145 28680  9280  3734  6683 48420 44668 27321  4354 19103
## [3625] 46044 35801 19478 46795 11437 21232 28305 46879 29071 35790 44642 21427
## [3637]  7770 30563  7441 21268 21540 23346  1427 12176 27119 27580 25777 48819
## [3649] 31486 18749 45108 35344 27331 11130  1901 23854  7822  6570 17359 45220
## [3661] 18239 27249  9389  7376 41659 45634 44190 42671 33694 26435 27021 30689
## [3673] 26350 12926 44427 16979 18739 41952  7566 42968 47812 39051 42096 33654
## [3685] 11692 22037 40378  7002    69 35354 22148 48638  2975 18975 11574 41555
## [3697] 27117 26175 41548 22217 46952  9378 10361  9722 18350 24997 16361 24070
## [3709] 44006 19062 20019 24627 41153 13100 14156 44514  2622  6113 47091 17435
```

```
## [3721] 41449 39708 41557 31118 10688 29432  5240 17959 22946  9227 39068 16252
## [3733] 15296  7279 44704  9808 23576  9865 35551 31815  6214 48019 43567 32339
## [3745] 17691 30420 30931 48566 38268 14099 47361 22541 12683   823 19470  8981
## [3757] 35338 22800 16243 26091 31582 35831 30716  9319 15740 12233  5179 44458
## [3769] 12043 21125 36286 23047 26928 36222 10723   134 31222  5284 27402  1193
## [3781] 45005 35440 11693 31618 37850 34499 44520 45978 32095  1833 35489  4186
## [3793]  9416 20174 11158  7127  5307 35000 12399 41630 28690 48629 38494 22096
## [3805] 16488 25450  1131 38469 32476  1651 19948 18645 25252 24013 18647 43188
## [3817] 13215 41869 29043 47889 10253   432 23921 17782 31984 37799 38876 30993
## [3829] 35294 37452 41865 15669 26156  2787 45181 19986 47078 45771 19476 24045
## [3841]  9110 38922 44065  3787  7206 19191 23269 28116 15779 32134 36640 17531
## [3853] 10461  3944 34611 19162 30925 48063  5832 24091 46498 33300 22726  3788
## [3865]  9268 47072  2383 38862 37007  1703 22179  8759 48045 42402 30621 23545
## [3877] 33994 42808 34090 10369 39833  2284 44570   576 25413 34823 18084 18783
## [3889] 34029 18600 44769  7214 24463 46691 17424  6987 22483 16749  4338 26061
## [3901] 38163 16440 46198 47679 38246 46805 13434 42670 28108  9717   630  9137
## [3913] 16102 37457 40896 34012 14371 13037 11425  7245 47739 19534 11016 18626
## [3925] 35991 47395 34536 33014 38775 15454 46307 32881  5674 36569 25325 19666
## [3937] 47552  5731 27594 14578  5342 31111 37713 30938 44055  4346  3012 47541
## [3949] 26571  3187 22070 11973  8775 48403  9271  2499 34801 28948  2434 37281
## [3961] 28648 36196 10222 40402 29474 38937  5295 25919 15715   435 24689 30839
## [3973] 29124 33646  7041 34684 24345  1752  6750 46569 29110 27077 25556 25170
## [3985] 10475 18145 16191  5928 13093 40855 46362 30887 32946 43364 14199 26337
## [3997] 40907 34667  7171 29585 28089 35083  6023 21306 13944 40495 22341 19520
## [4009] 32564 48227 38298  7197 37837 38431 27211 33627 42661  9277 13522   905
## [4021] 24359 41551 38694 47042 30188 21475 40674 34068  1014  5673 10093 13690
## [4033] 20464 44338 15766   438   218 35303 37552 33786 42233 28060 14198 36341
## [4045] 23866 48126 18679 21957 14328 16662 41205  1612 34614 43615 44705 41965
## [4057] 48584 38669 44892 37876  9149   360 45591  8294   319 21107  3647 32453
## [4069] 17112 18605 47744 21454  6666 26443  2908  8286   535  4195 32684 31445
## [4081]   599 36676  5373  8590 21070 10412 44331 13662 41720 15475 41920 42061
## [4093]  9215 42609 45211 14040  1710 14465  1892 16157 14408 24682  6586 25216
## [4105] 17419 31773  4450 36957  5606 43933 12132 15086  6784 39086 19673 31882
## [4117] 43484 45912   475 30705 24992 40355 30320 10129  1846  6267 16051 42518
## [4129] 21592 27379  6162 41429 23728 37493  9582  1318  8562 13031 45698 32139
## [4141] 45854 36666 48670 23067 46551 21805 37431 10754 48309  6653  8944  4477
## [4153] 32024 26469 23337 32011 12780 19507  2170  6189 38243 24643 40788  3307
## [4165] 11551 36501 36479 47032 41778  5864 45399  6608 19737 36013  2388 22722
## [4177] 19177 23075 35136  3269 37437 21412 16274 41007 47501 28417  1571 36285
## [4189] 45849 41844 41808 28794 34513 21417 22876 19483 30875 19597 13577  7704
## [4201] 22339  1976 25801  7377 47100 13836 46656  9883 22192  8941 36658  6878
## [4213] 25935 45836  4932 42283 33592 33482  2817  2123  2719 29503 36307 12096
## [4225]  8243 29880 34825   280 47121 45450 45900 17701  9798  4212 17958  1702
## [4237] 46194 26427 11575 25806 47650 28860 38397 36701  6965 42018 37580 13374
## [4249]  7302 20045 34505 13403 19067 18577 17293   907 37268  1819 43834 18715
## [4261]  6030 25265  5254 26894 12064 39329 43034 36589  2738 31302 21983 27974
## [4273] 16289 23441 22287 19784 10266 36656 31141  5218 43964 45492  6569  7513
## [4285] 42855  9872 46431  9911  9539 48572 47871 32381 17175 31557 46966 47503
## [4297] 43931  5192 45362 24505 43647 36607 35097 29929  2712 24567 25468 12956
## [4309] 40209 29104 41994  4939 11595 19413 44655 32065 21116  4317 43284 25620
## [4321] 20138 46429 12040  3075 29728 24578 36081 46806 47262 40832 12538 11301
## [4333] 21880 44298 21730 29438 37672  3959  5698 37626  1169 29749 36347  7971
## [4345] 38465 24619 28881 22256 23719  9256 27031 19974  6931 28912 32820 24948
## [4357] 12595 35597 30370 36084 29996 17388 41081 12430 25271 31584 18830 17922
```

```
## [4369] 29701 30421 21283  8964 42144 23729 26359 36137 32525 41763 48136 13268
## [4381]  9906 13582 42828 10606 21575 30219 42633 14823 29147 10662 46758  6707
## [4393] 18406 35719 22539 28626 17680 40338 35986 48634  4365  5056 30890  4449
## [4405] 38894 24112  6844  6840 17224 37928 26077 11469 29458 11038 15272  3786
## [4417] 20685  9015 21825 48805 45191  5123 26745 35209 13066 26679 32621 15254
## [4429] 29731 34612 15331 24518 35832 43446 23278 39486 38804 20933  8977 27798
## [4441] 29902 25617 16796  5536 14932 33734 19363 19690 15943 10143  5399  6610
## [4453] 13850  3220 36023 14368  7642 21017 17473 33192 27560 12126  3331 45862
## [4465] 30851 45939 31537 43433 46813 36199 21377 43387 39134 37418  6642  7393
## [4477]  2567  8125 32467  3146 27486 24237 33421 41575  8345 40139 21677 38090
## [4489] 24750  8041  6934 45950 24561  7958 16120 27061 13936 36664 18778 23525
## [4501] 43517  4151 17593 26668 32165  6862 27513 34554 22071 42871 16724 33806
## [4513]  4300 15843  8872  2023 25955  4146 19299 11873 19590 46306 33109 28685
## [4525] 13181 39424 39645 38105 25068 36826  4594 15947  3896 11789 24152 33171
## [4537] 27282 46962 13730 21438 24678 35118 38796 22403 29075 33283 35762 19745
## [4549] 18328  1825 48692 48185 36443  3574 28834 39241 27882 32508  6446 37805
## [4561] 42627  4841 23327  5455 11309 27289  3294 12889 17505 42734 10080 10803
## [4573]  9922 17307  6268  1750 14128 15648 10724 36140 31288 37637 44417 17669
## [4585] 21909  9484 24846 22605 11766 32126 36277 22326 18509 46489 19992 17751
## [4597] 13063 43909 35371 24701 14411 45348 21533 21067  7872 35028 17927 38273
## [4609] 27369 27240 43678 20332 12319 36885  2440 41830 17091 31217 19629 12378
## [4621] 12121  2880  6967  3698 15940 24759  5086 42237 20427  6899 34517 23589
## [4633]  9335  4410  8023 48711 10770  4353 47794 42019 48402 10232 16494  4172
## [4645]  4006 11709 20969 23087 19814  9838 38404 25952 38990 25261 47168 15214
## [4657] 27991 40098 10671 19361 12073 41950 37519  5849 13525 12768 32426 31956
## [4669] 23962 28968 24307 29346 29943 26968 15858 21444 30184 36252 29674 28961
## [4681] 41703 14336 32015 33412 19350  9551 31591  5306 30208 23595 35921 19694
## [4693] 27997 20739 42222  6129 43583 18777  9526 11541  9890  1661 15633  9369
## [4705] 32482 47270 29364  2128 36508    49  9827  2920 22929  2117 42002 37125
## [4717] 16425 26523 37515  6897  7280 12685 23657   106 17503 22785 27675 44003
## [4729] 15007  4861 35313  4734 31307 11088    50 13502 31717 29652 32617 44312
## [4741] 35760 43773  4702  3480 33099  8731 14136 14995 14241 30056 39039 47599
## [4753]   615 17083  7885 28165 14801 33825  9814 45068 46256  4648 39994 44379
## [4765] 48498  8309 47527 12686 37920 14444 12312 42826 40608 15888 47898 46623
## [4777] 41900 37473 19376 28550 22550 24958 10889 32314 12200 14260 38634  5617
## [4789]  5899 47513  7814 16882 21483 19514  6192  7163 15798 40187 38872 24511
## [4801] 29101 23915 43558 31485 39831 20434 30533 20882 19010 12290 47462  7392
## [4813]  2463 32221 12262  5915 20114 16914 30469  4538 20994 32131 35439 13486
## [4825] 46823 44341 16551 15570 17160 17946 43642  8557 37466  6798 40802 13365
## [4837] 15702   933 12353  3716 12592 48275 42853 15544  3921 39088 29837 33789
## [4849] 25747 26327  9407  2784 30294 26766 27585    10 39781  2148 18895 44716
## [4861] 29271 39517 12664 41883 39976 27987 30401  7486 18775 29304 41355  7960
## [4873]  9308 26911 47238 42672 32898  6911 16221  1519 38409 35012 42577 28571
## [4885] 32970 31383 28046 35120 37502 39624 17314 32465 20149  2898 35423 15473
## [4897] 18362 25021 16685  9935  8789 27215 47216 26279 47665 41633 24288 15760
## [4909] 39219 46970 47140  7276 28655 31812 39617 37105 38918  2242  6777 20942
## [4921] 31706 11294 40654 24442 17906 37443 37328   705 30899 27425 39813  2056
## [4933] 24571  5379 17279 23956 30700  9398  1155 14308 19471 20337 21927 23602
## [4945] 12527 10352 18066  1440 38649 36425 20006 31887  3829 44606  8511 24942
## [4957] 22453 33449  8201  7453  8776 18205 48433 46261 10584 40524  4622 33371
## [4969] 11804 38230  1491 38982 31540 21653 30885 26607   746 34041 45787 48166
## [4981]  7916 28479 15510 33990 18067 12250 12594 27962 28126 16787 26744 26412
## [4993] 20268 31534 48709 37076 16250  3743 42899 17331
##
```

```
## $class
## [1] "data.frame"
```

```
# Remove unwanted columns
dd <- dd[, !(names(dd) %in% c("fnlwgt", "income_integer", "income"))]
```

2. Initial Exploration

```
# DATA VISUALIZATION
attach(dd)
```

```
## The following objects are masked from dd (pos = 5):
##
##     age, cap_gain, cap_loss, edu_num, hours_week, marital,
##     native_country, occupation, race, relationship, sex, workclass
```

```
names(dd)
```

```
##  [1] "age"            "workclass"      "edu_num"        "marital"
##  [5] "occupation"     "relationship"   "race"           "sex"
##  [9] "cap_gain"       "cap_loss"       "hours_week"     "native_country"
```

```
# Check classes of variables
sapply(dd, class)
```

```
##            age      workclass        edu_num        marital     occupation
##      "integer"       "factor"      "integer"       "factor"       "factor"
##   relationship           race            sex       cap_gain       cap_loss
##       "factor"       "factor"       "factor"      "integer"      "integer"
##     hours_week native_country
##      "integer"       "factor"
```

3. Selection of Continuous Variables

```
# List of numeric variables
numeriques <- which(sapply(dd, is.numeric))
numeriques
```

```
##        age    edu_num   cap_gain   cap_loss hours_week
##          1          3          9         10         11
```

```
# Create data frame with continuous variables
dcon <- dd[, numeriques]
sapply(dcon, class)
```

```
##        age    edu_num   cap_gain   cap_loss hours_week
##  "integer"  "integer"  "integer"  "integer"  "integer"
```

```
# Note: If there were missing data, one should handle it before PCA
```

4. Principal Component Analysis on Continuous Variables

```
pc1 <- prcomp(dcon, scale = TRUE)
class(pc1)
```

```
## [1] "prcomp"
```

```
attributes(pc1)
```

```
## $names
## [1] "sdev"     "rotation" "center"   "scale"    "x"
##
## $class
## [1] "prcomp"
```

```
print(pc1)
```

```
## Standard deviations (1, .., p=5):
## [1] 1.1359158 1.0158971 0.9921132 0.9390746 0.9008322
##
## Rotation (n x k) = (5 x 5):
##                   PC1         PC2        PC3        PC4        PC5
## age        -0.3704399  0.23545900 -0.7962627  0.1200863  0.3985955
## edu_num    -0.5400647 -0.11720509  0.4760640 -0.3076046  0.6110119
## cap_gain   -0.4743027 -0.50977180 -0.2562210 -0.4152172 -0.5264172
## cap_loss   -0.2433951  0.81837612  0.1274668 -0.3697389 -0.3436054
## hours_week -0.5356306  0.03486056  0.2396494  0.7627886 -0.2694555
```

```
str(pc1)
```

```
## List of 5
##  $ sdev    : num [1:5] 1.136 1.016 0.992 0.939 0.901
##  $ rotation: num [1:5, 1:5] -0.37 -0.54 -0.474 -0.243 -0.536 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:5] "age" "edu_num" "cap_gain" "cap_loss" ...
##   .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
##  $ center  : Named num [1:5] 38.5 10.1 1023.7 91.3 40.2
##   ..- attr(*, "names")= chr [1:5] "age" "edu_num" "cap_gain" "cap_loss" ...
##  $ scale   : Named num [1:5] 13.84 2.56 7166.66 412.95 12.56
##   ..- attr(*, "names")= chr [1:5] "age" "edu_num" "cap_gain" "cap_loss" ...
##  $ x       : num [1:5000, 1:5] -0.053 0.794 0.714 0.154 0.954 ...
##   ..- attr(*, "dimnames")=List of 2
##   .. ..$ : chr [1:5000] "2986" "29925" "29710" "37529" ...
##   .. ..$ : chr [1:5] "PC1" "PC2" "PC3" "PC4" ...
##  - attr(*, "class")= chr "prcomp"
```

5. Inertia and Variance Percentage

```
# Calculate eigenvalues and inertia
pc1$sdev
```

```
## [1] 1.1359158 1.0158971 0.9921132 0.9390746 0.9008322
```

```
inerProj <- pc1$sdev^2
inerProj
```

```
## [1] 1.2903047 1.0320468 0.9842886 0.8818611 0.8114987
```

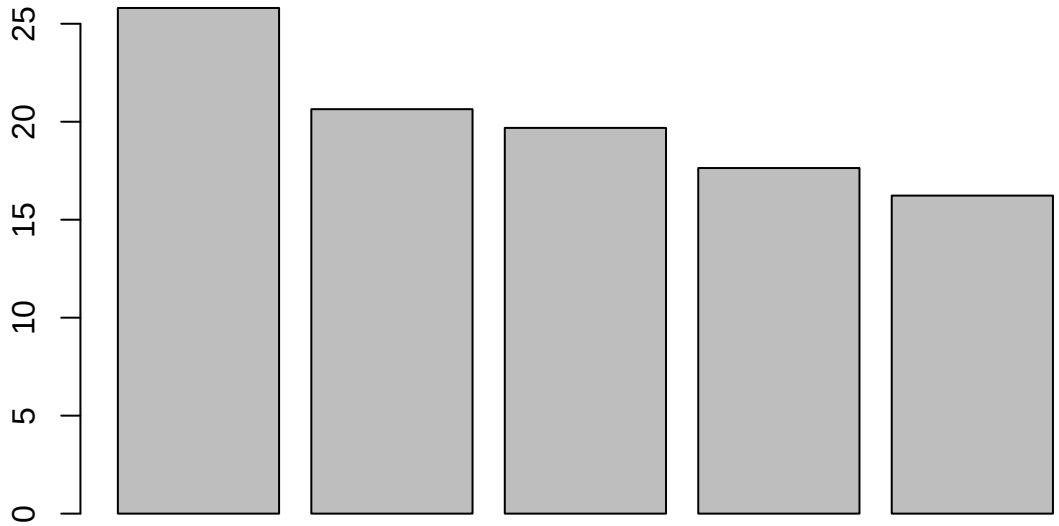
```
totalIner <- sum(inerProj)
totalIner
```

```
## [1] 5
```

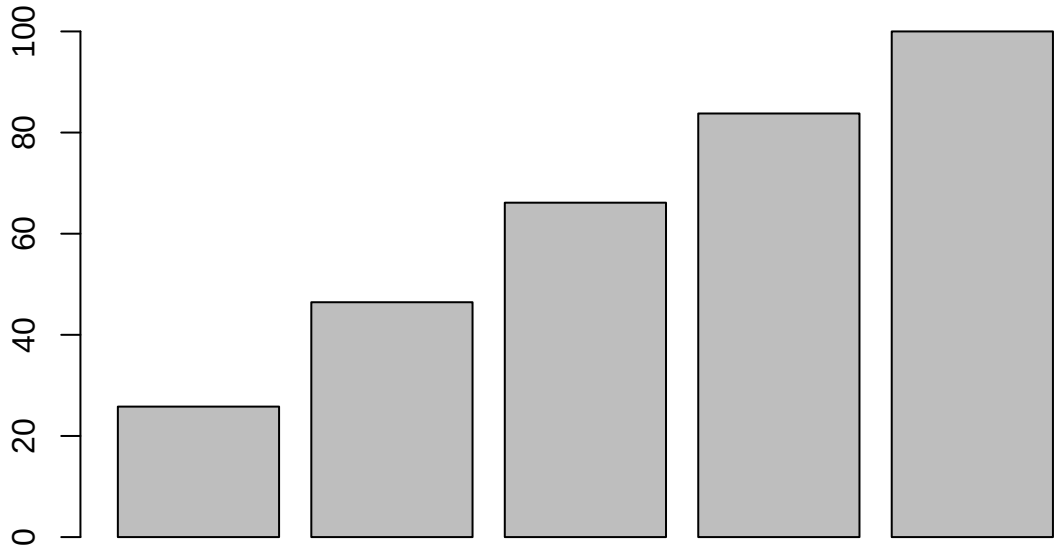
```
pinerEix <- 100 * inerProj / totalIner
pinerEix
```

```
## [1] 25.80609 20.64094 19.68577 17.63722 16.22997
```

```
# Plot individual explained variance
barplot(pinerEix)
```




```
# Plot cumulative explained variance
barplot(100 * cumsum(pc1$sdev[1:dim(dcon)[2]]^2) / dim(dcon)[2])
```




```
percInerAccum <- 100 * cumsum(pc1$sdev[1:dim(dcon)[2]]^2) / dim(dcon)[2]
percInerAccum
```

```
## [1]  25.80609  46.44703  66.13280  83.77003 100.00000
```

6. Selection of Significant Dimensions

```
# Choose first 4 components (~80% total variance)
nd <- 4
print(pc1)
```

```
## Standard deviations (1, .., p=5):
## [1] 1.1359158 1.0158971 0.9921132 0.9390746 0.9008322
##
## Rotation (n x k) = (5 x 5):
##                   PC1         PC2        PC3        PC4        PC5
## age        -0.3704399  0.23545900 -0.7962627  0.1200863  0.3985955
## edu_num    -0.5400647 -0.11720509  0.4760640 -0.3076046  0.6110119
## cap_gain   -0.4743027 -0.50977180 -0.2562210 -0.4152172 -0.5264172
## cap_loss   -0.2433951  0.81837612  0.1274668 -0.3697389 -0.3436054
## hours_week -0.5356306  0.03486056  0.2396494  0.7627886 -0.2694555
```

```
attributes(pc1)
```

```
## $names
## [1] "sdev"     "rotation" "center"   "scale"    "x"
```

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

```
pc1$rotation
```

```
##                  PC1         PC2        PC3        PC4        PC5
## age        -0.3704399  0.23545900 -0.7962627  0.1200863  0.3985955
## edu_num    -0.5400647 -0.11720509  0.4760640 -0.3076046  0.6110119
## cap_gain   -0.4743027 -0.50977180 -0.2562210 -0.4152172 -0.5264172
## cap_loss   -0.2433951  0.81837612  0.1274668 -0.3697389 -0.3436054
## hours_week -0.5356306  0.03486056  0.2396494  0.7627886 -0.2694555
```

7. Projections in the New Space

```
View(pc1$x)
dim(pc1$x)
```

```
## [1] 5000    5
```

```
dim(dcon)
```

```
## [1] 5000    5
```

```
dcon[2000,]
```

```
##       age edu_num cap_gain cap_loss hours_week
## 33671  42       9        0        0         40
```

```
pc1$x[2000,]
```

```
##           PC1           PC2           PC3           PC4           PC5
##  0.2590887659 -0.0004006728 -0.3945332354  0.2855167304  0.0051248000
```

```
# Store projections
Psi <- pc1$x[, 1:nd]
dim(Psi)
```

```
## [1] 5000    4
```

```
# Example access to projection of a single observation
Psi[2000,]
```

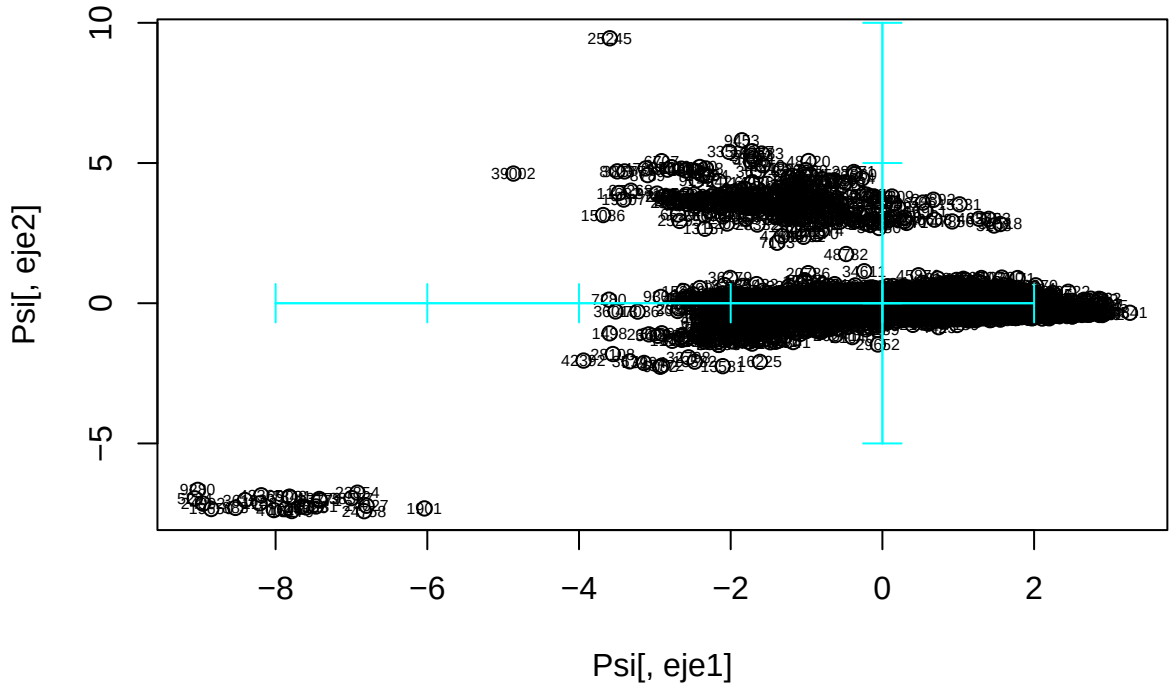
```
##           PC1           PC2           PC3           PC4
##  0.2590887659 -0.0004006728 -0.3945332354  0.2855167304
```

```
# Labels and indices
iden <- row.names(dcon)
etiq <- names(dcon)
ze <- rep(0, length(etiq))
```

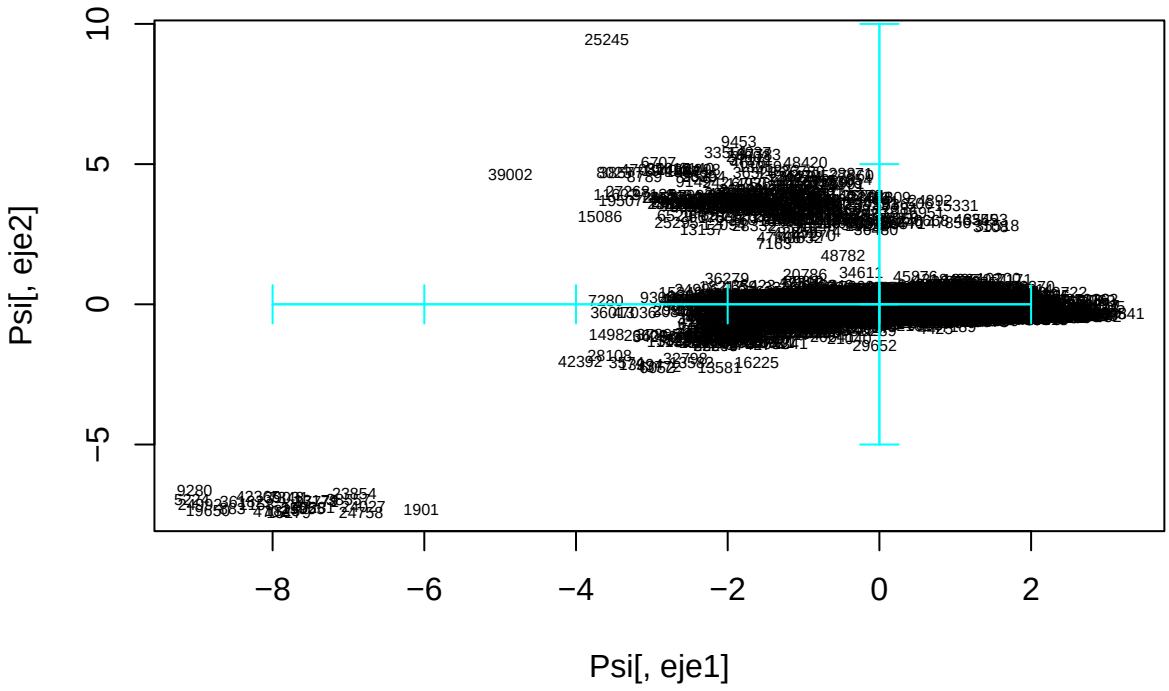
8. Plotting Individuals in the First Plane

```
eje1 <- 1
eje2 <- 2

# Plot with labels
```

```
plot(Psi[, eje1], Psi[, eje2])
text(Psi[, eje1], Psi[, eje2], labels = iden, cex = 0.5)
axis(side = 1, pos = 0, labels = FALSE, col = "cyan")
axis(side = 3, pos = 0, labels = FALSE, col = "cyan")
axis(side = 2, pos = 0, labels = FALSE, col = "cyan")
axis(side = 4, pos = 0, labels = FALSE, col = "cyan")
```

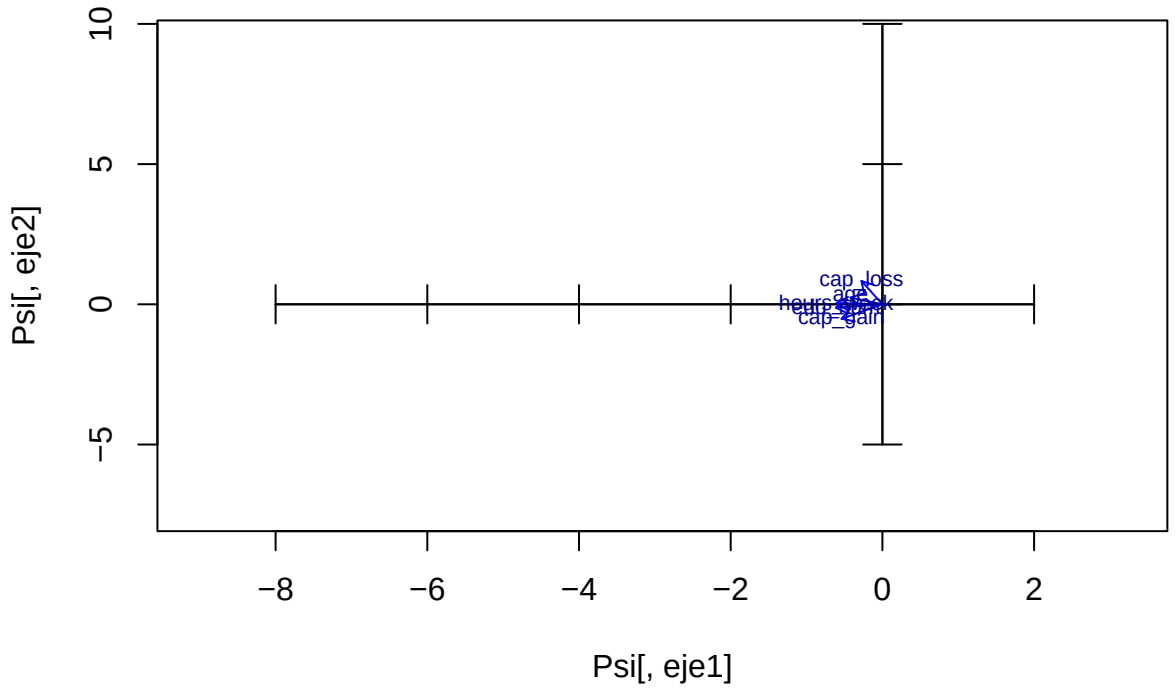



```
# Empty plot with only text
plot(Psi[, eje1], Psi[, eje2], type = "n")
text(Psi[, eje1], Psi[, eje2], labels = iden, cex = 0.5)
axis(side = 1, pos = 0, labels = FALSE, col = "cyan")
axis(side = 3, pos = 0, labels = FALSE, col = "cyan")
axis(side = 2, pos = 0, labels = FALSE, col = "cyan")
axis(side = 4, pos = 0, labels = FALSE, col = "cyan")
```

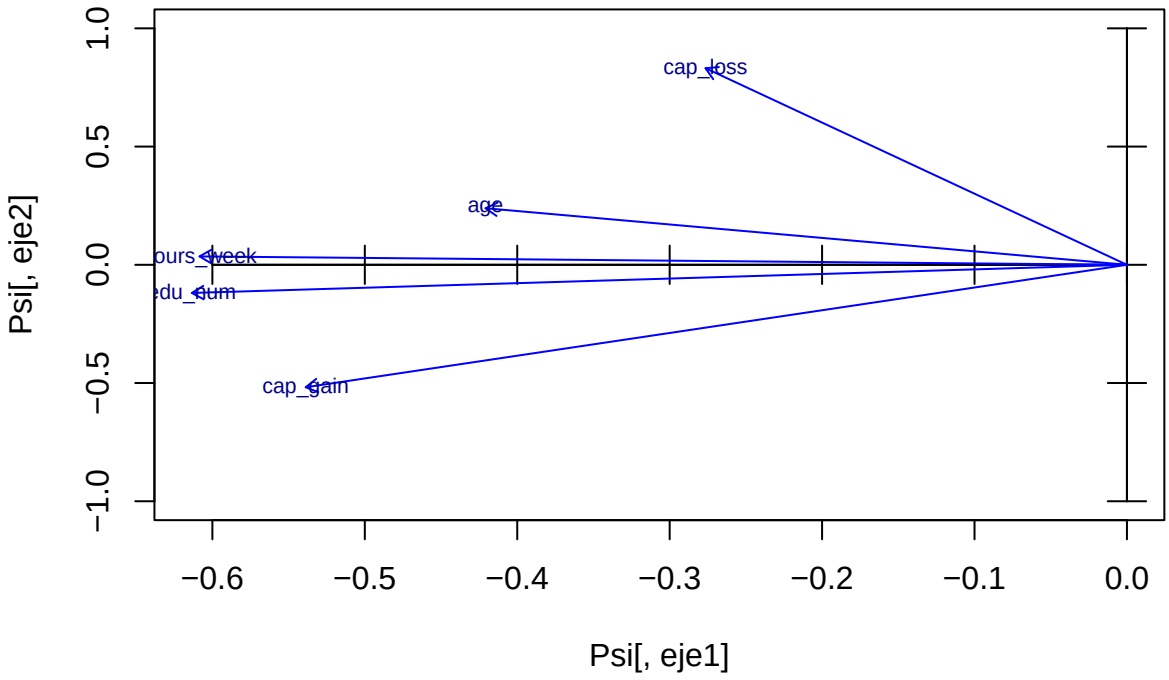


9. Projection of Continuous Variables on the Principal Plane

```
Phi <- cor(dcon, Psi)
View(Phi)
X <- Phi[, eje1]
Y <- Phi[, eje2]

plot(Psi[, eje1], Psi[, eje2], type = "n")
axis(side = 1, pos = 0, labels = FALSE)
axis(side = 3, pos = 0, labels = FALSE)
axis(side = 2, pos = 0, labels = FALSE)
axis(side = 4, pos = 0, labels = FALSE)
arrows(ze, ze, X, Y, length = 0.07, col = "blue")
text(X, Y, labels = etiq, col = "darkblue", cex = 0.7)
```

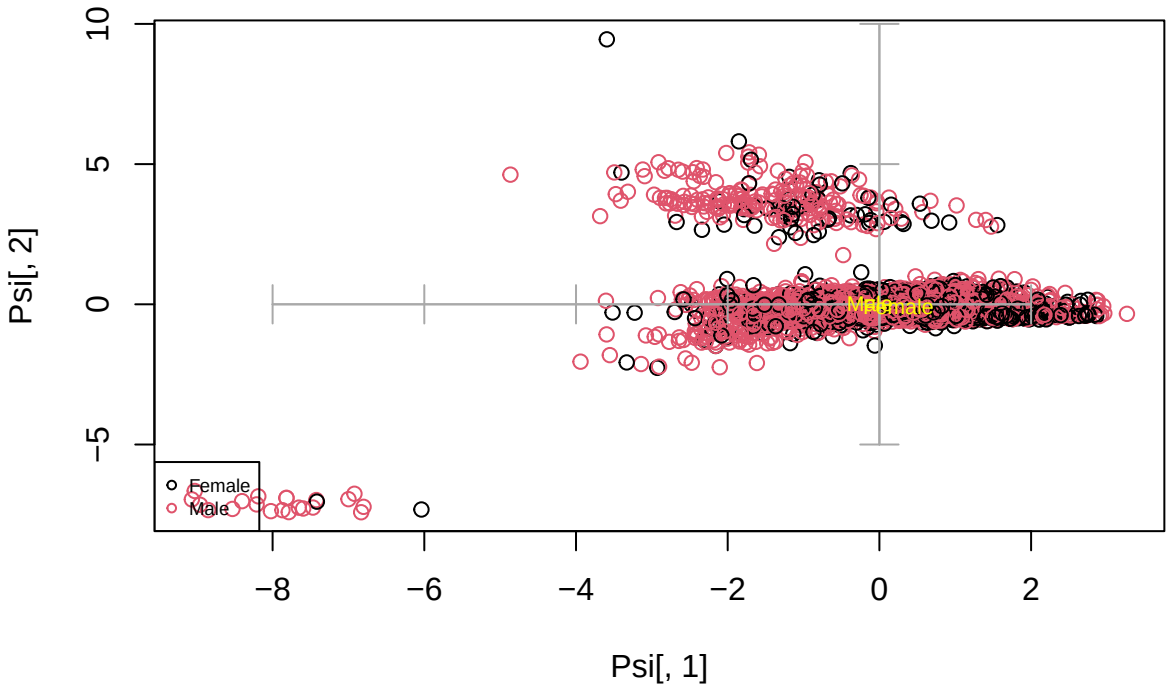



```
# Zoom using variable-based limits
plot(Psi[, eje1], Psi[, eje2], type = "n",
     xlim = c(min(X, 0), max(X, 0)),
     ylim = c(-1, 1))
axis(side = 1, pos = 0, labels = FALSE)
axis(side = 3, pos = 0, labels = FALSE)
axis(side = 2, pos = 0, labels = FALSE)
axis(side = 4, pos = 0, labels = FALSE)
arrows(ze, ze, X, Y, length = 0.07, col = "blue")
text(X, Y, labels = etiq, col = "darkblue", cex = 0.7)
```



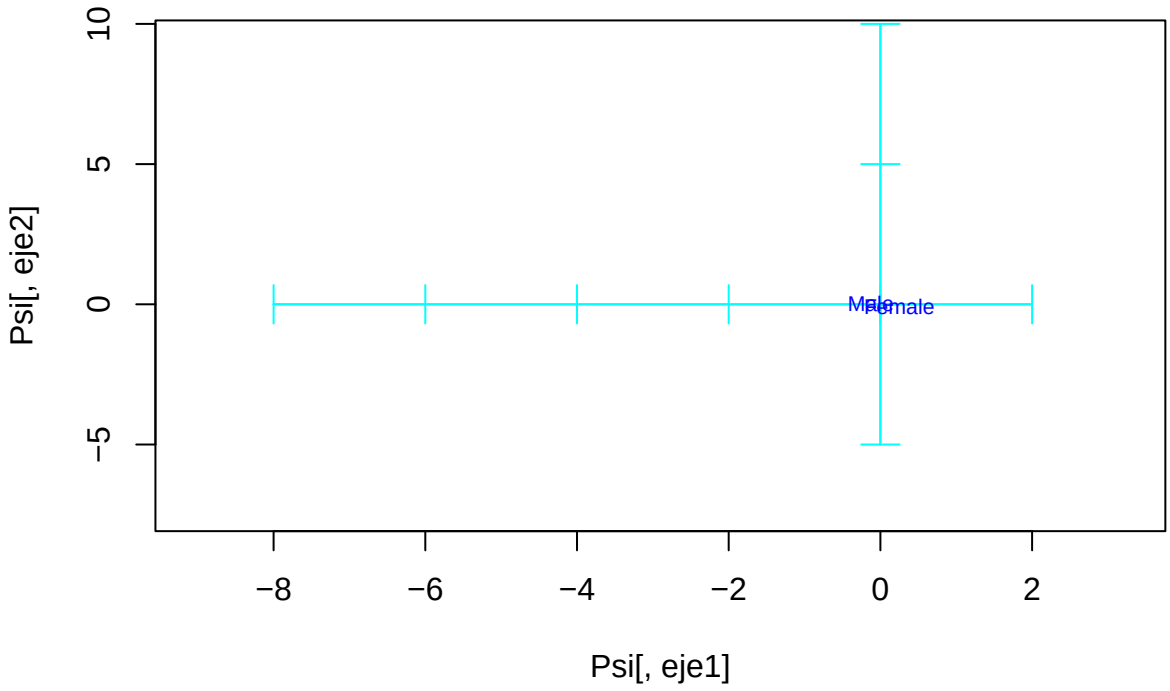
10. Projection of Qualitative Variables (Illustrative Category)

```r
# Example with variable in column 8
varcat <- factor(dd[, 8])
plot(Psi[, 1], Psi[, 2], col = varcat)
axis(side = 1, pos = 0, labels = FALSE, col = "darkgray")
axis(side = 3, pos = 0, labels = FALSE, col = "darkgray")
axis(side = 2, pos = 0, labels = FALSE, col = "darkgray")
axis(side = 4, pos = 0, labels = FALSE, col = "darkgray")
legend("bottomleft", levels(factor(varcat)), pch = 1, col = c(1, 2), cex = 0.6)

k <- 8 # gender (male or female)
varcat <- factor(dd[, k])
fdic1 <- tapply(Psi[, eje1], varcat, mean)
fdic2 <- tapply(Psi[, eje2], varcat, mean)
text(fdic1, fdic2, labels = levels(varcat), col = "yellow", cex = 0.7)
```



11. Projection of Centroids for a Qualitative Variable without Individuals

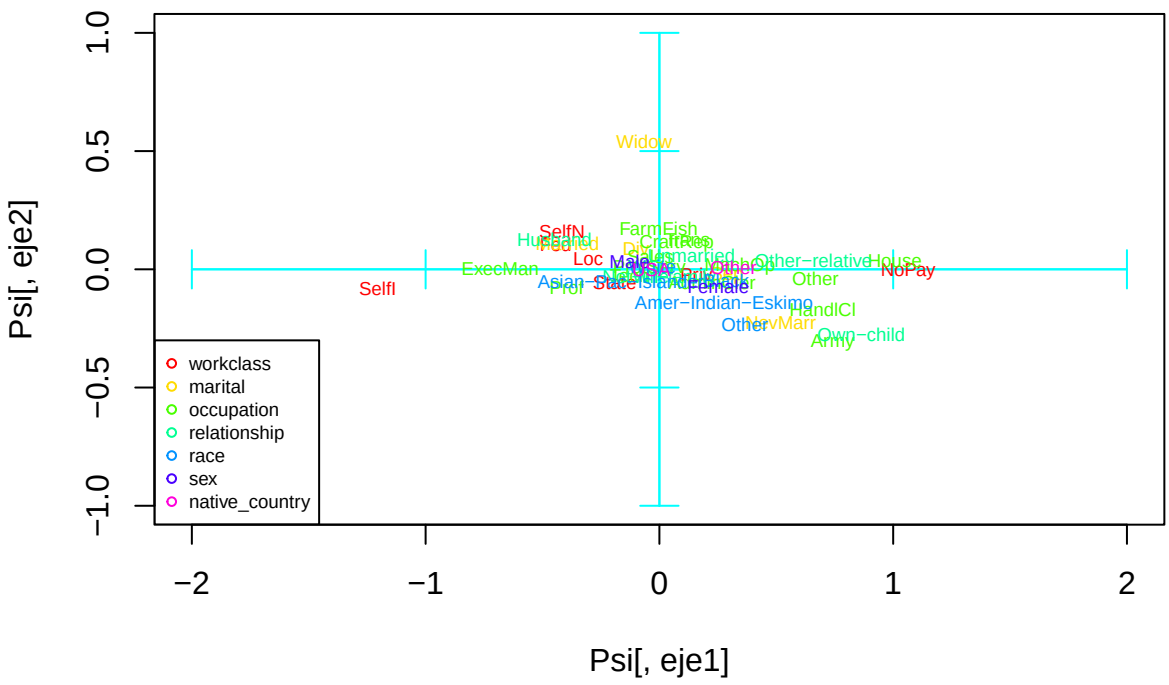
```r
# Empty plot
plot(Psi[, eje1], Psi[, eje2], type = "n")
axis(side = 1, pos = 0, labels = FALSE, col = "cyan")
axis(side = 3, pos = 0, labels = FALSE, col = "cyan")
axis(side = 2, pos = 0, labels = FALSE, col = "cyan")
axis(side = 4, pos = 0, labels = FALSE, col = "cyan")

# Same variable k = 8
# fdic1, fdic2 assumed computed above
text(fdic1, fdic2, labels = levels(varcat), col = "blue", cex = 0.7)
```



12. Joint Projection of All Qualitative Variables

```
x_center <- mean(Psi[, eje1])
y_center <- mean(Psi[, eje2])

plot(Psi[, eje1], Psi[, eje2], type = "n",
     xlim = x_center + c(-2, 2),
     ylim = y_center + c(-1, 1))
axis(side = 1, pos = 0, labels = FALSE, col = "cyan")
axis(side = 3, pos = 0, labels = FALSE, col = "cyan")
axis(side = 2, pos = 0, labels = FALSE, col = "cyan")
axis(side = 4, pos = 0, labels = FALSE, col = "cyan")

dcat <- c(2, 4:8, 12)
colors <- rainbow(length(dcat))
c <- 1
for (k in dcat) {
  seguentColor <- colors[c]
  fdic1 <- tapply(Psi[, eje1], dd[, k], mean)
  fdic2 <- tapply(Psi[, eje2], dd[, k], mean)
  text(fdic1, fdic2, labels = levels(factor(dd[, k])),
       col = seguentColor, cex = 0.6)
  c <- c + 1
}
```

```
legend("bottomleft", names(dd)[dcat], pch = 1, col = colors, cex = 0.6)
```



13. Custom Color Palette and Final Plot

```
colors <- c("red", "blue", "darkgreen",
            "orange", "violet", "magenta", "pink")

plot(Psi[, eje1], Psi[, eje2], type = "n", xlim = c(-1, 1), ylim = c(-3, 1))
axis(side = 1, pos = 0, labels = FALSE, col = "cyan")
axis(side = 3, pos = 0, labels = FALSE, col = "cyan")
axis(side = 2, pos = 0, labels = FALSE, col = "cyan")
axis(side = 4, pos = 0, labels = FALSE, col = "cyan")

arrows(ze, ze, X, Y, length = 0.07, col = "lightgray")
text(X, Y, labels = etiq, col = "gray", cex = 0.7)

c <- 1
for (k in dcat) {
  seguentColor <- colors[c]
  fdic1 <- tapply(Psi[, eje1], dd[, k], mean)
  fdic2 <- tapply(Psi[, eje2], dd[, k], mean)
  text(fdic1, fdic2, labels = levels(factor(dd[, k])),
       col = seguentColor, cex = 0.6)
  c <- c + 1
```

```
}
legend("bottomleft", names(dd)[dcat], pch = 19, col = colors, cex = 0.6)
```