R Lab. - Exercise 2

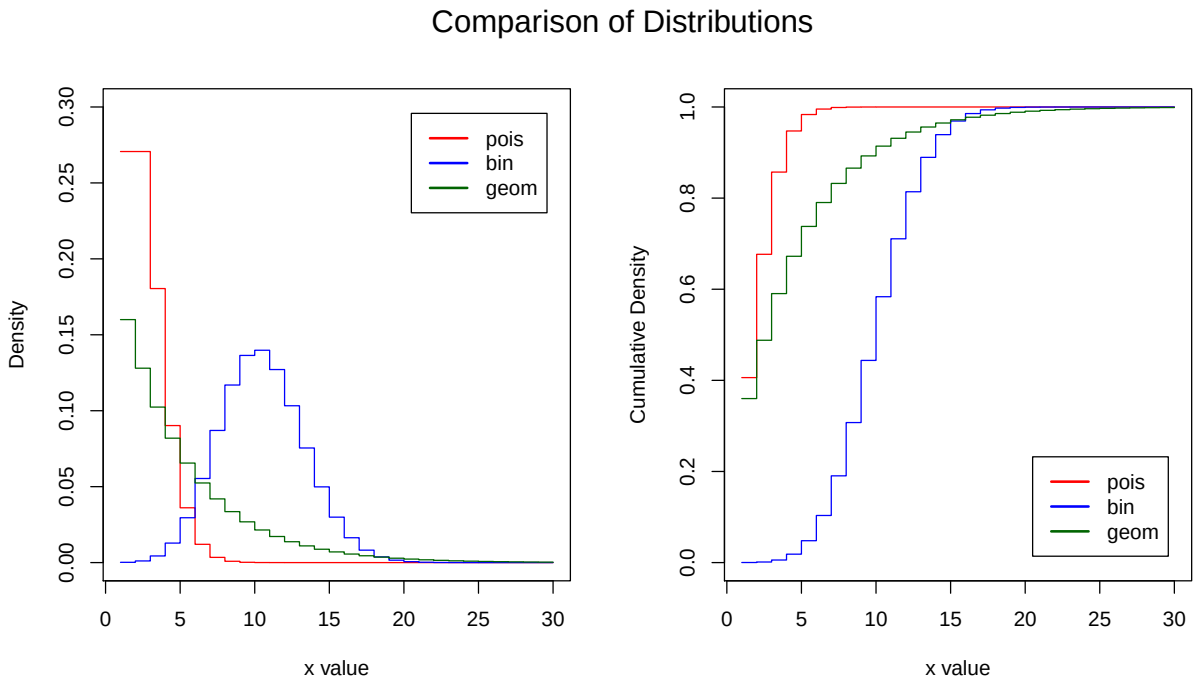
Michele Guadagnini - Mt. 1230663

April 22, 2020

Exercise 0 - Practice with the discrete probability distributions in R

Practice with the discrete probability distributions in R

```
x <- 1:30
poisx <- dpois(x, 2)
binx  <- dbinom(x, 50, 0.2)
geomx <- dgeom(x, 0.2)
ppoisx <- ppois(x, 2)
pbinx  <- pbinom(x, 50, 0.2)
pgeomx <- pgeom(x, 0.2)
colors <- c("red", "blue", "darkgreen")
labels <- c("pois", "bin", "geom")

par(mfrow=c(1,2), oma=c(0,0,1,0))
plot(x, poisx, type="s", xlab="x value", ylab="Density", ylim=c(0,0.3),
     col=colors[1])
lines(x, binx, type="s", col=colors[2])
lines(x, geomx, type="s", col=colors[3])
legend("topright", inset=.05, labels, lwd=2, lty=c(1, 1, 1), col=colors)
plot(x, ppoisx, type="s", xlab="x value", ylab="Cumulative Density", ylim=c(0,1),
     col=colors[1])
lines(x, pbinx, type="s", col=colors[2])
lines(x, pgeomx, type="s", col=colors[3])
legend("bottomright", inset=.05, labels, lwd=2, lty=c(1, 1, 1), col=colors)
mtext("Comparison of Distributions", outer=TRUE, cex=1.5, line=-2)
```



Exercise 1 - Concentration of a contaminant in tap water

A set of measurements have been performed on the concentration of a contaminant in tap water with two methods. Evaluate the expected values, E[X], and the variance, Var(X), for both methods.

```
x  <- c(15.58, 15.9, 16 , 16.1, 16.2)
p1 <- c(0.15, 0.21, 0.35, 0.15, 0.14)
p2 <- c(0.14, 0.05, 0.64, 0.08, 0.09)

Expectation1 <- sum(x*p1)
Var1 <- sum(x*x*p1) - sum(x*p1)**2
Expectation2 <- sum(x*p2)
Var2 <- sum(x*x*p2) - sum(x*p2)**2
```

For method 1:
the expected value is: 15.959
the variance is: 0.033979

For method 2:
the expected value is: 15.9622
the variance is: 0.0281672

Exercise 2 - Waiting time at the doctor's

The waiting time, in minutes, at the doctor's is about 30 minutes, and the distribution follows an exponential pdf with rate 1/30

A) simulate the waiting time for 50 people at the doctor's office and plot the relative histogram

```
Npp <- 50
lambda <- 1/30
x <- 1:Npp
wt <- rexp(x, lambda)

hist(wt, col="red", density=40, xlim=c(0,max(wt)+30), ylim=c(0,35),
     ylab="Persons", xlab="Waiting time [min]", main="Simulated waiting times")
```

B) what is the probability that a person will wait for less than 10 minutes?

```
theoProb10min <- pexp(10,lambda)
empProb10min  <- length(wt[wt<10])/length(wt)
```

Estimation from theoretical distribution: 0.2834687
Estimation from simulated data: 0.24

C) evaluate the average waiting time from the simulated data and compare it with the expected value

```
integrand <- function(z){z*dexp(z, lambda)}
integralWtavg <- integrate(integrand, 0, Inf)
theoricWtavg <- 1/lambda
empiricalWtavg <- mean(wt)
```

The estimation by integrating the distribution is: 30 min.
The theoretical expected waiting time is: 30 min.
The waiting time estimated from simulation is: 34.1034524 min.

D) what is the probability for waiting more than one hour before being received?

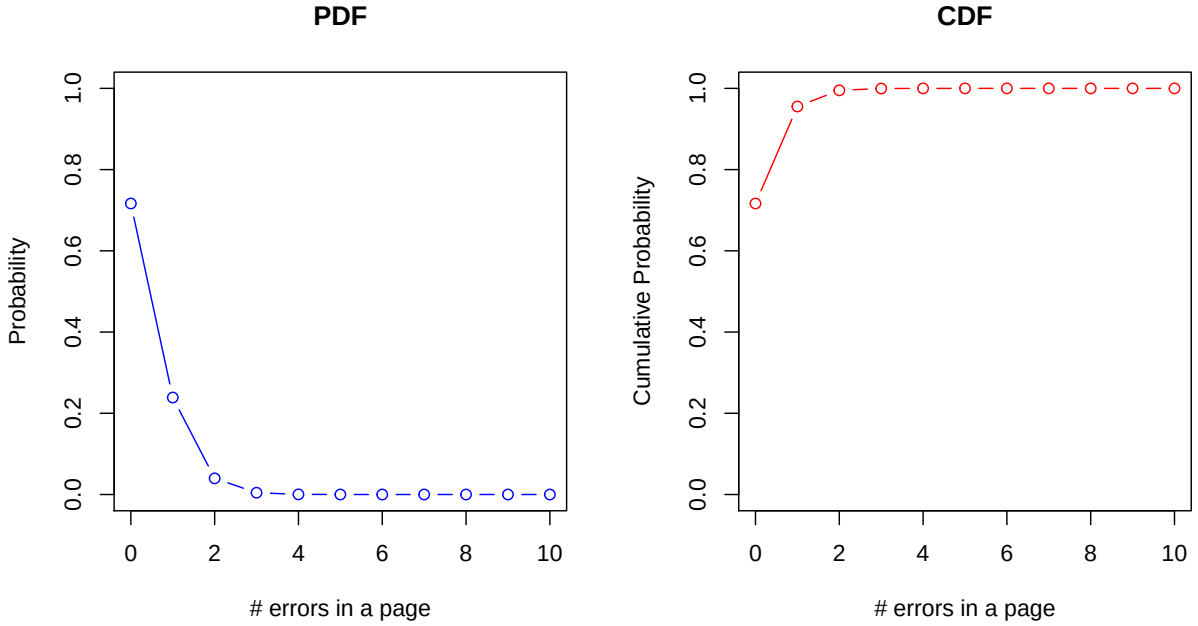
```
theoProb60min <- 1-pexp(60,lambda)
empProb60min  <- length(wt[wt>60])/length(wt)
```

The theoretical probability of waiting more then 60 minutes is: 0.1353353
The one estimated from simulation is: 0.2

Exercise 3 - Typos in a book

Suppose that on a book, on average, there is one typo error every three pages. If the number of errors follows a Poisson distribution, plot the pdf and cdf.

```
Nmax <- 10
rate <- 1/3
x <- seq(0, Nmax, length=Nmax+1)
poispdf <- dpois(x, rate)
poiscdf <- ppois(x, rate)

par(mfrow=c(1,2))
plot(x, poispdf, type="b", xlab="# errors in a page", ylab="Probability",
     main="PDF", col=4, ylim=c(0,1))
plot(x, poiscdf, type='b', xlab="# errors in a page", ylab="Cumulative Probability",
     main="CDF", col=2, ylim=c(0,1))
```



Calculate the probability that there is at least one error on a specific page of the book.

```
Prob.1err <- 1-ppois(0,rate)
```

The probability of finding one or more errors in a given page is 0.2834687.

Exercise 4 - Ace from a deck

We randomly draw cards from a deck of 52 cards, with replacement, until one ace is drawn. Calculate the probability that at least 10 draws are needed.

We need to exclude all the cases where we have less than 9 failures:

```
p <- 4/52
minfails <- 8
Prob  <- 1 - pgeom(minfails, p)
```

The probability is: 0.4865652

Exercise 5 - Italian mayors database

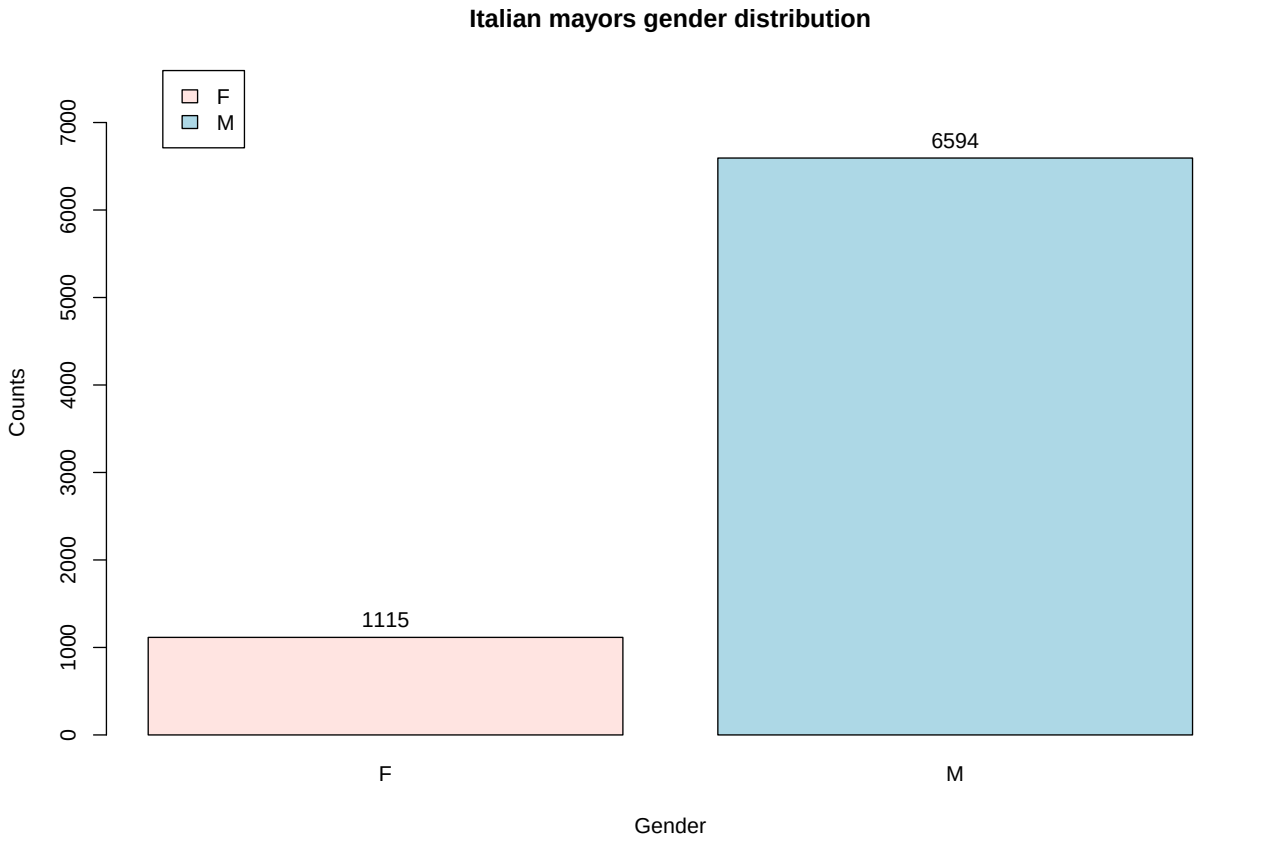
Importing the dataframe containing all the italian mayors currently in charge:

```
dfmayors <- read.csv2("sindaciincarica.csv", skip=2)
names(dfmayors)
```

```
##  [1] "codice_regione"          "codice_provincia"
##  [3] "codice_comune"           "denominazione_comune"
##  [5] "sigla_provincia"         "popolazione_censita"
##  [7] "titolo_accademico"       "cognome"
##  [9] "nome"                    "sesso"
## [11] "data_nascita"            "luogo_nascita"
## [13] "descrizione_carica"      "data_elezione"
## [15] "data_entrata_in_carica" "partito"
## [17] "titolo_studio"           "professione"
```

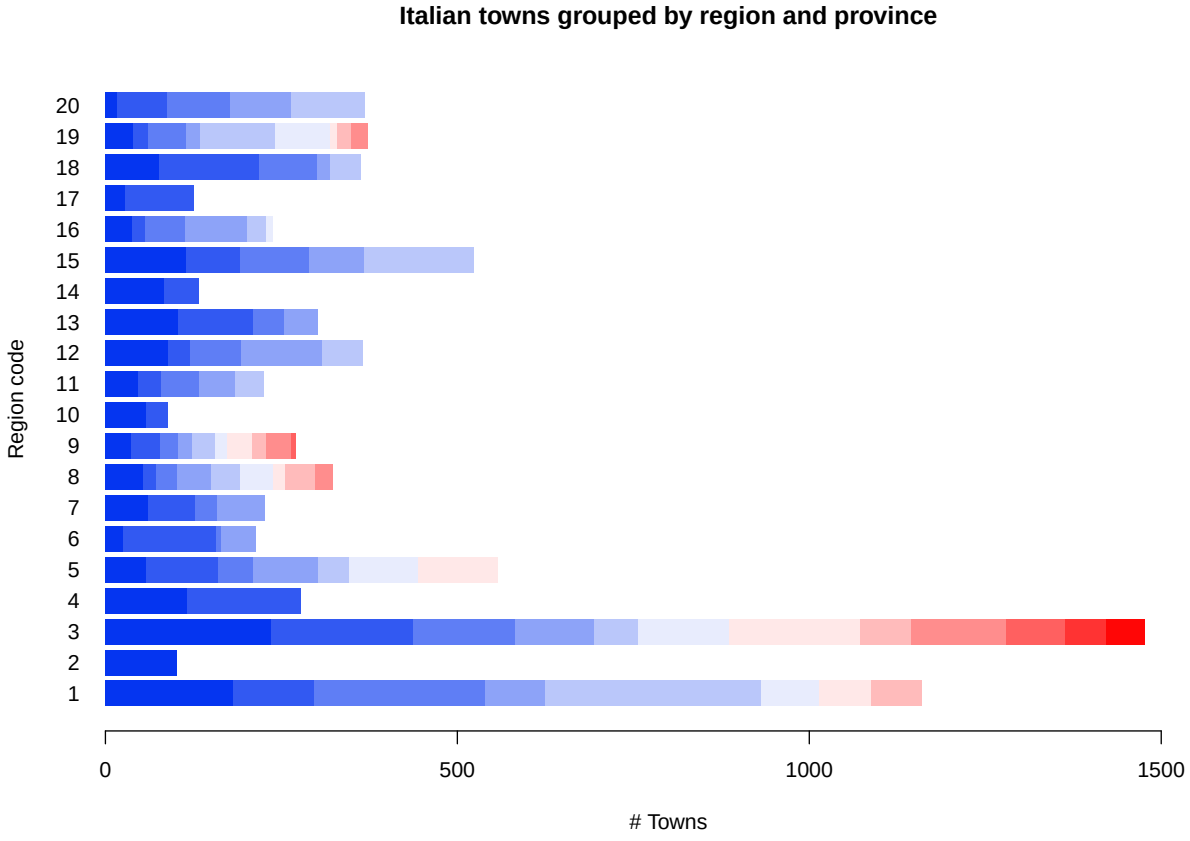
A) Plot the gender distribution among the mayors.

```
genddist <- table(dfmayors$sesso)

bp<-barplot(genddist, beside=TRUE, col=c("mistyrose", "lightblue"),
            xlab="Gender", ylab="Counts", legend=c("F","M"), ylim=c(0,max(genddist)+1000),
            args.legend=list(x="topleft", inset=c(0.05,0)))
text(bp, genddist+200, labels=genddist, xpd=TRUE)
title( main="Italian mayors gender distribution" )
```



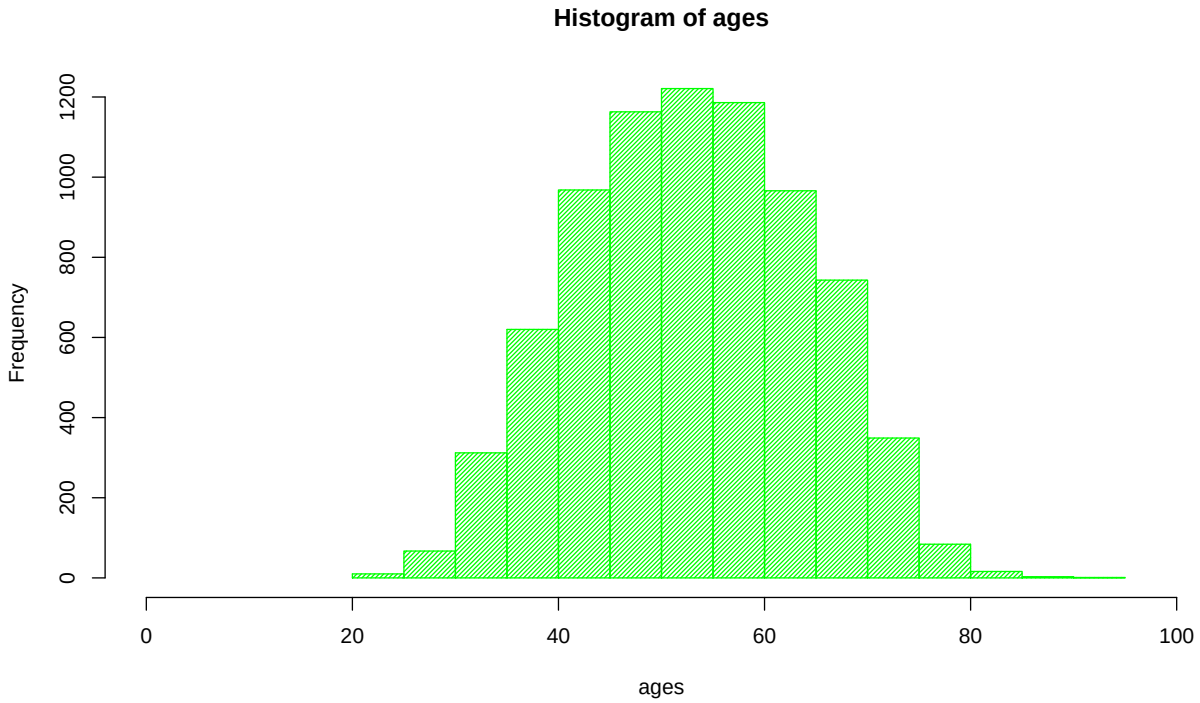
B) Plot the number of towns grouped per province (codice provincia) and per region (codice regione).

```
dfmayors <- tibble(dfmayors)

gdf <- group_split(dfmayors %>% group_by(dfmayors$codice_regione))
towntable <- vector("list", length(gdf))
for (i in 1:length(gdf)) {
    towntable[[i]] <- as.numeric(table(gdf[[i]]$codice_provincia,gdf[[i]]$codice_regione))
}
towntable <- as.data.frame(data.table::transpose(towntable), row.names=1:length(gdf))
color.ramp <- colorRampPalette(c("#0535F0", "#FFFFFF", "#FF0606"))(n=12)

barplot(t(as.matrix(towntable)), ylab="Region code", xlab="# Towns",
        main="Italian towns grouped by region and province",
        border=F, las=1, horiz=T, xlim=c(0, 1600), col=color.ramp)
```



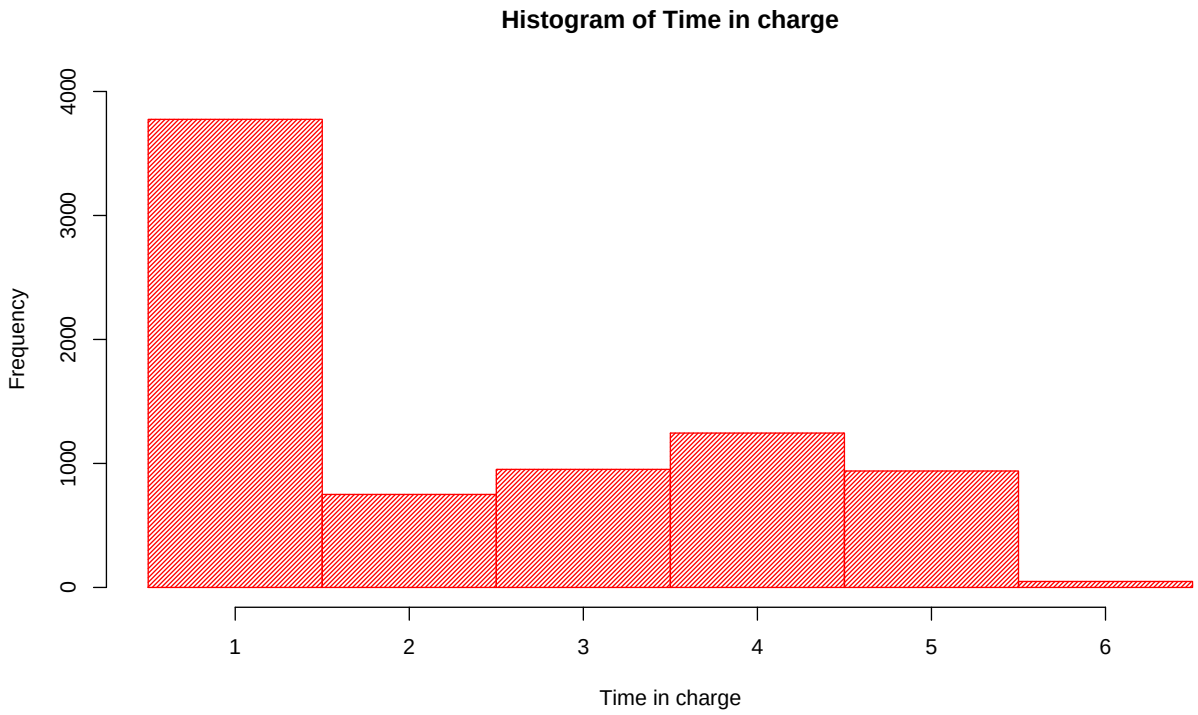
C) Plot a distributions of the age (years only) of the mayors.

```
cyear <- as.numeric(format(Sys.Date(), "%Y"))
ages <- cyear - as.numeric(format(as.Date(dfmayors$data_nascita, format="%d/%m/%Y"),"%Y"))

hist(ages, col="green", xlim=c(0,100), density=40)
```



D) Plot a distribution of the time the mayor is in charge. Since elections happen every 5 years, how many of them are going to complete their mandate this year? And how many in 2021?

```
cyear <- as.numeric(format(Sys.Date(), "%Y"))
elecyear <- format(as.Date(dfmayors$data_entrata_in_carica, format="%d/%m/%Y"),"%Y")
timeincharge <- cyear - as.numeric(elecyear)

breaks <- seq(0.5, 6.5, length=7)
htimeincharge <- hist(timeincharge, breaks=breaks, col="red", density=40,
           main="Histogram of Time in charge", xlab="Time in charge",
           ylim=c(0,4000))
```




```
outthisyear <- sum(htimeincharge$counts[5])
outnextyear <- sum(htimeincharge$counts[4])
total       <- sum(htimeincharge$counts)
```

The number of mayors completing their mandate this year are 939 over a total number of 7709.
The number of mayors completing their mandate next year are 1245 over 7709.