BSG - Homework 1

HENRY QIU LO & MEYSAM ZAMANI

November 17, 2019

1) SNP dataset

1. The file CHDCHR22RAW.raw contains all SNPs on chromosome 22 of a sample of Chinese individuals in Metropolitan Denver, CO, USA. This data has been extracted from the 1000 genomes project at www.internationalgenome.org.

2. Load this data into the R environment, with the read.table instruction. The first six columns contain non-genetical information. Extract the variables individual ID (the second column IID) and the sex of the individual (the 5th column sex). Create a dataframe that only contains the genetic information that is in and beyond the 7th column. Notice that the genetic variants are identifed by an "rs" identifier. The genetic data is coded in the (0, 1, 2) format with 0=AA, 1=AB, 2=BB.

FIrst step is loading data and do some preprocessing on it:

```
download.file("http://www-eio.upc.es/~jan/data/bsg/CHDCHR22RAW.raw","chinese.raw")
wholedataset <- read.table("chinese.raw", header = TRUE)
IDSex <- wholedataset[,c(2,5)]
dataset <- wholedataset[,c(7:ncol(wholedataset))]
IDSex$SEX[IDSex$SEX==1]<-"M"
IDSex$SEX[IDSex$SEX==2]<-"F"
dataset[dataset==0]<-"AA"
dataset[dataset==1]<-"AB"
dataset[dataset==2]<-"BB"
```

3. (1p) How many variants are there in this database? What percentage of the data is missing? How many individuals in the database are males and how many are females?

```
numVariants <- ncol(dataset)
numberRows <- nrow(dataset)
numberColumns <- ncol(dataset)
```

We are not sure how will the missing value will be stored, we just know that they must be different that these values

```
dataset[dataset!="AA" & dataset!="AB" & dataset!="BB"] <- NA
percentageMissing <- 100*sum(is.na(dataset))/(numberRows*numberColumns)

numberOfMales=length(IDSex$SEX[IDSex$SEX=="M"])
numberOfFemales=length(IDSex$SEX[IDSex$SEX=="F"])
```

```
numVariants
```

```
## [1] 16393
```

```
percentageMissing
```

```
## [1] 0
```

```
numberOfMales
```

```
## [1] 50
```

```
numberOfFemales
```

```
## [1] 59
```

There are 16393 variants. There are not missing data. 50 of the individuals are males and 59 are females.

4. (1p) Calculate the percentage of monomorphic variants. Exclude all monomorphics from the database for all posterior computations of the practical. How many variants do remain in your database?

Function to check whether a variant is monomorphic

```
#install.packages("genetics")
library(genetics)

is_monomorphic <- function(x) {
  genotypeOfX <- genotype(x,sep="")
  genotypeInfo <- summary(genotypeOfX)
  numberOfAlleles <- length(genotypeInfo$allele.names)
  return(numberOfAlleles > 1)
}
```

This can take some time. It looks in each column whether there are more than one type of allele.

```
booleanArrayNoMonomorphic <- apply(dataset,2,is_monomorphic)
datasetNoMonomorphic <- dataset[booleanArrayNoMonomorphic=="TRUE"]
numberNoMonomorphic <- ncol(datasetNoMonomorphic)
remainingVariants <- numberNoMonomorphic
numberMonomorphic <- numVariants - remainingVariants

percentageMonomorphic <- 100*numberMonomorphic/numVariants
```

```
percentageMonomorphic
```

```
## [1] 19.52663
```

```
remainingVariants
```

```
## [1] 13192
```

The percentage of monomorphic variatns are 19.52663%. It remains 13192 variants in my database.

5. (1p) Report the genotype counts and the minor allele count of polymorphism rs3729688 G, and calculate the MAF of this variant.

```
genotypeOf <- genotype(datasetNoMonomorphic$rs3729688_G,sep="")
  genotypeOfInfo <- summary(genotypeOf)
  genotypeCounts <- genotypeOfInfo$genotype.freq[,1]
  #also with genotypeCounts <- table(datasetNoMonomorphic$rs3729688_G)
  alleleCounts <- genotypeOfInfo$allele.freq[,1]
  minAlleleCount <- min(alleleCounts)
  alleleFrequency <- genotypeOfInfo$allele.freq[,2]
  minAlleleFrequency <- min(alleleFrequency)
  minAlleleFrequency[minAlleleFrequency==1] <- 0
```

```
minAlleleFrequency
```

```
## [1] 0.4541284
```

The genotype counts is A/A 29 A/B 61 B/B 19. The minor allele count is 99 corresponding to B. The MAF is 0.4541284.

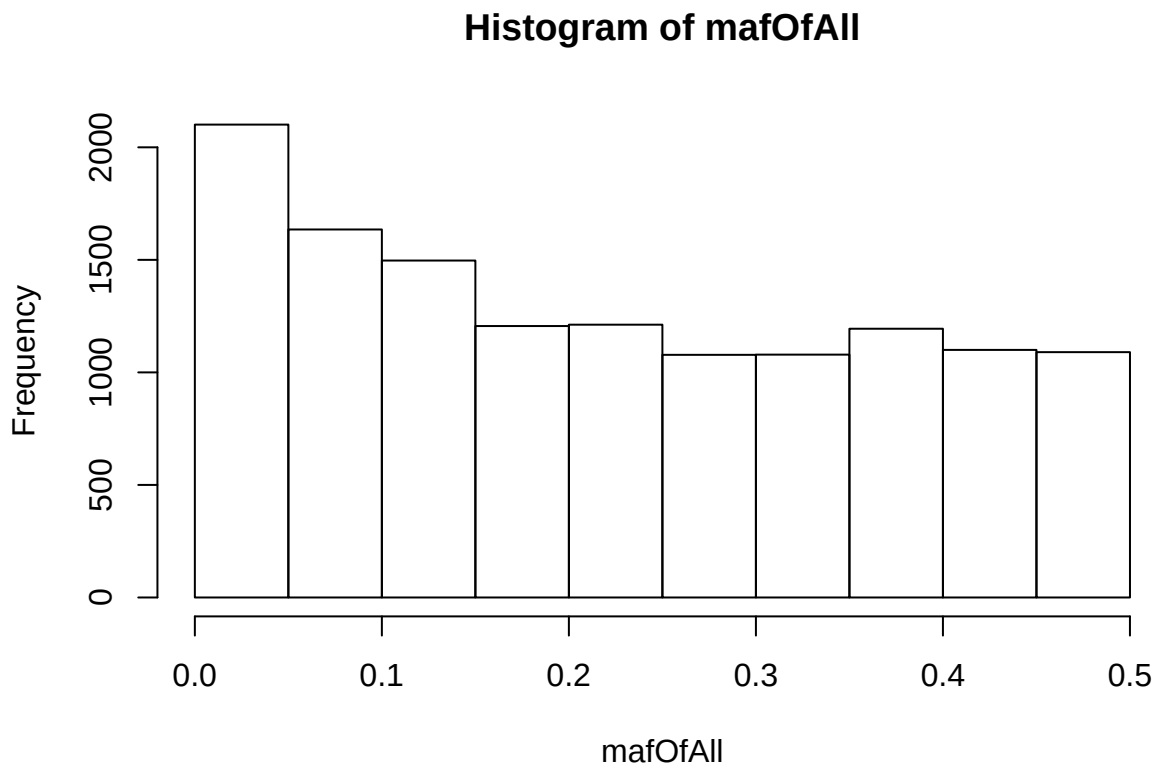
6. (2p) Compute the minor allele frequencies (MAF) for all markers, and make a histogram of it. Does the MAF follow a uniform distribution? What percentage of the markers have a MAF below 0.05? And below 0.01? Can you explain the observed pattern?

Function to compute the maf for each column of a matrix

```
maf <- function(x){
  x <- genotype(x,sep="")
  out <- summary(x)
  af1 <- min(out$allele.freq[,2],na.rm=TRUE)
  af1[af1==1] <- 0
  return(af1)
}
```

```
mafOfAll <- apply(datasetNoMonomorphic,2,maf)
#mafOfAll
```

```
hist(mafOfAll)
```




```
numberOfMarkersBelow005 <- length(mafOfAll[mafOfAll<0.05])
percentageOfMarkersBelow005 <-  100 * numberOfMarkersBelow005 / numberNoMonomorphic
numberOfMarkersBelow001 <- length(mafOfAll[mafOfAll<0.01])
percentageOfMarkersBelow001 <-  100 * numberOfMarkersBelow001 / numberNoMonomorphic
```

```
percentageOfMarkersBelow005
```

```
## [1] 15.92632
```

```
percentageOfMarkersBelow001
```

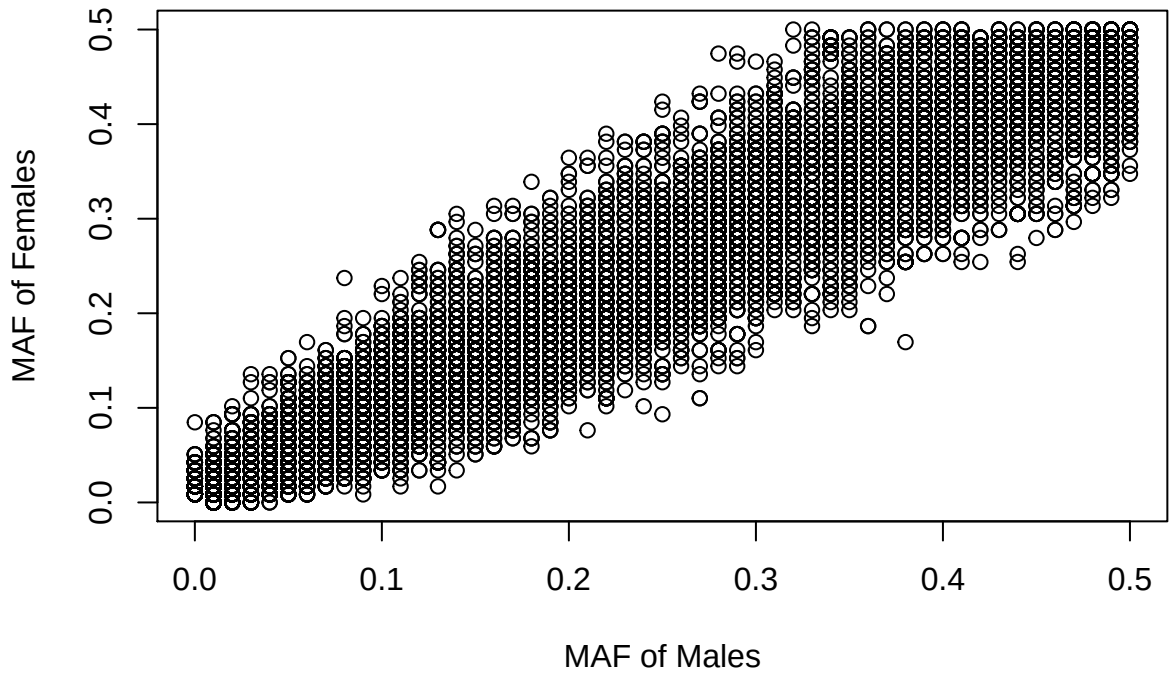
```
## [1] 5.965737
```

It seems that the distribution is uniform. The percentage of MAF below 0.03 is 15.92632%. The percentage of MAF below 0.01 is 5.965737%. From the historgram it seems that the major part of the MAF values are between 0.0 - 0.01.

7. (2p) Calculate the minor allele frequency for males and for females and present a scatterplot of these variables. What do you observe? Calculate and report their correlation coeficient.

```
datasetNoMonomorphicMales <- datasetNoMonomorphic[IDSex$SEX=="M",]
datasetNoMonomorphicFemales <- datasetNoMonomorphic[IDSex$SEX=="F",]
mafOfMales <- apply(datasetNoMonomorphicMales,2,maf)
mafOfFemales <- apply(datasetNoMonomorphicFemales,2,maf)
```

```
plot(mafOfMales,mafOfFemales,xlab="MAF of Males",ylab="MAF of Females")
```




```
correlationCoefficient <- cor(mafOfMales,mafOfFemales)
correlationCoefficient
```

```
## [1] 0.9486921
```

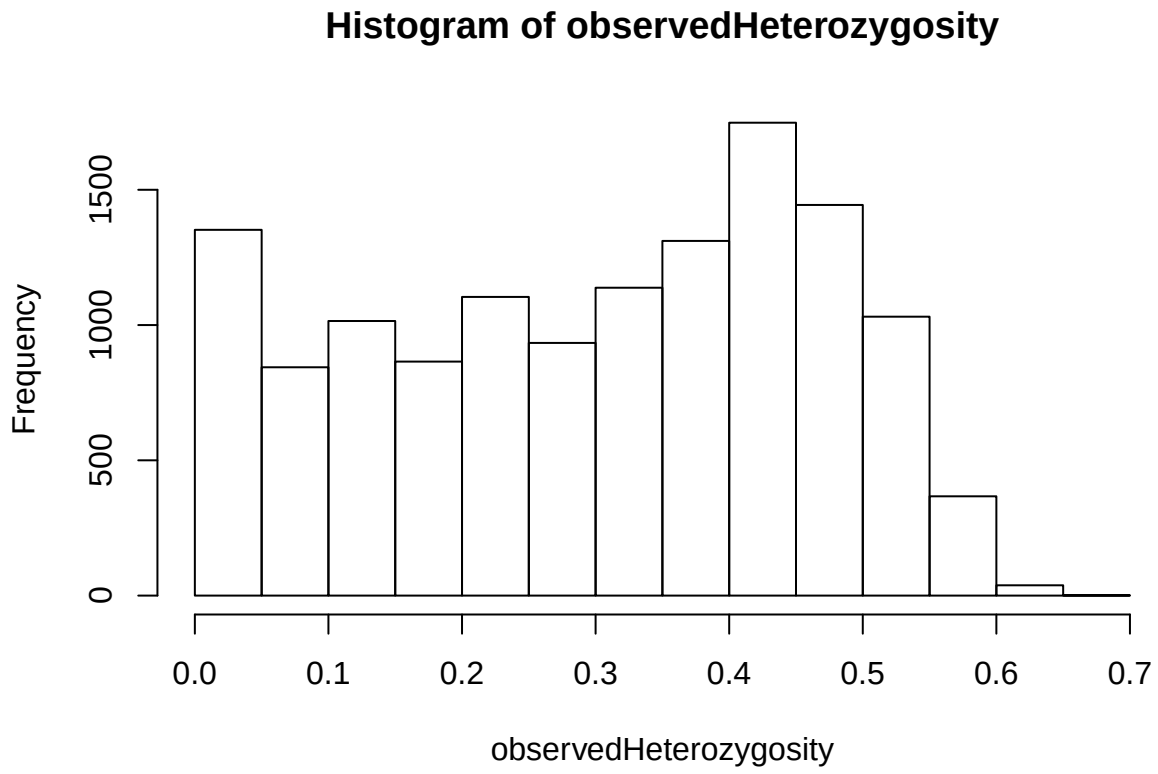
From the scatter plot we can see a clear correlation because there is a strong concentration at the diagonal of the plot. The correlation coefficient shows how high the correlation is: 0.9586921, almost a correlation of 95%.

8. (1p) Calculate the observed heterozygosity (Ho), and make a histogram of it. What is, theoretically, the range of variation of this statistic?

Function to get the observed heterozygosity of each marker.

```r
obs_hetero <- function(x) {
  genotypeOfX <- genotype(x,sep="")
  genotypeInfo <- summary(genotypeOfX)
  genotypeFrequency <- genotypeInfo$genotype.freq[,2]
  return(genotypeFrequency[2])
}
```

```r
observedHeterozygosity <- apply(datasetNoMonomorphic,2,obs_hetero)
hist(observedHeterozygosity)
```




```r
range(observedHeterozygosity)
```

```
## [1] 0.009174312 0.651376147
```

Theorically, the range variation is between 0.009174312 and 0.651376147.

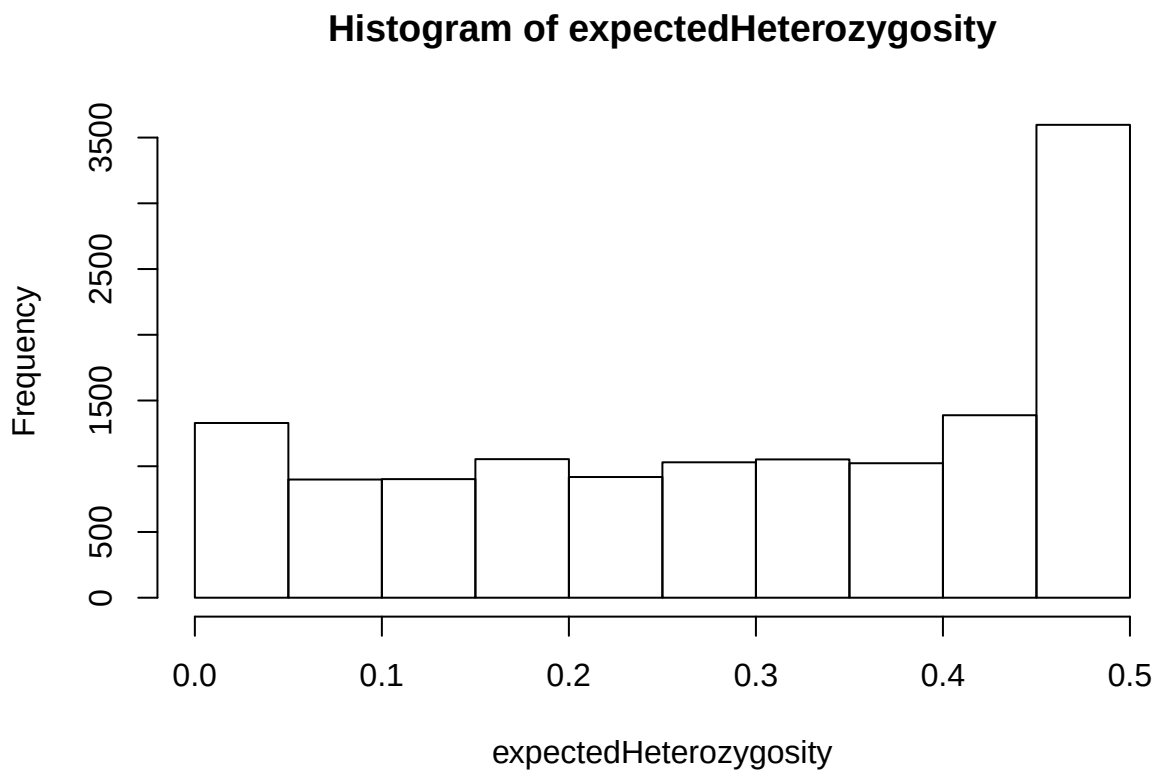
9. (2p) Compute for each marker its expected heterozygosity (He), where the expected heterozygosity for a bi-allelic marker is defined as 1-SUM P 2i where pi is the frequency of the ith allele. Make a histogram of the expected heterozygosity. What is, theoretically, the range of variation of this statistic? What is the average of He for this database?

Function to get the expected heterozygosity of each marker.

```r
exp_hetero <- function(x) {
  genotypeOfX <- genotype(x,sep="")
  genotypeInfo <- summary(genotypeOfX)
  k <- length(genotypeInfo$allele.names) #K is the number of allele
  sum = 0
```

```
  for(i in 1:k) {
    sum <- sum + genotypeInfo$allele.freq[,2][i] * genotypeInfo$allele.freq[,2][i]
  }
  return(1 - sum)
}
```

```
expectedHeterozygosity <- apply(datasetNoMonomorphic,2,exp_hetero)
hist(expectedHeterozygosity)
```




```
range(expectedHeterozygosity)
```

```
## [1] 0.009132228 0.500000000
```

```
mean(expectedHeterozygosity)
```

```
## [1] 0.2985011
```

2) STR dataset

1. The file FrenchStrs.dat contains genotype information (STRs) of individuals from a French population. The first column of the data set contains an identifier the individual. STR data starts at the second column. Load this data into the R environment.

First step is loading the data:

```
rm(list = ls())
data <- read.csv("FrenchSTRs.dat", sep= " ")
```

Theorically, the range variation is between 0.009132228 and 0.500000000. The average of He for this database is 0.2985011.

2. (1p) How many individuals and how many STRs contains the database?

```
dim(data)
```

```
## [1]  58 679
```

```
N <- length(unique(data$Individual))
N
```

```
## [1] 29
```

```
p <- ncol(data)-1
p
```

```
## [1] 678
```

```
data <- data[,2:(p+1)]
```

We have 29 individuals in the dataset, each individual has 2 rows and also there are 678 STRs.

3. (1p) The value -9 indicates a missing value. Replace all missing values by NA. What percentage of the total amount of datavalues is missing?

```
missings <- sum(data == -9)
percentage.missings <- missings/(2*N*p) * 100
percentage.missings
```

```
## [1] 4.206083
```

Totally we have 4.2% of missing values.

4. (2p) Write a function that determines the number of alleles for a STR. Determine the number of alleles for each STR in the database. Compute basic descriptive statistics of the number of alleles (mean, standard deviation, median, minimum, maximum).

```
### This funcion will return the number of alleles for a STR
n.alleles.str <- function(str){
  n.alleles <- length(unique(str[!is.na(str)]))
  return(n.alleles)
}
n.alleles <- apply(data, 2, n.alleles.str)
```

```
(mean.alleles <- mean(n.alleles))
```

```
## [1] 6.848083
```

```
(std.alleles <- sd(n.alleles))
```

```
## [1] 1.902091
```

```
(median.alleles <- median(n.alleles))
```

```
## [1] 7
```

```
(min.alleles <- min(n.alleles))
```

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

```
(max.alleles <- max(n.alleles))
```

```
## [1] 16
```

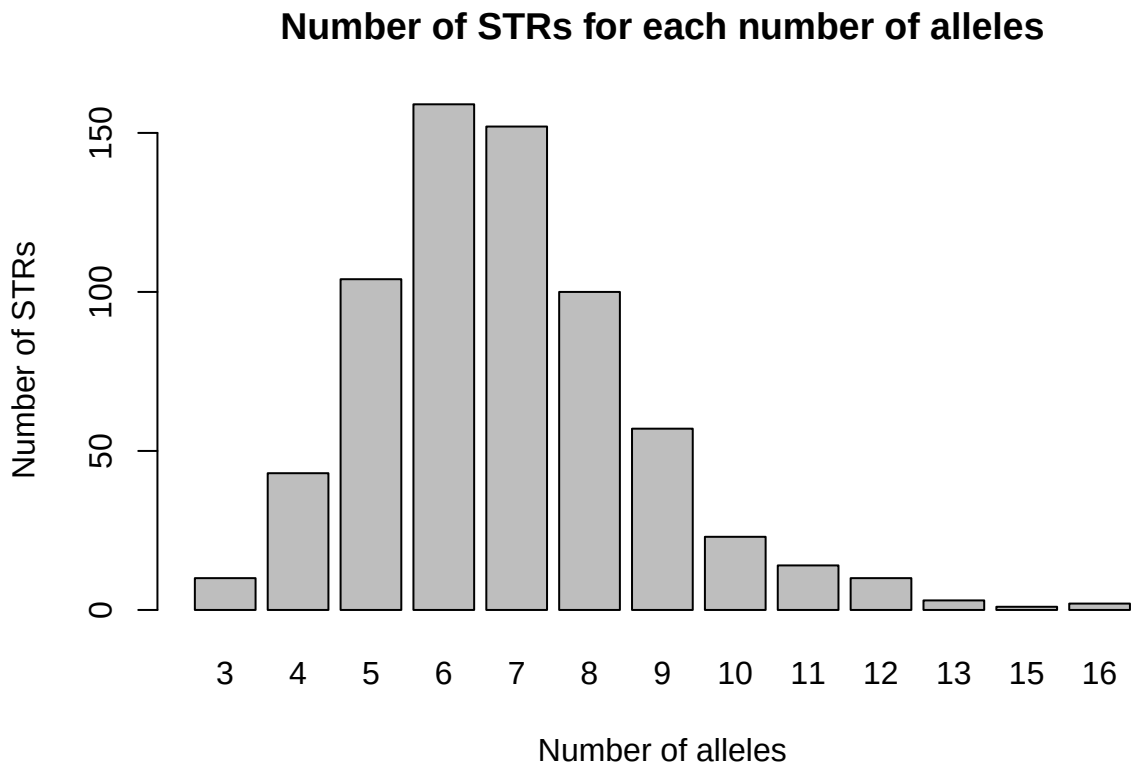
You can see that the mean of alleles is 6.848, the standard deviation is 1.902, thw median is 7, the minimum is 3 and the maximum is 16.

5. (2p) Make a table with the number of STRs for a given number of alleles and present a barplot of the number STRs in each category. What is the most common number of alleles for an STR?

```
table(n.alleles)
```

```
## n.alleles
##   3   4   5   6   7   8   9  10  11  12  13  15  16
##  10  43 104 159 152 100  57  23  14  10   3   1   2
```

```
barplot(table(n.alleles), main='Number of STRs for each number of alleles',
        xlab="Number of alleles", ylab="Number of STRs")
```

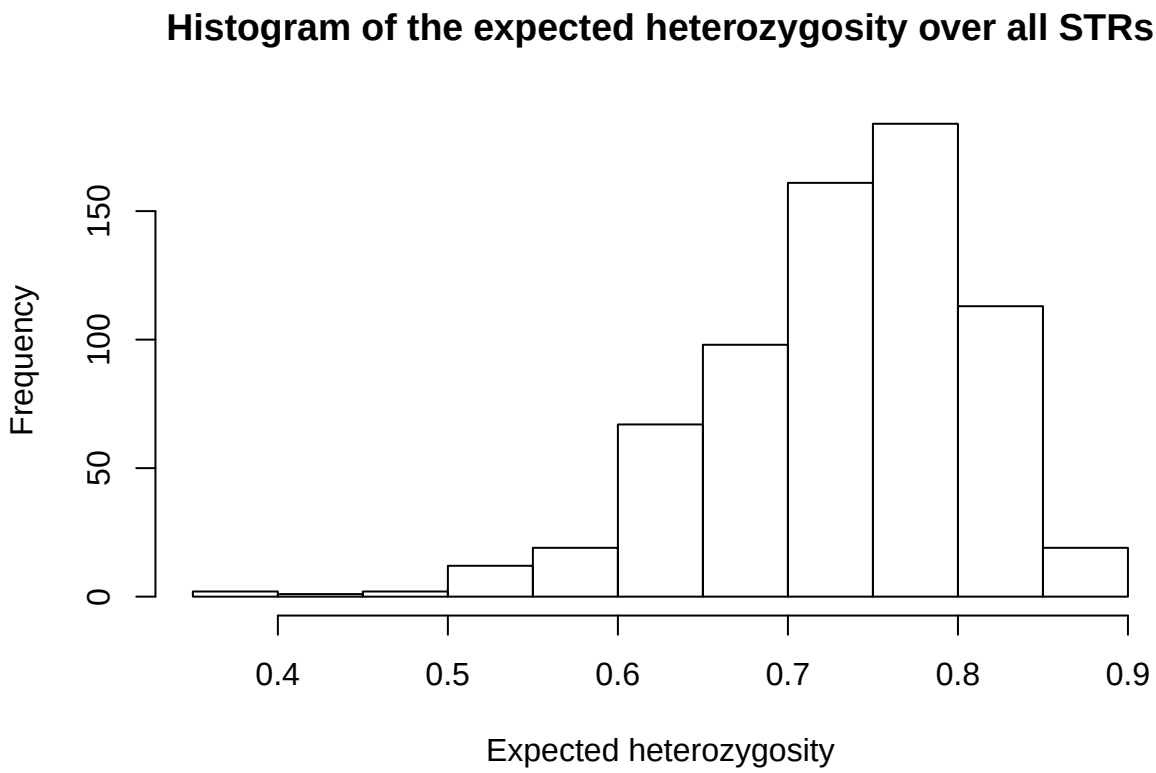


You can see with the barplot, the most common number of allels for an STR is 6.

6. (2p) Compute the expected heterozygosity for each STR. Make a histogram of the expected heterozygosity over all STRS. Compute the average expected heterozygosity over all STRs.

```
### Function for computing the expected heterozygosity for a given STR
expected.hetereozygozity.str <- function(str){
counts <- as.vector(table(str))
total <- sum(counts)
p <- counts/total
p <- p^2
return( 1 - sum(p))
}
e.h.str <- apply(data, 2, expected.hetereozygozity.str)
```

```r
hist(e.h.str, main="Histogram of the expected heterozygosity over all STRs",
     xlab='Expected heterozygosity', ylab="Frequency")
```




```r
mean(e.h.str)
```

```
## [1] 0.7328752
```

The average expected heterozygosity is 0.7328752

7. (2p) Compare the results you obtained for the SNP database with those you obtained for the STR database. What differences do you observe between these two types of genetic markers?

The first difference that we observe is in the expected heterozygosity. Since the SNP has values between 0 and 0.5 and the majority are very near to 0, the STR obtains values between 0 and 1 but nearer to 1 than 0. This is because the SNP has only two categories (A and B), thus if one is more frequent than the other, the expected heterozygosity will decrease very fast. On the other hand, the STR has more categories and the frequencies are more balanced than SNP, so the expected heterozygosity will stay high.