Practical 01 SG: Descriptive analysis of genetic markers

Ivan Almer, Lovro Katalinić

Hand-in: 20/11/2020

Resolve the following exercise in groups of two students. Perform the computations and make the graphics that are asked for in the practical below. Take care to give each graph a title, and clearly label x and y axes, and to answer all questions asked. You can write your solution in a Word or Latex document and generate a pdf file with your solution, or generate a solution pdf file with R Markdown. Take care to number your answers exactly as in this exercise. Upload your solution in **pdf format** to the web page of the course at raco.fib.upc.edu no later than the hand-in date.

You can make use of the R-package **genetics** (and other packages) to compute your answers, as you please. The first part of the practical is dedicated to the descriptive analysis of SNP data, whereas the second part is dedicated to the analysis of STR data. The datasets can be downloaded by clicking on their file names given below.

SNP dataset (10p)

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.
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?

```r
dataset = read.table("CHDCHR22.raw")

ids = dataset[,2]
ids = ids[2:length(ids)]
sexes = dataset[,5]
sexes = sexes[2:length(sexes)]

genetic_dataset = data.frame(dataset[,7:ncol(dataset)])
names = genetic_dataset[1,]
genetic_dataset = data.frame(genetic_dataset[2:nrow(genetic_dataset),])
names(genetic_dataset) = names

# Number of variants
num_var = ncol(genetic_dataset)
cat(paste('Number of variants:', num_var, '\n'))
```

```
## Number of variants: 16393
```

```
# Percentage of data missing
genetic_dataset[genetic_dataset == -9] <- NA
nmis = sum(is.na(genetic_dataset))
percentage_missing = nmis / (nrow(genetic_dataset) * ncol(genetic_dataset))
cat(paste('Percentage of data missing:', percentage_missing, '%\n'))
```

```
## Percentage of data missing: 0 %
```

```
# Number of ales and females
male_sign = 1
female_sign = 2
n_male = sum(sexes==male_sign)
n_female = sum(sexes==female_sign)
cat(paste('Male individuals: ', n_male, '\nFemale individuals: ', n_female))
```

```
## Male individuals:  50
## Female individuals:  59
```

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?

```
genetic_dataset_non_monomorphic = NULL

for(i in 1:ncol(genetic_dataset)) {
  is_monomorphic = length(unique(genetic_dataset[,i])) == 1
  if (is_monomorphic == FALSE) {
    col_name = names(genetic_dataset)[i]
    # add to dataframe
    if(is.null(genetic_dataset_non_monomorphic)) {
      genetic_dataset_non_monomorphic = data.frame(genetic_dataset[,i])
      names(genetic_dataset_non_monomorphic) = c(col_name)
    } else {
      names_before = names(genetic_dataset_non_monomorphic)
      genetic_dataset_non_monomorphic = cbind(genetic_dataset_non_monomorphic, data.frame(genetic_datase
      names(genetic_dataset_non_monomorphic) = append(names_before, col_name)
    }
  }
}


# There are 13192 Variants remaining in my database
perc = 1 - ncol(genetic_dataset_non_monomorphic) / ncol(genetic_dataset)
cat(paste('Percentage of monomorphic variants: ', perc, '\n'))
```

```
## Percentage of monomorphic variants:  0.195266272189349
```

```
cat(paste('Number of variants after excluding monomorphics: ', ncol(genetic_dataset_non_monomorphic)))
```

```
## Number of variants after excluding monomorphics:  13192
```

5. (1p) Report the genotype counts and the minor allele count of polymorphism rs3729688_G, and calculate the MAF of this variant.

```
## Loading required package: combinat
```

```
##
## Attaching package: 'combinat'
```

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

```
##
##     combn
```

```
## Loading required package: gdata
```

```
## gdata: read.xls support for 'XLS' (Excel 97-2004) files ENABLED.
```

```
##
```

```
## gdata: read.xls support for 'XLSX' (Excel 2007+) files ENABLED.
```

```
##
## Attaching package: 'gdata'
```

```
## The following object is masked from 'package:stats':
##
##     nobs
```

```
## The following object is masked from 'package:utils':
##
##     object.size
```

```
## The following object is masked from 'package:base':
##
##     startsWith
```

```
## Loading required package: gtools
```

```
## Loading required package: MASS
```

```
## Loading required package: mvtnorm
```

```
##
```

```
## NOTE: THIS PACKAGE IS NOW OBSOLETE.
```

```
##
```

```
##   The R-Genetics project has developed an set of enhanced genetics
```

```
##   packages to replace 'genetics'. Please visit the project homepage
```

```
##   at http://rgenetics.org for informtion.
```

```
##
```

```
##
## Attaching package: 'genetics'
```

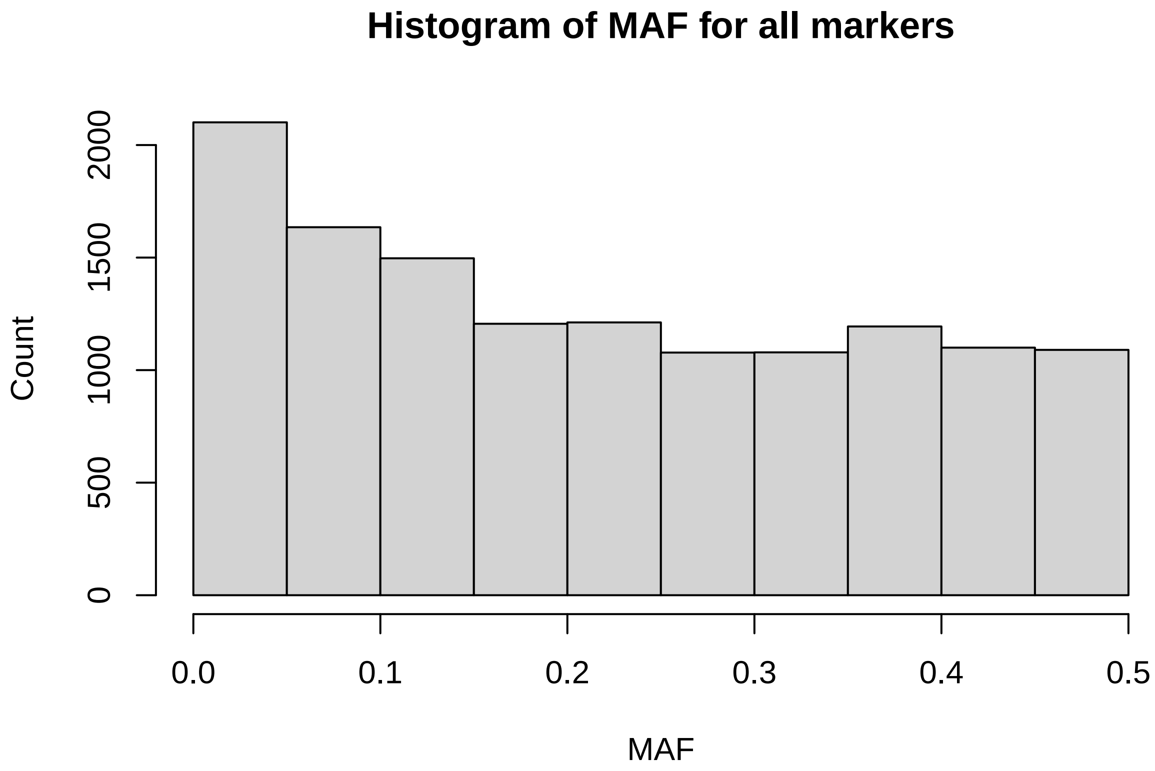
```
## The following objects are masked from 'package:base':
##
##     %in%, as.factor, order
```

```
##
## Number of samples typed: 109 (100%)
##
## Allele Frequency: (2 alleles)
##   Count Proportion
## A   119       0.55
## B    99       0.45
##
##
## Genotype Frequency:
##     Count Proportion
## A/A    29       0.27
## A/B    61       0.56
## B/B    19       0.17
##
## Heterozygosity (Hu)  = 0.4980764
## Poly. Inf. Content   = 0.3728869

##
## MAF of variant: 0.454128440366972
```

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?

```
maf.per.snp <- apply(genetic_dataset_non_monomorphic,2,maf)
hist(maf.per.snp, main="Histogram of MAF for all markers", xlab="MAF", ylab="Count")
```



```
cat(paste('Percentage of markers with MAF below the 0.05: ', sum(maf.per.snp < 0.05) / length(maf.per.s
```

```
## Percentage of markers with MAF below the 0.05:  0.159263189812007
```

```
cat(paste('Percentage of markers with MAF below the 0.01: ', sum(maf.per.snp < 0.01) / length(maf.per.s
```

```
## Percentage of markers with MAF below the 0.01:  0.0596573681018799
```

The MAF more or less follows uniform distribution, at least visually. It is not perfect since there is substantial number of variants with MAF below 0.1. There is 15.9% of variants with MAF below 0.05, and 6% variants with MAF below 0.01.What this means is that there are a lot more variants with near-monomorphic constitution.

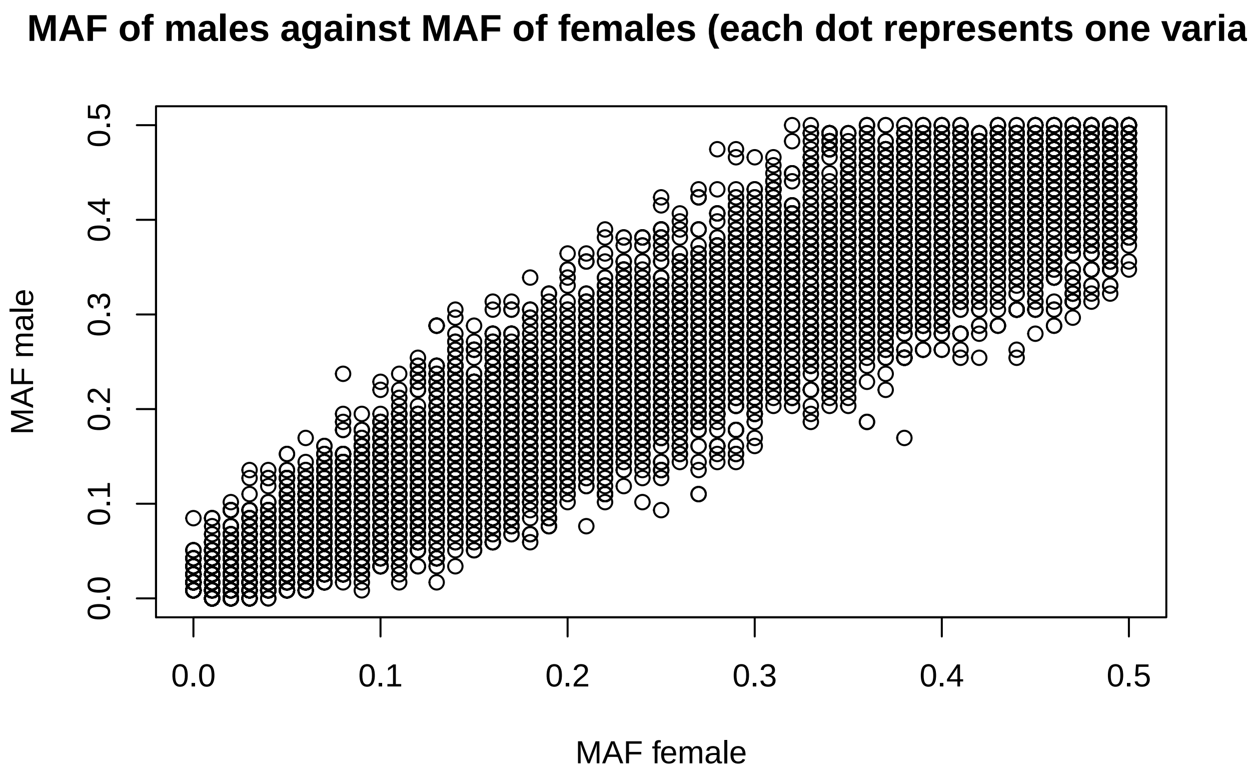
7. (2p) Calculate the minor allele frequency for males and for females and present a scatterplot of these variables. What do you observe? Calculate and report their correlation coefficient.

```
maf.male <- function(x){
  male_mask = sexes == male_sign
  x = x[male_mask]
  af1 = maf(x)
  return(af1)
}

maf.female <- function(x){
  female_mask = sexes == female_sign
  x = x[female_mask]
  af1 = maf(x)
  return(af1)
}

maf.per.snp.male = apply(genetic_dataset_non_monomorphic, 2, maf.male)
maf.per.snp.female = apply(genetic_dataset_non_monomorphic, 2, maf.female)

plot(maf.per.snp.male, maf.per.snp.female, main="MAF of males against MAF of females (each dot represen
```



```
cat(paste('Correlation coefficient between male and female minor MAFs: ', cor(maf.per.snp.male, maf.per
```

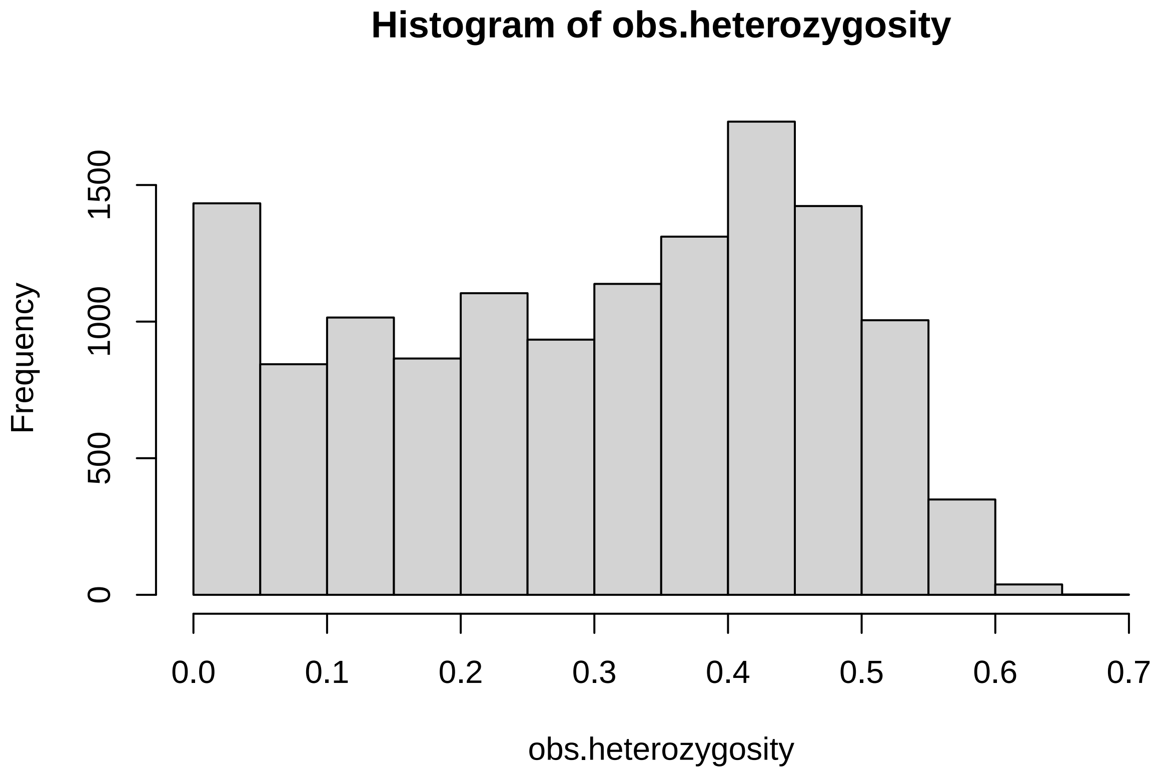
```
## Correlation coefficient between male and female minor MAFs:  0.948692122941391
```

We can observe that there exists some linear dependency between the minimum allele frequency among males and females. Since visually it makes sense that those 2 variables are correlated, we calculate the Pearson correlation coefficient which is near 0.95. That means that there is a high positive correlation between MAF in males and females (i.e. if one grows, other one also grows).

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

```
h.observed <- function(x) {
  x[x == 0] = "AA"
  x[x == 1] = "AB"
  x[x == 2] = "BB"
  x <- genotype(x,sep="")
  out <- summary(x)
  h.o = out$genotype.freq[,2]["A/B"]
  if(is.na(h.o)) {
    return(0.0)
  }
  return(h.o)
}

# Observed heterozygosity is genotype frequency f_AB
obs.heterozygosity = apply(genetic_dataset_non_monomorphic, 2, h.observed)
hist(obs.heterozygosity)
```



Theoretically the variation of this statistic can be in interval [0, 1.0] because H_o = f_AB, and f_AB can have frequency of 0 and also frequency of 1.0.

9. (2p) Compute for each marker its expected heterozygosity (H_e), where the expected heterozygosity for a bi-allelic marker is defined as $1 - \sum_{i=1}^{k} p_i^2$, where p_i is the frequency of the ith allele. Make a

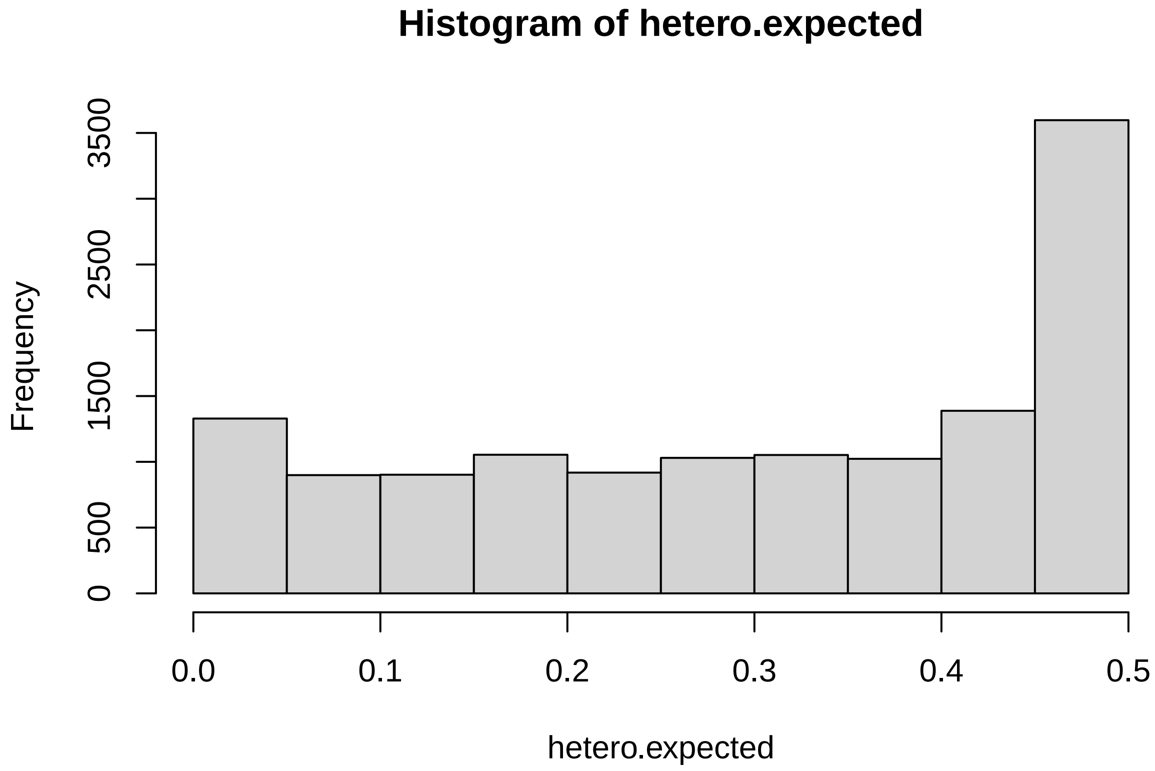
histogram of the expected heterozygosity. What is, theoretically, the range of variation of this statistic? What is the average of H_e for this database?

```
h.expected <- function(x) {
  x[x == 0] = "AA"
  x[x == 1] = "AB"
  x[x == 2] = "BB"
  x <- genotype(x,sep="")
  out <- summary(x)

  res = 1.0
  for(p in out$allele.freq[,2]) {
    res = res - p*p
  }

  return(res)
}

# Expected heterozygosity for all markers
hetero.expected = apply(genetic_dataset_non_monomorphic, 2, h.expected)
hist(hetero.expected)
```



```
# Average heterozygosity
avg.hetero = sum(hetero.expected) / length(hetero.expected)
cat(paste('The average H_e for this database is :', avg.hetero))
```

```
## The average H_e for this database is : 0.298501117356988
```

Theoretically the variation of this statistic depends on number of alleles. In this case it can be in interval [0,0.5]. But in case of 4 alleles it can be anywhere between [0, 0.75]. The average expected heterozygosity for this data-set is 0.2985.

STR dataset (10p)

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.

```r
French = read.table("FrenchStrs.dat")
dataset = French[,2:ncol(French)]
```

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

```r
n = nrow(dataset) / 2 # 2 rows for each individual
strs = ncol(dataset)
cat(paste('Number of individuals: ', n, '\n'))
```

```
## Number of individuals:  29
```

```r
cat(paste('Number of STRs: ', strs))
```

```
## Number of STRs:  678
```

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?

```r
dataset[dataset == -9] = NA

n.mis = sum(is.na(dataset))
n.total = nrow(dataset) * (ncol(dataset))

perc.mis = n.mis / n.total
cat(paste('Percentage of missing values: ', round(perc.mis*100, 2), '%\n'))
```

```
## Percentage of missing values:  4.21 %
```

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).

```r
n.alleles <- function(x) {
  length(unique(x[!is.na(x)]))
}

n.alleles.per.STR = apply(dataset, 2, n.alleles)

cat(paste('Standard deviation: ', sd(n.alleles.per.STR), '\n'))
```

```
## Standard deviation:  1.82338480266172
```

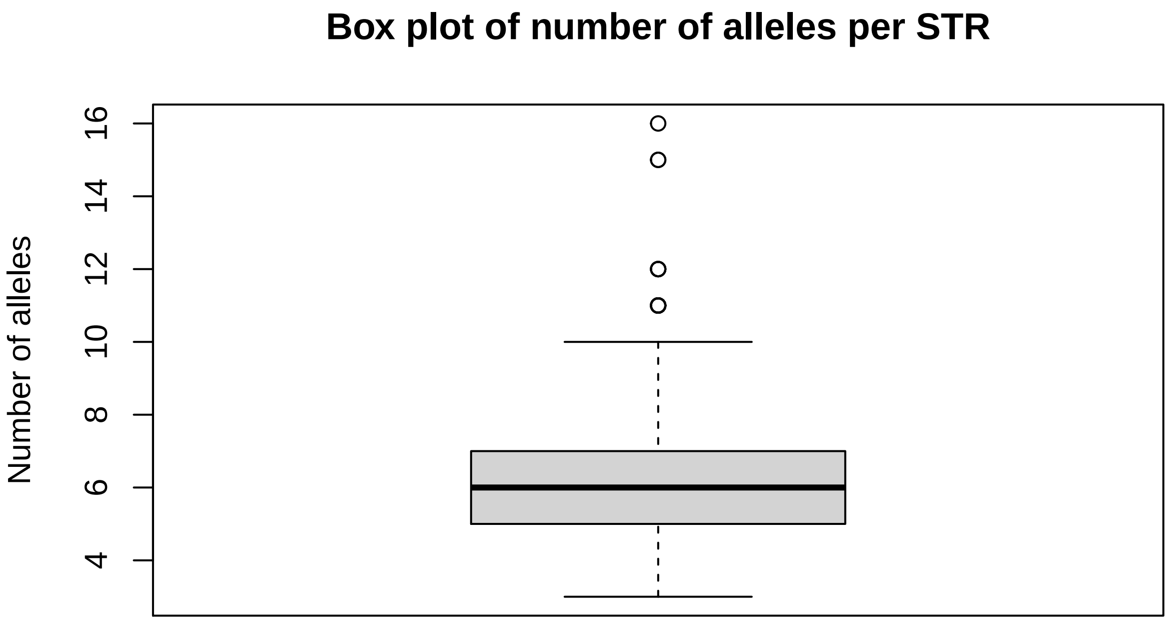
```r
cat('Other descriptive statistics: \n')
```

```
## Other descriptive statistics:
```

```r
summary(n.alleles.per.STR)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   3.000   5.000   6.000   6.375   7.000  16.000
```

```r
boxplot(n.alleles.per.STR, main='Box plot of number of alleles per STR', ylab='Number of alleles')
```

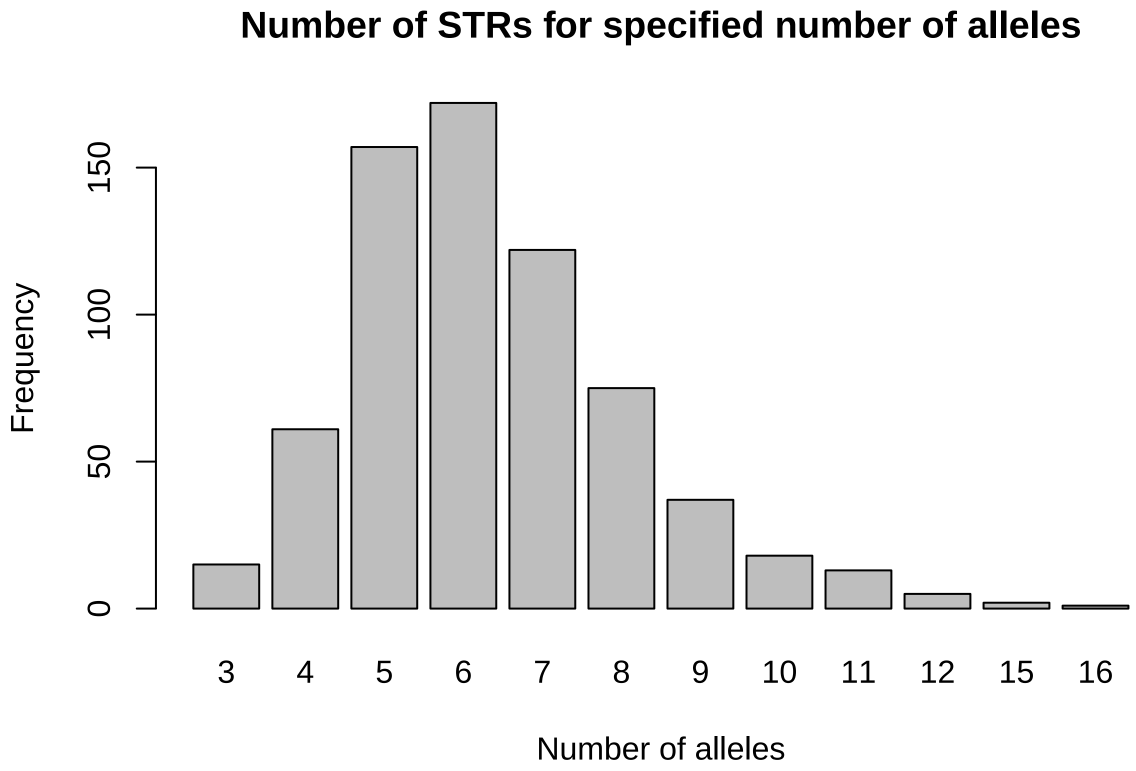


5. (2p) Make a table with the number of STRs for a given number of alleles and present a barplot of the number STRs in each category. What is the most common number of alleles for an STR?

```r
allele.counts = sort(unique(n.alleles.per.STR))
str.counts = c()

for(c in allele.counts) {
  num = sum(n.alleles.per.STR == c)
  str.counts = append(str.counts, num)
}

barplot(str.counts, names.arg=allele.counts, main='Number of STRs for specified number of alleles', xlab
```



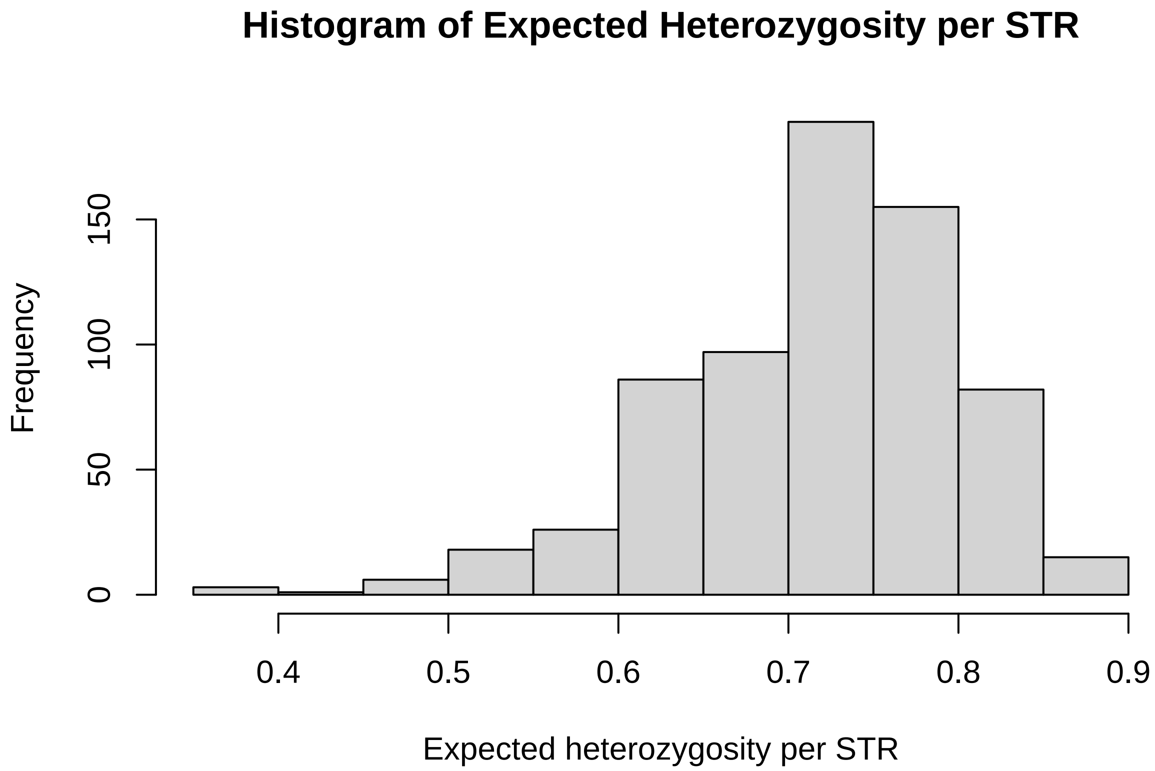
Most common number of alleles for a STR is 6.

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.

```
h.expected.str <- function(x) {
  x = x[!is.na(x)]
  tbl = table(x)
  total = sum(tbl)
  result = 1.0
  for(t in tbl) {
    result = result - (t/total)^2
  }
  result
}


h.expected.per.STR = apply(dataset, 2, h.expected.str)
he.df <- data.frame(h.expected.per.STR)
names(he.df)=c('Expected heterozygosity')
#(he.df)


hist(h.expected.per.STR, main='Histogram of Expected Heterozygosity per STR', xlab='Expected heterozygo
```



```r
cat(paste('Average expected heterozygosity over all STR: ', mean(h.expected.per.STR)))
```

```
## Average expected heterozygosity over all STR:  0.717266239372483
```

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?

When comparing SNP and STR databases, we can observe that SNP database has much more variants (16393) then STR database (678). While the SNP database does not have any missing values, there is 4.2% of data missing in the STR database.

Variants of SNP contain only 2 different alleles and therefore 3 different genotypes, while average number of alleles in STR variants is around 6. Higher number of alleles is the reason why STR's expected heterozygosity can theoretically be and is higher than SNP's.