B_01c.genomics

May 14, 2020

1 Bayesian methods in ecology and evolution

https://bitbucket.org/mfumagal/statistical_inference

1.1 day 1c: Bayesian applications in genomics

1.1.1 Reconstructing genomes from sequencing data

You are going to develop and implement a Bayesian approach to reconstruct genomes from data produced from high-throughput sequencing machines.

Specifically, you will be doing **genotype calling** from short-read NGS data.

Load the *R* functions needed with `source("Data/functions.R")`.

Among these functions, we provide one that calculates the likelihood of a certain sequence of bases for diploid individuals. This function is called *calcGenoLikes* and takes 5 paramaters in input: * the sequence itself (collection of bases) * the first allele of the genotype * the second allele of the genotype * the sequencing error rate * a boolean indicating whether the results should be returned in logarithmic scale (TRUE) or not (FALSE)

For instance, assuming that your sequence is `AAGAGGA`, your alleles are `A` and `G` (meaning that you want to calculate the likelihood for genotypes `{AA,AG,GG}`, and your sequencing error rate is 0.05, then the likelihood (not in logarithms) for each genotype is given by `calcGenoLikes("AAGAGGA", "A", "G", 0.05, FALSE)`

Complete all the following tasks using *R* when necessary. The key point of these exercises is to not recalculate quantities that you have already computed. The aim is that you should be able to

ICL

understand whether the likelihood or the prior is the same (or not) between different scenarios.

A)

Using Bayes' theorem, write the formula for the posterior probability of genotype G being AA given the sequencing data D. Write the explicit denominator assuming that your alleles are A and G and all possible genotypes are only AA, AG, GG.

```
In [5]: source("../Math/Data/functions.R")
        data <- "AAGAGGA"
        error_rate=0.05
        #function for likelihood
        calcGenoLikes("AAGAGGA", "A", "G", error_rate, FALSE)
        #function for prior
        prior <- function(allele_1,allele_2){
          prior <- c(1/3,1/3,1/3)
          names(prior) <- c(paste0(allele_1,allele_1),paste0(allele_1,allele_2),
                            paste0(allele_2,allele_2))
          return(prior)
        }
        #Posterior distribution
        prob_G_given_D <- function(allele_1,allele_2){
          posterior <- c()
          x <- calcGenoLikes(data, "A", "G", error_rate, FALSE)
          likelihood_value <- c() #likelihood value desired
          if (allele_1 == "A" & allele_2 =="A"){
            likelihood_value <- x[1]
          }else if (allele_1 == "G" & allele_2 == "G"){
            likelihood_value <- x[3]
          }else{
            likelihood_value <- x[2]
          }
          posterior <- (likelihood_value * prior(allele_1,allele_2)[1]) /
            sum(x*prior(allele_1,allele_2))
          return(posterior)
        }
        prob_G_given_D("A","A")
```

AA 3.77086226851852e-06 **AG** 0.00616207858546239 **GG** 6.61554783950617e-08

AA: 0.000611565663752487

B)

Assuming that your data is `AAAG`, your alleles are A and G, and the sequencing error rate is 0.01, calculate genotype posterior probability using a uniform prior, e.g. $P(G = AA) = P(G = AG) = P(G = GG) = ?$

```
In [9]: data <- "AAAG"
        error_rate = 0.01
        B_AA=prob_G_given_D("A","A")
        B_AA
        B_AG=prob_G_given_D("A","G")
```

```
    B_AG
    B_GG=prob_G_given_D("G","G")
    B_GG
```

AA: 0.0504699336283693
AG: 0.949529494208545
GG: 5.72163085721063e-07
C)

With the same assumptions as in point B, calculate genotype posterior probabilities using prior probabilitties based on Hardy Weinberg Equilibrium with a frequency of G of 0.1. Do you need to calculate a new likelihood or is it the same one as in point B?

```
In [20]: #HWE where f(G) = 0.1 thus f(A) = 0.9
         #GG=0.1^2, AA=0.9^2 and AG=2(0.1)(0.9)
         data <- "AAAG"
         error_rate = 0.01
         prior <- function(allele_1,allele_2){
           prior <- c(0.9^2,2*(0.1)*(0.9),(0.1)^2)
           names(prior) <- c(paste0(allele_1,allele_1),paste0(allele_1,allele_2),
                             paste0(allele_2,allele_2))
           return(prior)
         }
         C_AA=(prob_G_given_D("A","A"))
         C_AA
         C_AG=(prob_G_given_D("A","G"))
         C_AG
         C_GG=(prob_G_given_D("G","G"))
         C_GG
         sum(c(C_AA,C_AG,C_GG))
```

AA: 0.19301900783074
AG: 0.806980965154435
GG: 2.7014825176113e-08
1
D)

With the same assumptions as in point C, calculate genotype posterior probabilities using a prior based on Hardy Weinberg Equilibrium with a frequency of G of 0.1 and an inbreeding coefficient of 0.2. In this case, we need to modify our previous priors. Specifically, if f is the frequency of allele A and I is the inbreeding coefficient, then the prior probabilities for all genotypes are: * $p(AA) = f^2 + I \times f \times (1-f)$ * $p(AG) = 2 \times f \times (1-f) \times (1-I)$ * $p(GG) = (1-f)^2 + I \times f \times (1-f)$

Do you need to calculate a new likelihood or is it the same one as in points B and C?

```
In [19]: #HWE where f(G) = 0.1 thus f(A) = 0.9
         data <- "AAAG"
         error_rate = 0.01
         prior <- function(allele_1,allele_2){
           prior <- c((0.9^2 + 0.2 * 0.9 * 0.1),(2*(0.1)*(0.9) * (1-0.2)),((0.1)^2)
                      + 0.2 * 0.9 * 0.1)
           names(prior) <- c(paste0(allele_1,allele_1),paste0(allele_1,allele_2),
                             paste0(allele_2,allele_2))
           return(prior)
         }
         D_AA=(prob_G_given_D("A","A"))
         D_AA
         D_AG=(prob_G_given_D("A","G"))
         D_AG
         D_GG=(prob_G_given_D("G","G"))
         D_GG
         sum(c(D_AA,D_AG,D_GG))
```

AA: 0.23408461103795
AG: 0.76591529922172
GG: 8.97403294571994e-08
1

E)

With the same priors used in point D but with a sequencing error rate of 0.05, calculate the genotype posterior probabilities. Do you need to calculate a new likelihood or is it the same one as in point D?

```
In [18]: data <- "AAAG"
         error_rate = 0.05
         prior <- function(allele_1,allele_2){
           prior <- c((0.9^2 + 0.2 * 0.9 * 0.1),(2*(0.1)*(0.9) * (1-0.2)),((0.1)^2)
                      + 0.2 * 0.9 * 0.1)
           names(prior) <- c(paste0(allele_1,allele_1),paste0(allele_1,allele_2)
                             ,paste0(allele_2,allele_2))
           return(prior)
         }
         E_AA=(prob_G_given_D("A","A"))
         E_AA
         E_AG=(prob_G_given_D("A","G"))
         E_AG
         E_GG=(prob_G_given_D("G","G"))
         E_GG
         sum(c(E_AA,E_AG,E_GG))
```

AA: 0.600885166818381
AG: 0.399108579014721
GG: 6.25416689747521e-06
1

F)

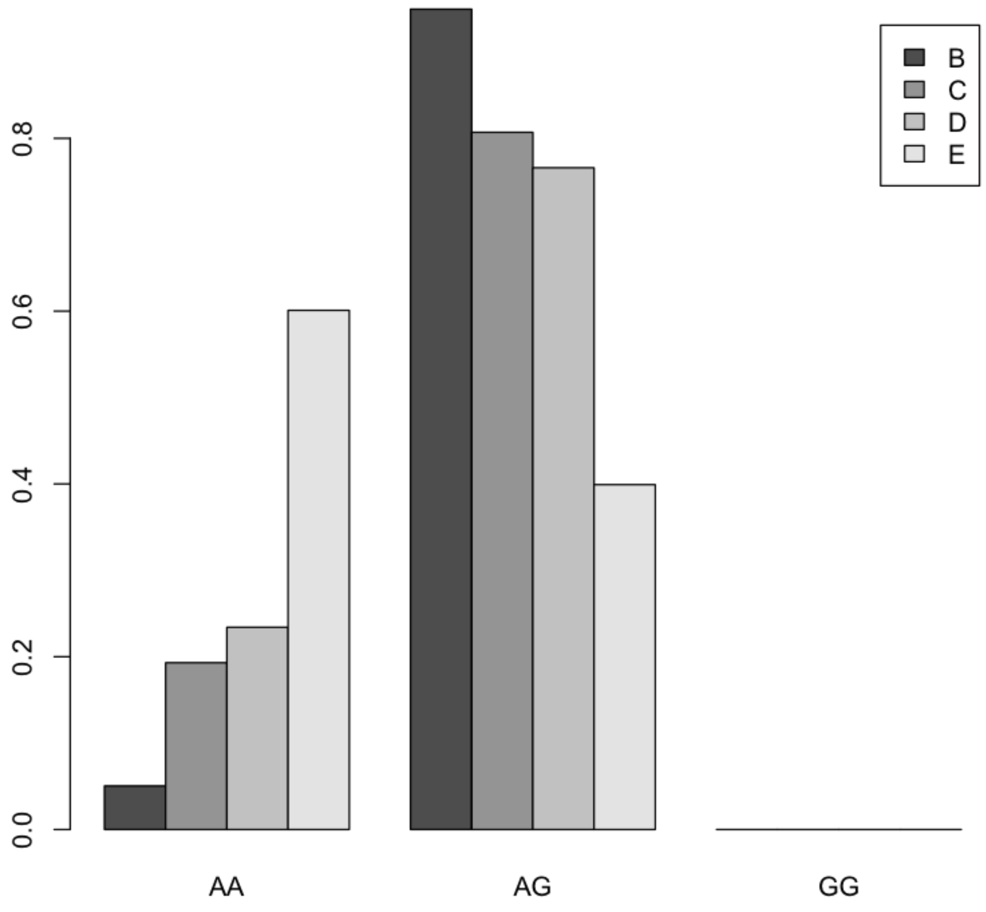
Plot all previous results (e.g. use a barplot with the 3 posterior probabilities for each scenario B-E).

```
In [29]: AA <- c(B_AA,C_AA,D_AA,E_AA)
         AG <- c(B_AG,C_AG,D_AG,E_AG)
```

```r
GG <- c(B_GG,C_GG,D_GG,E_GG)
result_matrix <- matrix(c(AA,AG,GG),nrow = 4,ncol = 3)
colnames(result_matrix) <- c("AA","AG","GG")
rownames(result_matrix) <- c("B","C","D","E")
result_matrix
barplot(result_matrix,beside=TRUE, legend=TRUE)
```

A matrix: 4 Œ 3 of type dbl

	AA	AG	GG
B	0.05046993	0.9495295	5.721631e-07
C	0.19301901	0.8069810	2.701483e-08
D	0.23408461	0.7659153	8.974033e-08
E	0.60088517	0.3991086	6.254167e-06



G)

Assuming that our collection of sequenced bases is `AAAGAGAAAAAAAGGGGGAAAGGA`, calculate the genotype posterior probabilities using the same priors as in point C and sequencing error rate of 0.05. What happens if we have more data? What is the **confidence** in our genotype inference?

```
In [26]: data <- "AAAGAGAAAAAAAGGGGGAAAGGA"
         error_rate = 0.05
         prior <- function(allele_1,allele_2){
           prior <- c(0.9^2,2*(0.1)*(0.9),(0.1)^2)
           names(prior) <- c(paste0(allele_1,allele_1),paste0(allele_1,allele_2),
                             paste0(allele_2,allele_2))
           return(prior)
         }
         #Posterior distribution
         prob_G_given_D <- function(allele_1,allele_2){
           likelihood <- calcGenoLikes(data, "A", "G", error_rate, FALSE)
           if (allele_1 == "A" & allele_2 =="A"){
             likelihood_value <- likelihood[1]
           }else if (allele_1 == "G" & allele_2 == "G"){
             likelihood_value <- likelihood[3]
           }else{
             likelihood_value <- likelihood[2]
           }
           y <- prior(allele_1,allele_2)
           if (allele_1 == "A" & allele_2 =="A"){
             prior_value <- y[1]
           }else if (allele_1 == "G" & allele_2 == "G"){
             prior_value <- y[3]
           }else{
             prior_value <- y[2]
           }
           posterior <- (likelihood_value * prior_value) /
             sum(likelihood*prior(allele_1,allele_2))
           return(posterior)
         }
         G_AA=(prob_G_given_D("A","A"))
         G_AG=(prob_G_given_D("A","G"))
         G_GG=(prob_G_given_D("G","G"))
         G_AA
         G_AG
         G_GG
         sum(G_AA,G_AG,G_GG)
```

AA: 7.83039005124002e-09
AG: 0.99999999216961
GG: 2.81870251493449e-21
1

H)

What happens if we have a lot of data? Assume that your sequenced bases are `bases <-`

`paste(c(rep("A",1e3),rep("G",1e3)), sep="", collapse="")`. Calculate the genotype likelihoods for this data. What is happening here?

It is convenient to use numbers in log-scale and you can do that by selecting TRUE as the last parameter in the *calcGenoLikes*. Remember that if you want to calculate proper probabilities (in log) you have to approximate the sum of logs.

Without calculating posterior probabilities, what is the effect of the prior here in your opinion?

```
In [31]: bases <- paste(c(rep("A",1e3),rep("G",1e3)), sep="", collapse="")
         calcGenoLikes(bases, "A", "G", error_rate, TRUE)
         print("The prior will have greater effect on the posterior as the
         likelihood log values are highly negative")
```

AA -4145.63785660962 **AG** -1454.09746447119 **GG** -4145.63785660957

```
[1] "The prior will have greater effect on the posterior as the likelihood log values are highl
```

```
In [ ]:
```

B_02c.population

February 28, 2020

1 Bayesian methods in ecology and evolution

https://bitbucket.org/mfumagal/statistical_inference

1.1 day 2c: population inferences from finite samples

We sequenced several genomes of bears and assigned each individual genotype. What is the frequency of a certain allele at the population level?

We have only a sample of the entire population of bears but we want to make inferences at the whole population level.

Our sample contains information for 100 individuals with the following genotypes: 63 AA, 34 AG, 3 GG. A frequentist estimate of the frequency of G is given by: $(34 + (3 \times 2))/200 = 40/200 = 0.20$.

What is the posterior distribution for the population frequency of G?

The first thing we need to do is define our likelihood model. We can imagine to randomly sample one allele from the population and each time the allele can be either G or not.

This is a set of Bernoulli trials and we can use of Binomial distribution as likelihood function.

The Binomial likelihood is

$$P(k|p,n) = \binom{n}{k} p^k (1-p)^{n-k} \tag{1}$$

where k is the number of successes (i.e. the event of sampling a G), p is the proportion of G alleles we have (i.e. the probability of a success), and n is the number of alleles we sample.

ICL

Recall that

$$\binom{n}{k} = \frac{n!}{k!(n-k)!} \tag{2}$$

Note that the combinatorial term does not contain p.

What is the maximum likelihood estimate of p?

You may recall that it is $\hat{p} = \frac{k}{n}$. Note that the combinatorial terms does not affect this estimate.

The second thing we need to do is define a prior probability for p.

What is the interval of values that p can take?

It is $[0, 1]$, as we express frequencies relative to the whole population/sample. It is convenient to choose a prior distribution which is conjugate to the Binomial.

A Beta distribution is a conjugate prior in this case.

Are certain values of p more likely to occur without observing the data?

If that it is not the case, can we use the Beta distribution to generate a noninformative prior?

We can choose $Beta(\alpha = 1, \beta = 1)$, which is defined as

$$P(p) = \frac{1}{B(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{3}$$

where $\frac{1}{B(\alpha,\beta)}$ is simply a normalisation term which does not depend on p.

The full model can be expressed as $P(p|k,n) \approx P(k|p,n)P(p)$.

The closed form for the posterior distribution given our choices for the likelihood and prior functions is

$$P(p|k,n) \approx p^{k+\alpha-1}(1-p)^{n-k+\beta-1} \tag{4}$$

The posterior distribution (Beta-Binomial model) is a Beta distribution with parameters $k+\alpha$ and $n-k+\beta$.

If we set $\alpha = \beta = 1$ then $P(p|k,n) = Beta(k+1, n-k+1)$. What are k and n?

n is the number of alleles we sample and k is the occurence of allele G in our sample.

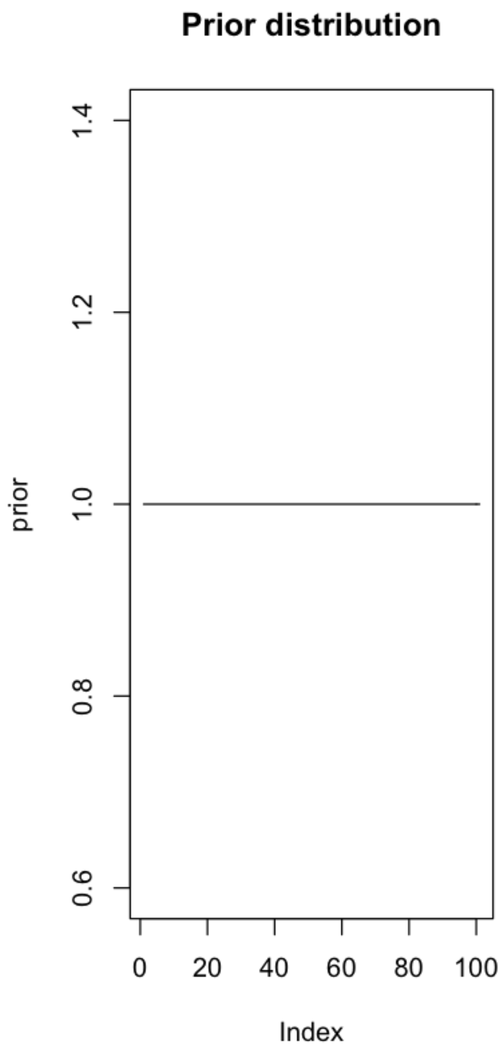
A)

Plot the posterior probability. Then calculate the maximum a posteriori value, 95% credible intervals, and notable quantiles.

What happens to the distribution if we have only 10 samples (with the sample allele frequency of 0.20)?

```
In [4]: #A
        #10 samples
        library(coda)
```

```
alpha <- 1
beta <- 1
n <- 10
K <- 2
x <- seq(0,1,0.01)

prior <- dbeta(x=x, shape1= alpha, shape2=beta)
par(mfrow=c(1,2))
plot(prior,main="Prior distribution",type="l")
likelihood <- dbinom(x=K, size=n, prob=x)
posterior_prob <- dbeta(x=x, shape1 = alpha + K, shape2= beta + n - K)
```



```
In [5]: MAP=x[which(posterior_prob==max(posterior_prob))]
        print(paste("MAP=",MAP))
```

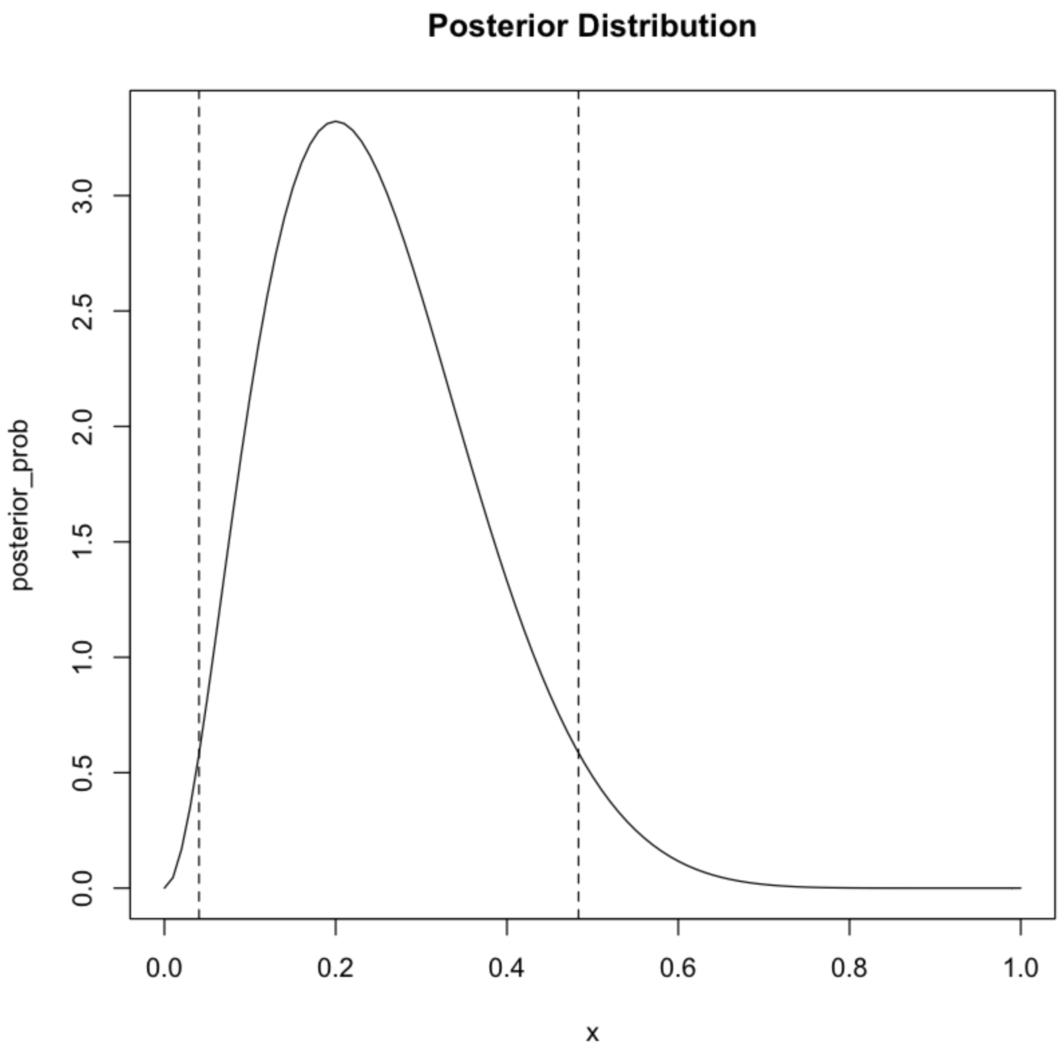
```
plot(x,posterior_prob,type="l",main="Posterior Distribution")

a <- rbeta(n=1e6, shape1=alpha + K, shape2=beta + n - K)
hpd <- HPDinterval(as.mcmc(a), prob=0.95)
hpd
abline(v=hpd, lty=2)

quantile(a)
```

```
[1] "MAP= 0.2"
```

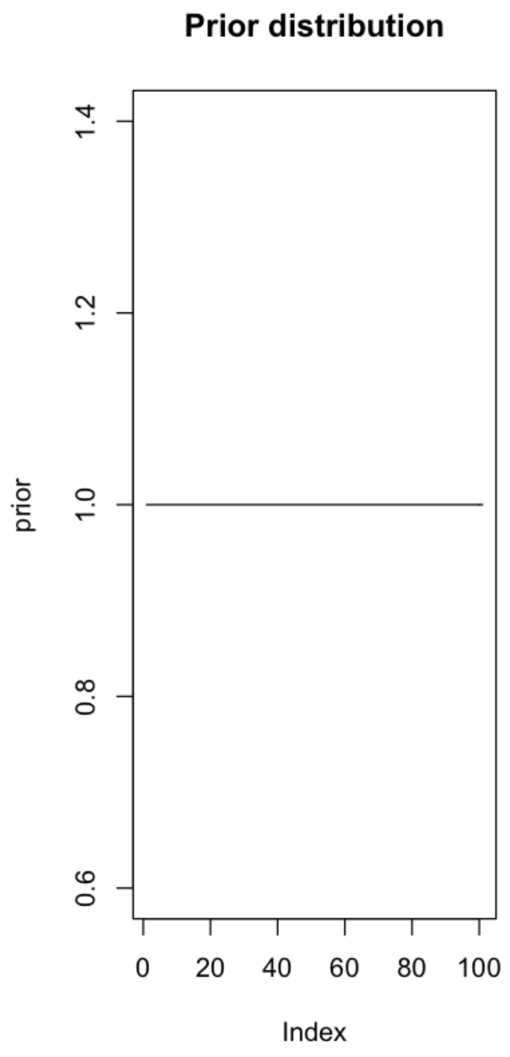
A matrix: 1 Œ 2 of type dbl

	lower	upper
var1	0.04039855	0.483534

0\% 0.00173026332411134 **25\%** 0.159589763808141 **50\%** 0.235861725827577 **75\%** 0.326248152138632 **100\%** 0.854277203683713



```
In [2]: #100 samples
        library(coda)
        alpha <- 1
        beta <- 1
        n <- 100
        K <- 20
        x <- seq(0,1,0.01)

        prior <- dbeta(x=x, shape1= alpha, shape2=beta)
        par(mfrow=c(1,2))
        plot(prior,main="Prior distribution",type="l")
        likelihood <- dbinom(x=K, size=n, prob=x)
        posterior_prob <- dbeta(x=x, shape1 = alpha + K, shape2= beta + n - K)
        MAP=x[which(posterior_prob==max(posterior_prob))]
```



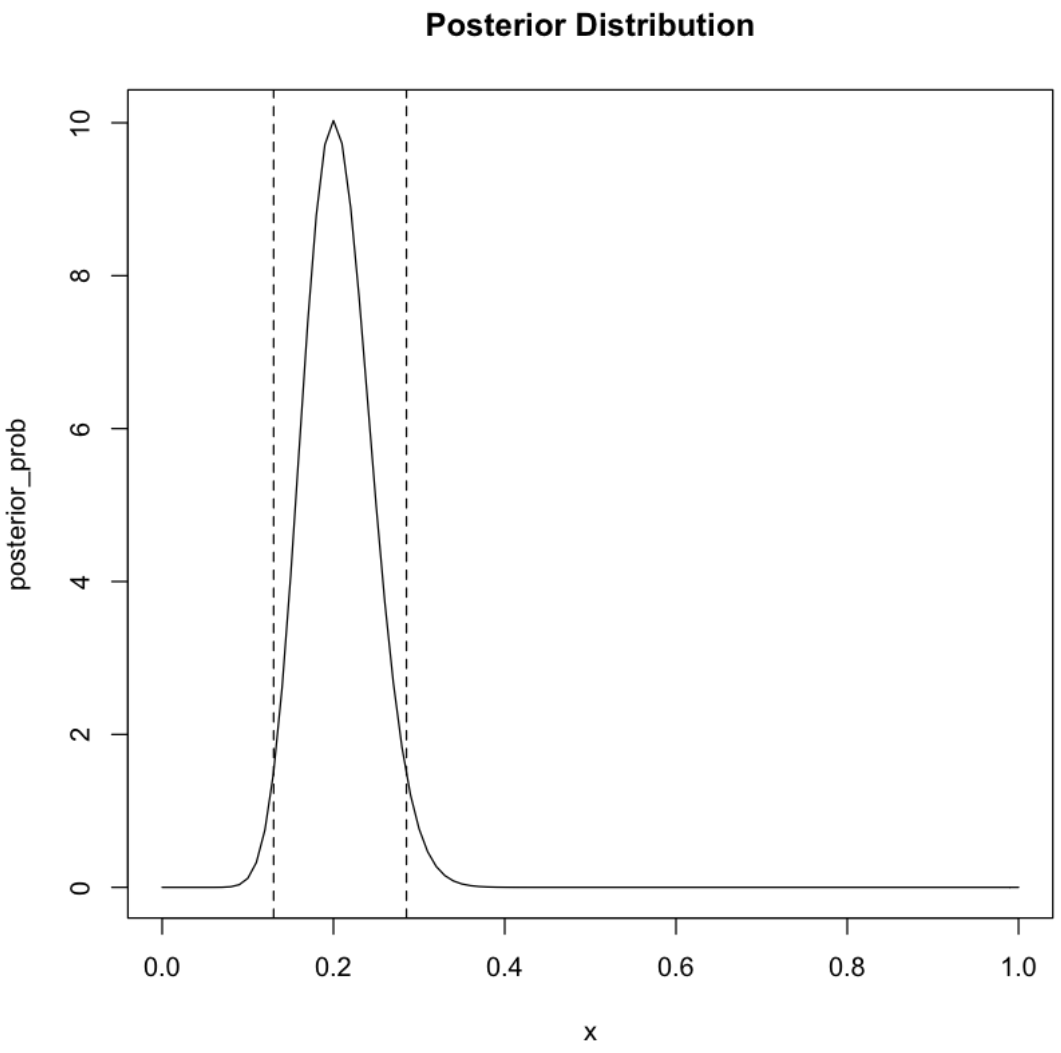
```
In [3]: plot(x,posterior_prob,type="l",main="Posterior Distribution")

        a <- rbeta(n=1e6, shape1=alpha + K, shape2=beta + n - K)
        hpd <- HPDinterval(as.mcmc(a), prob=0.95)
        hpd
        abline(v=hpd, lty=2)

        print(paste("MAP=",MAP))
        quantile(a)
```

A matrix: 1 Œ 2 of type dbl

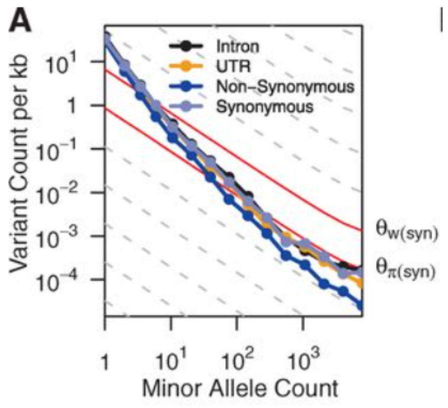
	lower	upper
var1	0.1301854	0.2852487

```
[1] "MAP= 0.2"
```

0\% 0.0627309514295545 **25\%** 0.177974213473729 **50\%** 0.203981847320473 **75\%** 0.231770138459138 **100\%** 0.413942957194519

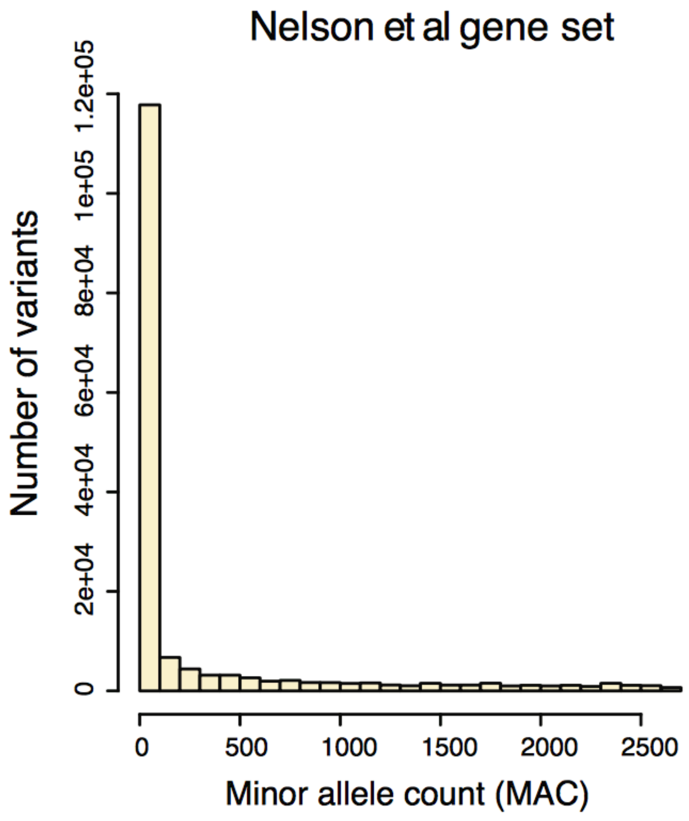


We can think of a more informative prior. The genome-wide distribution of allele frequencies for human populations as a particular shape. This is called a site frequency spectrum (SFS) or



allele frequency spectrum (AFS).

We can have another view at it by plotting the minor allele counts (MAC) distribution.



Does this distribution fit with a uniform prior? Can we use a conjugate (Beta) function to model this distribution?

Also, we don't know *a priori* whether the allele we are interested in is the minor allele. Therefore a prior distribution with more density at both low and high frequencies might be more appropriate.

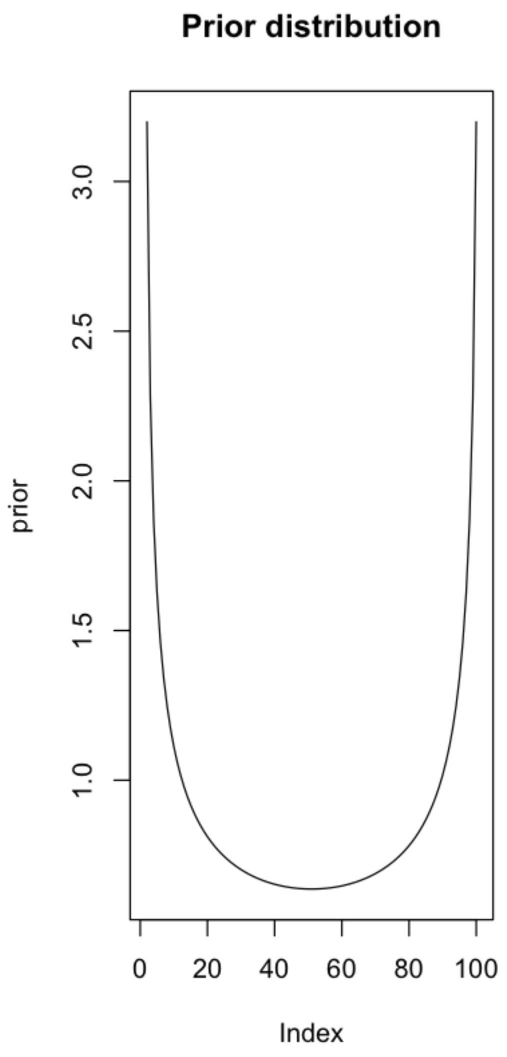
B)

Recalculate the posterior distribution of p using an informative prior (make your own choices regarding the parameter for the Beta distribution) both in the case of 100 and 10 samples.

Discuss how these results compare to the previous ones obtained in point A.

```
In [6]: #10 samples
```

```r
library(coda)
alpha <- 0.5
beta <- 0.5
n <- 10
K <- 2
x <- seq(0,1,0.01)

prior <- dbeta(x=x, shape1= alpha, shape2=beta)
par(mfrow=c(1,2))
plot(prior,main="Prior distribution",type="l")
likelihood <- dbinom(x=K, size=n, prob=x)
posterior_prob <- dbeta(x=x, shape1 = alpha + K, shape2= beta + n - K)
```



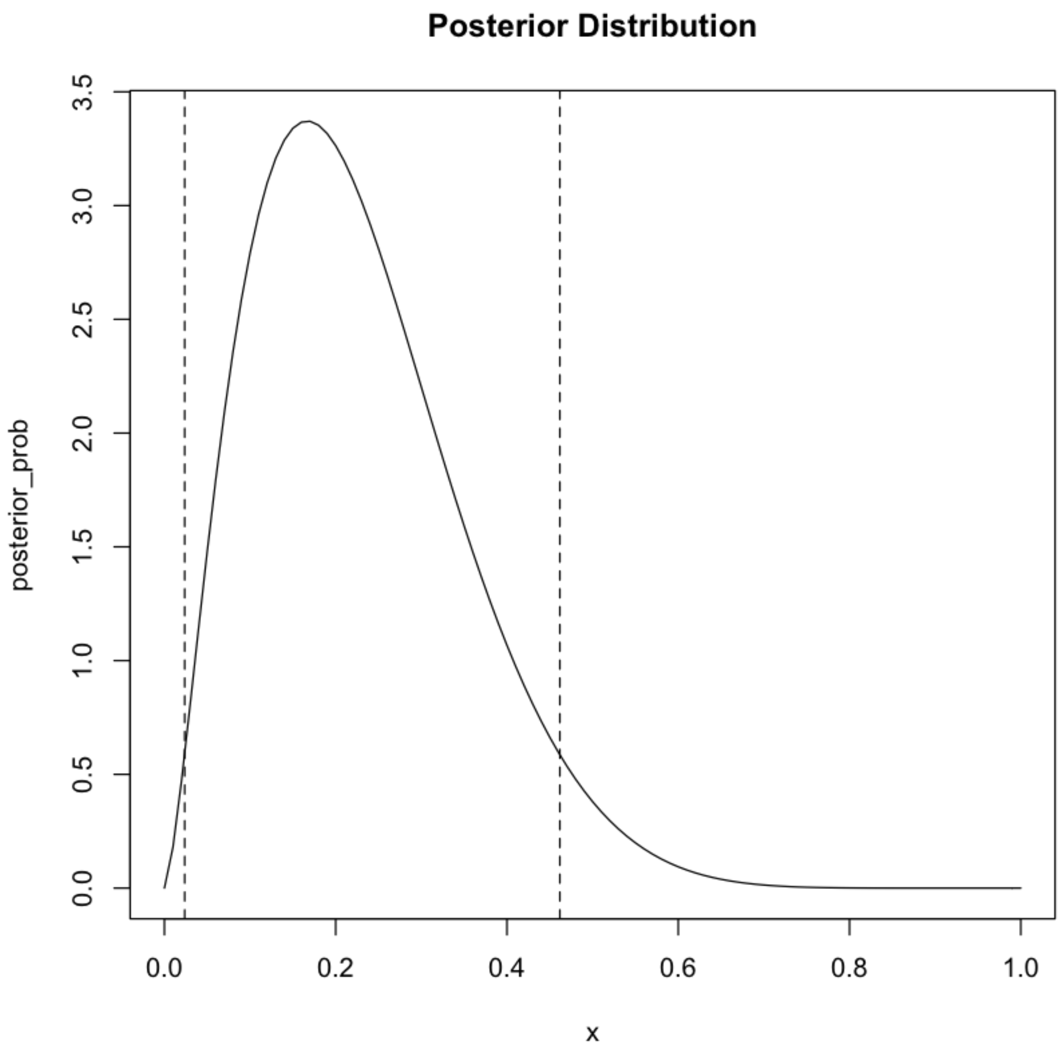
```
In [7]: MAP=x[which(posterior_prob==max(posterior_prob))]
        print(paste("MAP=",MAP))

        plot(x,posterior_prob,type="l",main="Posterior Distribution")

        a <- rbeta(n=1e6, shape1=alpha + K, shape2=beta + n - K)
        hpd <- HPDinterval(as.mcmc(a), prob=0.95)
        hpd
        abline(v=hpd, lty=2)

        quantile(a)
```

```
[1] "MAP= 0.17"
```

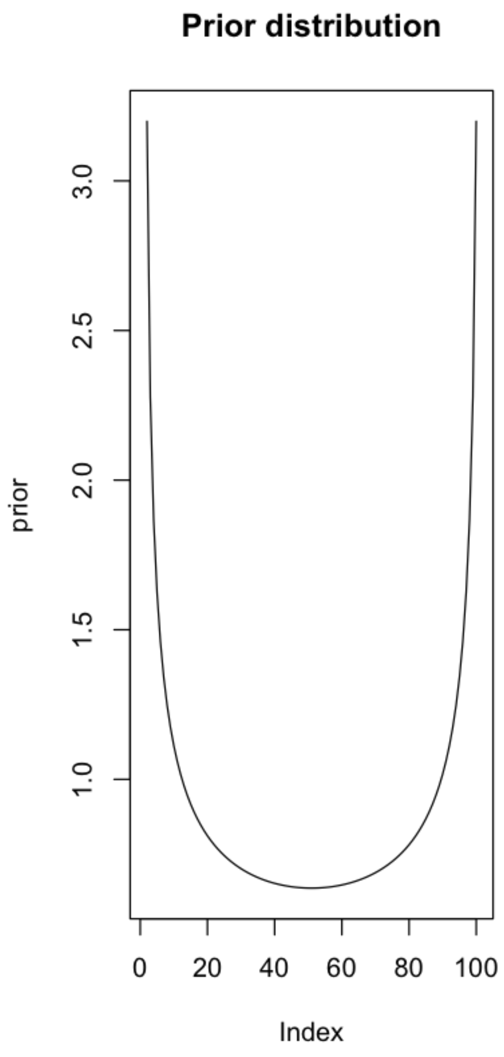
A matrix: 1 Œ 2 of type dbl

	lower	upper
var1	0.02367622	0.4617672

0\% 0.000798628598841223 **25\%** 0.135095392000748 **50\%** 0.210434729782217 **75\%** 0.302258185344709 **100\%** 0.870692372333148



```
In [8]: #100 samples
        library(coda)
        alpha <- 0.5
        beta <- 0.5
        n <- 100
        K <- 20
        x <- seq(0,1,0.01)

        prior <- dbeta(x=x, shape1= alpha, shape2=beta)
        par(mfrow=c(1,2))
        plot(prior,main="Prior distribution",type="l")
        likelihood <- dbinom(x=K, size=n, prob=x)
        posterior_prob <- dbeta(x=x, shape1 = alpha + K, shape2= beta + n - K)
```



```
In [9]: MAP=x[which(posterior_prob==max(posterior_prob))]


        plot(x,posterior_prob,type="l",main="Posterior Distribution")

        a <- rbeta(n=1e6, shape1=alpha + K, shape2=beta + n - K)
        hpd <- HPDinterval(as.mcmc(a), prob=0.95)
        hpd
        abline(v=hpd, lty=2)

        print(paste("MAP=",MAP))
        quantile(a)
```

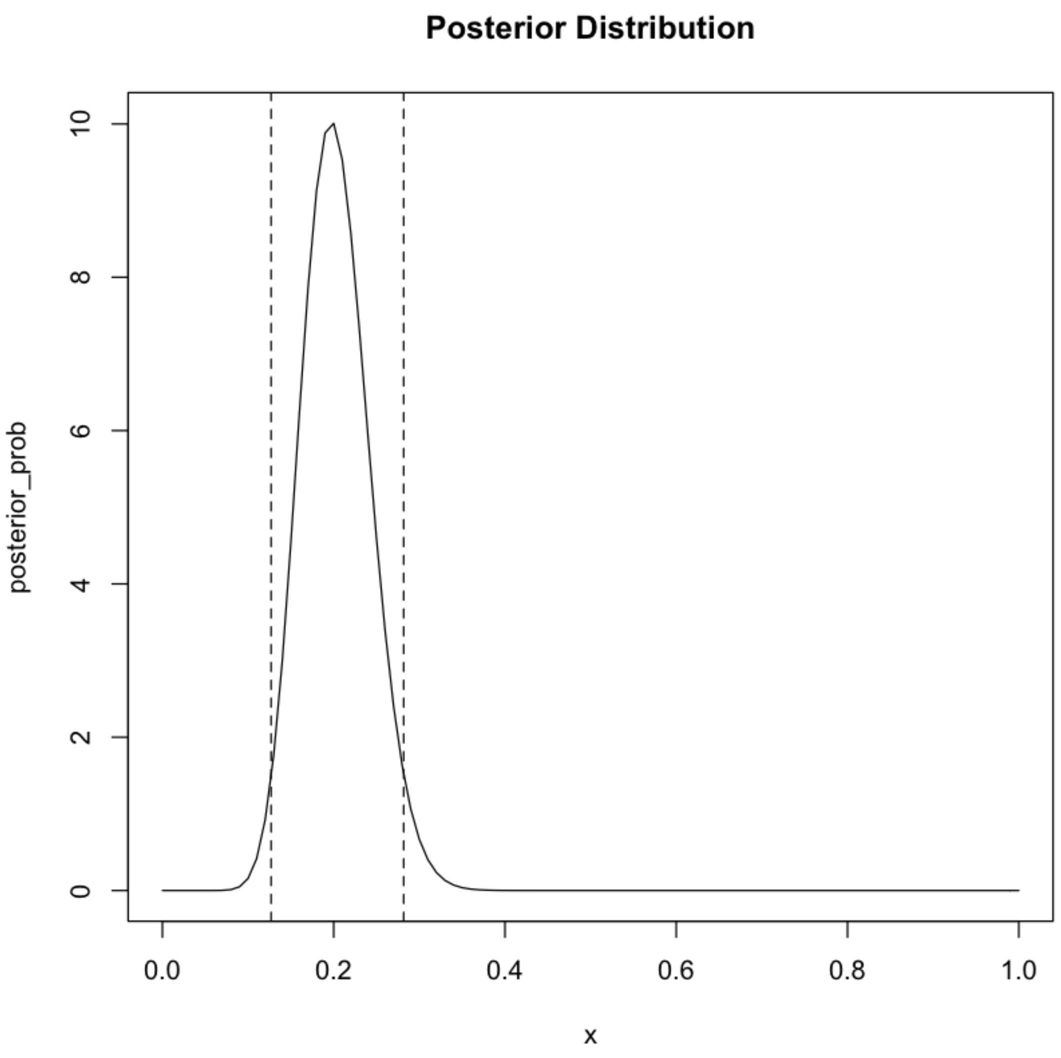
```
#changing from a uniform prior to a informative prior affected the distribution of the
#resulted in a the quantiles becomes larger, indicating that the the distribution has
#matters most for the sample size of 10 as the effect on the distribution will be grea
```

A matrix: 1 Œ 2 of type dbl

	lower	upper
var1	0.1269365	0.2817281

```
[1] "MAP= 0.2"
```

0\% 0.0514122186626254 **25\%** 0.175002181364613 **50\%** 0.200997497400279 **75\%** 0.228820184071584 **100\%** 0.450658353952743



C)

Calculate the Bayes factor for a model with $p <= 0.5$ vs a model with $p > 0.5$. Note that these models are equally probable a priori.

```
In [10]: library(coda)
         alpha <- 0.5
         beta <- 0.5
         n <- 10
         K <- 2
         posterior_prob1 <- pbeta(0.5, shape1= alpha + K , shape2 = beta + n - K)
         posterior_prob2 = 1 - posterior_prob1
         ratio = posterior_prob1 / posterior_prob2
         print(paste("BF=",ratio))
```

```
[1] "BF= 37.4074507188568"
```

B_04b.speciation

March 16, 2020

1 Bayesian methods in ecology and evolution

https://bitbucket.org/mfumagal/statistical_inference

1.1 day 4b: Bayesian estimation of speciation times

Preparation For this practical you need some R packages, namely `coda`, `abc`, `maps`, `spam`, `fields`. For plotting purposes you may also want to use `ggplot2`.

You will also need the software `ms` to be installed. You can find the executable for linux in `bin/ms`. If it doesn't work, you can compile the source `Software/ms.tar.gz` by `tar -xzvf ms.tar.gz; cd msdir; gcc -o ms ms.c streec.c rand1.c -lm`. Finally you need some data and R functions provided in `Data`.

I suggest to copy `functions.R` and `polar.brown.sfs*` in the workspace where you will run this practical without overwriting the repository.

You can work in teams for this practical. Actually, I encourage that you team up for this exercise (max 4 students per submission). Each student will have to submit her/his solution but with a clear indication of CID numbers of all members of the group.

Project In this practical you are going to estimate the divergence (or speciation) time between polar bears and brown bears using genomics data. You will be using Approximate Bayesian Computation (ABC) methods to inference such time.

ICL

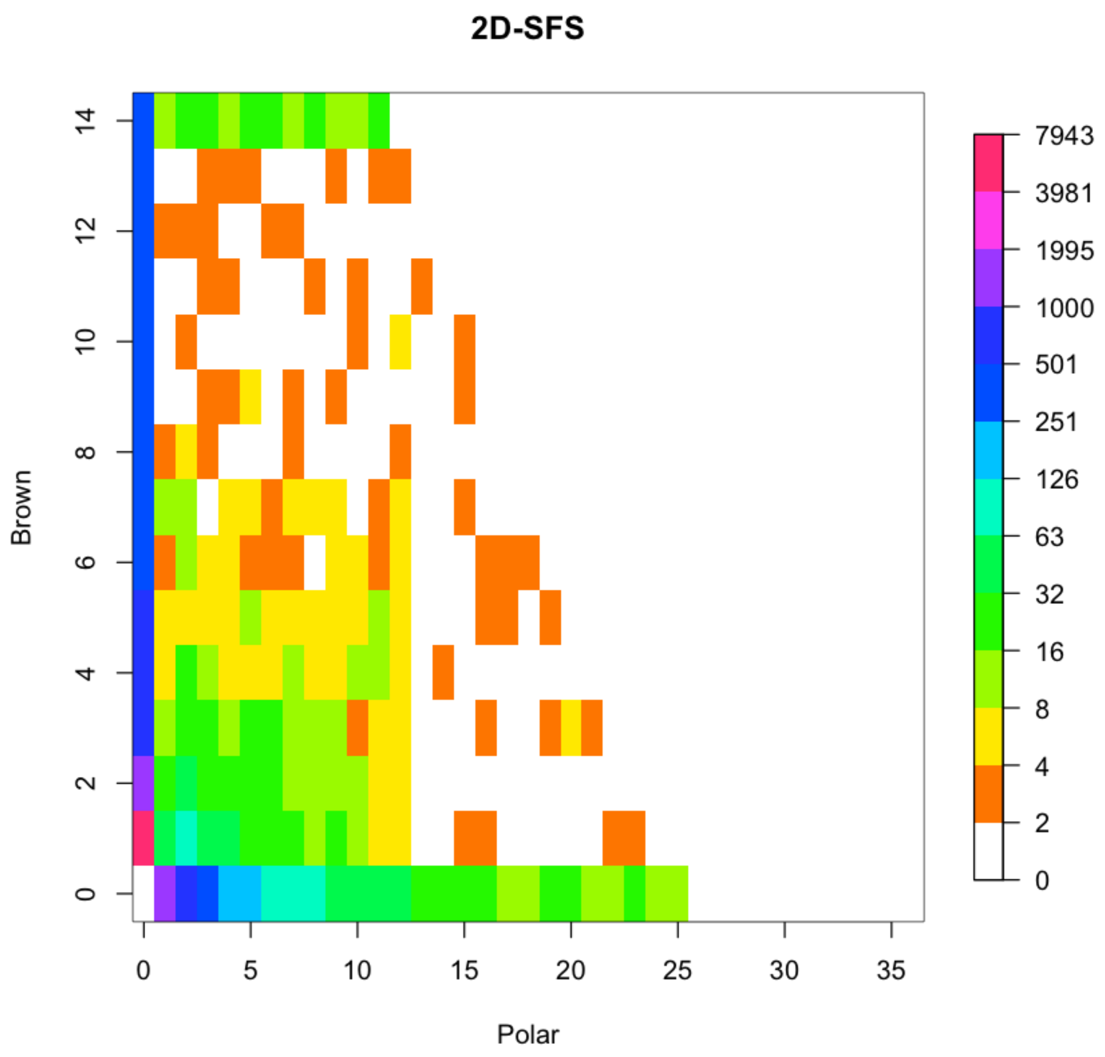
```
In [17]: # Open R and load all R functions and data needed:
         source("../Math/Data/functions.R")
         load("Data/polar.brown.sfs.Rdata")

         # Inspect the objects:
         ls()
```

1. 'binomial.likelihood' 2. 'calcGenoLikes' 3. 'calcSummaryStats' 4. 'Challenge_A' 5. 'Challenge_B' 6. 'Challenge_C' 7. 'Challenge_D' 8. 'Challenge_E' 9. 'Challenge_F' 10. 'chaos_game' 11. 'choose_two' 12. 'cluster_run' 13. 'count' 14. 'data' 15. 'doFST' 16. 'draw_fern' 17. 'draw_fern2' 18. 'draw_spiral' 19. 'draw_tree' 20. 'elbow' 21. 'email' 22. 'f' 23. 'fern' 24. 'fern2' 25. 'first' 26. 'first_scene' 27. 'fold2DSFS' 28. 'fout' 29. 'fromMStoSFS' 30. 'i' 31. 'init_community_max' 32. 'init_community_min' 33. 'intrinsic' 34. 'k' 35. 'leaf_area' 36. 'log.binomial.likelihood' 37. 'lt' 38. 'mat' 39. 'msDir' 40. 'N' 41. 'nam' 42. 'name' 43. 'nams' 44. 'nChroms.brown' 45. 'nChroms.polar' 46. 'neutral_generation' 47. 'neutral_generation_speciation' 48. 'neutral_step' 49. 'neutral_step_speciation' 50. 'neutral_time_series' 51. 'neutral_time_series_speciation' 52. 'nrSimul' 53. 'nrSites' 54. 'obsSummaryStats' 55. 'octaves' 56. 'p' 57. 'personal_speciation_rate' 58. 'plot2DSFS' 59. 'polar.brown.sfs' 60. 'preferred_name' 61. 'process_cluster_results' 62. 'question_12' 63. 'question_16' 64. 'question_21' 65. 'question_22' 66. 'question_8' 67. 'r' 68. 'res_sum' 69. 'res2' 70. 'result' 71. 'reynolds' 72. 'Ricker' 73. 'scale_data' 74. 'scale_sim_fst' 75. 'sd' 76. 'second' 77. 'second_scene' 78. 'sim_fst' 79. 'simulate' 80. 'simulatedSFS' 81. 'species_abundance' 82. 'species_richness' 83. 'spiral' 84. 'spongy_mesophyll' 85. 'stochrick' 86. 'stochrickvect' 87. 'sum_vect' 88. 't' 89. 'target' 90. 'third' 91. 'third_scene' 92. 'tree' 93. 'turtle' 94. 'username' 95. 'v_b' 96. 'volume_fraction' 97. 'x'

The file `polar.brown.sfs` includes the joint (2 dimensions) site frequency spectrum (SFS) between polar bears (on the rows) and brown bears (on the columns). This is based on real genomic data from 18 polar bears and 7 brown bears. The site frequency spectrum is a matrix $N \times M$

where cell (i, j) reports the number of sites with allele frequency $(i - 1)$ in polar bears and $(j - 1)$ in brown bears. If you want to see this file type `cat polar.brown.sfs` in your terminal.

```
In [18]: # You can plot this spectrum:
         plot2DSFS(polar.brown.sfs, xlab="Polar", ylab="Brown", main="2D-SFS")
```



Each population has $2n + 1$ entries in its spectrum, with n being the number of individuals. The number of chromosomes for each species (bears are diploids, like humans) can be retrieved as:

```
In [19]: nChroms.polar <- nrow(polar.brown.sfs)-1
         nChroms.polar
         nChroms.brown <- ncol(polar.brown.sfs)-1
         nChroms.brown
```

36
14

The only thing you need to remember about the site frequency spectrum is that we can easily calculate several summary statistics from it. These summary statistics can be used for inferences in an Approximate Bayesian Computation (ABC) framework.

For instance, from the site frequency spectrum, we can easily calculate the number of analysed sites (in this example all sites are polymorphic, and thus variable in our sample), which is simply the sum of all entries in the SFS.

```
In [20]: nrSites <- sum(polar.brown.sfs, na.rm=T)
         nrSites
```

16556

This value is important as we will condition the simulations to generate this number of sites for each repetition. In other words, when simulating data we will simulate exactly this number of polymoprhic sites to calculate the site frequency spectrum and all corresponding summary statistics afterwards.

I provide a function to easily calculate several summary statistics from a site frequency spectrum.

```
In [21]: obsSummaryStats <- calcSummaryStats(polar.brown.sfs)
         obsSummaryStats
         # These are the OBSERVED summary statistics! Keep them.
```

fst 0.565642690122126 **pivar1** 0.0524495986684921 **pivar2** 0.209206715005217 **sing1** 0.0884875573810099 **sing2** 0.366634452766369 **doub1** 0.047052428122735 **doub2** 0.0942256583715873 **pef** 0.00899975839574777 **puf** 0.991000241604252

These are the summary statistics available in this practical and their meaning is the following: * fst: population genetic differentiation; it measures how much species are genetically different; it goes from 0 (identical) to 1 (completely different); * pivar1: genetic diversity of species 1 (polar bears); * pivar2: genetic diversity of species 1 (brown bears); * sing1: number of singletons (sites with frequency equal to 1) for species 1 (polar bears); * sing2: number of singletons (sites with frequency equal to 1) for species 2 (brown bears); * doub1: number of doubletons (sites with frequency equal to 2) for species 1 (polar bears); * doub2: number of doubletons (sites with frequency equal to 2) for species 2 (brown bears); * pef: proportion of sites with equal frequency between polar bears and brown bears; * puf: proportion of sites with unequal frequency between polar bears and brown bears (note that puf=1-pef).

It is not important that you understand the significance (if any) of all these summary statistics in an evolutionary context. If interested, a nice review is "Molecular Signatures of Natural Selection" by Rasmus Nielsen (you can find a pdf copy in `Readings/Papers/Nielsen_2005.pdf`). However, some of these summary statistics might be more informative than others. It is your first goal to understand which summary statistics to keep.

The parameter you want to estimate is the divergence time between polar and brown bears (T). You first aim is to performs N simulations of data by drawing from a prior distribution of T and record (separately) the drawn values and the corresponding summary statistics generated by that value of T.

You can define how many simulations you want to perform (ideally a lot).

```
In [22]: nrSimul <- 1e3 # but change this accordingly
```

In [23]:
```
simulate
```

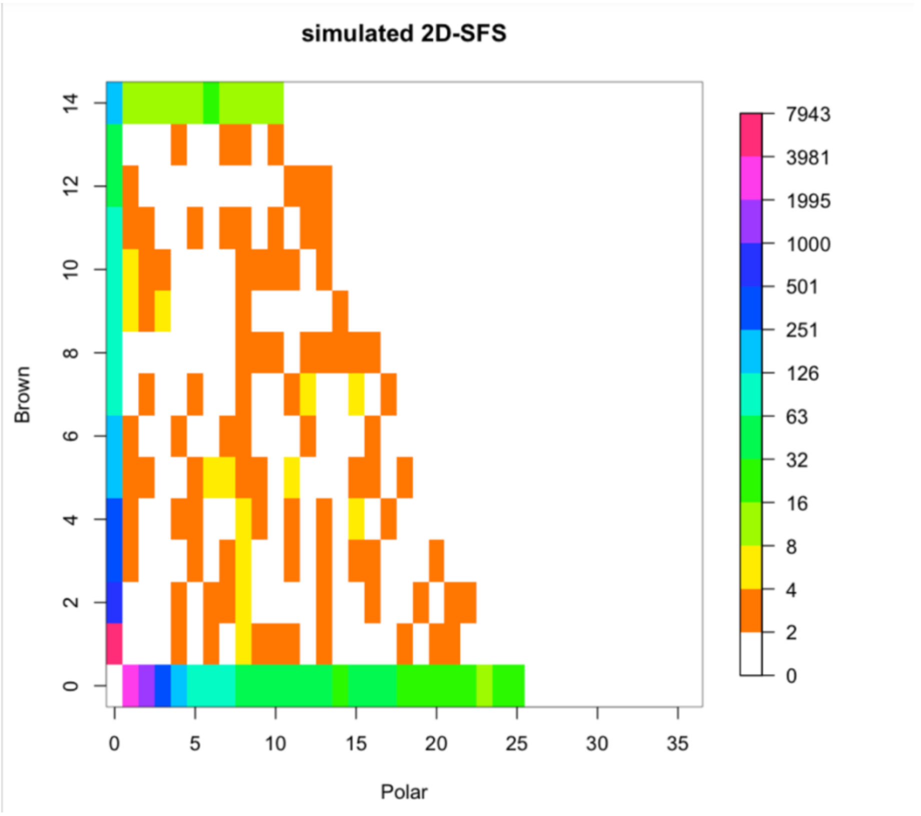
```
function (T, M, nr_snps, ms_dir, fout)
{
    gen_time <- 8.423
    ref_pop_size <- 68000
    Tcoal <- T/(gen_time * ref_pop_size * 4)
    cat("", file = fout)
    ms.command <- paste(ms_dir, "50", nr_snps, "-s 1 -I 2 36 14 -n 1 1 -n 2 6.8 -en 0.02269 1 0.07353 -en 0.05281 2
0.2941 -em 0.06459 1 2",
        M, "-em 0.1392 1 2 0 -en 0.1392 1 0.2941 -ej", Tcoal,
        "2 1 -en 0.3924 1 1.809 >", fout)
    system(ms.command)
}
```

This function takes as parameters: T (divergence time), M (migration rate), how many sites to simulate, the directory for `ms` program and the text file in output. This function simulates a joint evolutionary history for both polar and brown bears according to what we know in terms of their respective changes in size. However, you can set when they speciated (T in years ago) and the migration rate (M). (Note that the migration rate is scaled by the reference population size so a reasonable range of M is between 0 and 2.)

As an example, assuming T=200k and M=0 the command to simulate data and calculate summary statistics is the following:

In [24]:
```
# first, set the path to the "ms" software you installed
msDir <- "Software/msdir/ms" # this is my specific case, yours could be different

# second, set the name for the output text file
fout <- "ms.txt" # leave it like here

# then we can simulate data:
simulate(T=2e5, M=0, nrSites, msDir, fout)

# and finally calculate the summary statistics for this simulation
#(note that you need to specify the number of chromosomes for the two species)
simulatedSFS <- fromMStoSFS(fout, nrSites, nChroms.polar, nChroms.brown)
calcSummaryStats(simulatedSFS)

# you can even plot the simulated site frequency spectrum
plot2DSFS(simulatedSFS, xlab="Polar", ylab="Brown", main="simulated 2D-SFS")
```

```
   fst  0.413394578669613
pivar1  0.0578181985526755
pivar2  0.126645763031363
 sing1  0.232544092776033
 sing2  0.417431746798744
 doub1  0.0634211162116453
```



Based on the observed summary statistics 'fst', which measures how different polar and brown bears, in relation to the one calculated simulating T=2e5, can you make some initial (very rough) considerations on the most likely values of T (higher or lower than 200k years ago)?

You can use the `abc` package and the `abc` function to calculate the posterior distribution (as well as to compute the distance between observed and expected summary statistics).

In [25]:
```
library(abc)
?abc
```

As you can see, to perform an ABC analysis you need 3 objects:

- target: a vector of the observed summary statistics;
- param: a vector, matrix or data frame of the simulated parameter values;
- sumstat: a vector, matrix or data frame of the simulated summary statistics.

You already have 'target' as it is the vector of observed summary statistics called 'obsSummaryStatistics'.

You now have everything to estimate the divergence time. For simplicity assume that $M = 0$. Also, you are free to choose a rejection or local-regression method, as specified in the 'abc' function. This is not strictly required, but if you want to explore the estimation of two parameters simultaneously, you can estimate M by defining a prior for it, draw random samples jointly of T and M, calculate summary statistics, and so on.

Hints

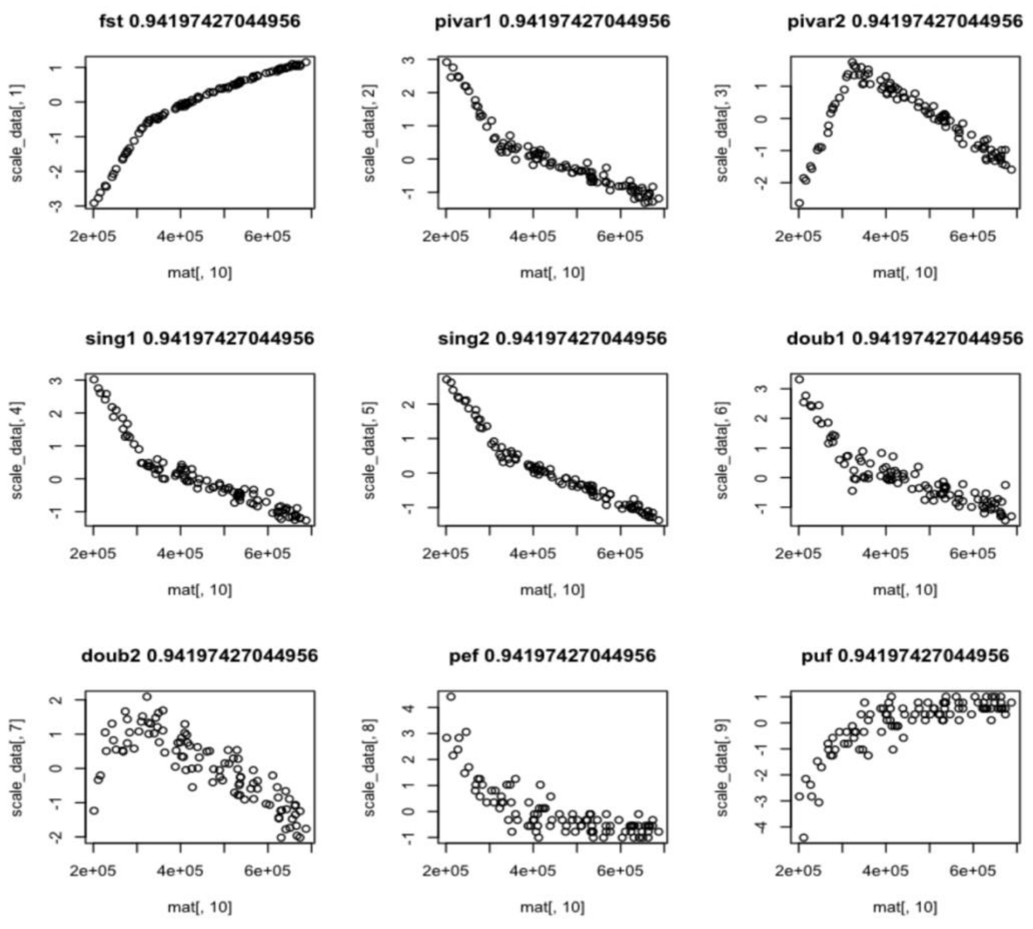
Please consider these points carefully when completing the project.

- Assess which summary statistics are more or less informative for the parameter estimation (e.g. after a first run of simulations with all parameters, look for correlations between the simulated parameter value and summary statistics).
- You can also look for correlations between summary statistics and eventually use only one of the pair if two summary statistics are highly correlated. If you are a pro, you can also perform a principal component or multidimensional scaling analysis (e.g. with package 'pls') and by using each statistic's loadings, you can create novel uncorrelated summary statistics which are linear combinations of the previous ones (this part is purely suggestive and it is not required to obtain the full score).
- Remember to scale your simulated (jointly with the observed) summary statistics separately, so that the mean is zero and standard deviation is one.
- Generate a plot with the posterior distribution of the parameter of interest. You can also show the chosen prior distribution on the same plot.
- Calculate the posterior mean, mode, median and other notable quantities (e.g. 95% HPD interval) to summarise the posterior distribution.
- I suggest you to use the 'abc' package in R instead of implementing methods (e.g. regression) yourself.
- A useful diagnostic plot to show is the distribution of sampled values from the prior: do they cover the whole range of the prior (and are they distributed as expected)?

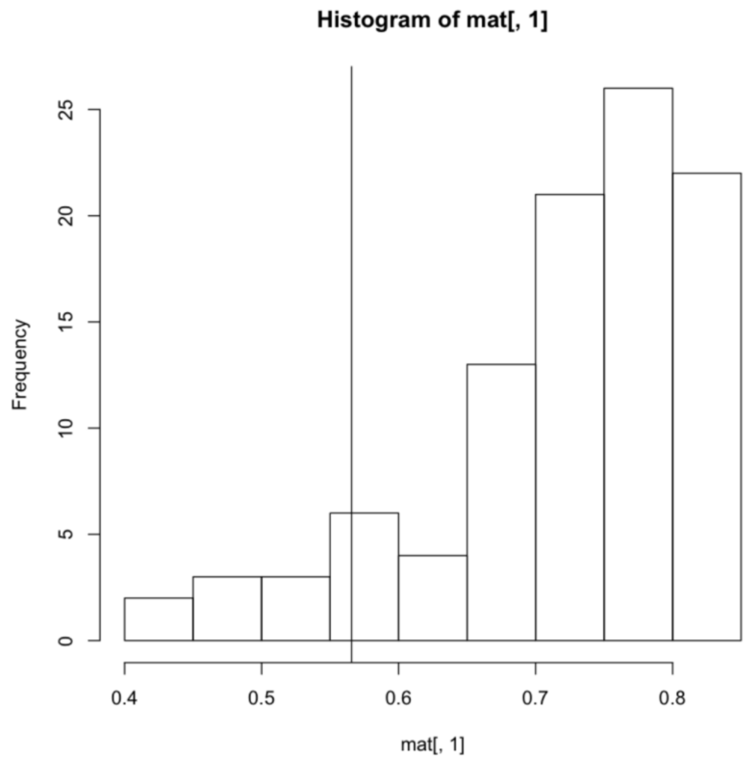
In [12]:
```
# second, set the name for the output text file
fout <- "ms.txt" # leave it like here
mat <- matrix(NA,nrow = nrSimul, ncol = 10)
nam <- c("fst", "pivar1", "pivar2", "sing1", "sing2", "doub1", "doub2","pef","puf","t")
colnames(mat) <- nam
for (i in 1:nrSimul){
  t= runif(1,2e5,7e5)
  # then we can simulate data:
  simulate(T=t, M=0, nrSites, msDir, fout)
  # and finally calculate the summary statistics for this simulation
  #(note that you need to specify the number of chromosomes for the two species)
  simulatedSFS <- fromMStoSFS(fout, nrSites, nChroms.polar, nChroms.brown)
  x=calcSummaryStats(simulatedSFS)
  mat[i,1] <- x[1]
  mat[i,2] <- x[2]
  mat[i,3] <- x[3]
  mat[i,4] <- x[4]
  mat[i,5] <- x[5]
  mat[i,6] <- x[6]
  mat[i,7] <- x[7]
  mat[i,8] <- x[8]
  mat[i,9] <- x[9]
  mat[i,10] <- t
}
```

In [13]:
```
#scaling the simulated data
scale_data <- matrix(NA,nrow = nrSimul, ncol = 9)
nams <- c("fst", "pivar1", "pivar2", "sing1", "sing2", "doub1", "doub2","pef","puf")
colnames(scale_data) <- nams
for (k in 1:dim(scale_data)[2]){
  scale_data[,k] <- scale(mat[,k])
}
```

In [14]:
```
par(mfrow=c(3,3))
#checking plots to decide which summary statistic to use
plot(mat[,10],scale_data[,1],main = paste(nam[1],cor(mat[,10],mat[,1])))
plot(mat[,10],scale_data[,2],main = paste(nam[2],cor(mat[,10],mat[,1])))
plot(mat[,10],scale_data[,3],main = paste(nam[3],cor(mat[,10],mat[,1])))
plot(mat[,10],scale_data[,4],main = paste(nam[4],cor(mat[,10],mat[,1])))
plot(mat[,10],scale_data[,5],main = paste(nam[5],cor(mat[,10],mat[,1])))
plot(mat[,10],scale_data[,6],main = paste(nam[6],cor(mat[,10],mat[,1])))
plot(mat[,10],scale_data[,7],main = paste(nam[7],cor(mat[,10],mat[,1])))
plot(mat[,10],scale_data[,8],main = paste(nam[8],cor(mat[,10],mat[,1])))
plot(mat[,10],scale_data[,9],main = paste(nam[9],cor(mat[,10],mat[,1])))
```

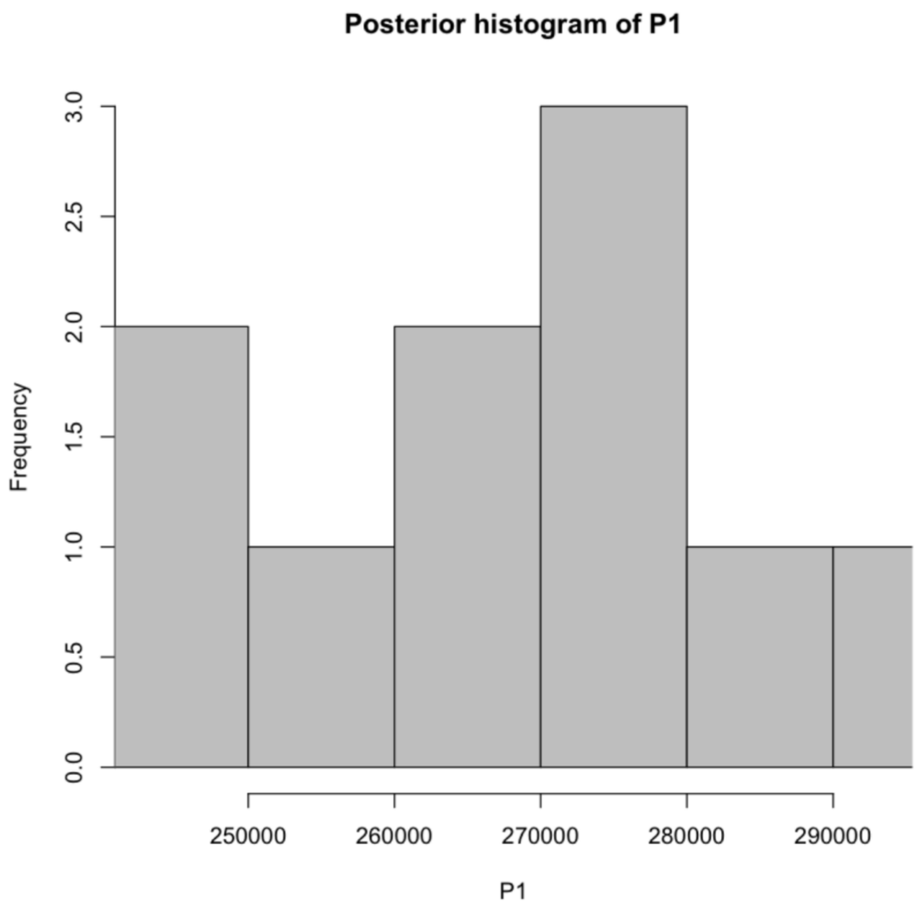


In [20]:
```
#plot simulated FST distribution and check where observed FST is
hist(mat[,1])
abline(v=as.numeric(obsSummaryStats[1]))
```



```
In [23]: library(abc)
         #Using FST
         #simulated 1000 due to time to run but increased tolerance to 10%
         #target
         target <- obsSummaryStats[1] #FST
         #simulated FST data with target to scale
         sim_fst <- matrix(NA,ncol = 1,nrow = length(mat[,1])+1)
         sim_fst <- c(mat[,1],as.numeric(target))
         scale_sim_fst <- scale(sim_fst)
         result <- abc(target=scale_sim_fst[101],param=mat[,10],sumstat=scale_sim_fst[1:100], tol = 0.10, method = "rejection")
         hist(result)
         res_sum <- summary(result)
```

```
Call:
abc(target = scale_sim_fst[101], param = mat[, 10], sumstat = scale_sim_fst[1:100],
    tol = 0.1, method = "rejection")
Data:
 abc.out$unadj.values (10 posterior samples)

                   P1
Min.:         242909.3
2.5% Perc.:   243646.1
Median:       270878.9
Mean:         268276.7
Mode:         275135.5
97.5% Perc.:  291113.2
Max.:         293404.4
```



Matthew Campos
CID: 01749686
Bayesian Statistics

What are the hypotheses to be tested?

The purpose of the research is to see how trophic interactions change in response to anthropogenic pressures. They consider how ecological and environmental patterns would shift with implementation of changes in fisheries. Using Bayesian Network, the focus was to identify the potential trade-offs among species and shifts in ecological structure which includes population and response. Before implementing new fishing rules, Bayesian network simulates alternative scenarios to reduce uncertainty with species effects and choose the best strategy. They look at change in productivity, temperature and species populations based on chosen strategy and its ecological consequences.

How did the authors suggest to use Bayes networks to test these hypotheses?

Bayesian network was used to simulate how different different changes in fisheries will affect different ecosystems. Model investigated how these hypothetical pressures will impact productivity and trophic interactions. The model was an “end-to-end” model representing different areas and their respective trophic dynamics. Each area incorporated data as well as knowledge of different species and their dynamics under different scenarios. The network combines both biotic and abiotic factors. It accounted for spatial heterogeneity and ecological complexity, accounting for correlation between area. They designate catch data into three categories, low, medium and high, and use bootstrap of historical data to predict the outcomes of different scenarios. The HV is represented by a discrete node which is parameterized using the Expectation Maximization algorithm, similar to a maximum likelihood.

They integrated topological features such as food webs with quantitative variation in species interactions and stochastic environmental stressors with a dynamic BN model to predict the outcomes. The model is then modified to make future projections of species and trophic groups. Using catch data from 1983 to 2015 as the prior, they use a Bayes Network where each scenario each with its own independent probability of outcome. Using conditional probability distribution, they can predict the probability of observing certain values. This allows for better predictions of the different interactions and stressors. They also used a Bayesian network because it would allow them to include a hidden variable which represents the complex interdependencies of species and environment. By adding this hidden variable, they can make improved inferences of the system as it captures higher order relationships.

What are the conclusions of the study?

Results showed there will be variability in the trends as the outcome is not consistent for the different scenarios in different sub-areas. The different species had different responses. For example, the expectation is that high fisheries catch would result in highest impact in predicted species population. This is the case for Cod as high and medium level fisheries catch had a similar trends of low population values, with the hidden variable being able to capture the variation in outcomes. However, for Whiting in a specific area, it had a different response to low fisheries catch scenarios, having the worst outcome. The difference in results highlights the spatial heterogeneity in terms of species responses to ecosystem change, as well as the spatial relationship between neighbouring areas and trophic interactions. They also investigated the effects on temperature on net primary production. Results showed that temperature influences the projection of productivity, which influences the trends of certain marine species. Different temperature increase scenarios were tested on the data and variability showed variation with species specific responses. The results showed that species higher up in the food chain will be most influenced by that the changes in productivity and temperature.

What did you learn on the use of Bayes network after reading this paper?

Bayesian network can be an accurate method for simulating and predicting future trends using historical data. It seems to be better than just maximum likelihood as it is able to account for uncertainties and higher-level factors not described in the data. It is important that the models are not too complex but also accurate in predictions. Bayes networks are able to account for the complexities found in the natural world and model the conditional dependence of factors. In the paper, they were able to include a hidden variable which accounts for species interactions with each other as well as the environment. As a result, it gives them high flexibility for accounting for latent effects and seeing how species change, and their dynamics, to reduce uncertainty. Therefore, they can more accurately see how the different scenarios affect not only the species present but also the overall ecosystem. By reducing the uncertainty, they have more robust results and can make better decisions with regards to fishery catch.