Assignment 2.1

Mahdi

September 30, 2020

1.2

1.2.1

The 1st column is African, the 2nd columns is European and the 3rd column is Asian.

```
## -- Attaching packages ---------------------------------------------- tidyverse 1.3.0 --
```

```
## v ggplot2 3.3.2     v purrr   0.3.4
## v tibble  3.0.3     v dplyr   1.0.0
## v tidyr   1.1.0     v stringr 1.4.0
## v readr   1.3.1     v forcats 0.5.0
```

```
## -- Conflicts ------------------------------------------------- tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

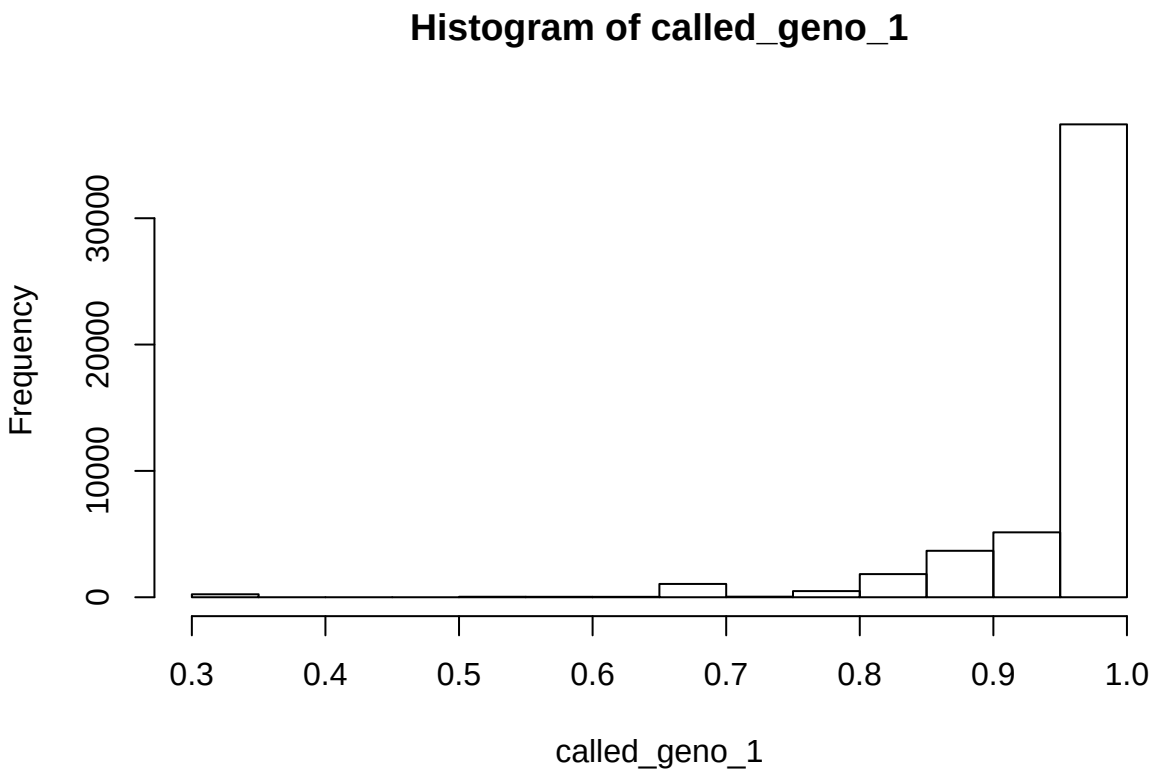
```
##      Ind60  Ind60.1  Ind60.2
## 1 0.000211 0.333261 0.666528
## 2 0.000009 0.199995 0.799996
## 3 0.000000 0.111109 0.888891
## 4 0.000000 0.058822 0.941178
## 5 0.333333 0.333333 0.333333
## 6 0.000000 0.030302 0.969698
```

1.2.2

Histogram of the posterior probabilities assuming uniform prior

```
get_posterior <- function(likelihood, prior){
  numerator <- likelihood * prior
  denominator <- apply(numerator, 1, sum)
  posterior <- numerator/denominator
  return(posterior)
}

posterior_1 <- get_posterior(lik_61, 1/3)
called_geno_1 <- apply(posterior_1, 1, max)
hist(called_geno_1)
```



Histogram of the posterior probabilities assuming frequency

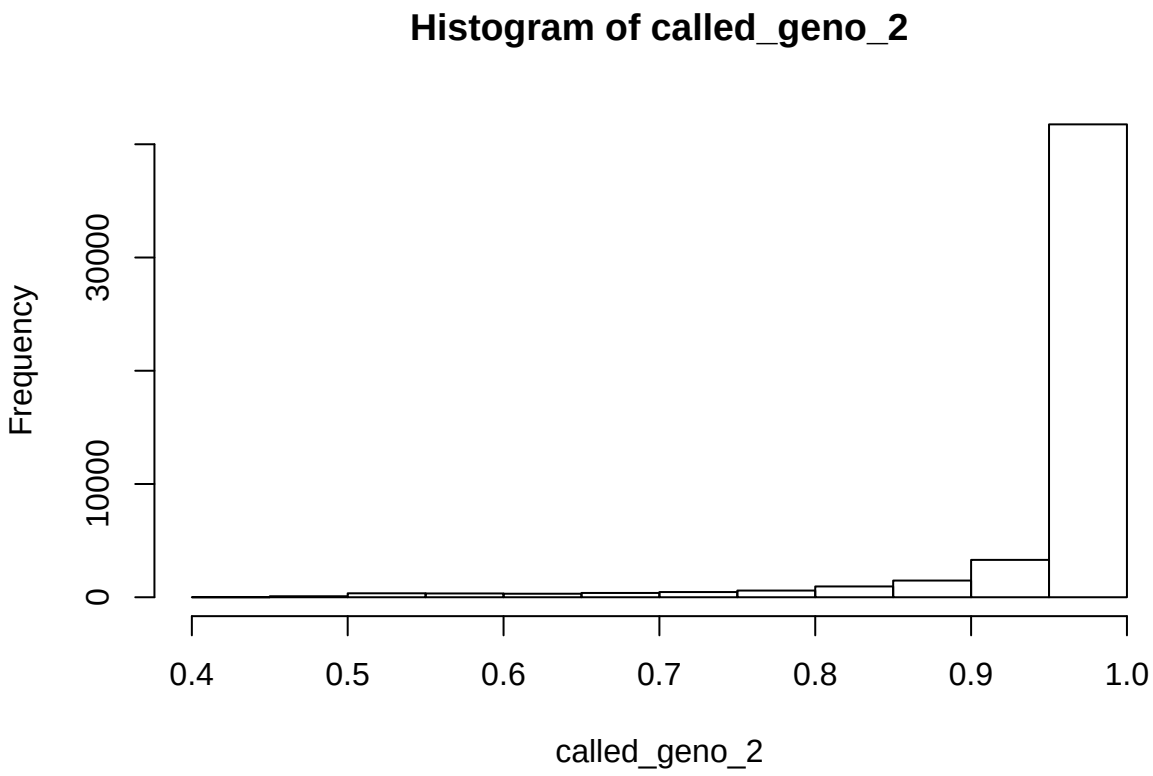
```
freq <- read.table("assign3.fopt.gz")
info[info$V2 == "NA12750",]
```

```
##     V1      V2
## 61 CEU NA12750
```

We can see that the individual is European so we use the second column of the frequency file.

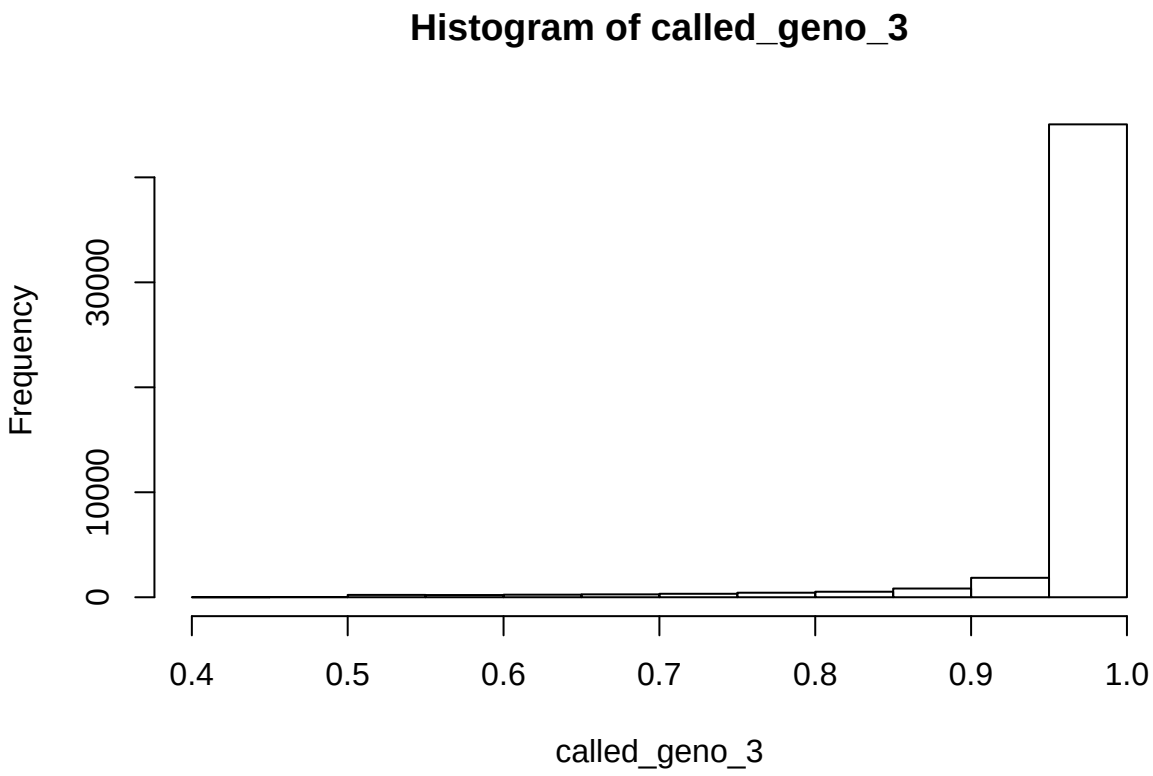
```
get_genotype_freq <- function(q){
  #q <- 1 - p
  p <- 1 - q
  return(data.frame("RR" = p^2, "RA" = 2*p*q, "AA" = q^2))
}

prior_2 <- get_genotype_freq(freq$V2)
posterior_2 <- get_posterior(lik_61, prior_2)
called_geno_2 <- apply(posterior_2, 1, max)
hist(called_geno_2)
```



Histogram of the posterior probabilities assuming Beagle

```
beagle_probs <- read.table("imputation.input.gz.gprobs.gz", header = T)
posterior_3 <- get_columns(beagle_probs, ind)
called_geno_3 <- apply(posterior_3, 1, max)
hist(called_geno_3)
```



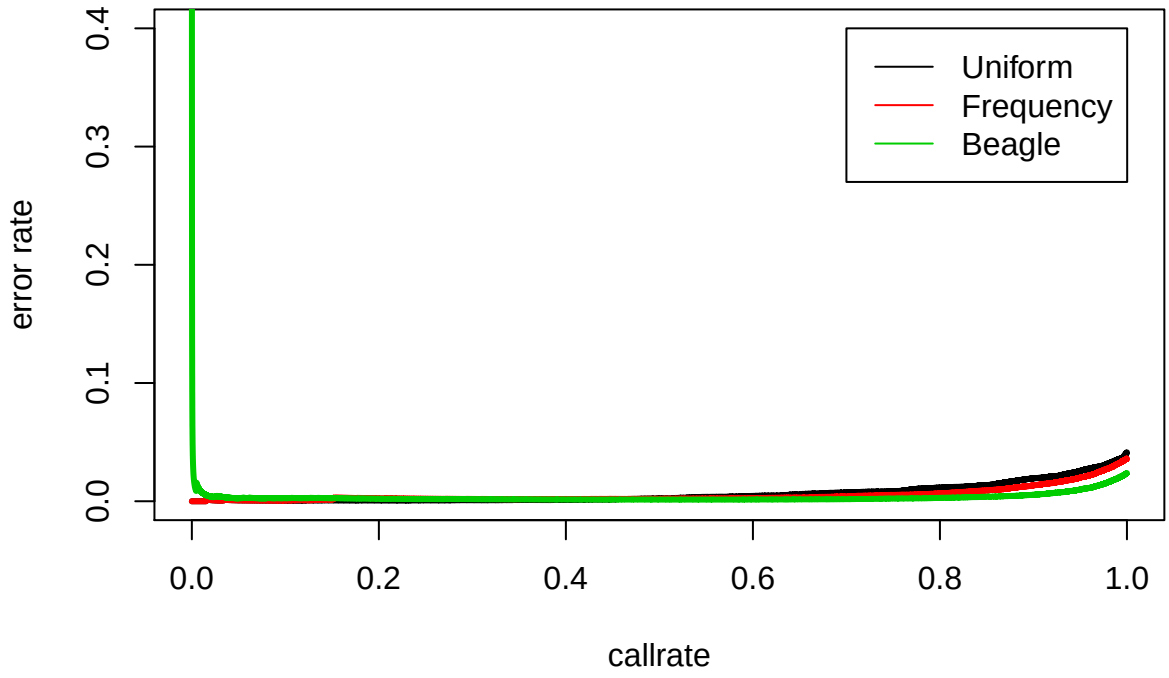
1.2.3

Make a plot with the accuracy of the three genotyping approaches

```
get_predictions <- function(post, max, true){
  pred <- sweep((post == max), 2, c(1,2,3), `*`)
  pred <- apply(pred, 1, sum) - 1
  return(pred == true)
}

true <- genotypes$NA12750
pred_1 <- get_predictions(posterior_1, called_geno_1, true)
pred_2 <- get_predictions(posterior_2, called_geno_2, true)
pred_3 <- get_predictions(posterior_3, called_geno_3, true)


plot(1,xlim=0:1,ylim=c(0,0.40),col="transparent",xlab="callrate",ylab="error rate")
plotAccuracy(pred_1,called_geno_1,lwd=3,col=1)
plotAccuracy(pred_2,called_geno_2,lwd=3,col=2)
plotAccuracy(pred_3,called_geno_3,lwd=3,col=3)
legend(0.70, 0.4, legend = c("Uniform", "Frequency", "Beagle"),
       col= c(1,2,3), lty=1)
```



Analysis for Ind2

```
ind2 <- 3
lik_3 <- get_columns(likelihoods, ind2)
true2 <- genotypes$NA19663

#get predictions and posteriors for 3 pops
get_pred_maxpost <- function(pop, lik, true){
  prior <- get_genotype_freq(pop)
  post <- get_posterior(lik, prior)
  max_post <- apply(post, 1, max)
  pred <- get_predictions(post, max_post, true)
  return(list(pred, max_post))
}

list_3_pops <- lapply(freq, get_pred_maxpost, lik_3, true2)

#get predictions and posteriors for combined pop
admix <- read.table("assign3.qopt")
admix_3 <- admix[3,]
prior_combined <- (admix_3$V1 * get_genotype_freq(freq$V1)) +
  (admix_3$V2 * get_genotype_freq(freq$V2)) +
  (admix_3$V3 * get_genotype_freq(freq$V3))

posterior_combined <- get_posterior(lik_3, prior_combined)
```

```r
called_geno_comb <- apply(posterior_combined, 1, max)
pred_comb <- get_predictions(posterior_combined, called_geno_comb, true2)

# plot
plot(1,xlim=0:1,ylim=c(0,0.40),col="transparent",xlab="callrate",ylab="error rate")
for (i in 1:length(list_3_pops)){
  plotAccuracy(list_3_pops[[i]][[1]],list_3_pops[[i]][[2]],lwd=3,col=i)
}
plotAccuracy(pred_comb,called_geno_comb,lwd=3,col=4)
legend(0, 0.4, legend = c("African", "European", "Asian", "Combined"),
       col= c(1,2,3,4), lty=1)
```