

Practical 5 Statistical Genetics

Antonin Rosa

Hendrik Wellmann

Question 1

```
library(data.table)

data <- fread("YRI6.raw", header = FALSE, sep = " ")

num_individuals <- nrow(data)
num_snps <- ncol(data) - 6

percentage_missing <- mean(is.na(data)) * 100

cat("Number of individuals:", num_individuals, "\n")

## Number of individuals: 85

cat("Number of SNPs:", num_snps, "\n")

## Number of SNPs: 56574

cat("Percentage of missing data:", percentage_missing, "%\n")

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

Question 2

```
genomic_data <- data[2:nrow(data), 7:ncol(data)]
genomic_data[, (1:ncol(genomic_data)) := lapply(.SD, as.numeric), .SDcols = 1:ncol(genomic_data)]
shared_mean <- matrix(c(NA), nrow = num_individuals, ncol = num_individuals)
shared_sd <- matrix(c(NA), nrow = num_individuals, ncol = num_individuals)
for (i in 1:num_individuals) {
  for (j in 1:num_individuals) {
    shared <- numeric(0)
    for (k in 1:nrow(genomic_data)){
      shared <- c(shared, 2 - abs(as.matrix(genomic_data[k,i, with=FALSE]) - as.matrix(genomic_data[k,j, with=FALSE])))
    }
    mean <- mean(shared)
    sd <- sd(shared)
    shared_mean[i,j] <- mean
    shared_sd[i,j] <- sd
  }
}

print(shared_mean[1:5, 1:5])
```

```

##      [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 2.000000 1.250000 1.190476 1.154762 1.273810
## [2,] 1.250000 2.000000 1.297619 1.476190 1.190476
## [3,] 1.190476 1.297619 2.000000 1.178571 1.107143
## [4,] 1.154762 1.476190 1.178571 2.000000 1.190476
## [5,] 1.273810 1.190476 1.107143 1.190476 2.000000

```

```
print(shared_sd[1:5, 1:5])
```

```

##      [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.0000000 0.6376727 0.6489322 0.6492637 0.6828570
## [2,] 0.6376727 0.0000000 0.6166322 0.6300926 0.6489322
## [3,] 0.6489322 0.6166322 0.0000000 0.7140705 0.6769502
## [4,] 0.6492637 0.6300926 0.7140705 0.0000000 0.6489322
## [5,] 0.6828570 0.6489322 0.6769502 0.6489322 0.0000000

```

Question 3

```

p0 <- matrix(c(NA), nrow = num_individuals, ncol = num_individuals)
p2 <- matrix(c(NA), nrow = num_individuals, ncol = num_individuals)
m <- matrix(c(NA), nrow = num_individuals, ncol = num_individuals)
for (i in 1:num_individuals) {
  for (j in 1:num_individuals) {
    shared <- numeric(0)
    for (k in 1:nrow(genomic_data)){
      shared <- c(shared, 2 - abs(as.matrix(genomic_data[k,i, with=FALSE]) - as.matrix(genomic_data[k,j])
    }
    p0[i,j] <- sum(shared == 0)/length(shared)
    p2[i,j] <- sum(shared == 2)/length(shared)
    m[i,j] = 1 - p0[i,j] + p2[i,j]
  }
}
print(p0[1:5, 1:5])

```

```

##      [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 0.0000000 0.10714286 0.13095238 0.14285714 0.1309524
## [2,] 0.1071429 0.00000000 0.08333333 0.07142857 0.1309524
## [3,] 0.1309524 0.08333333 0.00000000 0.17857143 0.1785714
## [4,] 0.1428571 0.07142857 0.17857143 0.00000000 0.1309524
## [5,] 0.1309524 0.13095238 0.17857143 0.13095238 0.0000000

```

```
print(p2[1:5, 1:5])
```

```

##      [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 1.0000000 0.3571429 0.3214286 0.2976190 0.4047619
## [2,] 0.3571429 1.0000000 0.3809524 0.5476190 0.3214286
## [3,] 0.3214286 0.3809524 1.0000000 0.3571429 0.2857143
## [4,] 0.2976190 0.5476190 0.3571429 1.0000000 0.3214286
## [5,] 0.4047619 0.3214286 0.2857143 0.3214286 1.0000000

```

```

print(m[1:5, 1:5])

##          [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 2.000000 1.250000 1.190476 1.154762 1.273810
## [2,] 1.250000 2.000000 1.297619 1.476190 1.190476
## [3,] 1.190476 1.297619 2.000000 1.178571 1.107143
## [4,] 1.154762 1.476190 1.178571 2.000000 1.190476
## [5,] 1.273810 1.190476 1.107143 1.190476 2.000000

print(shared_mean[1:5, 1:5])

##          [,1]      [,2]      [,3]      [,4]      [,5]
## [1,] 2.000000 1.250000 1.190476 1.154762 1.273810
## [2,] 1.250000 2.000000 1.297619 1.476190 1.190476
## [3,] 1.190476 1.297619 2.000000 1.178571 1.107143
## [4,] 1.154762 1.476190 1.178571 2.000000 1.190476
## [5,] 1.273810 1.190476 1.107143 1.190476 2.000000

print(all.equal(shared_mean, m))

## [1] TRUE

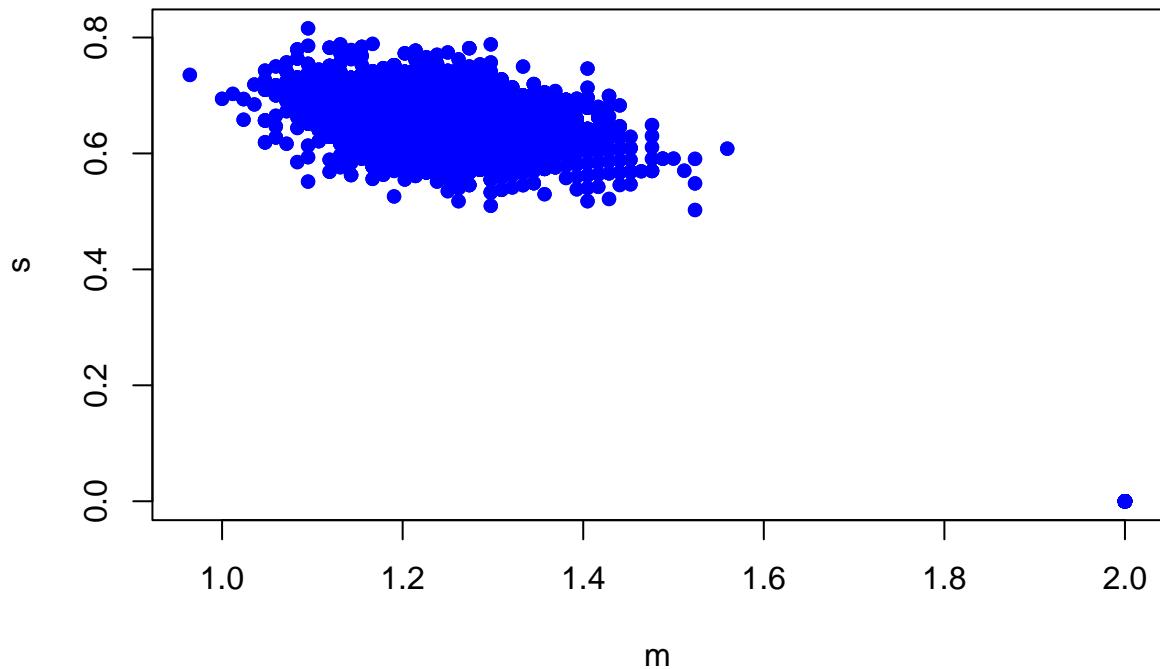
```

It holds because we can see that each element of m ($m = 1 - p_0 + p_2$) is equal to each element of `shared_mean`.

Question 4

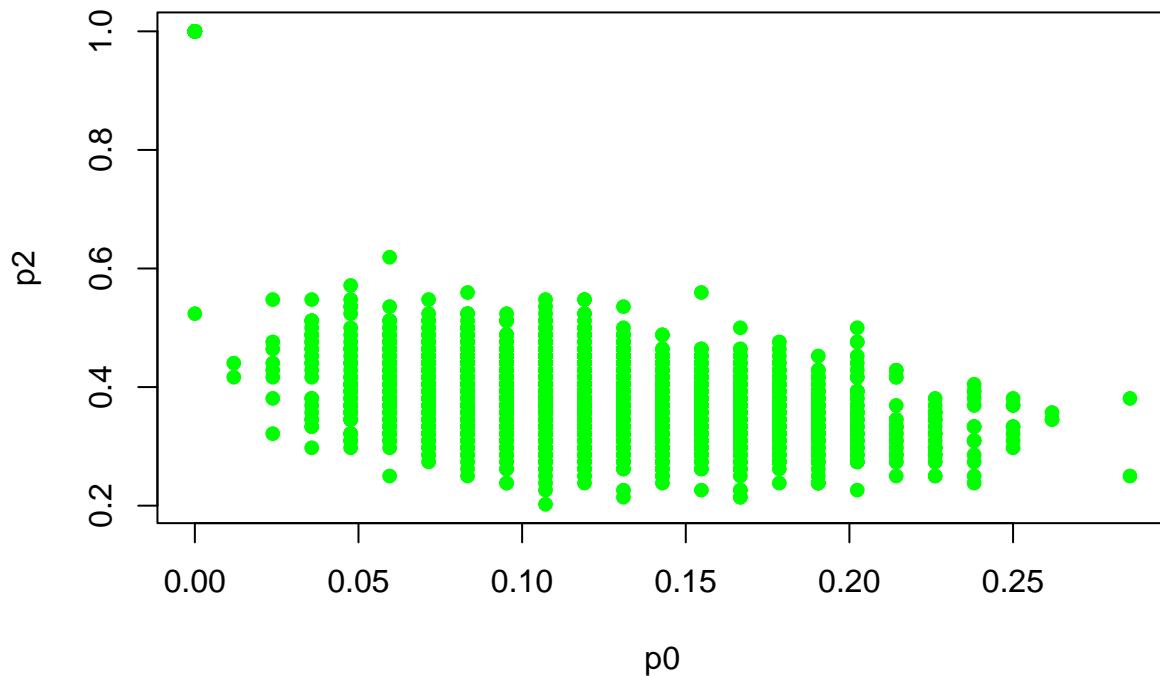
```
plot(shared_mean, shared_sd, main = "m vs s", xlab = "m", ylab = "s", col = "blue", pch = 16)
```

m vs s



```
plot(p0, p2, main = "p0 vs p2", xlab = "p0", ylab = "p2", col = "green", pch = 16)
```

p0 vs p2



m vs s : **Cluster in the Top Left:** The clustered points in this region could indicate a group of individuals with similar genetic characteristics. This grouping might be due to close familial relationships or specific genetic similarities.

Isolated Point in the Bottom Right: An isolated point suggests the presence of two individuals with identical genetic information. So either an error in the data or monozygotic twins.

p0 vs p2 : Cluster on the Entire Bottom Part: The concentration of points in the bottom part of the plot suggests that the majority of individuals share similar genetic characteristics in terms of $p0$ and $p2$. This could result from genetic similarities within this population.

Isolated Point in the Top Left: The presence of an isolated point in the top left suggests that there is an individual or a group of individuals that stand out from the others in terms of $p0$ and $p2$. This is the two individuals with identical genetic information again.

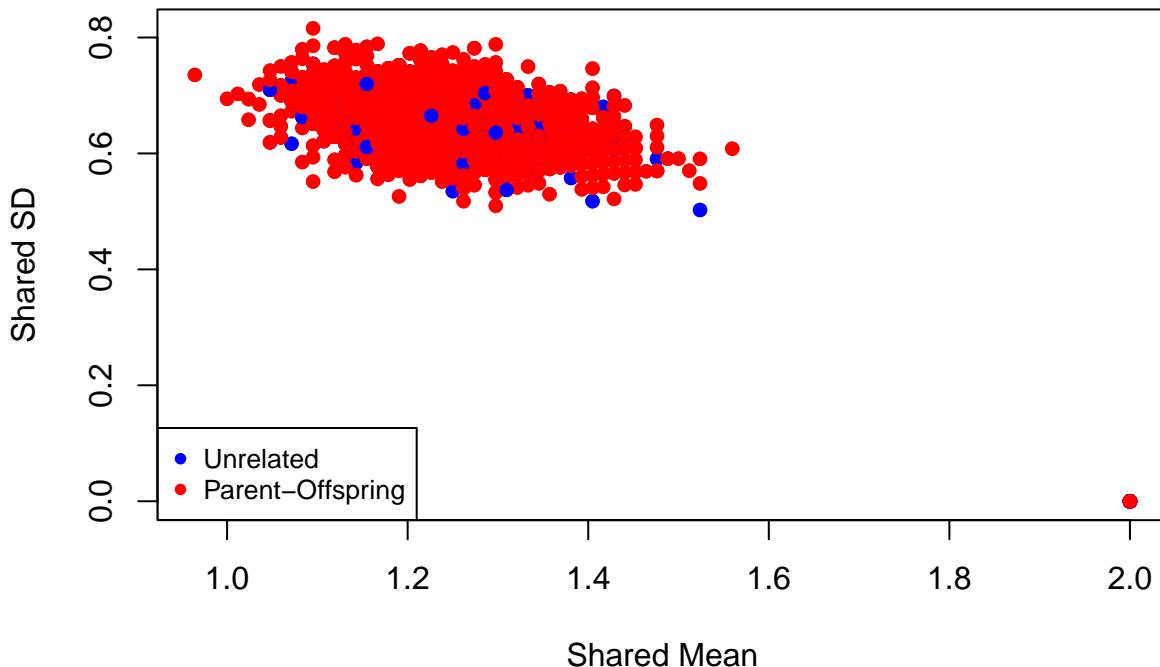
Question 5

```

m <- shared_mean
s <- shared_sd
family_relationship <- data[, c(3, 4)]
colors <- rep("blue", nrow(data) - 1)
parent_offspring_indices <- which(family_relationship[, 1] > 0 | family_relationship[, 2] > 0)
colors[parent_offspring_indices - 1] <- "red"
parent <- c()

```


Shared Mean vs Shared SD



We can see that all the majority of points present in the clusters have a family relationship. It affirms the hypothesis said before in question 4.

Question 6

```
library(SNPRelate)

## Lade nötiges Paket: gdsfmt

## SNPRelate -- supported by Streaming SIMD Extensions 2 (SSE2)

raw_data <- read.table("YRI6.raw", header = TRUE, sep = " ")
sample_ids <- raw_data[, 2]
snp_ids <- names(raw_data[, 7:ncol(raw_data)])
genotypes <- as.matrix(raw_data[, 7:ncol(raw_data)], drop = FALSE)
print(dim(genotypes))

## [1] 84 56574

print(length(sample_ids))

## [1] 84
```

```

print(length(snp_ids))

## [1] 56574

snpGDSCreateGeno("test.gds", genmat = t(genotypes), sample.id = sample_ids, snp.id = snp_ids)

genofile <- snpGDSOpen("test.gds")
snpset <- snpGDSLDpruning(genofile, ld.threshold=0.2)

## SNP pruning based on LD:
## Excluding 0 SNP on non-autosomes
## Excluding 0 SNP (monomorphic: TRUE, MAF: NaN, missing rate: NaN)
##     # of samples: 84
##     # of SNPs: 56,574
##     using 1 thread
##     sliding window: 500,000 basepairs, Inf SNPs
##     |LD| threshold: 0.2
##     method: composite
## Chromosome 1: 0.12%, 67/56,574
## 67 markers are selected in total.

snpset.id <- unlist(unname(snpset))
ibd <- snpGDSIBDML( genofile, maf=0.05, missing.rate=0.05, snp.id=snp_ids, num.thread=2)

## Identity-By-Descent analysis (MLE) on genotypes:
## Excluding 0 SNP (non-autosomes or non-selection)
## Excluding 0 SNP (monomorphic: TRUE, MAF: 0.05, missing rate: 0.05)
##     # of samples: 84
##     # of SNPs: 56,574
##     using 2 threads
## MLE IBD: the sum of all selected genotypes (0,1,2) = 4388685
## MLE IBD: Tue Dec 19 15:09:34 2023    0%
## MLE IBD: Tue Dec 19 15:10:04 2023    4%
## MLE IBD: Tue Dec 19 15:11:34 2023   16%
## MLE IBD: Tue Dec 19 15:12:04 2023   20%
## MLE IBD: Tue Dec 19 15:15:05 2023   42%
## MLE IBD: Tue Dec 19 15:17:05 2023   56%
## MLE IBD: Tue Dec 19 15:17:35 2023   59%
## MLE IBD: Tue Dec 19 15:18:36 2023   66%
## MLE IBD: Tue Dec 19 15:19:36 2023   73%
## MLE IBD: Tue Dec 19 15:20:06 2023   77%
## MLE IBD: Tue Dec 19 15:21:37 2023   86%
## MLE IBD: Tue Dec 19 15:23:07 2023   99%

ibd.coeff <- snpGDSIBDSelection(ibd)
family_info <- raw_data[, c(3, 4)]

colors <- ifelse(family_info[, 1] > 0 & family_info[, 2] > 0, "red", "blue")

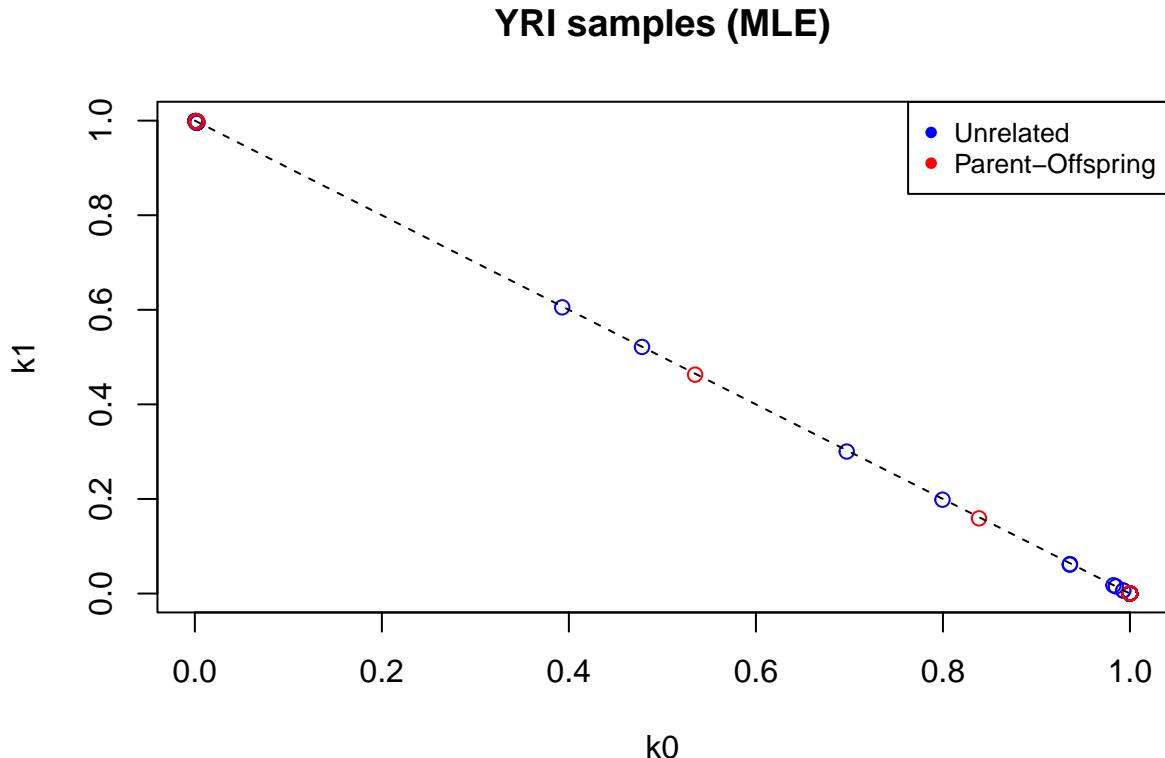
```

```

plot(ibd.coeff$k0, ibd.coeff$k1, col = colors, xlim = c(0, 1), ylim = c(0, 1),
     xlab = "k0", ylab = "k1", main = "YRI samples (MLE)")

legend("topright", legend = c("Unrelated", "Parent-Offspring"), col = c("blue", "red"), pch = 16, cex =
lines(c(0,1), c(1,0), col="black", lty=2)

```



Question 7

In IBD probability plot there are pairs with very low k1 values marked as Parent-Offspring. Parent-Offspring relationships are usually associated with a k0 of 0 and a k1 > 0.5. This suggests that there are incorrectly specified family relationships in the data.