answers.r

antoninrosa

2023-11-20

```r
# Question 1
library(HardyWeinberg)
```

```
## Loading required package: mice
```

```
##
## Attaching package: 'mice'
```

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

```
## The following objects are masked from 'package:base':
##
##     cbind, rbind
```

```
## Loading required package: Rsolnp
```

```
## Loading required package: nnet
```

```r
library(data.table)

file <- "TSIChr22v4.raw"
data <- fread(file, header = FALSE, sep = " ")
genetic_data <- data[, 7:ncol(data)]
first_elements <- sapply(genetic_data, function(x) x[1])
rs_columns <- grepl("^rs", first_elements)
genetic_data_rs <- genetic_data[, ..rs_columns, with = FALSE]
num_variants <- ncol(genetic_data_rs)
missing_percentage <- mean(is.na(genetic_data_rs)) * 100

cat("Number of variants in the database:", num_variants, "\n")
```

```
## Number of variants in the database: 1101343
```

```r
cat("Percentage of missing data:", missing_percentage, "%\n")
```

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

```r
# Question 2

monomorphic_variants <- sapply(genetic_data_rs, function(col) {
  cleaned_col <- col[-1]  # Remove the first element (variant name)
  length(unique(cleaned_col[!is.na(cleaned_col)])) == 1
})
percentage_monomorphic <- (sum(monomorphic_variants) / length(monomorphic_variants)) * 100
non_monomorphic_data <- genetic_data_rs[, !monomorphic_variants, with = FALSE]
```

```r
num_remaining_variants <- ncol(non_monomorphic_data)
cat("Percentage of monomorphic variants:", percentage_monomorphic, "%\n")
```

```
## Percentage of monomorphic variants: 81.0347 %
```

```r
cat("Number of remaining variants:", num_remaining_variants, "\n")
```

```
## Number of remaining variants: 208873
```

```r
# Question 3

variant_column_index <- which(genetic_data_rs[1, ] == "rs587756191_T")
variant_column <- genetic_data_rs[[variant_column_index]]

variant_column <- factor(variant_column, levels = c(0, 1, 2), labels = c("AA", "AB", "BB"))
print(table(variant_column))
```

```
## variant_column
##  AA  AB  BB 
## 106   1   0
```

```r
# Create a table with counts
chi_square_result <- HWChisq(table(variant_column))
```

```
## Warning in HWChisq(table(variant_column)): Expected counts below 5: chi-square
## approximation may be incorrect
```

```
## Chi-square test with continuity correction for Hardy-Weinberg equilibrium (autosomal)
## Chi2 =  106.2512 DF =  1 p-value =  6.495738e-25 D =  0.002336449 f =  -0.004694836
```

```r
print(chi_square_result)
```

```
## $chisq
## [1] 106.2512
## 
## $pval
## [1] 6.495738e-25
## 
## $D
## [1] 0.002336449
## 
## $p
## [1] 0.004672897
## 
## $f
## [1] -0.004694836
## 
## $expected
##           AA           AB           BB 
## 2.336449e-03 9.953271e-01 1.060023e+02 
## 
## $chi.contrib
##           AA           AB           BB 
## 1.060023e+02 2.465008e-01 2.336449e-03
```

```r
# Chi-square test with continuity correction
chi_square_correction_result <- HWChisq(table(variant_column))
```

```
## Warning in HWChisq(table(variant_column)): Expected counts below 5: chi-square
## approximation may be incorrect
```

```
## Chi-square test with continuity correction for Hardy-Weinberg equilibrium (autosomal)
## Chi2 =  106.2512 DF =  1 p-value =  6.495738e-25 D =  0.002336449 f =  -0.004694836
```

```r
print(chi_square_correction_result)
```

```
## $chisq
## [1] 106.2512
##
## $pval
## [1] 6.495738e-25
##
## $D
## [1] 0.002336449
##
## $p
## [1] 0.004672897
##
## $f
## [1] -0.004694836
##
## $expected
##           AA           AB           BB
## 2.336449e-03 9.953271e-01 1.060023e+02
##
## $chi.contrib
##           AA           AB           BB
## 1.060023e+02 2.465008e-01 2.336449e-03
```

```r
# Exact test
exact_test_result <- HWExact(table(variant_column))
```

```
## Haldane Exact test for Hardy-Weinberg equilibrium (autosomal)
## using SELOME p-value
## sample counts: nAA =  106 nAB =  1 nBB =  0
## H0: HWE (D==0), H1: D <> 0
## D =  0.002336449 p-value =  1
```

```r
print(exact_test_result)
```

```
## $pval
## [1] 1
##
## $prob
## 1
## 1
##
## $pofthesample
## 1
## 1
```

```r
# Permutation test
permutation_test_result <- HWPerm(table(variant_column))
```

```
## Permutation test for Hardy-Weinberg equilibrium
## Observed statistic: 0.002358439   17000 permutations. p-value: 1
```

```r
print(permutation_test_result)
```

```
## $stat
## [1] 0.002358439
##
## $pval
## [1] 1
```

```r
# Question 4
genetic_data_rs <- non_monomorphic_data
genotype_counts_matrix <- matrix(NA, ncol = 3, nrow = ncol(genetic_data_rs))
for (i in 1:ncol(genetic_data_rs)) {
  variant_column <- genetic_data_rs[[i]]
  genotype_counts <- table(variant_column)

  genotype_counts <- genotype_counts[c("0", "1", "2")]

  genotype_counts_matrix[i, ] <- as.numeric(genotype_counts)
}

colnames(genotype_counts_matrix) <- c("AA", "AB", "BB")

# Question 5
genetic_data_no_header <- genetic_data_rs[-1, ]

hwe_results <- c()

# Loop through each SNP column
for (col_name in names(genetic_data_no_header)) {
  # Extract the SNP column
  variant_column <- as.integer(genetic_data_no_header[[col_name]])
  variant_column <- factor(variant_column, levels = c(0, 1, 2), labels = c("AA", "AB", "BB"))

  # Create a matrix X with genotype counts for the SNP
  X <- matrix(table(variant_column), ncol = 3, byrow = TRUE)

  # Apply HWExactStats to the matrix X
  hwe_result <- HWExactStats(X, x.linked = FALSE, plinkcode = TRUE, midp = FALSE)

  # Collect the p-value for each SNP
  hwe_results <- c(hwe_results, hwe_result)
}

# Calculate the percentage of significant SNPs (p-value < 0.05)
significant_snps_percentage <- mean(hwe_results < 0.05) * 100

cat("Percentage of significant SNPs at alpha = 0.05:", significant_snps_percentage, "%\n")
```

```
## Percentage of significant SNPs at alpha = 0.05: 2.761965 %
```

```r
# Question 6

most_significant_index <- which.min(hwe_results)

most_significant_snp_value <- genetic_data_rs[[most_significant_index]][1]
```

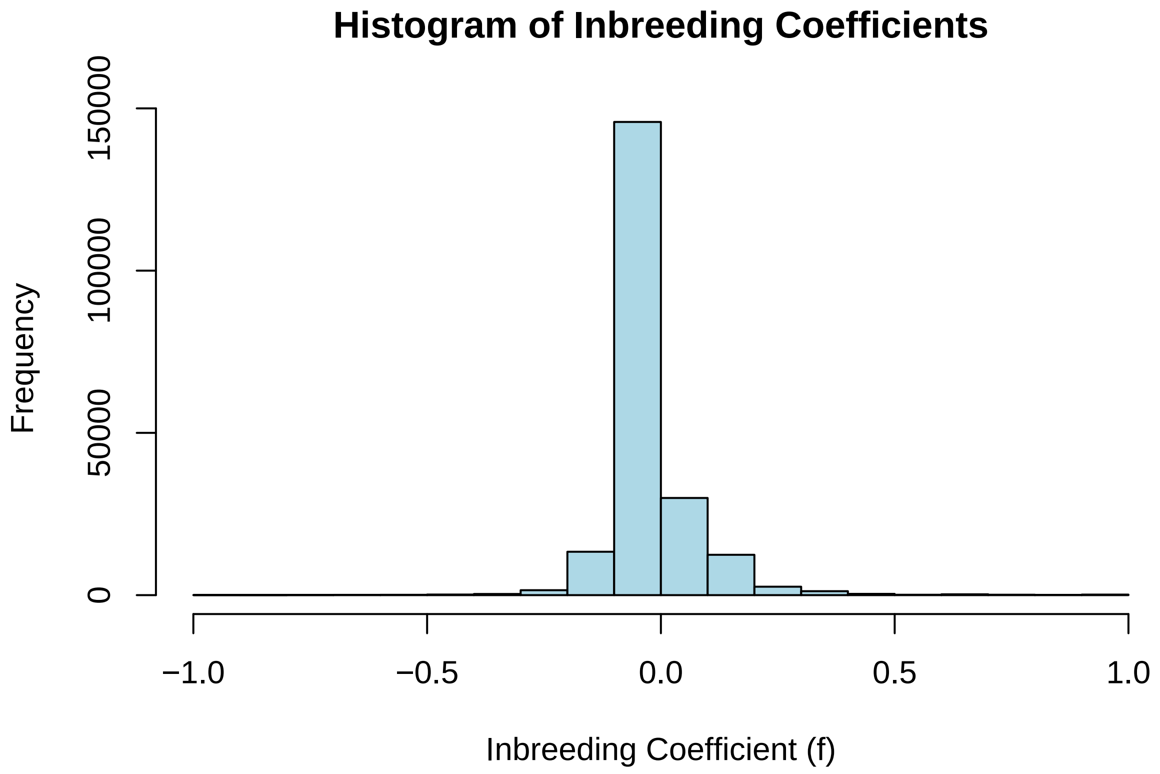
```r
most_significant_p_value <- hwe_results[most_significant_index]

cat("Most Significant SNP:", most_significant_snp_value, "\n")
```

```
## Most Significant SNP: rs2629366_C
```

```r
cat("P-value for Most Significant SNP:", most_significant_p_value, "\n")
```

```
## P-value for Most Significant SNP: 9.784766e-33
```

```r
# Question 7

inbreeding_coefficients <- c()

# Loop through each SNP column
for (col_name in names(genetic_data_no_header)) {
  # Extract the SNP column
  variant_column <- as.integer(genetic_data_no_header[[col_name]])
  variant_column <- factor(variant_column, levels = c(0, 1, 2), labels = c("AA", "AB", "BB"))

  # Create a matrix X with genotype counts for the SNP
  X <- matrix(table(variant_column), ncol = 3, byrow = TRUE)

  # Apply HWf to calculate the inbreeding coefficient
  f <- HWf(X)

  # Store the inbreeding coefficient
  inbreeding_coefficients <- c(inbreeding_coefficients, f)
}

# Make a histogram of f
hist(inbreeding_coefficients, main = "Histogram of Inbreeding Coefficients", xlab = "Inbreeding Coeffic
```



```
# Calculate and print descriptive statistics
mean_f <- mean(inbreeding_coefficients)
sd_f <- sd(inbreeding_coefficients)

cat("Descriptive Statistics for Inbreeding Coefficients:\n")
```

```
## Descriptive Statistics for Inbreeding Coefficients:
```

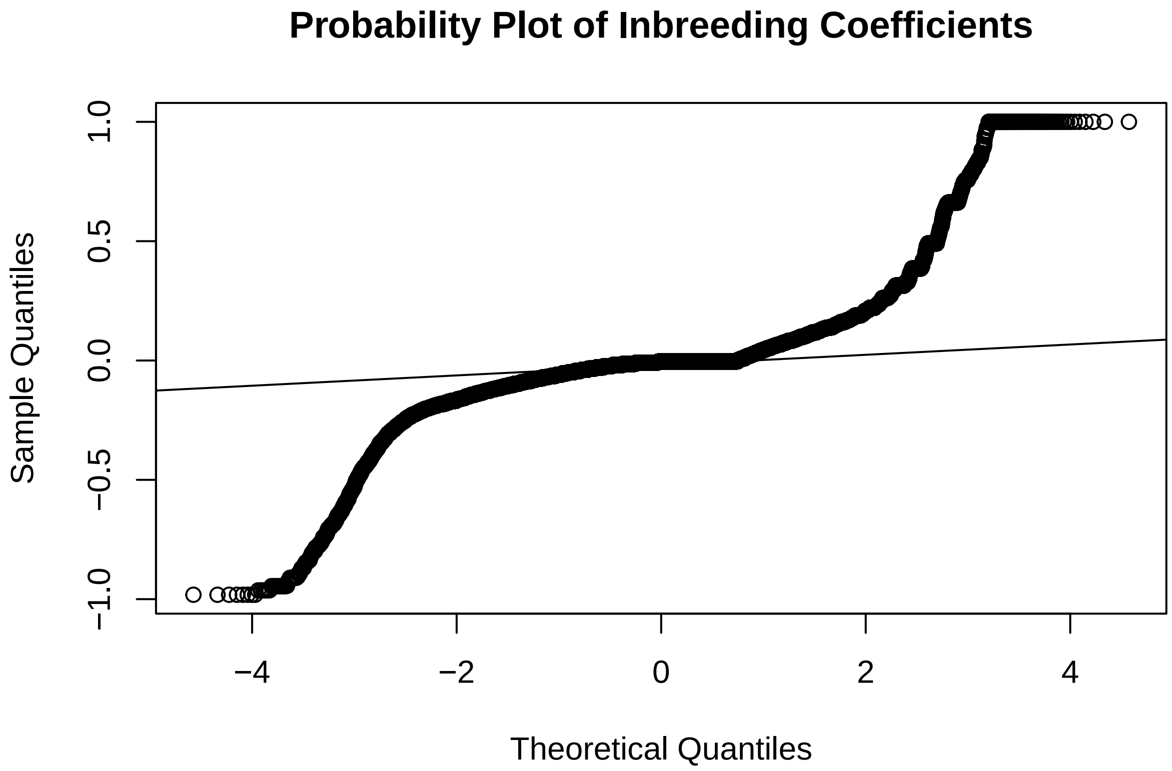
```
cat("Mean:", mean_f, "\n")
```

```
## Mean: -0.004668641
```

```
cat("Standard Deviation:", sd_f, "\n")
```

```
## Standard Deviation: 0.09486517
```

```
# Probability plot
qqnorm(inbreeding_coefficients, main = "Probability Plot of Inbreeding Coefficients")
qqline(inbreeding_coefficients)
```


Probability Plot of Inbreeding Coefficients
Sample Quantiles
1.0
0.5
0.0
−0.5
−1.0
−4
−2
0
2
4
Theoretical Quantiles

```
# Question 8
```