Problem Set 1

Valeriia Babaian: 23360186

Due: February 11, 2024

Question 1

The code shown below first generates 1,000 Cauchy random variables and obtains their empirical distribution, as indicated. Then, test statistic is calculated according to the given instructions, with ECDF standing for empirical and pnorm(data) for theoretical distribution. D equals to ≈ 0.135. *p-value* calculated from the Kolmogorov- Smirnoff CDF is low, allowing us to reject H_0 stating that the empirical distribution corresponds to the expected theoretical one with the mean equal to 0 and standard deviation equal to 1. Taken peculiarities of Cauchy variables, such as heavier tails, this result is not surprising.

```r
set.seed(123)
## generate 1,000 Cauchy random variables
data <- rcauchy(1000, location = 0, scale = 1)

# create empirical distribution of observed data
ECDF <- ecdf(data)
empiricalCDF <- ECDF(data)
# generate test statistic
D <- max(abs(empiricalCDF - pnorm(data)))

D #0.1347281
# Smaller values indicate similarity of the empirical distribution
# and the queried theoretical distribution.

# putting D in the p-value formula:
p_value <- sqrt(2*pi)/0.1347281*sum(exp(-((2*(1:1000)-1)^2*pi^2)/(8*0.1347281^2)))
p-value #5.65274e-29, reject H0 at 0.999 confidence level
```

Question 2

The code below estimates an OLS regression that uses BFGS optimization and compares it with regular `lm` results obtained in R. First, a function minimizing squared residuals is created (rss) to be used as a function in the BFGS optimization. The outcome of the function is the vector `beta` with two values: the intercept and the predictor coefficient.

```
1 set.seed (123)
2 data <- data.frame(x = runif(200, 1, 10))
3 data$y <- 0 + 2.75*data$x + rnorm(200, 0, 1.5)
4 
5 ## create a function that returns the residual sum of squares to be minimized.
6 # The function accepts beta as a vector with 2 values provided below:
7 rss <- function(beta, x, y) {
8   y_hat <- beta[1] + beta[2]*x
9   return(sum((y - y_hat)^2))
10 }
11 
12 # Optimize using built-in BFGS minimizing the RSS, assuming both coefficients are 0
13 bfgs <- optim(fn = rss, par = c(0, 0), x = data$x, y = data$y,  method = "BFGS")
14 
15 # Extract the best set of parameters found
16 bfgs <- bfgs$par
17 bfgs # 0.1391778 ; 2.7267000
18 
19 # Fit OLS via lm
20 m1 <- lm(y ~ x, data = data)
21 m1$coef #   0.1391874  ; 2.7266985
```

Best parameters obtained within BFGS optimization shown in the code output in the rows №17 and 21 are almost identical to each other.