Problem Set 1

Tolga Bag - 23371290

Due: February 11, 2024

Instructions

- Please show your work! You may lose points by simply writing in the answer. If the problem requires you to execute commands in `R`, please include the code you used to get your answers. Please also include the `.R` file that contains your code. If you are not sure if work needs to be shown for a particular problem, please ask.
- Your homework should be submitted electronically on GitHub in `.pdf` form.
- This problem set is due before 23:59 on Sunday February 11, 2024. No late assignments will be accepted.

Question 1

The Kolmogorov-Smirnov test uses cumulative distribution statistics test the similarity of the empirical distribution of some observed data and a specified PDF, and serves as a goodness of fit test. The test statistic is created by:

$$D = \max_{i=1:n} \left\{ \frac{i}{n} - F_{(i)}, F_{(i)} - \frac{i-1}{n} \right\}$$

where F is the theoretical cumulative distribution of the distribution being tested and $F_{(i)}$ is the ith ordered value. Intuitively, the statistic takes the largest absolute difference between the two distribution functions across all x values. Large values indicate dissimilarity and the rejection of the hypothesis that the empirical distribution matches the queried theoretical distribution. The p-value is calculated from the Kolmogorov- Smirnoff CDF:

$$p(D \leq d) = \frac{\sqrt{2\pi}}{d} \sum_{k=1}^{\infty} e^{-(2k-1)^2\pi^2/(8d^2)}$$

which generally requires approximation methods (see Marsaglia, Tsang, and Wang 2003). This so-called non-parametric test (this label comes from the fact that the distribution of the test statistic does not depend on the distribution of the data being tested) performs poorly in small samples, but works well in a simulation environment. Write an `R` function that implements this test where the reference distribution is normal. Using `R` generate 1,000 Cauchy random variables (`rcauchy(1000, location = 0, scale = 1)`) and perform the test (remember, use the same seed, something like `set.seed(123)`, whenever you're generating your own data).

As a hint, you can create the empirical distribution and theoretical CDF using this code:

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

```r

######################
# Problem 1
######################
# I prepare a function to perform Kolmogorov-Smirnoff test with normal reference
#distribution by researching online and with the help of ChatGPT.
ks_test <- function(data) {
  # Empirical cumulative distribution function per the code provided
  ECDF <- ecdf(data)
  empiricalCDF <- ECDF(data)

  # Theoretical cumulative distribution function (for normal distribution) per
  #the code provided
  theoreticalCDF <- pnorm(data)

  # I calculate the test statistic
  D <- max(abs(empiricalCDF - theoreticalCDF))

  # I calculate the p-value
  n <- length(data)
  p_value1 <- 1 - pnorm(sqrt(n) * D)

  # I return test statistic and p-value
  return(list(test_statistic = D, p_value1 = p_value1))
}

# settting seed for reproducibility as required
set.seed(123)

# I generate 1,000 Cauchy random variables
cauchy_data <- rcauchy(1000, location = 0, scale = 1)

# I perform the Kolmogorov-Smirnoff test
result <- ks_test(cauchy_data)
```

Test statistic is 0.1347281 and the p value is 1.019963. Test statistic is relatively small, indicating that the observed data and the normal distribution being tested against are relatively similar. In Kolmogorov-Smirnoff test, the null hypothesis is that the empirical distribution of the observed data matches the specified normal distribution. P is way above 0.05, so I fail to reject the null hypothesis, indicating that there is no significant difference between the observed datas empirical distribution and the specified normal distribution.

Question 2

Estimate an OLS regression in `R` that uses the Newton-Raphson algorithm (specifically `BFGS`, which is a quasi-Newton method), and show that you get the equivalent results to using `lm`. Use the code below to create your data.

```r
#####################
# Problem 2
#####################

set.seed (123)
data <- data.frame(x = runif(200, 1, 10))
data$y <- 0 + 2.75*data$x + rnorm(200, 0, 1.5)
# I define the objective function for OLS
objective_function <- function(beta, x, y) {
  y_pred <- beta[1] + beta[2] * x
  residuals <- y - y_pred
  sum(residuals^2)
}

# I am using the  BFGS algorithm to estimate OLS parameters: the initial guess
#and the optimum result per https://rpubs.com/aaronsc32/newton-raphson-method
# and a help from chatGPT
initial_guess <- c(0, 0)  # Initial guess for intercept and slope
optim_result <- optim(par = initial_guess, fn = objective_function,
                      x = data$x, y = data$y, method = "BFGS")

# I extract the coefficients from the optimization result
bfgs_intercept <- optim_result$par[1]
bfgs_slope <- optim_result$par[2]

# I print coefficients obtained using BFGS
cat("Intercept:", bfgs_intercept, "\n")
#Intercept is 0.1391778
cat("Slope:", bfgs_slope, "\n")
#slope is 2.7267

# I fit linear regression using lm() function for comparison
lm_model <- lm(y ~ x, data = data)
```

Intercept in the lm model is 0.1391874 and slope is 2.7266985, providing very identical methods (intercept of 0.1391778 and slope of 2.7267 for BFGS) in fitting the model to the data.