Problem Set 1

Applied Stats II

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:

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

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:

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

Answer Question 1

```
# Define function for the Kolmogorov-Smirnov test
ks_test <- function(data, theoretical_dist) {
n <- length(data)
empirical_cdf <- ecdf(data)
D <- max(abs(empirical_cdf(data) - theoretical_dist(data)))
p_value <- sqrt(2 * pi) / D * sum(exp(-(2 * (1:100) - 1)^2 * pi^2 / (8 * D
  ^2)))
return(list(test_statistic = D, p_value = p_value))
}
# Set the seed for reproducibility
set.seed(123)
# Generate 1,000 Cauchy random variables
cauchy_data <- rcauchy(1000, location = 0, scale = 1)
# Perform the Kolmogorov-Smirnov test using the normal distribution as the
  theoretical reference
result <- ks_test(cauchy_data, pnorm)
# View the test statistic and p-value
result$test_statistic
result$p_value
```

Explanation Answer 1

The test is used to compare the empirical distribution of observed data with a specified theoretical distribution, in this case, the normal distribution. The test statistic and the p-value obtained from this R code will help in determining whether the empirical distribution of the Cauchy random variables matches the normal distribution. The test statistic measures the largest absolute difference between the empirical and theoretical distribution functions, and the p-value indicates the level of similarity between the two distributions. A low p-value suggests that the two distributions are significantly different. The R code is used to compare two sets of numbers to see if they come from the same type of pattern. If the two sets of

numbers are very different, the test will give a small number. If they are very similar, the test will give a big number.

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.

```
D <- max(abs(empirical_cdf(data) - theoretical_dist(data)))
p_value <- sqrt(2 * pi) / D * sum(exp(-(2 * (1:100) - 1)^2 * pi^2 / (8 * D
  ^2)))
return(list(test_statistic = D, p_value = p_value))
```

Answer Question 2

```
# Set the seed for reproducibility
set.seed(123)
# Create the data
data <- data.frame(x = runif(200, 1, 10))
data$y <- 0 + 2.75*data$x + rnorm(200, 0, 1.5)
# Perform OLS regression using BFGS algorithm
model_bfgs <- optim(c(0, 0), function(beta) sum((data$y - beta[1] - beta[2]*
  data$x)^2), method = "BFGS")
# Extract the coefficients from the BFGS optimization
coefficients_bfgs <- model_bfgs$par
# Fit the OLS regression using lm
model_lm <- lm(y ~ x, data = data)
# Extract the coefficients from the lm model
coefficients_lm <- coef(model_lm)
# Compare the coefficients
coefficients_bfgs
coefficients_lm
```

Explanation Answer 2

This code involves estimating an OLS regression in R using the Newton-Raphson algorithm (specifically BFGS) and demonstrating its equivalence to using lm. This R code is used to estimate a linear relationship between two variables (x and y) using two different methods (BFGS and lm) and compare the results to show that they are equivalent. The BFGS algorithm is a type of optimization algorithm used to find the best fit for the linear relationship, while the lm function is a built-in function in R that performs the same task. The code shows that both methods produce the same coefficients, thereby demonstrating their equivalence.