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 1

```
1
2  # Create function to calculate P value
3
4  calculate_p_value <- function(data) {
5    # Create empirical distribution
6    ECDF <- ecdf(data)
7    empiricalCDF <- ECDF(data)
8
9    # Calculate the theoretical CDF assuming a standard normal distribution
10   theoreticalCDF <- pnorm(data)
11
12   # Calculate the test statistic D
13   D <- max(abs(empiricalCDF - theoreticalCDF))
14
15   # Initialize p-value
16   p_value <- 0
17
18   # Loop to calculate the sum in the formula
19   for (k in 1:1000) {
20     p_value <- p_value + sqrt(2 * pi) / D * exp(-(((2 * k) - 1)^2) * pi^2 /
   (8 * D^2))
21   }
22
23   return(p_value)
24 }
25
26 set.seed(123)
27 data <- rcauchy(1000)
28
29 p_value <- calculate_p_value(data)
30 p_value = 5.652523e-29
31
32
33 # Perform Kolmogorov-Smirnov test
34 ks_result <- ks.test(data, "pnorm")
35
36 # Print the result
37 print(ks_result)
38
39   Asymptotic one-sample Kolmogorov-Smirnov test
40
41 data:  data
42 D = 0.13573, p-value = 2.22e-16
43 alternative hypothesis: two-sided
44
45
46
```

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.

```
set.seed (123)
data <- data.frame(x = runif(200, 1, 10))
data$y <- 0 + 2.75*data$x + rnorm(200, 0, 1.5)
```

Answer 2

```
set.seed(123)
data <- data.frame(x = runif(200,1,10))
data$y <- 0 + 2.75*data$x + rnorm(200, 0, 1.5)

# Run with LM

model <- lm(y ~ x, data = data)
summary(model)

Coefficients:
             Estimate   Std. Error t value Pr(>|t|)
(Intercept)  0.13919    0.25276    0.551    0.582
x            2.72670    0.04159   65.564   <2e-16 ***
---

Residual standard error: 1.447 on 198 degrees of freedom
Multiple R-squared:  0.956, Adjusted R-squared:  0.9557

# Estimate an OLS in R using Newton-Raphson (BFGS)

nr_function <- function(beta, x, y) {
  y_hat <- beta[1] + beta[2] * x
  sum((y - y_hat)^2)
}

initial_guess <- c(0, 1) # Initial guess for intercept and slope
result <- optim(par = initial_guess, fn = nr_function, x = data$x, y = data$
  y, method = "BFGS")

# Extract the coefficients
coefficients <- result$par
print(coefficients)

# Result is the same as the LM result
[1] 0.139187 2.726699
```

```
36
```