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:

```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
#####################
# Function to calculate Kolmogorov-Smirnov test statistic and p-value
ks_test_custom <- function(data) {
  n <- length(data)
  sorted_data <- sort(data)
  empirical_cdf <- (1:n) / n
  theoretical_cdf <- pnorm(sorted_data)

  D <- max(abs(empirical_cdf - theoretical_cdf))

  # Calculate p-value using Kolmogorov-Smirnov CDF
  p_value <- sqrt(2*pi)/D * sum(exp(-(2*(1:n)-1)^2*pi^2 / (8*D^2)))

  return(list(D = D, p_value = p_value))
}

# Set seed for reproducibility
set.seed(123)

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

# Perform the custom Kolmogorov-Smirnov test
ks_result <- ks_test_custom(cauchy_data)

# Print test result
print(ks_result)
```

```
The result below:
$D
[1] 0.1347281
$p_value
[1] 5.652523e-29
```

Define the null hypothesis:
the null hypothesis would be that the observed data follows a normal distribution.
Calculate the p-value: If the p-value is less than the chosen significance level (e.g.,$\alpha = 0.05$), reject the null hypothesis and conclude that the observed data does not follow a normal distribution. Otherwise, fail to reject the null hypothesis.
We can see that the p-value(5.652523e-29) is not equal or below the $\alpha = 0.05$, the random data not follow the normally distributed.

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

```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)

# Set seed for reproducibility
set.seed(123)

# Create data
data <- data.frame(x = runif(200, 1, 10))
```

```r
data$y <- 0 + 2.75 * data$x + rnorm(200, 0, 1.5)

# Define objective function for OLS regression
ols_obj <- function(beta, x, y) {
  y_pred <- beta[1] + beta[2]*x
  residuals <- y - y_pred
  sum(residuals^2)
}

# Initial values for beta (intercept and slope)
beta_init <- c(1, 1)

# Optimize objective function using BFGS algorithm
optim_result <- optim(par = beta_init, fn = ols_obj, method = "BFGS", x = data$x, y = data$y)

# Extract estimated coefficients
beta_hat <- optim_result$par
```

```
print(beta_hat)
[1] 0.1391874 2.7266985

 print(summary(lm_result)$coefficients)
Estimate Std. Error    t value      Pr(>|t|)
(Intercept) 0.1391874 0.25275645  0.5506778  5.824754e-01
x           2.7266985 0.04158811 65.5643750 3.134842e-136
```

We can see that the result of Newton-Raphson and OLS regression is same. It indicates that the optimization algorithm has successfully converged to the same solution as the standard OLS regression method.