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:

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

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:

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

Kolmogorov-Smirnov test function

```
# Create Kolmogorov-Smirnov test function
kolsmir <- function(data, p_value = 0.05){
  # Sort input data
  data_sorted <- sort(data)

  # Calculate empirical CDF values
  ecdf_values <- (1: length(data_sorted)) / length(data_sorted)

  # Calculate theoretical CDF values
  tcdf_values <- pcauchy(data_sorted, location = 0, scale = 1)

  # Calculate maximum absolute difference between empirical and theoretical
  kolsmir_statistic <- max(abs(ecdf_values - tcdf_values))

  # Calculate p-value
  kolsmir_p_value <- (sqrt(2 * pi)/kolsmir_statistic) * sum(exp( -((2 * nrow(
    data_sorted)) - 1)^2 * pi^2 / (8 * kolsmir_statistic^2)))

  # Confirm
  kolsmir_result <- kolsmir_p_value < p_value

  return(list("KS_Test_Statistic" = kolsmir_statistic, "KS_P-Value" = kolsmir_
    p_value, "KS_Test_Result" = kolsmir_result))
}
```

Implement custom function with the generated data. Test confirms TRUE.

```
# Implement Kolmogorov-Smirnov test function with generated data
data_result <- kolsmir(data = data)
print(data_result)
```

Lastly, Compare the given Reference values with the Custom Outputs

```
# Compare Reference and Custom Function Outputs
KS_compare <- data.frame("Custom Function" = data_result$KS_Test_Statistic, "
    Reference" = D, "Difference" = abs(data_result$KS_Test_Statistic - D))
# Show how the Test Statistics compare between my custom function and the
print(KS_compare)
```

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.

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

# Use optim() function to estimate OLS regression
# Set starting values for the parameters as c(1, 1) to be optimised over
bfgs_model <- optim(par = c(1, 1),
```

The parameters are set to $c(1, 1)$ as a starting point for the algorithm to optimise over in order to reach the MLE. Note the RSS function is defined as a lambda function, though alternatively could be defined beforehand within the main or same function environment.

```

# Use optim() function to estimate OLS regression
# Set starting values for the parameters as c(1, 1) to be optimised over
bfgs_model <- optim(par = c(1, 1),
                    # Set BFGS algorithm for the Newton-Raphson optimisation method
                    method = "BFGS",
                    # Lambda function calculates the residual sum of squares
                    # (ie differences between predicted and observed values)
                    fn = function(coef) {
```

Then the built-in lm() function is used for comparison:

```

# Generate OLS regression using lm() for comparison
```

Now the coefficients are extracted for comparison:

```

# Extract coefficients for comparison
lm_coefficients <- lm_compare$coefficients
```

Lastly, demonstrate the coefficients are the same, confirming my custom model functions correctly.

```

# Compare outputs using all.equal() function in base R, ignoring differences in attributes such as column names.
model_check <- all.equal(lm_coefficients, bfgs_coefficients, check.attributes = FALSE)
```