Problem Set 1

Yajie Dong

February 11, 2024

Question 1

Step 1: Understanding the Test Statistic

The test statistic D measures the maximum absolute difference between the empirical distribution function(EDF) of the sample and the theoretical cumulative distribution function (CDF) of the hypothesized distribution. This statistic is pivotal in measuring the deviation of the observed data from the expected theoretical model.

Step 2: Calculating D

The calculation of D involves sorting the sample data, computing the EDF, and identifying the maximum discrepancy between the EDF and the theoretical CDF, as denoted by:

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

where $F_{(i)}$ represents the value of the theoretical CDF at the i-th ordered data point.

Now, Let's calculating it in R:

Generate Sample Data:

```r
# Load necessary library
if (!requireNamespace("stats", quietly = TRUE)) install.packages("stats")
library(stats)

# 1. Set the seed for reproducibility
set.seed(123)

# 2. Generate 1,000 Cauchy random variables
data <- rcauchy(1000, location = 0, scale = 1)
```

Calculate the Empirical CDF:

```r
# 3. Create the empirical CDF from the generated data
ECDF <- ecdf(data)
empiricalCDF <- ECDF(data)
```

Calculate the Theoretical CDF:

It's crucial to align the theoretical CDF with the distribution under test. Here, we assume a standard normal distribution for comparison purposes, although the choice of distribution should be guided by the context of the data or specific research questions.

```
# 4. Calculate the theoretical CDF using a normal distribution
# Assuming a standard normal distribution (mean=0, sd=1)
theoreticalCDF <- pnorm(data, mean=0, sd=1)
```

Calculate the Test Statistic$D$:

```
# 5. Determine the test statistic D
D <- max(abs(empiricalCDF - theoreticalCDF))

# Display the test statistic
print(paste("Test Statistic D:", D))
```

The result we get from R:

$$\text{Test Statistic } D = 0.13472806160635 \tag{2}$$

Interpretation of the test statistic $D = 0.13472806160635$:
Given the test statistic $D = 0.13472806160635$, this represents the maximum distance between the empirical Cumulative Distribution Function (CDF) from our sample and the theoretical CDF of a standard normal distribution. This distance highlights how much our observed data differs from the expected under the normal model. The size of D suggests a significant deviation, potentially indicating a notable difference that, pending the p-value analysis in the next step, may lead to rejecting the null hypothesis that the distributions are identical.

Step 3: Using `ks.test()` in R

The `ks.test()` function offers a comprehensive approach to performing the Kolmogorov-Smirnov test, yielding both the test statistic and the p-value, thereby facilitating the validation of the manually computed D statistic.

The p-value is calculated from the Kolmogorov-Smirnoff CDF:

$$p(D \leq d) = \frac{\sqrt{2\pi}}{d} \sum_{k=1}^{\infty} e^{-(2k-1)^2\pi^2/(8d^2)} \tag{3}$$

Now, Let's do that in R:

```
# 6. Using ks.test for validation (This is optional and serves as a check)
ks_result <- ks.test(data, "pnorm", mean = 0, sd = 1)

# Display ks.test results
print(ks_result)
```

The result we get from R:

Test	Asymptotic one-sample Kolmogorov-Smirnov test
Data	data
D Statistic	0.13573
p-Value	2.22×10^{-16}
Alternative Hypothesis	two-sided

Table 1: Summary of the Kolmogorov-Smirnov test results.

Note: The minor discrepancy between the manually calculated D and the D from `ks.test()` is likely due to computational nuances but does not affect the overall conclusion.

Step 4: Interpretation and Conclusion

The small p-value and test statistic D suggest a significant divergence from the null hypothesis, implying the observed data's distribution differs notably from the normal model. This outcome aligns with expectations given the Cauchy distribution's distinct properties versus the normal distribution. The choice of Cauchy for this analysis highlights the Kolmogorov-Smirnov test's adeptness at capturing distributional variances, especially between dissimilar distributions, underscoring its value in diverse analytical settings.

Question 2

The task is to estimate an Ordinary Least Squares (OLS) regression in R using the Newton-Raphson algorithm, specifically the BFGS method, which is a quasi-Newton method. We are expected to demonstrate that the results obtained using this approach are equivalent to those produced by the lm function in R. Now, let's do that step-by-step:

Step 1: Data Preparation in R:

```r
# Load necessary libraries
library(stats)  # For lm and optim functions

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

Step 2: Implement OLS Regression Using `lm`:

Use R's `lm` function to perform a linear regression with y as the response variable and x as the predictor. Let's do that in R:

```r
# OLS Regression using lm
lm_model <- lm(y ~ x, data = data)
print(summary(lm_model))
```

Term	Estimate	Std. Error		
(Intercept)	0.13919	0.25276		
x	2.72670	0.04159		
Residuals				
Min	1Q	Median	3Q	Max
-3.1906	-0.9374	-0.1665	0.8931	4.8032
Residual standard error: 1.447 on 198 degrees of freedom				
Multiple R-squared: 0.956, Adjusted R-squared: 0.9557				
F-statistic: 4299 on 1 and 198 DF, p-value: ¡ 2.2e-16				

Table 2: Summary of OLS Regression Results

Interpretation Of R output OLS Regression using `lm`:

The `lm` model's coefficients indicate that for every one-unit increase in x, y increases by approximately 2.727 units, assuming the model is a good fit to the data.

The p-value for the slope coefficient is $< 2 \times 10^{-16}$, indicating that the relationship between x and y is statistically significant.

The R^2 value is 0.956, suggesting that approximately 95.6% of the variability in y can be explained by x in this linear model.

Step 3: Implement OLS Regression Using Newton-Raphson (BFGS Method):

1.we use the **Residual Sum of Squares (RSS)** to help the BFGS method find the best-fitting regression model by minimizing the differences between the observed data and our model's predictions.

Let't do that in R:

```r
# Function to calculate residuals sum of squares (RSS)
rss <- function(beta, data) {
  y_pred <- beta[1] + beta[2] * data$x
  return(sum((data$y - y_pred) ^ 2))
}
```

2.Implement OLS Regression Using Newton-Raphson (BFGS Method)in R:

```r
# Initial parameter estimates for the Newton-Raphson algorithm
initial_params <- c(0, 1)  # Intercept and slope

# OLS Regression using Newton-Raphson (BFGS Method)
optim_result <- optim(par = initial_params, fn = rss, data = data, method = "BFGS")
print(optim_result)
```

Parameter	Value
par[1]	0.139187
par[2]	2.726699
Statistic	**Value**
value	414.4577
Counts	
function	28
gradient	5
Convergence	0
Message	NULL

Table 3: Optimization Results from R

Interpretation Of R output OLS Regression using Newton-Raphson (BFGS Method):

Optimized Coefficients (using the `optim` function):

- Intercept: 0.139187
- Slope (Coefficient of x): 2.726699

The BFGS method yielded nearly identical coefficients to those produced by the `lm` function, demonstrating the accuracy and effectiveness of this optimization method for OLS regression.

Step 4: Compare Results:

Finally,the coefficients from both the `lm` model and the BFGS optimization are displayed for comparison.

Let't do that in R:

```r
# Compare the coefficients
cat("Coefficients from lm:\n")
print(coef(lm_model))
cat("Coefficients from BFGS:\n")
print(optim_result$par)
```

Method	Intercept	x
Coefficients from `lm`	0.1391874	2.7266985
Coefficients from BFGS	0.139187	2.726699

Table 4: Comparison of Coefficients from `lm` and BFGS Optimization

Interpretation:

The equivalence of the coefficients from both methods confirms that the Newton-Raphson algorithm with the BFGS method is a valid approach for estimating OLS regression parameters, yielding results consistent with the conventional `lm` function in R. The very close match between the two sets of coefficients (intercept and slope) illustrates that both approaches are robust and reliable for linear regression analysis, at least for this dataset.

Step 5: Conclusion:

We have successfully demonstrated that the Newton-Raphson algorithm with the BFGS method can be used to estimate the coefficients of an OLS regression model, yielding results equivalent to those obtained from the standard `lm` function in R.