# Applied Stats II - Problem Set 1

Imelda Finn (22334657)

Due: February 12, 2023

Code in PS1_ImeldaFinn.R

## Question 1

The Kolmogorov-Smirnov test uses cumulative distribution statistics to 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 $i$th 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).

The code for the kolmogorov_smirnov function is:

```
# function to return values for kolmogorov-smirnov test
#   - one-sample, comparison to normal PDF
# input: empirical data
# output: D-statistic, D+, D-, K-statistic (=D*sqrt(N)), P(K)
```

```
kolmogorov_smirnov <- function (dat) {
  # get number of elements in data
  N <- length(dat)
  # convert data to ECDF
  ECDF <- ecdf(dat)
  empiricalCDF <- ECDF(dat)
  # calculate test statistic (D)
  Dmin <- min(empiricalCDF - pnorm(dat))
  Dmax <- max(empiricalCDF - pnorm(dat))
  D <- max(abs(c(Dmin, Dmax)))

  #calculate critical value (K-statistic)
  K <- D * sqrt(N)

  # get probability of calculated test statistic
  k = seq(1,N)

  #calculate q value for P(K)
  qval <- sum(exp(-1*((2*k - 1)^2)*(pi^2)/(8 * K^2)))
  qval <- qval * sqrt(2 * pi) * (1/K)

  # return results
  res <- tibble('D'= D, 'Dmax' = Dmax, 'Dmin' = Dmin,
    'K'=K, 'pval'=1-qval)
  return(res)
}
```

The test data was generated using the `rcauchy` function, and is summarised in Table 1.

```
    # create empirical distribution of observed data
    set.seed(123)
    N <- 1000
    data <- rcauchy(n=N, location = 0, scale = 1)

```

Table 1: Data Summary Table

| Statistic | N | Mean | St. Dev. | Min | Max |
|---|---|---|---|---|---|
| Observed_CDF | 1,000 | 0.500 | 0.289 | 0.001 | 1.000 |
| Normal | 1,000 | 0.505 | 0.358 | 0.000 | 1.000 |

Table 2: Kolmogorov-Smirnov Test results

| | value |
|---|---|
| D | 0.13473 |
| Dmax | 0.12356 |
| Dmin | -0.13473 |
| K | 4.26048 |
| pval | 0 |

one sample, two sided, normal; alpha 0.05; D alpha: 0.04301

A hypothesis test was carried out using results from the `kolmogorov_smirnov`, function:

```
ks_results <-kolmogorov_smirnov(data)
# print(ks_results)
# A tibble: 1 x 5
       D  Dmax    Dmin     K      pval
   <dbl> <dbl>   <dbl> <dbl>     <dbl>
1 0.135 0.124 -0.135  4.26 2.22e-16

```

## Hypothesis Test

1. $H_0$ the observed data is normally distributed
2. $H_a$ the observed data is not normally distributed
3. the test statistic $D$ is 0.135
4. the $\alpha$ value is 0.05
5. the critical value $K$ is 4.26
6. the `pvalue` is $2.22 \times 10^{-16}$ (ie the probability of observing these values if the data was normally distributed is approximately 0)
7. as `pvalue` is less than $\alpha$ we **reject** the null hypothesis ($D_{\alpha,N} = 0.04301$, we reject $H_\alpha$ if $D > D_{\alpha,N}$)[1]

There is insufficient evidence to support the hypothesis that our observed data is normally distributed.

The results are summarised in Table 2.

The results were checked using the `cont_ks_test` function from the `KSgeneral` package, and the same results were obtained.[2]

```
# KSgeneral::cont_ks_test(data, "pnorm")

  One-sample Kolmogorov-Smirnov test

data:  data
D = 0.13573, p-value < 2.2e-16
alternative hypothesis: two-sided
```

[1]https://www.itl.nist.gov/div898/handbook/eda/section3/eda35g.htm
[2]these agreed with the results with the default `ks.test` function.

# 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.

From Eliason[3], when $Y \sim N(X\beta, \sigma^2)$ OLS and MLE estimators for the $\beta$ are equivalent. Therefore we can apply MLE estimation using the optim function with method $= BGFS$.

The data was generated as follows:

```
set.seed (123)
data2 <- data.frame(x = runif(200, 1, 10))
data2$y <- 0 + 2.75*data2$x + rnorm(200, 0, 1.5)


```

[3]Eliason, Scott R, *Maximum Likelihood Estimation: Logic and Practice*, 1993, p17

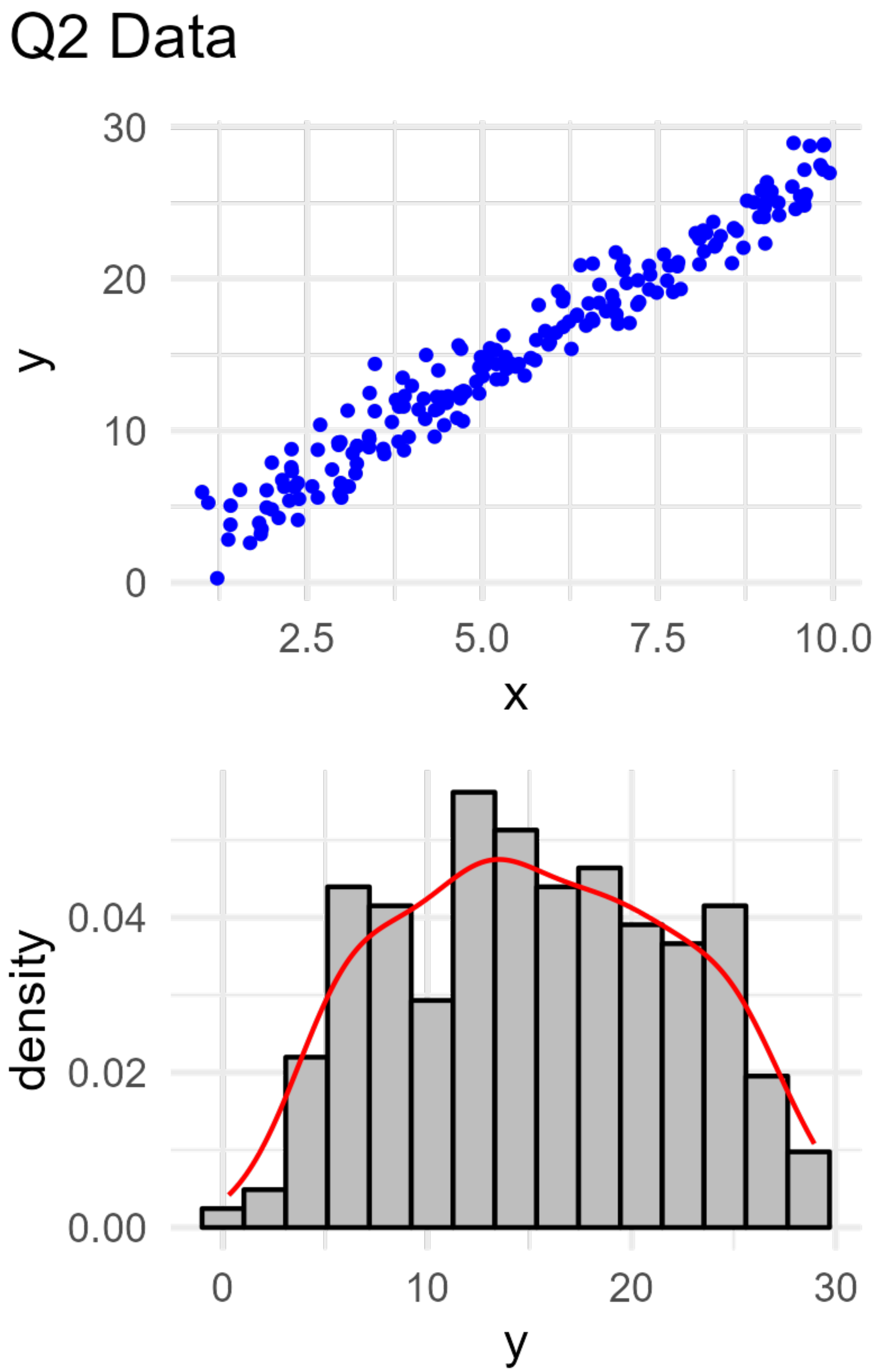


Figure 1: Q2 Data

The code to implement the Maximum Likelihood Estimation/OLS is:

```r
# define likelihood function to be solved by optim
linear.lik <- function(theta, y, X){
  n <- nrow(X )
  k <- ncol(X )
  beta <- theta[ 1 : k ]
  sigma2 <- theta[ k + 1 ] ^ 2
  e <- y - X%*%beta
  logl <- -.5 *n* log(2 * pi)-.5 *n* log(sigma2)-(( t(e)%*%
                                                      e)/(2 * sigma2))
  return(-logl )
}

#  use optim to get MLE estimators for parameters
linear.MLE <- optim(fn = linear.lik, par = c(theta=1, y=1, X=1),
        hessian =TRUE, y = data2$y, X= cbind(1, data2$x), method = "BFGS")

# get se for the parameters
linear.MLE$se <- sqrt(diag(solve(linear.MLE$hessian)))
```

The results from the function are $\hat{\beta}_0 = 0.1398324, \hat{\beta}_1 = 2.7265559, \hat{\sigma}^2 = 1.4390716$ :

```r
    linear.MLE$par
    #theta          y          X
    #0.1398324   2.7265559  -1.4390716

    linear.MLE$se
    #       theta              y            X
    #0.25140690  0.04136606  0.07191798

```

The linear regression model was called:

```r
linear.lm <- lm(y ~ x, data2)
```

The results for the MLE and linear models are in Table 3:
The prediction equation is: $y = 0.1398324 + 2.7265559 \times x$.
$\hat{y} = 0.13983(\theta) + 5.55753(\bar{x}) * 2.72656(\beta) = 15.29274$
$\bar{y} = 15.29289$

Table 3:

| | *Dependent variable:* | |
|---|---|---|
| | y | |
| | MLE | Linear |
| theta | 0.1398324 | |
| | (0.251) | |
| y | 2.7265559 | |
| | (0.041) | |
| X | -1.4390716 | |
| | (0.072) | |
| x | | $2.727^{***}$ |
| | | (0.042) |
| Constant | | 0.139 |
| | | (0.253) |
| Observations | 200 | 200 |
| $R^2$ | | 0.956 |
| Adjusted $R^2$ | | 0.956 |
| Residual Std. Error (df = 198) | | 1.447 |
| F Statistic | | $4{,}298.687^{***}$ (df = 1; 198) |
| *Note:* | | $^{*}$p<0.1; $^{**}$p<0.05; $^{***}$p<0.01 |