MATH20811 - Practical Statistics

Coursework 3 Submission

In this report, I will be performing two types of tests: Goodness of Fit and Monte Carlo Sampling according to the questions provided on Blackboard. In this coursework, I aim to showcase the skills I have gained throughout the first semester of my second year learning Practical Statistics.

Part 1: Goodness of Fit

The aim for this section of the report is to investigate whether the univariate sample data given on Blackboard could be regarded as a random sample from a N(0,1) distribution.

Histogram

To start off, the data from the .txt file was standardised using its sample mean and sample deviation then to produce a histogram and a kernel density estimate (KDE) for a comparison with N(0,1)'s pdf. The product is shown below, following by the R codes used to compute the result.

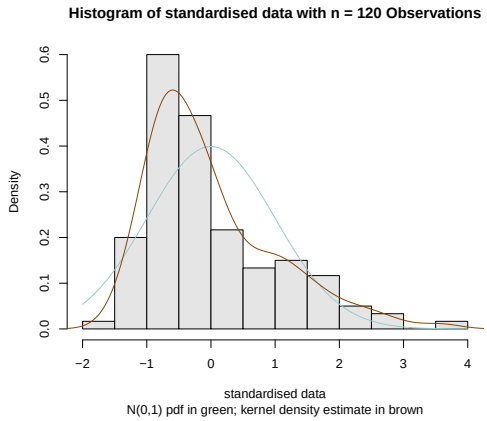

Figure 1: Histogram of standardised data with the N(0,1) pdf and KDE

```
> data <- read.table("~/Desktop/cw3_data.txt")$V1 #extract data
    from .txt file on Blackboard

    > n <- length(data)
    > mu <- mean(data)
    > sd <- sd(data)

    > x <- (data - mu)/sd #obtain standardised data

    > min(x);max(x) #for graph plotting
    [1] -1.505587
    [1] 3.572706

    > hist(x, freq = F, xlim = c(-2, 4), main = "Histogram of
    standardised data with n = 120 Observations", sub = "N(0,1)
    pdf in green; kernel density estimate in brown",
    col = "grey90", xlab = "standardised data")

    > xx <- seq(from = -2, to = 4, length.out = 400)
    > dxx <- dnorm(xx)
    > lines(xx, dxx, col = "paleturquoise3") #N(0,1) pdf
    > lines(density(x), col = "darkorange4") #KDE
```

It can be seen that from the graph, that the KDE of the standardised data did not match well with the pdf of the Normal distribution, although there was an exception of some hint of similarities at the tails of the graph, but this is not strong enough to show that it is a good example of a N(0,1) distribution.

Q-Q Plot

Off to a bad start with the histogram, a Normal quantile-quantile plot (Q-Q plot) of the standardised data was then constructed to cotinue to search for more evidences to conclude the investigation, along with a reference line for a Normality comparison.

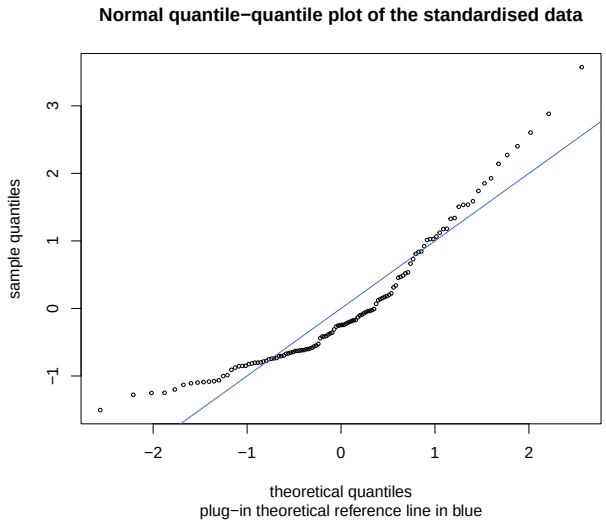

Figure 2: Q-Q plot of the standardised dat

R codes used to compute the plot are as such:

```
> sq <- sort(x)
> a = 0
> for(i in 1:n){
    a[i] = ((i-3/8)/(n+1/4))
  }
> tq <- qnorm(a)

> plot(tq, sq, main = "Normal quantile-quantile plot of
the standardised data", sub = "plug-in theoretical
reference line in blue", cex = 0.5, xlab = "theoretical
quantiles", ylab = "sample quantiles")
> abline(0, 1, col = "royalblue") #reference line
```

Judging by the Q-Q plot, the results are similar to what was concluded from the histogram plot - it is very likely that the standardised data is not a random sample of a N(0,1) distribution. The majority of the plot did not lie along the reference line, not being close to their corresponding theoretical quantiles, indicating an unfit of Normality.

Kolmogorov-Smirnov Test

A Kolmogorov-Smirnov (KS) test was also carried out to assess the validity of the assumption, along with the largest absolute difference between the empirical CDF and the N(0,1)'s cdf. The result conducted using the code *ks.test* is as follows:

	D	pvalue
Data	0.14511	0.01278

Table 1: KS test of the standardised data

the R codes:

```
> test <- ks.test(x, y = pnorm, alternative = c("two.sided"))

#converting to a dataframe to make it easier to transfer to
LaTeX
> testdf <- data.frame(
    D = test$statistic,
    pvalue = test$p.value
  )

> rownames(testdf) <- c("Data")

> install.packages("xtable")
> library(xtable)
> xtable(testdf, digits = 5
```

The test results further proved that the standardised data cannot be regarded as a random sample from the N(0,1) distribution at 5% significance level as its p-value is smaller than 0.05, which backed up the comments given on the two graphs previously.

The standardised data value where the maximum absolute difference between the empirical CDF and the N(0,1) CDF lies can be computed by using the directional deviations (D_n^+, D_n^-):

```
> x.ecdf <- (1:n)/n
> y <- pnorm(sq)
>
> diff1 = x.ecdf - y
> md1 = max(diff1)
> dnplus = max(md1, 0)
> ks1 = sq[dnplus == diff1]
>
> diff2 = y - x.ecdf
> md2 = max(diff2)
> md2 = md2 + (1/n)
> dnminus = max(md2, 0)
> ks2 = sq[dnminus == diff2 + (1/n)]
> ks = max(dnminus, dnplus)
> ks
[1] 0.1451055
> if(dnminus < dnplus) ksstat = abs(ks1)
> if(dnminus > dnplus) ksstat = abs(ks2)
> ksstat
[1] 0.008620076
```

From the R code above, the standardised data value as mentioned is 0.008620076, which will be indicated on the next graph. Additionally, the maximum value between D_n^+ and D_n- is 0.1451055, which matched the D-value of the KS test performed and shown in Table 1.

Empirical CDF

Lastly, the empirical cdf (ECDF) of the standardised data and the N(0,1) cdf will be showcased, along with the indication of the point where the maximum difference between two plots occur, as mentioned.

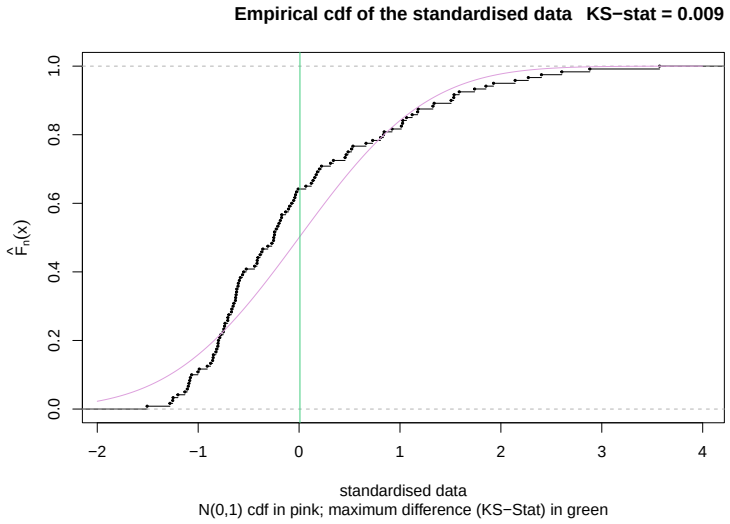

Figure 3: Empirical cdf of the standardised data

```
> plot.ecdf(x, main = "Empirical cdf of the standardised
data", cex = 0.3, ylab = "", xlab = "standardised data",
sub = "N(0,1) cdf in pink; maximum difference (KS-Stat)
in green")

> pxx <- pnorm(xx)
> lines(xx, pxx, col = "plum", type = "l")
> mtext(text = expression(hat(F)[n](x)), side = 2,
line = 2.5)

> abline(v = ksstat, col = "seagreen3", cex = 0.3)
> title(paste("KS-stat = ", as.character(round(ksstat,
digits = 3)), sep = ""), adj = 1, cex = 0.3)
```

Once again, the ECDF and N(0,1) do not match due to the large discrepancies, leading to the rejection of the assumption of the standardised data being a sample of the N(0,1) distribution.

Test Simulation

Now, I would like to show how the graphs and test data would look like if the sample data was indeed a random sample of the N(0,1) distribution by running a function in R to simulate the sampling distribution when the N(0,1) null distribution is true using n=120. The sampling distribution was simulated as such:

```
> sampling <- function(N, sample.size){
simksstat <- numeric(N)
for (i in 1:N){
  simx <- rnorm(sample.size)
  simtest <- ks.test(simx, y = "pnorm")
  simksstat[i] <- simtest$statistic
    }
return(simksstat)
 }
> sim <- sampling(10000, n)
```

Using this function, a histogram and KDE of the distribution, as well as comparison of the observed value and estimated 5% critical value can be constructed and shown in one graph. The observed value was obtained by carrying out a KS test on the sampling distribution with the sample sample size as the data given on Blackboard i.e. *rnorm(120)*. And the 5% critical value range, which is computed by the 0.025 and 0.975 quantile of the simulated distribution. The KS test results are as such:

	D	pvalue
Data	0.07659	0.48216

Table 2: KS test for the observed value (D) of the simulated data

For reference, the rest of the code conducted in R were as follows:

```
#compute KS test for observed value
> simx <- rnorm(120)
> test2 <- ks.test(simx, y = "pnorm")

> test2.df <- data.frame(
D = test2$statistic,
pvalue = test2$p.value
)
> rownames(test2.df) = c("Data")
> xtable(test2.df, digits = 5)

#5% critical value range
> crit.val <- quantile(sim, c(0.025, 0.975))
> crit.val
      2.5%      97.5%
0.04223996 0.13472574

#plot histogram and indicate values obtained from above
> hist(sim, freq = F, main = "Histogram of the estimated
sampling distribution", xlab = "simulated sampling
distribution", ylim = c(0, 20))

> lines(density(sim), col = "slateblue4")
> abline(v = crit.val, col = "indianred")
> abline(v = test2$statistic, col = "chocolate1")
> legend("topright", c("observed value = 0.077", "5%
critical value range = (0.042, 0.135)", fill =
c("chocolate1", "indianred")))
```

These results and R codes leads to the graph:

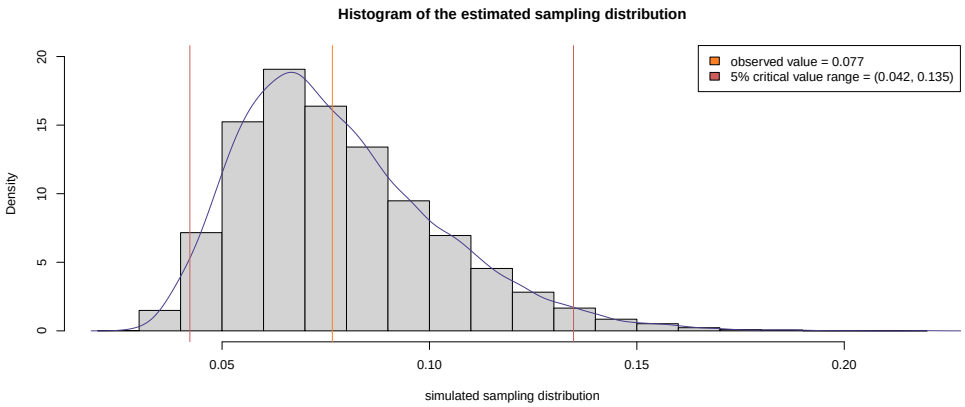

Figure 4: Histogram of the estimated sampling distribution

From the graph, the observed value is seen to lie within the critical value range, meaning that it is in the non-rejection range, which then proved the N(0,1) null distribution being true. Furthermore, the KDE of the sampling distribution was almost identical to the N(0,1) pdf, which further supports the proof that this sampling distribution a sample of a N(0,1) distribution.

To conclude this part of the report, the .txt file given on Blackboard cannot be regarded as a random sample from the N(0,1) distribution, as shown as the discrepancies of its KDE and the N(0,1) pdf in Figure 1, Q-Q plot and theoretical reference in Figure 2, also the empirical CDF and N(0,1) CDF in Figure 3. This can be proven supported by the null hypothesis rejection due to the p-value of the KS test being lesser than 0.05 in Table 1. These graphs and table can be compared with the Test Simulation part of the report, where the results of the simulated distribution being a sample of N(0,1) were shown, in Table 2 and Figure 4.

Part 2: Monte Carlo Sampling

Monte Carlo Integration and Estimation

In this section, I was asked to express the integral I = $\int_0^\infty x^4 e^{-x}\,dx$ as *E(h(x))* and use Monte Carlo integration to approximate I ($\hat{I}$), then propose a Monte Carlo estimate for the Gamma function $\Gamma(5)$. The calculations were done as follows:

$$\begin{aligned} \text{E(h(x))} &= \int_0^\infty x^4 e^{-x}\,dx = \tfrac{1}{b-a}\int_a^b h(x)\,dx \\ &= \int_1^0 (-ln(u))^4 \tfrac{u}{-u}\,du \\ &= \int_0^1 (-ln(u))^4\,du \end{aligned}$$

side working: let u = e^{-x};
$\frac{du}{dx} = -e^{-x}$ = -u;
x = 0, u = 1; x = ∞, u = 0;

Therefore, the integral can be expressed as *E(h(x))*, as required, with h(x) = $(-ln(x))^4$

Now, I could be approximated using Monte Carlo estimation by implementing the equation

$$\hat{I} = \frac{b-a}{N}\sum_{i=1}^{N} h(U_i)$$

and its estimated standard error:

$$\boldsymbol{V}[\hat{I}] = \sqrt{\frac{(b-a)^2}{N} Var[h(U_i)]}$$

both computed in R as such:

```
> N = 1000000
> x = runif(N, 0, 1)
> hx = log(x)^4

> iest = (1-0)*mean(hx)

> var.iest = (1-0)^2/N * var(hx)

> se.iest = sqrt(var.iest)

> iest;se.iest
[1] 23.99455
[1] 0.1945264
```

```
#check the accuracy of estimation
> f <- function(x){
    x^4 * exp(-x)
  }

> I = integrate(f, 0, Inf)
> I
24 with absolute error < 2.2e-05
```

The estimated value of I was ideal due to the small standard error and being shown almost accurate to the actual value of the integral.

The Gamma function $\Gamma(5)$ function equates the integral I as well, as

$$\Gamma(z) = \int_0^\infty e^{-t} t^{z-1}\, dt$$

$$\Longrightarrow$$

$$\Gamma(5) = \int_0^\infty e^{-t} t^{5}\, dt$$

For further prove:

```
> gamma(5)
[1] 24
```

Hence, the Monte Carlo estimate for $\Gamma(5)$ is the same as the calculation computed by R from above.

Monte Carlo Estimate and Rejection Sampling

In the last section, the mixture Normal distribution with pdf

$$f(x) = \frac{3}{5} f_1(x) + \frac{2}{5} f_2(x), -\infty < x < \infty$$

where

$$f_1(x) = \frac{1}{2\sqrt{2}\pi} e^{-\frac{(x-5)^2}{8}}$$

and

$$f_2(x) = \frac{1}{\sqrt{2}\pi} e^{-\frac{x^2}{2}}$$

will be considered and used to obtain a sample of size N with rejection sampling, and estimate the probability P(0<x<5) via Monte Carlo Integration.

Firstly, a Normal proposal N(2.5, 10) with pdf

$$g(x) = \frac{1}{\sqrt{2}0\pi} e^{-\frac{(x-2.5)^2}{20}}$$

was assumed for the rejection sampling scheme, a bound, M with choices of 1, 1.5, 2, 2.5, 3 were given to pick the most suitable bound. This will be conducted by visually comparing f(x) with Mg(x)s in R, with the codes as so:

```
#compute f(x)
> x = seq(from = -4, to = 10, length.out = 400)
> f1x <- exp(-(x-5)^2/8)/(2*sqrt(2*pi))
> f2x <- exp(-x^2/2)/sqrt(2*pi)
> fx <- 3*f1x/5 + 2*f2x/5

> plot(x, fx, type = "l", ylim = c(0, 0.4), main =
"Comparison with f(x) with Mg(x) for M = 1, 1.5,
2, 2.5, 3", ylab = "f(x), Mg(x)")

> gx <- exp(-(x-2.5)^2/20)/sqrt(20*pi)

#adding Mg(x) according to different values of M
> lines(x, 1*gx, col = "coral1")
> lines(x, 1.5*gx, col = "cadetblue2")
> lines(x, 2*gx, col = "burlywood2")
> lines(x, 2.5*gx, col = "slateblue2")
> lines(x, 3*gx, col = "sienna2")
> legend("topright", c("M=1", "M=1.5", "M=2",
"M=2.5", "M=3" ), fill = c("coral1", "cadetblue2",
"burlywood2", "slateblue2", "sienna2") )
```

And the plot for comparison turned out to be as such:

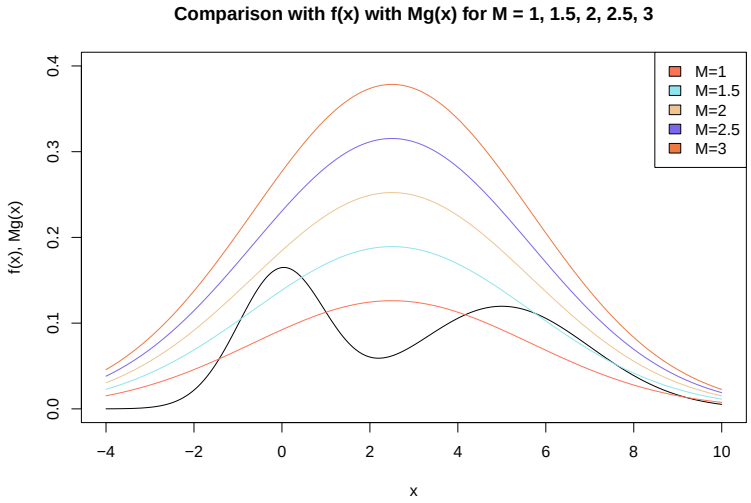

Figure 5: Comparison of f(x) and multiple Mg(x)

It can be concluded that M=2 is the most suitable bound as it is the smallest value that covers f(x)'s density plot i.e. keeps acceptance rate at its highest.

To obtain a sample of size N from the mixture Normal distribution, a simulated sample was written in R to output the simulated data, acceptance probability of the rejection sampling algorithm, and a comparison plot of the density of the simulated data to f(x). It was written and ran as such:

```
> rej = function(M, N){
    res = numeric(N)
    niter = 0
    count = 1
    while (niter<N){
      count = count + 1
      x = rnorm(1, 2.5, sqrt(10))
      y = runif(1, 0, M * dnorm(x, 2.5, sqrt(10)))
      f1x <- exp(-(x-5)^2/8)/(2*sqrt(2*pi))
      f2x <- exp(-x^2/2)/sqrt(2*pi)
      fx <- 3*f1x/5 + 2*f2x/5
      if(y < fx){
        niter = niter + 1
        res[niter] = x
      }
    }
    z = seq(from = -4, to = 10, length.out = 400)
    f1z <- exp(-(z-5)^2/8)/(2*sqrt(2*pi))
    f2z <- exp(-z^2/2)/sqrt(2*pi)
    fz <- 3*f1z/5 + 2*f2z/5
    plot(z, fz, type = "l", main = "Comparison of simulated
    data's density to f(x)", xlab = "x", ylab = "f(x)",
    sub = "density in pink (M=2)")
    lines(density(res), col ="deeppink")
    out = list(res, N/count)
  }

> N = 100000
> res = rej(2, N)
> res[[2]] #acceptance rate
[1] 0.4990941
```

The acceptance rate appeared to be 0.4990941 at N = 100000, which is very ideal as the theoretical acceptance rate is to be $\frac{1}{M} = 0.5$ in this case. This is also reflected from the density graph that is constructed in the function written:

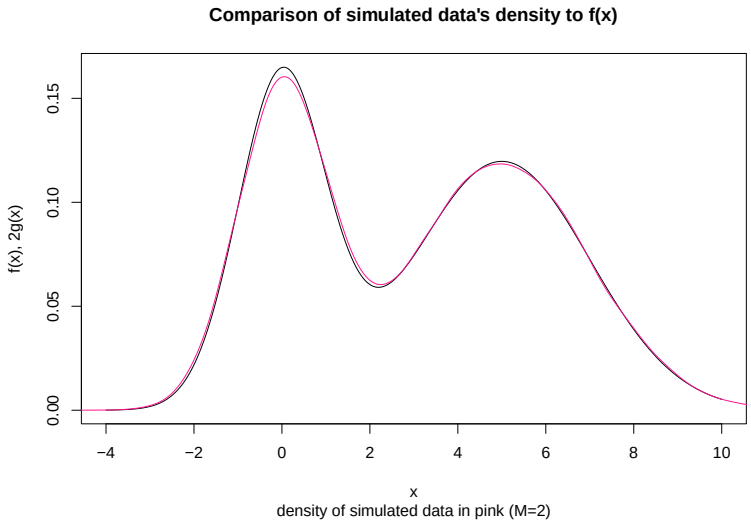

Figure 6: Comparison of density of simulated data and f(x)

This figure indicates that the proposal, along with M=2 are good at approximating a sample of f(x), judging by the similarities, almost identical density plots between f(x) and the simulated data, which also explains the high acceptance rate.

Lastly, the Monte Carlo etimate of the probability P(0<x<5) along with its 95% Confidence Interval is to be computed in R using the sample simulated previously. The results are as such:

```
> mu <- mean(res[[1]])
> sd <- sd(res[[1]])

> mc.prob <- function(a, b, level, N){
    x <- runif(N, a, b)
    fx <- dnorm(x, mu, sd)
    iest = (b-a) * mean(fx)
    est.var = ((b-a)^2)*var(fx)/N
    est.se = sqrt(est.var)
    ci = 0
    ci[1] = iest - qnorm((1+level)/2)*est.se
    ci[2] = iest + qnorm((1+level)/2)*est.se
    results = list(Probability = iest, CI = ci,
```

```
    Conf_Level = level)
    return(results)
  }

> mc.prob(0, 5, 0.95, N)
$Probability
[1] 0.5949392

$CI
[1] 0.5944797 0.5953987

$Conf_Level
[1] 0.95

#actual probability
> pnorm(5, mu, sd)- pnorm(0, mu, sd)
[1] 0.594527
```

The value of the probability turned out to be 0.5949392, which lies within the 95% Confidence Interval of (0.5944797, 0.5953987), and is also very close to the actual probability, 0.594527. Hence, it can be concluded that the Monte Carlo Integration, Estimate and Rejection Sampling were computed successfully.