MATH20811 - Practical Statistics

Coursework 3 Submission

This report aims to analyse data that was collected by the Swedish Committee on Analysis of Risk Premium in Motor Insurance. The two variables that we will be studying are `claims`, as the independent variable and `payment`, as the dependent variable. With $N = 63$ observations, these variables are assumed to be modeled by a simple linear regression,

$$y_i = \alpha + \beta x_i + \epsilon_i \quad i = 1, ..., N$$

where

- y_i denotes the response variable, `payment`
- x_i denotes the predictor, `claims`
- ϵ_i are random errors assumed to follow $N(0, \sigma^2)$, and σ^2 is unknown
- α and β are unknown parameter values

We specifically aim to look at the suitability of the simple linear regression model for the dataset as well as the assumed distribution of the random errors. Therefore this report is split into two sections.

1 Simple Linear Regression

1. ```
   > claims = swdins$claims
   > paym = swdins$payment
   > plot(claims, paym, xlab = "Claims", ylab = "Payment")
   ```

   With the listed R codes, we take a look at the aforementioned model for `payment` against `claims`, by creating a scatterplot in Figure 1. Based on the scatter, we first notice that the data points are very concentrated in the bottom left of the graph, with only a couple values located in the top right corner. It is possible that those very few points are outliers in the dataset. However, the longer we look at it, we can tell that the data points also seem to be concentrated along some straight line graph with some values being deviated by random errors. Therefore there could be a possibility that the values in the far right are not outliers, and that some values deviating from the straight line are. This is the exact description of the simple linear regression model. Hence, there *is* apparent suitability of using the model for the dataset.

2. In this part of the report we will be writing a function in R to obtain the parameter estimates, $\hat{\alpha}$ and $\hat{\beta}$, of the model using the least squares method. To complement the analysis, we will also be including the fitted values, $\hat{y}_i$, estimated residuals, $\hat{\epsilon}_i$ as well as its degrees of freedom. Before the function is presented, some formulas used to calculate these values are listed below.
   ```

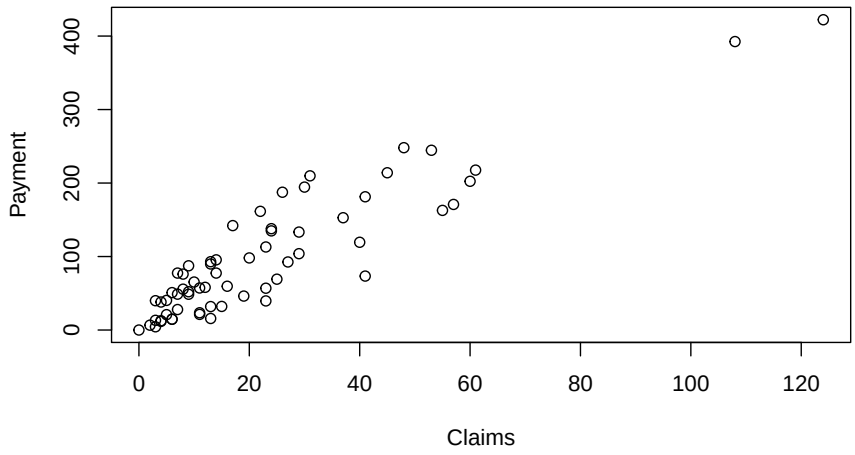

Figure 1: Analysis of Risk Premium in Motor Insurance

$$\hat{\beta} = \frac{cov(x,y)}{cov(x,x)} \tag{1}$$

$$\hat{\alpha} = \frac{\sum_{i=1}^{n} y_i}{N} - \hat{\beta}\frac{\sum_{i=1}^{n} x_i}{N} \tag{2}$$

$$\hat{y}_i = \hat{\alpha} + \hat{\beta} x_i \tag{3}$$

$$\hat{\epsilon}_i = y_i - \hat{y}_i \tag{4}$$

With all these formulas, we write a function to obtain the aforementioned vectors and values with user input x and y values, in this case the `claims` and `payment` respectively;

```
> slinreg = function(x,y){
+   N = length(x)
+   par.est = numeric(2)
+   y.hat = numeric(N)
+   res = numeric(N)
+   for (i in 1:N) {
+     b.hat = (sum(x*y)-sum(x)*sum(y)/N)/(sum(x^2)-(sum(x)^2/N))
+     a.hat = sum(y)/N - b.hat*(sum(x)/N)
+     y.hat[i] = a.hat + b.hat*x[i]
+     res[i] = y[i] - y.hat[i]
+   }
+   par.est = c(a.hat, b.hat)
+   resdf =N-2
+   output = list(par.est, y.hat, res, resdf)
+   names(output) = c("parameter.estimates", "fitted.values",
    "residuals","residual.degrees.of.freedom")
```

```
+   return(output)
+ }

> slinreg(claims, paym)
$parameter.estimates
[1] 19.994486  3.413824

$fitted.values
 [1] 388.68743  84.85713  64.37419 443.30861 156.54743 214.58243  98.51243
    67.78802 173.61655
[10]  54.13272  37.06360 183.85802  57.54654  98.51243  43.89125  26.82213
    101.92625  40.47743
[19]  30.23596  98.51243  40.47743  50.71890  50.71890  30.23596 118.9953
    43.89125  33.64978
[28]  88.27096  43.89125  33.64978  19.99449 105.34007  40.47743  37.06360
    95.09860  57.54654
[37] 228.23772  60.96037  33.64978  74.61566  64.37419 224.82390 159.96125
    146.30596 207.75478
[46] 159.96125  57.54654 112.16772  47.30507  30.23596  78.02949  64.37419
    64.37419  71.20184
[55]  47.30507 118.99537 122.40919 101.92625  50.71890 125.82302  67.78802
    200.92713 108.75390

$residuals
 [1]   3.8125698 -38.6571334 -48.6741920 -21.1086072 -37.1474282
    -43.6824287 -41.6124276
 [8]   9.7119844  40.3834540  11.1672786 -16.1636036  64.2419834
    -34.0465449 -58.9124276
[15]   4.9087493 -20.2221329  32.9737488  10.4225729 -25.8359564
    14.4875724 -25.6774271
[22]  -2.0188978   1.3811022 -17.0359564 -15.0953690  33.6087493
    -21.8497800   9.8290430
[29] -15.9912507   4.4502200 -19.9944858 -36.1400748 -25.8774271
    3.2363964  66.4013959
[36]  -0.3465449 -10.6377229  -2.8603685 -21.0497800 -15.0156627
    25.5258080 -22.4238994
[43]  21.3387483   6.4940425 -44.9547816 -86.5612517 -36.2465449
    -19.5677219  28.7949258
[50]   9.6640436  64.0705137  28.6258080 -32.4741920 -39.1018392
    8.2949258  14.3046310
[57]  72.0908074  35.9737488  36.6811022  83.9769839  27.7119844
    43.6728656  78.7461017

$residual.degrees.of.freedom
[1] 61
```

These values can be confirmed by extracting information from the built in R function;

```
> lmswdins = lm(paym~claims)
> lmswdins$coefficients
> lmswdins$fitted.values
> lmswdins$residuals
```

To then ease the further calculations and analysis, we store the values obtained from the function in the objects below.

```
a.hat = slinreg(claims, paym)$parameter.estimates[1]
b.hat = slinreg(claims, paym)$parameter.estimates[2]
fit.val = slinreg(claims, paym)$fitted.values
est.res = slinreg(claims, paym)$residuals
```

3. i. To further investigate the suitability of the regression model, it is useful to draw the straight line mentioned in part 1 using the parameter estimates obtained from the function above, producing Figure 2.

```
> slinreg(claims, paym)$parameter.estimates
[1] 19.994486  3.413824
> abline(a = a.hat, b = b.hat)
```

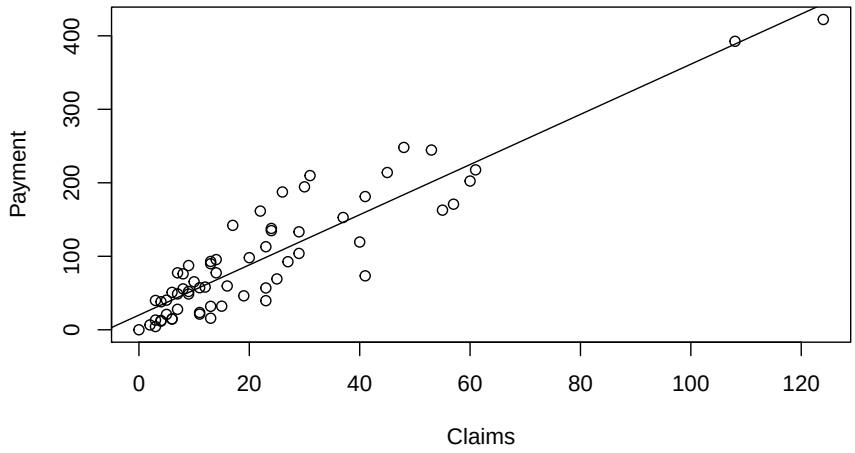

Figure 2: Analysis of Risk Premium in Motor Insurance

Dissecting the scatterplot by intervals, we see that in the first 20 `claims`, the data points fall very close either on or around the straight line, implying that the deviations are quite small. These deviations seem to increase from 21 to 60 numbers of `claims`, and so some of those points could be outliers in the data. Finally, the two points on the top right corner, around 101 to 125 `claims` are plotted very close to the drawn straight line. This figure has given us a clearer visual on the regression model, confirming the comments made in part 1.

ii. It was mentioned briefly that the estimated residuals follow a Normal distribution with mean 0 and variance σ^2. We will now calculate the estimated value of σ^2, $\hat{\sigma}^2$ denoted by `sigma.hat.sq` in R by using the error sum of squares;

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} \hat{\epsilon}_i}{N-2}$$

```
> sse = sum((paym - fit.val)^2)
> sse
[1] 78796.74
> sigma.hat.sq = sse/(N-2); sigma.hat.sq
[1] 1291.75
```

iii. Next, we conduct a hypothesis test, the F-test, at 5% significance level on whether β is zero or otherwise to see if the simple linear regression is a suitable model for the dataset.

$$H_0 : E[Y|x] = \alpha \text{ vs } H_A : E[Y|x] = \alpha + \beta x$$

The test statistic is calculated with the listed formula and is assumed to follow the F distribution with the following parameters when H_0 is true.

$$F = \frac{\sum_{i=1}^{n} \hat{y}_i - \bar{y}}{\hat{\sigma}^2} \sim F(1, N-2)$$

```
> ssr=sum((fit.val - mean(paym))^2) # regression SS
> ssr
[1] 394021.5
>
> F=ssr/(sse/(N-2))
> F
[1] 305.0293
```

We can then find the p-value to the test statistic and compare it to the significance value,

```
> 1-pf(F, df1=1, df2=N-2)
[1] 0
```

As we can see, the p-value is very small and is less than 0.05. Therefore, there is significant evidence to reject the H_0, in favour of the H_A. The true β value is *not* equals to zero, implying that the simple linear regression model *is* suitable for this dataset.

iv. Finally for the regression model part of this report, we can construct a 95% confidence interval for the mean response, $E[Y|x]$ when $x = 80$. The formula is as follows,

$$[\hat{\alpha} + \hat{\beta}x \pm t_{1-\frac{\alpha}{2},N-2}\hat{\sigma}^2 \sqrt{\frac{1}{N} + \frac{(x-\bar{x}^2)}{cov(x,x)}}]$$

```
> surd = (1/N) + ((80 - mean(claims))^2/(sum(claims^2)-(sum(claims)^2/N)))
    ; surd
[1] 0.1122919
> t = qt(0.975, N-2); t
[1] 1.999624
>
> ub = a.hat + 80*b.hat + t*sqrt(sigma.hat.sq)*sqrt(surd)
> lb = a.hat + 80*b.hat - t*sqrt(sigma.hat.sq)*sqrt(surd)
>
> lb; ub
[1] 269.0173
[1] 317.1834
```

From the graph in Figure 2, we can see that the y-value for $x = 80$ does fall in the range of the confidence interval. This also falls in line with the hypothesis test we did previously, that the true β value is not 0.

2 Standardised Residuals

For the second part of this report, we will be focusing on the standardised residuals of the data, obtained by the `rstandard` function in R.

4. First, we must inspect the assumption made on the distribution of the standardised residuals by using diagnostic plots. Some assumptions made that will be compared to the key characteristics of the plot are as follows;

   – Linearity of the data
   – Independent errors
   – Normally distributed errors
   – Homoscedasticity (equal variances of the errors)

```
> std.res = rstandard(lmswdins)
> plot(fit.val, std.res, xlab = "Fitted values",
    ylab = "Standardised Residuals")
> abline(h = 0)
> abline(h = 2, col = "red")
> abline(h = -2, col = "blue")
```

There are three lines in Figure 3 that are used as reference lines, the red, blue, and black. First off, we see that there are no particular patterns that the data points form which implies that the regression model fits well with the data in terms of linearity and independently distributed errors. The number of observations is quite little which makes it hard to tell if the variance is constant throughout, however, since they do not form any pattern, it is safe to assume that the variability is constant. Most of the standardised residuals fall within -2 and

2, with many points being close to zero. This tells us that they are approximately Normally distributed. The very few points that lie outside of that interval can be regarded as outliers since the deviation is quite large.

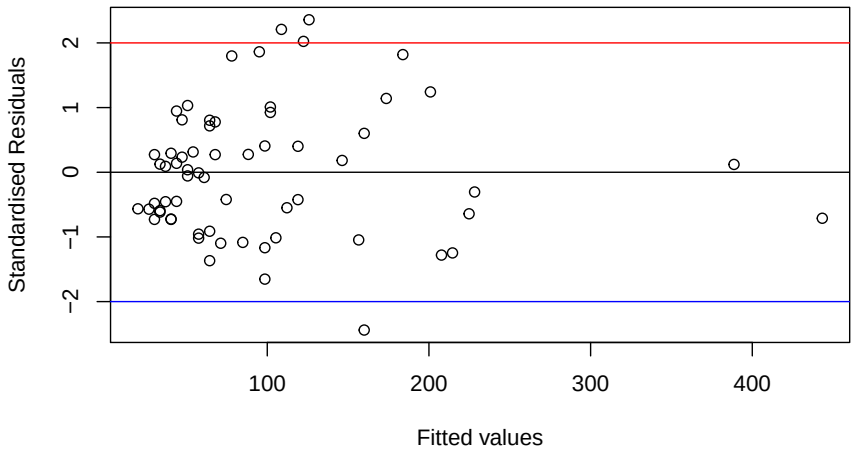

Figure 3: Fundamental Diagnostic Plot; Standardised Residuals against Fitted Values

Another useful diagnostic plot would be plotting the standardised residuals against the covariates, `claims`. We see in Figure 4, plotted using the following R codes, that the points clearly look like random scatter in the first 60 claims. At the right end of the graph it is quite hard to tell since there are only two points evident. They also seem to be symmetrically distributed about the black reference line added when the standardised residual is equal to zero, its supposed mean value. It is also important to note that most of the residuals fall between -2 and 2, implying it is Normal.

```
> plot(claims, std.res, xlab = "Claims", ylab = "Standardised Residuals")
> abline(h = 0)
```

5. To further investigate the standardised residuals' Normality, we will construct a Normal quantile-quantile plot of the standardised residuals. Starting off with sorting the data, we then compute the theoretical quantile estimate using the Type 9 estimator where the probabilities, p_k are defined as

$$p_k = \frac{k - \frac{3}{8}}{N + \frac{1}{4}} \quad k = 1, ..., N$$

These probabilities are then computed in the `qnorm` function, leaving the mean and variance to its default value 0 and 1 respectively, to obtain the theoretical quantile estimate values. Finally we plot the sample order statistic against the theoretical values and check for linearity.

```
> std.res.ord <- sort(std.res)
```

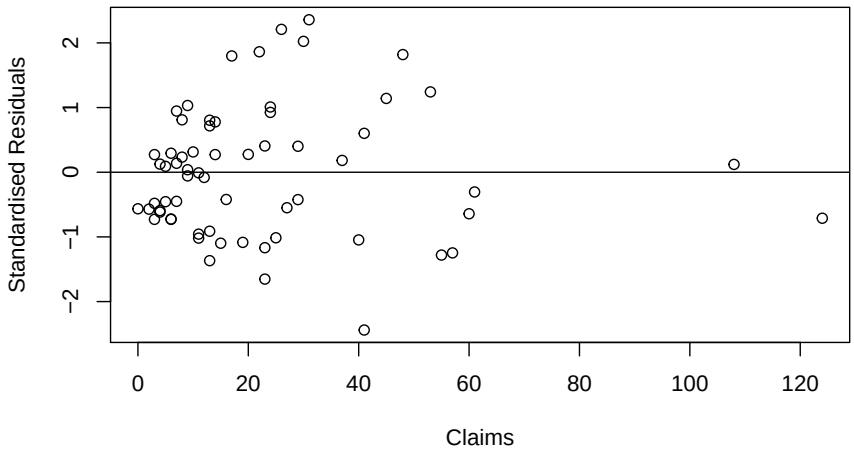

Figure 4: Diagnostic Plot; Standardised Residuals against covariates

```
> n <- length(std.res); n
[1] 63
> k = seq(1:n)
> pk <- (k - 3/8)/(n + 1/4)
> q_th <- qnorm(pk)
>
> plot(q_th, std.res.ord, pch = 1, cex = 0.5, xlab = "Theoretical Quantile
    Estimate", ylab = "Sample Order Statistic")
> abline(a = mean(std.res), b = sd(std.res), col = "red")
```

A red reference straight line graph with the mean of standardised residuals as the intercept and their standard deviation as the gradient, is added to accommodate the analysis. Looking at the quantile-quantile plot in Figure 5, the data points are fairly linear as most of them fall very closely to the reference line. There are only a few points in the intervals -2 to -1 and 1 to 2, that deviate a bit further than the rest of the data points. Therefore Normality is a tenable assumption for the standardised residuals.

6. Another way we can check the standardised residuals' Normality is by conducting a Kolmogorov-Smirnov (KS) test with the following null hypothesis

$$H_0 : \text{The standardised residuals is a random sample from } N(0,1)$$

$$\text{vs } H_A : H_0 \text{ is not true}$$

The alternate hypothesis is rather vague, hence there are some possibilities that it could imply, the data is a Normal random sample with different parameters; or the data is a random sample not from a Normal distribution; or the data might not even be a random sample entirely. To calculate the KS test statistic, D_n, we require two sets of directional deviations, D_n^+ and D_n^-;

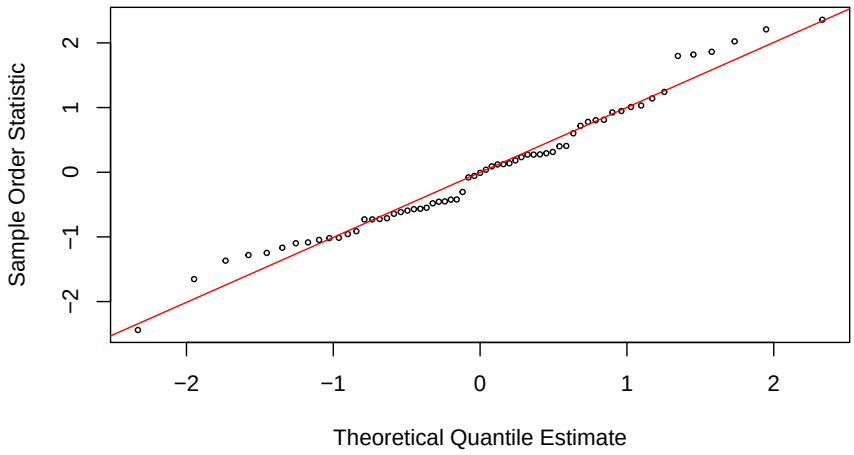

Figure 5: Normal quantile-quantile plot

$$D_n^+ = \sup_x (S_n(x) - F_0(x))$$
$$D_n^- = \sup_x (F_0(x) - S_n(x))$$

where $S_n(x)$ is the empirical cumulative distribution function (cdf) also known as $\hat{F}_n(x)$, and $F_0(x)$ is the distribution specified in H_0, in this case, the standard normal distribution. With them, we can then find the formula to obtain D_n, we have,

$$D_n = \sup_x |S_n(x) - F_0(x)| = \max(D_n^+, D_n^-)$$

Therefore we get these formulas below,

$$D_n^+ = \max\{\max_{1\le i\le n}(\frac{i}{n} - F_0(X_{(i)}), 0\}$$
$$D_n^- = \max\{\max_{1\le i\le n}(F_0(X_{(i)} - \frac{i-1}{n}), 0\}$$

All these formulas are then applied in R as follows;

```
> std.res.ecdf<-(1:N)/N
> y <- pnorm(std.res.ord)
>
> diff1 = std.res.ecdf-y
> md1 = max(diff1)
> dn.plus = max(md1,0)
> dn.plus
```

```
[1] 0.1077294
> std.res.KS1 = std.res.ord[dn.plus==diff1]
> std.res.KS1
        40
-0.4214452
>
> diff2 = y - std.res.ecdf
> md2 = max(diff2)
> md2 = md2+(1/N)
> dn.minus = max(md2,0)
> dn.minus
[1] 0.05914349
> std.res.KS2 = std.res.ord[dn.minus==diff2+(1/n)]
> std.res.KS2
      51
1.797923
>
> KSstat=max(dn.minus, dn.plus)
> KSstat
[1] 0.1077294
>
> if(dn.minus < dn.plus) std.res.KS = std.res.KS1
> if(dn.minus > dn.plus) std.res.KS = std.res.KS2
> std.res.KS
        40
-0.4214452
```

And finally we obtain $D_n = 0.1077294$ and the standardised residual value where this occurs is -0.4214452. We will come to a conclusion to this test later on in part 8 where we look at the test statistic's sampling distribution.

7. Next, it is useful to take a look at the empirical cdf of the standardised residuals by using the ordered statistics of the standardised residuals. Then the step function $\hat{F}_n(x)$ is given by

$$\hat{F}_n(x) = \begin{cases} 0 & x < x_{(1)} \\ \frac{i}{n} & x_{(1)} \leq x \leq x_{(62)} \\ 1 & x \geq x_{(63)} \end{cases}$$

To make a meaningful analysis we superimpose the plot with the Normal cdf as well as mark where the KS statistic occurs. This is done by the following R code lines;

```
> x2 = seq(from=-4, to=4, length.out=600)
> px2 = pnorm(x2)
>
> plot(std.res.ord, (1:N)/N, type="s", xlim=c(-4, 4), xlab="x", ylab="")
> segments(x0 = std.res.ord[1], y0 = 0, x1 = std.res.ord[1], y1 = 1/N,
    col = "black")
```

```
> segments(x0 = std.res.ord[n], y0 = 1, x1 = 4, y1 = 1, col="black")
>
> lines(x2, px2, col="blue")
> legend('topleft', c('empirical cumulative distribution function',
    'standard normal cdf'), lty=1 ,col=c('black', 'blue'),
    bty='n',cex= 0.85)
> mtext(text = expression(hat(F)[n](x)), side = 2, line = 2.5)
```

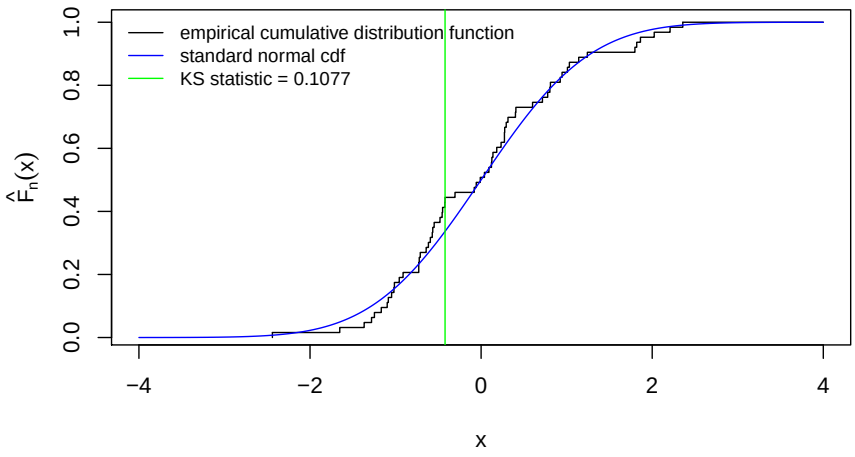

Figure 6: Empirical cdf of the standardised residuals superimposed by the N (0, 1) cdf

Based on Figure 6, we note that the variability of the empirical cdf compared to the Normal cdf is fairly small, as they seem to be pretty close to each other. However, there are a few instances where $\hat{F}_n(x)$ looks to be quite far off from the Normal cdf. This is where we are able to locate where the KS statistic occurs, at the standardised residual valued -0.4214. Here, $\hat{F}_n(x)$ is larger than the distribution specified under H_0 in part 6 by 0.1077 stated in the legend.

8. Finally, this report ends with a conclusion for part 6. In order to come to a conclusion on whether or not to accept H_0 we will simulate $B = 5000$ values of the KS test statistic conducted on vector `ysim` of length $N = 63$, consisting of random samples from a standard Normal distribution. Then we construct a histogram and its kernel density estimate for a visual representation of the sampling distribution of the test statistic. Using the R code lines below, we get the plot in Figure 7.

```
> B = 5000
> ysim = numeric(N)
> test.sim = 0
> for (i in 1:B){
+   for (k in 1:N) {
+     ysim[k] <- rnorm(1)
+   }
```

```
+   test.sim[i] <- ks.test(ysim, y = "pnorm",
    alternative = c("two.sided"))$statistic
+ }
>
> hist(test.sim, freq = FALSE, main = "", xlab = "x")
> lines(density(test.sim), col = "red")
> legend('topright', c('simulated test statistic'), lty = 1,
    col = c('red'), bty = 'n',cex = 0.85)
```

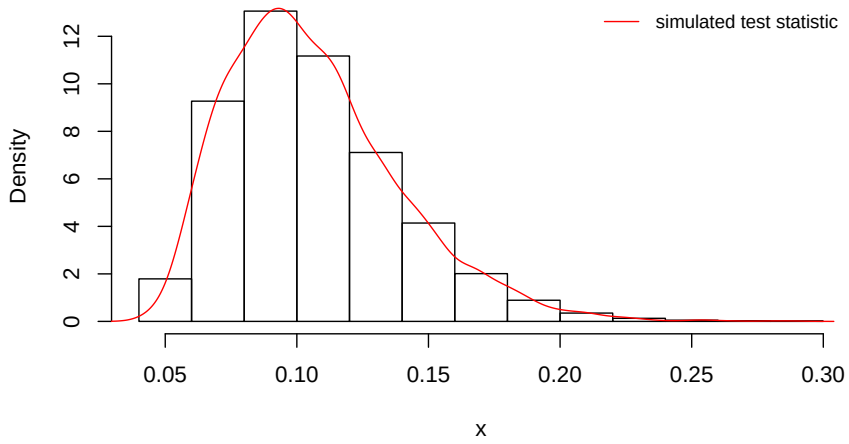

Figure 7: Histogram and Kernel density estimate of simulated KS test statistic

From the density estimate, we are able to obtain a critical value to be compared with the KS test statistic we obtained earlier in part 6. The `quantile` function in R with the default Type 7 gives us

```
> quantile(test.sim, probs = 0.95)
     95%
0.168571
```

Since the critical value, 0.1686 is greater than the KS statistic, 0.1077, there is not enough evidence to be rejecting H_0. Therefore, we conclude that the standardised residuals *is* a random sample from the standard Normal distribution.