Problem 1(a)

我們要生成$\int_0^1 \frac{e^{-x}}{1+x^2}dx$,並且令其為 I, $\frac{e^{-x}}{1+x^2}$ 為 h(x)，因為這裡的 X 來自 uniform distribution, 所以演算法為

1/(1-0)* I= 1/(1-0)* $\int_0^1 \frac{e^{-x}}{1+x^2}dx$

1/(1-0)* I= $\int_0^1 \frac{e^{-x}}{1+x^2} \times 1/(1-0)dx$

I= $(1-0) * \int_0^1 \frac{e^{-x}}{1+x^2} \times 1/(1-0)dx$

I= $(1-0) * E(h(x))$

E(x)=integrate x*f(x)

E(h(x))=integrate h(x)f(x)

步驟:

1. 生成 x1, x2,…
2. 計算 h(x1), h(x2), …
3. 計算 E(h(x1)), E(h(x2)), … 為(1/n)*sum(h(xi))
4. 計算 I= $(1-0) * E(h(x))$

Problem 1(b)

我們要生成$\int_0^1 \frac{e^{-x}}{1+x^2}dx$,並且令其為 I, $\frac{e^{-x}}{1+x^2}$ 為 h(x)，因為這裡的 X 來自 exponential distribution, 所以演算法為

步驟:

1. 生成從 exponential 分布的隨機變數 x1, x2,…
2. 計算 weight=h(x)/g(x)
3. 得到期望值為 mean(weight*x)/mean(w)

Problem 1(c)

從結果上來看，1(a)的結果比 1(b)的要好，這可能是因為在 1(b)的 weight 的期望值應該要接近 1, 但是實際的值卻只有 0.64 左右。

```
> mean(w)
[1] 0.6400261
```

| 題目 | 結果 |
|---|---|
| 1(a):實際積分 | 0.5247971 with absolute error < 5.8e-15 |
| 1(a):importance function | 0.5250281 |
| 1(b):importance function | 0.5351034 |

Problem 2(a)

題目要求要 simulate 200 次迴歸分析的 random sample，而該回歸式為:

$Y_i = \beta_0 + \beta_1 X_i + \epsilon_i$, $\epsilon_i \sim N(0, \delta_\epsilon^2)$, $X_i \sim N(0, \delta_X^2)$, $n = 500$, $\beta_0 = 1$, $\beta_1 = 2$, $\delta_\epsilon^2 = 1$, $\delta_X^2 = 2$

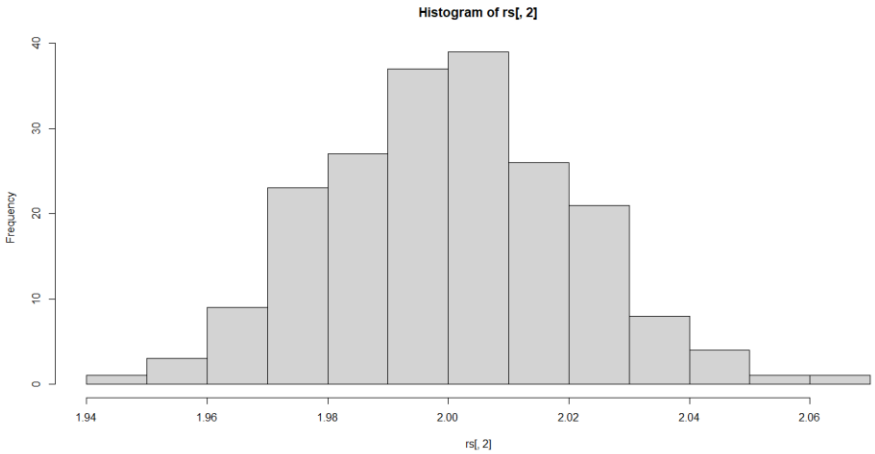
演算法步驟為:

1. 產生 500 組的$\epsilon_i \sim N(0,1)$, 500 組的 $X_i \sim N(0,2)$
2. 設定 Y=1+2*X+e
3. 並將每一次產生的 500 組 X 值跟 Y 值分別存入不同的向量中
4. 利用 lm 的函式得到所需要的 beta0 和 beta1，並將其存入向量中
5. 利用 residual 的函式得到每一次 random sample 的 residual，並將其存入向量中
6. 利用 for loop 重複以上步驟並產生兩百組 beta1, beta0 的 random sample

最後得到 beta1 的平均值為:

```
> mean(rs[,2])
[1] 2.001404
```

而畫出來的 beta1 直方圖為:




Problem 2(b)

從圖上可以看出由上述方法所產生的 beta1 的分布非常像常態分佈，並且中心在 2 附近，而這也符合理論的期望，因為理論上來說這些隨機產生的變數經過回歸分析之後所得到的係數應該要接近原本的係數(在這裡 beta1 的平均應該要非常接近 2，因為 X 的係數是 2)，並且 beta1 的分布應該要像是中心在 2 的常態分佈。

Problem 2(c)

這一題要求要用兩種方式計算 beta1 的變異數

首先，第一種方法是採用 asymptotic beta1 的變異數，

因此演算法為:

1. 利用 1(a)方法得到 200 個 beta1 的樣本
2. 並將 200 個 beta1 的樣本取變異數

最終得到的結果為:

```
> var(rs[,2])
[1] 0.0005290695
```

第二，另一種方法為採用 empirical variance，

演算法為:

1. 根據定義，理論上期望值的計算為

$\beta_1 \sim N\left(\beta_1, \frac{\delta^2}{S_{XX}}\right)$, 在這裡$\delta^2 = 1, S_{XX} = \sum(\mathrm{X_i}\text{-}0)^2$

最終得到的結果為:

```
> mean(empvar)
[1] 0.0005051891
```

Problem 2(d)

題目要求要利用兩種方式做 bootstrap

第一種方式為 nonparametric sampling，演算法為:

1. 產生一個 NULL 的 list boot
2. 這裡需要兩層 for loop:
3. 內層的 for loop 為從前面 sample 的 X 值和 Y 值中任意取 500 個值，並且取後放回
4. 接著利用 lm 的函式產生 beta1 值
5. 利用內層的迴圈跑出 200 個 beta1 值，並將其存進 boot 中
6. 利用外層的迴圈再將內層迴圈的步驟進行 200 次，並將其存進 boot 中
7. 最終我們會得到 200*200=的 40000 個 beta1
8. 最後將其取 var 並得到結果

```
> var(boot)
[1] 0.0005264373
```

第一種方式為 residual sampling，演算法為:

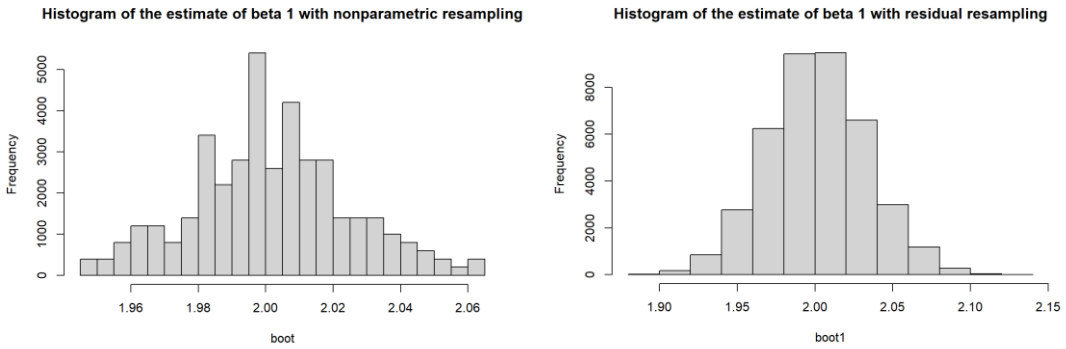
1. 產生一個 NULL 的 list boot1
2. 這裡需要兩層 for loop:
3. 內層的 for loop 為從前面 sample 的 X 值和 Y 值中任意取 500 個值，並且

取後放回

4. 接著利用(a)題的 beta 值
5. 利用(a) 中的 residual
6. 並利用公式 Y* =residual+ beta1*X+beta0 得出新的 Y*
7. 利用 lm 的函式跑出 Y*和 X 的迴歸分析並得到新的 beta1 值
8. 利用內層的迴圈跑出 200 個 beta1 值，並將其存進 boot1 中
9. 利用外層的迴圈再將內層迴圈的步驟進行 200 次，並將其存進 boot1 中
10. 最終我們會得到 200*200=的 40000 個 beta1
11. 最後將其取 var 並得到結果

```
> var(boot1)
[1] 0.001017509
```

比較:




1. 比較上述兩種方法，在這個回歸模型中，residual sampling 比起 nonparametric sampling 獲得了較大的變異數

| residual sampling | nonparametric sampling |
|---|---|
| 0.001017509 | 0.0005264373 |

2. 根據上圖，可以看出 residual sampling 的圖更像常態分佈
3. 若是和前面 beta 1 的直方圖做比較的話，會發現 nonparametric sampling 的直方圖和之前的結果更加相近

Problem 2(e)

觀察與討論:

1. 受到擾動之後所得到的 beta1 的變異數會比受到擾動之前大。
2. 由於 Asymptotic var.的樣本數只有 200 個，所以變異數算起來會比 bootstrap(樣本數為 40000)的還要大一些。
3. 若是當 beta1 要當 M estimator 的時候，變異數必須要等於 0，而在這裡可

以發現平均出來的結果都很接近 0，不論是擾動前或是擾動後。

4. 從受到擾動的結果觀察，Boot var.的值比其他兩個小，這代表他在變異數受到擾動的情況相較另兩種方法比較不敏感。
5. 不論是擾動前或是擾動後，利用 nonparametric sampling 之後所得到的變異數值都比較小，而使用 residual sampling 會產生最大的變異數。
6. 在受到擾動之後，boot(non-parametric sampling)和 boot1(residual sampling)的分布看起來十分相似，但是 boot 擾動之後的變異數仍然小於 boot1，並且和理論值最為接近，因此在這個回歸模型中可以利用無母數的方法估計 coefficients.

| | 受到擾動前 | 受到擾動後 |
|---|---|---|
| 理論值 | 0.0005051891 | 0.0005051891 |
| Asymptotic var. | 0.0005290695 | 0.0006099797 |
| Boot var. | 0.0005264373 | 0.0005802581 |
| Boot1 var. | 0.001017509 | 0.001121274 |