

Statistical Computing

and

Simulation HW3

授課教授: 余清祥教授

學生: 統計碩一 106354003 林健宏

統計碩一 106354012 曹立諭

**目錄**

[# Question 01. 2](#_Toc513161773)

[(a) Monte-Carlo Integration 2](#_Toc513161774)

[(b) Hit or miss 2](#_Toc513161775)

[(c) Antithetic Variate 3](#_Toc513161776)

[(d) Importance Sampling 3](#_Toc513161777)

[(e) Control variate 3](#_Toc513161778)

[(f) Stratified Sampling 3](#_Toc513161779)

[(g) 結論 5](#_Toc513161780)

[# Question 02. 6](#_Toc513161781)

[# Question 03. 8](#_Toc513161782)

[# Question 04. 9](#_Toc513161783)

[# Question 05. 11](#_Toc513161784)

[(a) Kernel smooth 12](#_Toc513161785)

[(b) Lowess 12](#_Toc513161786)

[(c) Spline smooth 13](#_Toc513161787)

[(d) Running mean smooth 14](#_Toc513161788)

[(e)結論 14](#_Toc513161789)

[附錄 (R code) 15](#_Toc513161790)

[1. 15](#_Toc513161791)

[2. 18](#_Toc513161792)

[3. 21](#_Toc513161793)

[4. 22](#_Toc513161794)

[5. 25](#_Toc513161795)



Experiment with as many variance reduction techniques as you can think of to apply the problem of evaluating .

|  |  |
| --- | --- |
| (圖一) | (圖二) |
|  |  |

以Monte-Carlo Integration為基準，並使用其他變異數縮減方法去降低估計的誤差，使用方法有以下幾種：Hit or miss、Antithetic Variate、Importance Sampling、Control variate以及Stratified Sampling。每一個方法皆取1000個隨機樣本生成一個，再生成1000樣本 ，並運用其平均去估計真實的及計算樣本變異數。

**使用方法介紹**

1. Monte-Carlo Integration：

P(X1 )可以經由Monte-Carlo Integration模擬，x從uniform(0,1)抽取1000筆資料，並將每筆資料帶入 得到每筆資料的對應值，再取這1000筆資料對應值的平均值為P(X 1 )的估計值。因此P( X > 1)的估計值為0.5 - P(X 1 )的估計值。我們重複這動作1000次取得1000個P( X > 1)的估計值，計算這1000個P( X > 1)的估計值的平均與變異數為0.24997910與0.00000272，此為Monte-Carlo Integration。

1. Hit or miss

從X ~ Cauchy中隨機抽取1000筆資料，計算幾筆資料超過1的比例即為 P( X > 1)的估計值，重複這個動作直到取得1000筆 P( X > 1)的估計值，最後取這1000筆估計值的平均與變異數，分別為0.2498500與0.000193，此為Hit or miss 估計方法。

1. Antithetic Variate

做法類似Monte-Carlo Integration，x 從Uniform(0,1)抽取500筆，再將這500筆資料用1 – x的方式生成另外500筆資料，形成1000筆資料。後續步驟雷同Monte-Carlo Integration的步驟，得到P( X > 1)的估計值的平均與變異數為0.24999466與0.00000004，此為Antithetic Variate縮小變異數方法。

1. Importance Sampling

先選找另一個函數g(x)（此題使用的為），令Y = ，再利用Monte Carlo方法去估計E(Y)。當Y越接近常數，變異數縮減效果越好。因此我們透過此方法，形成1000筆P( X > 1)的估計值，並取這1000筆P( X > 1)的估計值的平均數與變異數為0.24997706與0.00000244，

此為Importance Sampling縮小變異數方法。

1. Control variate

先找一個與本題函數類似的函數（此題使用g(x) = ），這個函數要能求取期望值，利用  + a( g(x) – E(g(x)))，a = x從uniform(0,1)隨機抽取1000筆，將這1000筆x代入 並取平均即為P(X1 )的估計值，0.5 - 為P( X > 1)的估計值。總共取1000筆P( X > 1)的估計值，並取並取這1000筆P( X > 1)的估計值的平均數與變異數為0.24982097與0.00000983，此為Control variate縮小變異數方法。

1. Stratified Sampling

做法類似於Monte-Carlo Integration，我們將x的範圍(0,1)切成5等分，即(0, 0.2), (0.2, 0.4), …, (0.8,1)，每做一次P(X 1 )的估計值，要抽取1000個x樣本，這邊將x範圍等分成5等份，因此樣本也平均分配成每一區間各抽取200個樣本並帶入，目的是為了讓樣本的分散程度足夠、代表性足夠，將這5組資料分別取平均後得到   , …, ，再將這5個估計值取平均即為的估計值 而的估計值為 0.5 - 。重複上述動作1000次，最後取這1000筆估計值的平均與變異數，分別為0.25001367與0.00000009，此為Stratified Sampling 估計方法。

實際 *P*( *X* > 1) = 0.25

以Monte-Carlo Integration為基準，並使用其他變異數縮減方法去降低估計的誤差，使用方法有以下幾種：Hit or miss、Antithetic Variate、Importance Sampling、Control variate以及Stratified Sampling

（其中用於Control variate的另一函數為, x = 0 ~ 1，Stratified Sampling將定義域分為5層並平均分配抽樣數，用於Importance Sampling的另一函數為。）

每一個方法皆取1000個隨機樣本生成一個，再生成1000樣本 ，並運用其平均去估計真實的及計算樣本變異數。

統整六種方法估計P( X > 1)的估計值樣本平均數與樣本變異數：

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | MonteCarlo | Hit or miss | Antithetic | Importance | Control | Stratified(5) |
|  | 0.24997910 | 0.2498500 | 0.24999466 | 0.24997706 | 0.25017903 | 0.25001367 |
|  | 0.00000272 | 0.0001932 | 0.00000004 | 0.00000244 | 0.00000983 | 0.00000009 |

另外亦有探討抽樣樣本數的問題，分別取100、1000、10000樣本數，去探討樣本平均以及樣本變異數之間的關係。

在此以表格方式呈現

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n = 100 | Monte-Carlo | Hit or miss | Antithetic | Importance | Control | Stratified(5) |
|  | 0.25028315 | 0.24937000 | 0.25002894 | 0.24989915 | 0.24878815 | 0.2500383 |
|  | 0.00002736 | 0.00185055 | 0.00000041 | 0.00002551 | 0.00009001 | 0.0000008 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n = 1000 | Monte-Carlo | Hit or miss | Antithetic | Importance | Control | Stratified(5) |
|  | 0.24997910 | 0.2498500 | 0.24999466 | 0.24997706 | 0.25017903 | 0.25001367 |
|  | 0.00000272 | 0.0001932 | 0.00000004 | 0.00000244 | 0.00000983 | 0.00000009 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| n = 10000 | Monte-Carlo | Hit or miss | Antithetic | Importance | Control | Stratified(5) |
|  | 0.25001537 | 0.24978600 | 0.2499985 | 0.24997728 | 0.25001918 | 0.25000063 |
|  | 0.00000025 | 0.00001847 | 0.0000001 | 0.00000024 | 0.00000099 | 0.00000001 |

1. 結論：

實際 P( X > 1) = 0.25，由上表可知，每個方法都估計的數值相差無幾，只有Antithetic Variate與Stratified Sampling的樣本變異數來得非常小，而Control variate與Hit or miss來得相對大滿多，其中Control variate可能與取得函數有關，若是能取得更適合的函數，也許可以將樣本變異數降低更多。

而抽樣樣本數也是一個問題，當樣本數由小至大會發現樣本數的樣本變異數會下降很快，會近乎接近0。因此，在本題使用的估計方法以Antithetic Variate與Stratified Sampling為優，樣本數則是越多誤差越小。

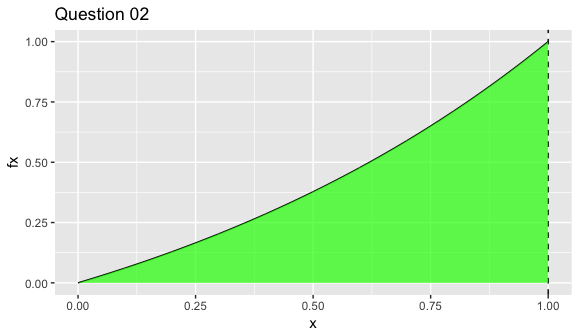


Hammersley and Handscomb (1964) used the integration of $

as a test problem of variance reduction techniques (which is about 0.4180233).

Achieve as large a variance reduction as you can. (They achieved 4 million.)

實際 θ = 0.4180233



以Monte-Carlo Integration為基準，並使用其他變異數縮減方法降低估計的誤差，使用方法有以下幾種： Antithetic Variate、Importance Sampling、Control variate以及Stratified Sampling。每一個方法皆取1000個隨機樣本生成一個，再生成1000樣本 ，並運用其平均去估計真實的及計算樣本變異數。

其中本題運用方法方式與第一題類似，而用於Control variate的另一函數為，Stratified Sampling將x的範圍0~1平均分為5層並平均分配抽樣數，

用於Importance Sampling的另一函數為 。

另外本題亦有多做縮減變異數混搭，在Importance Sampling的方法之中，

套入Antithetic Variate抽取x樣本的方法以藉此來縮小只做Importance Sampling的變異數。

統整六種方法估計θ =估計值的樣本平均數與樣本變異數：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Monte-Carlo | Antithetic | Importance | Control | Stratified(5) |
|  | 0.41787975 | 0.41805356 | 0.41824956 | 0.4180233 | 0.41804309 |
|  | 0.00008453 | 0.00000281 | 0.00015477 | 9.253717e-36 | 0.00000345 |

結論：

實際，由上表可知，每個方法估計θ =的數值相差無幾，

多落在0.417~0.418附近。

接著比較每個方法變異數，以Antithetic Variate*、*Stratified Sampling與Control variate的變異數最小，而Importance Sampling的變異數則是六個之中最大的，可能與Importance Sampling使用的另一函數（）有關，若是能取得更適合的函數，便可以將樣本變異數降下來。

在本題中，我們亦有做縮減變異數方法的混搭，透過兩個方法的結合（Importance Sampling與Antithetic Variate），來讓Importance Sampling方法中的變異數進一步縮減，從0.00015477縮減至0.00004137。因此本題以Antithetic Variate*、*Stratified Sampling與Control variate三個方法為優，另外能透夠增加抽樣樣本數來降低所有方法之變異數。



Let be independent exponential random variables each with mean 1,

and consider the quantity $\theta$ defined by .

Propose at least three simulation methods to estimate $\theta$ and compare their variances.

以Monte-Carlo Integration為基準，並使用其他變異數縮減方法去降低估計的誤差，使用方法有以下幾種：Hit or miss、Antithetic Variate以及Stratified Sampling。在使用Monte-Carlo Integration、Antithetic Variate以及Stratified Sampling時，需要的機率分配形式，透過變數變換後，

的機率分配：

先計算)的估計值，再透過的方式取得估計值。

以Monte-Carlo Integration為例，先從uniform(0,21.6)抽取1000筆樣本，帶入y的機率形式中並乘上21.6後去取平均數即為一個)的估計值，再計算)的估計值即為的估計值。

重復此動作1000次，取得1000筆的估計值後，取此1000筆估計值的平均數與變異數，即為0.16960651與0.00018478。

而Antithetic Variate與Stratified Sampling做法類似Question 01的做法，便能取得該方法所估計估計值的平均數與變異數。

Hit or miss方法則是直接從分配中抽取5筆資料，並計算的數值，

重複取1000筆並計算超過21.6的比例為一個的估計值。

重復抽取1000筆的估計值，取其平均數與變異數為0.16906200與0.00015192。

在此以表方式統整四種方法估計 估計值的樣本平均數與樣本變異數：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Monte-Carlo | Antithetic | Stratified(5) | Hit or miss |
|  | 0.16810602 | 0.16730692 | 0.16894173 | 0.16816900 |
|  | 0.00019783 | 0.00029715 | 0.00002004 | 0.00013109 |

**結論：**

在本題四種方法中**，**估計的 大多都落在0.168附近，唯獨Antithetic Variate與0.168有一點落差，變異數的部分則是Stratified Sampling最小，Antithetic Variate最大。

因此，本題以Stratified Sampling為優。



First, simulate 100 observations from and then use 3 density estimating methods to smooth the observations. You need to specify the parameters in the smoothing methods, and compare the results.

我們使用了下列三個方法來估計我們的密度函數，分別是

1. **Histogram**

我們將亂數取出來min(x)、max(x) 作為我們兩端的範圍，將此範圍切割成適當的大小m

(依照亂數的個數決定)，，

此密度函數的估計值為

為 裡的個數

其中h為環寬，寬度會影響整個估計函數矩形的形狀及大小

1. **The Naïve Density Estimator**

我們將亂數取出來min(x)、max(x) 作為我們兩端的範圍，將此範圍切割成適當的大小m

(依照亂數的個數決定)，，

此密度函數的估計值為

其中h為環寬，寬度會影響整個估計函數曲線的起伏程度

1. **Kernel Estimator**

我們將亂數取出來min(x)、max(x) 作為我們兩端的範圍，將此範圍切割成適當的大小m

(依照亂數的個數決定)，，

此密度函數的估計值為

其中h為環寬，寬度會影響整個估計函數曲線的平滑程度，

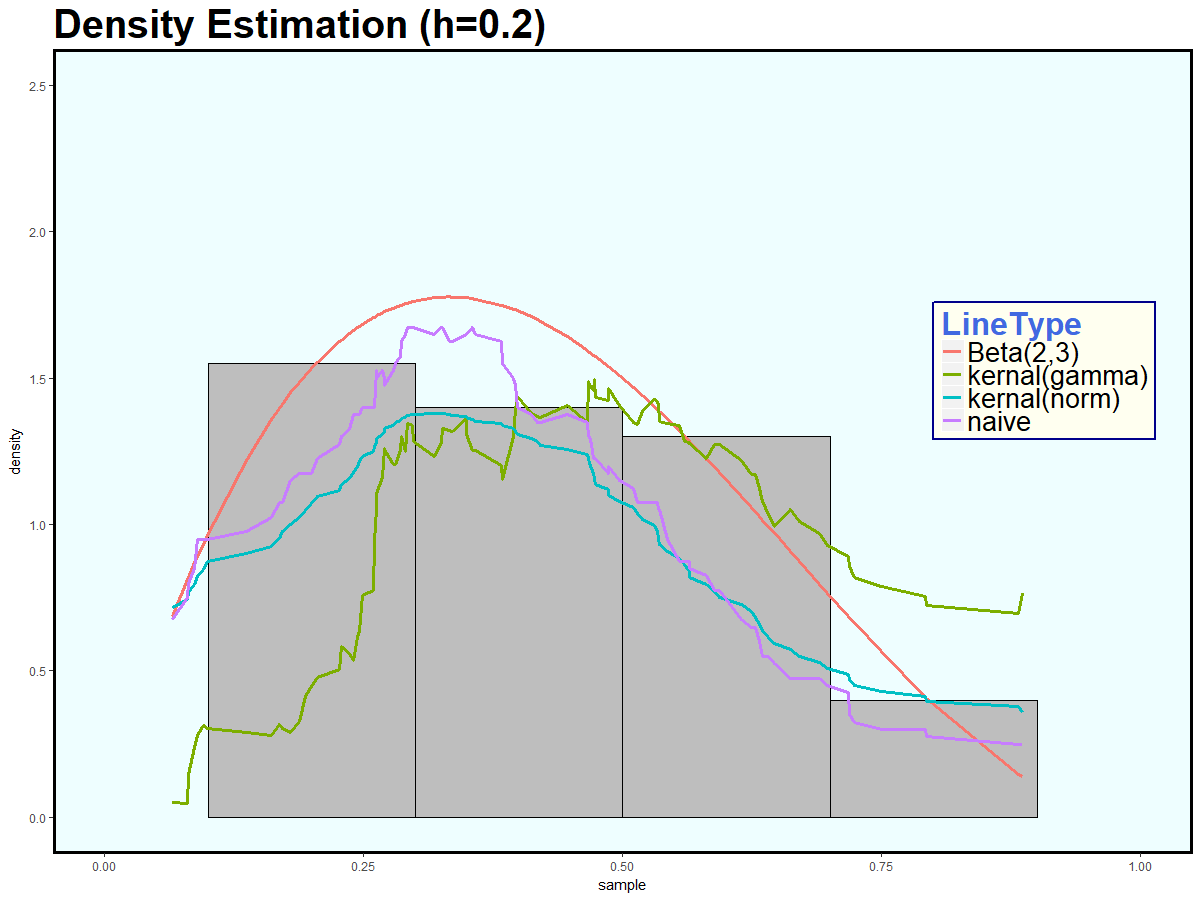
K為一機率密度函數，這裡我們使用標準常態分配作為核密度函數，

並用此算式來計算新的估計值，並將結果繪製成圖表。

這是我們模擬出的x散佈圖，接者將這先模擬出的亂數帶入上面提及之方法函數，將各方法所得出的新值繪製在相同圖表上比較，



我們使用上述三種方法來將模擬出的新數值進行繪製，並將圖形堆疊至相同圖表進行比較



( 橘色線為 Beta(2,3) ; 綠色線為 kernel(Gamma) ; 藍綠色線為 kernel(norm) ; 紫色線為 naïve )

從上述兩圖中我們可以發現不同h的選擇會讓曲線的平滑程度跟起伏程度有所改變，以此題分配模擬出的100個亂數中，我們發現h=0.1所繪製的圖表較為準確，而h=0.2所繪製的圖表其高度都有所下降，與前述的圖表比較有稍微平滑一點。

這裡我們選擇了兩種核密度函數來估計函數，從繪製的圖形中我們可以看出，使用Gamma函數來估計函數沒有比使用常態函數來估計的好，這裡是個之後可以再深入探討的點，該使用哪種核密度函數來估計會是比較好的方法，而在此題之中，我們一致認定使用常態分配的函數會有較好的準確程度



Let be 100 equally spaced points on and let with .

Apply at least 3 linear smoothers and compare the differences, with respect to mean squares error

(i.e., and variance) from 1,000 simulation runs.

這裡我們使用了下列四種方法使模擬點的曲線變得更平滑

1. **Kernel Smoothers**

根據上式函數，我們將模擬值帶入並計算出新的估計值，因為K為機率密度函數，

當我們將新的估計值繪出線段時會較原本模擬值來的平滑許多。

1. **Lowess (局部加權迴歸)**

我們將模擬值依序排列，可以設定每次使用多少個鄰近模擬值，配飾出該點的迴歸線，

並計算該點新的模擬值，最後將其繪出成較平滑的線段。

1. **Spline smooth**

首先將原始觀察值分成k+1個區間，其中為這k+1個區段內多項式的交接點，

我們一般在每個區間內使用一個三次多項式，並利用模擬值來配飾出各區間的三次式，

再依照各區間及三次式匯出成較平滑的線段。

(其線段會一節點選擇而有不同的平滑程度)

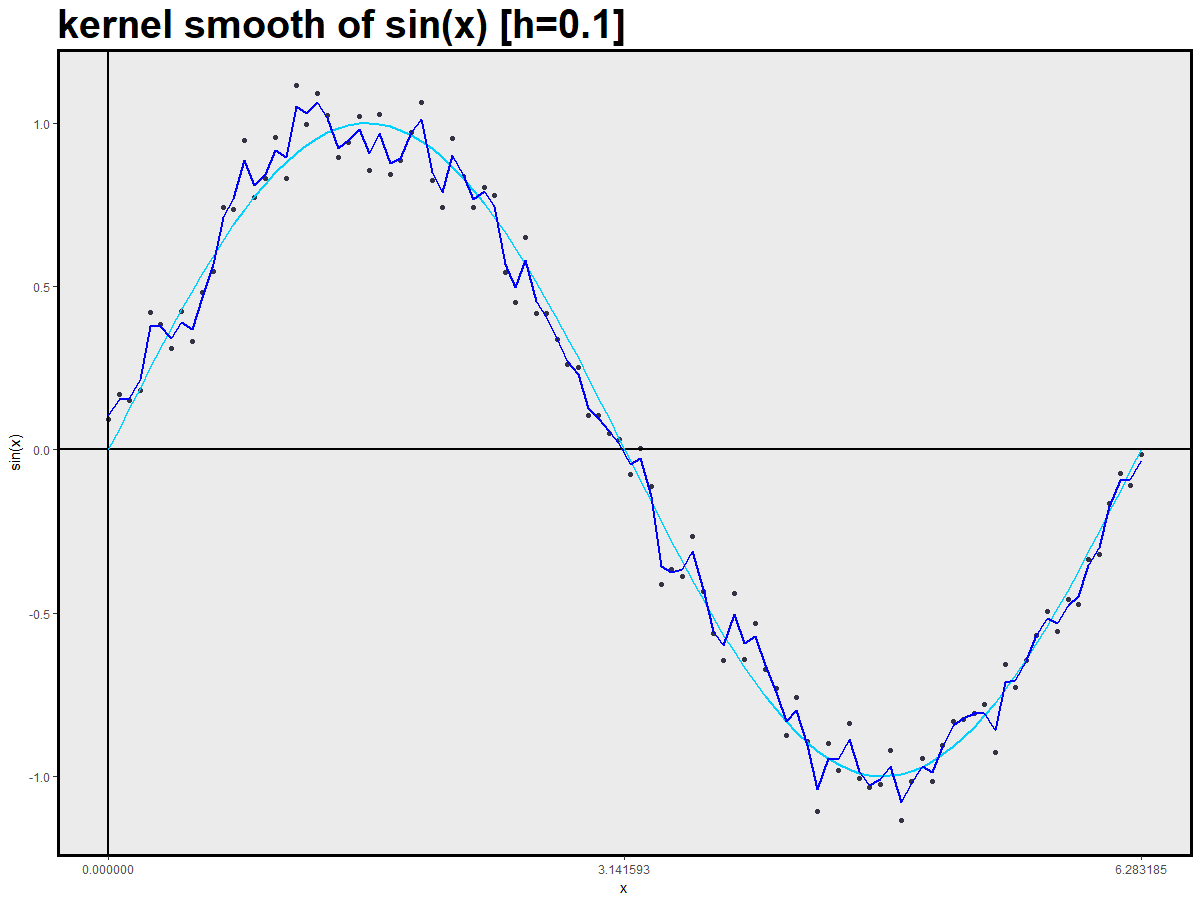
1. Running means (移動平均法)

我們將模擬值依序列出，並設定我們想要的移動平均的個數，其新的模擬值為該點周圍K個模擬值的平均，最後將其新值繪出成線段。

當曲線使用這四種方法平滑後，我們分別模擬不同方法1000次，並計算各方法之均方誤差MSE，其為預測誤差平方值總和的平均。

(a) Kernel smooth

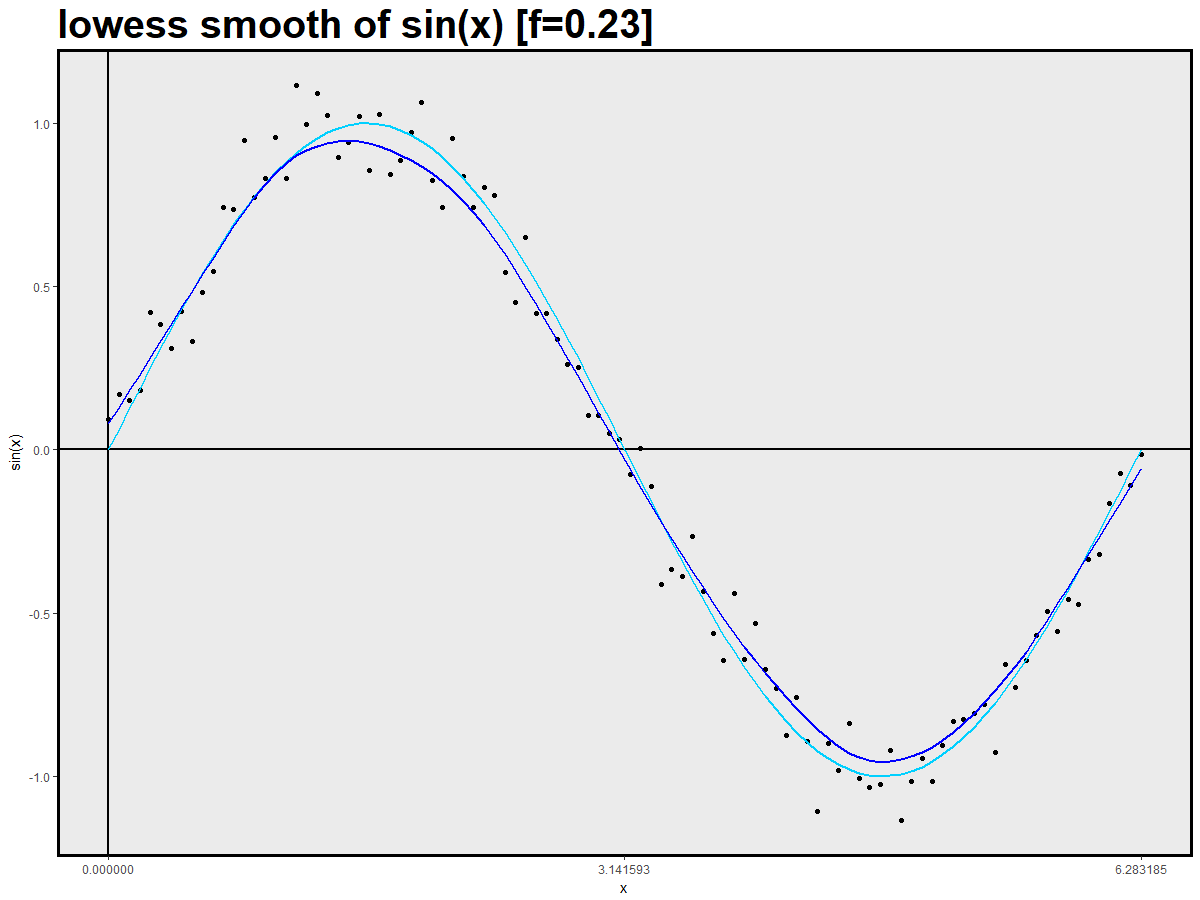
在這裡我們使用了核密度函數為標準常態分配的方法來繪製線段，可以看出線段有比模擬值點雨點間的連線來的稍微平滑，如果我們選取的核密度函數不同，則會有不同的平滑程度。



(淺藍色線段為 sin(x) ; 深藍色線段為 kernel smooth line)

(b) Lowess

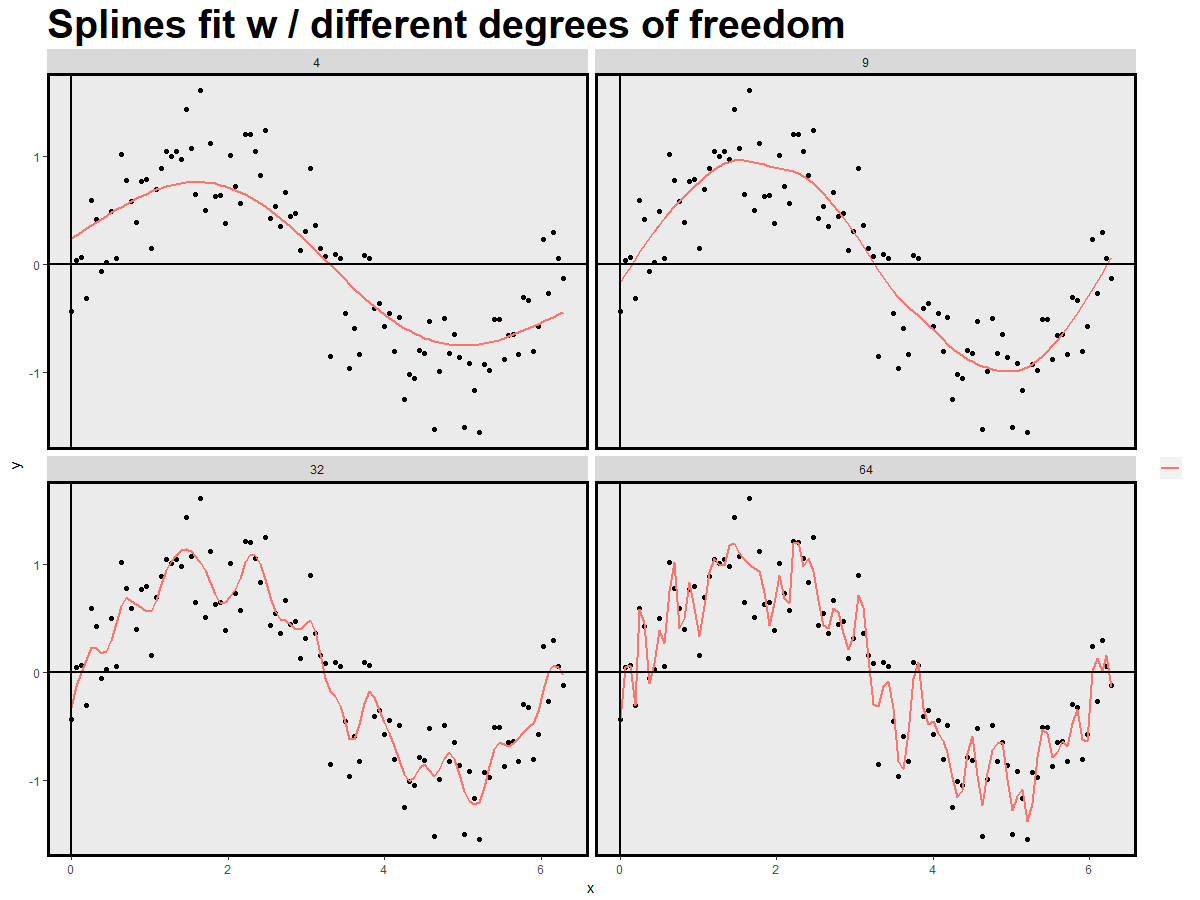
這裡我們先在R軟體設定f變數為0.23，這給出了影響每個值平滑的圖中點的比例，而較大的值會使平滑度更高，我們找到一個適合我們原本圖形的平滑程度的比例，並將其繪製出線段。



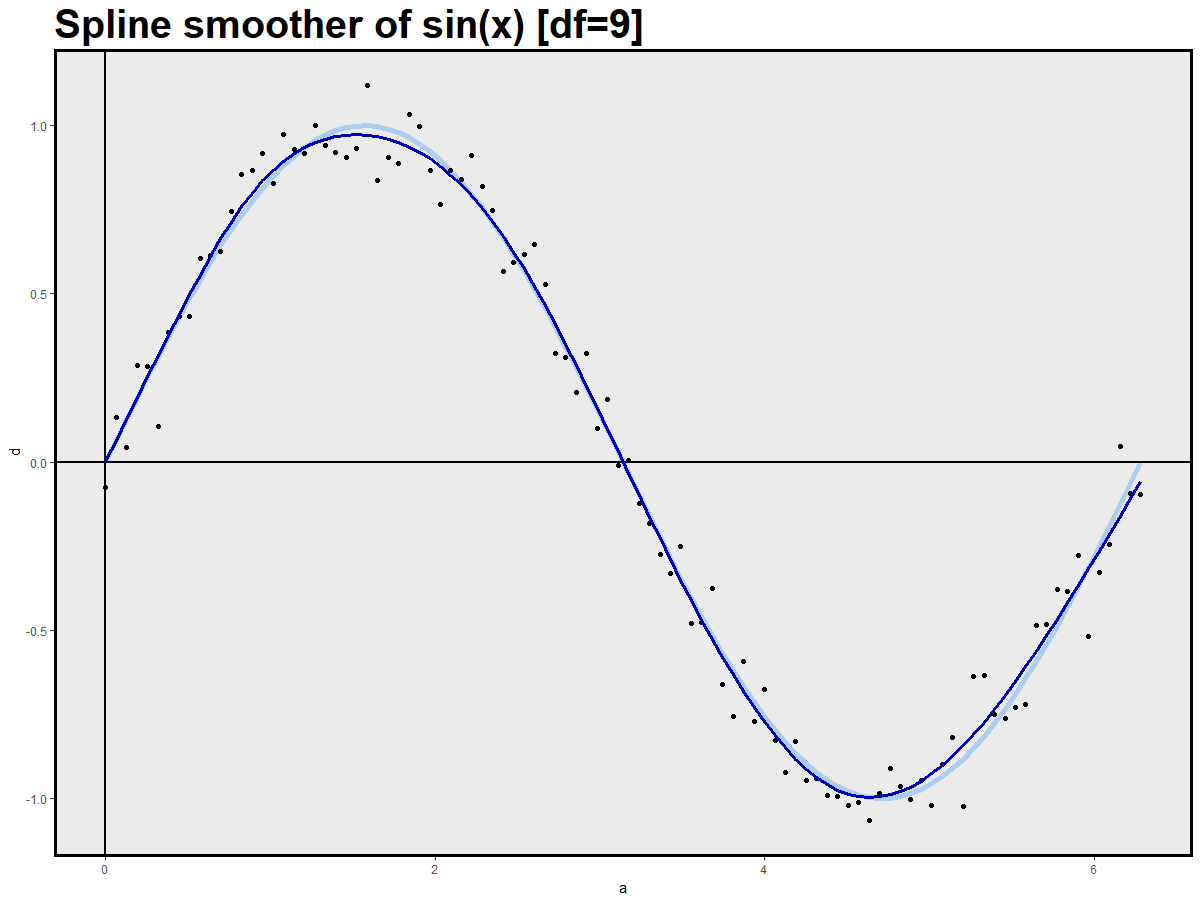
(淺藍色線段為 sin(x) ; 深藍色線段為 lowess smooth line)

(c) Spline smooth

我們先按照節點選取的數量分別畫出平滑線來觀看，並發現不同節點的選擇會讓平滑線的曲度不一樣，這裡我們選取了自由度分別為4、9、32、64，共四組來畫出不同自由度下的平滑曲線。

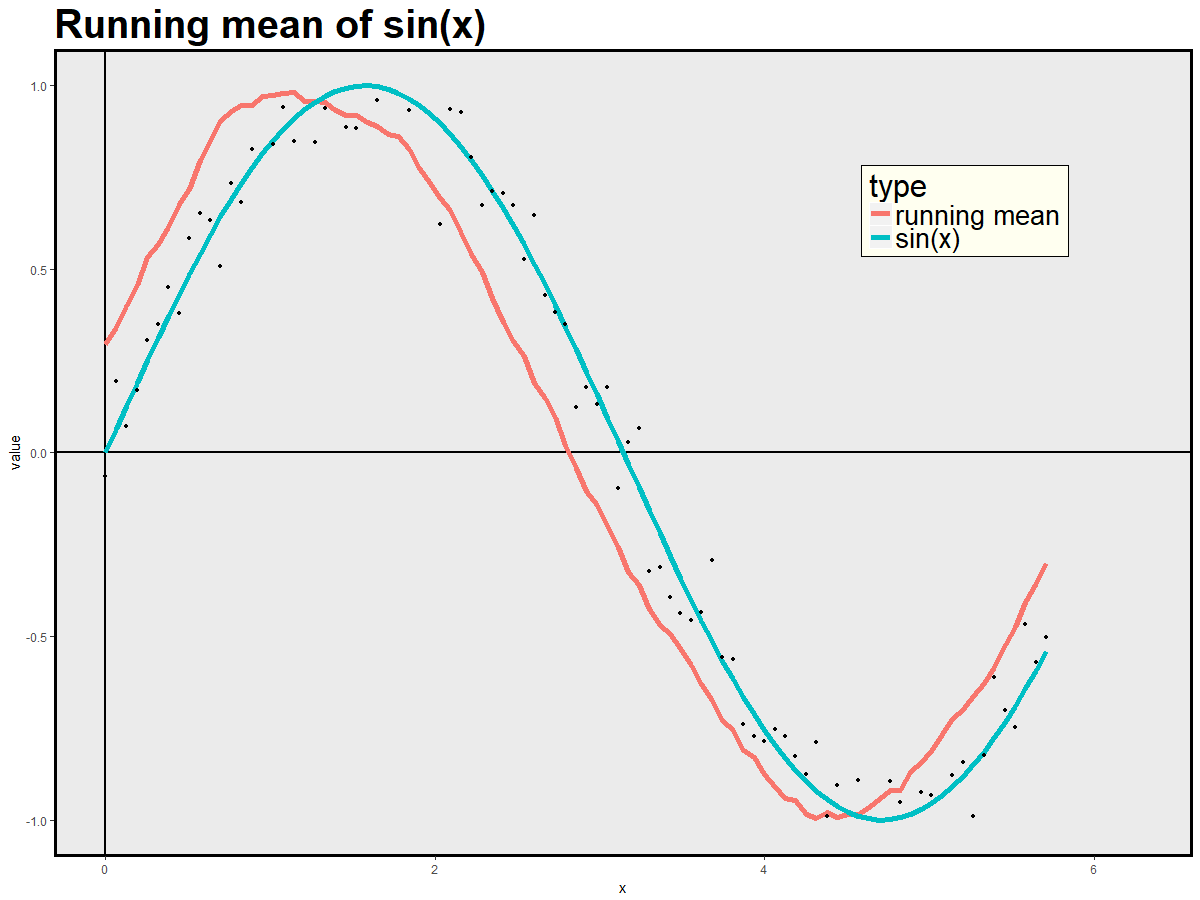


我們發現當自由度選擇過大時，平滑線的曲線反而會變得震盪許多，接著我們將各自由度帶入計算其MSE，發現在模擬1000次後的結果中，自由度為9的平滑曲線其MSE最小，並將其自由度繪出圖形觀看，發現其曲線與真實的sin函數圖形幾乎重疊。



(淺藍色線段為 sin(x) ; 深藍色線段為 Spline smooth line)

(d) Running mean smooth



(淺藍色線段為 sin(x) ; 紅色線段為 Running mean smooth line)

(e)結論

我們使用這四種方法，模擬1000次並計算其MSE，呈現在下列表格中

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Kernel smooth | Lowess smooth | Cubic Spline | Running Means |
|  |  |  |  |  |

從上圖表格我們可以發現Spline smoother 的MSE最小，而從我們上述所繪出的線段判斷來說，Spline smooth 的線段確實很靠近我們想要估計的sin函數線段。

我們覺得Cubic Spline smooth的方法最好，但如果當模擬值過多時，我們發現其運算會很耗費時間，因為當結點數越多時，其矩陣的大小就越大，會因為矩陣的大小影響運行速度。

* 附錄 (R code)

Github : <https://github.com/CaoCharles/Statistical-Computing-and-Simulation-HW3>

R Markdown :



