统计计算 HW3

朱强强 17064001

2020 年 4 月 14 日

写出使用以下两种抽样密度，计算蒙特卡洛积分 $\int g(x)dx = \int_0^1 xdx$ 的算法，并比较方差。

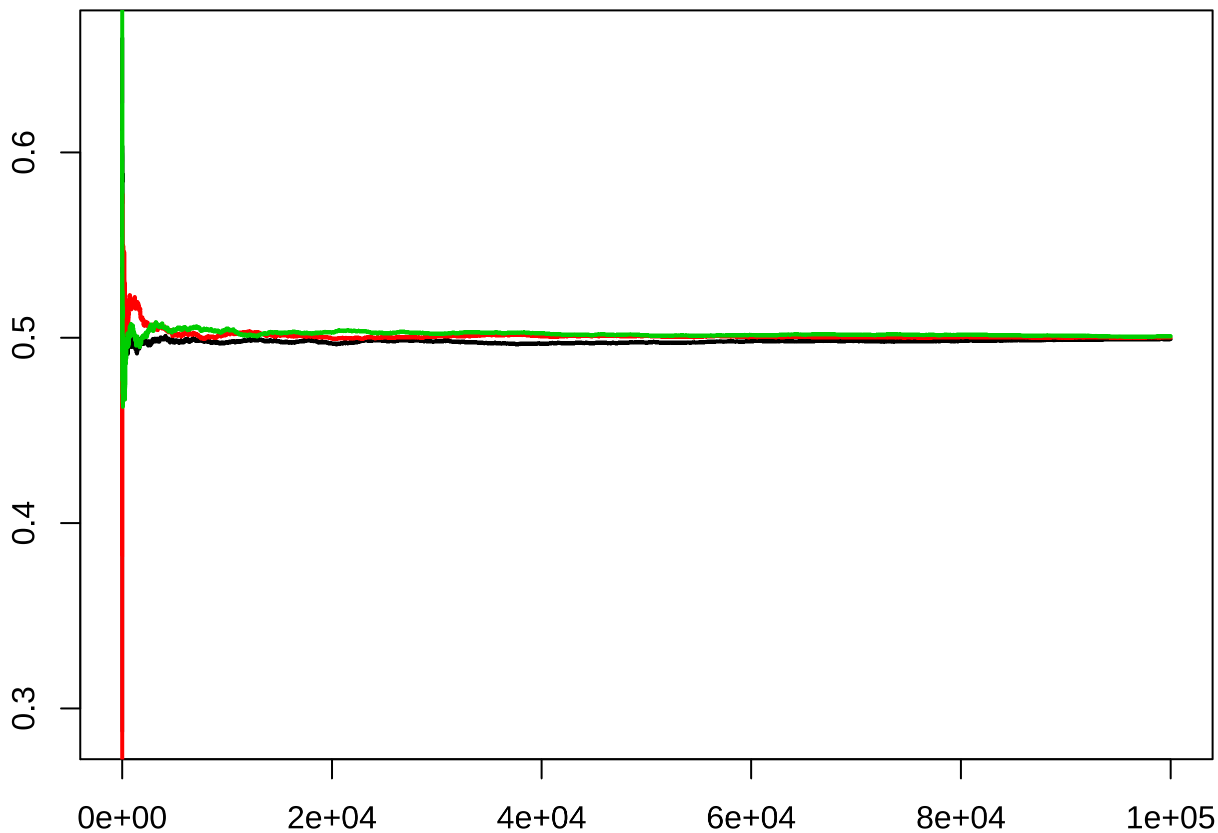
- $p(x) = 1, 0 < x < 1$
- $q(x) = 2x, 0 < x < 1$

当 $p(x) = 1, 0 < x < 1$ 时，用逆变换法产生 $f(x) = 2x$ 的随机数，

```
> set.seed(123)
> n <- 1e5
> x1 <- runif(n)
> cat("Estimate=", mean(x1), "; var=", var(x1),
+     "; True=", 0.5, sep="", "\n")
>
> set.seed(456)
> x2 <- runif(n)
> cat("Estimate=", mean(x2), "; var=", var(x2), "; True=", 0.5, sep="")
```

```
Estimate=0.4992992; var=0.08317841; True=0.5
Estimate=0.5001497; var=0.08377207; True=0.5
```

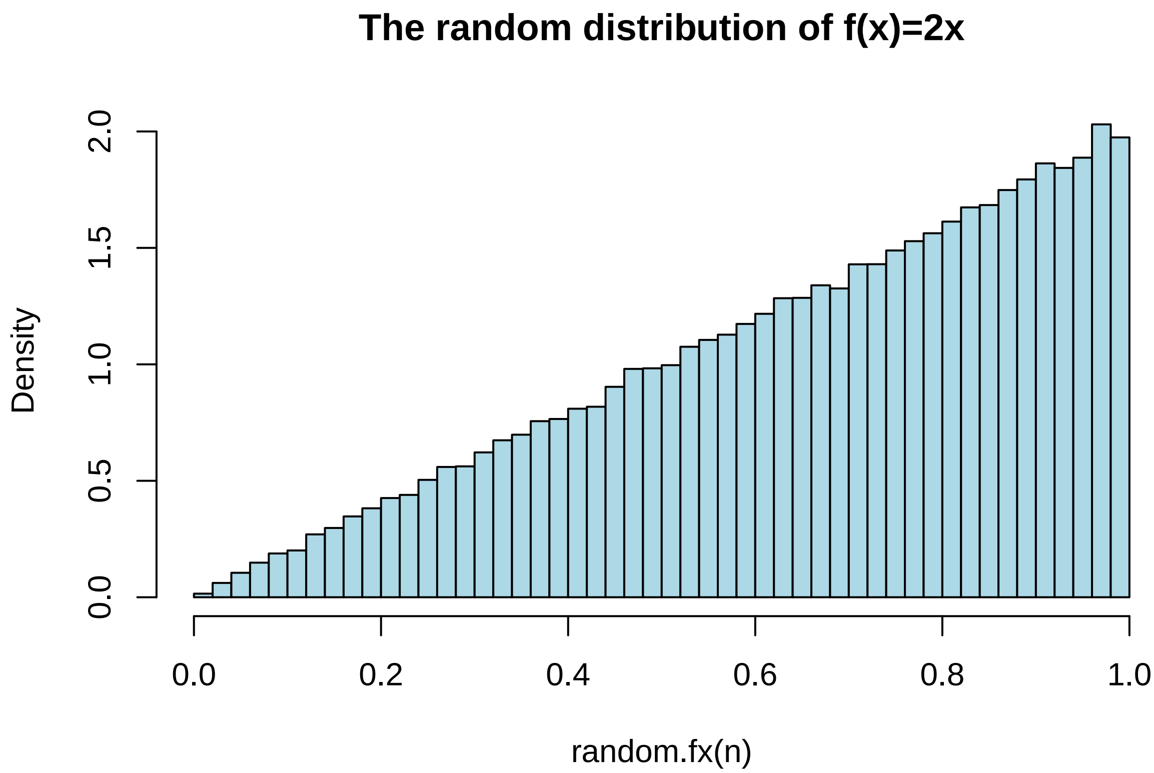
```
> par(mar=c(2, 2, 1, 1))
> hat.theta.n1 <- cumsum(x1)/(1:n)
> plot(hat.theta.n1, type="l", lwd=2)
> hat.theta.n2 <- cumsum(x2)/(1:n)
> lines(hat.theta.n2, lwd=2, col=2)
> x3 <- runif(n)
> hat.theta.n3 <- cumsum(x3)/(1:n)
> lines(hat.theta.n3, lwd=2, col=3)
```



当 $q(x) = 2x, 0 < x < 1$ 时

$$F(x) = x^2$$
$$F^{-1}(x) = \sqrt{x}$$

```
> h <- 0.5
> n <- 1e5
>
> # 逆变换法产生f(x)=2x的随机数
> random.fx <- function(n) {
+   u <- runif(n)
+   invFu <- sqrt(u)
+   return(invFu)
+ }
>
> hist(random.fx(n), nclass=50,
+      main="The random distribution of f(x)=2x", prob=T,
+      col="lightblue")
```




```
> set.seed(123)
> x1 <- random.fx(n) * h
> cat("Estimate=", mean(x1), "; var=", var(x1),
+     "; True=", 0.5, sep="", "\n")
>
> set.seed(456)
> x2 <- random.fx(n) * h
> cat("Estimate=", mean(x2), "; var=", var(x2), "; True=", 0.5, sep="")
```

```
Estimate=0.3331424; var=0.01384109; True=0.5
Estimate=0.3333106; var=0.01394164; True=0.5
```

由上面结果可知，抽样密度为 $p(x) = 1, 0 < x < 1$ 生成的蒙特卡罗积分比 $q(x) = 2x, 0 < x < 1$ 生成的蒙特卡洛积分更接近于真实值。后者的方差明显比前者的方差小。