节选第七章

7.8. It can be shown that if we add random numbers until their sum exceeds 1, then the expected number added is equal to e. That is, if

$$N = min\left\{n : \sum_{i=1}^{n} U_i > 1\right\} \tag{1}$$

then $E[N] = e$.

(a) Use this preceding to estimate e, using 1000 simulation runs.

(b) Estimate the variance of the estimator in (a) and give a 95 percent confidence interval estimate of e.

(a)-(b) 代码如下：

```
f <- function(n) {
  result <- vector(length = n)
  for(i in 1:n) {
    sum <- 0
    count <- 0
    while(sum <= 1) {
      r <- runif(1, min = 0,  max = 1)
      sum <- sum + r
      count <- count + 1
    }
    result[i] <- count
  }
  mean <- mean(result)
  var <- var(result)
  print(mean)
  print(var)
  # 计算e的95%置信区间
  dL <- mean - 1.96*sqrt(var)/sqrt(n)
  dU <- mean + 1.96*sqrt(var)/sqrt(n)
  print(c(dL, dU))
}
```

2

f(1000)

第 8 题运行截图

```
> f(1000)
[1] 2.725
[1] 0.7461211
[1] 2.671462 2.778538
```

图 1: 第 8 题运行截图

经 1000 次模拟,e 值为 2.725, 其方差为 0.7461211.
e 的 95% 置信区间为 [2.671462,2.778538]

7.9. Consider a sequence of random numbers and let M denote the first one that is less than its predecessor. That is

$$M = min\{n : U_1 \leq U_2 \leq ... \leq U_{n-1} > U_n\}$$

(a) Argue that $P\{M > n\} = \frac{1}{n!} \quad n \geq 0$.
(b) Use the identity $E[M] = \sum_{n=0}^{\infty} P\{M > n\}$ to show that $E[M] = e$.
(c) Use part (b) to estimate e, using 1000 simulation runs.
(d) Estimate the variance of the estimator in (c) and give a 95 percent confidence interval estimate of e.
解:(a) 这里介绍三种理解方法:
法一:
4 个人排队一共有 $A_4^4 = 4!$ 种站法, 其中只有一种是完全按照学号顺序来的, 即 $P\{M > 4\} = \frac{1}{4!}$. 以此类推, 50 个人排队做核酸一共有 $A_{50}^{50} = 50!$ 种站法, 在这么多站法中, 只有 1 种是完全按学号顺序站的, 那么翻译过来就是 $P\{M > 50\} = \frac{1}{50!}$, 因此:$P\{M > n\} = \frac{1}{A_n^n} = \frac{1}{n!} \quad (n \geq 0)$.
法二:
这个 (a) 题其实可以抽象成有 n 个人在” 乱” 站. 假设全班 50 个同学要站队去核酸检测, 第一个来站岗的同学学号是 44, 第二个来的学号是 21, 他有两种选择: 站 44 前面或 44 后面, 概率各是 $\frac{1}{2}$(因为是乱站, 所以来站队的人站在哪里是没有选择倾向的, 故概率各是 $\frac{1}{2}$, 而非 $\frac{43}{49}$ 和 $\frac{6}{49}$, 最后两个人是否按学号顺序站是站完之后再去观测的, 因此概率各是 $\frac{1}{2}$). 假如 21 站在 44 后面, 那么翻译过来就是 $P\{M = 2\}$. 假如 21 站在 44 前面, 那么翻译过来就

是 $P\{M>2\}$(其值为 $\frac{1}{2}$). 想让排队继续就必须 21 站在 44 前面. 现在学号为 07 的同学下来了, 他有三种选择: 站 21 前面 (07 21 44)、站 21 和 44 中间 (21 07 44)、站 44 后面 (21 44 07), 概率各为 $\frac{1}{3}$, 第一种情况翻译过来就是 $P\{M>3\}$, 其值为 $\frac{1}{2}*\frac{1}{3}=\frac{1}{6}=\frac{1}{3!}$. 再来一个学号为 49 的同学, 他选择站哪里其实就是个插空问题, 前面下来的 3 位同学构成了 4 个空, 49 号选择站哪里概率都是 $\frac{1}{4}$, 假如他选择完毕后 4 个人的站队次序为 (07 21 44 49), 那么翻译过来就是 $P\{M>4\}$, 其值为 $\frac{1}{2}*\frac{1}{3}*\frac{1}{4}=\frac{1}{24}=\frac{1}{4!}$. 以此类推, 可以推导出 $P\{M>n\}=\frac{1}{n}*\frac{1}{n-1}*...*\frac{1}{3}*\frac{1}{2}=\frac{1}{n!}\quad(n\geq 0)$.

法三:(最没技术含量的解法, 建议跳过)

假设 4 个同学要排队去核酸检测, 规定下来的人只能按先来后到的顺序依次往后站. 第一个来站岗的人学号是未知的, 每个人都有 $\frac{1}{4}$ 的概率最先来排队. 现在我们假设第一个人学号是 1, 那么后面来的第一个人, 学号比 1 大的概率是 $\frac{3}{3}$, 假设第一个人学号是 2, 后面来的第一个人学号比 2 大的概率就是 $\frac{2}{3}$, 依次类推, 第二个来的人比第一个来的人学号大的概率就是

$$\frac{\frac{3}{3}+\frac{2}{3}+\frac{1}{3}+\frac{0}{3}}{4}=\frac{1}{2}=\frac{1}{2!}$$

然后又来了一个人, 重复上面的思想, 第三个人比第二个人学号大的概率为:

$$\frac{\frac{3}{3}(\frac{\frac{2}{2}+\frac{1}{2}+\frac{0}{2}}{3})+\frac{2}{3}(\frac{\frac{1}{2}+\frac{0}{2}}{2})+\frac{1}{3}(\frac{\frac{0}{2}}{1})+\frac{0}{3}}{4}=\frac{1}{6}=\frac{1}{3!}$$

以此类推, 不难发现通项公式为以下形式的嵌套, 但是嵌套完求解起来相当复杂

$$A_n=\frac{\frac{n-1}{n-1}+\frac{n-2}{n-1}+...+\frac{1}{n-1}+\frac{0}{n-1}}{n}$$

再对分子的每一项都乘上 A_{n-1}, 再对 A_{n-1} 的分子的每一项都乘上 A_{n-2}, 再对 A_{n-2} 的分子的每一项都乘上 A_{n-3}, 以此类推, 直至无法再嵌套为止, 最后可以通过数学归纳法求解得到 $P\{M>n\}=\frac{1}{n!}\quad(n\geq 0)$

解:(b) 这题直接用泰勒公式

$$f(x)=f(0)+f'(0)x+\frac{f''(0)}{2!}x^2+\frac{f'''(0)}{3!}x^3+...+\frac{f^{(n)}(0)}{n!}x^n+o(x^n)$$

构造 $f(x)=e^x$, 将 $x=1$ 代入泰勒公式, 得到:

$$f(1)=1+1+\frac{1}{2!}+\frac{1}{3!}+...+\frac{1}{(n-1)!}+\frac{1}{n!}=e$$

解:(c)-(d)

```
f <- function(n) {
  result <- rep(1,n)
  for(i in 1:n) {
    r <- runif(10, min = 0, max = 1)
    for(j in 1:10) {
    result[i] = result[i] + 1
      if(r[j+1] > r[j]) {
        break
      }
    }
  }
  mean <- mean(result)
  var <- var(result)
  dL <- mean - 1.96*sqrt(var)/sqrt(n)
  dU <- mean + 1.96*sqrt(var)/sqrt(n)
  print(mean)
  print(c(dL, dU))
}
f(1000)
```

第 9 题 (c)-(d) 运行截图

```
> f(1000)
[1] 2.722
[1] 2.667348 2.776652
```

图 2: 第 9 题 (c)-(d) 运行截图

经 1000 次模拟,e 值为 2.722, 其 95% 置信区间为 [2.667348,2.776652]

7.13. Let $X_1, ..., X_n$ be independent and identically distributed random variables having unknown mean μ, For given constants $a < b$, we are interested in estimating $p = P\{a < \sum_{i=1}^{n} X_i/n - \mu < b\}$.

(a) Explain how we can use the bootstrap approach to estimate p.

(b) Estimate p if $n = 10$ and the values of the X_i are 56,101,78,67,93,87,64,72,80,and

69. Take $a = -5, b = 5$.

解:(a) 题

Step1: 有放回地从 n 个数中随机抽取 n 次, 抽取的 n 个数构成一组自助样本

Step2: 重复 Step1 至少 1000 次, 得到 1000 个 $\sum_{i=1}^{n} X_i/n$ 统计量

Step3: 将 1000 个 $\sum_{i=1}^{n} X_i/n$ 统计量分别减去 $\mu = \frac{X_1+X_2+...+X_n}{n}$, 留下其值在 [-5,5] 之间的, 再除以自助样本的个数 (1000), 即为 p

解:(b) 题, 代码如下:

```
f <- function(a, b, data, n) {
  result <- vector(length = n)
  count <- 0
  for(i in 1:n) {
    r <- sample(data, length(data), replace = TRUE)
    result[i] <- mean(r)
    if(((mean(r) - mean(data)) > a) &&
       ((mean(r) - mean(data)) < b)) {
      count <- count + 1
    }
  }
  print(mean(result))
  print(count/n)
}
data <- c(56,101,78,67,93,87,64,72,80,69)
f(-5,5,data,66666)
```

第 13 题 (b) 运行截图

```
> f(-5,5,data,66666)
[1] 76.72828
[1] 0.7605826
```

图 3: 第 13 题 (b) 运行截图

经 66666 次自助抽样, 得到 p 值为 0.7605826

In the following three exercises $X_1, ..., X_n$ is a sample from a distribution whose variance is (the unknown) σ^2. We are planning to estimate σ^2 by the sample vari-

ance $S^2 = \sum_{i=1}^{n}(X_i - \overline{X})^2/(n-1)$, and we want to use the bootstrap technique to estimate $Var(S^2)$.

7.14. If $n = 2$ and $X_1 = 1$ and $X_2 = 3$, what is the bootstrap estimate of $Var(S^2)$?

解: 代码如下:

```
f <- function(data, n) {
  result <- vector(length = n)
  for(i in 1:n) {
    r <- sample(data, length(data), replace = TRUE)
      result[i] <- sum((r-mean(r))^2)/
      (length(data)-1)
  }
  varS2 <- sum((data-mean(data))^2) / (length(data)-1)
  print("估计量有偏的方差如下")
  print(var(result))
  print("估计量无偏的方差如下")
  print(mean((result-varS2)^2))
}
data <- c(1,3)
f(data,66666)
```

第 14 题运行截图

```
> f(data,66666)
[1] "估计量有偏的方差如下"
[1] 0.9999961
[1] "估计量无偏的方差如下"
[1] 2.0087
```

图 4: 第 14 题运行截图

这个 14 题有些表意不明, 没有说 $\hat{S^2}$ 是否是 S^2 的无偏估计

$$MSE(\hat{S^2}) = E(\hat{S^2} - S^2)^2 = Var(\hat{S^2}) + (E(\hat{S^2}) - S^2)^2$$

如果是有偏的, 那么上式的 $(E(\hat{S^2}) - S^2)^2 > 0$, 此时 $Var(S^2)$ 直接是自助样本的生成的 S^2 的方差, 其值为 0.9999961; 如果是无偏的, 那么上式的 $(E(\hat{S^2}) - S^2)^2 = 0$, 此时 $Var(S^2) = MSE(\hat{S^2})$, 其值为 2.0087.

7.15. If $n = 15$ and the data are 5,4,9,6,21,17,11,20,7,10,21,15,13,16,8 approximate(by a simulation) the bootstrap estimate of $Var(S^2)$.
解: 本题代码与 14 题一模一样, 把传入的参数变一下即可
第 15 题运行截图

```
> f(data,66666)
[1] "估计量有偏的方差如下"
[1] 58.51285
[1] "估计量无偏的方差如下"
[1] 63.78225
```

图 5: 第 15 题运行截图

同样, 如果 $\hat{S^2}$ 是 S^2 的有偏估计, 那么结果为 58.51285; 如果是无偏估计, 结果为 63.78225.

7.16. Consider a single-server system in which potential customers arrive in accordance with a Poisson process having rate 4.0. A potential customer will only enter if there are three or fewer other customers in the system when he or she arrives. The service time of a customer is exponential with rate 4.2. No additional customers are allowed in after time $T = 8$. (All time units are per hours.) Develop a simulation study to estimate the average amount of time that an entering customer spends in the system. Using the bootstrap approach, estimate the mean square error of your estimator.
考虑一个单服务器系统, 其中潜在客户根据泊松过程到达, 泊松过程的速率为 4.0. 只有当系统中有三个或更少的其他客户时, 潜在客户才会进入. 客户的服务时间服从参数为 4.2 的指数分布. $T = 8$ 之后, 不允许有新客户进入系统.(所有时间单位均为每小时) 开发一项模拟研究, 以估计进入客户在系统中花费的平均时间, 并使用自助法估计其均方误差.
解: 先上代码

```
f <- function(lambda, T) {
  n <- 10000
  SPEND <- vector(length = n)
  h <- function(lambda, T) {
    g <- function(lambda, T) {
```

```
  S <- vector(length = 0)
  t = 0
  C = 0
  for(i in 1:100) {
    Df = runif(1)
    x = -log(Df)
    t = t+x/lambda
    if(t > T) {
      re <- list(count = C, time=S)
      return(re)
    }
    C = C+1
    S <- append(S, t)
  }
}
result <- g(lambda,T)
finishT <- vector(length=0)
spendT <- vector(length=0)
t <- result$time[1]
r <- rexp(1, rate = 4.2)
spendT <- append(spendT, r)
finishT <- append(finishT, r+t)
C <- 1
for(i in 2:result$count) {
  t <- result$time[i]
  while((C > 0) &&
        (t >= finishT[length(finishT)-C+1])) {
    C <- C-1
  }
  if(C >= 4) {
    next
  }
  r <- rexp(1, rate = 4.2)
```

```
      finishT <- append(finishT,
                  r+max(finishT[length(finishT)], t))
      spendT <- append(spendT,
                finishT[length(finishT)]-t)
      C <- C+1
    }
    return(mean(spendT))
  }
  for(i in 1:n) {
    SPEND[i] <- h(lambda,T)
  }
  mean <- mean(SPEND)
  sd <- vector(length = n)
  for(i in 1:n) {
    sd[i] <- (SPEND[i]-mean)^2
  }
  mse <- mean(sd)
  print(mean)
  print(mse)
}
f(4,8)
```

全过程解释:
在此之前, 需要注意两个地方:(1) 泊松过程产生的是潜在客户, 而不是最终进入到系统内的客户;(2) 这题的均方误差 $E(\hat{\theta}-\theta)^2$, 其中的 θ 是没办法知道其真实值的, 需要用自助样本的均值去替代 (开始我是不确定的, 后来去知网和谷歌学术上看, 很多人都是这么干的)
程序完整流程: 首先需要生成 $\lambda = 4$ 的泊松过程, 每个值代表着潜在客户到达系统的时间 (注意用词, 这里没说是进入系统). 具体要不要进入系统还要看前面排队的人多不多, 本场景规定当系统内的人数小于或等于 3 人时, 潜在客户才会进入系统, 否则直接 say 拜拜. 本题的难点就在于怎么判断前面有多少人? 于是我们引入 $finishT$ 变量, 用来记录每个进入系统的客户 (千万注意不是潜在客户) 的服务完成时间. 当某客户到达系统的时间大于前面第 4 个进入客户的 $finishT$ 时, 该客户才会进入系统 (实际代码不会这么写, 看

正数第 31-34 行代码即可, 这种写法一举两得, 既可以判断该潜在客户是否能进入系统, 又能实时更新当前潜在客户到达之时系统有多少人).
如果能进入系统, 最终目的是记录其在系统内花费的时间, 这个如果正向思维硬算那可太麻烦啦, 我们可以先计算出该进入客户的离开时间, 再用离开时间减去到达时间就可以计算出在系统内花费的时间了. 于是这个问题就简化为了求该进入客户的离开时间. 具体求法: 分两种情况, 第一种情况是来了没人直接就接受服务, 那么离开时间就是" 到达时间 + 服务时间"; 第二种情况就是来了需要等, 那么离开时间就是" 上一进入系统的客户的离开时间 + 服务时间"
故经过 10000 次模拟, 可以求解得出: 进入系统的客户的平均花费时间为 $\theta = 0.565968$, 均方误差为 $E(\hat{\theta} - \theta)^2 = 0.03476586$
第 16 题运行截图

```
> f(4,8)
[1] 0.565968
[1] 0.03476586
```

图 6: 第 16 题运行截图

16 题附图: 单次输出结果
从上到下的输出结果依次是: 进入客户的到达 (进入) 时间; 进入客户的离开时间; 泊松过程产生的潜在客户数量和潜在客户的到达时间

```
> f(4,8)
 [1] 0.1000629 0.2532870 0.3287330 0.4765461 0.6857056 1.1896856 1.3239551
 [8] 1.4712557 2.0813225 2.4479711 2.6027069 3.6333977 3.6399654 3.7177157
[15] 4.4811056 4.9608925 5.3026868 5.3355327 6.0244518 6.1723857 6.3194629
[22] 6.9252012 7.3996519 7.4316363 7.5435302 7.5507759
 [1] 0.3210782 0.3989658 0.4125784 0.6082221 0.9514259 1.4552406 1.6322871
 [8] 1.7086626 2.3622735 2.4672707 2.6881798 4.0230715 4.2943845 4.3278177
[15] 4.7262081 5.4591160 6.2144955 6.6797857 7.2081737 7.2547056 7.3684487
[22] 7.3734644 7.9389381 8.1333404 8.3475877 8.3969388
$count
[1] 30

$time
 [1] 0.1000629 0.2532870 0.3287330 0.4765461 0.6857056 1.1896856 1.3239551
 [8] 1.4712557 2.0813225 2.4479711 2.6027069 3.6333977 3.6399654 3.7177157
[15] 4.4811056 4.9608925 5.3026868 5.3355327 6.0244518 6.1723857 6.3194629
[22] 6.3681027 6.9252012 7.0881414 7.1011123 7.1392349 7.3996519 7.4316363
[29] 7.5435302 7.5507759
```

图 7: 16 题单次输出结果