Analytics 511 Homework 1

Arif Ali

September 16, 2015

Note: R code is located after the written solutions in Appendix

Problem 1

- a = 9
- b = 28.4605
- c = 17
- d = 23409

Problem 2

As p increases, myattempts system times tend to go down whereas rgeom remained about the same regardless of the p input. This is probably because rgeom is a built in function within R.

Problem 3

Some noticable observations is that when p = 0.01, the difference between the estimated standard deviation and the bulit in one is greater by approximately 2. However, as p increases the difference is less. Please see R code in the appendix.

Problem 4

Please see R code in the appendix.

Problem 5

The main argument against SAS is that in order to use SAS, the user has to pay an expensive fee. While many larger companies can afford to pay for the license, individuals in smaller organizations are at a disadvantage. Since R is open sourced and free, many individuals across different disciplines and organizations can collaborate to make it better whereas the updates to SAS are controlled mainly by one company.

However, because of this one organization, the syntax behind SAS is more uniformed when compared to R, especially with respect to packages. In addition, in R there are multiple ways to do the same thing, which can be confusing when collarborating with different individuals or groups. This actually can impede learning R because the lack of central control of syntax means basic structures are just good practice instead of a requirement.

In terms of technique, both blogs cited tend to agree that R is more robust. While analysis done in SAS can be done in R, the converse isn't true. However, R is still not as favored among big companies as SAS is mainly because of a culture aspect. So while R is both free and more robust, SAS has more industry backing. The one noticable conclusion is that most of these blog posts actually tend to conclude both are worth knowing; the comment sections tend to be where the debate of which is better is.

Sources:

- http://thomaswdinsmore.com/2014/12/15/sas-versus-r-part-two/
- http://www.r-bloggers.com/sas-vs-r-the-right-answer-to-the-wrong-question/

Problem 6

Part A

$$P(in\ Sample) = \frac{1000}{100 * 10^6} = 1 * 10^{-5}$$

Part B

$$P(not\ In\ Any\ Of\ 2000\ Samples) = \left(\frac{100 * 10^6 - 1000}{100 * 10^6}\right)^{2000} \approx 0.98$$

Part C

$$P(at\ Least\ One\ Sample) = 1 - P(not\ In\ Any\ Sample) = 1 - \left(\frac{100 * 10^6 - 1000}{100 * 10^6}\right)^{n} = 0.50 \Longrightarrow$$

$$\left(\frac{100 * 10^6 - 1000}{100 * 10^6}\right)^{n} = 0.50 \Longrightarrow n = 69315$$

Problem 7

A plug-in estimator for p requires generating data from rgenom. From the data we are able to determine the mean of the square of results from the data denoted as A.

$$\hat{p} = \frac{\sqrt{1 + 8A} - 1}{2A}$$

Approximately, 77.6% of the estimated phats are greater than the original p meaning that there is a positive bias. The average of those phats is greater than p = .3 by approxmiately 0.21if we assume every undefined value of phat (2A=0) is converted to 0.

Problem 8

$$E(\hat{X}_6) = 0.01560057$$

Bonus

$$E(X_i) = \left(\frac{1}{2}\right)^{i} \therefore E(X_6) = \left(\frac{1}{2}\right)^{6} = 0.015625\ and\ E(X_4) = \left(\frac{1}{2}\right)^{4} = 0.0625$$

```
####Problem 1
sinResult = sin(1.23)
a = as.numeric(tail(unlist(strsplit(as.character(floor(sinResult*10^10)), "")), n = 1))
a
#[1] 9
b = sqrt(a^2+a^3)
b
#[1] 28.4605
c = nchar(floor(exp((log(b))^3)))
c
#[1] 17
d = sum((1:c)^3)
#[1] 23409
####Problem 2
mytoss = function(p){
  u = runif(1)
  x = as.numeric(u<p)
  return(x)
}
myattempts = function(p){
  counter <- 1
  while (mytoss(p) == 0){
    counter <- counter + 1
  }
  return(counter)
}
pValues = c(0.01, 0.25,0.5,0.75, 0.99)
problem2 = matrix(nrow = 2, ncol = length(pValues))
for(i in 1:length(pValues)){
  avgMyattemptsTimeTimes = c()
  avgRgeomTimes = c()
  myattemptsTime = mean(replicate(10000, system.time(myattempts(pValues[i]))[3]))
  rgeomTimes = mean(replicate(10000, system.time(1 + rgeom(1,pValues[i]))[3]))
  problem2[1,i] = myattemptsTime
  problem2[2,i] = rgeomTimes
}
pValues
#   [1] "0.01" "0.25" "0.5"  "0.75" "0.99"
problem2
#       [,1]      [,2]     [,3]     [,4]     [,5]
#[1,] 0.0011956 0.0001309 9.32e-05 8.44e-05 7.12e-05
#[2,] 0.0000642 #0.0000666 5.55e-05 5.91e-05 5.80e-05
####Problem 3
pValue = c(0.01, 0.2,0.4,0.6,0.8, 0.99)
estimatedpValue = 1:length(pValue)
for(i in estimatedpValue){
  estimatedpValue[i] <- sd(replicate(10000, 1+rgeom(1,pValue[i])))
}
actual_sd_value = sqrt(1-pValue)/pValue
estimatedpValue
#[1] 101.2936308   4.5308430   1.9487092   1.0566343   0.5579683   0.1098053
```

```
actual_sd_value
#[1] 99.4987437  4.4721360  1.9364917  1.0540926  0.5590170  0.1010101
####Problem 4
meanRexp = function(n){
  mean(rexp(n))
}
sapply (rep (20 , times = 10) , meanRexp)
####Problem 6
1000/(100*10^6)
#[1] 1e-05
((100*10^6-1000)/(100*10^6))^2000
#[1] 0.8187226
notInSample = function(n){
  ((100*10^6-1000)/(100*10^6))^n
}
n = 1
InSample = notInSample(1)

while(InSample>.50){
  n = n+1
  InSample = notInSample(n)
}
n
####Problem 7
problem7 = 1:100000
for(i in 1:100000){
aa = rgeom(4,.3)
problem7[i] = mean(aa^2)}
phat = (sqrt(1+8*(problem7))-1)/(2*(problem7))
phat[problem7==0] = 0
##Formula
mean(phat - .3>=0)
#[1] 0.77672
mean(phat) - .3
#[1] 0.2118334
####Problem 8
startrunifs = mean(replicate(100000,runif(1, max = 1)))
print(startrunifs)
#[1] 0.7391124
EstimatesOfX = c(startrunifs)
for(i in 1:5){
  startrunifs = mean(replicate(100000, runif(1, max = startrunifs)))
  EstimatesOfX = c(EstimatesOfX,startrunifs)
}
tail(EstimatesOfX, n = 1)
#[1] 0.03814304
####Bonus
(.5)^6
#[1] 0.015625
(.5)^4
#[1] 0.0625
```