Homework #6 Math 160

Ethan Ashby

Due: 24 Mar, 2021

1. Create solutions using `:::: {.solution}` and `::::` in the .Rmd file.
2. If the .css file is not working for you, type out the Problem headings using `# Problem 1`, `# Solution`, `# Problem 2`, and so on.
3. Knit to an .html file.
4. Print out the .html file inside a browser (such as Chrome or Safari) to a .pdf file.
5. Upload the .pdf file to Gradescope.
6. Mark the pages that your questions are on.
7. Be sure to box your answers with the `\boxed{put in box}` $\LaTeX$ command.
8. Be sure to give nonintegral answers to four significant digits.

Problem 1

Write code to draw (X, Y) from the uniform set of directions in $\mathbb{R}^2$. Using 10^6 draws, estimate $\mathbb{P}(Y \geq 0.7)$, reporting your answer as $a \pm b$.

Solution

To sample uniformly from the set of directions in $\mathbb{R}^2$, we draw $U \sim$ Unif$[0, 1]$, and return $(\cos(\tau U), \sin(\tau U))$.

```
set.seed(1234567)
#function to sample uniformly over directions in r2
samp_unif_dir_r2<-function(){
  u<-runif(1)
  return(c(cos(2*pi*u), sin(2*pi*u)))
}

#return the y coordinates for 10^6 runs
ys<-replicate(10^6, samp_unif_dir_r2()[2])

tibble(
  est=mean(ys>0.7),
  se=sd(ys>0.7)/sqrt(length(ys))
) %>% kable()
```

est	se
0.253559	0.000435

My estimate for $\mathbb{P}(Y \geq 0.7)$ is $\boxed{0.25356 \pm 0.00044}$.

Problem 2

Now consider (X, Y) uniform over the unit disc, so Unif($\{(x, y) : x^2 + y^2 \leq 1\}$). Consider converting to polar coordinates to $R = \sqrt{X^2 + Y^2}$ and $\theta \sim$ Unif($[0, \tau]$).

a. Find, for $r \in [0, 1]$, $\mathbb{P}(R \leq r)$.

b. Write code to sample from R using the inverse transform method.

c. Draw (X, Y) uniformly from the unit circle by first drawing (R, θ) and converting from polar coordinates to Cartesian coordinates.

d. Estimate $\mathbb{P}(Y \geq 0.7)$ for (X, Y) uniform over the unit disc using 10^5 samples. Report your answer as $a \pm b$.

Solution

a. For $r \in [0, 1]$, the area of the enclosed disk is πr^2. The area of the whole unit disc is π, and since we're sampling over the unit disc, the probability that $R \leq r$ is proportional to the area of the smaller disk divided by the area of the unit disc: $\mathbb{P}(R \leq r) = \frac{\pi r^2}{\pi} = \boxed{r^2}$.

b. I will do this by sampling uniformly over possible areas of discs with $r \in [0, 1]$

```
itm_sample_r<-function(){
  #sample random uniform, corresponding to area of smaller disk
  u<-runif(1)
  #now find the radius associated with the uniform area
  r=sqrt(u)
  return(r)
}
```

c.

```
xy_draw_polar<-function(){
  R=itm_sample_r()
  theta=2*pi*runif(1)
  x=cos(theta)*R
  y=sin(theta)*R
  return(c(x,y))
}
```

d.

```
set.seed(1234567)
unit_circ_ys<-replicate(10^5, xy_draw_polar()[2])
tibble(
  est=mean(unit_circ_ys>0.7),
  se=sd(unit_circ_ys>0.7)/sqrt(length(unit_circ_ys))
) %>% kable()
```

est	se
0.09397	0.0009227

My estimate of $\mathbb{P}(Y \geq 0.7) = \boxed{0.09397 \pm 0.00093}$.

I verified this estimate graphically too!

Problem 3

Consider taking a asymmetric random walk on the integers mod 5 ($\{0, 1, 2, 3, 4\}$), using M where $\mathbb{P}(M = -1) = 0.6$ and $\mathbb{P}(M = 1) = 0.4$. So with probability 0.6 add 1 to the current state, and if it reaches 5 replace it with a 0. Else (with probability 0.4) add -1 to the current state, and if it reaches -1 replace it with a 4.

What is the stationary distribution of this chain and why?

Solution

NOTE: The problem has two different definitions of the transition probabilities. In the first statement $\mathbb{P}(M = 1) = 0.4$, but in the second statement, with probability 0.6 we add 1 to the current state. I will choose to follow directions from the first statement.

The stationary distribution of this asymmetric random walk is $\boxed{\text{uniform}}$ on the integers mod 5, because this represents a random walk on a finite group (which we know has a uniform stationary distribution).

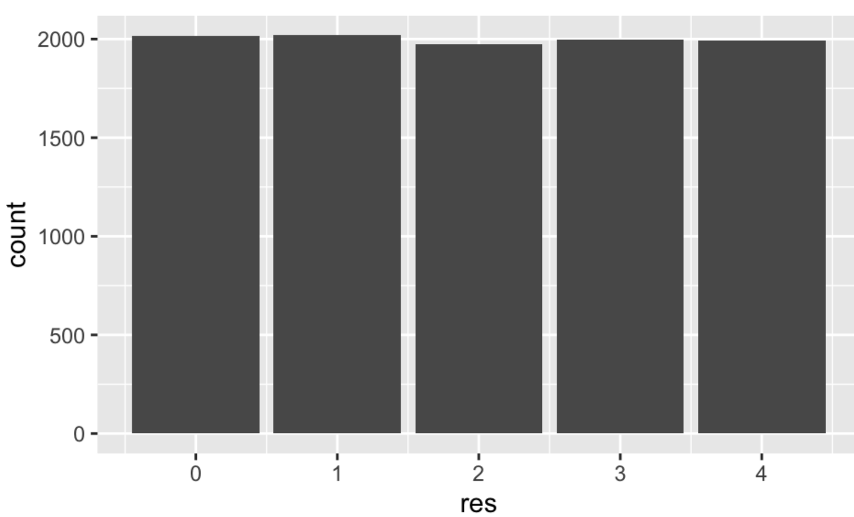
Problem 4

Continuing the last problem, implement the chain and take 10^4 steps starting from state 0. Make a histogram of the states visited by the chain.

Solution

I will illustrate this with a simulation of the markov chain below:

```
set.seed(1234567)
#step function
p3_step<-function(x, m){
  newstate=x+m
  if(newstate<5 & newstate>-1){
    return(newstate)
  }
  if(newstate==5){
    return(0)
  }
  if(newstate==-1){
    return(4)
  }
}

#markov chain
p3_mchain<-function(steps=10^4){
  burnin=steps
  datasteps=steps
  m1=2*(runif(burnin)>0.6)-1
  x<-0
  for(i in 1:burnin){
    x<-p3_step(x, m1[i])
  }
  m2=2*(runif(burnin)<0.6)-1
  data<-c(x, rep(0, datasteps))
  for(i in 1:datasteps){
    x<-p3_step(x, m2[i])
    data[i+1]<-x
  }
  return(data)
}

#now mchain
res <- p3_mchain()

#plot as barchart
ggplot() +
  geom_bar(aes(res))
```



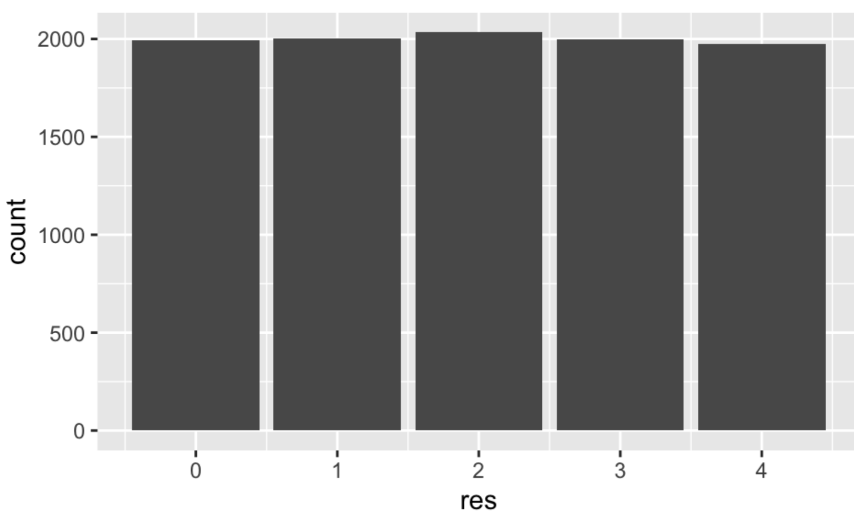
Oh yeah! This matches my expectation of a uniform distribution over this group.

Problem 5

Repeat the last problem, but change the move so that there is a 0.9 chance that $M = -1$, and a 0.1 chance that $M = 1$.

Solution

```
#markov chain with different transition probs
p5_mchain<-function(steps=10^4){
  burnin=steps
  datasteps=steps
  m1=2*(runif(burnin)>0.9)-1
  x<-0
  for(i in 1:burnin){
    x<-p3_step(x, m1[i])
  }
  m2=2*(runif(burnin)<0.6)-1
  data<-c(x, rep(0, datasteps))
  for(i in 1:datasteps){
    x<-p3_step(x, m2[i])
    data[i+1]<-x
  }
  return(data)
}

#now mchain
res <- p5_mchain()

#plot as barchart
ggplot() +
  geom_bar(aes(res))
```



We still get a uniform stationary distribution!

Problem 6

Write Acceptance Rejection code to sample (X, Y) uniformly from the unit disc. Use your algorithms with 10^5 samples to estimate $\mathbb{P}(Y \geq 0.7)$. Report your answer as $a \pm b$.

Solution

```
set.seed(1234567)
#acceptance rejection algorithm for sampling from unit disc
ar_unit_disc<-function(){
  a<-FALSE
  while(!a){
    x=2*runif(1)-1
    y=2*runif(1)-1
    a<-(x^2+y^2 <= 1)
  }
  return(c(x,y))
}

ys_ar<-replicate(10^5, ar_unit_disc()[2])

tibble(
  est_prob=mean(ys_ar>=0.7),
  se=sd(ys_ar>=0.7)/sqrt(length(ys_ar))
) %>% kable()
```

est_prob	se
0.09276	0.0009174

The AR estimate for $\mathbb{P}(Y \geq 0.7) = \boxed{0.09276 \pm 0.00092}$.

Problem 7

Write code to draw $(X_1, \ldots, X_n)$ uniformly from directions in $\mathbb{R}^4$. Using 10^4 draws, estimate $\mathbb{P}(X_1 + \cdots + X_4) > 1/2$. Report your answer as $a \pm b$.

Solution

We want to sample uniform over all directions in $\mathbb{R}^4$. TO do this, we need to sample $Z_1, \ldots, Z_4 \sim N(0, 1)$.

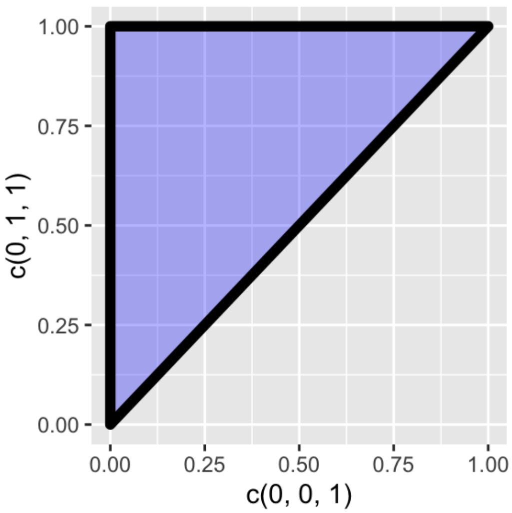
```
set.seed(1234567)
draw_r4<-function(){
  zs<-rnorm(4)
  u_dir<-zs/sqrt(sum(zs^2))
  return(u_dir)
}

r4_unifs<-replicate(10^4, draw_r4())
tibble(
  est_prob=mean(colSums(r4_unifs)>0.5),
  se=sd(colSums(r4_unifs)>0.5)/sqrt(dim(r4_unifs)[2])
) %>% kable()
```

est_prob	se
0.3464	0.0047585

My estimate for $\mathbb{P}(X_1 + \cdots + X_4 > 1/2) = \boxed{0.3464 \pm 0.0048}$.

Problem 8

Suppose that (X, Y) is uniform over the triangle in $\mathbb{R}^2$ with vertices $(0, 0)$, $(0, 1)$, and $(1, 1)$.



a. Write code for one step in a random walk with partially reflecting boundaries that takes as input the current state and two iid standard normal random variables, and attempts to add a move that is a two-dimensional multivariate normal with mean 0, correlation 0, and variance for each component 0.2^2.

b. Use 10 replications of your chain for 10^5 data gathering steps to estimate $\mathbb{E}[X]$.

Solution

a.

```r
set.seed(1234567)
#step function for walking around the triangle
triangle_walk<-function(x, m1, m2){
  if(x[1]+0.2*m1<=x[2]+0.2*m2 & between(x[2]+0.2*m2, 0, 1) & between(x[1]+0.2*m1, 0, 1)){
    #scaled normal
    x<-c(x[1]+0.2*m1, x[2]+0.2*m2)
  }
  else{
    x<-x
  }
  return(x)
}

triangle_mchain<-function(steps=10^5){
  burnin=steps
  datasteps=steps
  #i just cherry picked a starting spot somewhere in the middle of the triangle
  x<-c(0.25, 0.75)
  m1s<-rnorm(datasteps)
  m2s<-rnorm(datasteps)
  #run burnin
  for(i in 1:burnin){
    x<-triangle_walk(x, m1s[i], m2s[i])
  }
  #output
  output_states<-matrix(ncol=2, nrow=datasteps+1)
  output_states[1,]<-x
  #regenerate normals and
  m1s<-rnorm(datasteps)
  m2s<-rnorm(datasteps)
  for(i in 1:datasteps){
    x<-triangle_walk(x, m1s[i], m2s[i])
    output_states[i+1,]<-x
  }
  return(output_states)
}

#replicate chain 10 times and only store x's
res<-replicate(10, triangle_mchain(steps=10^5)[,1])

tibble(
  mean=mean(apply(res, FUN=mean, MARGIN=2)),
  se=sd(apply(res, FUN=mean, MARGIN=2))/sqrt(10)
) %>% kable()
```

mean	se
0.3334294	0.0010063

My estimate for the $\mathbb{E}[X]$ is $\boxed{0.3334 \pm 0.0010}$.

Problem 9

Consider the six dimensional hypersphere $\{(x_1, \ldots, x_6) : x_1^2 + \cdots + x_6^2 \leq 1\}$.

a. Implement a random walk Markov chain whose stationary distribution is uniform over the six dimensional hypersphere, using standard normal random variables to make your moves.

b. Using 10 replications of a chain with $10\,000$ data gathering steps, estimate the probability that a uniform point is at least 0.9 distance from the origin.

Solution

a.

```
step_hypersphere_6<-function(x, m){
  if(sum((x+m)^2)<=1){
    return(x+m)
  }
  else{
    return(x)
  }
}

hypersphere_6_mchain<-function(steps=10^5){
  burnin<-steps
  datasteps<-steps
  #burnin
  ms<-replicate(burnin, rnorm(6))
  x<-c(0,0,0,0,0,0)
  for(i in 1:burnin){
    x<-step_hypersphere_6(x, ms[,i])
  }
  #output and datagathering
  output<-matrix(ncol=6, nrow=datasteps+1)
  output[1,]<-x
  ms<-replicate(burnin, rnorm(6))
  for(i in 1:datasteps){
    x<-step_hypersphere_6(x, ms[,i])
    output[i+1,]<-x
  }
  return(output)
}
```

b.

```
set.seed(1234567)
data<-replicate(10, hypersphere_6_mchain())
lengths<-matrix(nrow=dim(data)[1], ncol=10)
for(i in 1:10){
  lengths[,i]<-apply(data[,,i], FUN=function(x){sqrt(sum(x^2))}, MARGIN=1)
}

tibble(
  means=mean(apply(lengths, FUN=function(x){mean(x>0.9)}, MARGIN=2)),
  se=sd(apply(lengths, FUN=function(x){mean(x>0.9)}, MARGIN=2))/sqrt(10)
) %>% kable()
```

means	se

means	se
0.4847332	0.0060249

My estimate of the probability of a point being at least 0.9 from the origin as being $\boxed{0.4847 \pm 0.0061}$.

Problem 10

Continuing the last problem, modify your chain so that each of the standard normals uses has variance σ^2 instead of 1. Find roughly a value of σ so that the chance of accepting a move in in the chain (again using 10 replications of a chain run with 10^5 data gathering steps) is close to $1/4$ (within a percentage point.)

Solution

```
set.seed(1234567)
#use acceptance_assessment argument flag to specify if you want to assess the acceptance r
ate
hypersphere_6_sigma<-function(steps=10^5, sigma=1, acceptance_assessment=TRUE){
  burnin<-steps
  datasteps<-steps
  #burnin
  ms<-replicate(burnin, rnorm(6, sd=sigma))
  x<-c(0,0,0,0,0,0)
  for(i in 1:burnin){
    x<-step_hypersphere_6(x, ms[,i])
  }
  #output and datagathering
  if(acceptance_assessment==FALSE){
    output<-matrix(ncol=6, nrow=datasteps+1)
    output[1,]<-x
    ms<-replicate(burnin, rnorm(6, sd=sigma))
    for(i in 1:datasteps){
      x<-step_hypersphere_6(x, ms[,i])
      output[i+1,]<-x
    }
    return(output)}
  #assess acceptance
  if(acceptance_assessment==TRUE){
    accept<-rep(0, datasteps)
    ms<-replicate(datasteps, rnorm(6, sd=sigma))
    for(i in 1:datasteps){
      xold<-x
      x<-step_hypersphere_6(x, ms[,i])
      #if x is different from old, accept
      accept[i]<-mean(x!=xold)
    }
    return(accept)
  }
}

accept_vec<-replicate(10, mean(hypersphere_6_sigma(sigma=0.385)))

tibble(
  mean=mean(accept_vec),
  se=sd(accept_vec)/sqrt(10)
) %>% kable()
```

mean	se
0.249327	0.0004051

$\boxed{\sigma = 0.385}$ produces a markov chain that accepts with probability $\boxed{0.24933 \pm 0.00041}$.