Laplace's Law of Succession

Yong

Dec. 8, 2020

1 Motivation

Recently I was re-visiting the conditional prob and here is an experiment drawn from the textbook, with some added ingredients (e.g. some derivations and simulations).

2 Intro

The background is as follows: repeat n $i.i.d$ binomial experiments, each trial with probability p of success; however, p is unknow and is draw from uniform U(0,1). So the overall experiment consists of 2 steps, first draw a random p from U(0,1), and then perform n binomial trials with success rate p. So it can be termed 'uniform-binomial' experiments.
The final goal is to calculate some statistics such as the probability of the total number of successes $S_n = k$ (i.e. $P(S_n = k)$), and the conditional distribution of P (the random var representing p) given k time successes in n trials (i.e. $f(P|S_n = k)$).
If we consider P to be either discrete or continuous, then there are two classes of scenarios. We will first go through the continuous one and then discrete.

3 Continuous uniform-binomial

In this case, P is drawn from U(0,1).

3.1 The distribution of S_n

Conditioned on $P = p$, we have the binomial prob:

$$P(S_n = k|P = p) = \binom{n}{k} p^k (1-p)^{n-k} \tag{1}$$

And the overall probability $P(S_n = k)$ can thus be calculated by the average conditional probabilities:

$$P(S_n = k) = \int P(S_n = k|P = p) f(p) dp = \int_0^1 \binom{n}{k} p^k (1-p)^{n-k} f(p) dp = \int_0^1 \binom{n}{k} p^k (1-p)^{n-k} dp = \frac{1}{n+1} \tag{2}$$

which is uniform on $[0, n]$.

3.2 The conditional distribution of P given that $S_n = k$

Here we aim to find $P(P|S_n = k)$, i.e. after observing $S_n = k$), what's the distribution of P? We use the Bayes' rule.

$$P(P \in dp|S_n = k) = \frac{P(S_n = k|P \in dp) * P(P \in dp)}{P(S_n = k)} \tag{3}$$

We already know that $P(S_n = k|P \in dp) = \binom{n}{k} p^k (1-p)^{n-k} dp$. And the marginal P is uniform on (0,1), i.e., $P(P \in dp) = 1$. Also, $P(S_n = k) = 1/(n+1)$ by 1. Thus, the above prob reduces to:

$$P(P \in dp|S_n = k) = \frac{P(S_n = k|P \in dp) * P(P \in dp)}{P(S_n = k)} = (n+1) * \binom{n}{k} p^k (1-p)^{n-k} dp \tag{4}$$

As a result:

$$f(p|S_n = k) = (n+1) * \binom{n}{k} p^k (1-p)^{n-k} \tag{5}$$

A close look at the above conditional density finds that, P(P=p$|S_n = k$) is a beta pdf with parameters k+1 and n-k+1, i.e. P=p$|S_n = k \sim beta(k+1, n-k+1)$.

3.3 Prob of next trial is a success, after observing k successes in n trials

Given n trials has produced k successes, what's the prob that the next trail is a success?
First, by the independent assumption, P(next trial is a success $|S_n = k, P = p) = p$.
By the average conditional probability principle, we have:
P(next trial is a success$|S_n = k) = \int_0^1$ P(next trial is a success$|S_n = k, P = p) f(p|S_n = k) dp = \int_0^1 p f(p|S_n = k) dp$. The last term is the expectation of P, i.e. E(P$|S_n = k$). According to 5, we know P=p$|S_n = k \sim beta(k+1, n-k+1)$, and as per the beta statistics, we have:

$$E(P|S_n = k) = \frac{k+1}{n+2} = P(next\ trial\ is\ a\ success|S_n = k) \tag{6}$$

4 Discrete uniform-binomial

The setting is similar to the continuous case; the only difference is that, now P can only take discrete values in [0,1]. The scene is, suppose there are N+1 boxes labelled b=0,1,2,...,N, and Box b contains b black and N-b white balls. A box is picked randomly, then n balls are drawn at random with replacement from the picked box. S_n denotes the total number of black balls in the n trials. We perform similar analysis to the continuous case to find:
(i) the distribution of S_n
(ii) the conditional dist P(p$|S_n = k$)

4.1 the distribution of S_n

We see P can take n+1 values uniformly from the set 0,1/N,2/N,...,1, each with probability 1/(N+1), i.e. P(P=p)=1/(N+1). Then for a given $P = p$, we have the binomial event $S_n = k|P = p \sim B(n,p)$:

$$P(S_n = k|P = p) = \binom{n}{k} p^k (1-p)^{n-k} \tag{7}$$

Using the average conditional probability principle, we get $P(S_n = k)$:

$$P(S_n = k) = \sum_p P(S_n = k|P = p) * P(P = p) = \sum_p \binom{n}{k} p^k (1-p)^{n-k} * \frac{1}{N+1} = \binom{n}{k} \frac{1}{(N+1)N^n} \sum_{b=0}^{N} b^k (N-b)^{n-k} \tag{8}$$

We can further explore the limiting distribution of $P(S_n = k)$:

$$P(S_n = k) = \sum_p \binom{n}{k} p^k (1-p)^{n-k} * \frac{1}{N+1} = \binom{n}{k} \sum_p p^k (1-p)^{n-k} * \frac{1}{N+1} \tag{9}$$

Notice that, the summation $\sum_p p^k (1-p)^{n-k} * \frac{1}{N+1}$ is $\sum_p p^k (1-p)^{n-k} dp$, it becomes an integral when $N \to \infty$, i.e.: $\sum_p p^k (1-p)^{n-k} * \frac{1}{N+1} = \int_0^1 p^k (1-p)^{n-k} dp$, which is a beta integral, i.e. $\sum_p p^k (1-p)^{n-k} * \frac{1}{N+1} =$

$\int_0^1 p^k(1-p)^{n-k}dp = B(k+1, n-k+1)$. The beta integral can be calculated using the gamma function Γ:

$$B(k+1, n-k+1) = \frac{\Gamma(k+1)\Gamma(n-k+1)}{\Gamma(k+1+n-k+1)} \tag{10}$$

And the gamma function has the property $\Gamma(k+1) = k!$, which simplifies the above equation to be:

$$B(k+1, n-k+1) = \frac{\Gamma(k+1)\Gamma(n-k+1)}{\Gamma(k+1+n-k+1)} = \frac{1}{\binom{n}{k}} * \frac{1}{n+1} \tag{11}$$

Therefore, when $N \to \infty$,

$$P(S_n = k) \to \frac{1}{n+1} \tag{12}$$

which demonstrates that, the limiting distribution of S_n is uniform on the set 0,1,2,..., n-1,n.

4.2 the conditional pmf P(p$|S_n = k$)

Again, using Bayes' rule:

$$P(P = p|S_n = k) = \frac{P(S_n = k|P = p) * P(P = p)}{P(S_n = k)} = \frac{\binom{n}{k}p^k(1-p)^{n-k} * \frac{1}{N+1}}{\sum_p \binom{n}{k}p^k(1-p)^{n-k} * \frac{1}{N+1}} = \frac{p^k(1-p)^{n-k}}{\sum_p p^k(1-p)^{n-k}} \tag{13}$$

If $N \to \infty$, we use the result in 12 and get:

$$P(P = p|S_n = k) = \frac{P(S_n = k|P = p) * P(P = p)}{P(S_n = k)} = \frac{\binom{n}{k}p^k(1-p)^{n-k} * \frac{1}{N+1}}{\frac{1}{n+1}} = \frac{n+1}{N+1}\binom{n}{k}p^k(1-p)^{n-k} \tag{14}$$

In a Bayesian setting, you might also want to reason about p, given $S_n = k$, i.e. which box has bee picked up based on $p = b/N$.

5 Simulation

The exhaustive approach to simulate the experiments is to first generate p from U(0,1), and then repeat n binomial trials.

5.1 S_n

5.1.1 A fake simulation

A cheating simulation is to use directly formula 2. Essentially, we need to evaluate the integration in 2 - this can be done using Monte-Carlo method, with uniform DGP. The process of generating the distribution of S_n is as follows (Algo.**??**):

Algorithm 1: Find $f(S_n)$

specify n, *e.g.* n=1000
for $k = 0 : n$ **do**
  generate p from U(0,1)
  sum = 0
  for 1:1000 **do**
    sum += $\binom{n}{k}p^k(1-p)^{n-k}$ (evaluate integrant)
  end for
  $P(S_n = k) = \frac{1}{1000}sum$
  (MC: e.g. evaluate the integrant 1000 times to obtain 1 MC value)
end for
plot [k,$P(S_n = k)$]

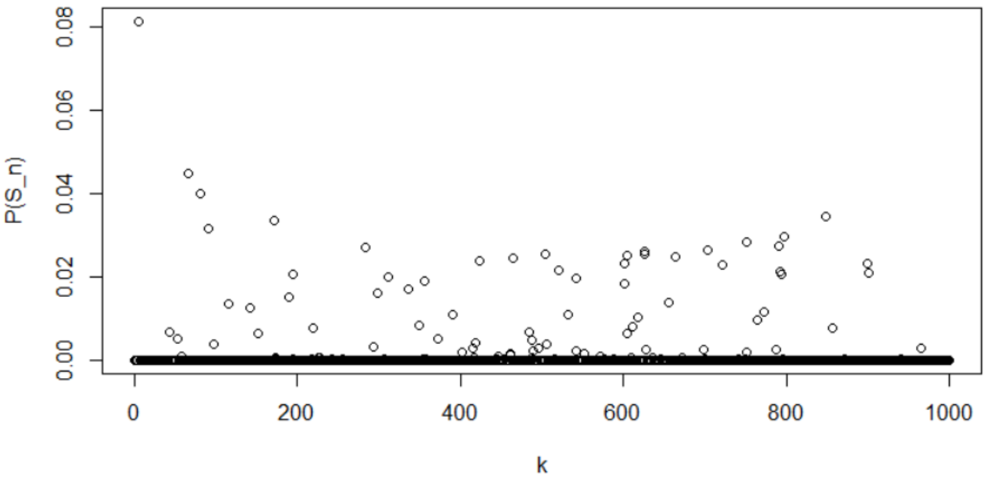

Figure 1: Simulation: distribution of S_n

If we chose $n = 1e3$, from 2 we see that, theoretically $P(S_n)$ would be uniform with pmf $\frac{1}{n+1} \approx 0.001$. The simulation results are show in Fig.1

However, as implied before, this is a spurious simulation because in doing the simulation, we have referred to the theoretical result 2 - the only useful thing we are doing here is using MC to evaluate integral.

5.1.2 a true simulation

So we have to perform the exhaustive experiments, i.e. first sample $p \sim U(0,1)$, then sample $X \sim B(n,p)$ (1e3 samples for each p value), and repeat this 1e3 times for allow multiple p values to have chances to be sampled. The experimental design and computation is show in following: (Algo.2):

Algorithm 2: Find $f(S_n)$

```
specify n, e.g. n=1000
k = 0:n
initialize tabular histogram counter as a function of k
for i = 1 : 1000 do
  sample p from U(0,1)
  sample 1000 X_is from B(n,p)
  update histogram counter
end for
plot [k,histogram counter]
```

The simulation results are shown in Fig.2. We can see the frequencies for each k value fluctuates around 1e3, which is roughly uniform.

5.2 $f(p|S_n = k)$

Then we move on to find the conditional distribution $f(p|S_n = k)$. We can follow the the same path of generating p and binomial sample $X \sim B(n,p)$. The difference here is, this time we need to record p and S_n at the same time to form a table, which is the joint distribution of P and S_n. I sampled p 1e3 ties, and for each value of p, I generated 1e3 binomial samples. As the joint distribution is more informative than marginal distributions, we can derive both marginal distributions form the joint distribution (NB: not the other way round). The exhaustive simulation is described in Algo.3 :

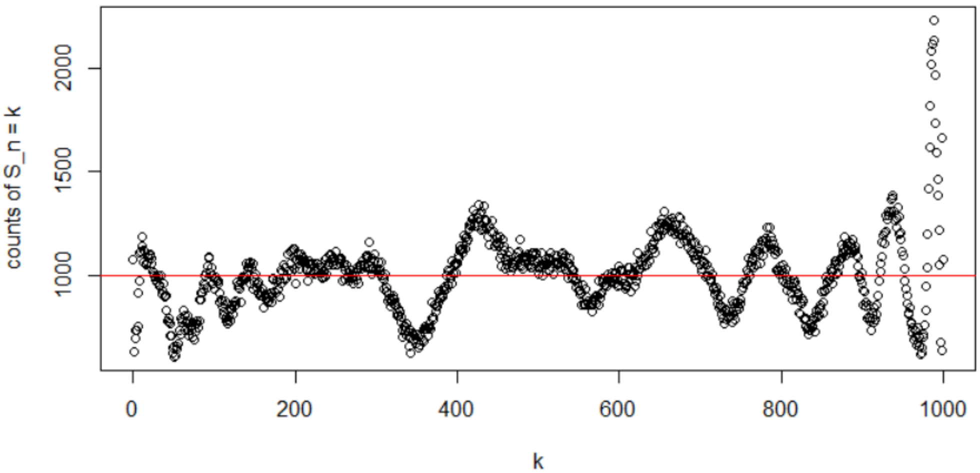

Figure 2: Simulation: distribution of S_n

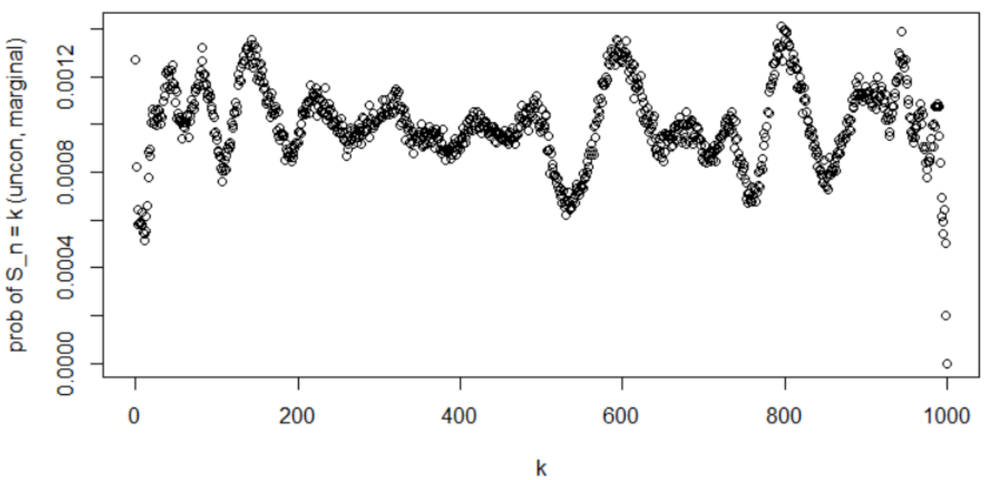

Figure 3: $S_(n)marginal$

Algorithm 3: Find marginals $f(S_n)$, $f(p)$ and conditional $f(p|S_n = k)$

---

specify n, *e.g.* n=1000
k = 0:n
initialize a table (matrix) M, in which row number ranges from 1 to 1000 (corresponding to p), column number ranges from 1 to 1001 (corresponding to k)
for $i = 1 : 1000$ **do**
  sample p from U(0,1)
  sample 1000 X_is from B(n,p)
  count the pair $(k, freq)$ in X_is)
  update the corresponding numbers in the row $P = p$ of M
end for
calculate marginal and conditional distributions by summation and normalisation

---

The resulted simulation-based marginal distributions are shown in Fig.3 and Fig.4.

Again, these marginal probabilities match our theoretical expectations: their long-term averages are expected to be $P(S_n = k) = \frac{1}{n+1}$ and $1/n$ respectively.

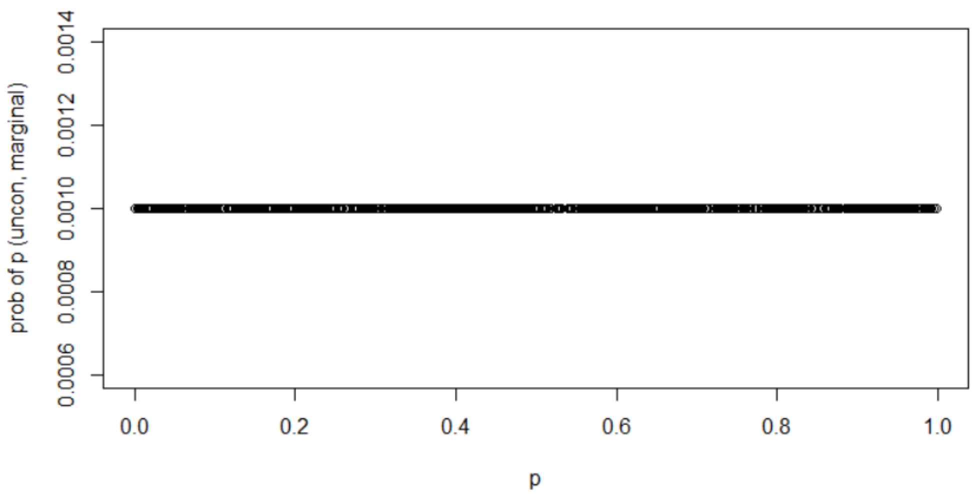

Figure 4: *pmarginal*

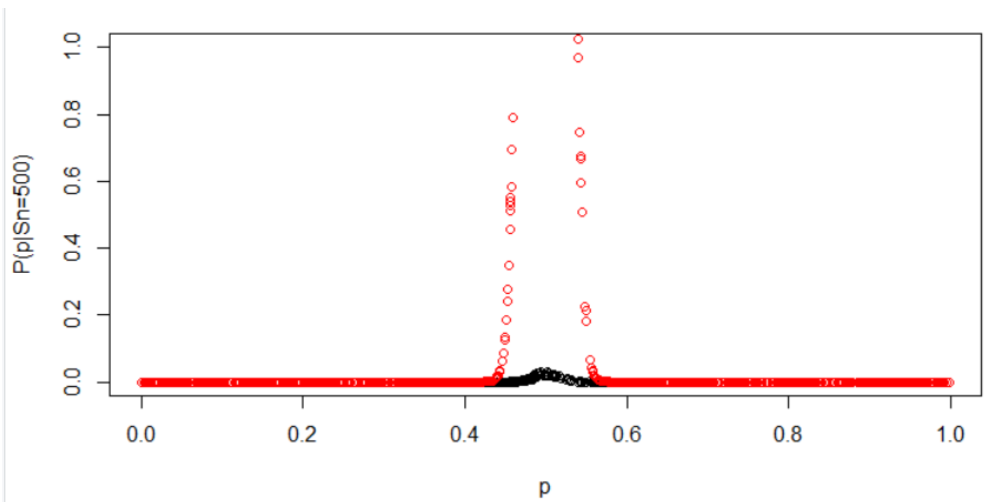

Figure 5: $P(p|S_n = 500)$

So now the conditional profiles can be worked out based on the joint distribution. For example, if we are interested in $P(p|S_n = 500)$, we have a cut-off line in the table at $S_n = 500$, and extract the cross-sectional p series values and then normalize it by $P(S_n = 500)$ (which is essentially the sum of the sliced p series values), we arrive at Fig.

As an comparison, theoretical analysis in Eq.5 gives a distribution over p: $P(p|S_{1000} = 500) = (1000 + 1) * \binom{1000}{500} p^{500}(1-p)^{1000-500}$, which is plotted in red in Fig.5 as an comparison. The theoretical and simulation-based conditional distributions agree that, if 500 successes have been observed out of 1000 independent binomial trials, then the most probable sampled probability p in B(1000,p) is 0.5. The absolute value, however, diverges due to the limited number of data points simulated. In the discrete setting, this means the most probable that, after we have repeated the Binomial trials 1000 times without any knowledge of which box has been picked up in the first step (i.e. p), given the results, it would be reasonable to estimate that, the most probable one that has been picked in the first step would be the one with half black and half white balls.

5.3 Alternative approach for data generation

Instead of using the 2-stage sequentially dependent data generation approach (i.e. first sample p then generate $X_i \sim B(n,p)$), which is exhaustive, there exists an alternative simple data generation approach that directly generate data pairs (p, S_n) from the joint distribution $f(p, S_n)$:

$$p = U(0,1),\ and\ S_n = \sum_{i=1}^{n} I(U_i < p) \tag{15}$$

where I is the indicator variable with value 1 if the condition satisfied. So S_n is a sum of n independent indicator variables (events), which is the total number of successful events in n trials. And each trial has probability p to succeed. We thus have the following algorithm: (Algo.4):

Algorithm 4: Find marginals $f(S_n)$, $f(p)$ and conditional $f(p|S_n = k)$

---

specify n, *e.g.* n=1000
k = 0:n
initialize a table (matrix) M, in which row number ranges from 1 to 1000 (corresponding to p), column number ranges from 1 to 1001 (corresponding to k)
for $i = 1 : 1000$ **do**
  sample p from U(0,1)
  sample 1000 X_is from B(n,p)
  $S_n = \sum_{i=1}^{n} I(X_i < p)$
  update (+1) the corresponding cell (p, k) of M
end for
calculate marginal and conditional distributions by summation and normalisation

---

The results are shown in Fig.6, Fig.7, and Fig.**??**.

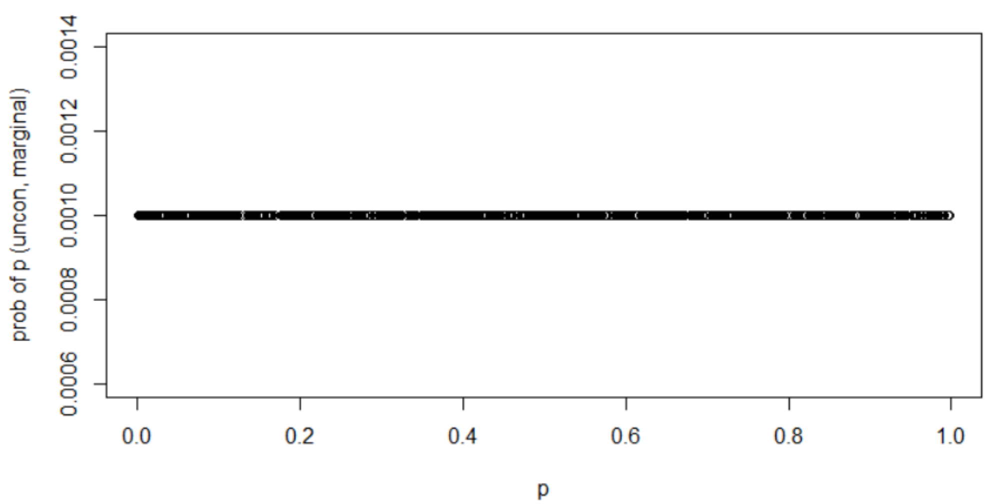

Figure 6: *pmarginal*

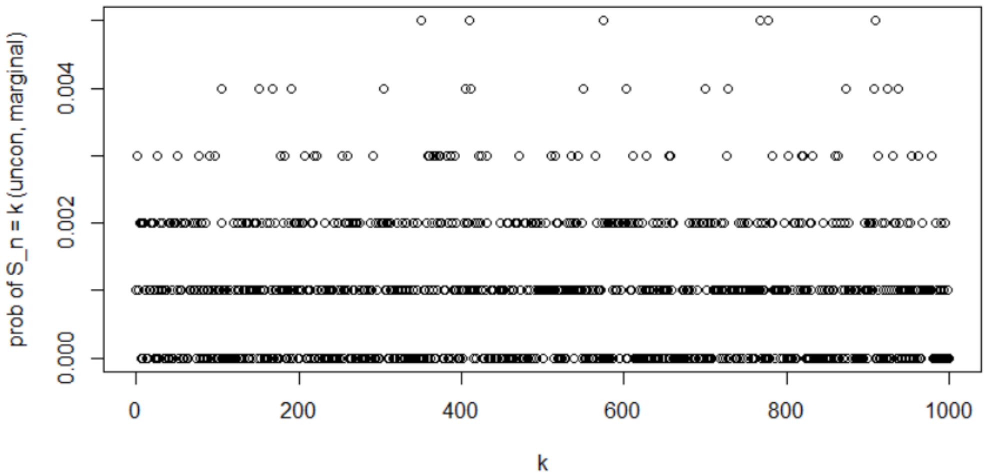

Figure 7: $S_(n) marginal$

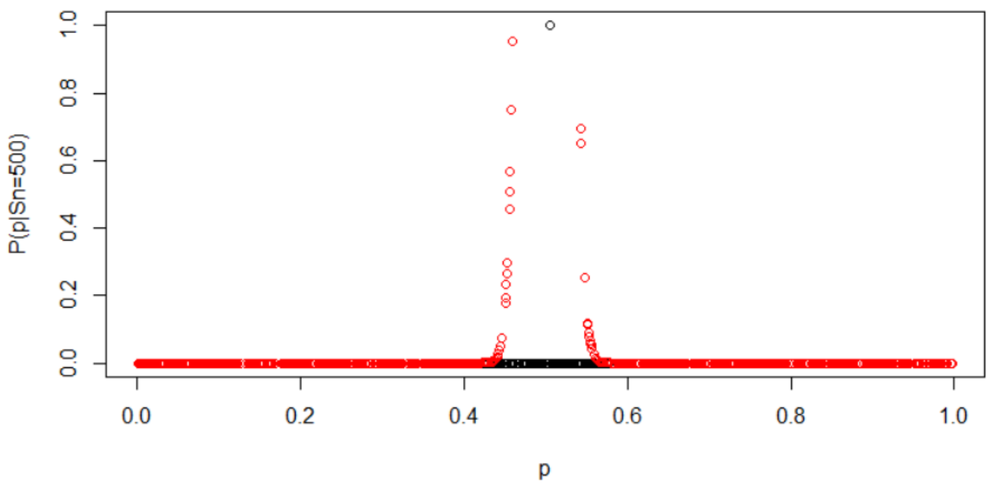

Figure 8: $P(p|S_n = 500)$

6 conclusion

Sorry, hurry to go home for dinner so will write up next time.

7 Reference

$< Probability >$ the textbook.
All typos and errors are mine.

Appendix A Appendice: R codes used in simulations

A.1 Code 1

```
if(!require("pracma")){install.packages("pracma");require("pracma")}

##### Sn spurious ####
n=1e3
MC_no = 1e3
p_vec=vector()
Sn_k_vec=vector()
for(k in 1:n){
  p=runif(1, min = 0, max = 1)
  p_vec = c(p_vec, p)
  sum=0
  for (j in 1:MC_no) {
    sum = sum + nchoosek(n, k)*p^k*(1-p)^(n-k)
  }
  Sn_k = sum/MC_no
  Sn_k_vec=c(Sn_k_vec,Sn_k)
}
plot(1:n,Sn_k_vec, xlab = "k", ylab = "P(S_n)")


##### Sn real ####
n=1e3
k=0:n
# p_vec=seq(0,1,0.01)

Sn_k_vec= rep(0,length(k))
for(i in 1:1e3){
  p=runif(1, min = 0, max = 1)
  samples=rbinom(1e3, size=n, prob=p)
  histogram=hist(samples,breaks=0:(n+1))
  histogram$breaks
  Sn_k_vec = Sn_k_vec + histogram$counts
}
plot(k, Sn_k_vec, xlab = "k", ylab = "counts of S_n = k")
abline(h=1000, col="red")
```

Listing 1: R codes for Fig.1 and Fig.2

A.2 Code 2

```
##### f(p|Sn) ####
set.seed(111)
n=1e3
k=0:n

p_vec = rep(NA, 1e3)
p_marginal_vec = rep(NA, length(p_vec))
Sn_k_vec= rep(0,length(k))
p_Sn_matrix = matrix(nrow = length(p_vec), ncol = length(Sn_k_vec))
for(i in 1:1e3){
  p=runif(1, min = 0, max = 1)
  p_vec[i] = p
  samples=rbinom(1e3, size=n, prob=p)
  histogram=hist(samples,breaks=0:(n+1))
  histogram$breaks
  Sn_k_vec = Sn_k_vec + histogram$counts
  p_Sn_matrix[i,] = histogram$counts
}
p_marginal_vec = rowSums(p_Sn_matrix)/(sum(colSums(p_Sn_matrix)))
plot(k, Sn_k_vec/sum(Sn_k_vec), xlab = "k", ylab = "prob of S_n = k (uncon, marginal)")
plot(p_vec, p_marginal_vec, xlab = "p", ylab = "prob of p (uncon, marginal)")
# P(p|Sn=500)
fixed_k=500
con_prob = p_Sn_matrix[,fixed_k+1]/sum(p_Sn_matrix[,fixed_k+1])
summary(con_prob)
plot(p_vec, con_prob, xlab = "p", ylab = "P(p|Sn=500)", ylim = c(0,1))
# theoretical
theoretical_con_prob_vec = (n+1)*nchoosek(n, fixed_k)*p_vec^fixed_k*(1-p_vec)^(n-fixed_k)
points(p_vec, theoretical_con_prob_vec, col="red")
summary(theoretical_con_prob_vec)
```

Listing 2: R codes for Fig.3, Fig.4 and Fig.5

A.3 Code 3

```
#### alternative data generation approach (sampling from joint distribution)
set.seed(111)
n=1e3
k=0:n

p_vec = rep(NA, 1e3)
Sn_k_vec = rep(0,length(k))
p_Sn_matrix = matrix(data=0, nrow = length(p_vec), ncol = length(Sn_k_vec))
for(i in 1:1e3){
  p=runif(1, min = 0, max = 1)
  p_vec[i] = p
  samples = runif(n, min = 0, max = 1)
  Sn = sum(samples<p)
  Sn_k_vec[Sn] = Sn_k_vec[Sn] + 1
  p_Sn_matrix[i,Sn] = p_Sn_matrix[i,Sn] + 1
}
plot(k, Sn_k_vec/sum(Sn_k_vec), xlab = "k", ylab = "prob of S_n = k (uncon, marginal)")
p_marginal_vec = rowSums(p_Sn_matrix)/(sum(colSums(p_Sn_matrix)))
plot(p_vec, p_marginal_vec, xlab = "p", ylab = "prob of p (uncon, marginal)")
# P(p|Sn=500)
fixed_k=500
con_prob = p_Sn_matrix[,fixed_k+1]/sum(p_Sn_matrix[,fixed_k+1])
summary(con_prob)
plot(p_vec, con_prob, xlab = "p", ylab = "P(p|Sn=500)", ylim = c(0,1))
# theoretical
theoretical_con_prob_vec = (n+1)*nchoosek(n, fixed_k)*p_vec^fixed_k*(1-p_vec)^(n-fixed_k)
points(p_vec, theoretical_con_prob_vec, col="red")
summary(theoretical_con_prob_vec)
```

Listing 3: R codes for Fig.6, Fig.7, and Fig.8