Assignment04

Zhen Liu

2022-10-23

1.Probability theory

1.1 Rules of probability

Q1

P({a}) = 0.4 P({b}) = 0.1 P({c})=0.5

Q2

rule 1: P(∅) = 0, since q ∈ [0,1], hence P({0}) = 1−q ≥ 0, P({1}) = q ≥ 0, P({0,1}) = 1 satisfy that P(A) ≥ 0 for any event A ∈ E rule 2: Ω = {0,1} and P({0,1}) = 1 which satisfy P(Ω) = 1 for sample space Ω rule 3: P({0}) = 1−q, P({1}) = q, P({0,1}) = 1 = P({0}) + P({1}) which satisfy rule 3.

1.2 Deriving new properties from the rules of probability

Q1 Union of a finite sequence of disjoint events

Based on lecture 10 rule 3 proof,

a finite probability sapce is triple (Ω,E,p) and E = {A ⊆ Ω} Ω is finite,

$$P(\cup_{i=1}^{\infty} A_i) = \sum_{i=1}^{\infty}(\sum_{i=1}^{n} p_i \mathbf{1}A_i(w_i))$$

hence

$$P(\cup_{i=1}^{n} A_i) = \sum_{i=1}^{n} p_i \mathbf{1}A_i(w_i) = \sum_{i=1}^{n} \mathbf{p}(A_i)$$

Q2 Probability of a complement

Based on rule 2 and rule 3

$$P(S^c) + P(S) = 1.$$

hence

$$P(S^c) = 1 - P(S)$$

Q3 The union bound

P(A∪B)=P(A)+P(B)−P(A∩B)≤P(A)+P(B)

$$\text{It is possible that } P(S_{i-1} \cap S_{i-2}) > 0 \text{ hence} P(\cup_{i=1}^{\infty} S_i) < \sum_{i=1}^{\infty} P(S_i)$$

Q4

$$A \cup B = (A \cap B) \cup (A \cap B^c) \cup (A^c \cap B)$$

$$P(A) = P(A \cap B) + P(A \cap B^c)$$

$$P(B) = P(A \cap B) + P(B \cap A^c)$$

Based on rule 3

$$P(A \cup B) = P[(A \cap B) \cup (A \cap B^c) \cup (A^c \cap B)] = P(A \cap B) + P(A \cap B^c) + P(A^c \cap B) = P(A) + P(B) - P(A \cap B)$$

2. Finite probability spaces

2.1 Sampling with replacement

Q1

$$P(A_{z,22}) = \binom{22}{z} * (0.3)^z * (0.7)^{22-z}$$

Q2

```
prob_red_spheres<-function(z){
  pro<- choose(22,z)*(0.3^z)*(0.7^(22-z))
  return(pro)
}
```

```
prob_red_spheres(10)
```

```
## [1] 0.05285129
```

Q3

```
library(tidyverse)
```

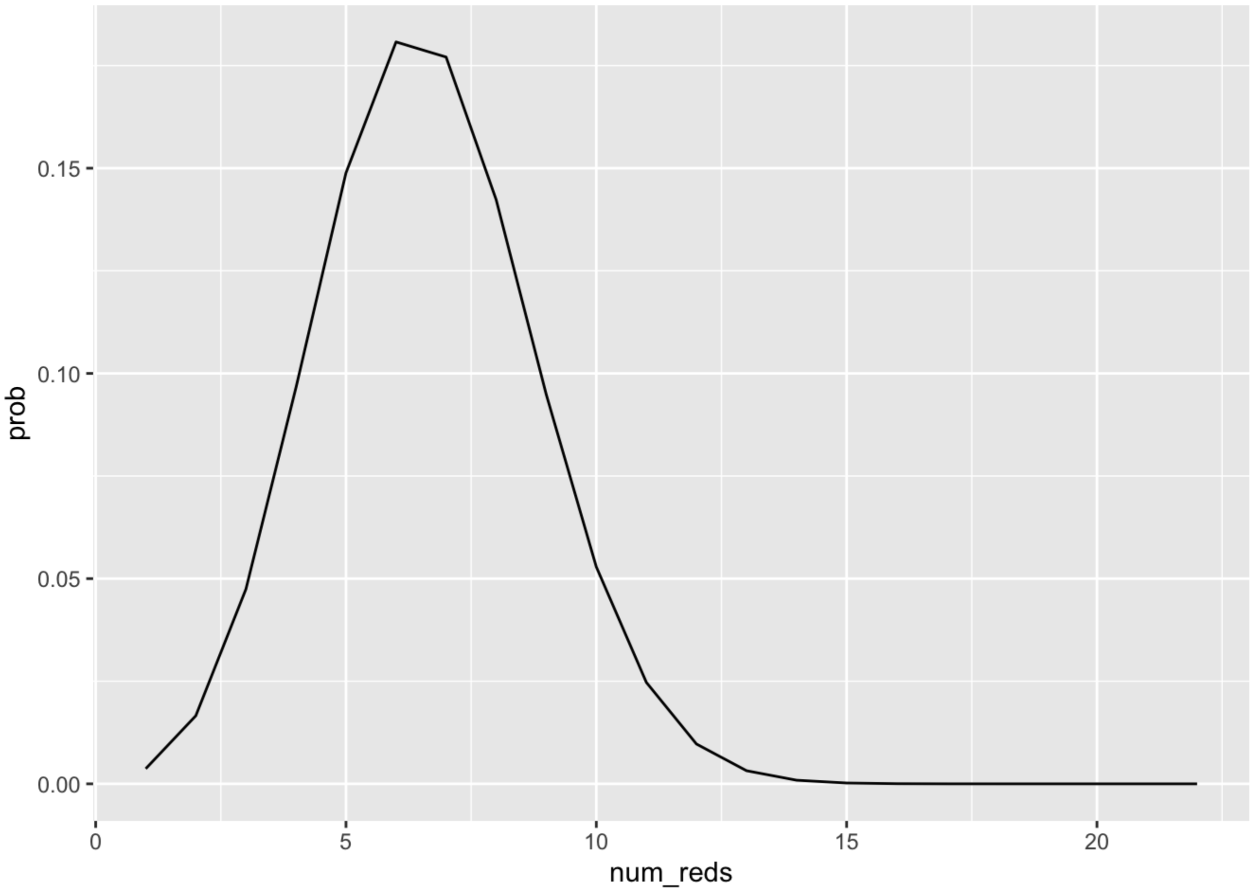
```
## ── Attaching packages ─────────────────────────────────────── tidyverse 1.3.2 ──
## ✔ ggplot2 3.3.6      ✔ purrr   0.3.4
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
```

```
num_reds<- c(seq(22))
prob<-c(prob_red_spheres(num_reds))


prob_by_num_reds<- data.frame(num_reds,prob)

prob_by_num_reds %>% head(3)
```

```
##   num_reds        prob
## 1        1 0.003686403
## 2        2 0.016588812
## 3        3 0.047396606
```

Q4

```
library(ggplot2)
ggplot(prob_by_num_reds,aes(x=num_reds,y=prob)) + geom_line()
```



Q5

```
sample(10, 22, replace=TRUE)
```

```
##  [1]  3  5  6 10  5  6  4  3  9  5  6  4  8  6  1 10  8  4  7  2  6  2
```

```
## Setting the random seed just once
set.seed(0)
for(i in 1:5){
print(sample(100,5,replace=FALSE))
# The result may well differ every time
}
```

```
## [1] 14 68 39  1 34
## [1] 87 43 14 82 59
## [1] 51 97 85 21 54
## [1] 74  7 73 79 85
## [1]  37  89 100  34  99
```

```
## Resetting the random seed every time
```

```
for(i in 1:5){
set.seed(1)
print(sample(100,5,replace=FALSE))
# The result should not change
}
```

```
## [1] 68 39  1 34 87
## [1] 68 39  1 34 87
## [1] 68 39  1 34 87
## [1] 68 39  1 34 87
## [1] 68 39  1 34 87
```

```
num_trials<-1000 # set the number of trials
set.seed(0) # set the random seed
sampling_with_replacement_simulation<-data.frame(trial=1:num_trials) %>%
mutate(sample_balls = map(.x=trial, ~sample(10,22, replace = TRUE)))
# generate collection of num_trials simulations
```

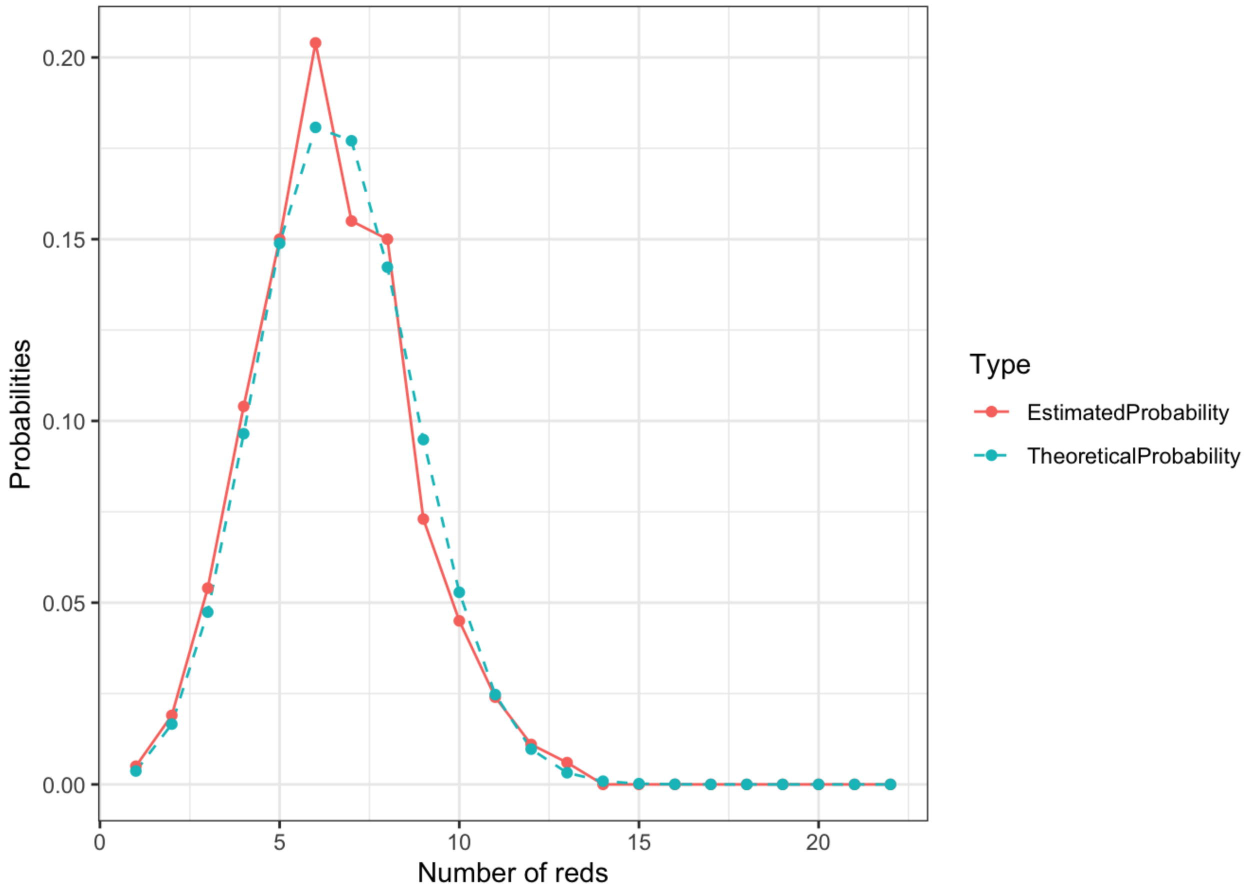
```
sampling_with_replacement_simulation<-sampling_with_replacement_simulation%>%
  mutate(num_reds = map_dbl(.x=sample_balls,~sum(.x<=3)))
```

Q6

```
num_reds_in_simulation<-sampling_with_replacement_simulation %>%
pull(num_reds)
# we extract a vector corresponding to the number of reds in each trial
prob_by_num_reds<-prob_by_num_reds %>%
mutate(predicted_prob=map_dbl(.x=num_reds,~sum(num_reds_in_simulation==.x))/num_trials)
# add a column which gives the number of trials with a given number of reds
```

Q7

```
prob_by_num_reds %>%
rename(TheoreticalProbability=prob, EstimatedProbability=predicted_prob) %>%
pivot_longer(cols=c("EstimatedProbability","TheoreticalProbability"),
names_to="Type",values_to="count") %>%
ggplot(aes(num_reds,count)) +
geom_line(aes(linetype=Type, color=Type)) + geom_point(aes(color=Type)) +scale_linetype_man
ual(values = c("solid", "dashed"))+
theme_bw() + xlab("Number of reds") + ylab("Probabilities")
```



2.2 Sampling without replacement

Q1

1. First set a random seed;

```
set.seed(0)
```

2.

```
num_trials_no<-1000
sample(100,10,replace = FALSE)
```

```
##  [1]  14  68  39   1  34  87  43 100  82  59
```

3.

```
sampling_without_replacement_simulation<-data.frame(trial=1:num_trials_no) %>%
mutate(sample_collections = map(.x=trial, ~sample(100,10, replace = FALSE)))
```

4.

```
sampling_without_replacement_simulation<-sampling_without_replacement_simulation%>%
  mutate(reds = map_dbl(.x=sample_collections,~sum(.x<=50)))%>%
  mutate(blues = map_dbl(.x=sample_collections,~sum(.x>50 & .x<=80)))%>%
  mutate(greens = map_dbl(.x=sample_collections,~sum(.x>80 & .x<=100)))
```

5.

```
missing_values<-pmin(sampling_without_replacement_simulation$reds,sampling_without_replacem
ent_simulation$blues,sampling_without_replacement_simulation$greens)
```

6.

```
missing_values%>%
  table(missing_values==0)
```

```
##
## .   FALSE TRUE
##   0     0  124
##   1   365    0
##   2   393    0
##   3   118    0
```

```
portion<- 125/1000
print(portion)
```

```
## [1] 0.125
```