Assignment 4

Insert My Name

2022-10-19

```
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()
```

1. Probability theory

Q1:P({a})=0.5 p({b})=0.1 p({c})=0.4 # 2. Finite probability spaces

2.1 (Q1)

@ref(eq:binom) \[\begin{equation} P\left(A\right) = \binom{22}{z} \left(\frac{3}{10}\right)^{z}\left(\frac{7}{10}\right)^{22-z} \end{equation}\]

2.1 (Q2)

```
prob_red_spheres<-function(z){
  k<-z
  y<-1
  x<-1
  up<-22
  w<-22-z
  h<-1
  while(up>=1){
    x<-x*up
    up=up-1
  }
  while(w>=1){
    h<-w*h
    w=w-1
  }
  while(k>=1){
    y<-y*k
    k=k-1
```

```
  }
  c<-x/(y*h)
  left<-round(0.3,1)
  right<-round(0.7,1)
  leftx<-1
  rightx<-1
  for(i in seq(1,z)){
    leftx = left*leftx
  }
  for(i in seq(1,22-z)){
    rightx = right*rightx
  }
  return(c*leftx*rightx)
}
prob_red_spheres(10)
```

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

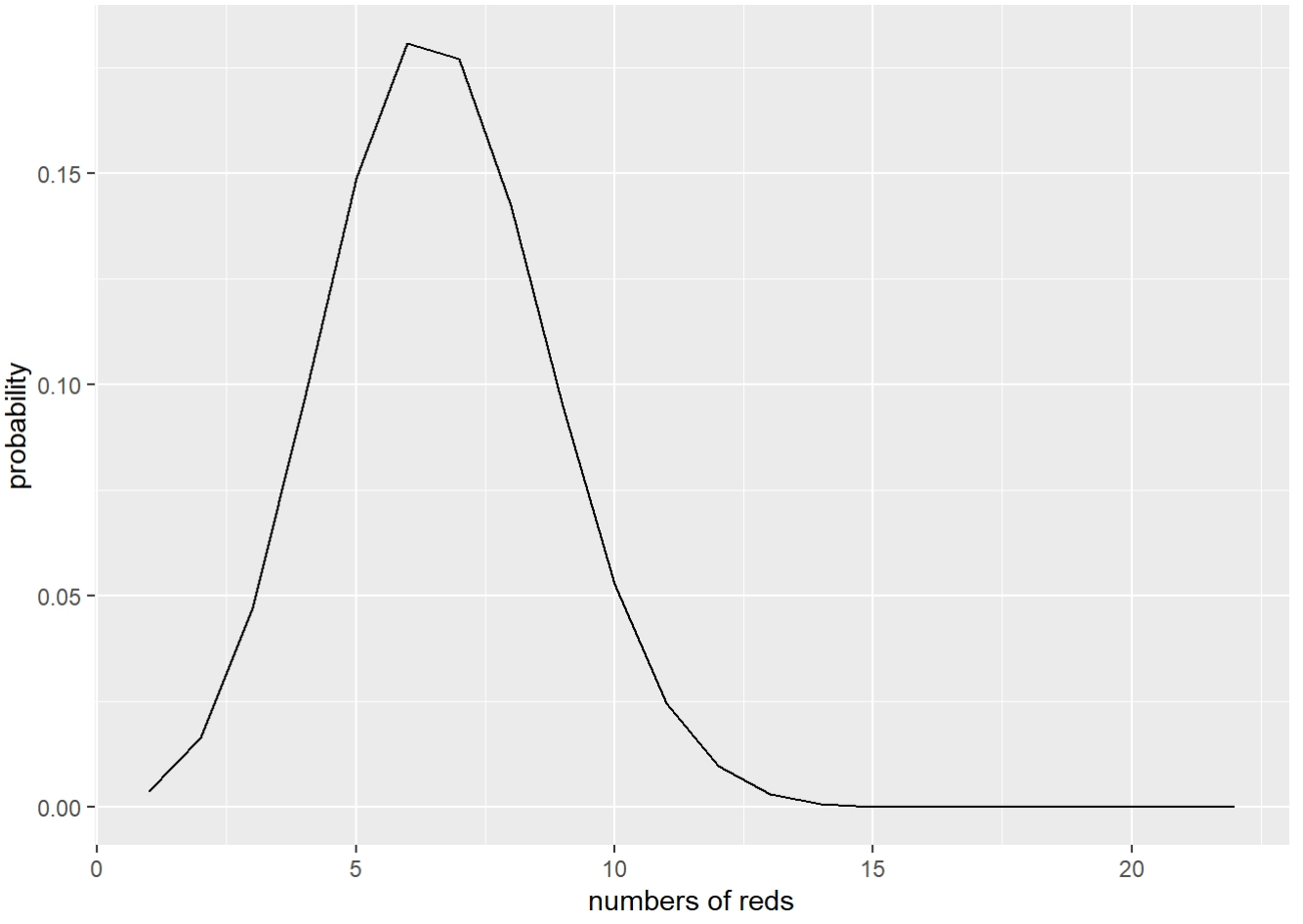
2.1 (Q3)

```
ok <- function(x){
  return(map_dbl(x,prob_red_spheres))
}
num_reds<-seq(1,22,by=1)
prob<-ok(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
```

2.1 (Q4)

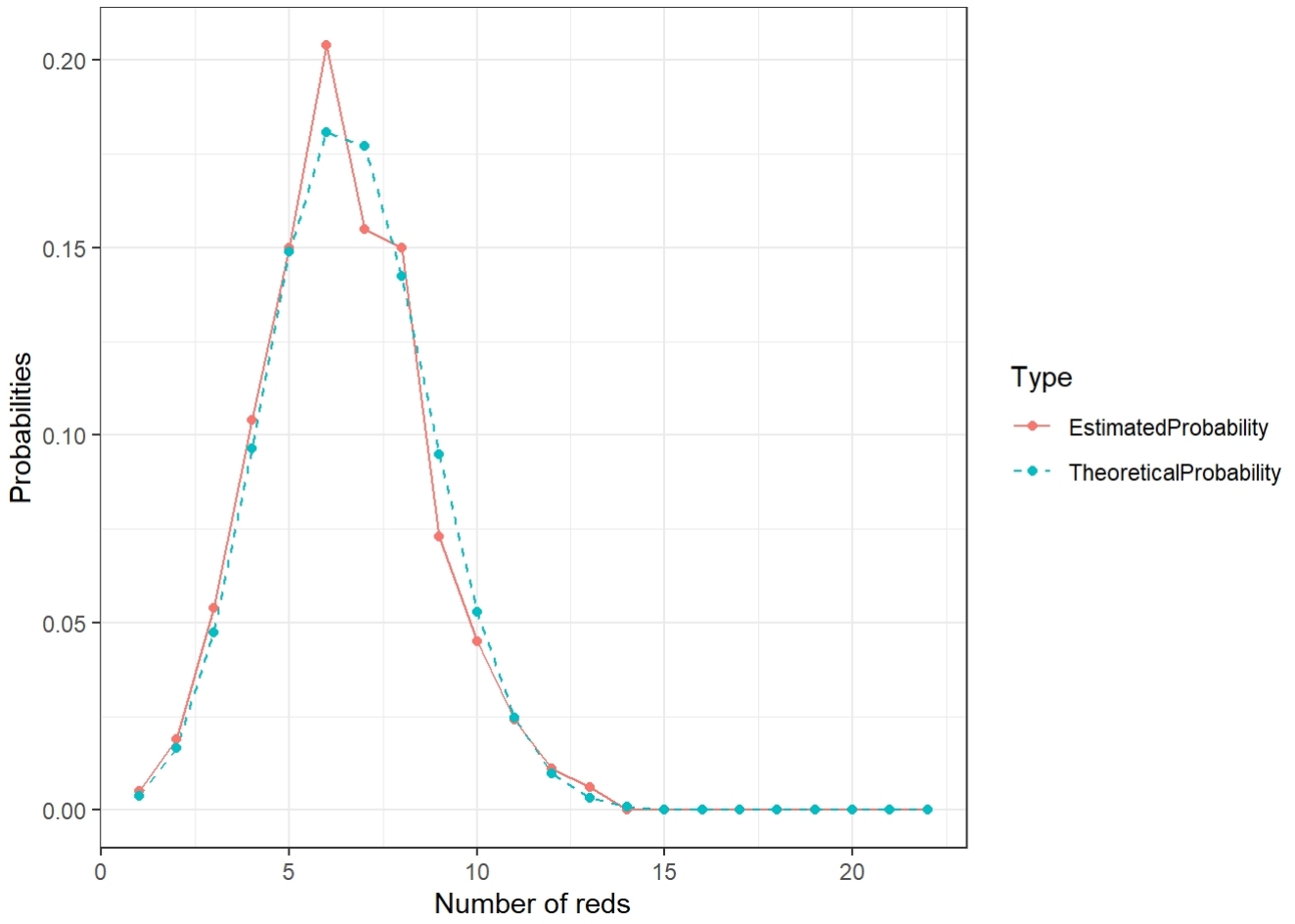
```
univar_plot <- ggplot(data=prob_by_num_reds, aes(x=num_reds,y=prob)) + xlab("numbers of reds")
univar_plot+ylab("probability")+geom_line()
```



2.1 (Q5)&(Q6)

```
library(dplyr)
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
suum<-function(x){
  w<-0
  for(i in x){
    if(i<=3){
      w<-w+1
    }
  }
  return(w)
}
sampling_with_replacement_simulation<-sampling_with_replacement_simulation%>%mutate(num_reds=map_dbl(sample_balls,suum))
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)
```

```
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_manual(values = c("solid", "dashed"))+
theme_bw() + xlab("Number of reds") + ylab("Probabilities")
```



2.2 Sampling without replacement

Q1

```
num_trials<-10 # set the number of trials
set.seed(0) # set the random seed
sampling_out_replacement_simulation<-data.frame(trial=1:num_trials) %>%
mutate(sample_balls = map(.x=trial, ~sample(100,10,replace = FALSE)))
suum<-function(x){
  w<-0
  for(i in x){
    if(i<=50){
```

```
      w<-w+1
    }
  }
  return(w)
}
suumb<-function(x){
  w<-0
  for(i in x){
    if(i>=50 & i<=80){
      w<-w+1
    }
  }
  return(w)
}
suumg<-function(x){
  w<-0
  for(i in x){
    if(i>=80&i<=100){
      w<-w+1
    }
  }
  return(w)
}
sampling<-sampling_out_replacement_simulation%>%mutate(num_reds=map_dbl(sample_balls,suum))
numr<-sampling %>%pull(num_reds)
samplingb<-sampling_out_replacement_simulation%>%mutate(num_b=map_dbl(sample_balls,suumb))
numb<-samplingb %>%pull(num_b)
samplingg<-sampling_out_replacement_simulation%>%mutate(num_gs=map_dbl(sample_balls,suumg))
numg<-samplingg %>%pull(num_gs)
pmin(numg,numb,numr)
```

```
##  [1] 2 2 1 2 0 2 1 0 2 0
```

```
c<-pmin(numg,numb,numr)
x<-1
for(i in c){
  if(i==0){
    x = x+1
  }
}
x/length(c)
```

```
## [1] 0.4
```

```
# we extract a vector corresponding to the number of reds in each trial
```