lab2.2- Nhi Nguyen

Counting successes with if statements

+1 code: "'{r, eval = TRUE} balls = rep(c("R","W"), c(3,3)) number.draws = 2 replicates = 10

successes = vector("numeric", replicates) set.seed(5011)

for (k in 1:replicates) { draw = sample(balls, size = number.draws, replace = FALSE) if (draw[1] == 'W' & draw[2] == 'R') successes[k]=1 }

successes table(successes)

w1r1 = (successes == 1) table(w1r1) sum(w1r1)/replicates

```
a.
We have vector balls with function: balls = rep(c("R","W"), c(3,3)) which means there are balls "R" appe
To modify the bag to contain 5 "R" and 2 "W", we can run with the function: balls = rep(c("R","W"), c(5

b.
"draw" is how we create a sample for the vector "balls" (which has 3 "R" and 3 "W"). For each time choos

c.
A success in that case means that we have W ball in the 1st time and R one after that. So if-statement h

d.
When a set of 2 draws has W and R respectively on first and second pick for the k(th) pick of all 10 rep

e.
'''{r}
balls = rep(c("W","R"), c(3,3))
number.draws = 2
replicates = 10000

successes = vector("numeric", replicates)
set.seed(5011)

for (k in 1:replicates) {
  draw = sample(balls, size = number.draws, replace = FALSE)
  if (draw[1] == 'W' & draw[2] == 'R') successes[k]=1
}

successes
table(successes)

w1r1 = (successes == 1)
table(w1r1)
sum(w1r1)/replicates
```

we have the result here 0.3095

f. `{r, eval = TRUE} P.w = 3/6 P.r = 3/5 P = P.w * P.r P`

+2

a. P = 2C1 * 3/6 * 3/5 = 0.6
b.

```
balls = rep( c('w','r'), c(3,3))
num.draws = 2
num.rep = 10000

success = vector("numeric", num.rep)
set.seed(5011)

for (k in 1:num.rep)  {
  draw = sample(balls, size = num.draws, replace = F)
  if (draw[1] == 'w' & draw[2] == 'r' | draw[1] == 'r' & draw[2] == 'w') success[k] = 1
}

success
table(success)

sum(success) / num.rep
```

We got 0.6028

c. "'{r, eval = TRUE} balls = rep(c('w','r'), c(3,3)) num.draws = 2 num.rep = 1000

success = vector("numeric", num.rep) set.seed(5011)

for (k in 1:num.rep) { draw = sample(balls, size = num.draws, replace = F) if (draw[1] != draw[2]) success[k] = 1 }

success table(success)

sum(success) / num.rep

```
use condition (draw[1] != draw[2]) instead which means 2 balls must be different, so 1 R and 1 W on wha

### Simulating a population with if statements

+3
a.b.d.e.
```{r}
population = 10000
pop = rep(c(0,1), c(population/2, population/2))

p.tallifW = 0.03
p.tallifM = 0.2

sex = vector("numeric", population)
```
```

```
tall = vector("numeric", population)

set.seed(2021)

sex = sample(pop, size = population, replace = T)

for (k in 1:population) {
  if (sex[k] == 0) {
    tall[k] = sample(c(0,1), prob = c(1-p.tallifM, p.tallifM), size = 1, replace = F)
  }
  if (sex[k] == 1) {
    tall[k] = sample(c(0,1), prob = c(1-p.tallifW, p.tallifW), size = 1, replace = F)
  }
}
```

"'{r, eval = TRUE} addmargins(table(sex, tall)) addmargins(table(sex))

p.tallW = sum(sex==1 & tall==1)/population p.tallW

p.tall = sum(tall)/population p.tall

```
c.
tall vector requires the use of if-statements because it depends on which is the sex with different prol
f.
prob that next person walking through the door is W and >6feet is 0.015
prob that next person walking through the door is >6feet is 0.115

#4
a.
P.AA = 0.81
P.Aa = 0.18
p.aa = 0.01

'''{r}
population = 10000

p.fr.A = 0.9
p.fr.a = 0.1

set.seed(2021)

setAA = vector("numeric", population)
setaa = vector("numeric", population)
setAa = vector("numeric", population)

for (k in 1:population) {
  draw = sample(c(0,1), prob = c(0.9,0.1), size = 2, replace = T)
  if (draw[1] == 0 & draw[2] == 0) setAA[k] = 1
  if (draw[1] != draw[2]) setAa[k] = 1;
  if (draw[1] == 1 & draw[2] == 1) setaa[k] = 1;
}
```

```
{r, eval = TRUE} AA = (setAA == 1) sum(AA)/ population Aa = (setAa == 1) sum(Aa)/ population
aa = (setaa == 1) sum(aa)/ population
```

b.

```
prob.AA = 0.81
prob.Aa = 0.18
prob.aa = 0.01
prob.AAd = 0.8
prob.Aad = 0.4
prob.aad = 0.1
population = 10000

set.seed(2021)

disease = vector("numeric", population)

genotype = sample(c("AA","Aa","aa"), size = population, prob = c(prob.AA,prob.Aa, prob.aa), replace = T

for (k in 1:population){
  if (genotype[k] == "AA") disease[k] = sample(c(0,1), size = 1, prob = c(1-prob.AAd,prob.AAd))
  if (genotype[k] == "Aa") disease[k] = sample(c(0,1), size = 1, prob = c(1-prob.Aad,prob.Aad))
  if (genotype[k] == "aa") disease[k] = sample(c(0,1), size = 1, prob = c(1-prob.aad,prob.aad))
}

addmargins(table(genotype, disease))
```

i. `{r, eval = TRUE} sum(disease)/population`
ii. `{r, eval = TRUE} sum(genotype == "AA")*prob.AAd / sum(disease)`
iii. Answer for i: p.d = prob.AA * prob.AAd + prob.Aa * prob.Aad + prob.aa * prob.aad = 0.81x0.8 + 0.18x0.4 + 0.01x0.1 = 0.721

Answer for ii: P(A|B) = P(A intersect B) / P(B) (B is the event that disease occurs for all genotype A is the next event to choose which one is AA) = 0.81x0.8 / p.d = 0.8987