Assignment 1

1 Blood types genetics

Ex 1.1

$\Omega = \{\mathrm{I^A, I^Ai, I^BI^B, I^Bi, I^AI^B, ii}\}$

Ex 1.2

$\mathrm{P(ii)} = 0.25$
$\mathrm{P(I^Ai, ii)} = 0.75$

Ex 1.3

$\mathrm{P(C = I^AI^B)} = 0.25$

Ex 1.4

Phenotypes: AB, A, B
Distribution: AB = 0.25; A = 0.375; B = 0.375

Ex 1.5

$\mathrm{P(I^Bi, I^BI^B \mid B = I^AI^B)} = 0.375$

1.1 Some simulation in R

Ex 2.1

```r
sample(c("ii", "iB", "iA", "BB", "AA", "AB"), size = 1, replace = TRUE)
```

Ex 2.2

```r
offspringG <- function(Parent1, Parent2){
  Allele1 <- sample(unlist(strsplit(Parent1, split = "")), replace = TRUE, size=1)
  Allele2 <- sample(unlist(strsplit(Parent2, split = "")), replace = TRUE, size=1)
  return(paste(Allele1, Allele2, sep = ""))
}


offspringP <- function(Genotype){
  Phenotype <- ""
  if (Genotype == "ii"){
    Phenotype <- "Type O"
  }
  if (Genotype == "AB" | Genotype == "BA"){
    Phenotype <- "Type AB"
  }
  if (Genotype == "iB" | Genotype == "Bi" | Genotype == "BB"){
    Phenotype <- "Type B"
  }
  if (Genotype == "iA" | Genotype == "Ai" | Genotype == "AA"){
    Phenotype <- "Type A"
  }
  return(Phenotype)
}
```

Ex 2.3

```
offspingiAiA <- c()
for(i in 1:1000){
  offspingiAiA[i] <- offspringG("iA", "iA")
}
sum(offspingiAiA == "ii")/length(offspingiAiA)
```

```
## [1] 0.231
```

```
sum(offspingiAiA != "AA")/length(offspingiAiA)
```

```
## [1] 0.758
```

Ex 2.4

```
secondGen <- c()

for(i in 1:1000){
  A <- offspringG("iA", "iB")
  B <- offspringG("AB", "AB")
  C <- offspringP(offspringG(A, B))
  secondGen[i] <- C
}

head(secondGen, 50)
```

```
##  [1] "Type B"  "Type A"  "Type AB" "Type A"  "Type A"  "Type AB" "Type A"
##  [8] "Type A"  "Type B"  "Type A"  "Type A"  "Type AB" "Type A"  "Type A"
## [15] "Type A"  "Type AB" "Type A"  "Type B"  "Type AB" "Type B"  "Type B"
## [22] "Type B"  "Type AB" "Type B"  "Type A"  "Type B"  "Type A"  "Type A"
## [29] "Type B"  "Type B"  "Type A"  "Type AB" "Type AB" "Type B"  "Type B"
## [36] "Type A"  "Type A"  "Type A"  "Type B"  "Type B"  "Type A"  "Type AB"
## [43] "Type B"  "Type B"  "Type B"  "Type B"  "Type A"  "Type A"  "Type A"
## [50] "Type A"
```

2 A case study of DNA sequence

Ex 3.1

$\Omega = \{AA, AT, AC, AG, TA, TT, TC, TG, CA, CT, CC, CG, GA, GT, GC, GG\}$

Ex 3.2

$P(ACG) = (0.25 \times 0.25 \times 0.25) \times 3 = 0.047$

Ex 3.3

$\Omega = \{\emptyset, \mathrm{A}, \mathrm{T}, \mathrm{C}, \mathrm{AA}, \mathrm{AT}, \mathrm{AC}, \mathrm{AAA}, \ldots\}$
$\mathrm{P}(\text{length } 10 \ \mathrm{G}^{\mathrm{C}}) = (0.75)^{10} = 0.05$

Ex 3.4

1: $\mathrm{P}(\mathrm{CG}|\mathrm{C}) = 0.5 \times 0.1 = 0.05$
2: $\mathrm{P}(\mathrm{CG}|\mathrm{C}) = 0.3 \times 0.4 = 0.12$

2.1 Some simulations in R

Ex 4.1

```
ACGsim <- function(reps){
  count = 0

  for(i in 1:reps){
    sqce_5 <- paste(sample(c("A", "T", "G", "C"), replace = TRUE, size = 5), collapse = "")

    if(grepl("ACG", sqce_5, fixed = TRUE)){
      count <- count + 1
    }
  }
  return(count/reps)
}
ACGsim(10000)
```

```
## [1] 0.0457
```

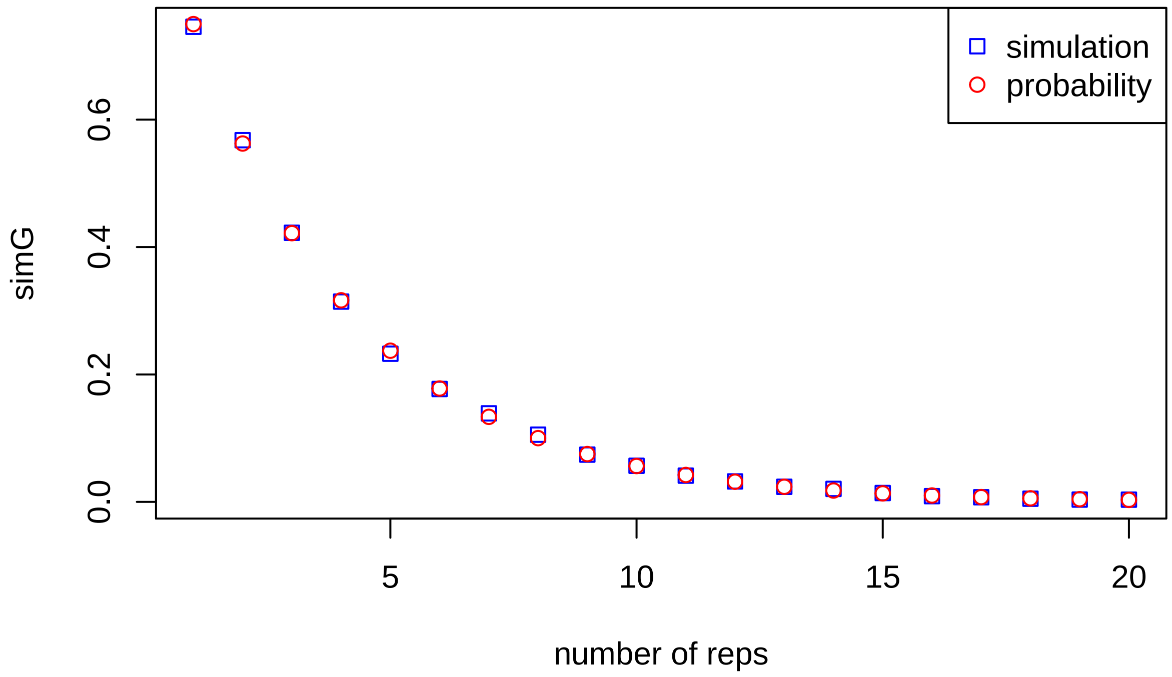
Ex 4.2

```
Gsim <- function(reps, length = 10){
  sqces_with_G <- 0
  for(i in 1:reps){
    seq <- sample(c("A", "C", "T", "G"), replace =TRUE, size = length)
    if ("G" %in% seq){
      sqces_with_G <- sqces_with_G + 1
    }
  }
  sqces_with_no_G <- reps - sqces_with_G
  return(sqces_with_no_G/reps)
}

Gsim(10000)
```

```
## [1] 0.0521
```

Ex 4.3

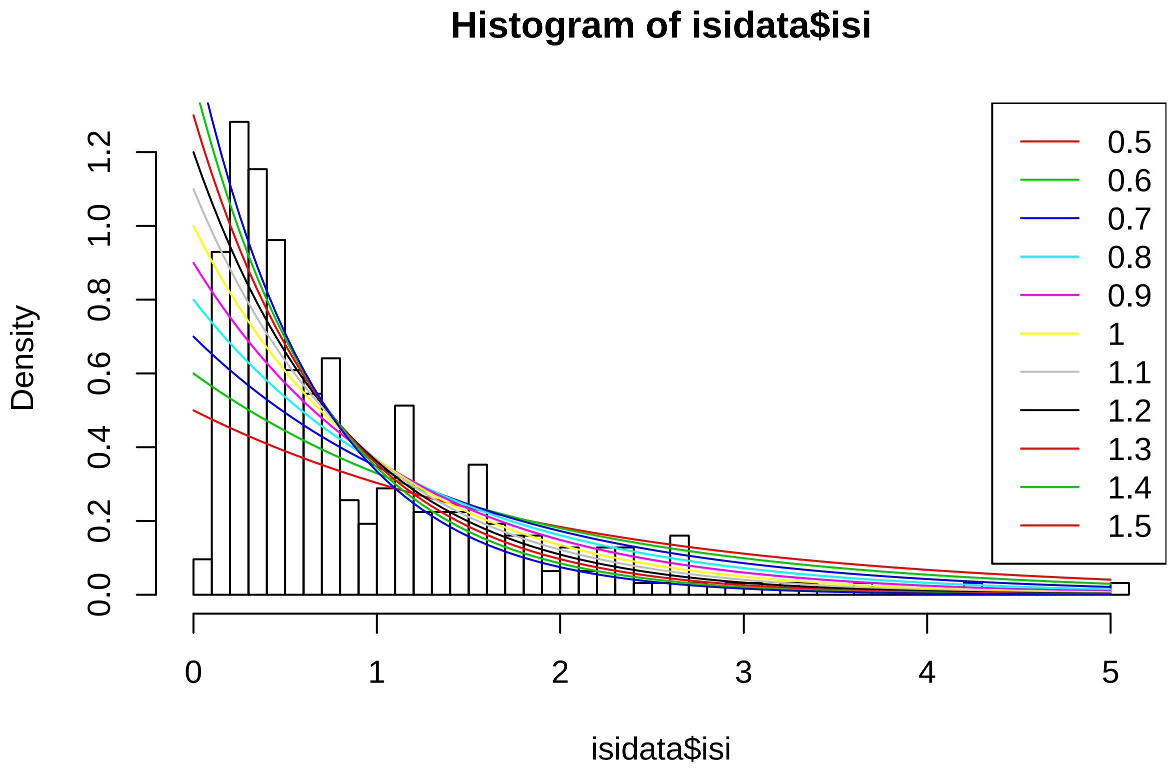
```
simG <- c()
pG <- c()
for(i in 1:20){
  simG[i] <- Gsim(10000, length = i)
  pG[i] <- dbinom(i, size = i, prob = 0.75)
}

plot(1:20, simG, col = "blue", pch = 0, xlab = "number of reps")
points(1:20, pG, col = "red", pch = 1)
legend("topright",legend = c("simulation", "probability"),
       col = c("blue", "red"), pch = c(0, 1))
```



3 A case study of neuronal data

Ex 5.1 and 5.2

```
isidata <- read.table("neuronspikes.txt", col.names = "isi")
hist(isidata$isi, breaks = 50, probability = TRUE)

for(i in seq(0.5, 1.5, 0.1)){
  curve(dexp((x), rate = i), from = 0, to = 5, col= (i*10)-3, add = TRUE)
}

legend("topright", legend=c(seq(0.5,1.5,0.1)),
       col=c(2:length(seq(0.5,1.5,0.1))), lty = 1)
```



A λ value of 1.5 fits the data the best.

Ex 5.3

```
pexp(1, rate = 1.5)
```

```
## [1] 0.7768698
```

```
pexp(1.5, rate = 1.5) - pexp(0.5, rate = 1.5)
```

```
## [1] 0.3669673
```

4 Brain cell database

Ex 6.1

The option na.strings converts empty strings to NA values.

Ex 6.2

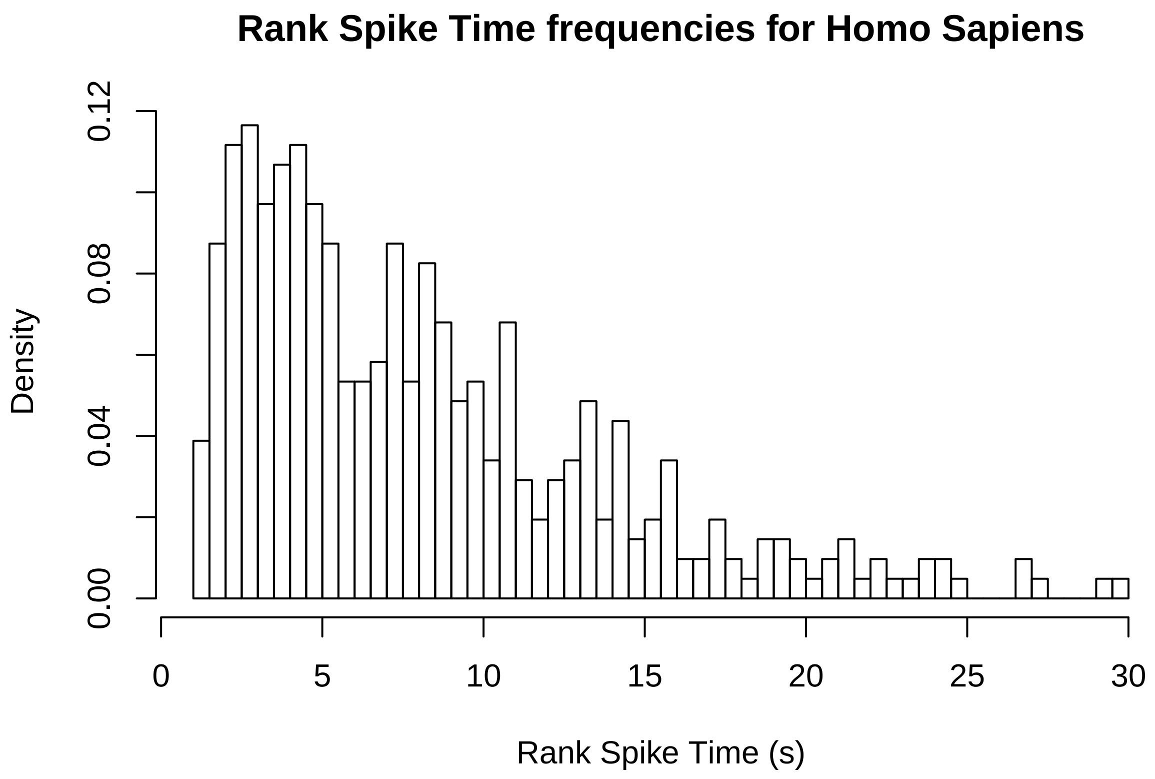
The donor species column indicates which values are humans and which are mice.

```
sum(cells$donor__species == "Homo Sapiens") / length(cells[,1])
```

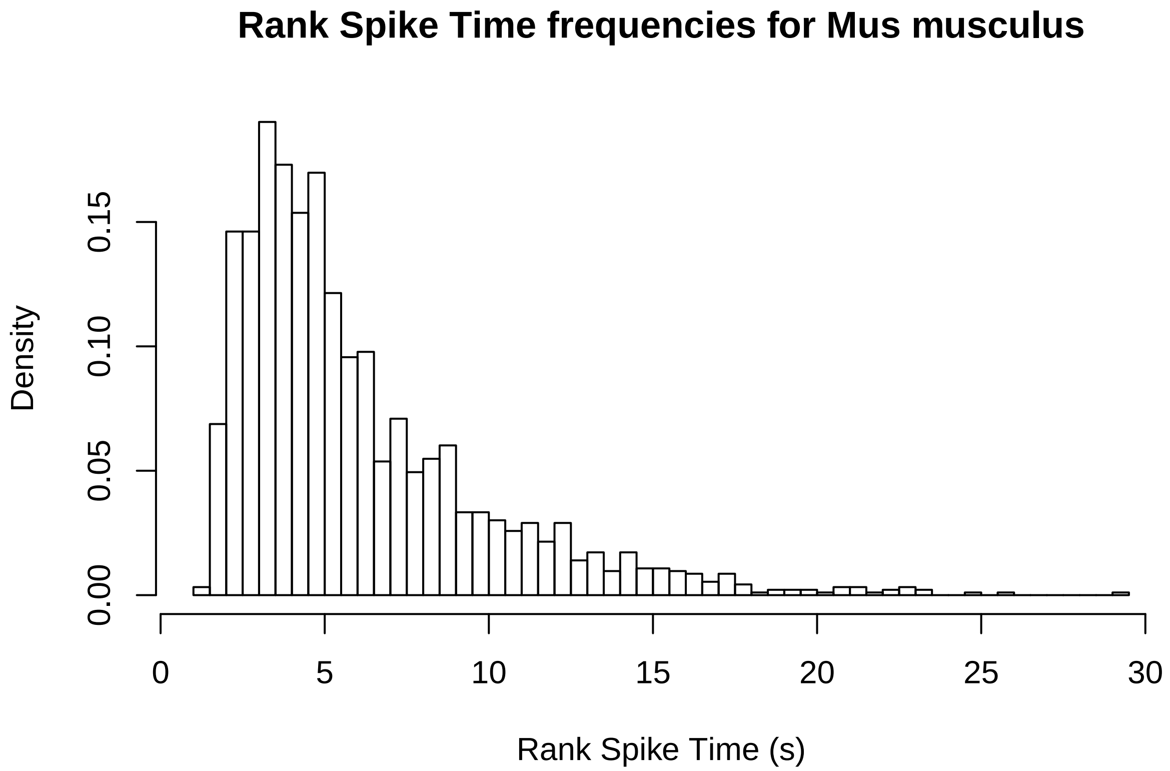
```
## [1] 0.1770253
```

Ex 6.3

```
hist(cells$ef__peak_t_ramp[cells$donor__species == "Homo Sapiens"],
     proba = TRUE,  xlab = "Rank Spike Time (s)",
     breaks = 50, main = "Rank Spike Time frequencies for Homo Sapiens")
```

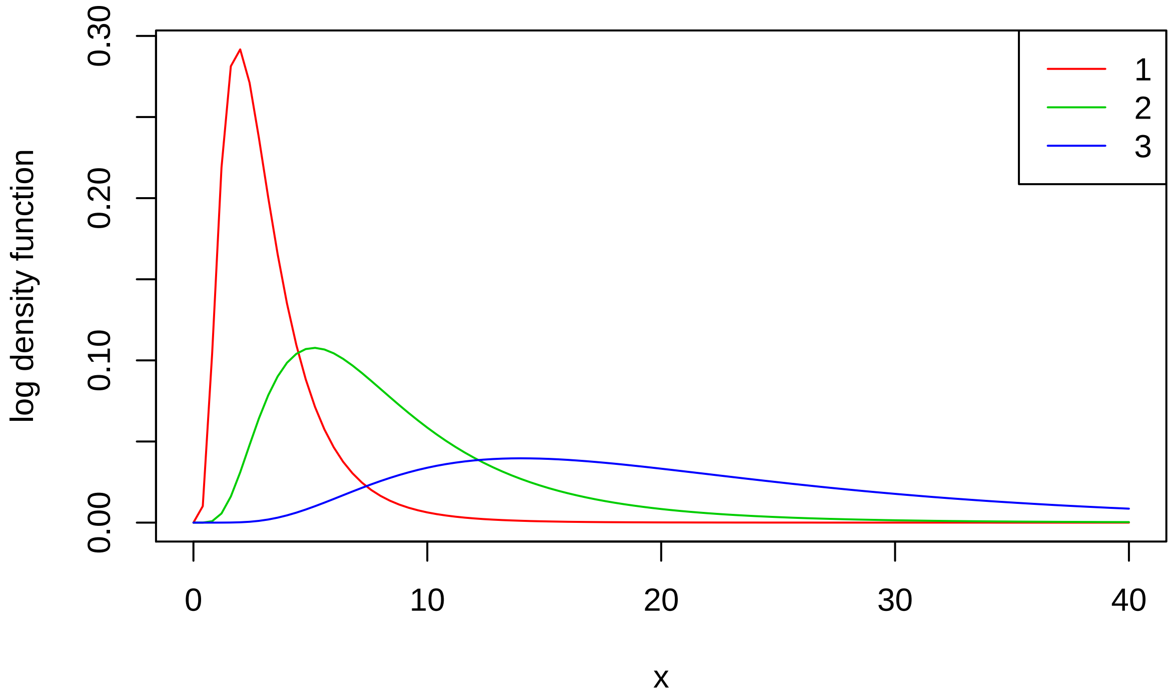


```
hist(cells$ef__peak_t_ramp[cells$donor__species == "Mus musculus"],
     proba = TRUE, xlab = "Rank Spike Time (s)",
     breaks = 50, main = "Rank Spike Time frequencies for Mus musculus")
```



Ex 6.4

```
first_plot = FALSE

for(i in 1:3){
  curve(dlnorm(x, sdlog = 0.6, meanlog = i), from = 0, to = 40,
        add = first_plot, col = i+1, ylab = "log density function")
  first_plot = TRUE
}

legend("topright", legend = 1:3, col = 2:4, lty = 1)
```

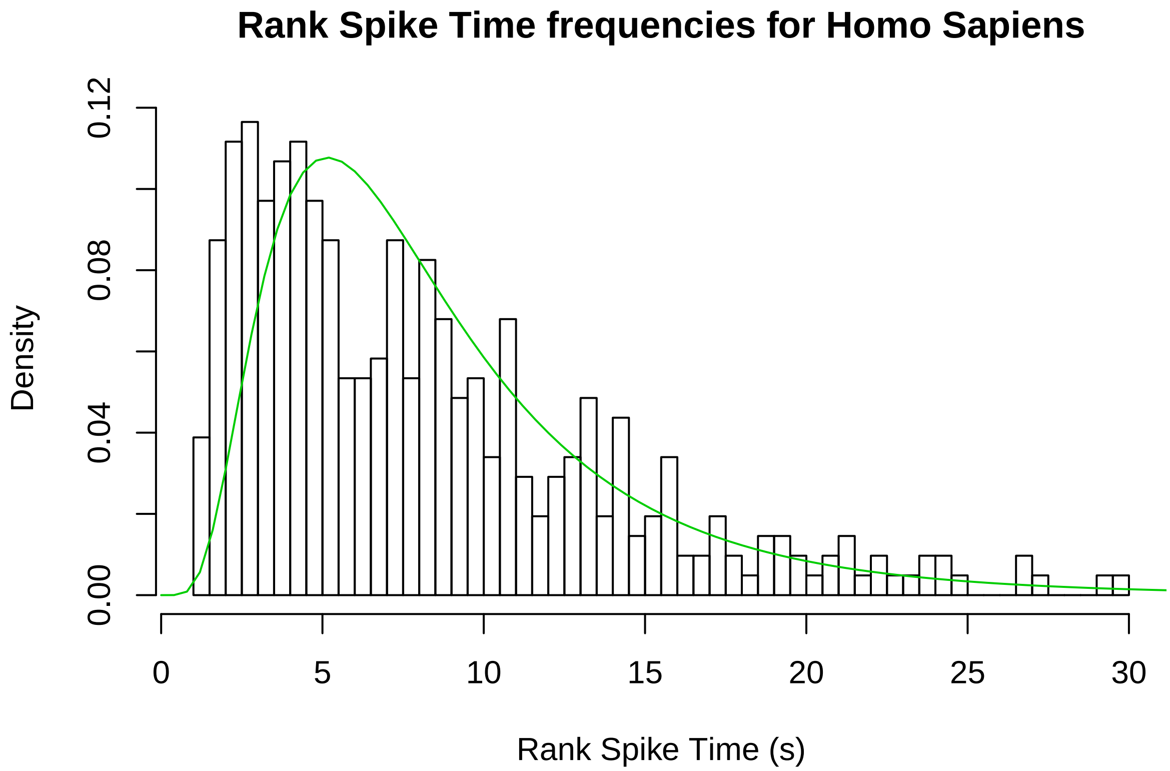


Ex 6.5

meanlog = 2 is best, based on the fit of the corresponding density function in the plot.

Ex 6.6

```r
hist(cells$ef__peak_t_ramp[cells$donor__species == "Homo Sapiens"],
     proba = TRUE,  xlab = "Rank Spike Time (s)",
     breaks = 50, main = "Rank Spike Time frequencies for Homo Sapiens")

curve(dlnorm(x, sdlog = 0.6, meanlog = 2),
      from = 0, to = 40, add = TRUE,
      col = 3, ylab = "log density function")
```



Yes, the density fits the data.

Ex 6.7

```r
sum(cells$donor__species == "Homo Sapiens" & cells$donor__sex == "Male")
```

```
## [1] 234
```

```r
sum(cells$donor__species == "Homo Sapiens" & cells$donor__sex == "Female")
```

```
## [1] 179
```