Week 1

gherardo varando

November 27, 2018

Ex 1 Blood types

In classical genetics, we recognize three alleles for the ABO gene that regulates ABO blood type: i, I^A and I^B . Any individual inherits a complete set of chromosomes from its two parents and thus the genotype of an individual is a pair of alleles (the order is not important).

1.1 Sample space

Describe the sample space related to the genotype of an individual

$$\Omega = \{\{i, i\}, \{i, I^A\}, \{i, I^B\}, \{I^A, I^B\}, \{I^A, I^A\}, \{I^B, I^B\}\}$$

$$|\Omega| = 6 = 2 \times 3 \qquad \text{(not requested)}$$

1.2 Probability for the offspring

Suppose both parents have genotype $iI^A = \{i, I^A\}$, compute the probability that the offspring will have genotype $ii = \{i, i\}$ and the probability that at least one of the two alleles will be i

We can represent the situation with a table, that we can generate in R as

```
parent1 <- c("i", "i^A")
parent2 <- c("i", "i^A")
offspring <- apply(expand.grid(parent1, parent2), MARGIN = 1,
                   FUN = function(x){ return(paste(x[1], x[2]))})
matrix(data = offspring , nrow = 2,
       dimnames = list(parent1 = parent1, parent2 = parent2))
```

```
##        parent2
## parent1 i       i^A
##     i   "i i"   "i i^A"
##     i^A "i^A i" "i^A i^A"
```

The possible genotypes for the offspring are,

$$ii, iI^A, I^AI^A$$

The probabilities of each *cell* in the table is 0.25 thus, the probabilities for the possible genotypes are,

$$P(ii) = 0.25 \quad P(I^AI^A) = 0.25 \quad P(iI^A) = 0.25 + 0.25 = 0.5$$

Or similarly we can use the independence assumption to say that

$$P(ii) = P(\text{parent 1 gives } i \text{ and parent2 gives } i) =$$

$$P(\{\text{parent 1 gives } i\} \cap \{\text{ parent2 gives } i\}) =$$

$$P(\{\text{parent 1 gives } i\}) \times P(\{\text{ parent2 gives } i\}) = 0.5 \times 0.5$$

And so on for the other probabilities.

As you can see $P(ii) + P(iI^A) + P(I^A I^A) = 1$.

Now we can easily compute the required probabilities,

$$P(ii) = 0.25$$

and

$$P(\text{at least one alleles is } i) = P(\{ii\} \text{ or } \{iI^A\}) = P(\{ii\} \cup \{iI^A\})$$
$$= P(ii) + P(iI^A) = 0.75$$

Since the two events $\{ii\}$ and $\{iI^A\}$ are disjoint.

1.3 Two generations

- **Parents with genotypes iI^A and iI^B generate offspring A**
- **Parents with genotypes $I^A I^B$ generate offspring B**
- **Parents A and B generate individual C**

Compute the probability of genotype of C being $I^A I^B$

$$P(\text{genotype of } C = I^A I^B) =$$
$$P(A \to C = I^A \text{ and } B \to C = I^B) + P(A \to C = I^B \text{ and } B \to C = I^A)$$

where we use the notation $K \to M = zzz$ to denote the events where parent K gives allele zzz to offspring M.

We proceed now to obtain the probabilities

$$P(A \to C = I^A \text{ and } B \to C = I^B) \qquad \text{(first case)}$$

$$P(A \to C = I^B \text{ and } B \to C = I^A) \qquad \text{(second case)}$$

Since the events of two different parents are independent, we have that

$$P(A \to C = I^A \text{ and } B \to C = I^B) = P(A \to C = I^A) \times P(B \to C = I^B) \qquad \text{(first case)}$$

To obtain for example $P(A \to C = I^A)$ we need to consider the possible genotypes for A,

Individual A can have one of the four possible genotypes,

$$ii, iI^A, iI^B, I^A I^B$$

And

$$P(A = ii) = P(A = iI^A) = P(A = iI^B) = P(A = I^A I^B) = 0.25$$

Moreover we have that,

$$P(A \to C = I^A | A = ii) = 0$$
$$P(A \to C = I^A | A = iI^A) = 0.5$$
$$P(A \to C = I^A | A = iI^B) = 0$$
$$P(A \to C = I^A | A = I^A I^B) = 0.5$$

Thus using the chain rule ($P(A, B) = P(A|B)P(B)$) we obtain,

$$P(A \to C = I^A) = (0 \times 0.25 + 0.5 \times 0.25 + 0 \times 0.25 + 0.5 \times 0.25 = 0.25$$

For individual B we have that,

$$P(B = I^A I^A) = P(I^B I^B) = 0.25$$

and

$$P(B = I^A I^B) = 0.5$$

and moreover,

$$P(B \to C = I^B | B = I^A I^A) = 0$$
$$P(B \to C = I^B | B = I^A I^B) = 0.5$$
$$P(B \to C = I^B | B = I^B I^B) = 1$$

Thus now using again the chain rule,

$$P(B \to C = I^B) = 0 \times 0.25 + 0.5 \times 0.5 + 1 \times 0.25 = 0.5$$

We can now obtain from equation (first case) that,

$$P(A \to C = I^A \text{ and } B \to C = I^B) = 0.25 \times 0.5 = 0.125$$

Similarly we need to obtain,

$$P(A \to C = I^B \text{ and } B \to C = I^A) = P(A \to C = I^B) \times P(B \to C = I^A) \qquad \text{(second case)}$$

The same reasoning will lead us to,

$$P(A \to C = I^B) = 0.25$$

and

$$P(B \to C = I^A) = 0.5$$

Thus,

$$P(A \to C = I^B \text{ and } B \to C = I^A) = 0.125$$

And finally

$$P(\text{genotype of } C = I^A I^B) =$$
$$P(A \to C = I^A \text{ and } B \to C = I^B) + P(A \to C = I^B \text{ and } B \to C = I^A) =$$
$$0.125 + 0.125 = 0.25$$

1.4 Phenotypes

Which are the possible phenotypes for individual C ?

The possible genotypes are,

$$\{iI^A, iI^B, I^AI^B, I^AI^A, I^BI^B\} = \Omega \setminus \{ii\}$$

Then the possible phenotypes are:

- **Type B** from iI^B or I^BI^B
- **Type A** from iI^A or I^AI^A
- **type AB** from I^AI^B

The only phenotypes that individual C can not be is **type 0**.

Can you describe the distribution of the phenotype of C?

We have that,

$$P(\text{phenotype } C = \text{ Type AB }) = P(C = I^AI^B) = 0.25$$

Moreover it is obvious that,

$$P(\text{phenotype } C = \text{ Type A }) = P(\text{phenotype } C = \text{ Type B })$$

since the problem is *symmetric*. And since the sum of the probabilities has to be 1, we obtain that,

$$P(\text{phenotype } C = \text{ Type A }) = (1 - 0.25)/2 = 0.375$$

and

$$P(\text{phenotype } C = \text{ Type B }) = (1 - 0.125)/2 = 0.375$$

1.5 Conditional probability

Ex 2 Some simulations

2.1 Random genotype

How can we simulate a random individual genotype ?

We use the `sample` function in R, and we represent genotypes with vectors.

```r
randomGenotype <- function(alleles = c("i", "A", "B")){
  sample(alleles, size = 2, replace = T)
}

randomGenotype()
```

```
## [1] "A" "A"
```

2.2 Random offspring and phenotype

```r
randomOffspring <- function( parent1, parent2){
  c( sample(parent1, size = 1), sample(parent2, size = 1))
}
```

```
p1 <- randomGenotype()
p2 <- randomGenotype()

randomOffspring(p1, p2)
```

```
## [1] "B" "B"
```

```
genotype2phenotype <- function(genotype){
  if (all(genotype == "i")){
   return("Type O")
  }else if ("i" %in% genotype){
      return(paste("Type", genotype[genotype != "i"] ))
  }else if (genotype[1] == genotype[2]){
    return(paste("Type", genotype[1]))
  }else{
    return("Type AB")
  }
}
```

Lets test it,

```
genotype2phenotype(c("i","i"))
```

```
## [1] "Type O"
```

```
genotype2phenotype(c("A","i"))
```

```
## [1] "Type A"
```

```
genotype2phenotype(c("i","B"))
```

```
## [1] "Type B"
```

```
genotype2phenotype(c("A","A"))
```

```
## [1] "Type A"
```

```
genotype2phenotype(c("B","B"))
```

```
## [1] "Type B"
```

```
genotype2phenotype(c("A","B"))
```

```
## [1] "Type AB"
```

Or in a more compact way,

```
#generate all possible genotypes
possible <- expand.grid(c("i","A","B"), c("i","A","B") )

possible$phen <- apply(possible, MARGIN = 1, FUN = genotype2phenotype)

possible
```

```
##   Var1 Var2    phen
## 1    i    i  Type O
## 2    A    i  Type A
## 3    B    i  Type B
## 4    i    A  Type A
## 5    A    A  Type A
## 6    B    A Type AB
## 7    i    B  Type B
## 8    A    B Type AB
## 9    B    B  Type B
```

2.3

Approximate from simulation the probabilities of Ex 1.2

```
parent1 <- parent2 <- c("i", "A")

N <- 10000
## or you can use sapply or a for.
offsprings <- t(replicate(N, randomOffspring(parent1, parent2)))

pii_est <- nrow(offsprings[ offsprings[,1] == "i" &
                              offsprings[,2] == "i",  ]) / N
pii_est
```

```
## [1] 0.2527
```

```
pi_est <- nrow(offsprings[ offsprings[,1] == "i" |
                              offsprings[,2] == "i",  ]) / N
pi_est
```

```
## [1] 0.7435
```

2.4

Code the two generation scenario and sample phenotype of C

```
twoGenC <- function(){
A <- randomOffspring(c("i","A"),  c("i","B"))
B <- randomOffspring(c("A","B"), c("A","B"))

C <- randomOffspring(A, B)
return(list(A = A, B = B, C = C))
}

twoGenC()
```

```
## $A
## [1] "i" "B"
##
## $B
## [1] "A" "B"
##
## $C
## [1] "i" "A"
```

Sample 1000 phenotypes of C,

```
phenotypesC <- replicate(1000, genotype2phenotype(twoGenC()$C))
table(phenotypesC)
```

```
## phenotypesC
##  Type A Type AB  Type B
##     385     249     366
```

```r
table(phenotypesC) / 1000
```

```
## phenotypesC
##  Type A Type AB  Type B
##   0.385   0.249   0.366
```

Increase the number of sample

```r
N <- 100000
phenotypesC <- replicate(N, genotype2phenotype(twoGenC()$C))
table(phenotypesC)
```

```
## phenotypesC
##  Type A Type AB  Type B
##   37615   24895   37490
```

```r
table(phenotypesC) / N
```

```
## phenotypesC
##  Type A Type AB  Type B
## 0.37615 0.24895 0.37490
```

Ex 3 DNA sequences

Let

$$\Omega = \{A, T, C, G\}$$

The sample space of a single DNA nucleotide type.

3.1

Describe the sample space of DNA sequences of length 2

$$\Theta = \Omega \times \Omega = \Omega^2 = \{(A, A), (A, T), \ldots, (A, G), (T, A), (T, T) \ldots (G, G)\}$$

3.2

Describe the event of observing the motif ACG in a DNA sequence of length 5

$$E = \{ACG * *, *ACG*, * * ACG\}$$

where $*$ indicates whatever nucleotide in Ω.

Compute the probability of this event

Observe that the events $ACG * *$, $*ACG*$, $* * ACG$ are disjoints. Thus,

$$P(E) = P(ACG * *) + P(*ACG*) + P(* * ACG)$$

Moreover,

$$P(ACG * *) = P(*ACG*) = P(* * ACG) = \left(\frac{1}{4}\right)^3$$

since the events in different positions are independent.

Thus,

$$P(E) = 3\left(\frac{1}{4}\right)^3$$

```r
3 * (0.25)^3
```

```
## [1] 0.046875
```

3.3

Suppose we observe from a given position in the genome. We keep observing until we find the nucleotide G. Describe the sample space of the observed DNA sequence. What is the probability that the DNA sequence observed before observing G has length 10?

$$S = \{G, *G, **G, ***G, \dots\}$$

$$P(\text{length sequence before observing G } = 10) = (1-0.25)^{10}(0.25)$$

```r
(1-0.25)^(10) * (0.25)
```

```
## [1] 0.01407838
```

3.4

Now we remove the independence assumption. Suppose that the probability of observing A,T,C or G given the previous nucleotide being C, is respectively 0.2, 0.2, 0.5, 0.1. In general in different locations of the DNA sequence, this conditional probabilities will be different and this fact is important for biological sequence detecting.

Suppose we observe C at a given location, what is the probability of observing the motif CG immediately after?

$$P(\text{observing CG starting from position 2}|\text{observe C in position 1}) = ?$$

Lets introduce three *extended* random variable D_1, D_2, D_3, indicating the nucleotide in position 1,2 and 3.

$$D_1, D_2, D_3$$

We want to compute,

$$P(D_2 = C, D_3 = G|D_1 = C) = P(D_3 = G|D_2 = C, D_1 = C)P(D_2 = C|D_1 = C) =$$
$$0.1 \times 0.5 = 0.05$$

Suppose now in a different region of the DNA sequence, the conditional probabilities of A, T, C, G given C become 0.1, 0.2, 0.3, 0.4, then what is in that region the probability of observing CG given that we observed C in the previous location?

Similarly,

$$P(D_2 = C, D_3 = G|D_1 = C) = P(D_3 = G|D_2 = C, D_1 = C)P(D_2 = C|D_1 = C) =$$
$$0.4 \times 0.3 = 0.12$$

Ex 4 Some simulations

4.1

```
simDNA <- function(length = 1, nucleotides = c("A", "T", "C", "G")){
  sample(nucleotides, size = length, replace = TRUE)
}

estMotifProb <- function(length = 5, motif = c("A", "C", "G"), N = 1000){
  S<-t(replicate(N, simDNA(length)))

  sum(apply(S, MARGIN = 1, FUN = function(dna){
    length(grep(pattern = paste0(motif,collapse = ""),
                x = paste0(dna, collapse = ""), value = FALSE)) > 0
  } )) / N
}

estMotifProb(5, c("A", "C", "G"), 1000)
```

```
## [1] 0.044
```

4.2

```
tryBefore <- function(before = "G", max = Inf){
  x <- simDNA(1)
  k <- 0
  while (x != before & k <= max ){
    x <- simDNA(1)
    k <- k + 1
  }
  return(k)
}

N <- 100000
sum(replicate(N, tryBefore("G", 11)) == 10) / N
```

```
## [1] 0.01464
```

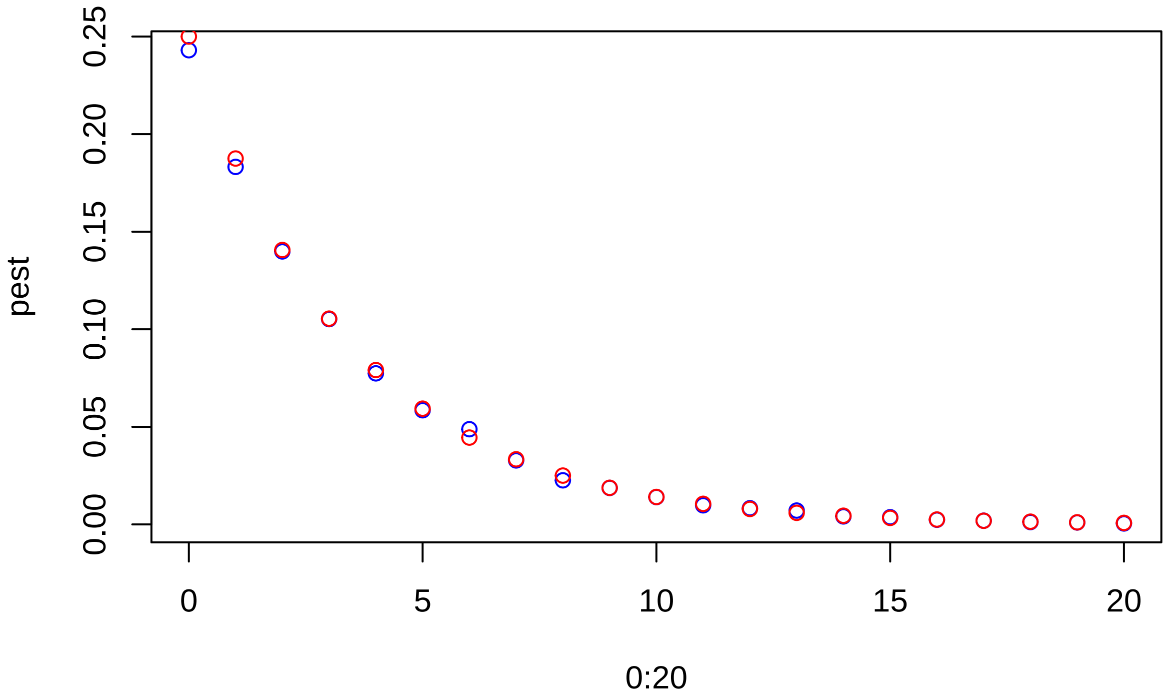
4.3

```
N <- 10000

##estimation as before
pest <- sapply(0:20, function(i){
  sum(replicate(N, tryBefore("G", i + 1)) == i) / N
})

## real prob
preal <- sapply(1:21, function(i){
  return( 0.75^(i-1) * 0.25)
})
```

```r
plot(0:20,pest, col = "blue")
points(0:20, preal, col = "red")
```



Ex 5 ISI data set

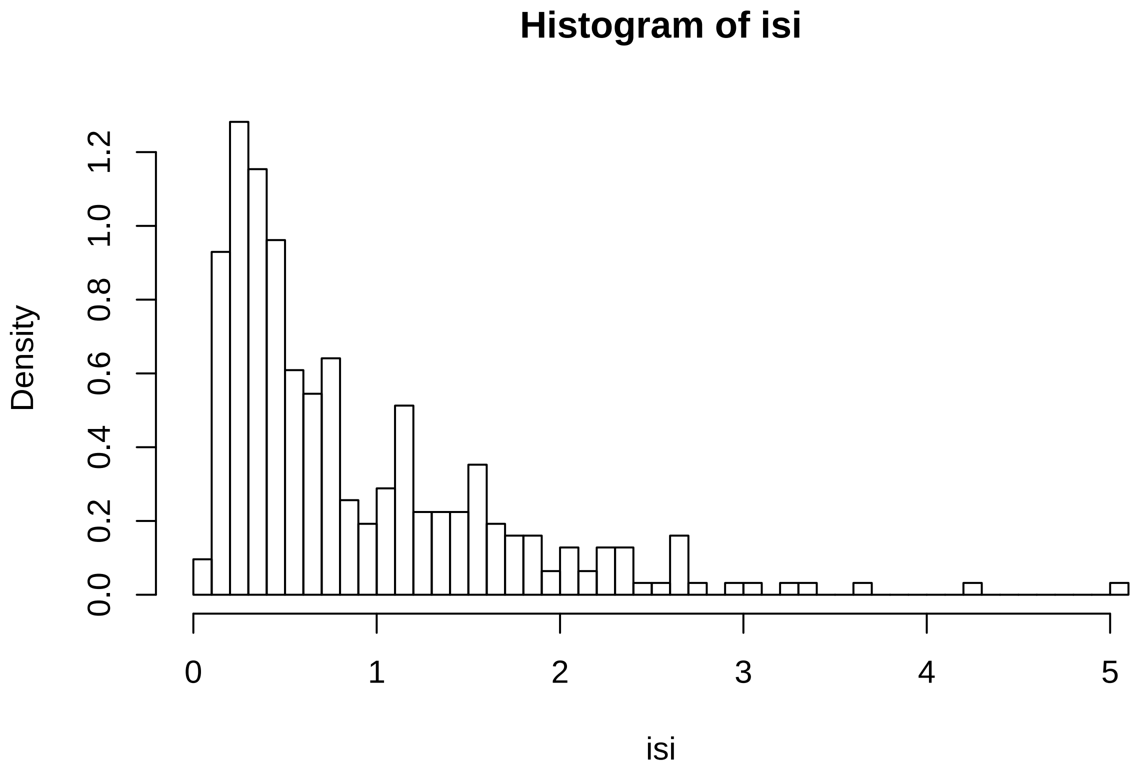
We load the data

```r
isidata <- read.table("neuronspikes.txt", col.names = "isi")
isi <- isidata$isi ##here we have the vector of observations
```

5.1

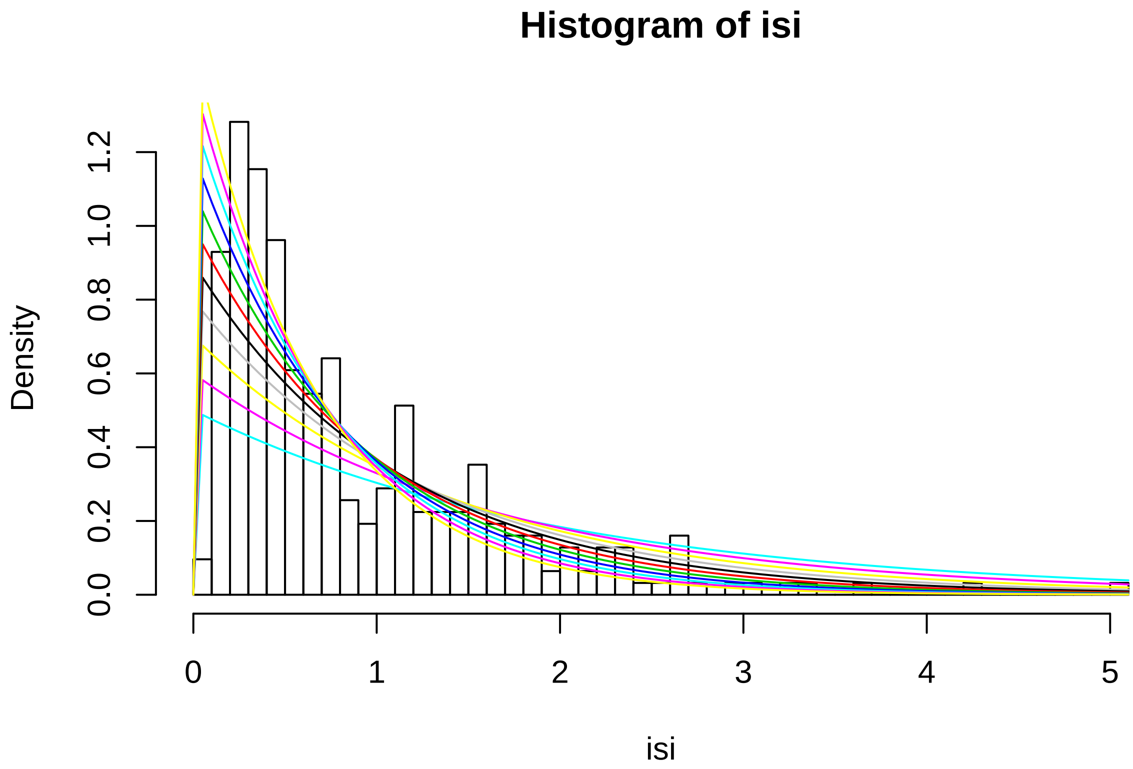
Plotting the histogram

```r
hist(isi, breaks = 50, probability = TRUE)
```



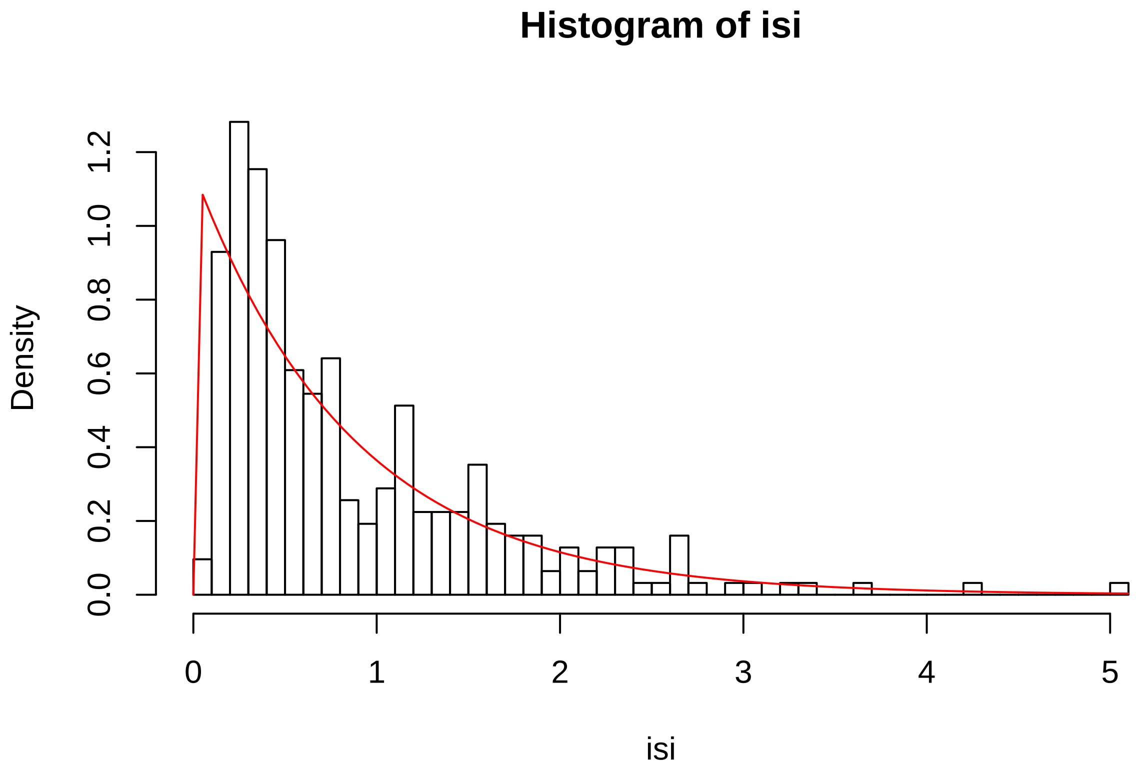
5.2

```
hist(isi, breaks = 50, probability = TRUE)
for (rate in seq(0.5, 1.5, 0.1)){
  curve(dexp(x, rate = rate), add = TRUE, col = rate*10)
}
```



Maybe a nice parameter can be $\lambda = 1.15$

```
hist(isi, breaks = 50, probability = TRUE)
curve(dexp(x, rate = 1.15), col = "red", add = TRUE)
```



5.3

```
rate <- 1.15

pexp(1, rate = rate)
```

```
## [1] 0.6833632
```

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

```
## [1] 0.3845318
```

Ex 6 Brain cell types data set

We load the brain cell data set,

```
cells <- read.csv("cell_types.csv", na.strings = "")
```

6.1

We use the parameter `na.strings = ""` to specify which string (in this case the empty strig) is used as a placeholder for the `NA` value.

6.2

To find which observations are from human donors or mice, firts of all we locate the appropriate variable in the data, and secondly we check which are the possible values.

Lets list all the variables,

```
colnames(cells)
```

```
##  [1] "line_name"
##  [2] "specimen__id"
##  [3] "specimen__name"
##  [4] "specimen__hemisphere"
##  [5] "structure__id"
##  [6] "structure__name"
##  [7] "structure__acronym"
##  [8] "structure_parent__id"
##  [9] "structure_parent__acronym"
## [10] "structure__layer"
## [11] "nr__max_euclidean_distance"
## [12] "nr__number_stems"
## [13] "nr__number_bifurcations"
## [14] "nr__average_contraction"
## [15] "nr__average_parent_daughter_ratio"
## [16] "nr__reconstruction_type"
## [17] "nrwkf__id"
## [18] "erwkf__id"
## [19] "ef__fast_trough_v_long_square"
## [20] "ef__upstroke_downstroke_ratio_long_square"
## [21] "ef__adaptation"
## [22] "ef__f_i_curve_slope"
## [23] "ef__threshold_i_long_square"
## [24] "ef__tau"
## [25] "ef__avg_isi"
## [26] "ef__avg_firing_rate"
## [27] "ef__ri"
## [28] "ef__peak_t_ramp"
## [29] "ef__vrest"
## [30] "si__height"
## [31] "si__width"
## [32] "si__path"
## [33] "csl__x"
## [34] "csl__y"
## [35] "csl__z"
## [36] "csl__normalized_depth"
## [37] "cell_reporter_status"
## [38] "m__glif"
## [39] "m__biophys"
## [40] "m__biophys_perisomatic"
## [41] "m__biophys_all_active"
## [42] "tag__apical"
## [43] "tag__dendrite_type"
## [44] "morph_thumb_path"
## [45] "ephys_thumb_path"
## [46] "ephys_inst_thresh_thumb_path"
## [47] "donor__age"
## [48] "donor__sex"
## [49] "donor__disease_state"
## [50] "donor__race"
## [51] "donor__years_of_seizure_history"
## [52] "donor__species"
## [53] "donor__id"
## [54] "donor__name"
```

we are interested in `donor__species`

Since string are converted to factors ina data.frame (thanks to `read.csv` function), we can check the `levels`,

```
levels(cells$donor__species)
```

```
## [1] "Homo Sapiens" "Mus musculus"
```

So we can extract the observations from humans and from mice,

```
human <- cells$donor__species == levels(cells$donor__species)[1]
mouse <- cells$donor__species == levels(cells$donor__species)[2]
```

We check how many they are,

```
sapply( list(human = human, mouse = mouse), sum)
```

```
## human mouse
##   413  1920
```

The proportion of human cells in the data set is,

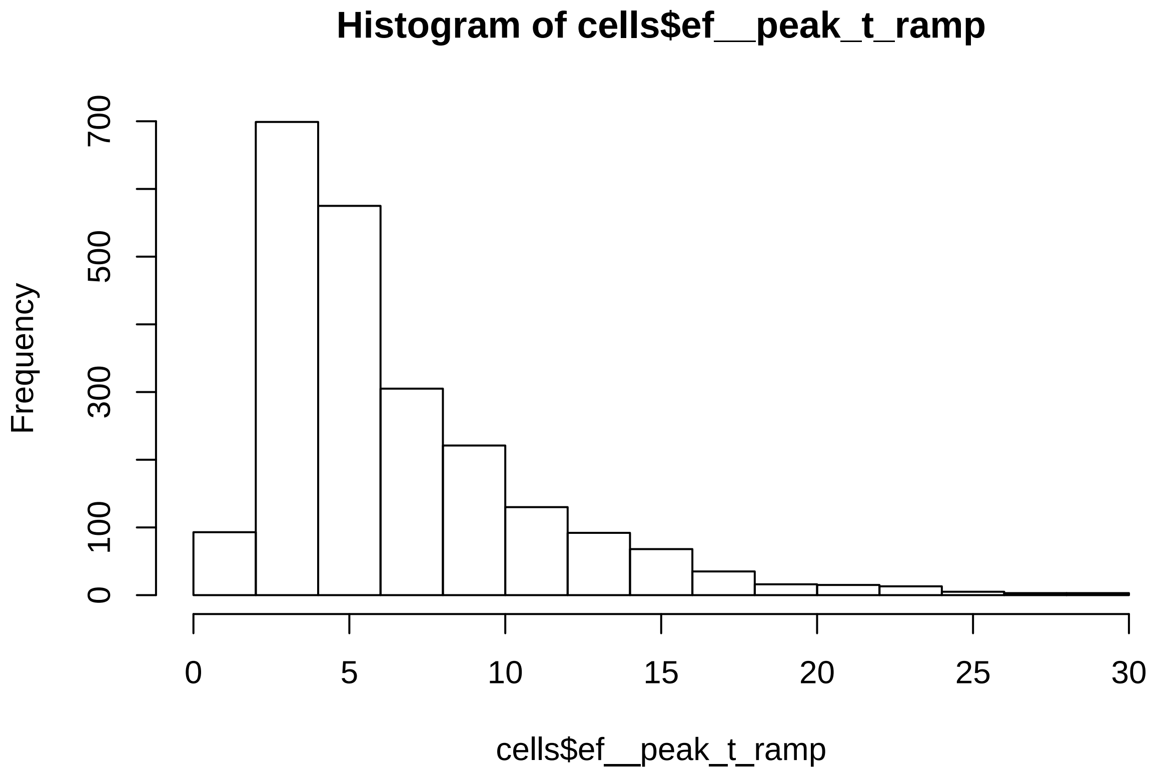
```
sum(human) / nrow(cells)
```

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

6.3

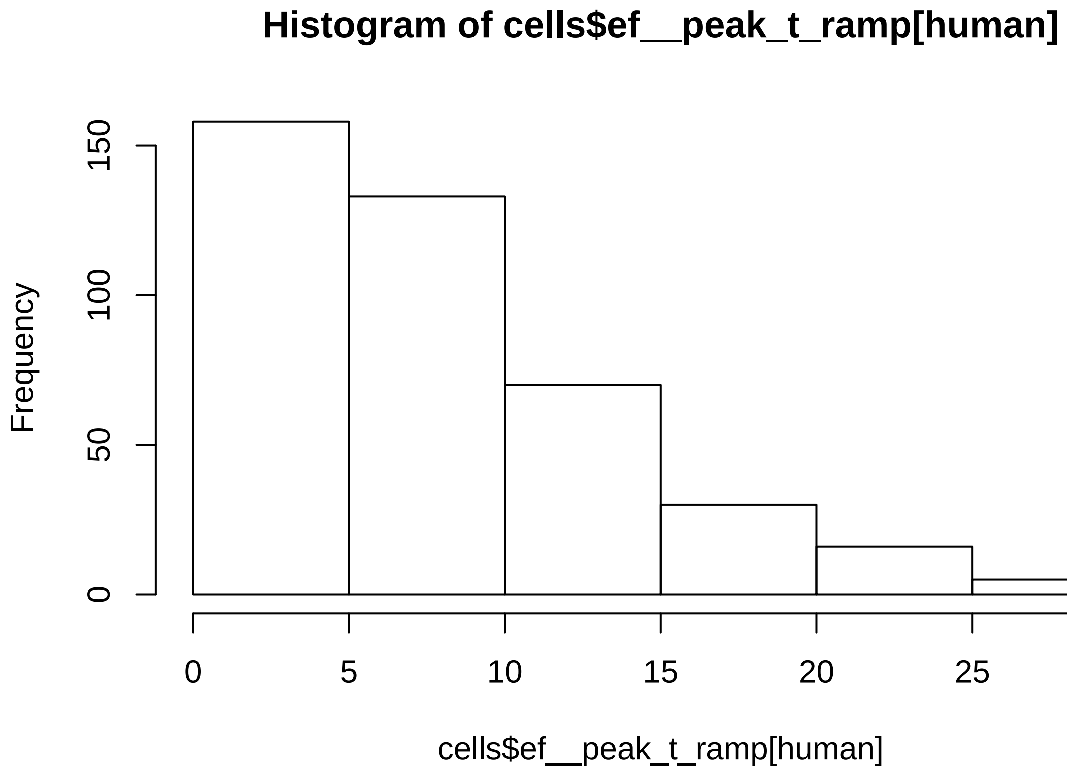
Histogram for the ramp spike time,

```
hist(cells$ef__peak_t_ramp)
```



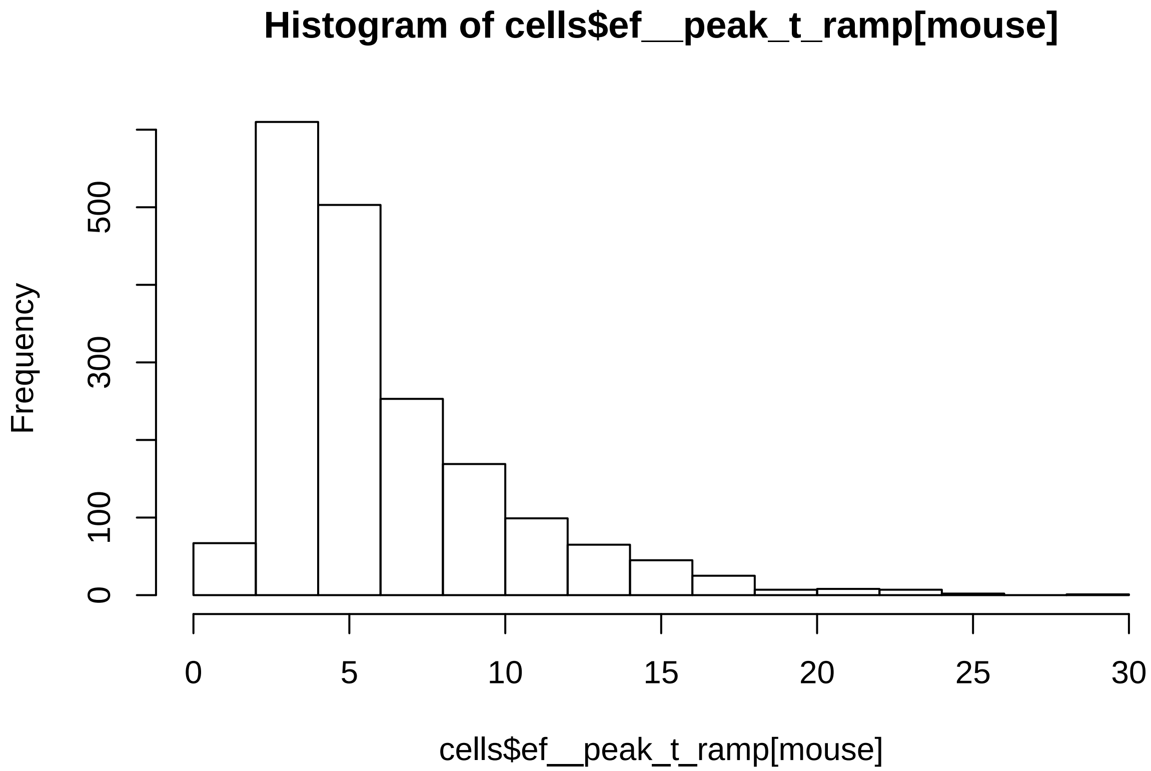
Separately for human cells,

```
hist(cells$ef__peak_t_ramp[human])
```



and mice,

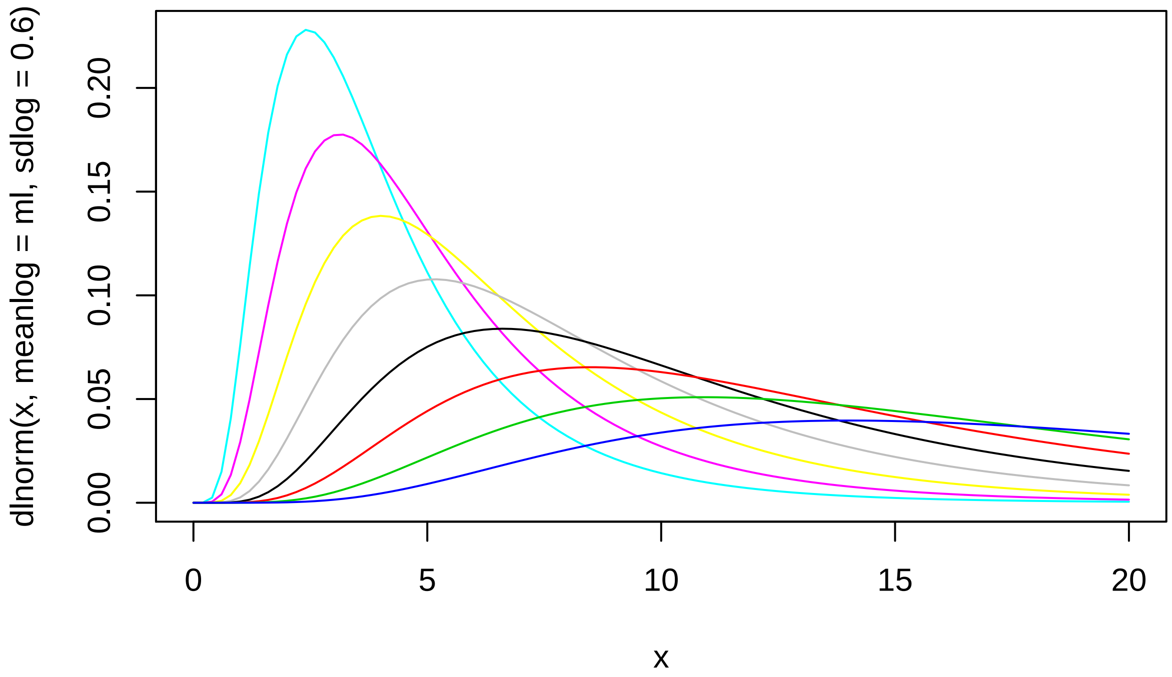
```
hist(cells$ef__peak_t_ramp[mouse])
```



6.4

Plotting log-normal densities,

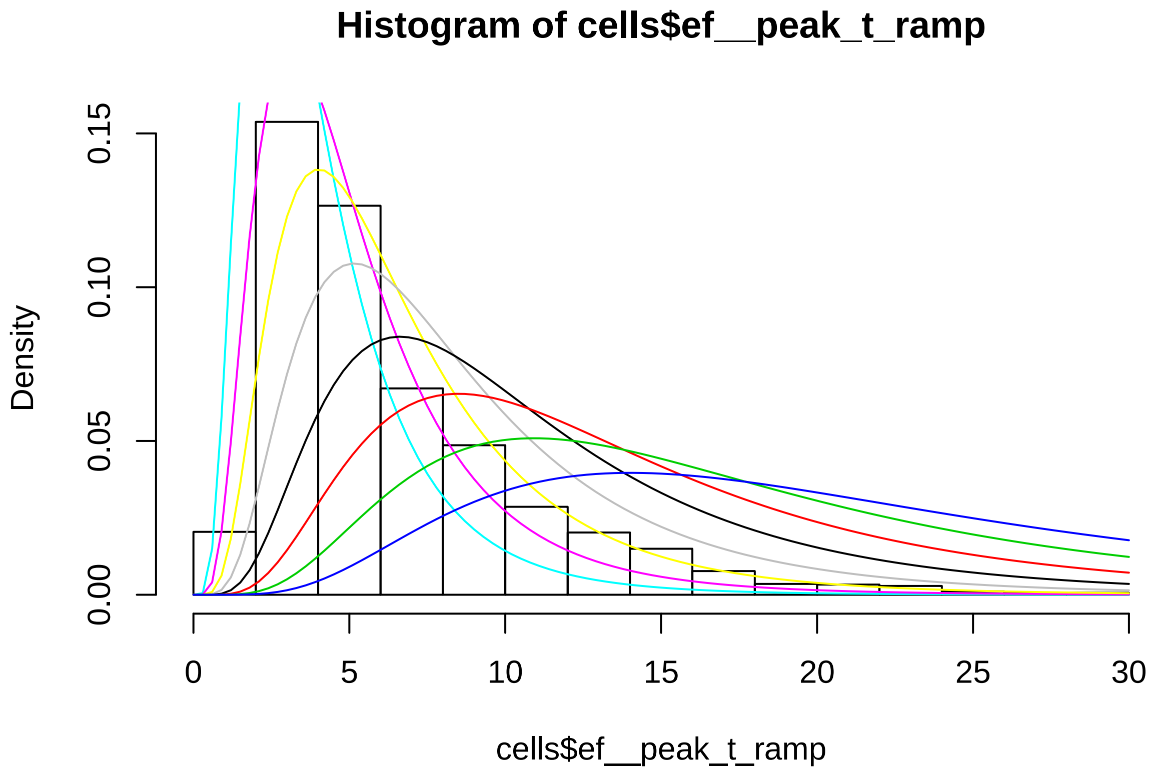
```
firstplot <- F
for (ml in seq(1.25, 3, 0.25)){
  curve(dlnorm(x,meanlog =  ml, sdlog = 0.6),
        from = 0, to = 20, add = firstplot, col = ml*4 )
  firstplot <- TRUE
}
```



6.5

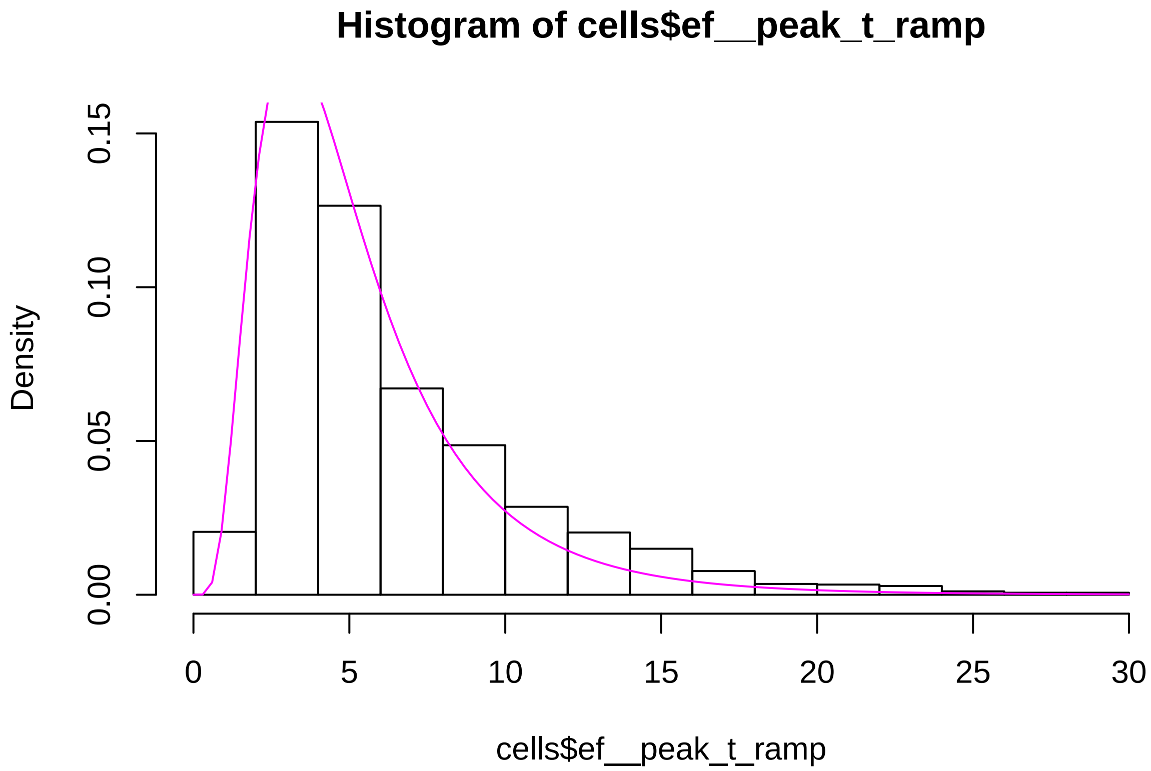
Plot densities on top of histogram,

```
hist(cells$ef__peak_t_ramp, freq = FALSE)
for (ml in seq(1.25, 3, 0.25)){
  curve(dlnorm(x,meanlog =  ml, sdlog = 0.6),
        add = TRUE, col = ml*4 )
 }
```

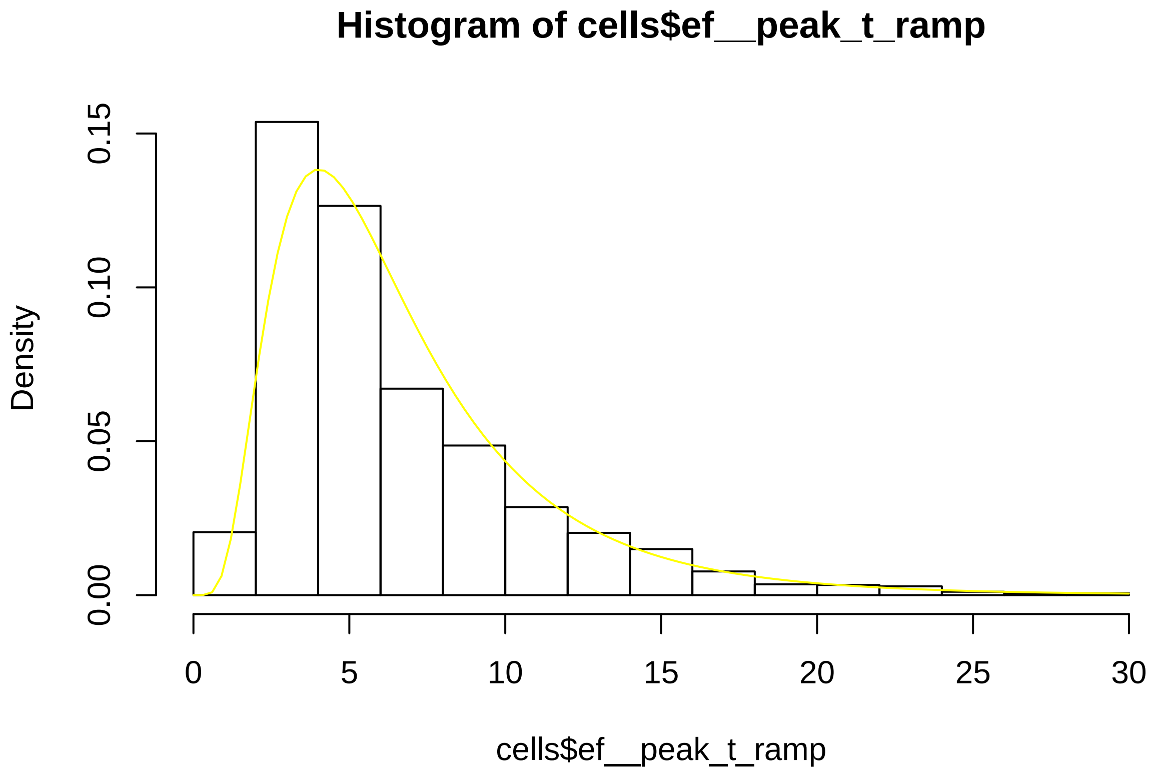


Probably meanlog = 1.5 or 1.75

```
ml <- 1.50
hist(cells$ef__peak_t_ramp, freq = FALSE)
curve(dlnorm(x,meanlog =  ml, sdlog = 0.6),
        add = TRUE, col = ml * 4)
```



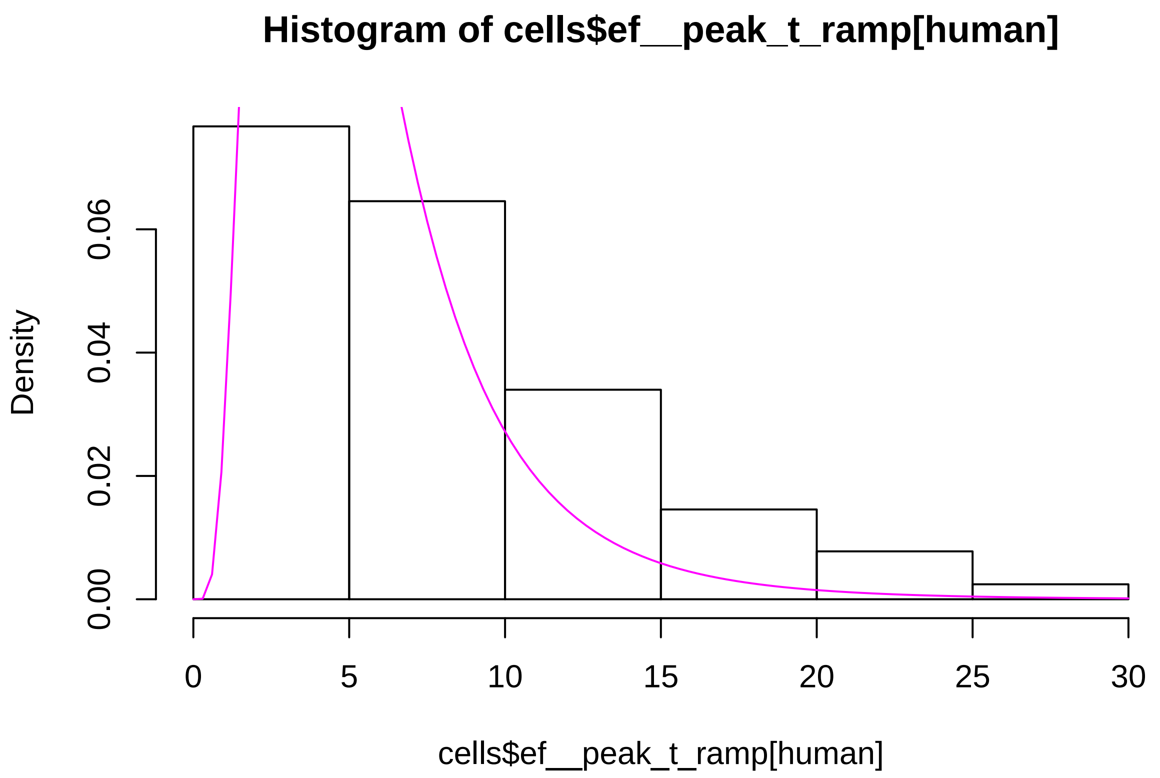
```
ml <- 1.75
hist(cells$ef__peak_t_ramp, freq = FALSE)
curve(dlnorm(x,meanlog =  ml, sdlog = 0.6),
        add = TRUE, col = ml * 4)
```



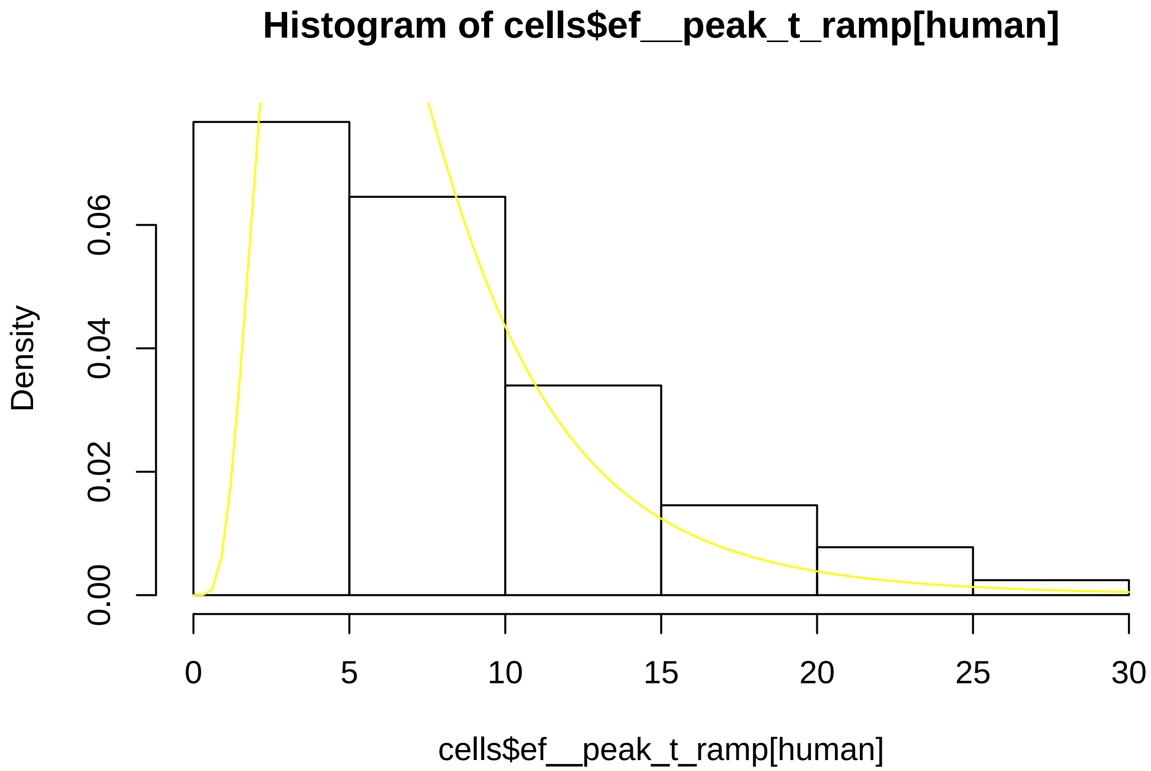
6.6

Only on top of the human observations,

```
ml <- 1.50
hist(cells$ef__peak_t_ramp[human], freq = FALSE)
curve(dlnorm(x,meanlog =  ml, sdlog = 0.6),
        add = TRUE, col = ml * 4)
```



```r
ml <- 1.75
hist(cells$ef__peak_t_ramp[human], freq = FALSE)
curve(dlnorm(x,meanlog =  ml, sdlog = 0.6),
        add = TRUE, col = ml * 4)
```


Histogram of cells$ef__peak_t_ramp[human]
Density
0.00
0.02
0.04
0.06
0
5
10
15
20
25
30
cells$ef__peak_t_ramp[human]