Homework

Statistics warm-up

Consider a random variable X with a probability density function

$$f(x) = \frac{c}{\sqrt{x(b-x)}}, \quad 0 < x < b,$$

where c is a normalising constant and b is a parameter.

- Find c such that the probability density function is valid.
- Find the corresponding cumulative distribution function, $F(x)$ and its inverse $F^{-1}(x)$.
- Describe a procedure to generate samples distributed according to X, given a set of uniformly distributed samples $u_1, \dots, u_n \sim \mathcal{U}[0,1]$.
- Set the seed to 42 and generate 1,000 samples distributed according to X using 1,000 uniformly distributed i.i.d. $\mathcal{U}[0,1]$ samples, with $b = 5$. Plot a histogram of the resulting samples.
- Assume that we have a set of samples $X_1, \dots, X_n$ which are i.i.d. as X with b unknown. Find the log-likelihood function for the parameter b, $\ell(b; x_1, \dots, x_n)$ and find its derivative with respect to b, $\frac{d\ell}{db}$.
- Based on the likelihood, or otherwise, give a statistic based on $X_1, \dots, X_n$ which is a reasonable estimator for the unknown parameter b. Explain your choice.
- The file `samples.rds` contains samples $X_1, \dots, X_n$ distributed according to X. Report the estimated value of $\hat{b}$ based on those samples.

Arrays

You are given a multi-dimensional array `array.rds`. It contains penetrance curves for various cancers and genes. In simple terms, penetrances are how likely one will develop a cancer given that they have a certain corresponding gene mutation. Other variables in `data` describe different sub-populations. For example, the probabilities corresponding to `Brain` cancer and gene `APC` are the probabilities for which a person will develop brain cancer, given that they have a mutation in the `APC` gene.

- Read in the data and give the dimensions of the array.

```r
array <- readRDS("array.rds")
dim(array) # 18 26  8  2 94
```

```
## [1] 18 26  8  2 94
```

```r
str(array)
```

```
##  num [1:18, 1:26, 1:8, 1:2, 1:94] 3.98e-05 2.80e-07 0.00 5.00e-08 0.00 ...
##  - attr(*, "dimnames")=List of 5
##   ..$ Cancer: chr [1:18] "Brain" "Breast" "Cervical" "Colorectal" ...
##   ..$ Gene  : chr [1:26] "APC" "ATM" "BARD1" "BMPR1A" ...
##   ..$ Race  : chr [1:8] "All_Races" "AIAN" "Asian" "Black" ...
##   ..$ Sex   : chr [1:2] "Female" "Male"
##   ..$ Age   : chr [1:94] "1" "2" "3" "4" ...
```

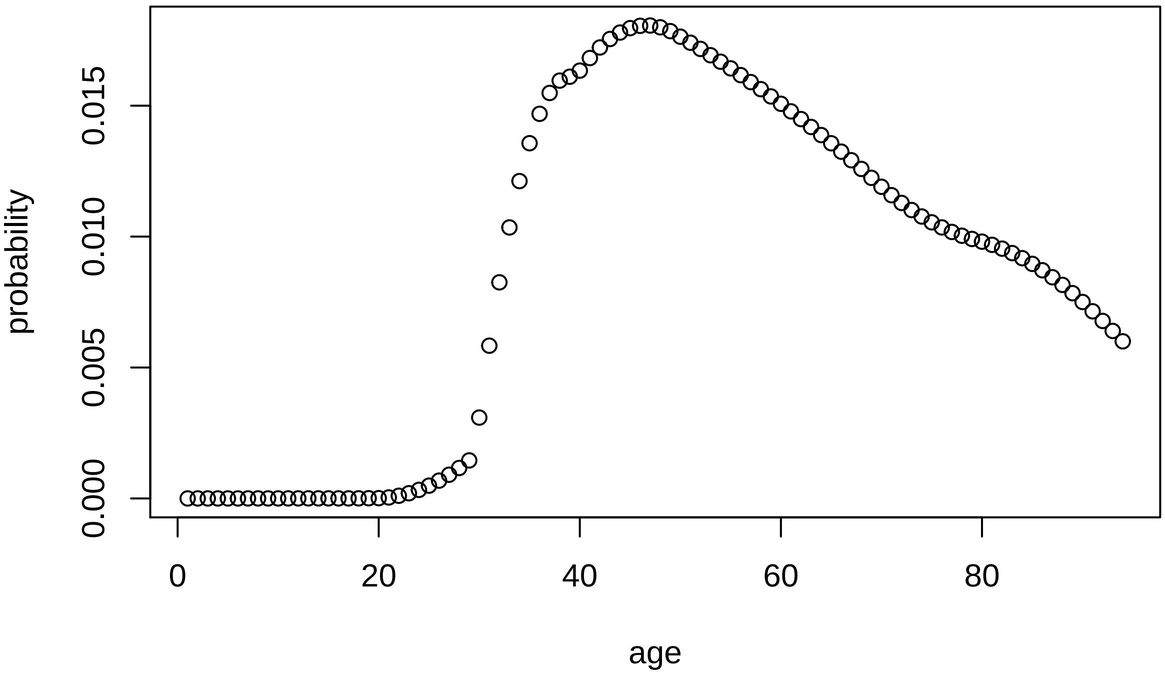
Hint: Use the `str` function.

- Subset the array for the penetrances associated with `Breast` cancer and the `BRCA2` gene for a female with the default race `All_Races`. Then plot the penetrance curve (probability versus age).

```
array_sub <- array[c("Breast"),c("BRCA2"),c("All_Races"),c("Female"),]
length(array_sub)
```

```
## [1] 94
```

```
plot(1:94,array_sub,xlab="age", ylab="probability")
```



- Subset the array for the penetrances associated with `Colorectal` cancer and the `PALB2` gene for an Asian male. What is the probability that a person from this subpopulation at age 50 will develop colorectal cancer in the next 10 years given that he has tested positive for a PALB2 mutation but is otherwise disease free?

```
array_sub_2 <- array[c("Colorectal"),c("PALB2"),c("Asian"),c("Male"),]

risk_sum <- 0
array_sub_2[50]
```

```
##         50
## 0.00051693
```

```
for (i in 50:60){
  risk_year <- array_sub_2[i]
  risk_sum <- risk_sum+risk_year
}
risk_sum
```

```
##         50
## 0.00812721
```

Hint: The probability over a period of time is calculated by summing the yearly risks.

Family pedigrees

Read in the `.rdata` file `pedigree.rda`. Each `data.frame` represents a family. Each individual is uniquely identified by the first column called `ID`. Their sex is coded as `0` for females and `1` for males. Individuals' mother and father are indicated in the `MotherID` and `FatherID` columns. A value of `NA` in these columns means that this person is a so-called 'founder' or that a certain parent is missing.

Each pedigree can be thought of as a family tree. For example, a visualisation of a sample pedigree is shown below. The colours indicate affliction status for cancers as labelled in the legend.

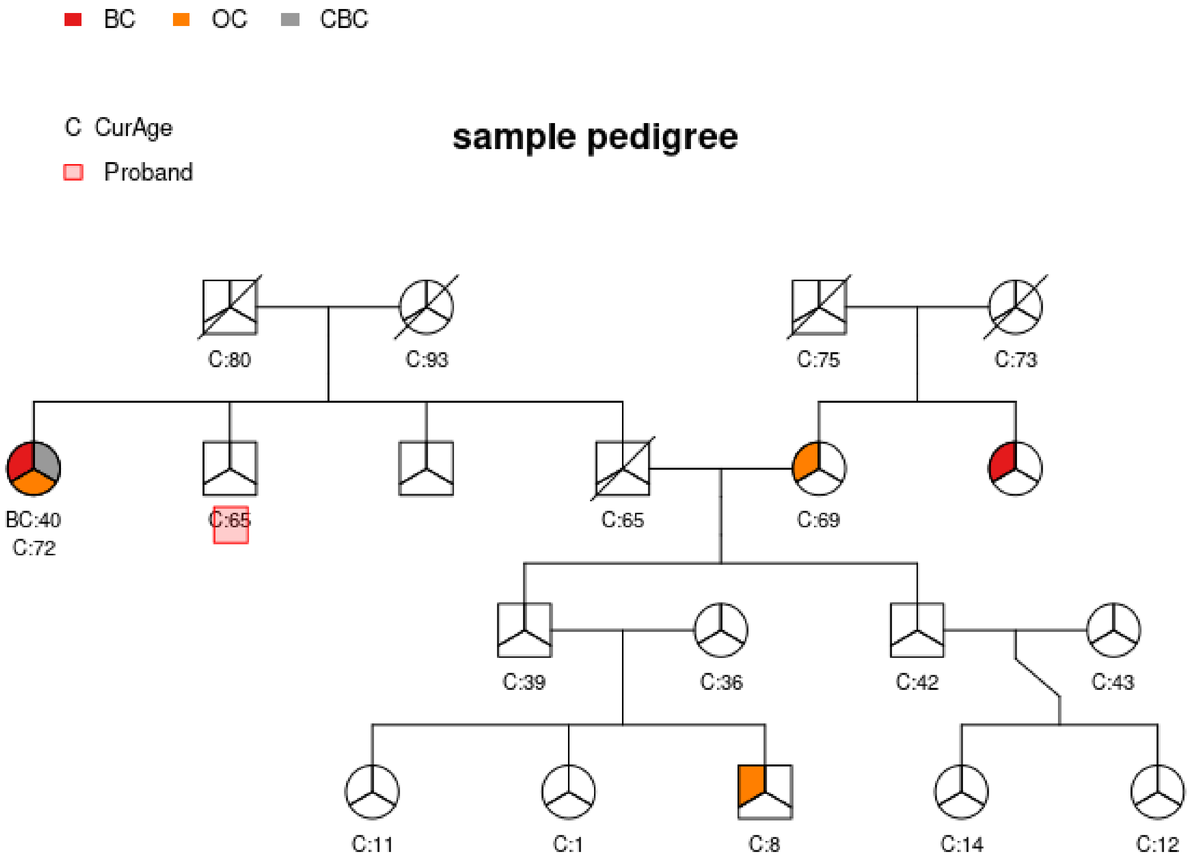


Figure 1:

In the following exercises, you are encouraged to modularise and comment on your code.

```r
pedi <- load("pedigree.rda")
# str(pedi)
# fam10
# fam50
# fam75
# fam100
```

- Write an R function(s) to count the number of unique nuclear families there are in a certain pedigree. A nuclear family is defined as the set of two parents and all of their children.

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.0 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
# fam10
# fam10_new <- fam10 %>% unite("pair",MotherID:FatherID, sep=":",na.rm=TRUE)
# fam50
# fam50_new <- fam50 %>% unite("pair",MotherID:FatherID, sep=":",na.rm=TRUE)
# unique(fam10_new$pair)
# length(unique(fam10_new$pair))
# fam100_new <- fam100 %>% unite("pair",MotherID:FatherID, sep=":",na.rm=TRUE)
# length(unique(fam100_new$pair))
# rowSums(is.na(fam100[,c("MotherID","FatherID")]))
family_count <- function(data){
  data <- data %>% unite("pair",MotherID:FatherID, sep=":",na.rm=TRUE,remove=FALSE)
  if (rowSums(is.na(data[,c("MotherID","FatherID")])) %in% c(2)){
    count <- length(unique(data$pair))-1
  }
  else
  {
    count <- length(unique(data$pair))
  }
  return(count)
}
```

```
# fam10$MotherID[1:9][fam10$FatherID==2]
family_count_update <- function(data){
  data$count <- NA
  if (rowSums(is.na(data[1,c("MotherID","FatherID")]))==2){
    data$count[1] <- 0
  }
  else{data$count[1] <- 1}

  for (i in 2:nrow(data)){

    if(rowSums(is.na(data[i,c("MotherID","FatherID")]))==2){
      data$count[i] <- data$count[i-1]
    }
  else{
    if (data$MotherID[i] %in% data$MotherID[1:i]){
      if (data$FatherID[i] %in% data$FatherID[1:i-1][data$MotherID==data$MotherID[i]]){
        data$count[i] <- data$count[i-1]
      }
      else{
        data$count[i] <- data$count[i-1]+1
      }
    }
    else{data$count[i] <- data$count[i-1]+1}
  }
  }
return(data$count[nrow(data)])
}
family_count_update(fam10)
```

```
## [1] 4
```

```
family_count_update(fam50)
```

```
## [1] 10
```

```
# family_count(fam100)
family_count_update(fam75)
```

```
## [1] 18
```

```
family_count_update(fam100)
```

```
## [1] 26
```

- Report the number of nuclear families for the pedigrees contained in the `.rda` file.

fam10, fam50, fam75 and fam100 has 4, 10, 18 and 26 nuclear families.