# Statistics for (detrital) thermochronology

Pieter Vermeesch

3 September 2023

## Contents



# 1 Introduction to R

This short course will include some practical exercises that use the programming environment R. You can download the teaching materials from https://github.com/pvermees/Thermo2025/, including a copy of these notes, example input files and an auxiliary script called helper.R with R functions to generate synthetic fission track and U-Th-He data. This section of the handout provides the basic knowledge of R that is needed to use these resources.

## 1.1 RStudio

R is a free and open statistical programming language that was inspired by the commercial S language. R is popular for data science. We will use the RStudio IDE (Integrated Development Environment) to interact with R because it provides a uniform user experience across a wide range of operating systems,

including Windows, MacOS and Linux (Figure 1). R and RStudio can be downloaded free of charge from https://posit.co/downloads/.

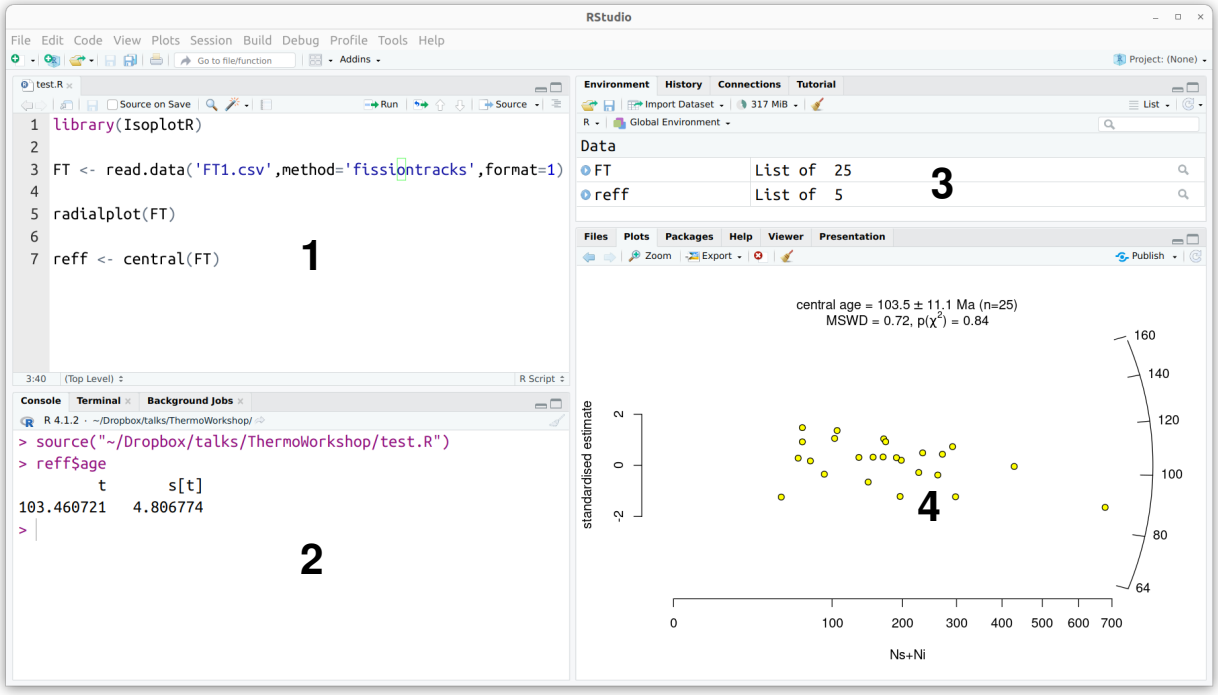


Figure 1: RStudio has four sub-windows: 1) a scripting editor, 2) the console, 3) the workspace environment and command history, and 4) plots, file browser and documentation viewer.

## 1.2 IsoplotR

IsoplotR is an R package for radiometric geochronology (Vermeesch 2018). It includes functionality for fission tracks, U-Th-He and many other chronometers. The package can be installed from the Comprehensive R Archive Network (CRAN), by entering the following command in the console:

```
install.packages("IsoplotR")
```

Additionally, a graphical user interface for IsoplotR can be installed as:

```
install.packages("IsoplotRgui")
```

Once the package has been installed (which only needs to be done once), it can be added to the R workspace by running the following command at the console (which needs to be done for every R-session that uses the package):

```
library("IsoplotR")
```

IsoplotR can be used in three different ways:

1. Online, at https://isoplotr.es.ucl.ac.uk
2. Offline, by entering the following code in the console:

   ```
   IsoplotRgui::IsoplotR()
   ```

3. Using the command-line API (Applications Programming Interface).

The remainder of this tutorial will introduce the API for `IsoplotR` (and another package called `provenance`), which enables power users to create automation scripts and extend the functionality of the two packages beyond the capabilities of the visual interfaces.

## 1.3 Basic R

1. We will start this tutorial from the `R` console (panel 1 in Figure 1). First, do some arithmetic:

```
1 + 1
```

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

`R` prints the result (`2` in this case) to the console[1]. Here are some other arithmetic operations:

```
sqrt(2)
```

```
## [1] 1.414214
```

```
exp(log(10))
```

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

```
13%%5
```

```
## [1] 3
```

2. An arrow operator is used to assign a value to a variable. Note that the arrow can point both ways:

```
foo <- 2
4 -> bar
foo <- foo*bar
foo
```

```
## [1] 8
```

3. Create a vector of numbers:

```
myvec <- c(2,4,6,8)
myvec*2
```

```
## [1]  4  8 12 16
```

Query the third value of the vector:

```
myvec[3]
```

Change the third value of the vector:

```
myvec[3] <- 100
```

Change the second and the third value of the vector:

```
myvec[c(2,3)] <- c(100,101)
```

Create a sequence of numbers:

```
seq(from=1,to=10,by=1)
```

```
##  [1]  1  2  3  4  5  6  7  8  9 10
```

Equivalently:

[1] The save space, the notes will not always reproduce the output.

```
seq(1,10,1)
seq(1,10)
seq(to=10,by=1,from=1)
seq(to=10)
1:10
```

Create a 10-element vector of twos:

```
rep(2,10)
```

4. Create a 2 × 4 matrix of ones:

```
mymat <- matrix(1,nrow=2,ncol=4)
```

Change the third value in the first row of `mymat` to 3:

```
mymat[1,3] <- 3
```

Change the entire second column of `mymat` to 2:

```
mymat[,2] <- 2
```

Remove the first column from `mymat`:

```
mymat[,-1]
```

```
##      [,1] [,2] [,3]
## [1,]    2    3    1
## [2,]    2    1    1
```

Give names to the rows:

```
rownames(mymat) <- c('first','second')
```

Use the names:

```
mymat['first',]
```

```
## [1] 1 2 3 1
```

The transpose of `mymat`:

```
t(mymat)
```

```
##      first second
## [1,]     1      1
## [2,]     2      2
## [3,]     3      1
## [4,]     1      1
```

Element-wise multiplication (`*`) vs. matrix multiplication (`|%*%|`):

```
mymat*mymat
```

```
##        [,1] [,2] [,3] [,4]
## first     1    4    9    1
## second    1    4    1    1
```

```
p <- mymat %*% t(mymat)
p
```

```
##        first second
## first     15      9
```

```
## second     9      7
```

The inverse and determinant of a square matrix:

```
invp <- solve(p)
det(invp %*% p)
```

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

5. Lists are used to store more complex data objects:

```
mylist <- list(v=myvec,m=mymat,nine=9)
mylist$v
```

```
## [1] 2 4 6 8
```

or, equivalently:

```
mylist[[1]]
mylist[['v']]
```

Data frames are list-like tables:

```
myframe <- data.frame(
    period=c('Cz','Mz','Pz','PC'),
    SrSr=c(0.708,0.707,0.709,0.708),
    fossils=c(TRUE,TRUE,TRUE,FALSE)
    )
myframe
```

```
##   period  SrSr fossils
## 1     Cz 0.708    TRUE
## 2     Mz 0.707    TRUE
## 3     Pz 0.709    TRUE
## 4     PC 0.708   FALSE
```

You can access the items in `myframe` either like a list or like a matrix:

```
myframe$period == myframe[,'period']
```

```
## [1] TRUE TRUE TRUE TRUE
```

6. Save data to a text (`.csv`) file:

```
write.csv(myframe,file='timescale.csv',row.names=FALSE)
```

Read data from a `.csv` file:

```
myframe2 <- read.csv(file='timescale.csv',header=TRUE)
```

Type `myframe2` at the console to verify that the contents of this new variable match those of `myframe`.

7. Plot the first against the second row of `mymat`:

```
plot(x=mymat[1,],y=mymat[2,])
```

Draw lines between the points shown on the existing plot:

```
lines(mymat[1,],mymat[2,])
```

Create a new plot with red lines but no points and a 1:1 aspect ratio for the X- and Y-axis:

```
plot(mymat[1,],mymat[2,],type='l',col='red',asp=1)
```

Save the currently active plot as a vector-editable `.pdf` file:

```
dev.copy2pdf(file="trigonometry.pdf")
```

8. If you want to learn more about a function, type `help` or `?`:

```
help(c)
?plot
```

You can also define your own functions:

```
cube <- function(n){ return(n^3) }
```

Using the newly created function:

```
cube(2)
```

```
## [1] 8
```

```
result <- cube(3)
```

9. Type the following lines in RStudio's file editor (panel 1 of Figure 1) and save them in a file called `myscript.R`:

```
# the 'print' function is needed to show intermediate
# results when running commands from a .R file
print(pi)
```

You can run this code by going back to the console and typing:

```
source("myscript.R")
```

```
## [1] 3.141593
```

Note that everything that follows the `#`-symbol is ignored by `R`.

10. Conditional statements. Replace the contents of `myscript.R` with:

```
toss <- function(){
 r <- runif(1) # create a random number between 0 and 1
     if (r<0.5){
        print("head")
     } else {
        print("tail")
     }
}
```

Save and run in the console:

```
source('myscript.R')
toss()
```

11. Add the following function to `myscript.R`:

```
fibonnaci <- function(n=5){ # 5 is the default value
   if (n < 3) { stop("n must be at least 3") }
   # seed the output vector with 0 and 1:
   s <- c(0,1)
   # loop through all numbers from 3 to n:
   for (i in 3:n){
      s[i] <- s[i-1] + s[i-2]
   }
```

```
    return(s)
}
```

Save and run in the console to calculate the first `n` numbers in the Fibonnaci series:

```
source('myscript.R')
fibonnaci(10)
```

# 2 Ages vs. dates

In statistics, an important distinction is made between the *true* values of estimated parameters, and their *estimates*. In geochronology, these concepts are referred to as *ages* and *dates*, respectively[2].

## 2.1 Age estimation and error propagation

Consider the following fission track data:

| true values | measured data |
|---|---|
| $\rho_s = 1 \times 10^6$ cm$^{-2}$ | $N_s = 15$ |
| $\rho_i = 2 \times 10^6$ cm$^{-2}$ | $N_i = 42$ |
| $\rho_d = 2.5 \times 10^6$ cm$^{-2}$ | $\hat{\rho}_d = 2.55 \pm 0.05 \times 10^6$ cm$^{-2}$ |
| $\zeta = 350$ yr$^{-1}$cm$^2$ | $\hat{\zeta} = 355 \pm 10$ yr$^{-1}$cm$^2$ |

This gives rise to an age of:

$$t = \frac{1}{\lambda_{238}} \ln\left(1 + \frac{1}{2}\lambda_{238}\zeta\rho_d\frac{\rho_s}{\rho_i}\right) = 215 \text{ Ma} \tag{1}$$

and a date of:

$$\hat{t} = \frac{1}{\lambda_{238}} \ln\left(1 + \frac{1}{2}\lambda_{238}\hat{\zeta}\hat{\rho}_d\frac{N_s}{N_i}\right) = 160 \text{ Ma} \tag{2}$$

It is impossible to use the date without some estimate of its *precision*, which provides a way to guess its likely proximity to the true age. For the fission track method, the analytical uncertainty (standard error) of a date is estimated as follows:

$$s[\hat{t}] = \hat{t}\sqrt{\frac{1}{N_s} + \frac{1}{N_i} + \left(\frac{s[\hat{\zeta}]}{\hat{\zeta}}\right)^2 + \left(\frac{s[\hat{\rho}_d]}{\hat{\rho}_d}\right)^2} = 48 \text{ Ma} \tag{3}$$

The U-Th-He method is based on the following ingrowth equation:

$$\begin{aligned} [\text{He}] &= a(t)[\text{U}] + b(t)[\text{Th}] \\ \text{where } a(t) &= 8\frac{137.818}{138.818}(e^{\lambda_{238}t} - 1) + 7\frac{1}{138.818}(e^{\lambda_{235}t} - 1) \\ \text{and } b(t) &= 6(e^{\lambda_{232}t} - 1) \end{aligned} \tag{4}$$

in which [U], [Th] and [He] have the same atomic units (e.g., fmol, µmol/g). Equation 4 does not have an analytical solution but can be solved iteratively. Error propagation is done by first order Taylor approximation:

$$s[\hat{t}] = \frac{\sqrt{a(\hat{t})^2 s[\text{U}]^2 + 2a(\hat{t})b(\hat{t})s[\text{U,Th}] + b(\hat{t})^2 s[\text{Th}]^2 + s[\text{He}]^2}}{8\frac{137.818}{138.818}\lambda_{238}e^{\lambda_{238}\hat{t}} + 7\frac{\lambda_{235}}{138.818}e^{\lambda_{235}\hat{t}}} \tag{5}$$

where $s$[U,Th] is the covariance of the U and Th measurements.

[2] In reality, the terms 'age' and 'date' are often used interchangeably, because their meaning is usually clear from the context.

Using IsoplotR, geochronological data can be loaded into memory with the read.data() function. Before running following code, it is important to add IsoplotR to the workspace and set the working directory to the location of your input files. This can either be done by clicking on the Session → Set Working Directory menu in RStudio and ticking the IsoplotR box in the Packages tab (fourth panel of Figure 1), or by entering the following commands at the beginning of your R script:

```
library(IsoplotR)
setwd("/home/pieter/Thermo2023") # replace the string with your own path
```

Then load the FT.csv file into memory as follows:

```
FTdat <- read.data("data/FT.csv",method="fissiontracks")
```

Calculate the dates[3] and uncertainties:

```
FTdates <- age(FTdat)
head(FTdates) # print the first few rows
```

```
##              t    err[t]
## [1,] 124.73242 22.962810
## [2,] 111.33252 18.014183
## [3,]  87.67540  9.001938
## [4,] 110.56597 34.352213
## [5,]  89.03396 19.546294
## [6,]  85.52453 13.365990
```

Similarly, for U-Th-He data:

```
Hedat <- read.data("data/UThHe.csv",method="U-Th-He")
Hedates <- age(Hedat)
```

## 2.2 Populations

A single date is not enough for geochronology in general, and for detrital thermochronology in particular. As we saw at the end of the previous section, the age() function returns a table of age estimates and uncertainties. This section will introduce three ways to visualise distributions of multiple dates. Let us consider the following five age *populations*, where the word 'population' refers to the (unknowable) distribution of the true ages that would be observed if an infinite number of aliquots (i.e. apatite crystals) were dated with infinite precision and no bias:

1. A single discrete age component: the true ages of all the apatite crystals from this sample are exactly 100 Ma.
2. A continuous distribution of ages: the true ages span a range of values, with most grains being ca. 100 Ma, but some being younger or older.
3. A discrete mixture of two age components: half of the sample consists of 100 Ma grains, the other half of the crystals are 200 Ma.
4. A mixture of discrete and continuous age components: 50% of the grains have a true age of 50 Ma, and the remaining 50% have a range of ages around 100 Ma. This population contains no true ages younger than 50 Ma, so the older subpopulation is *truncated* at 50 Ma.
5. A mixture of continuous age components: 50% of the grains are derived from a continuous population with a true fission ages around 100 Ma, whereas the remaining 50% are derived from a 300 Ma peak, with tails to higher and lower values. Note that, unlike population 4, crystals from the young peak are allowed to be older than those from the old peak and vice versa.

[3] IsoplotR uses the age() function to calculate dates because date() already exists as an R function!

The Probability Density Functions (PDFs) of the five populations are shown in Figure 2.

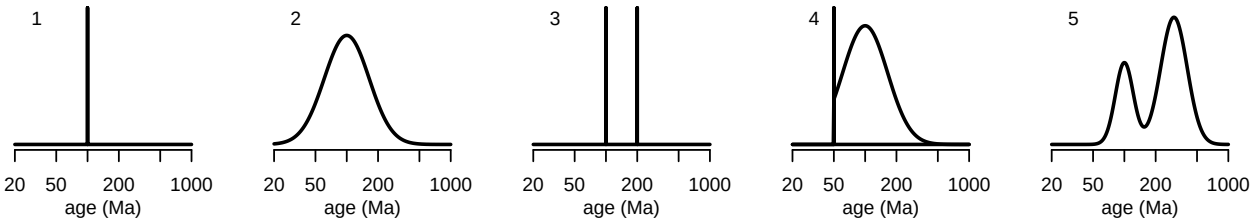


Figure 2: Five theoretical age populations.

You can reproduce these plots using the `getpop()` function of the auxiliary `helper.R` script. For example:

```
source("helper.R")
getpop(pop=4,plot="ages")
```

The PDFs of the age populations (Figure 2) are unknowable truths, the essential features of which we must try to infer from a finite number of imprecisely dated samples. It is important to note that the distribution of the dates is not the same as the distribution of the ages. Let us explore this crucial fact with another function of `helper.R`, called `FTsamp()`. This function can be used to draw synthetic samples from the populations of Figure 2. For example, to draw a 50-grain fission track sample from population 1:

```
FTsample <- FTsamp(pop=1,ng=50)
```

`FTsamp()` returns a list containing the following items:

```
names(FTsample)
```

```
## [1] "x"      "rhoD"   "zeta"   "format" "t.true"
```

`FTsample$t.true` is a vector containing the true ages. Inspecting this list at the console for the above example reveals that it contains 50 identical values of 100 Ma, exactly as expected from Figure 2:

```
head(FTsample$t.true)
```

```
## [1] 100 100 100 100 100 100
```

`FTsample$x` contains random realisations of the spontaneous and induced track counts for the 50 synthetic grains:

```
head(FTsample$x)
```

```
##      Ns  Ni
## [1,] 16  85
## [2,] 20  80
## [3,] 14 132
## [4,] 11  74
## [5,] 23  83
## [6,] 16  85
```

`FTsample` is actually a special type of list. It behaves as an 'object' of class `fissiontracks`, which is recognised by `IsoplotR` and can be directly passed on to its `age()` function:

```
FTdates <- age(FTsample)
head(FTdates)
```

```
##              t   err[t]
## [1,]  81.83135 22.30031
## [2,] 108.45749 27.11437
```

```
## [3,]  46.23531 12.99569
## [4,]  64.70793 20.91004
## [5,] 120.10903 28.30255
## [6,]  81.83135 22.30031
```

From the first six entries of `FTdates` you can already see that, whereas the ages are all the same, the dates are not! Importantly, the addition of analytical uncertainty has caused the dates to be *more dispersed* than the ages. `helper.R` also contains a function called `UThHesamp()` to generate synthetic U-Th-He data.

Under certain simplifying assumptions, it is possible to predict the PDF of the dates from the PDF of the ages (Figure 3) by setting `plot="dates"` in `getpop()`:

```
getpop(pop=1,plot="dates",method="fissiontracks")
```

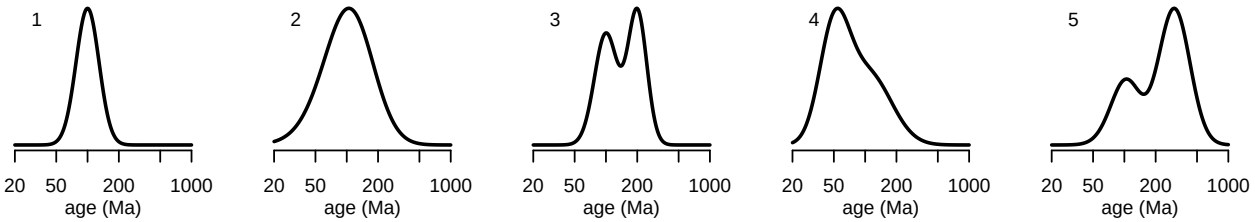


Figure 3: Predicted PDFs of the fission track dates for the age distributions of Figure 2, under the simplifying assumption that $N_s + N_i = 100$.

Similarly, for the U-Th-He method:

```
getpop(pop=1,plot="dates",method="U-Th-He")
```

which produces Figure 4.

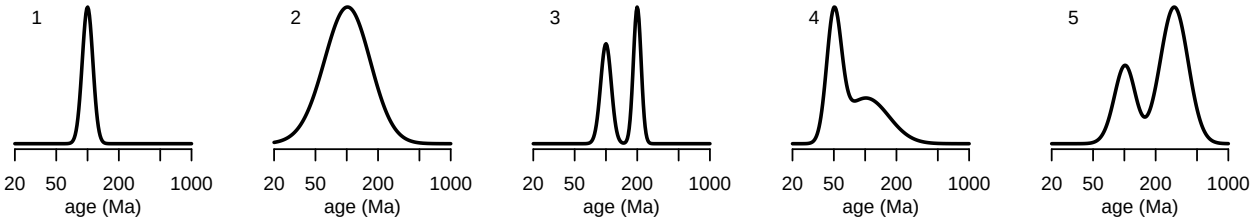


Figure 4: Predicted PDFs of the U-Th-He dates for the age distributions of Figure 2.

Note how the PDFs of the ages (Figure 3) are more similar to the PDFs of the U-Th-He dates (Figure 4) than to the PDFs of the fission track dates (Figure 3). This is because the U-Th-He method exhibits much smaller analytical uncertainties than the fission track method.

# 3 Plotting data

The previous section showed that the PDF of the dates is unknowable, just like the PDF of the ages. This section will introduce three graphical devices that are used to visualise the population of the dates. Section 4 will introduce statistical tools to infer properties of the ages populations from the dates.

## 3.1 Kernel Density Estimates

A KDE is a graphical device that produces a continuous line, which approximates the PDF of the dates from a limited number of values (Vermeesch 2012).

$$\mathrm{KDE}(x) = \frac{1}{nh\sqrt{2\pi}} \sum_{j=1}^{n} \exp\left[-\frac{1}{2}\left(\frac{x-\hat{t}_j}{h}\right)^2\right] \tag{6}$$

where $h$ is the so-called *bandwidth* of the KDE, which serves a similar purpose as the bin width of a histogram. Using `IsoplotR`:

```
kde(FTdates,log=TRUE)
```

where the logarithmic transformation is useful to prevent the KDE to overlap into physically impossible negative values. Collecting five random samples of 25 values from population 1 produces five KDEs that all look slightly different from each other (Figure 5).

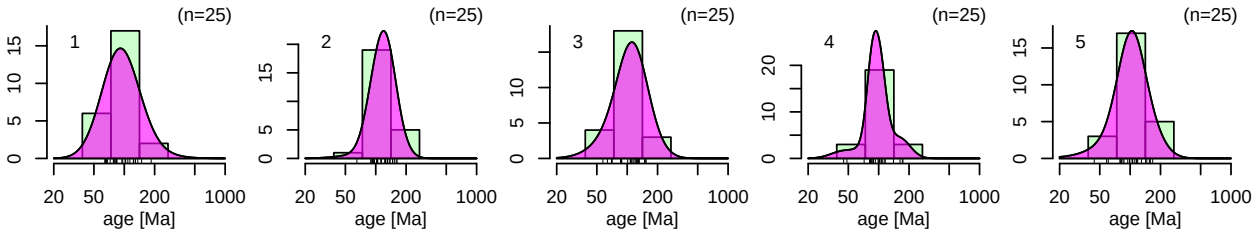


Figure 5: KDEs of five random fission track samples from population 1. To ensure comparability with Figure 3, $N_s + N_i = 100$ for each sample. Although all five samples were drawn from the same population, their KDEs look slightly different.

Generating KDEs for all five populations (Figure 6) produces density estimates that hardly resemble the theoretically predicted PDFs of Figure 3.

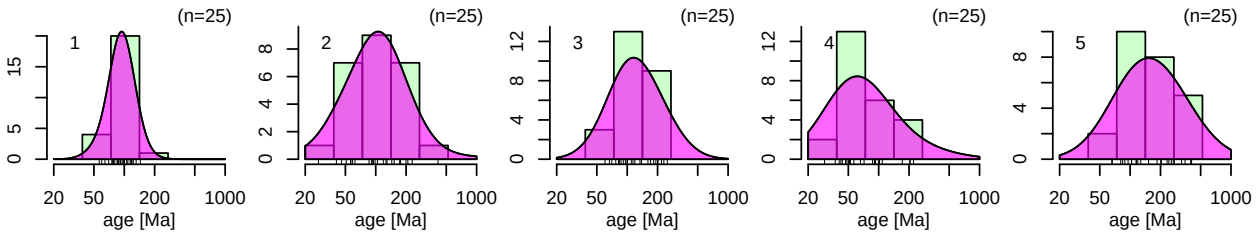


Figure 6: KDEs of five random fission track samples from populations 1 through 5. To ensure comparability with Figure 3, $N_s + N_i = 100$ for each sample. Note how the multimodality of populations 3, 4 and 5 has been smoothed out and lost.

For the U-Th-He method:

```
kde(Hedates,log=TRUE)
```

Repeating this for five random samples from the age populations of Figure 2 yields five KDEs (Figure 7) that resemble the PDFs of Figure 4 to different degrees.

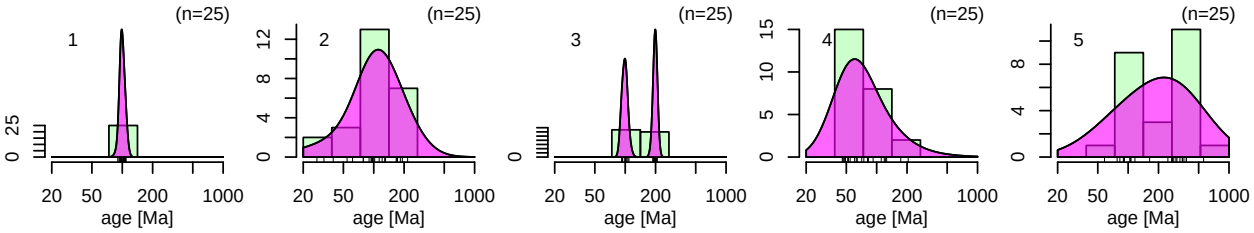


Figure 7: KDEs of five random U-Th-He samples from populations 1 through 5.

The KDEs of the first three samples of Figure 7 resembles their respective populations reasonably well. The same cannot be said about samples 4 and 5, whose KDEs smooth out their bimodal populations.

*Exercise*: Generate some KDEs of your own using `helper.R`'s `FTsamp()` and `UThHesamp()` function, and `IsoplotR`'s `kde()` function. Type `?kde` at the console to view the documentation. Use the optional arguments to complete the following tasks:

1. Construct a KDE for a random fission track sample from population 5, first using a linear scale and then using a logarithmic scale.
2. Create KDEs for U-Th-He population 4 using sample sizes of 20, 100 and 500, plotted on a logarithmic scale.
3. Create KDEs for U-Th-He population 4 using custom bandwidths of 10, 1 and 100 Ma, plotted on a linear scale.

## 3.2 Cumulative Age Distributions

Constructing a KDE requires choosing a kernel bandwidth. Although a plethora of 'automatic' bandwidth selection algorithms exist, none of these work optimally in all cases. This is evident from the mismatch between the shape of the KDEs and PDFs of fission track samples 3, 4 and 5 (Figure 6), and U-Th-He samples 4 and 5 (Figure 7).

The issue of bandwidth selection can be completely avoided by plotting the data as an Empirical Cumulative Distribution Function (ECDF) or Cumulative Age Distribution (CAD):

$$\mathrm{CAD}(x) = \frac{1}{n}\sum_{j=1}^{n} I(\hat{t}_j < x) \tag{7}$$

where $I(*) = 1$ if $*$ is 'true' and $I(*) = 0$ if $*$ is 'false'. Using `IsoplotR`:

```
cad(FTdates)
```

Figure 8 plots the fission track samples of Figure 6 as CADs.

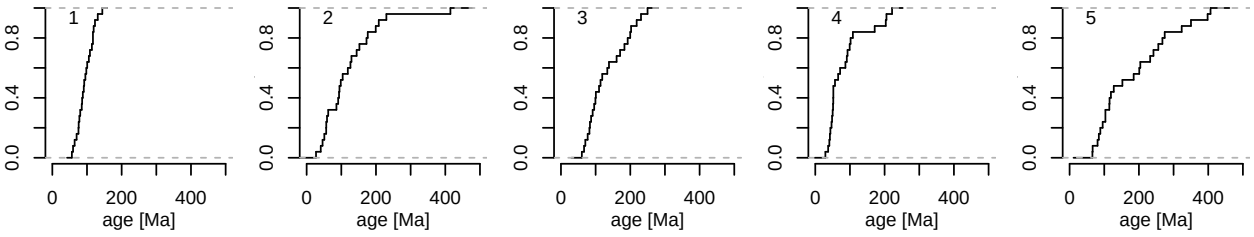


Figure 8: CADs of the samples of Figure 6.

Figure 9 repeats the same exercise for the U-Th-He samples of Figure 7.

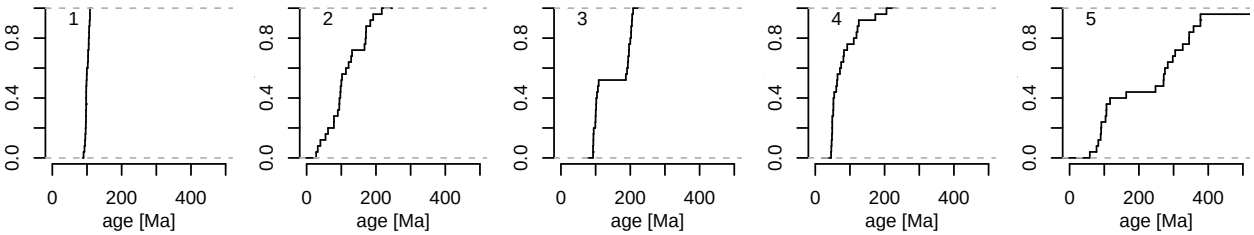


Figure 9: CADs of the samples of Figure 7.

Peaks of the KDEs correspond to steep segments of the cumulative distributions. Because the cumulative step functions do not require kernel smoothing, they make it easier to identify multiple modes in a distribution of noisy data. It is indeed easy to spot the two modes of samples 3, 4 and 5 in Figure 9. Unfortunately, things are not so straightforward for the fission track data of Figure 8.

It is interesting to note that the CAD has a similar appearance to an age-elevation profile. In fact, under certain idealised conditions (zero analytical uncertainties, linear hypsometry and uniform erosion), it can be shown that the CAD of a detrital age population is an unbiased estimator of the age-elevation profile. Of course, in reality, these idealised conditions never exist. Nevertheless, it is still possible to infer useful information about age-elevation profiles from the CAD of detrital fission track and U-Th-He data. Conversely, if the age-elevation profile is known, then deviations of the observed CAD from forward modelled predictions can be used to map the spatial distribution of erosion in a river catchment (Vermeesch 2007).

## 3.3 Radial plots

Sections 3.1 and 3.2 showed that neither KDEs nor CADs are very useful for fission track data. This is because fission track are extremely imprecise and *heteroscedastic*, i.e. their precision varies from grain to grain. Radial plots are an alternative graphical device that was created specifically to deal with this type of data (Galbraith 1988, 1990). Unlike KDEs and CADs, radial plots jointly visualise the values with their analytical uncertainties. A radial plot is a scatter plot of $(x_i, y_i)$ values, where

$$\begin{aligned} x_i &= 1/s[z_i] \\ \text{and } y_i &= \frac{z_i - z_\circ}{s[z_i]} \end{aligned} \tag{8}$$

in which $z_i$ is some transformed version of the data of interest (e.g. the logarithm of the $N_s/N_i$-ratios) and $z_\circ$ is some reference value such as the mean. The slope of a line connecting the origin of this scatter plot with any of the $(x_i, y_i)$s is proportional to $z_i$ and, hence, a function of the date $t_i$. In `IsoplotR`, radial plots are produced with the `radialplot` function:

```
radialplot(FTdat)
```

To understand the interpretation of radial plots, consider the following toy example consisting of four sets of fission track counts:

| # | $N_s$ | $N_i$ | $N_s/N_i$ | $N_s + N_i$ |
|---|---|---|---|---|
| 1 | 10 | 90 | 0.1 | 100 |
| 2 | 50 | 50 | 1 | 100 |
| 3 | 1 | 9 | 0.1 | 10 |
| 4 | 5 | 5 | 1 | 10 |

Table 1: Four synthetic fission track counts. Note how the $N_s/N_i$-ratios of grains 1 and 3 are lower than those of grains 2 and 4; and how the sum of the counts $(N_s + N_i)$ for grains 1 and 2 is greater than those of grains 3 and 4. Therefore, grains 1 and 3 are younger than grains 2 and 4; whilst the dates of grains 1 and 2 are more precise than those of grains 1 and 2.

The radial plot simultaneously displays the dates (which are proportional to the angle of the data relative to

the origin) and their analytical precision (which scales with the horizontal position of the plot symbols in the radial plot (Figure 10).

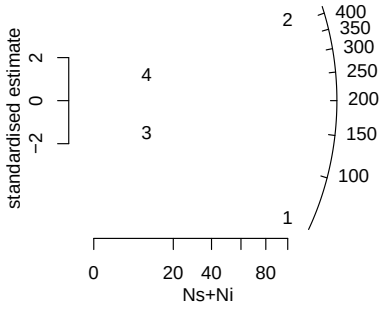


Figure 10: Radial plot for the toy example of Table 1, with $\zeta = 350$ yr cm$^2$ and $\rho_D = 2.5 \times 10^6$ cm$^{-2}$. Connecting the origin of the diagram to the plot symbols for grains 1 and 3 (low $N_s/N_i$-ratios) forms a line with a negative slope, which projects onto the radial scale at a younger date than a line connecting the origin to grains 2 and 4 (high $N_s/N_i$-ratios, positive slope). Grains 1 and 2 (high $N_s + N_i$-values, right hand side of the radial plot) are more precise than grains 3 and 4 (low $N_s + N_i$-values, left hand side of the radial plot).

Plotting five random samples from the synthetic populations of 2 on radial plots (Figure 11) preserves the bimodality of populations 3, 4 and 5 better than the KDEs and CADs of Figures 6 and 8.

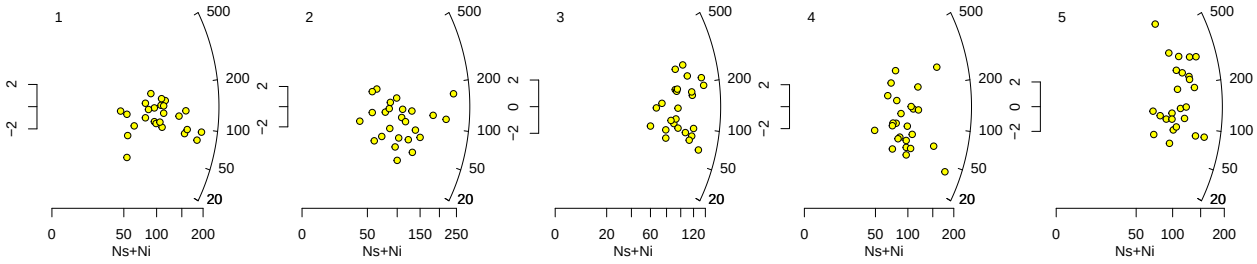


Figure 11: Radial plots for five random fission track samples of the populations shown in Figure 3. Unlike the samples shown in Figures 6 and 8, $N_s + N_i$ was allowed to vary between grains in this example. The bimodality of populations 3, 4 and 5 can more easily be recognised in these diagrams than the KDEs or CADs.

Besides visualising the distribution of the dates, an equally important feature of the radial plot is its ability to visually assess the dispersion of the data. Consider, for example, samples 1 and 2 in Figure 11. Both of these datasets are centred around 100 Ma, and are characterised by similar analytical precision, with $N_s + N_i$ ranging from 50 to 200 for both samples.

The key difference between the two samples is the spread of the dates. The dates of sample 1 range from 50 to 150 Ma, whereas those of sample 2 range from 20 to 300 Ma. The left hand side of the radial plots contains a '2-sigma' error bar around the origin. The plot symbols for sample 1 all fall within a band of this width. In contrast with this, the plot symbols for sample 2 fall outside the 2-sigma band of their radial plot. This provides graphical evidence that sample 2 is *overdispersed* with respect to the analytical uncertainties, whereas sample 1 is not. Section 4.1 will formalise this visual assessment with a statistical text.

*Exercise*: The datasets of Figure 11 are saved in five files called `FT1.csv` through `FT5.csv`.

1. Using these input files, reproduce the radial plots of Figure 11.
2. Investigate the effects of changing the parameters called `to`, `from`, `z0` and `transformation`. See `?radialplot` for details.

Although radial plots are most popular in fission track thermochronology, they can also be used to visualise other types of heteroscedastic data, including U-Th-He dates (Figure 12).

```
radialplot(Hedat)
```

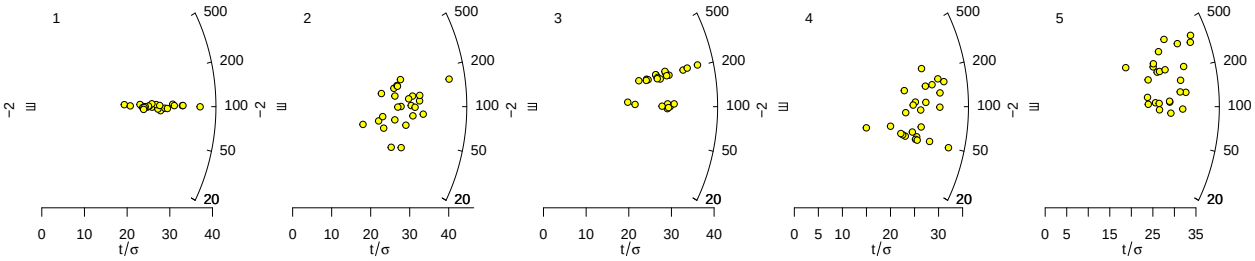

Figure 12: Radial plots for five random U-Th-He samples of the populations shown in Figure 4. These datasets are saved in files `UThHe1.csv` through `UThHe5.csv`.

Note how the 2-sigma bands of Figure 12 are much narrower than those of Figure 11. This reflects the higher precision of the U-Th-He measurements compared to the fission track data.

## 3.4 Helioplots

The U-Th-He ingrowth equation (Equation 4) is scale invariant in the sense that the age ($t$) does not depend on the absolute amounts of U, Th and He, but only on their *relative* amounts. This means that we can renormalise the U-Th-He composition of a sample to unity and plot it on a ternary diagram (Vermeesch 2008). Figure 13 plots the five samples of Figure 12 in this way.

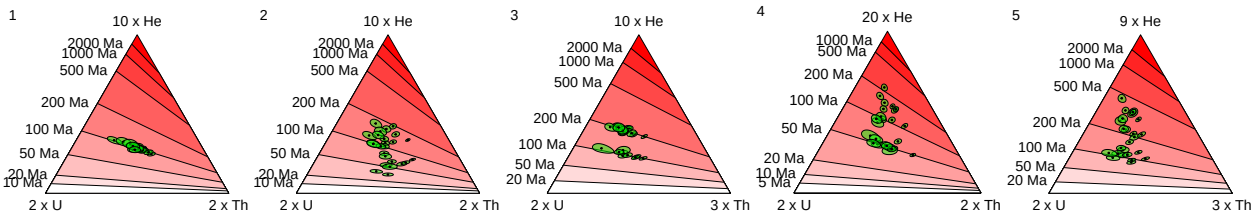

Figure 13: The five random U-Th-He samples of Figure 12 shown as ternary 'helioplots'.

Alternatively (and equivalently), the same U-Th-He data can also be visualised on bivariate logratio diagrams (Figure 14).

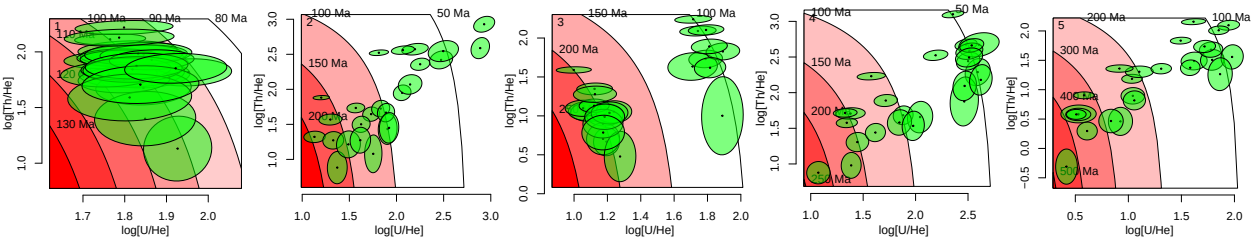

Figure 14: The same data as Figure 13 shown as bivariate logratio plots.

In `IsoplotR`:

```
helioplot(Hedat,logratio=TRUE)
```

The benefits of the logratio treatment will be discused in more detail in Section 4.6 of these notes.

# 4 Mixture models

The graphical devices of Section 3 are tools to assess the shape of the PDF of the dates (Figures 3 and 4) from a random selection of samples. However, the scientifically most important information is not contained in the PDF of the dates, but in the PDF of the ages (Figure 2). This Section will introduce statistical techniques to parameterise age populations.

## 4.1 The $\chi^2$ test for homogeneity

The $\chi^2$ test for homogeneity is a way to decide whether a dataset is derived from Figure 2's populations 1 or 2. To this end, we calculate the Chi-square test statistic. For fission track data using the External Detector Method (EDM), this statistic can be directly calculated from the raw fission track counts:

$$\chi^2_{\text{stat}} = \frac{1}{N_s N_i} \sum_{j=1}^{n} \frac{(N_{sj} N_i - N_{ij} N_s)^2}{N_{sj} + N_{ij}} \tag{9}$$

For other data, such as U-Th-He dates and ICP-MS based fission track dates, an alternative formula can be used:

$$\chi^2_{\text{stat}} = \sum_{j=1}^{n} \left( \ln[\hat{t}_j] \frac{\hat{t}_j}{s[\hat{t}_j]} \right)^2 - \frac{\left( \sum_{j=1}^{n} \ln[\hat{t}_j] \left[ \frac{s[\hat{t}_j]}{\hat{t}_j} \right]^2 \right)^2}{\sum_{j=1}^{n} \left( \frac{\hat{t}_j}{s[\hat{t}_j]} \right)^2} \tag{10}$$

If the data come from the discrete age population 1, then the value of the $\chi^2_{\text{stat}}$ is derived from a (noncentral) Chi-square distribution with $n-1$ degrees of freedom. If $n$ is reasonably large ($n > 20$, say), the expected value of $\chi^2_{\text{stat}}$ is approximately equal to the number of degrees of freedom. Put in another way, if $n > 1$ and the data are derived from population 1, then the Mean Square of the Weighted Deviates (MSWD, a.k.a. the *reduced Chi-square statistic*) is approximately 1:

$$\text{MSWD} = \frac{\chi^2_{\text{stat}}}{n-1} \approx 1 \text{ if } n > 20 \tag{11}$$

The probability of observing a Chi-square statistic of $\chi^2_{\text{stat}}$ *or greater* under a Chi-square distribution with $n-1$ degrees of freedom is called the 'p-value'. A p-value of less than 0.05 is usually taken as sufficient evidence to reject the hypothesis of a discrete age distribution such as population 1 in favour of a more dispersed age distribution such as population 2. For samples with $n > 20$, p-values below 0.05 correspond to MSWD values that exceed $1 + 2\sqrt{2/(n-1)}$.

`IsoplotR` reports the results of the Chi-square test for age homogeneity in the legend of its radial plots. Inspecting fission track samples 1 and 2 of Figure 11 and U-Th-He samples 1 and 2 of Figure 12:

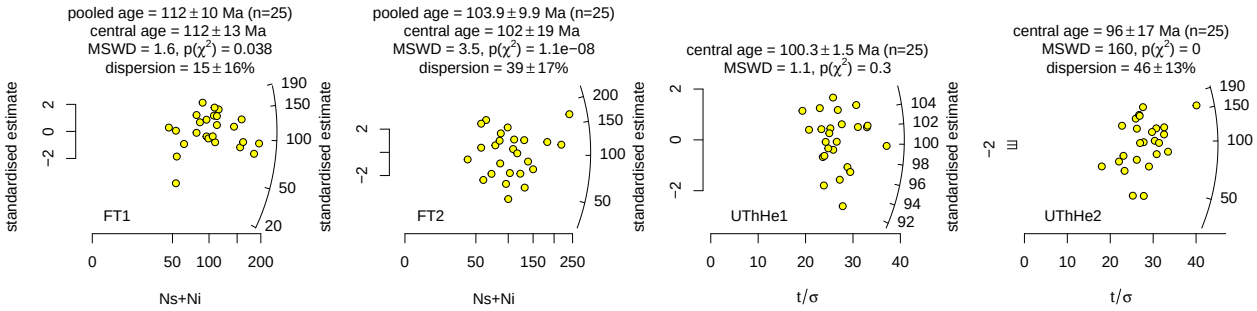


Figure 15: The same radial plots as panels 1 and 2 of Figure 11 (leftmost two panels) and Figure 12 (rightmost two panels), but with their respective legends. These legends report the MSWD and p-values, as well as the central age and (for the second and fourth panels) the dispersion parameter. The latter two quantities are discussed in Section 4.2.

Inspection of Figure 15 leads to the following observations:

1. Samples `FT2` and `UThHe2` (which are shown in the second and fourth panel) are characterised by MSWD-values that are much greater than 1, and p-values that are smaller than 0.05. Hence they fail the Chi-square test. These two samples were both drawn from population 2 of Figure 2. Thus, the test has correctly identified that these two samples are not compatible with a discrete age population.
2. Samples `FT1` and `UThHe1` do not fail the Chi-square test. This does not necessarily prove that they were drawn from a discrete age population (although, in this case, they actually were). All we can say is that we cannot rule out that they were drawn from a discrete population.
3. The MSWD value of sample `UThHe2` is much higher than that of sample `FT2`, and its p-value is much lower. So although both datasets were drawn from the same age population, the evidence for excess dispersion is much stronger for the high precision U-Th-He data than for the low precision fission track data.

## 4.2 The random effects model

Thanks to the Chi-square test, it is possible to distinguish unimodal discrete age distributions such as population 1 from more complex age distributions such as populations 2 though 5. Let us now consider population 2 in more detail. Whereas population 1 can be completely characterised by a single number (the age of the discrete peak), population 2 can be described by two parameters, controlling the position and dispersion of the data. These two parameters are commonly referred to as $\mu$ and $\sigma$, respectively (Galbraith and Laslett 1993).

For EDM-based fission track data, $\mu$ and $\sigma$ are defined as the mean and standard deviation $\ln[\rho_s/\rho_i]$.

$$\ln[\rho_s/\rho_i] \sim \mathcal{N}(\mu, \sigma^2) \tag{12}$$

For ICP-MS based fission track data and U-Th-He dates, $\mu$ and $\sigma$ are defined as the mean and standard deviation of $\ln[t]$:

$$\ln[t] \sim \mathcal{N}(\mu, \sigma^2) \tag{13}$$

Given some fission track counts or U-Th-He measurements, $\mu$ and $\sigma$ can be estimated (as $\hat{\mu}$ and $\hat{\sigma}$) using Equation 12 or 13, respectively, by treating the parameters as unknowns. This is called the 'method of maximum likelihood'. For EDM-based fission track data, the 'central age' is then defined as the age corresponding to a $\rho_s/\rho_i = \exp[\hat{\mu}]$. For other data (including U-Th-He), the central age is simply given by $\exp[\hat{\mu}]$ directly. As the name suggests, the 'dispersion' parameter $\hat{\sigma}$ quantifies the spread of the ages around the central value.

Further investigation of Figure 15 shows that only the radial plots for datasets `FT2` and `UThHe2` report the dispersion, because only these two datasets fail the Chi-square test. For datasets `FT1` and `UThHe1`, there is insufficient evidence that the dispersion is greater than zero.

Although samples `FT2` and `UThHe2` are characterised by vastly different MSWDs and p-values, their estimated dispersion parameters overlap within uncertainty. This reflects the fact that they were both derived from exactly the same age population (i.e. population 2 of Figure 2). Note, again, that the PDFs of their dates (which are shown in Figures 3 and 4) are NOT the same. In summary, the random effects model has successfully estimated key properties of the age distribution from the distribution of the dates.

In `IsoplotR`, the random effects model can either be accessed indirectly via the `radialplot()` function, or directly via the `central()` function. For fission track data:

```
FT2dat <- read.data("data/FT2.csv",method="fissiontracks")
result <- central(FT2dat)
message("MSWD=",result$mswd,", date=",result$age[1],", dispersion=",result$disp[1])
```

```
## MSWD=3.53003182876946, date=102.171999445594, dispersion=0.391601932947489
```

Similarly, for U-Th-He data:

```
He2dat <- read.data("data/UThHe2.csv",method="U-Th-He")
result <- central(He2dat)
message("MSWD=",result$mswd,", date=",result$age[1],", dispersion=",result$disp[1])
```

```
## MSWD=159.304575024221, date=95.6086385073217, dispersion=0.463091664898562
```

*Exercise*: Apply the random effects model to the fission track *dates* rather than the raw fission track data. Do this using both the `radialplot()` and `central()` functions. How different are these approximate results from the exact results obtained using the raw data?

## 4.3 Finite mixtures

The following code applies the random effects model to fission track samples `FT3` through `FT5`:

```
for (fname in c('FT3.csv','FT4.csv','FT5.csv')){
    path <- file.path('data',fname)
    result <- central(read.data(path,method="fissiontracks"))
    message(fname,": MSWD=",result$mswd,", p-value=",result$p.value)
}
```

```
## FT3.csv: MSWD=3.41689052902655, p-value=2.9062384943046e-08
```

```
## FT4.csv: MSWD=6.25887927514974, p-value=0
```

```
## FT5.csv: MSWD=7.04432662513856, p-value=0
```

All three samples fail the Chi-square test, because none of them were derived from a discrete age distribution such as population 1. However, the Chi-square test does not distinguish between populations 2, 3, 4 and 5.

The random effects model assumes that the age distribution looks like the unimodal distribution of population 2. Different statistical models are required to describe populations 3, 4 and 5. Population 3 consists of a finite mixture of two discrete age components at 100 Ma and 200 Ma. The distribution of the dates blurs out these two components (Figure 3.3) to the point where the two modes are unrecognisable in a KDE of a 25-grain sample (Figure 6.3). Once again, mathematical statistics comes to the rescue (Galbraith and Green 1990).

It is possible to capture age population 3 with three parameters: the (logs of the) $\rho_s/\rho_i$-ratios or ages of the two components ($\mu_1 = \ln[(\rho_s/\rho_i)_1]$ or $\ln[t_1]$ and $\mu_2 = \ln[(\rho_s/\rho_i)_2]$ or $\ln[t_2]$) and the fraction of the population that belongs to population 1 ($\pi$, where the fraction belonging to population 2 is given by $1 - \pi$).

In `IsoplotR`, this calculation can be done by the `radialplot()` function (Figure 16). To fit two peaks to the third dataset of Figure 11:

```
FT3dat <- read.data("data/FT3.csv",method="fissiontracks")
radialplot(FT3dat,k=2)
```

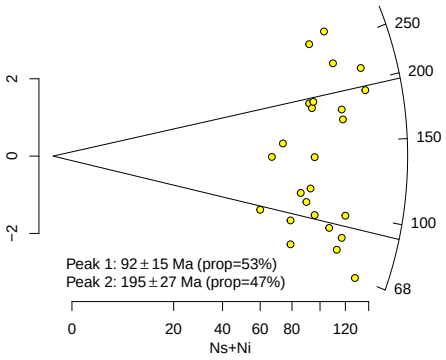


Figure 16: Finite mixture of sample `FT3` from fission track population 3, which consists of a 50/50 mixture of 100 and 200 Ma age components. Uncertainties are reported as 90% confidence intervals.

Alternatively, the numerical output of the peak fitting algorithm can also be obtained directly with the `peakfit()` function:

```
result <- peakfit(FT3dat,k=2)
cbind(t(result$peaks),t(result$props))
```

```
##           t      s[t]         p      s[p]
## 1  92.46128  7.846665 0.5287914 0.1134234
## 2 195.44752 13.670866 0.4712086 0.1134234
```

which also reports the standard errors of the estimated proportions.

## 4.4 How many components to fit?

The parameter `k` of the `radialplot()` and `peakfit()` functions controls the number of components in the finite mixture model. However, in many cases, the number of components in a natural sample is unkown. `IsoplotR` provides a mechanism to choose the 'optimal' number of finite components that can be fitted to a detrital population:

```
result <- peakfit(FT3dat,k='auto')
cbind(t(result$peaks),t(result$props))
```

```
##           t      s[t]         p      s[p]
## 1  92.46128  7.846665 0.5287914 0.1134234
## 2 195.44752 13.670866 0.4712086 0.1134234
```

which produces exactly the same result as `k=2`. It is tempting to use `k="auto"` as the default option. However, doing so can produce misleading results. Suppose that the detrital age distribution (which is unknown) consists of a continuous mixture such as population 2 of Figure 2. Fitting a finite mixture to the continuous mixture of sample `FT2`:

```
FT2dat <- read.data("data/FT2.csv",method="fissiontracks")
result <- peakfit(FT2dat,k='auto')
result$peaks
```

```
##              1          2
## t    60.068034 134.417978
## s[t]  7.213805   9.075505
```

The `peakfit()` algorithm has returned two age components, even though the population has only one (continuous) age component. The two component ages are statistically significant but geologically meaningless. To make matters worse, the number of statistically justifiable discrete age components increases with sample size. To illustrate this phenomenon, let us use `helper.R`'s `FTsamp()` function to generate three samples containing 20, 100 and 500 values, respectively, from population 2 (Figure 17).

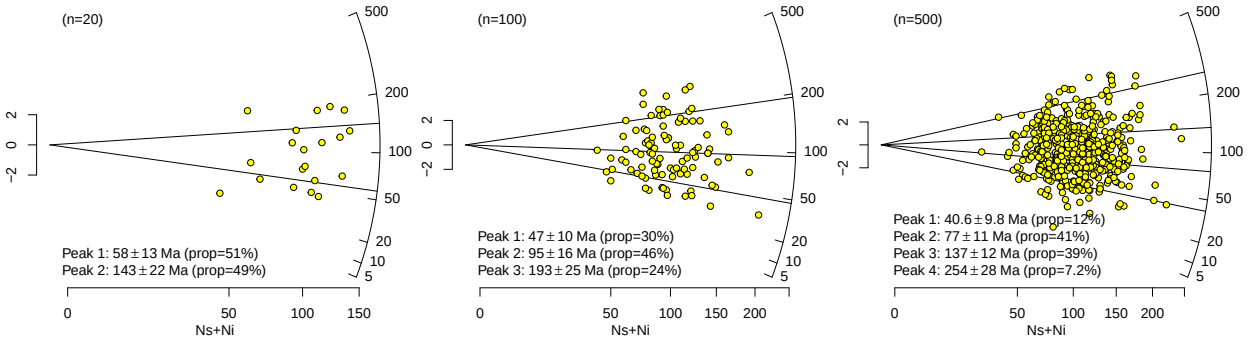


Figure 17: Applying the finite mixture model to continuous mixtures from population 2 yields statistically valid but geologically meaningless results. The number of 'automatically' fitted components increases with sample size ($n$).

In summary, the `auto` setting is of limited value. It should not be used to 'prove' the presence of finite mixtures. The greatest value of the `auto` setting is to protect oneself against over-interpreting data: if the automated peak fitting returns two age components, then there is definitely *not* enough evidence to justify the presence of three age components!

## 4.5 The minimum age model

The '*lag time*' is the difference between the youngest age component and the depositional age of a sedimentary deposit. Lag times are often used to track tectonic exhumation rates over time. It should be clear that the finite mixture models of Section 4.3 are not ideally suited for lag time estimation. So let us abandon this model in favour of a model that describes the functional form of population 4 (Figure 2) with four parameters:

1. $\gamma = \ln[(\rho_s/\rho_i)_m]$ (for EDM data) or $= \ln[t_m]$ (for other data), where $(\rho_s/\rho_i)_m$ is the $\rho_s/\rho_i$-ratio and and $t_m$ the age of the (discrete) youngest component;
2. $\mu = \ln[(\rho_s/\rho_i)_c]$ (for EDM data) or $= \ln[t_c]$ (for other data), are the means of truncated (at $\gamma$) normal distributions describing the continuous distribution of a hypothesised older age population;
3. $\sigma$ is the standard deviation of the continuous mixture of older ages (before truncation);
4. $\pi$ represents the fraction of the population belonging to the discrete youngest age component, so that $1 - \pi$ is the fraction of the population belonging to the older component.

These four parameters can be estimated using the method of maximum likelihood, i.e. by finding the parameters that minimise the misfit with the data. To increase numerical stability, it is useful to reduce the number of free parameters from four to three by making the simplifying assumption that $\gamma = \mu$. This usually has an only minor effect on the accuracy of the minimum age estimate (Figure 18).

In `IsoplotR`, this minimum age model can be used by setting `k='min'` in `radialplot()` and `peakfit()`:

```
FT4dat <- read.data("data/FT4.csv",method="fissiontracks")
radialplot(FT4dat,k='min')
```

which uses the 3-parameter version of the minimum age module. To use the 4-parameter version:

```
radialplot(FT4dat,k='min',np=4)
```

Figure 18 reanalyses the synthetic datasets of Figure 17 using the minimum age model.

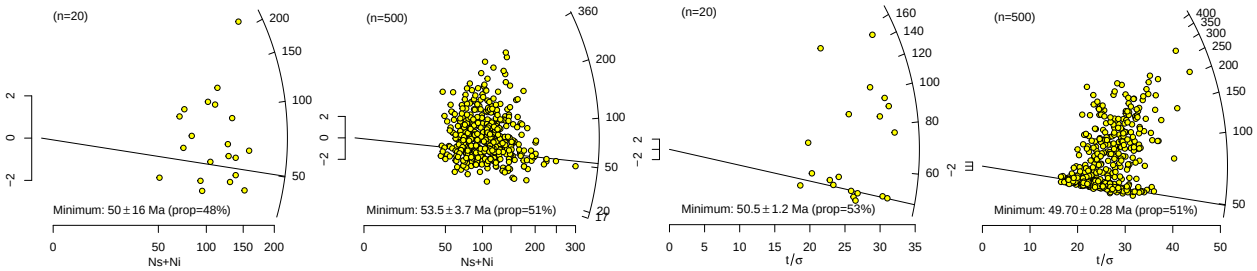


Figure 18: Applying the minimum age model to three random samples from population 3. This produces more stable results, which do not systematically depend on sample size.

## 4.6 Logratios

As discussed in Section 3.4, the U-Th-He ingrowth equation is scale invariant, allowing U-Th-He compositions to be visualised on ternary diagrams and logratio plots (Figures 13 and 14). This section will show that the logratio diagram is the 'natural' space within which to process U-Th-He (and other geochronological) datasets. To illustrate the benefits of using logratios, consider the following synthetic dataset of U-He measurements (assuming Th=0):

```
U <- signif(runif(10),4)
He <- signif(runif(10),4)
rbind('U'=U,'He'=He)
```

```
##      [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  [,10]
## U  0.9042 0.3353 0.2882 0.7574 0.9392 0.4527 0.8208 0.8665 0.4707 0.4032
## He 0.4270 0.1374 0.6048 0.2298 0.3523 0.4440 0.7648 0.7290 0.7319 0.5436
```

where the `signif()` function rounds the random numbers to four significant digits for the sake of brevity. Let us now calculate the U/He ratios and the He/U ratios:

```
ratios <- rbind('U/He'=U/He,'He/U'=He/U)
print(signif(ratios,4))
```

```
##        [,1]   [,2]   [,3]   [,4]   [,5]   [,6]   [,7]   [,8]   [,9]  [,10]
## U/He 2.1180 2.4400 0.4765 3.2960 2.6660 1.0200 1.0730 1.1890 0.6431 0.7417
## He/U 0.4722 0.4098 2.0990 0.3034 0.3751 0.9808 0.9318 0.8413 1.5550 1.3480
```

It is easy to see that $1/(\mathrm{U/He}) = \mathrm{He/U}$ (for example, $1/2.118 = 0.4722$. However, when we compare the arithmetic means of the U/He and He/U vectors:

```
signif(rowMeans(ratios),4)
```

```
##   U/He   He/U
## 1.5660 0.9316
```

Then we find that $1/1.566 = 0.6385 \neq 0.9316$!

The solution to this puzzling inconsistency is simple: take logarithms. Calculate the U/He and He/U logratios:

```
signif(log(ratios),3)
```

```
##        [,1]   [,2]   [,3]  [,4]   [,5]     [,6]     [,7]   [,8]   [,9]  [,10]
## U/He   0.75  0.892 -0.741  1.19  0.981  0.0194  0.0707  0.173 -0.441 -0.299
## He/U -0.75 -0.892  0.741 -1.19 -0.981 -0.0194 -0.0707 -0.173  0.441  0.299
```

Averaging the logratios and exponentiating produces two geometric means:

```
signif(exp(rowMeans(log(ratios))),4)
```

```
##   U/He   He/U
## 1.2970 0.7713
```

then we see that $1/1.297 = 0.7713$, an altogether more satisfying result.

The logratio approach can be generalised from two to three variables. Thus, the three components of the U-Th-He method can be mapped to two logratios and back:

$$\mathrm{U} = \frac{e^x}{e^x + e^y + 1}, \mathrm{Th} = \frac{e^y}{e^x + e^y + 1}, \mathrm{He} = \frac{1}{e^x + e^y + 1} \rightleftarrows x = \ln\left[\frac{\mathrm{U}}{\mathrm{He}}\right], y = \ln\left[\frac{\mathrm{Th}}{\mathrm{He}}\right] \tag{14}$$

Taking the (weighted) mean of the logratios and subjecting the resulting value to an inverse logratio transformation gives rise to a compositional mean. Plugging this compositional mean into Equation 4 and solving it for time gives rise to a 'barycentric' age. Note that, in previous versions of `IsoplotR`, this value was called a 'central' age (Vermeesch 2008). However, it was recently renamed to avoid confusion with the central age of Section 4.2.

In `IsoplotR`, this calculation is done by the `helioplot()` and `central()` functions.

```
result <- central(Hedat,compositional=TRUE)
message("barycentric age= ",signif(result$age["t"],4),
        ", MSWD = ",signif(result$mswd,4),
        ", p.value = ", signif(result$p.value,4))
```

```
## barycentric age= 11.27, MSWD = 5.379, p.value = 5.429e-14
```

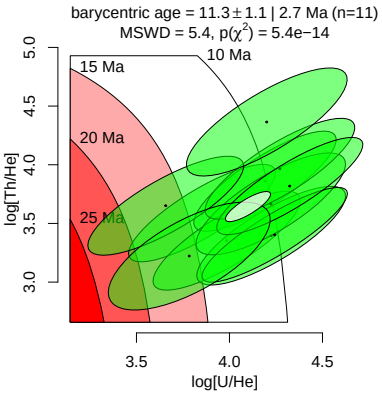


Figure 19: Logratio-plot of some U-Th-He data with indication of the weighted mean composition (white ellipse) and barycentric U-Th-He age.

The 'compositional' MSWDs of the helioplot tend to be significantly higher (and the p-values lower) than the MSWDs (and p-values) of the central ages that are shown on radial plots.

# 5 Mineral fertility estimation with `provenance`

The previous sections reviewed techniques that can be applied to individual samples of bedrock or sediment. Most studies, however, usually involve multiple samples. This is especially the case for detrital thermochronology studies that aim to map out sediment budget in river catchments. The `provenance` package includes some useful functions for such studies. As the name suggests, `provenance` is an `R` package for statistical provenance analysis (Vermeesch, Resentini, and Garzanti 2016). It can be installed as follows:

```
install.packages("provenance")
```

and used in two different ways:

1. Using a query based user interface:

```
provenance::provenance()
```

2. Using the command-line API.

```
library(provenance)
```

## 5.1 Grain size estimation

One of the principal aims of detrital thermochronology is to determine catchment-wide exhumation rates. These exhumation rates are often used to estimate mass balances between different catchments. The accuracy of these mass balances crucially depends on the apatite or zircon concentrations of the river catchments of interest. Consider, for example, the confluence of two rivers: one draining a catchment dominated by carbonate lithologies and another draining a zircon-rich granite. The zircon age spectrum of the pooled sediment will be completely dominated by the granite catchment, whilst being blind to the exhumation that is taking place in the carbonate catchment. In order to get a handle on mineral fertility, it is extremely useful to measure the concentration of apatite or zircon in the samples (Malusà, Resentini, and Garzanti 2016).

The first steps of any thermochronological study are sampling and mineral separation. Both steps may involve sieving samples. It is important to choose an appropriate grain size window, to avoid the risk of (partially) missing the apatite or zircon fraction. This risk exists because the mineralogical composition of siliciclastic sediment strongly depends on grain size: small grains of dense zircon ($\rho = 4.6 - 4.7$ g/cm$^3$) are hydraulically equivalent to larger grains of less dense quartz ($\rho = 2.65$ g/cm$^3$). The 'size shift' between minerals of different density can be predicted using the Minsorting algorithm, which is implemented in the `provenance` package.

The following code snippet predicts the grain size distributions of the quartz, apatite and zircon in a well sorted sand with an average Krumbein grain size of $\phi = 2 \pm 0.5(1\sigma)$. `read.compositional()` and `minsorting()` are functions of the `provenance` package and `densities` is a table with mineral densities that is included with it. Type `?read.compositional`, `?minsorting` and `?densities` at the console for futher details.

```
comp <- read.compositional("data/mineralogy.csv")
grainsizes <- minsorting(comp,dens=densities,sname="N1",
                         phi=2,sigmaphi=0.5,medium="freshwater",by=0.05)
plot(grainsizes,components=c("Q","ap","zr"))
```

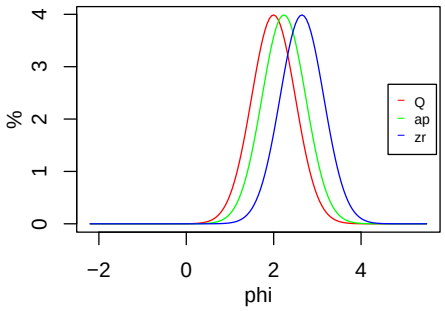


Figure 20: Predicted grain size distributions for quartz, apatite and zircon. The mean grain size for the bulk sample is $\phi = 2$ with a standard deviation of $s[\phi] = 0.5$. However, dense apatite and zircon are significantly finer grained than the light quartz, resulting in a size shift of 0.65 $\phi$-units between quartz and zircon, and a size shift of 0.25 $\phi$-units between quartz and apatite.

## 5.2 Source Rock Density (SRD) correction

After mineral separation, the zircon or apatite fraction can be weighed and divided by the total mass of the sample. This provides useful estimates of the zircon and apatite fertilities of the sample catchment. It is also useful to weigh the bulk density of the sample to detect the effects of *hydraulic sorting* (Garzanti, Andò, and Vezzoli 2008; Garzanti, Andò, and Vezzoli 2009).

With the exception of some exotic geological settings such as ophiolites, the bulk density of most silicilastic source materials is remarkably constant, between 2.65 and 2.75 g/cm$^3$. Placer deposits, on the other hand, can easily reach bulk densities of 4 g/cm$^3$ or more.

Hydraulic sorting of minerals changes their relative abundances in predictable ways. Heavy mineral enrichment in placers and heavy mineral depletion in anti-placers can significantly bias mineral fertility estimates. `provenance` implements methods to undo these biases by restoring the compositon of a sedimentary assemblage to some hypothesised initial bulk density.

```
rescomp <- restore(comp,dens=densities,target=2.71)
message("original zr fraction: ",signif(comp$x['N8','zr'],4),
        "%, restored zr fraction: ",signif(rescomp$x['N8','zr'],4),"%")
```

```
## original zr fraction: 0.3166%, restored zr fraction: 0.04076%
```

# Bibliography


Galbraith, R. F. 1988. "Graphical Display of Estimates Having Differing Standard Errors." *Technometrics* 30 (3): 271–81.

———. 1990. "The Radial Plot: Graphical Assessment of Spread in Ages." *Nuclear Tracks and Radiation Measurements* 17: 207–14.

Galbraith, R. F., and P. F. Green. 1990. "Estimating the Component Ages in a Finite Mixture." *Nuclear Tracks and Radiation Measurements* 17: 197–206.

Galbraith, R. F., and G. M. Laslett. 1993. "Statistical Models for Mixed Fission Track Ages." *Nuclear Tracks and Radiation Measurements* 21 (4): 459–70.

Garzanti, E., S. Andò, and G. Vezzoli. 2008. "Settling Equivalence of Detrital Minerals and Grain-Size Dependence of Sediment Composition." *Earth and Planetary Science Letters* 273 (1): 138–51.

Garzanti, E., S. Andò, and G. Vezzoli. 2009. "Grain-size dependence of sediment composition and environmental bias in provenance studies." *Earth and Planetary Science Letters* 277: 422–32. https://doi.org/10.1016/j.epsl.2008.11.007.

Malusà, M. G., A. Resentini, and E. Garzanti. 2016. "Hydraulic Sorting and Mineral Fertility Bias in Detrital Geochronology." *Gondwana Research* 31: 1–19.

Vermeesch, P. 2007. "Quantitative geomorphology of the White Mountains (California) using detrital apatite fission track thermochronology." *Journal of Geophysical Research (Earth Surface)* 112 (F11): 3004. https://doi.org/10.1029/2006JF000671.

———. 2008. "Three New Ways to Calculate Average (U-Th)/He Ages." *Chemical Geology* 249: 339–47.

———. 2012. "On the Visualisation of Detrital Age Distributions." *Chemical Geology* 312-313: 190–94. https://doi.org/10.1016/j.chemgeo.2012.04.021.

———. 2018. "`IsoplotR`: a free and open toolbox for geochronology." *Geoscience Frontiers* 9: 1479–93.

Vermeesch, P., A. Resentini, and E. Garzanti. 2016. "An R package for statistical provenance analysis." *Sedimentary Geology*.