



Supplement of

Luminescence age calculation through Bayesian convolution of equivalent dose and dose-rate distributions: the D_e - D_r model

Norbert Mercier et al.

Correspondence to: Norbert Mercier (norbert.mercier@u-bordeaux-montaigne.fr)

The copyright of individual parts of the supplement might differ from the article licence.

Supplement S1: Luminescence age calculation through
Bayesian convolution of equivalent dose and dose-rate
distributions: the De_Dr model

Norbert Mercier, Jean-Michel Galharret, Chantal Tribolo, Sebastian Kreutzer, Anne Philippe

Version: 2022-03-07

S1 Scope

This document provides a brief example of how to model OSL ages by combining dose rate and equivalent dose distributions with the R package 'Luminescence' (Kretzler et al., 2012). The functions underpinning functionality were contributed by Anne Philippe, Jean-Michel Galharret, Norbert Mercier, and Chantal Tribolo. The integration into the R package 'Luminescence' was carried out by Sebastian Kretzler. To run this example, you need the package 'Luminescence' version $\geq 0.9.16$.

S2 Preparation

S2.1 Load the package 'Luminescence'

First we start R and load the R package 'Luminescence' into our R session.

```
library(Luminescence)
```

S2.2 Creating example data set

Now we create an artificial dataset. To ensure that the creation of this dataset is reproducible, we set a random seed.

Please note: In our example, the sizes of the equivalent-dose dataset and the dose rate dataset are identical. However, the modelling can also handle datasets of unequal length.

```
## set random seed to keep that example reproducible
set.seed(8765)
```

```
## create dose rate dataset in Gy/ka
Dr <- stats::rlnorm(1000, 1, 0.3)
```

```
## create equivalent dose dataset
## and add uncertainties in Gy
## Here the results of 50 times (50 ka) the
## dose rate
De <- 50 * sample(Dr, 50, replace = TRUE)
s <- stats::rnorm(50, 7, 2) * De/100
```

Now that we have created the dataset, we can visualise the numbers using base R or a function from the R package 'Luminescence':

```
par(mfrow = c(1,2))
plot(density(Dr), xlab = "Dr [Gy/ka]")
plot_KDE(data = data.frame(De, s))
```

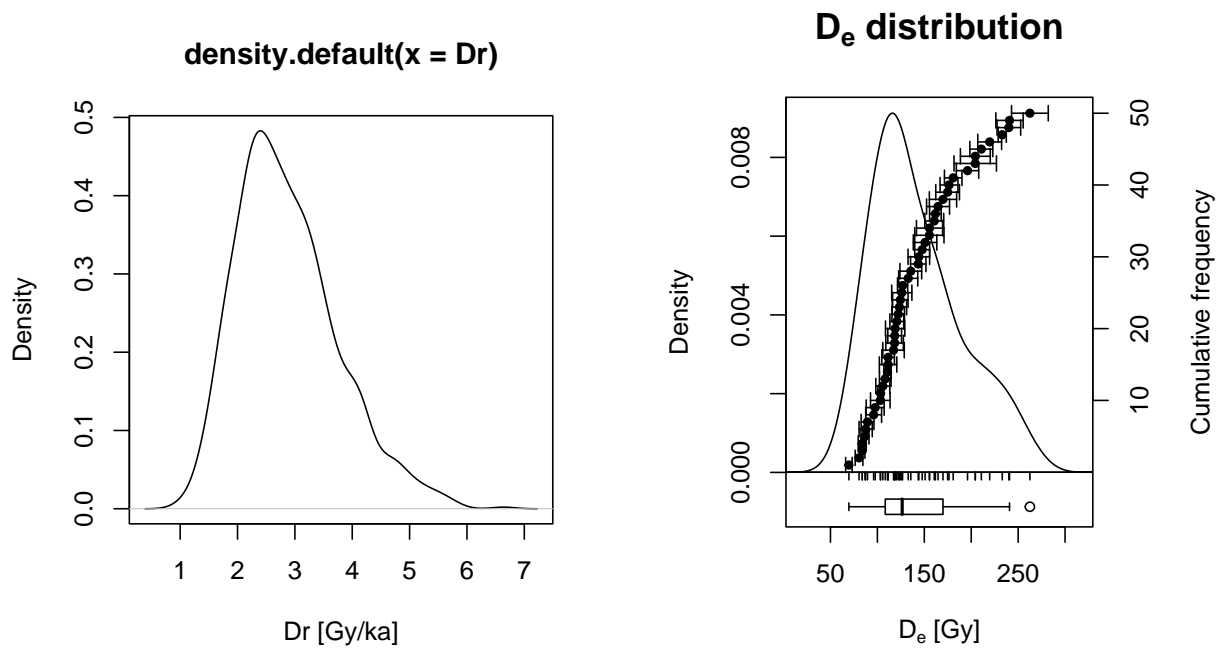


Fig. S1: Kernel density plots of the D_r (left) and the D_e (right) distribution.

S3 Age modelling

With that dataset being created, we can quickly run the model using `thecombine_De_Dr()` function. The parameters passed with the argument `method_control` allow us to fine-tune the MCMC process. Here we have selected 50000 iterations and three MCMC chains. We also opted for an output plot showing the detected outliers.

```
results <- combine_De_Dr(  
  Dr = Dr,  
  int_OD = 0.1,  
  De = De,  
  s = s,  
  cdf_ADr_quantiles = TRUE,  
  outlier_analysis_plot = TRUE,  
  Age_range = c(0,100),  
  method_control = list(  
    n.iter = 50000,  
    n.chains = 3))
```

```
##  
## [combine_De_Dr()]  
## (1) Running Bayesian modelling 'Individual Age Model' ... DONE  
## >> Outliers detected: 12/50 (24%)  
## (2) Running Bayesian modelling 'Bayesian Central Age Model' ... DONE  
## (3) Age results (presumably in ka)  
## -----  
## Age (HPD) : 47.90  
## Age (CI 68%): 45.47 : 50.51  
## Age (CI 95%): 43.18 : 53.18  
## -----
```

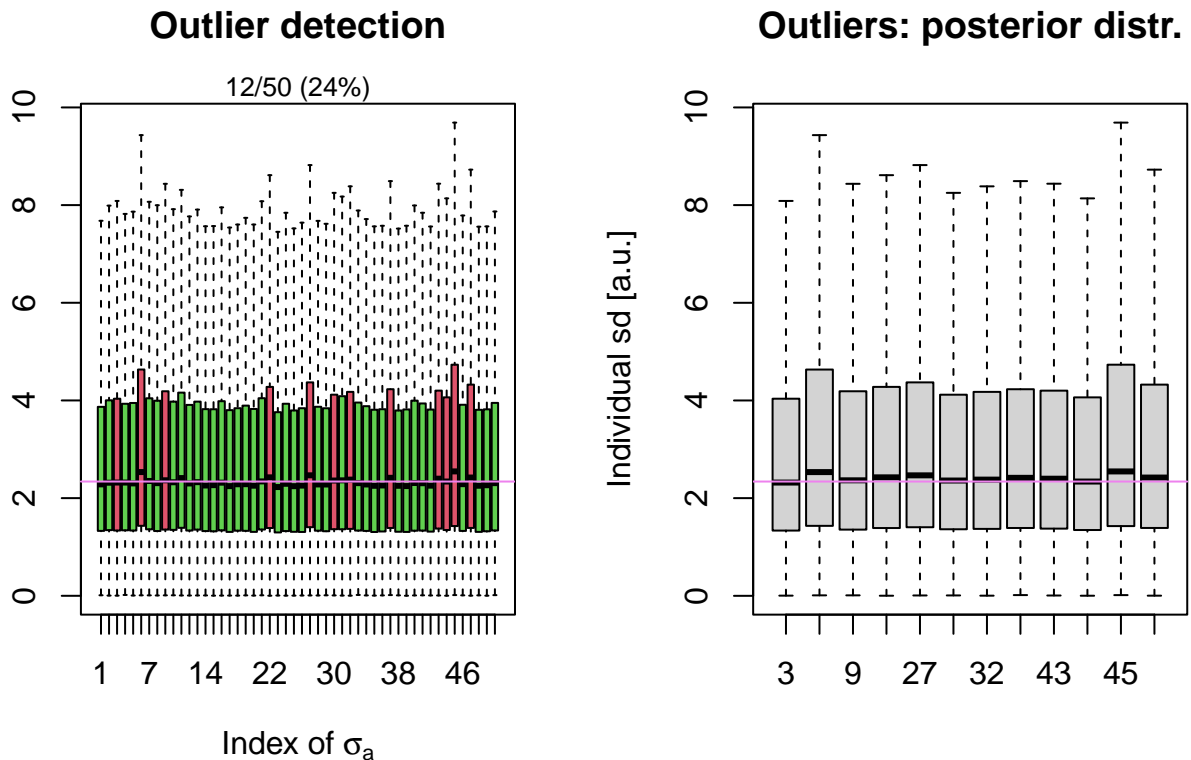


Fig. S2: Typical plot output of `combine_De_Dr()`.

What do the plots show? **The first two plots** refer to the outlier detection, in case the argument `outlier_analysis_plot = TRUE`. Displayed are so-called boxplots illustrating the variation of σ_a , which

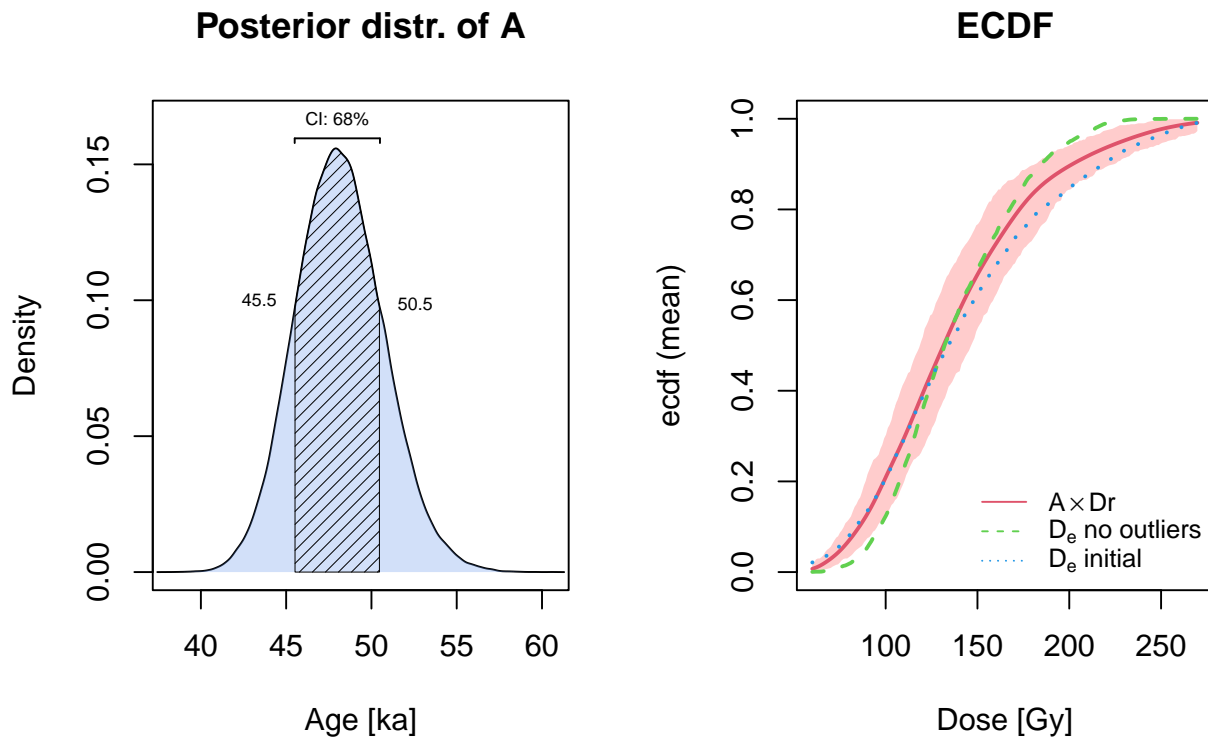


Fig. S3: Typical plot output of `combine_De_Dr()`.

is the individual standard deviation of the posterior age distributions. We have 50 D_e values. Hence we have 50 age posterior age distributions. If an outlier is detected (the detection depends on the selected algorithm), the corresponding boxplot is coloured in red. The standard deviation distribution of the outlier values is shown once more in the close-up view (the right-hand plot).

The following plots show a kernel density plot of the posterior ages (left) with the selected credible interval (here 68%) and a plot of the empirical cumulative distribution function (ecdf). The latter one allows assessing the quality of the modelling by comparing the cumulative D_e distribution (the input data with or without the identified outliers), with the modelled D_e distribution ($A \times Dr$). Ideally, the latter two are very similar.

S3.1 Show me more

The modelling output contains an `RLum.Results` object, which can be inspected further.

```
results

##
## [RLum.Results-class]
##   originator: combine_De_Dr()
##   data: 9
##     .. $Ages : numeric
##     .. $Ages_stats : data.frame
##     .. $outliers_index : integer
##     .. $cdf_ADr_mean : numeric
##     .. $cdf_ADr_quantiles : matrix
##     .. $cdf_De_no_outlier : numeric
##     .. $cdf_De_initial : numeric
##     .. $mcmc_IAM : NULL
##     .. $mcmc_BCAM : NULL
##   additional info elements: 3
```

The object contains the sample ages, which form the output distribution, the index of the detected outliers and the empirical cumulative distributions for the D_e without outliers and from the initial D_e . Example:

```

head(results$Ages)

## [1] 48.40817 48.28127 49.81658 50.77291 47.49147 46.28086

results$outliers_index

## [1] 3 6 9 22 27 30 32 37 43 44 45 47

head(results$cdf_ADr_mean)

## [1] 0.0072258 0.0084614 0.0099402 0.0116680 0.0136490 0.0158632

head(results$cdf_De_no_outlier)

## [1] 4e-04 6e-04 8e-04 8e-04 8e-04 1e-03

head(results$cdf_De_initial)

## [1] 0.0212 0.0232 0.0250 0.0266 0.0286 0.0308

```

These values might be used, for instance, to recreate Fig. 5 in the main text of the manuscript manually.

Furthermore, on purpose, we do not return all modelling values to keep the model function concise and usable without overwhelming the user with too much information. However, if you want to inspect the applied models and even build your own R script based on the models, you can run:

```

## show models
writeLines(results@info$model_IAM)

## model{
##     for( i in 1 : N ) {
##         D_e[i] ~ dnorm(a[i] * mu[z[i]], tau[i])
##         tau[i] <- 1 / (a[i] * sigma[z[i]]) ^ 2
##         z[i] ~ dcat(theta)
##         De[i] ~ dnorm(D_e[i], prec2[i])
##         a[i] ~ dnorm(A, prec_a[i])
##         u[i] ~ dunif(0, 1)
##         prec_a[i] <- 1 / s02 * u[i] / (1 - u[i])
##         prec2[i] <- 1 / (s2[i])
##         sig_a[i] <- 1 / sqrt(prec_a[i])
##     }
##     A ~ dunif(Amin, Amax)
## }

writeLines(results@info$model_BCAM)

##
## model{
##     for( i in 1:J) {
##         D_e[i] ~ dnorm(A*mu[z[i]],tau[i])
##         tau[i]<-1/(A*sigma[z[i]])^2
##         z[i] ~ dcat(theta)
##         De[i] ~ dnorm(D_e[i], prec2[i])
##         prec2[i] <- 1/(s2[i])
##     }
##
##     A ~ dunif(Amin,Amax)
## }

```

S3.2 I want to check the Bayesian process

The function is designed to provide an optimal amount of information, however, sometimes one may want to gain more insight into the Bayesian process. This can be achieved by activating `method_control`

= list(return_mcmc = TRUE), which avoids that the information from the Bayesian modelling are discarded (to save memory).

In the following example we return the MCMC information for a reduced dataset (here only the first 8 D_e for illustrative reasons). No plot and terminal output is shown.

```
results_BayesianInformation <- combine_De_Dr(  
  Dr = Dr,  
  outlier_threshold = 1,  
  int_OD = 0.1,  
  De = De[1:8],  
  s = s[1:8],  
  verbose = FALSE,  
  plot = FALSE,  
  outlier_analysis_plot = FALSE,  
  Age_range = c(0,100),  
  method_control = list(  
    return_mcmc = TRUE,  
    n.iter = 5000,  
    n.chains = 2))
```

To visualise the Bayesian modelling information, we can use the functionality of the package 'coda' (Plummer et al., 2006), for example, to create a trace plot.

```
par(mfrow = c(3,3))  
coda::traceplot(  
  x = results_BayesianInformation$mcmc_BCAM,  
  smooth = TRUE)
```

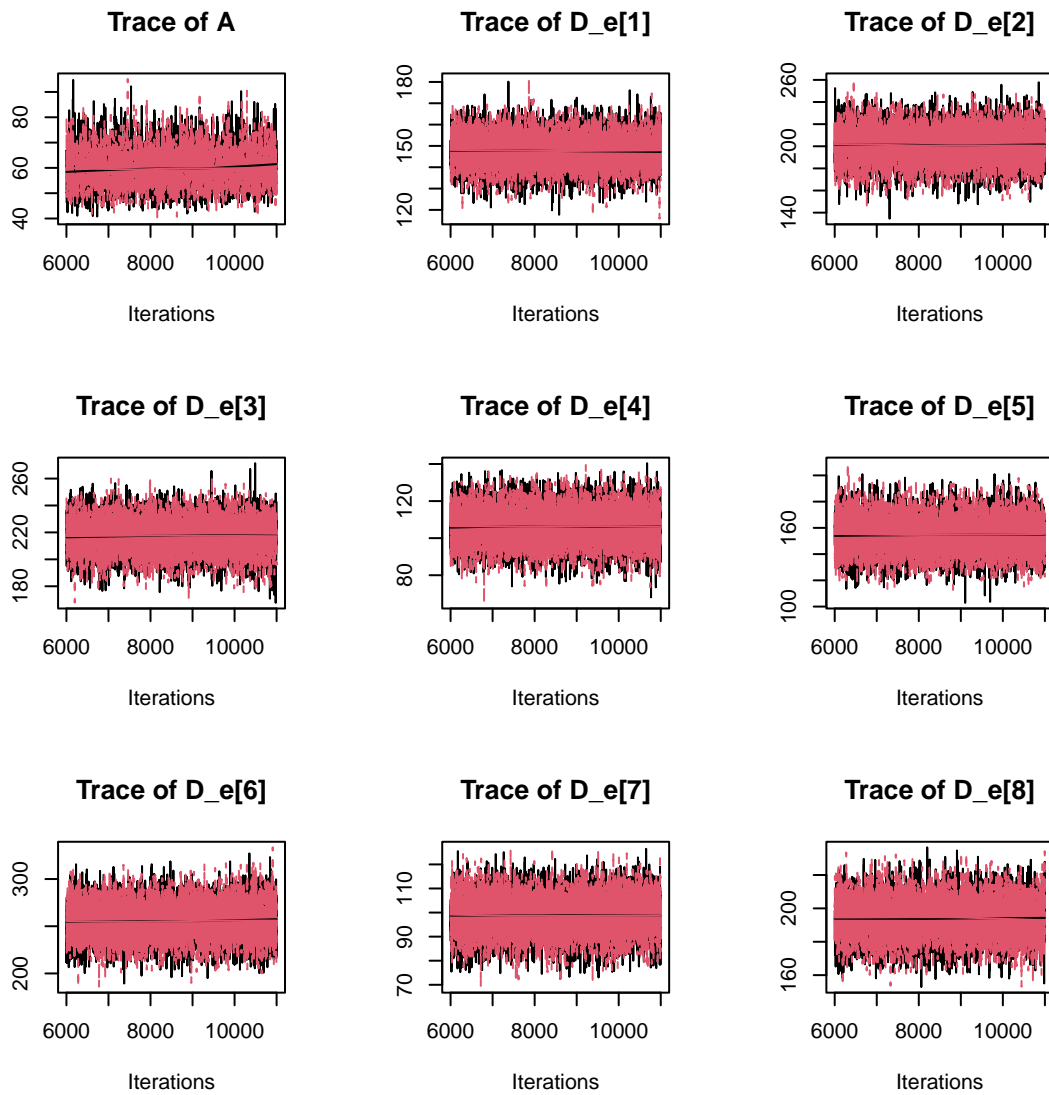



Fig. S4: MCMC diagnostic trace plots as returned by `coda :: traceplot()`.

S3.3 I am not happy with the output plot

Suppose you are unhappy with the plot output. In that case, the OSL age summary can be shaped in various ways using the function `plot_OSLAgeSummary()` and, of course, standard R plot functionality. In the brief example below, we will add the age values as points in ascending order to the plot. The exciting part is that the plot itself also returns a few statistical parameters, such as the central value and the credible interval.

```
plot_OSLAgeSummary(  
  object = results,  
  level = 0.99,  
  main = "Ages distribution",  
  polygon_col = rgb(0, 1, 0, 0.4),  
  polygon_density = 10)  
  
##  
## [plot_OSLAgeSummary()]  
## Credible Interval (99 %): 41.9 : 54.9  
## Bayes estimate (posterior mean ± sd): 48.2 ± 2.6  
  
##  
## [RLum.Results-class]  
## originator: plot_OSLAgeSummary()  
## data: 3  
## .. $Estimate : numeric  
## .. $Credible_Interval : matrix  
## .. $level : numeric  
## additional info elements: 1
```

```
points(  
  sort(results$Ages),  
  seq(0,0.15, length.out = length(results$Ages)),  
  pch = 20,  
  col = rgb(0,0,0,0.1))
```

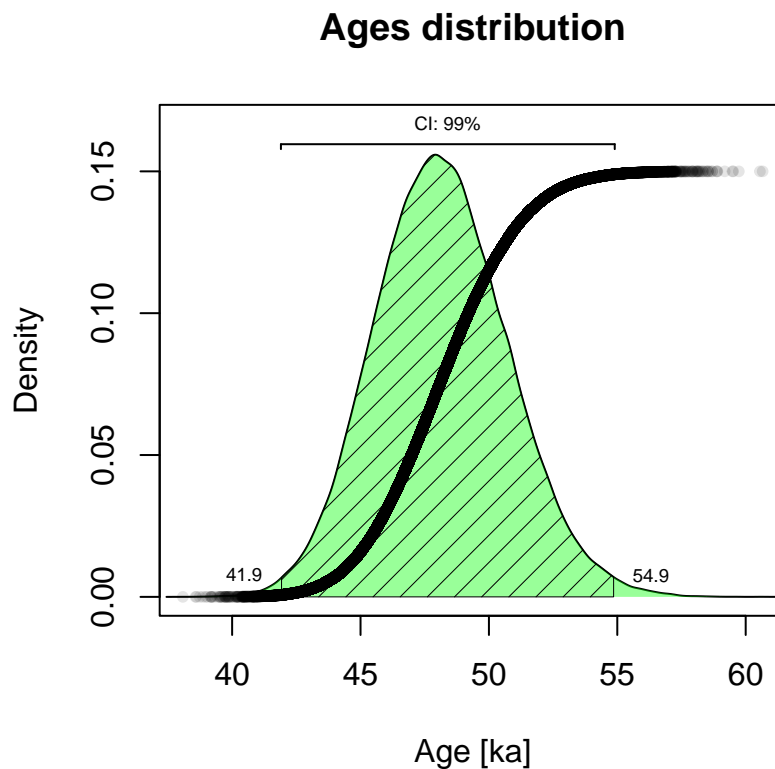


Fig. S5: Output `plot_OSLAgeSummary()`.

S4 Incooperating systematic errors

In all samples above, individual errors were considered random. However, one may want to add systematic error to the final age. This can be done relatively quickly using standard R functionality. For more details, see the main text.

```
p <- 0.1 # case of a relative total systematic error of 10%
res <- results@data$Ages
res <- res * rnorm(length(res), 1, p)
plot_OSLAgeSummary(res, level = 0.68)
```

```
##
## [plot_OSLAgeSummary()]
## Credible Interval (68 %): 42.6 : 53.2
## Bayes estimate (posterior mean ± sd): 48.2 ± 5.5
```

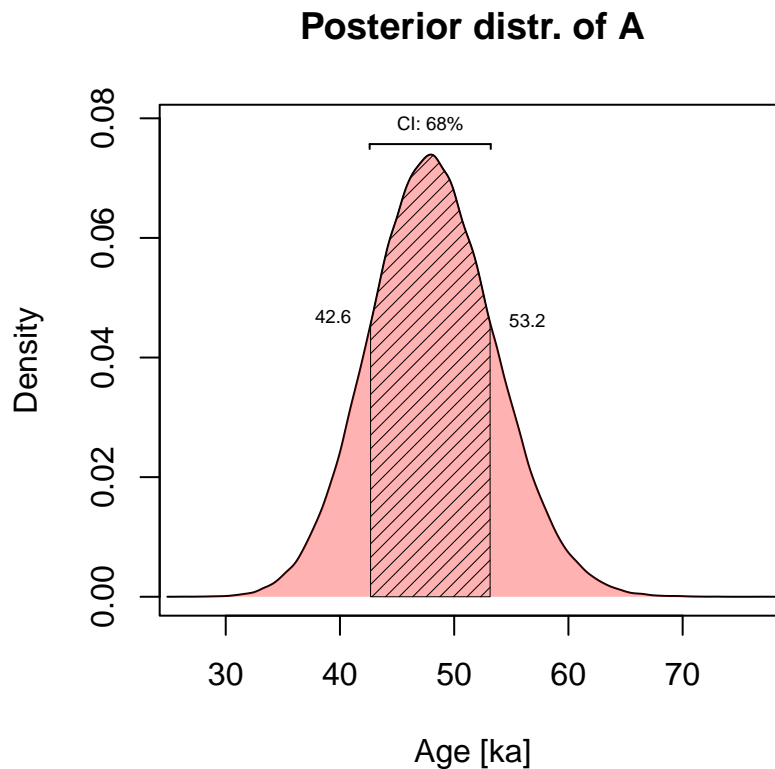


Fig. S6: Adding systematic error to the final age.

```
##
## [RLum.Results-class]
## originator: plot_OSLAgeSummary()
## data: 3
## .. $Estimate : numeric
## .. $Credible_Interval : matrix
## .. $level : numeric
## additional info elements: 1
```

References

- Kreutzer, S., Schmidt, C., Fuchs, M. C., Dietze, M., Fischer, M., and Fuchs, M.: Introducing an R package for luminescence dating analysis, *Ancient TL*, 30, 1–8, 2012.
- Plummer, M., Best, N., Cowles, K., and Vines, K.: CODA: Convergence diagnosis and output analysis for MCMC, *R News*, 6, 7–11, 2006.

Supplement S2: Code and results maintext Tables 1 to 3

Norbert Mercier, Jean-Michel Galharret, Chantal Tribolo, Sebastian Kreutzer, Anne Philippe

Version: 2022-03-07

S1 Scope

This document records code and results used to produce the results reported in Tables 1-3. Please note, because of the stochastic nature of the process the numbers will always vary a little bit if repeated.

S2 General settings

```
## load package
library(Luminescence)

## set variables according to what is written
## in the manuscript Mercier et al., preprint

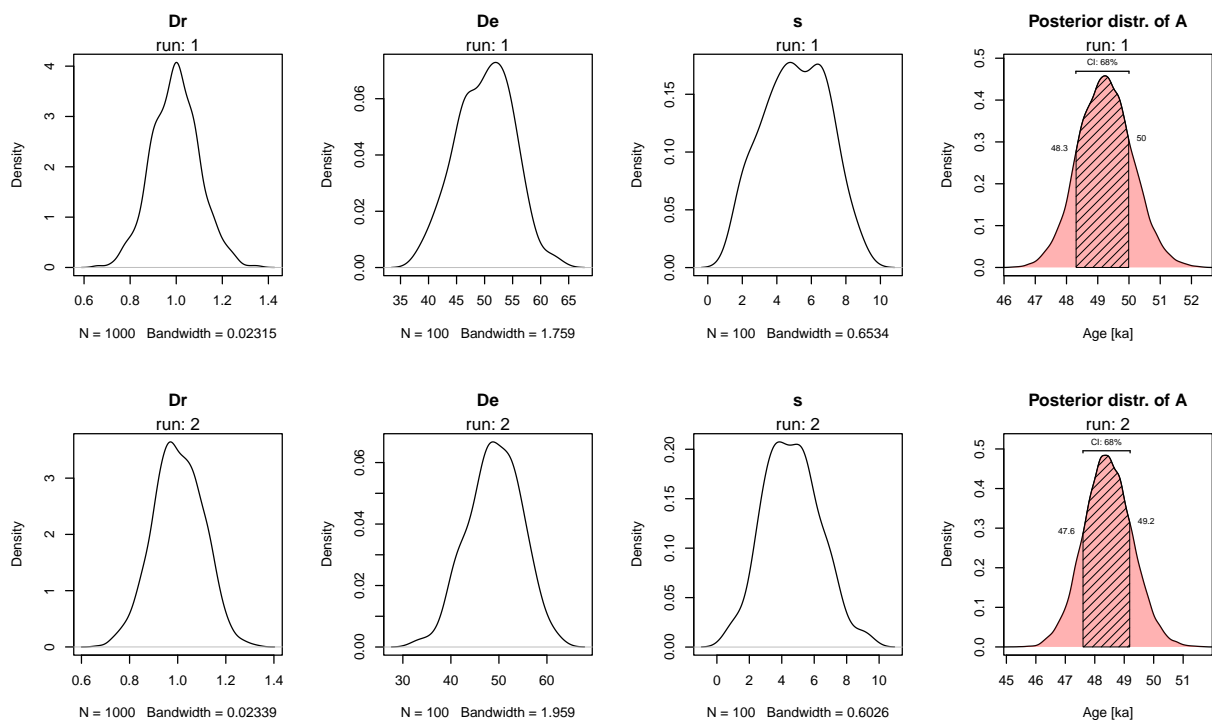
##general settings
nb_aliquots <- 100
mean_s <- 10
sigma_s <- mean_s/3
supposed_age <- 50
int_OD <- 0.1

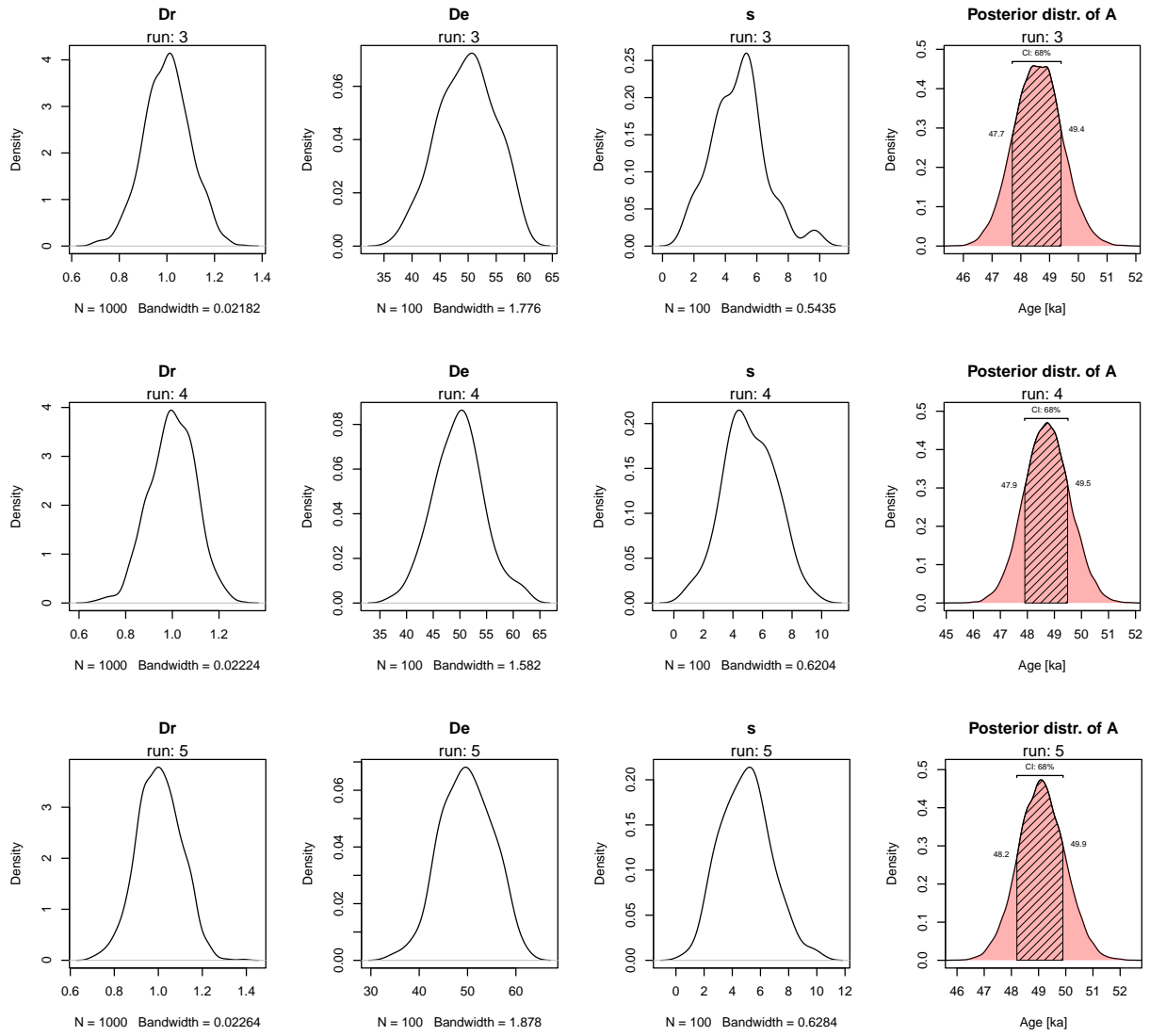
## set Bayesian modelling parameters
n.iter <- 5000
age_range <- c(0,100)
```

S3 Table 1

S3.1 Norm(1000, 1, 0.1)

```
results1 <- lapply(1:5, function(i){  
  # set parameters  
  Dr <- stats::rnorm(1000, 1, 0.1)  
  De <- supposed_age * sample(Dr, nb_aliquots, replace = TRUE)  
  s <- stats::rnorm(length(De), mean_s , sigma_s)  
  s <- abs(s) * De/100  
  
  ## run modelling  
  results <- combine_De_Dr(  
    De = De,  
    s = s,  
    Dr = Dr,  
    int_OD = int_OD,  
    Age_range = age_range,  
    method_control = list(  
      n.iter = n.iter  
    ),  
    verbose = FALSE,  
    plot = FALSE)  
  
  return(list(Dr, De, s, results))  
})
```





	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	49.23	48.33	50.04	47.57	50.95	0
	2	48.37	47.62	49.24	46.82	50.07	1
	3	48.45	47.74	49.39	46.97	50.21	0
	4	48.74	47.86	49.53	47.09	50.38	0
	5	49.09	48.25	49.92	47.43	50.78	1
mean	—	48.78	47.96	49.62	47.17	50.48	—
sd	—	0.38	0.31	0.35	0.32	0.38	—

S3.2 Norm(1000, 1, 0.1) + Norm(200, 1.4, 0.05)

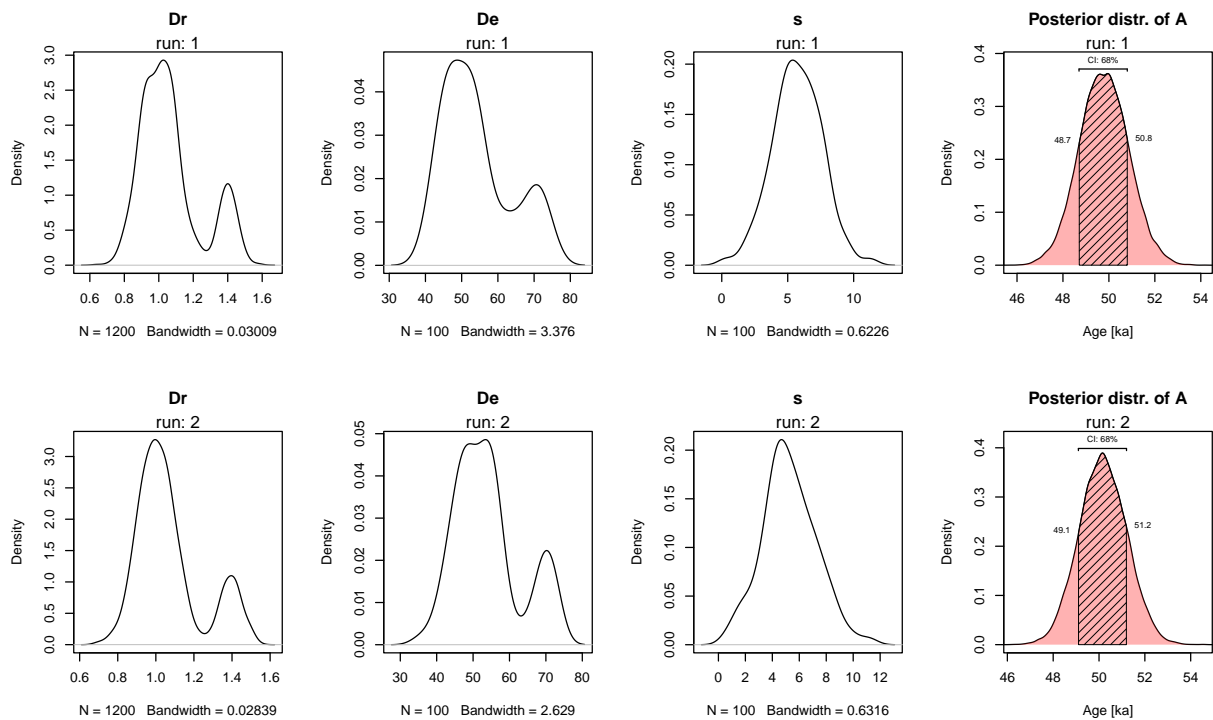
```

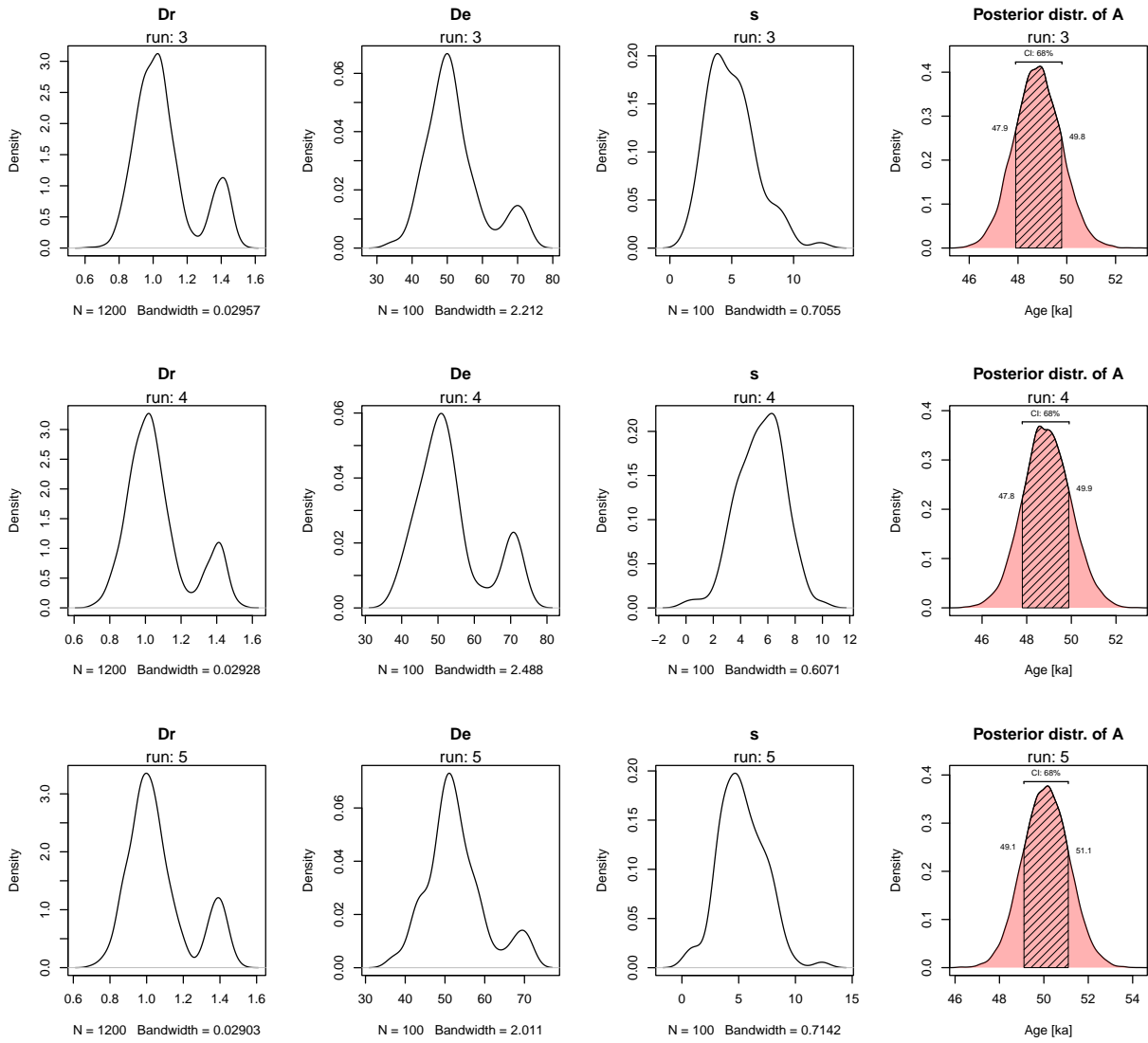
results2 <- lapply(1:5, function(i){
  # set parameters
  Dr <- c(stats::rnorm(1000, 1, 0.1), stats::rnorm(200, 1.4, 0.05))
  De <- supposed_age * sample(Dr, nb_aliquots, replace = TRUE)
  s <- stats::rnorm(length(De), mean_s , sigma_s)
  s <- abs(s) * De/100

  ## run modelling
  results <- combine_De_Dr(
    De = De,
    s = s,
    Dr = Dr,
    int_OD = int_OD,
    Age_range = age_range,
    method_control = list(
      n.iter = n.iter
    ),
    verbose = FALSE,
    plot = FALSE)

  return(list(Dr, De, s, results))
})

```





	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	49.95	48.71	50.82	47.68	51.86	5
	2	50.15	49.14	51.19	48.15	52.20	7
	3	48.90	47.86	49.76	46.92	50.69	4
	4	48.58	47.82	49.94	46.82	51.04	17
	5	50.16	49.05	51.12	48.06	52.12	5
mean	—	49.55	48.52	50.56	47.53	51.58	—
sd	—	0.75	0.64	0.67	0.63	0.68	—

S3.3 log-Norm(1000, 1, 0.1)

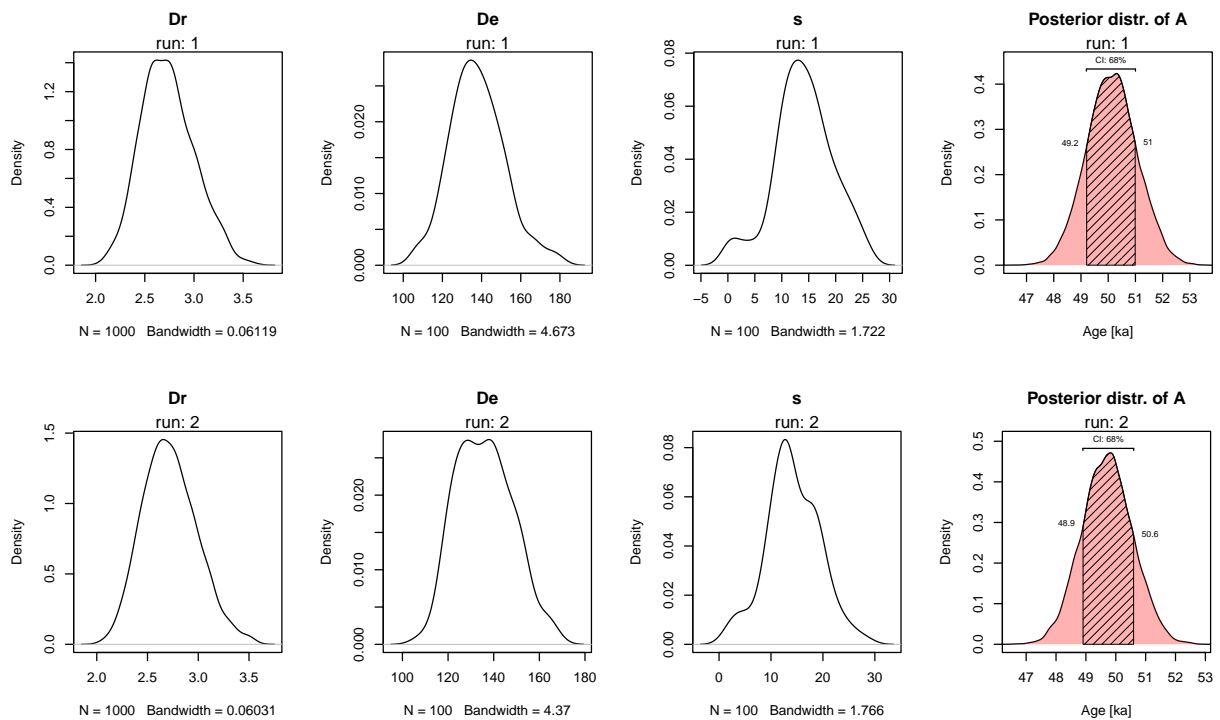
```

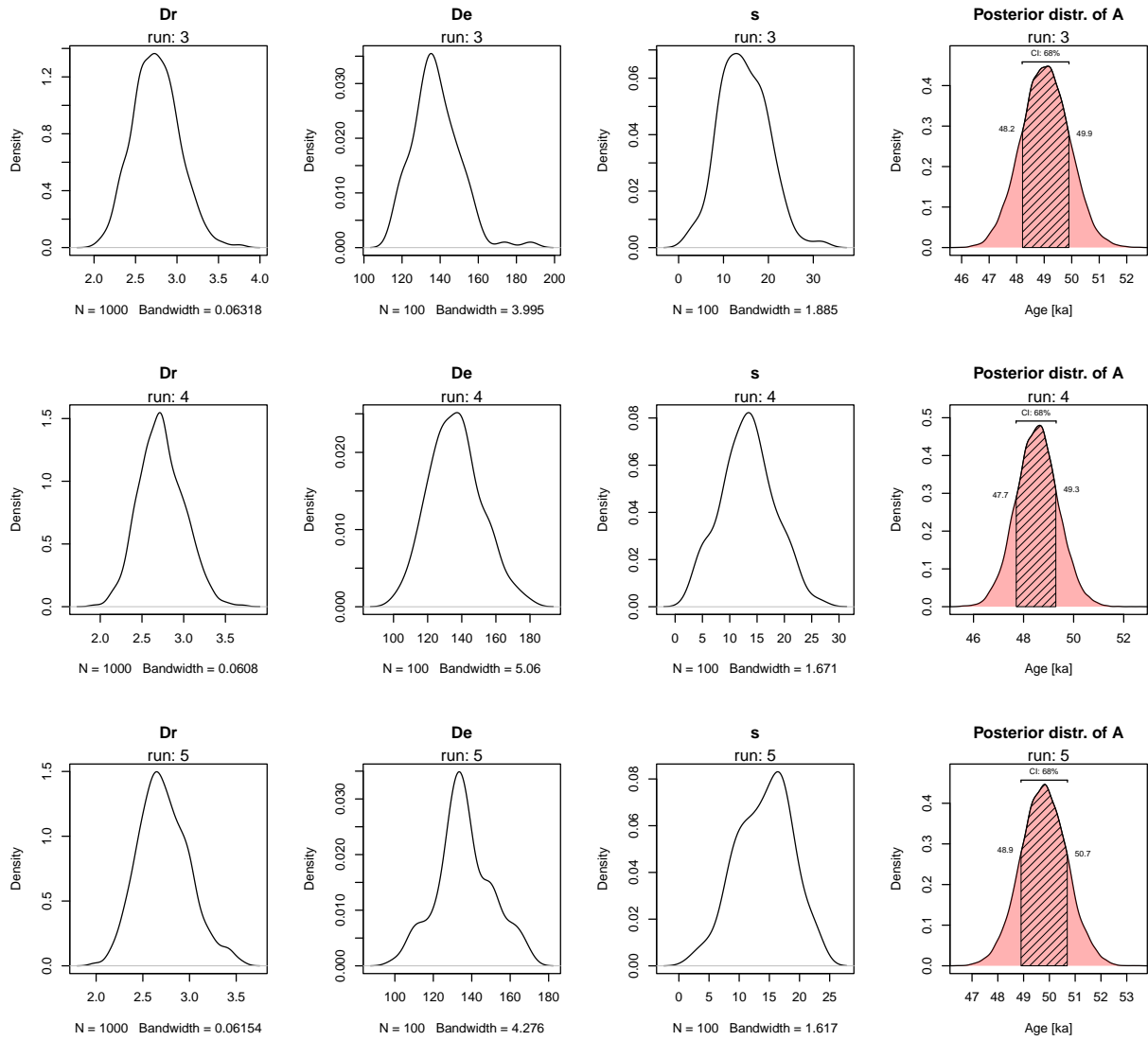
results3 <- lapply(1:5, function(i){
  # set parameters
  Dr <- c(stats::rlnorm(1000, 1, 0.1))
  De <- supposed_age * sample(Dr, nb_aliquots, replace = TRUE)
  s <- stats::rnorm(length(De), mean_s , sigma_s)
  s <- abs(s) * De/100

  ## run modelling
  results <- combine_De_Dr(
    De = De,
    s = s,
    Dr = Dr,
    int_OD = int_OD,
    Age_range = age_range,
    method_control = list(
      n.iter = n.iter
    ),
    verbose = FALSE,
    plot = FALSE)

  return(list(Dr, De, s, results))
})

```





	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	50.31	49.19	51.02	48.34	51.97	10
	2	49.82	48.86	50.56	48.07	51.39	0
	3	49.14	48.17	49.90	47.31	50.71	1
	4	48.65	47.69	49.33	46.89	50.19	1
	5	49.81	48.90	50.66	48.02	51.56	0
mean	—	49.55	48.56	50.29	47.73	51.16	—
sd	—	0.65	0.62	0.67	0.60	0.71	—

S3.4 log-Norm(1000, 1, 0.1) + log-Norm(200, 1.4, 0.05)

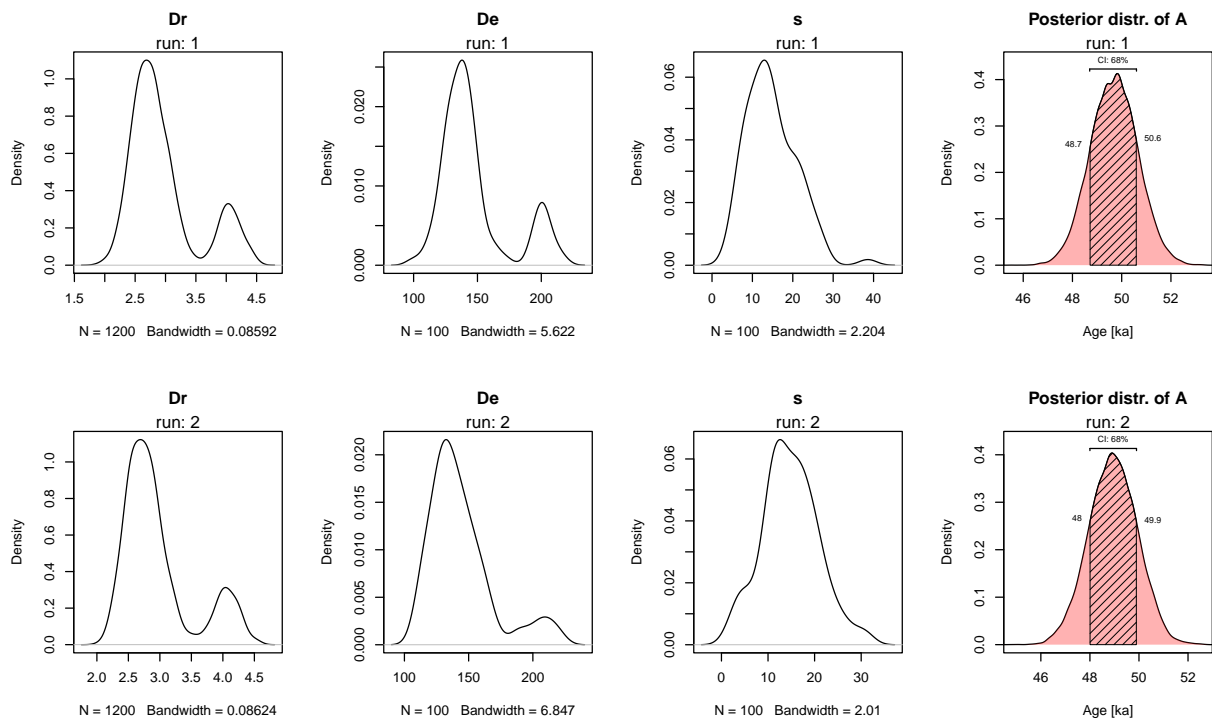
```

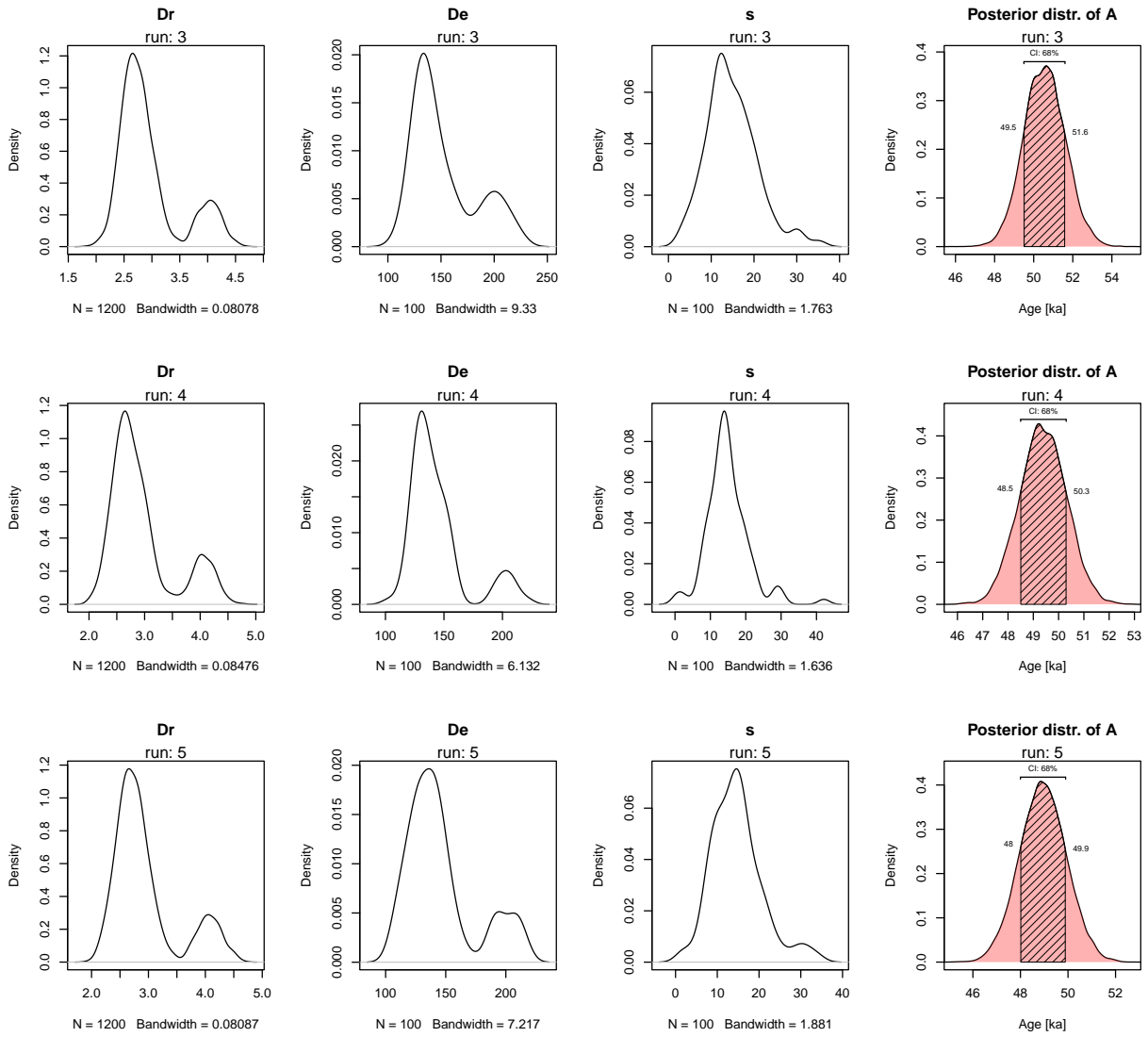
results4 <- lapply(1:5, function(i){
  # set parameters
  Dr <- c(stats::rlnorm(1000, 1, 0.1), stats::rlnorm(200, 1.4, 0.05))
  De <- supposed_age * sample(Dr, nb_aliquots, replace = TRUE)
  s <- stats::rnorm(length(De), mean_s , sigma_s)
  s <- abs(s) * De/100

  ## run modelling
  results <- combine_De_Dr(
    De = De,
    s = s,
    Dr = Dr,
    int_OD = int_OD,
    Age_range = age_range,
    method_control = list(
      n.iter = n.iter
    ),
    verbose = FALSE,
    plot = FALSE)

  return(list(Dr, De, s, results))
})

```





	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	49.82	48.73	50.63	47.81	51.56	5
	2	48.90	47.97	49.93	47.01	50.85	6
	3	50.64	49.49	51.56	48.48	52.61	9
	4	49.22	48.47	50.31	47.55	51.16	8
	5	48.85	47.97	49.90	47.01	50.85	11
mean	—	49.48	48.53	50.47	47.57	51.41	—
sd	—	0.75	0.63	0.68	0.61	0.73	—

S4 Table 2

S4.1 Norm(1000,1,0.1) - 50 x Norm(20,1.3,0.05)

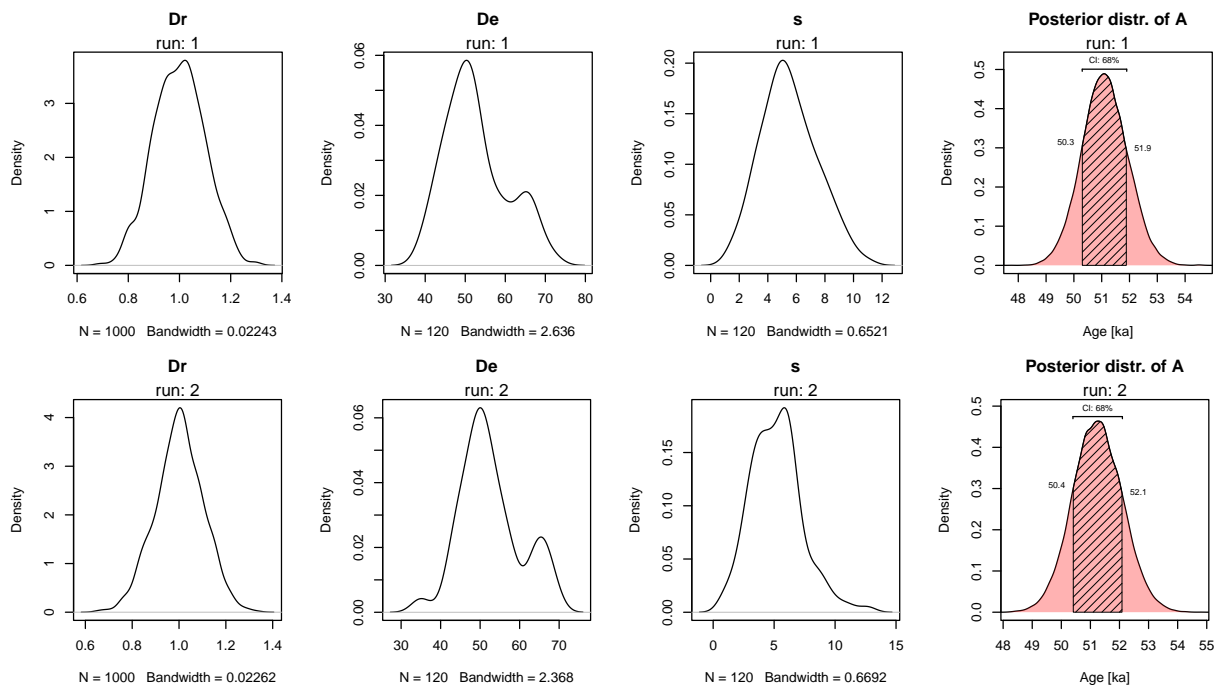
```

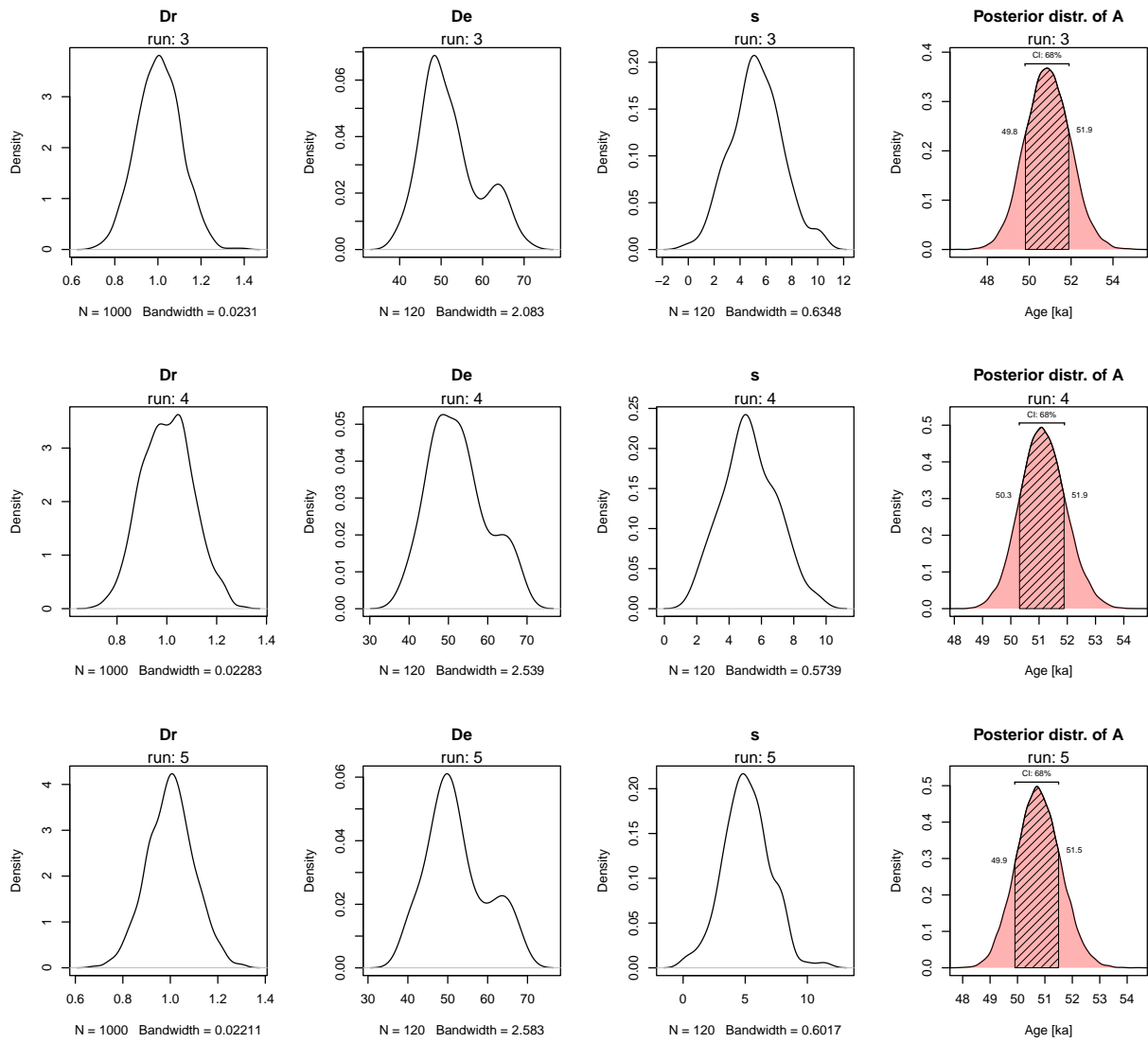
results_table2_a <- lapply(1:5, function(i){
  # set parameters
  Dr <- c(stats::rnorm(1000, 1, 0.1))
  De <- c(sample(Dr, nb_aliquots, replace = TRUE), rnorm(20,1.3,0.05))
  De <- supposed_age * De
  s <- stats::rnorm(length(De), mean_s , sigma_s)
  s <- abs(s) * De/100

  ## run modelling
  results <- combine_De_Dr(
    De = De,
    s = s,
    Dr = Dr,
    int_OD = int_OD,
    Age_range = age_range,
    method_control = list(
      n.iter = n.iter
    ),
    verbose = FALSE,
    plot = FALSE)

  return(list(Dr, De, s, results))
})

```





	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	51.09	50.27	51.89	49.49	52.70	3
	2	51.26	50.38	52.06	49.57	52.91	6
	3	50.86	49.80	51.92	48.79	52.97	56
	4	51.08	50.31	51.90	49.59	52.76	0
	5	50.70	49.95	51.54	49.17	52.32	9
mean	—	51.00	50.14	51.86	49.32	52.73	—
sd	—	0.22	0.25	0.19	0.34	0.25	—

S4.2 log-Norm(1000,1,0.1) - 50 x log-Norm(20,1.3,0.05)

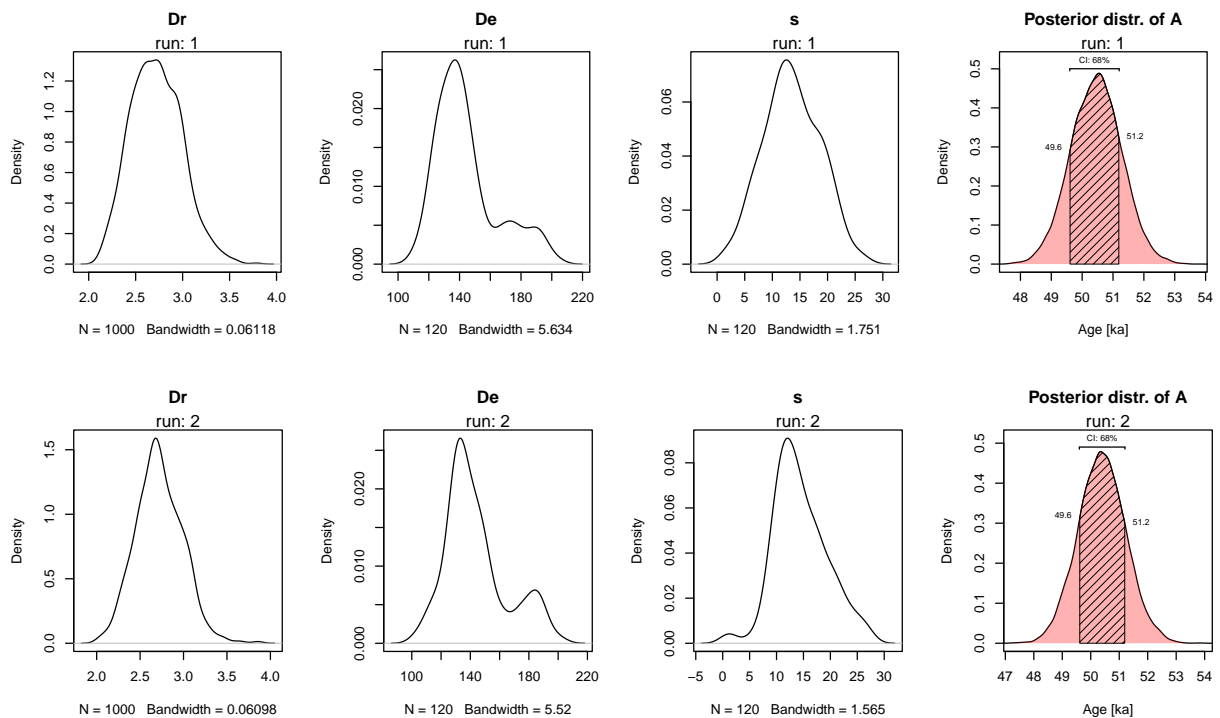
```

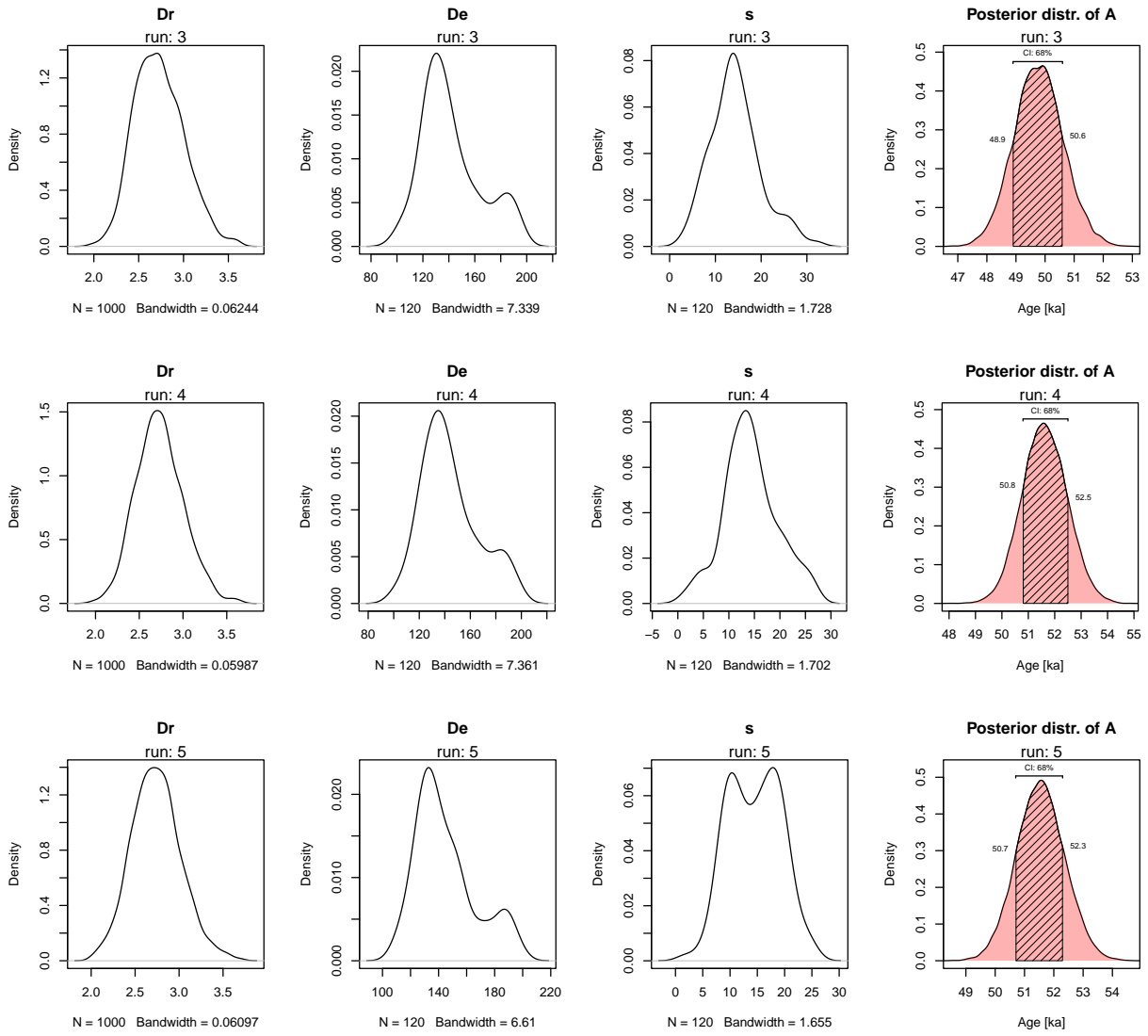
results_table2_b <- lapply(1:5, function(i){
  # set parameters
  Dr <- c(stats::rlnorm(1000, 1, 0.1))
  De <- c(sample(Dr, nb_aliquots, replace = TRUE), rlnorm(20,1.3,0.05))
  De <- supposed_age * De
  s <- stats::rnorm(length(De), mean_s , sigma_s)
  s <- abs(s) * De/100

  ## run modelling
  results <- combine_De_Dr(
    De = De,
    s = s,
    Dr = Dr,
    int_OD = int_OD,
    Age_range = age_range,
    method_control = list(
      n.iter = n.iter
    ),
    verbose = FALSE,
    plot = FALSE)

  return(list(Dr, De, s, results))
})

```





	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	50.55	49.62	51.24	48.83	52.04	9
	2	50.35	49.57	51.21	48.73	52.03	13
	3	49.91	48.93	50.59	48.14	51.45	9
	4	51.59	50.76	52.45	49.96	53.29	10
	5	51.56	50.71	52.32	49.95	53.15	5
mean	—	50.79	49.92	51.56	49.12	52.39	—
sd	—	0.75	0.79	0.80	0.80	0.79	—

S4.3 Norm(1000,1,0.1) - 50 x Norm(20,1.4,0.05)

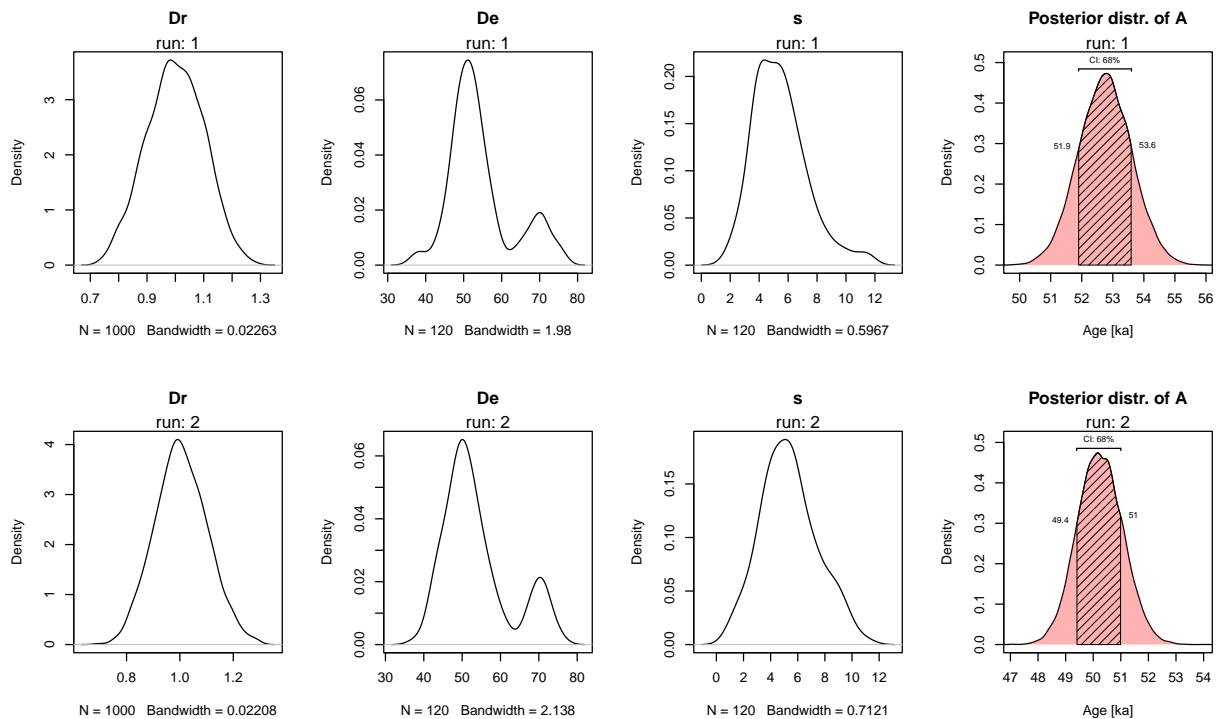
```

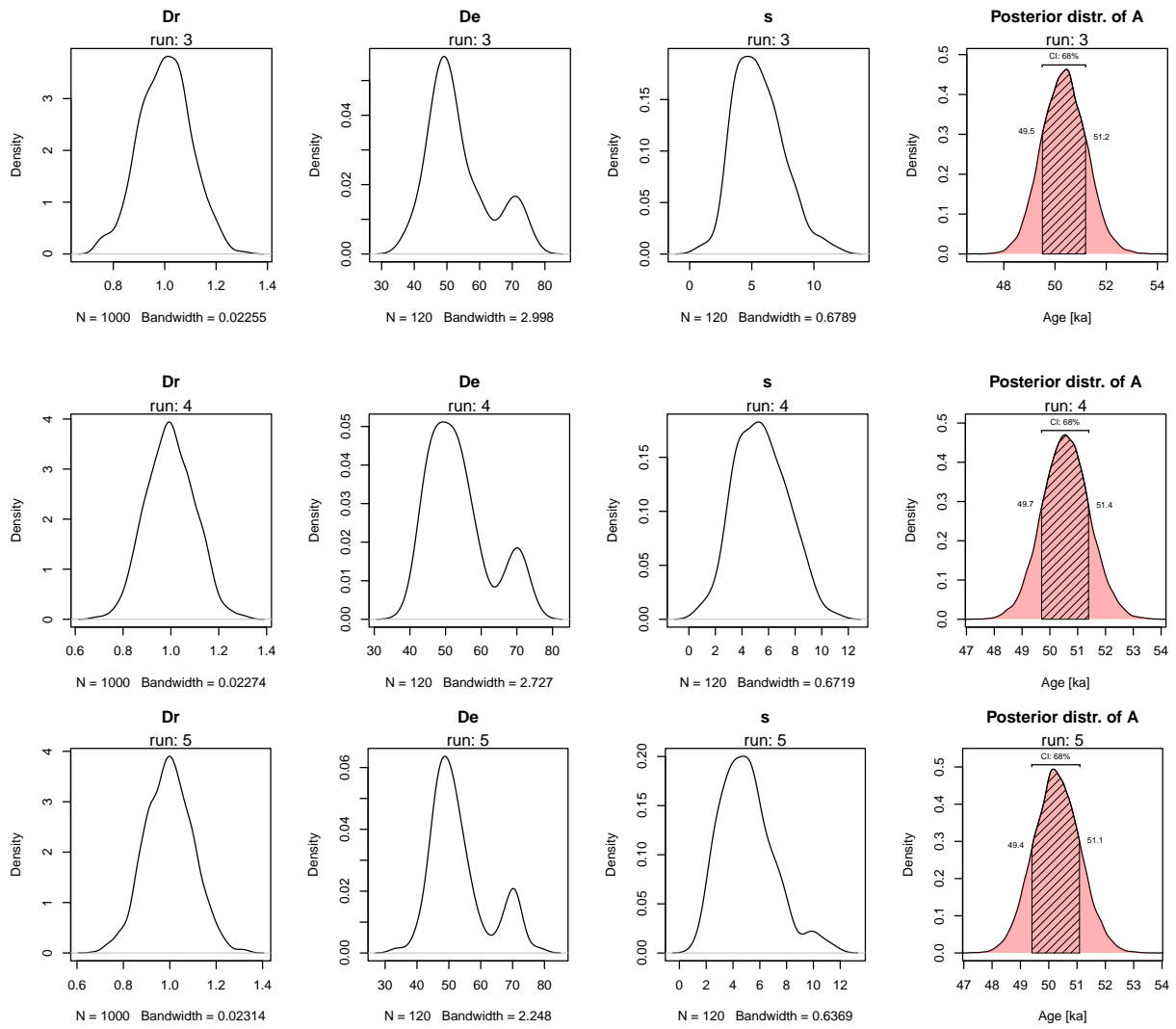
results_table2_c <- lapply(1:5, function(i){
  # set parameters
  Dr <- c(stats::rnorm(1000, 1, 0.1))
  De <- c(sample(Dr, nb_aliquots, replace = TRUE), rnorm(20,1.4,0.05))
  De <- supposed_age * De
  s <- stats::rnorm(length(De), mean_s , sigma_s)
  s <- abs(s) * De/100

  ## run modelling
  results <- combine_De_Dr(
    De = De,
    s = s,
    Dr = Dr,
    int_OD = int_OD,
    Age_range = age_range,
    method_control = list(
      n.iter = n.iter
    ),
    verbose = FALSE,
    plot = FALSE)

  return(list(Dr, De, s, results))
})

```





	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	52.79	51.92	53.59	51.08	54.41	6
	2	50.17	49.41	51.04	48.67	51.87	11
	3	50.45	49.46	51.18	48.67	52.01	12
	4	50.55	49.72	51.38	48.92	52.25	11
	5	50.16	49.45	51.07	48.69	51.90	12
mean	—	50.82	49.99	51.65	49.21	52.49	—
sd	—	1.11	1.08	1.09	1.05	1.08	—

S4.4 log-Norm(1000,1,0.1) - 50 x log-Norm(20,1.4,0.05)

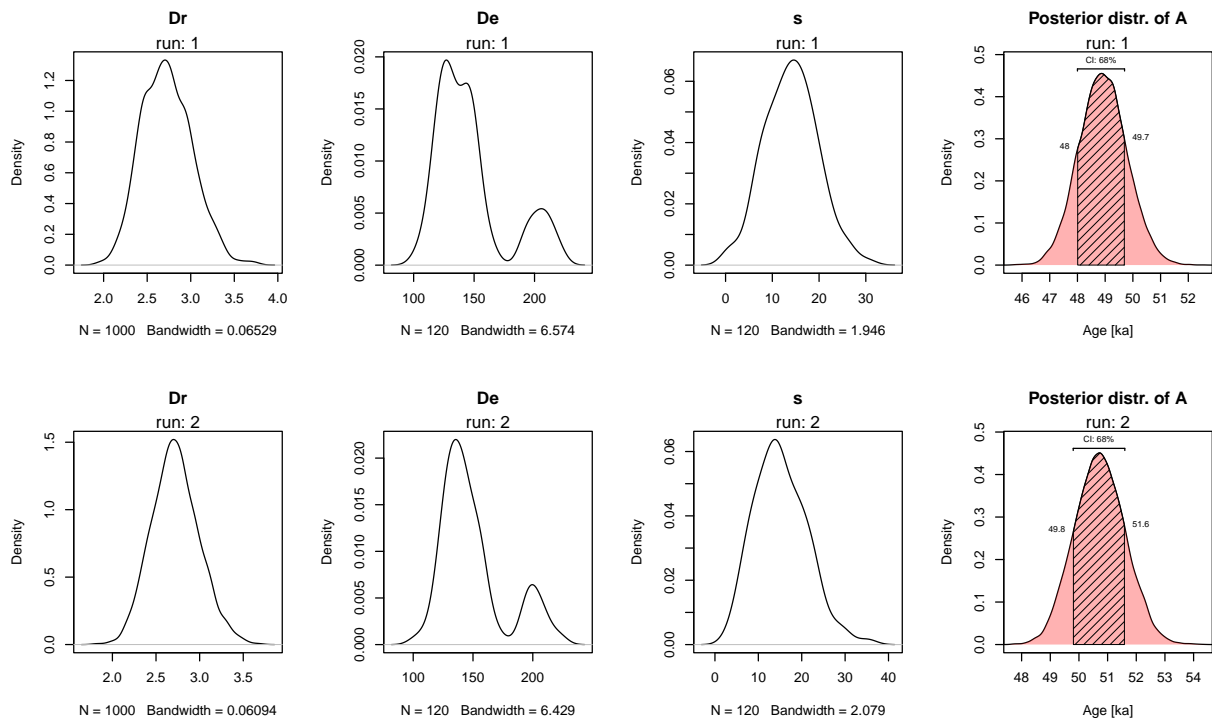
```

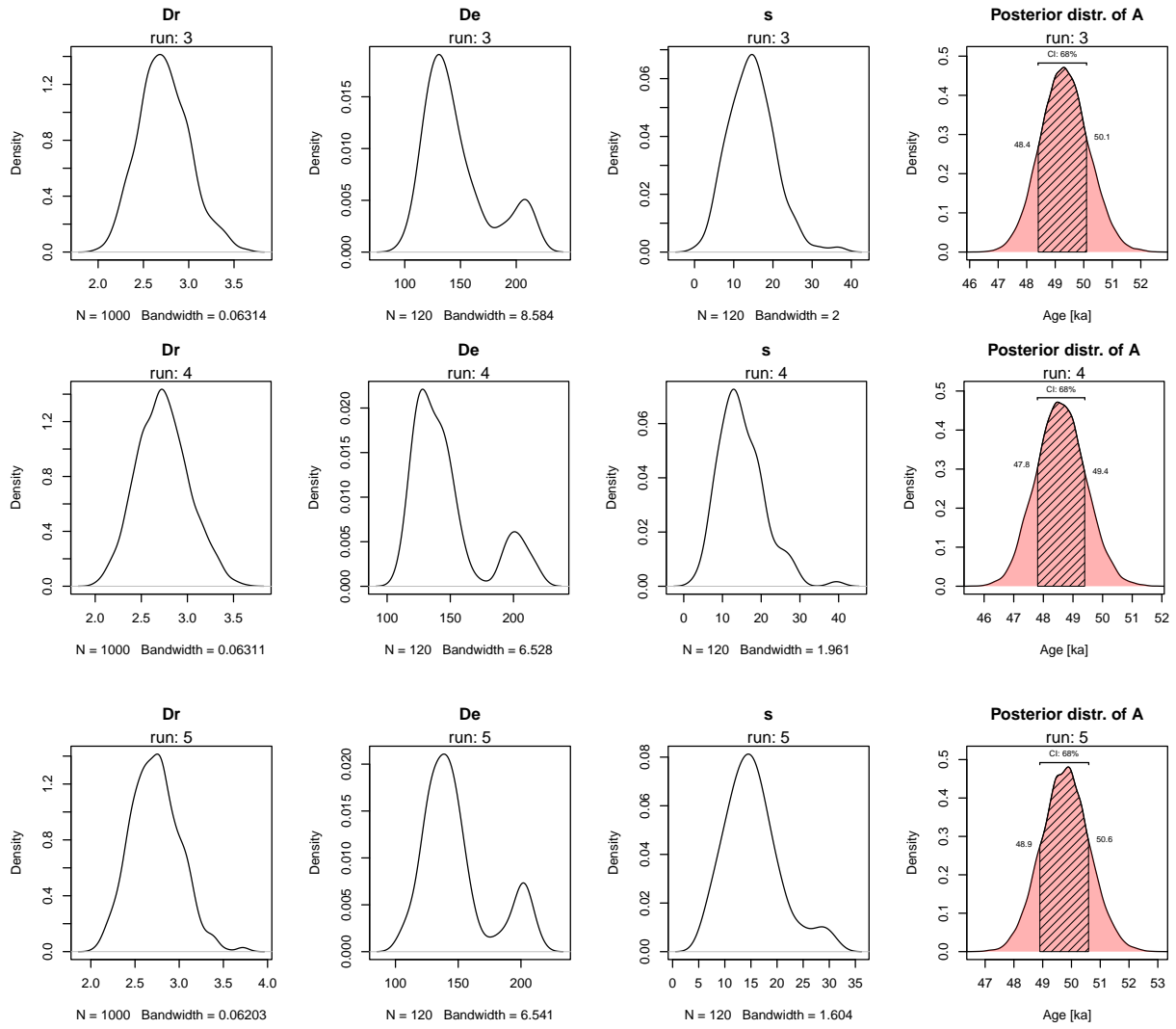
results_table2_d <- lapply(1:5, function(i){
  # set parameters
  Dr <- c(stats::rlnorm(1000, 1, 0.1))
  De <- c(sample(Dr, nb_aliquots, replace = TRUE), rlnorm(20,1.4,0.05))
  De <- supposed_age * De
  s <- stats::rnorm(length(De), mean_s , sigma_s)
  s <- abs(s) * De/100

  ## run modelling
  results <- combine_De_Dr(
    De = De,
    s = s,
    Dr = Dr,
    int_OD = int_OD,
    Age_range = age_range,
    method_control = list(
      n.iter = n.iter
    ),
    verbose = FALSE,
    plot = FALSE)

  return(list(Dr, De, s, results))
})

```





	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	48.87	48.02	49.74	47.23	50.62	23
	2	50.72	49.84	51.59	49.02	52.44	17
	3	49.30	48.45	50.11	47.68	50.98	20
	4	48.48	47.75	49.41	46.99	50.22	18
	5	49.89	48.94	50.61	48.14	51.42	16
mean	—	49.45	48.60	50.29	47.81	51.13	—
sd	—	0.88	0.82	0.85	0.81	0.86	—

S4.5 Norm(1000,1,0.1) - 50 x Norm(20,1.5,0.05)

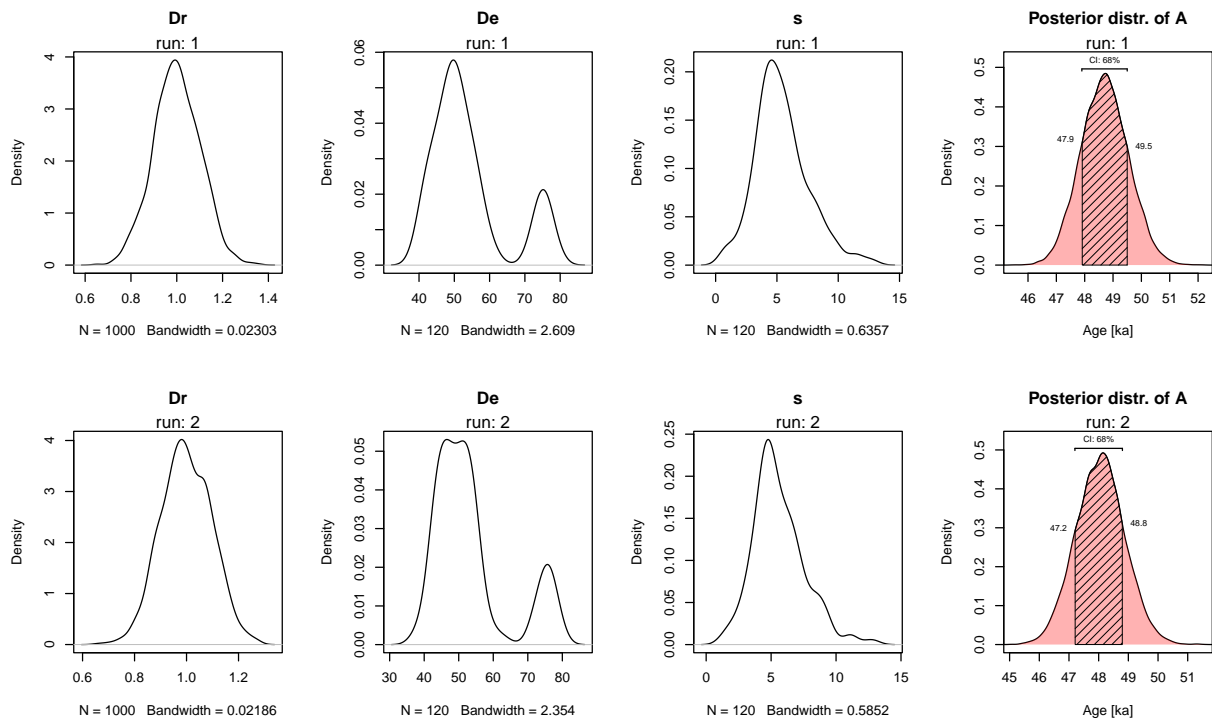
```

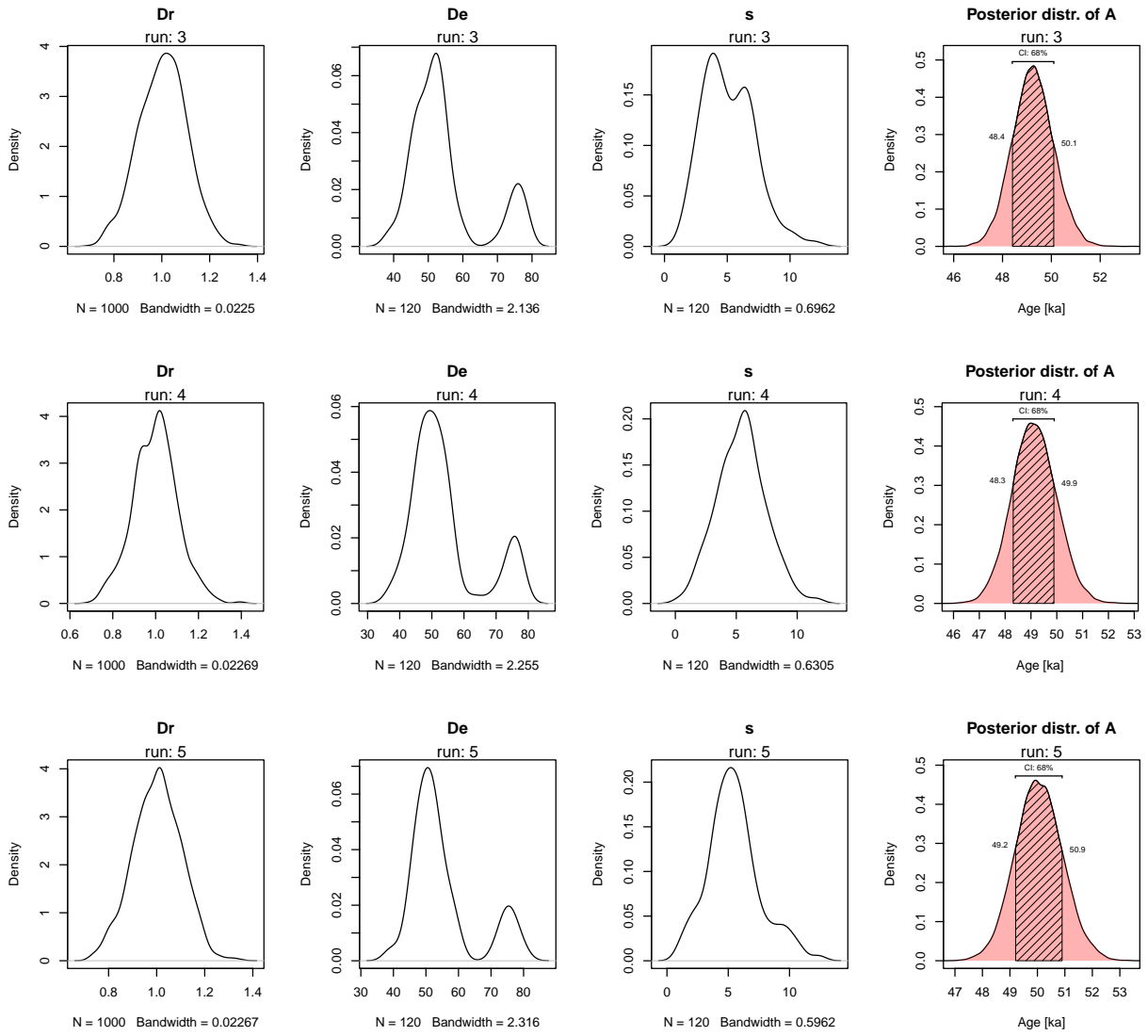
results_table2_e <- lapply(1:5, function(i){
  # set parameters
  Dr <- c(stats::rnorm(1000, 1, 0.1))
  De <- c(sample(Dr, nb_aliquots, replace = TRUE), rnorm(20,1.5,0.05))
  De <- supposed_age * De
  s <- stats::rnorm(length(De), mean_s , sigma_s)
  s <- abs(s) * De/100

  ## run modelling
  results <- combine_De_Dr(
    De = De,
    s = s,
    Dr = Dr,
    int_OD = int_OD,
    Age_range = age_range,
    method_control = list(
      n.iter = n.iter
    ),
    verbose = FALSE,
    plot = FALSE)

  return(list(Dr, De, s, results))
})

```





	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	48.73	47.86	49.51	47.07	50.31	20
	2	48.14	47.21	48.84	46.41	49.63	20
	3	49.28	48.39	50.05	47.60	50.92	19
	4	49.00	48.25	49.95	47.43	50.79	21
	5	49.93	49.19	50.88	48.37	51.75	16
mean	—	49.02	48.18	49.85	47.38	50.68	—
sd	—	0.66	0.73	0.75	0.72	0.78	—

S4.6 log-Norm(1000,1,0.1) - 50 x log-Norm(20,1.5,0.05)

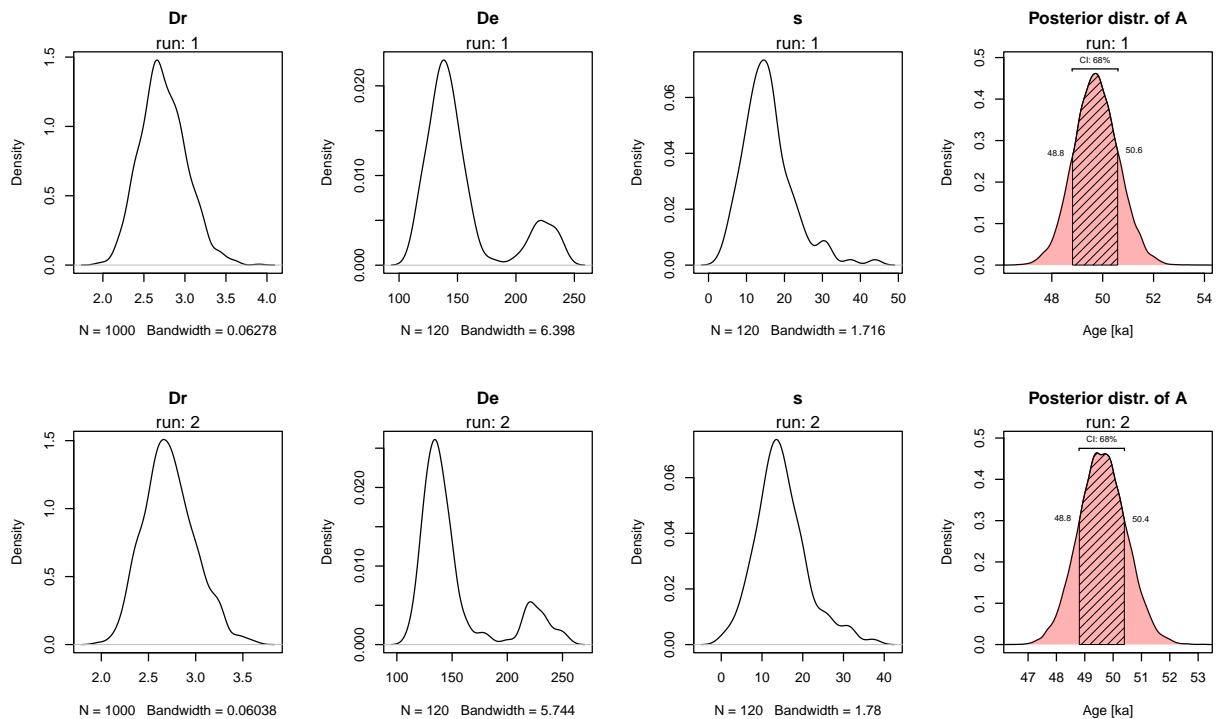
```

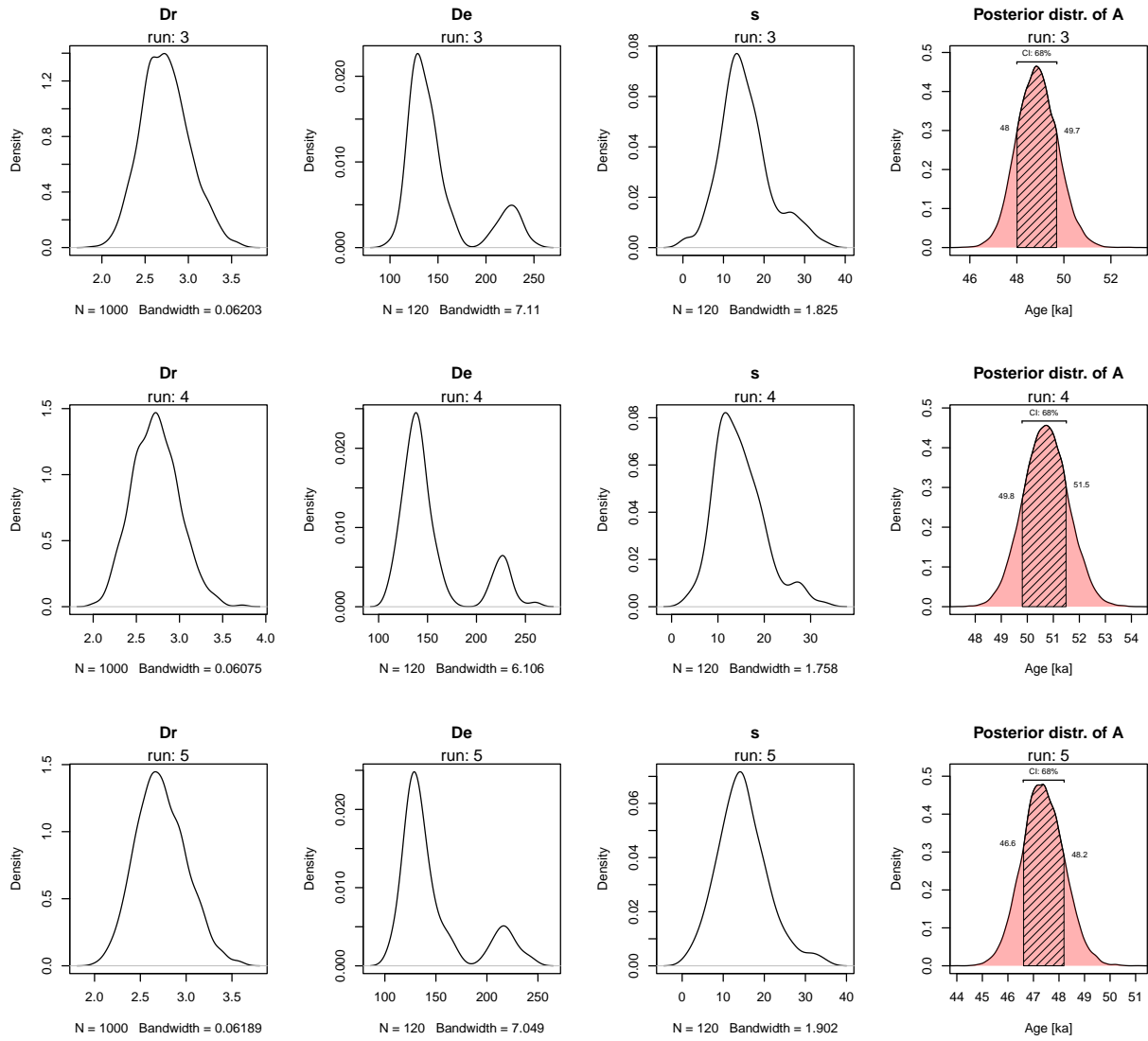
results_table2_f <- lapply(1:5, function(i){
  # set parameters
  Dr <- c(stats::rlnorm(1000, 1, 0.1))
  De <- c(sample(Dr, nb_aliquots, replace = TRUE), rlnorm(20,1.5,0.05))
  De <- supposed_age * De
  s <- stats::rnorm(length(De), mean_s , sigma_s)
  s <- abs(s) * De/100

  ## run modelling
  results <- combine_De_Dr(
    De = De,
    s = s,
    Dr = Dr,
    int_OD = int_OD,
    Age_range = age_range,
    method_control = list(
      n.iter = n.iter
    ),
    verbose = FALSE,
    plot = FALSE)

  return(list(Dr, De, s, results))
})

```





	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	49.72	48.85	50.57	48.06	51.49	20
	2	49.43	48.76	50.44	47.99	51.28	22
	3	48.83	47.99	49.69	47.20	50.56	22
	4	50.72	49.84	51.53	49.03	52.39	20
	5	47.36	46.55	48.17	45.76	48.97	20
mean	—	49.21	48.40	50.08	47.61	50.94	—
sd	—	1.24	1.22	1.26	1.22	1.28	—

S4.7 Norm(1000,1,0.1) - 50 x Norm(20,1.6,0.05)

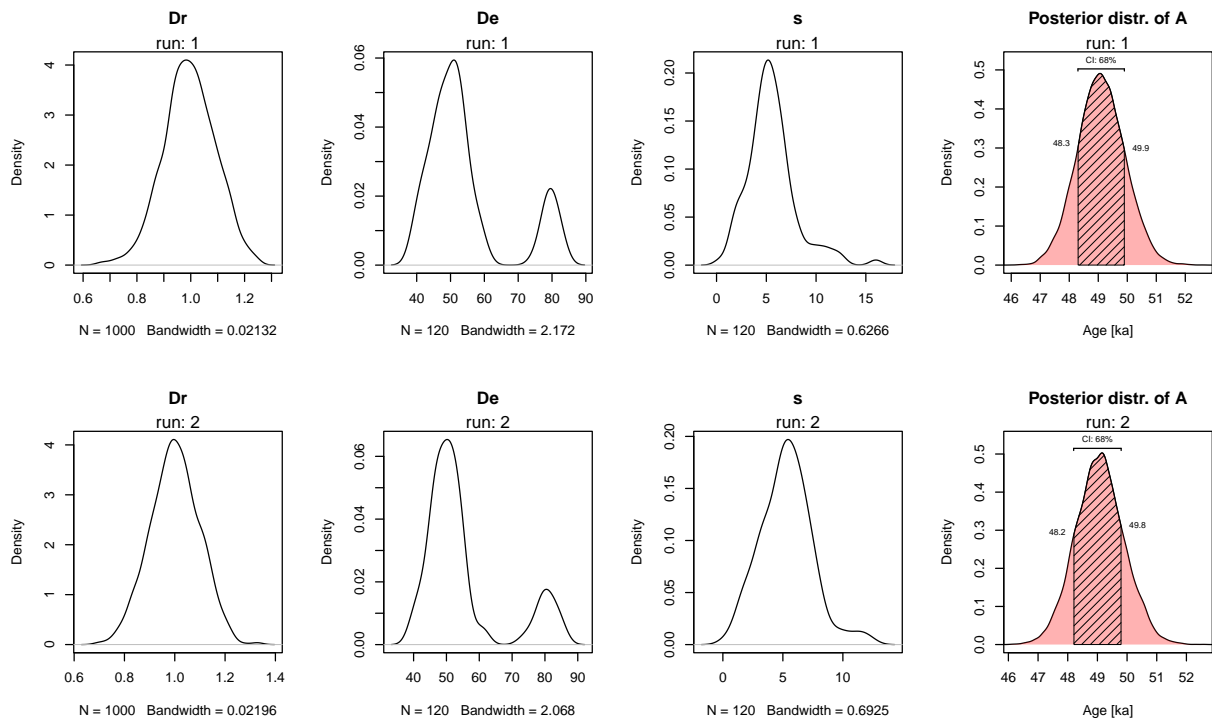
```

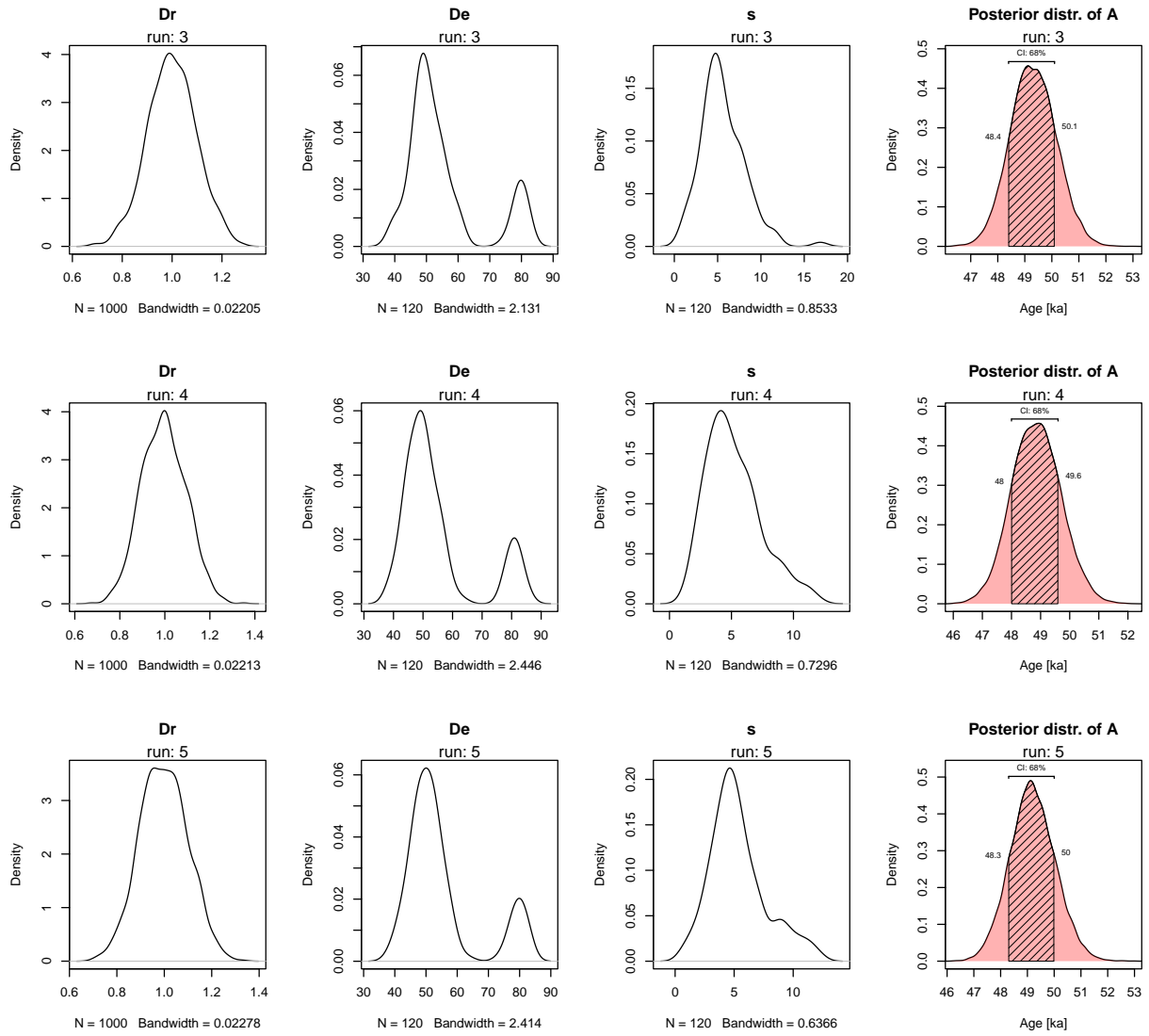
results_table2_g <- lapply(1:5, function(i){
  # set parameters
  Dr <- c(stats::rnorm(1000, 1, 0.1))
  De <- c(sample(Dr, nb_aliquots, replace = TRUE), rnorm(20,1.6,0.05))
  De <- supposed_age * De
  s <- stats::rnorm(length(De), mean_s , sigma_s)
  s <- abs(s) * De/100

  ## run modelling
  results <- combine_De_Dr(
    De = De,
    s = s,
    Dr = Dr,
    int_OD = int_OD,
    Age_range = age_range,
    method_control = list(
      n.iter = n.iter
    ),
    verbose = FALSE,
    plot = FALSE)

  return(list(Dr, De, s, results))
})

```





	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	49.06	48.28	49.88	47.50	50.71	19
	2	49.16	48.22	49.84	47.49	50.70	22
	3	49.12	48.45	50.12	47.67	50.97	20
	4	48.95	47.99	49.65	47.24	50.53	20
	5	49.12	48.32	49.99	47.53	50.84	21
mean	—	49.08	48.25	49.89	47.49	50.75	—
sd	—	0.08	0.17	0.18	0.16	0.16	—

S4.8 log-Norm(1000,1,0.1) - 50 x log-Norm(20,1.6,0.05)

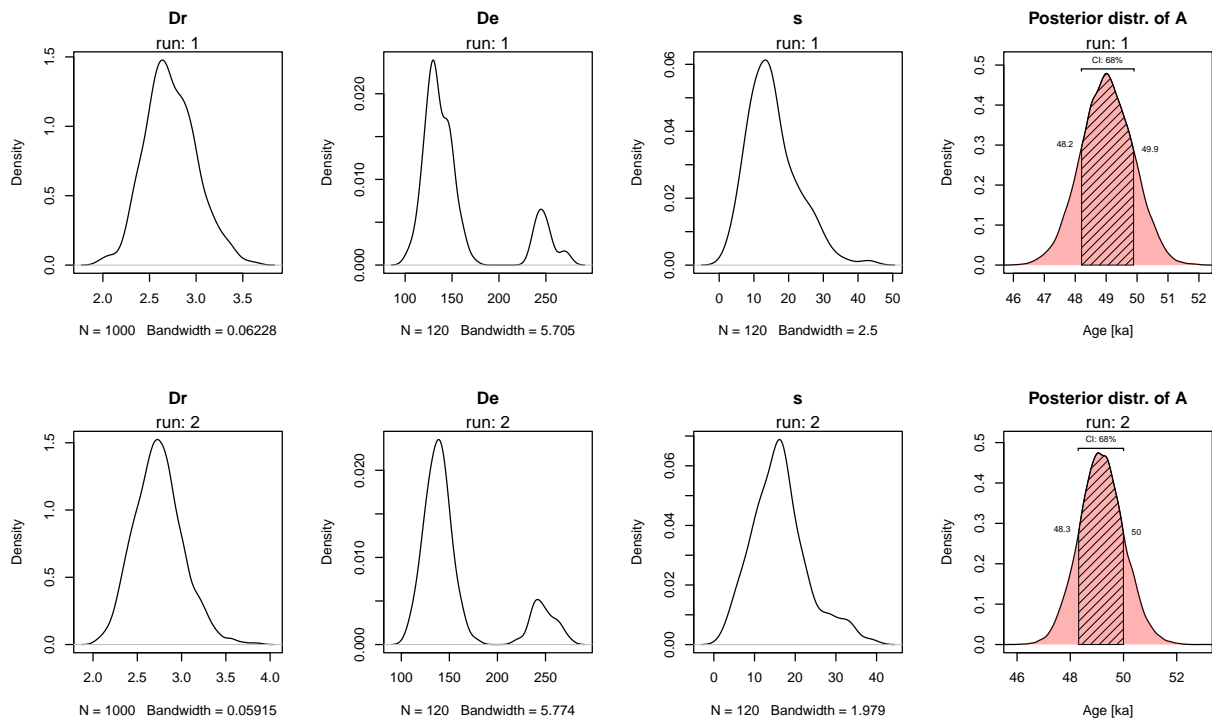
```

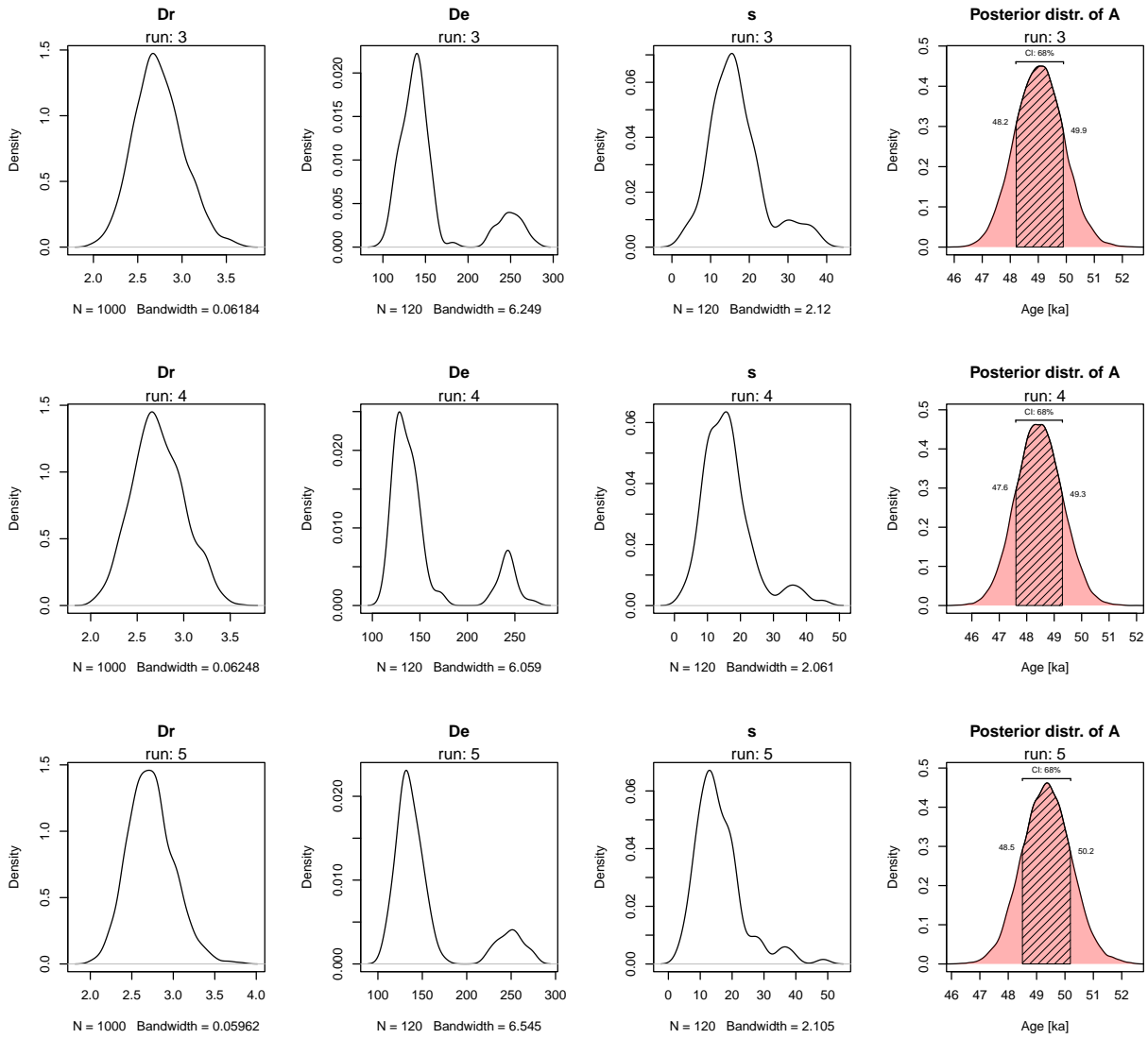
results_table2_h <- lapply(1:5, function(i){
  # set parameters
  Dr <- c(stats::rlnorm(1000, 1, 0.1))
  De <- c(sample(Dr, nb_aliquots, replace = TRUE), rlnorm(20,1.6,0.05))
  De <- supposed_age * De
  s <- stats::rnorm(length(De), mean_s , sigma_s)
  s <- abs(s) * De/100

  ## run modelling
  results <- combine_De_Dr(
    De = De,
    s = s,
    Dr = Dr,
    int_OD = int_OD,
    Age_range = age_range,
    method_control = list(
      n.iter = n.iter
    ),
    verbose = FALSE,
    plot = FALSE)

  return(list(Dr, De, s, results))
})

```





	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	49.01	48.19	49.87	47.40	50.68	21
	2	49.06	48.32	49.97	47.50	50.81	20
	3	49.08	48.17	49.88	47.34	50.69	21
	4	48.33	47.59	49.26	46.84	50.10	20
	5	49.37	48.47	50.18	47.66	51.05	20
mean	—	48.97	48.15	49.83	47.35	50.67	—
sd	—	0.38	0.33	0.34	0.31	0.35	—

S5 Table 3

S5.1 Norm(1000, 1, 0.1) - add an increasing number of De outliers

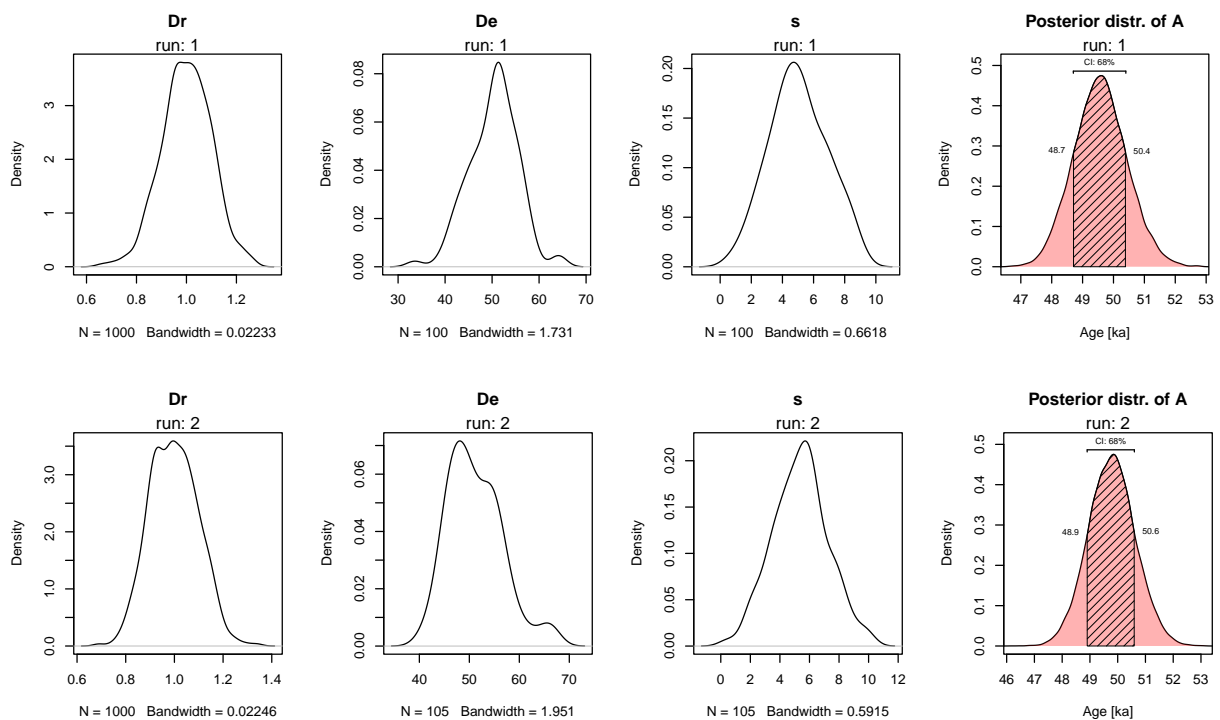
```

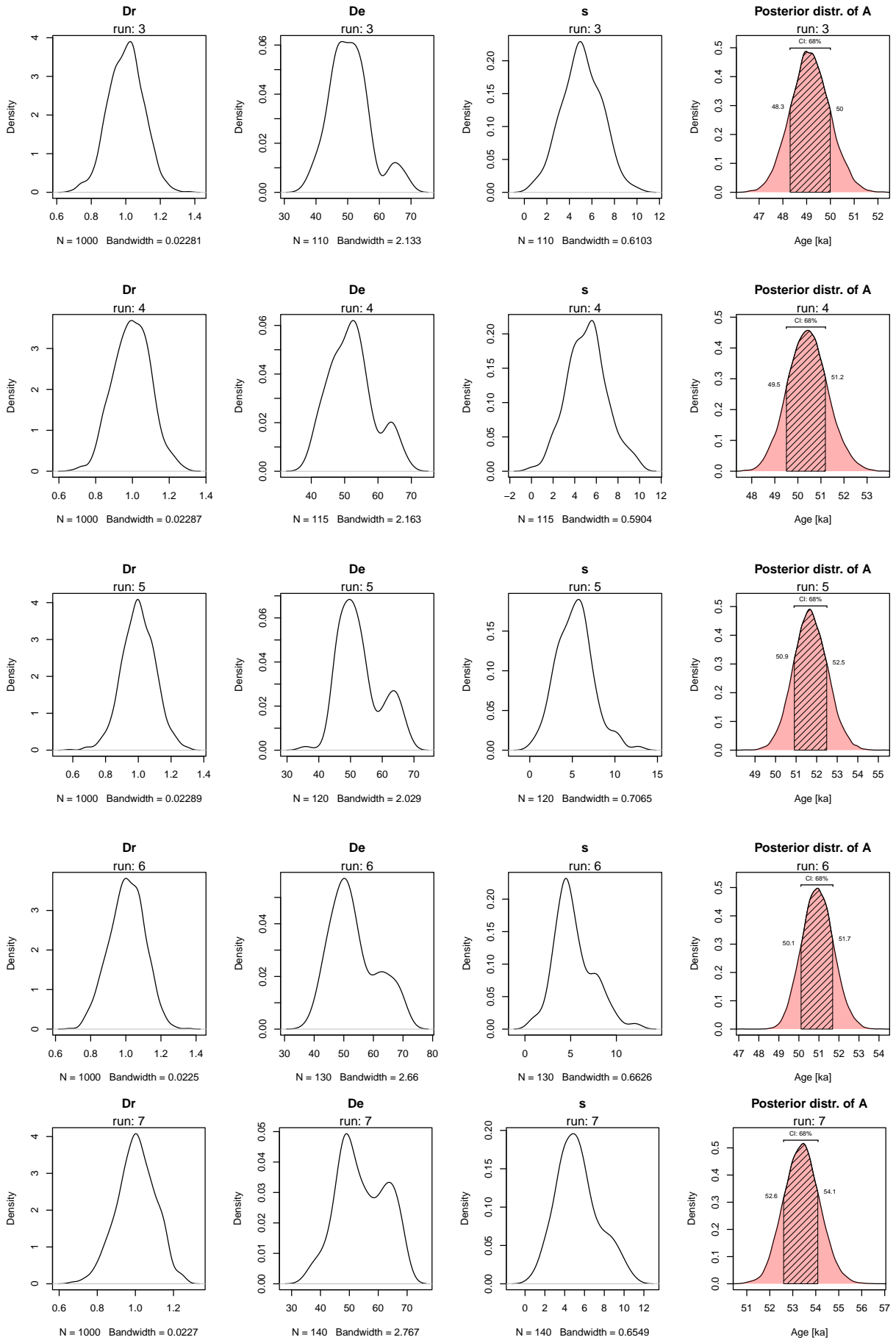
results_table3_a <- lapply(c(0,5,10,15,20,30,40,50), function(i){
  # set parameters
  Dr <- c(stats::rnorm(1000, 1, 0.1))
  De <- c(sample(Dr, nb_aliquots, replace = TRUE), rnorm(i,1.3,0.05))
  De <- supposed_age * De
  s <- stats::rnorm(length(De), mean_s , sigma_s)
  s <- abs(s) * De/100

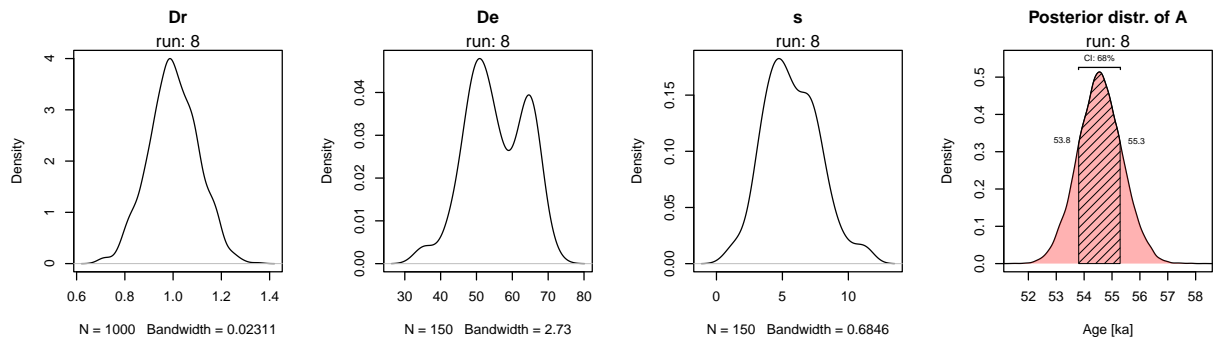
  ## run modelling
  results <- combine_De_Dr(
    De = De,
    s = s,
    Dr = Dr,
    int_OD = int_OD,
    Age_range = age_range,
    method_control = list(
      n.iter = n.iter
    ),
    verbose = FALSE,
    plot = FALSE)

  return(list(Dr, De, s, results))
})

```







	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	49.61	48.69	50.39	47.89	51.26	2
	2	49.85	48.92	50.58	48.12	51.46	1
	3	48.98	48.32	49.96	47.56	50.79	4
	4	50.43	49.53	51.23	48.72	52.14	9
	5	51.66	50.86	52.49	50.09	53.33	4
	6	50.94	50.14	51.70	49.40	52.47	6
	7	53.46	52.60	54.14	51.88	54.89	6
	8	54.56	53.76	55.34	53.00	56.13	5
mean	—	51.19	50.35	51.98	49.58	52.81	—
sd	—	1.95	1.95	1.91	1.97	1.87	—

S5.2 log-Norm(1000, 1, 0.1) - add an increasing number of De outliers

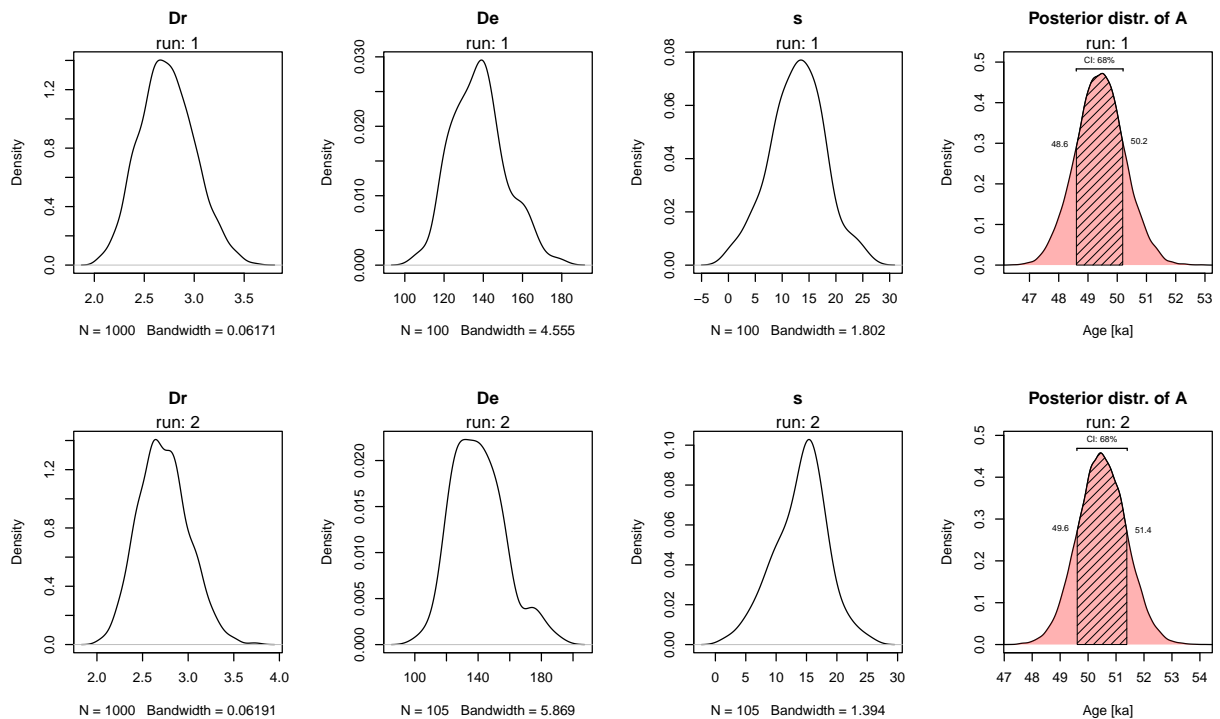
```

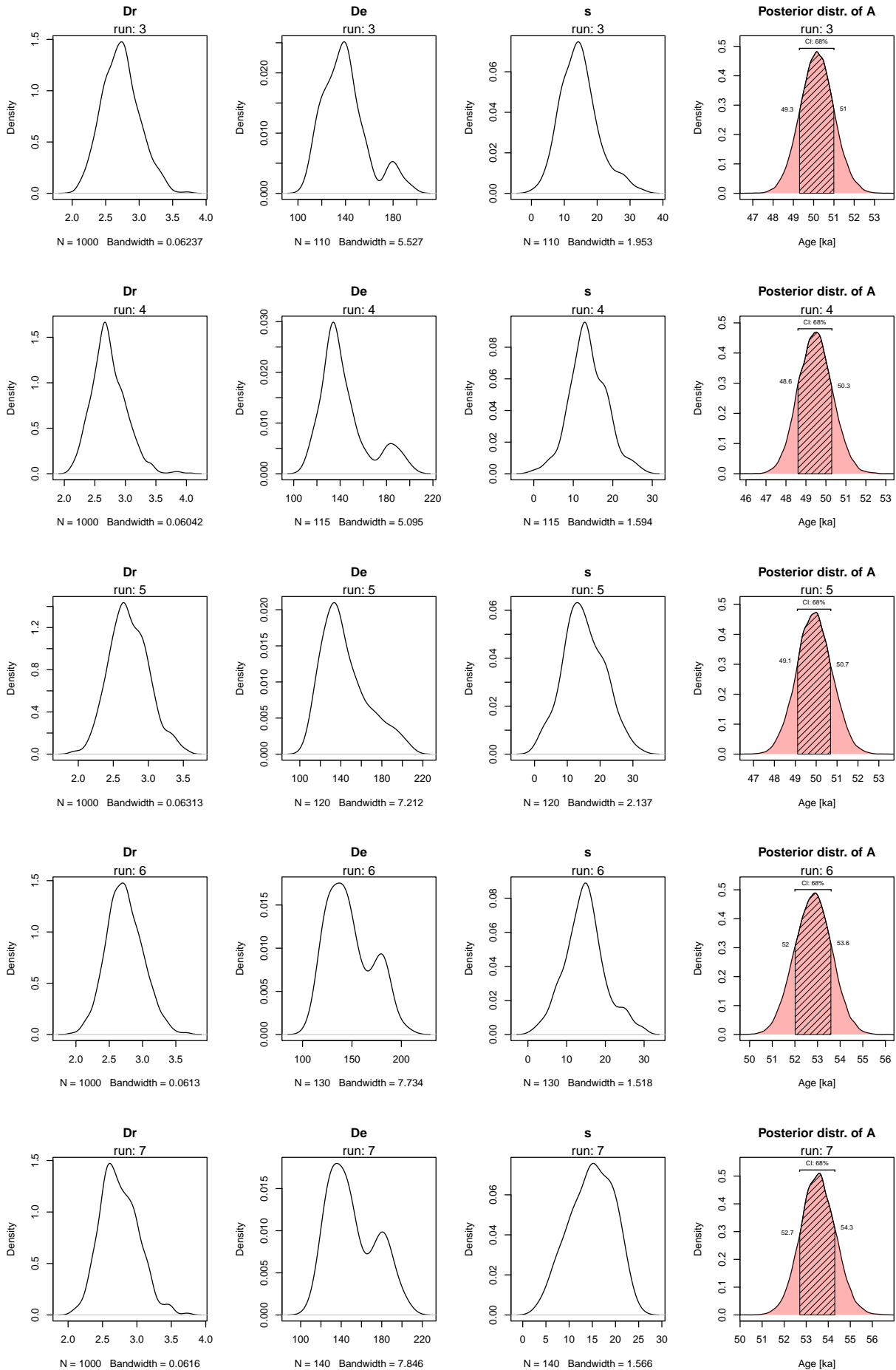
results_table3_b <- lapply(c(0,5,10,15,20,30,40,50), function(i){
  # set parameters
  Dr <- c(stats::rlnorm(1000, 1, 0.1))
  De <- c(sample(Dr, nb_aliquots, replace = TRUE), rlnorm(i,1.3,0.05))
  De <- supposed_age * De
  s <- stats::rnorm(length(De), mean_s , sigma_s)
  s <- abs(s) * De/100

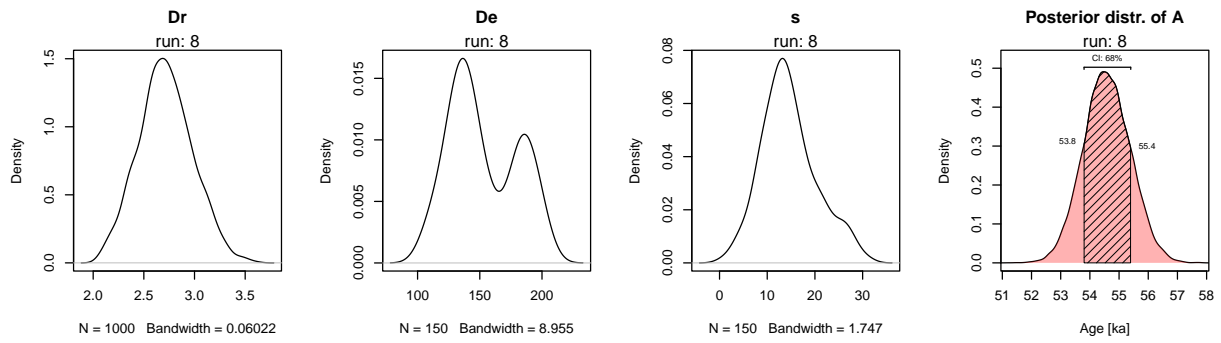
  ## run modelling
  results <- combine_De_Dr(
    De = De,
    s = s,
    Dr = Dr,
    int_OD = int_OD,
    Age_range = age_range,
    method_control = list(
      n.iter = n.iter
    ),
    verbose = FALSE,
    plot = FALSE)

  return(list(Dr, De, s, results))
})

```







	RUN	HPD	CI_68_lower	CI_68_upper	CI_95_lower	CI_95_upper	OUTLIERS
	1	49.49	48.59	50.23	47.74	51.05	4
	2	50.45	49.64	51.36	48.80	52.20	2
	3	50.16	49.33	50.97	48.53	51.80	3
	4	49.50	48.59	50.26	47.83	51.13	15
	5	50.00	49.05	50.72	48.25	51.55	10
	6	52.89	52.00	53.60	51.20	54.35	4
	7	53.59	52.75	54.31	52.00	55.08	5
	8	54.46	53.79	55.38	53.00	56.14	18
mean	—	51.32	50.47	52.10	49.67	52.91	—
sd	—	2.00	2.06	2.02	2.07	1.98	—