

# STAT4DS / HOMEWORK 3 - PART B

*Maria Luisa Croci, Valerio Antonini*

---

## DEEP-SKY-BAYES

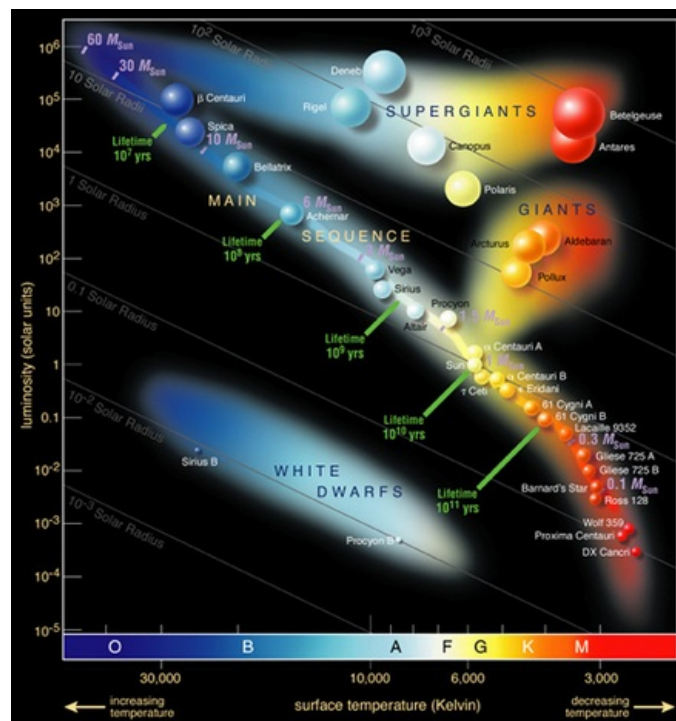
### Index

#### Brief Introduction

0. Summaries
1. Build the Model
2. OpenBUGS: Inference on  $\mu$  and  $\sigma$
3. JAGS: Inference on  $\mu$  and  $\sigma$
4. Change of the priors

#### Brief Introduction

The goal of the the following paper is to establish the stellar initial mass function (IMF) of 208 stars which belongs to the massive young cluster NGC 6611 aka *The Eagle Nebula*. The evolution of a star is generally explained through the *Hertzsprung - Russell diagram* (figure above) that shows the relationship between the stars' absolute magnitudes or luminosities (brightness) versus their stellar classifications or effective temperatures (color). Stars collapse from red giants to dwarf, and then, in the course of their lifetimes, move down along the so called main sequence, the diagonal that runs from the upper left to the lower right of the H-R diagram.



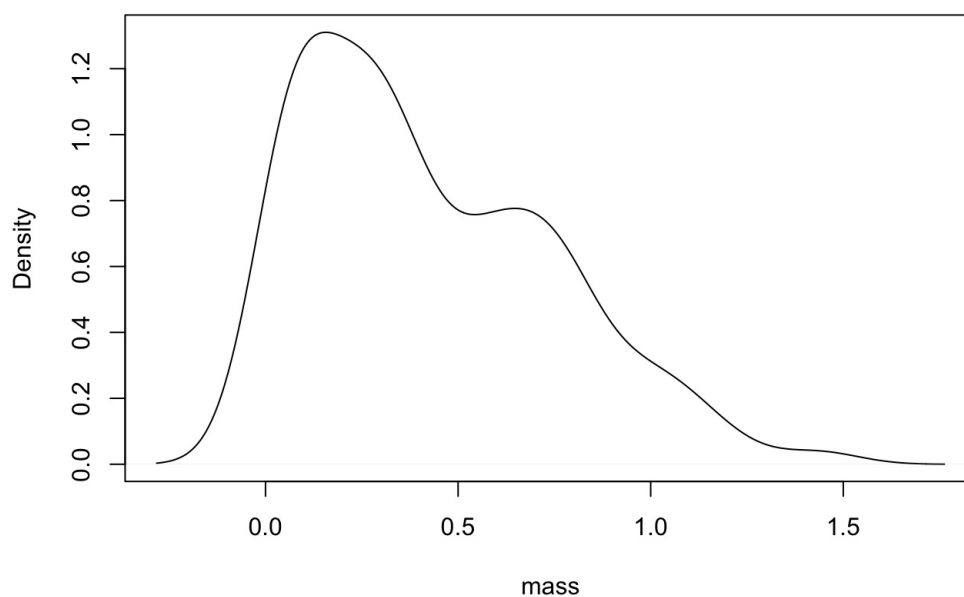
In astronomy, the *initial mass function* (IMF) is an empirical function that describes the initial distribution of masses for a population of stars. The IMF is often given as a probability distribution function (PDF) for the mass at which a star enters the main sequence.

We will perform our analysis assuming a lognormal distribution on the observations of the stellar mass.

## 0. Summaries

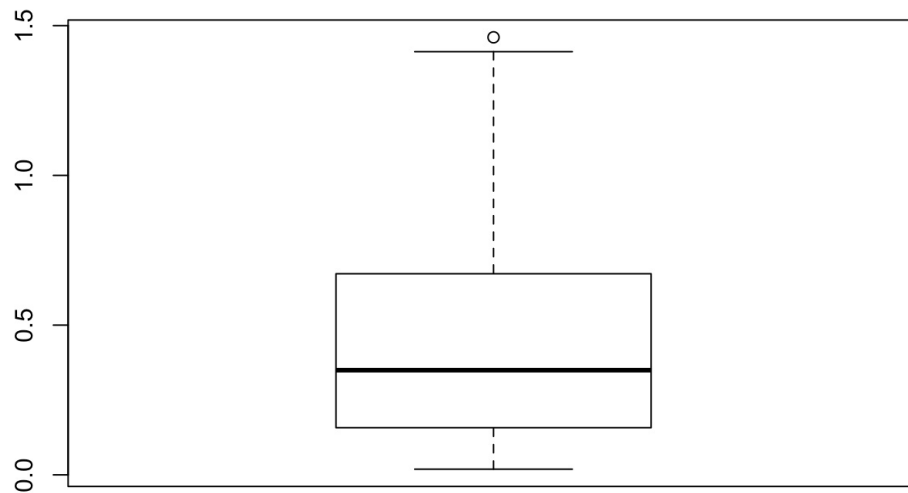
Let's start opening and exploring the dataset. We have 208 observation on the stellar masses distribute as follow:

**Distribution of stellar mass**



Looking quickly at the following boxplot we expect to have values that goes from around 0 up to 1.5 and a median little less than 0.5. Moreover we expect to have an asymmetric distribution since the first and third quantile don't have the same distance from the median (we will check this with the skewness).

## Boxplot of the stellar masses



Using in R the function `summary()` we can confirm our result:

```
library(dplyr)

summary(dat)

##      Mass
##  Min.   :0.0190
## 1st Qu.:0.1578
##  Median:0.3495
##   Mean  :0.4274
## 3rd Qu.:0.6713
##   Max.   :1.4610
```

Other important statistics are the skewness and the kurtosis, they respectively measure the asymmetry and the tailedness of the probability distribution. So technically these are used to describe the shape of a distribution.

```
library(moments)
print(paste0("skewness = ", round(skewness(dat$Mass), 3), " and kurtosis = ", round(kurtosis(dat$Mass), 3)))

## [1] "skewness = 0.73 and kurtosis = 2.811"
```

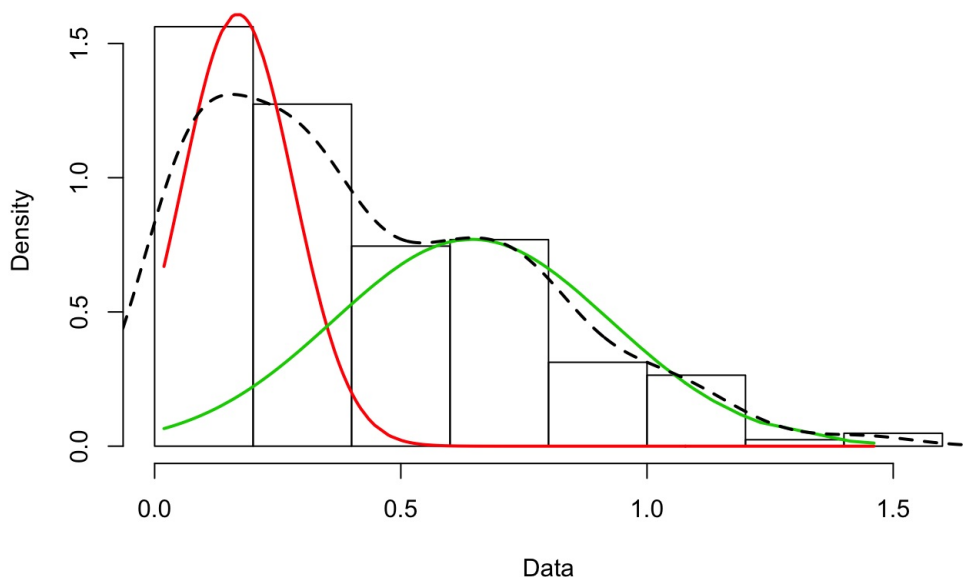
Looking at our outcome we can say that the distribution is moderately skewed and shorter, tails are thinner than the normal distribution.

In order to extend the information about the shape, we can show that our distribution is a mixture of normals:

```
library(mixtools)
x <- dat$Mass
mixmdl = normalmixEM(x) #extract means and variances for the normals distributions

plot(mixmdl, which=2) #plot the two normals red and green
lines(density(x), lty=2, lwd=2) #plot our data (the dotted one) and show that it fits as a mixture
```

## Density Curves



### 1. Build the Model

Since  $X \sim \text{LogNorm}(\mu, \sigma)$ , we could assume a non-informative Gaussian and Gamma priors for the mean and variance in order to build the model as a Doodle in OpenBUGS.

OpenBUGS is a software application for the Bayesian analysis of complex statistical models that use *Markov chain Monte Carlo* (MCMC) methods. These methods comprise a class of algorithms for sampling from a probability distribution. Through a Markov Chain, it can be obtained a sample of the desired distribution by observing the chain after a number of steps. The more steps there are, the more closely the distribution of the sample matches the actual desired distribution.

After several research, we have found that good non-informative prior distributions are as follow:

- $\mu \sim N(0, 0.0001)$
- $\sigma \sim \Gamma(0.0001, 0.0001)$

In the file *doodle.odc* there is the model as a doodle.

---

### 2. OpenBUGS: Inference on $\mu$ and $\sigma$

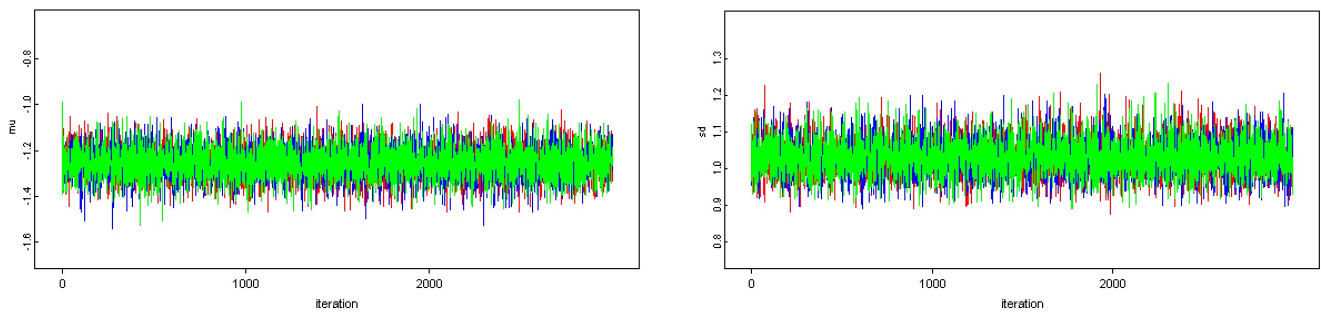
Since OpenBUGS has a different syntax from R, we worked on the precision that is the inverse of  $\sigma^2$  and we decided to run 3 chains using the initial values the same values  $\mu = 0$  and  $\sigma = 1$ .

Why do we run three chains? Because running multiple chains is one way to check the convergence of *Monte Carlo-Markov Chain*.

These are our results:

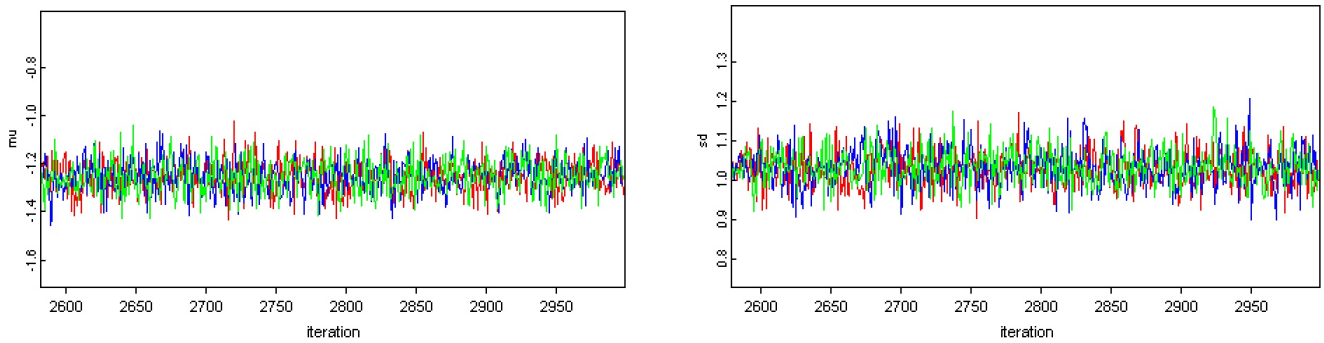
– On OpenBUGS the command *history* generates the following time series-type plot, where the chains are separated by colour.

## Time Series



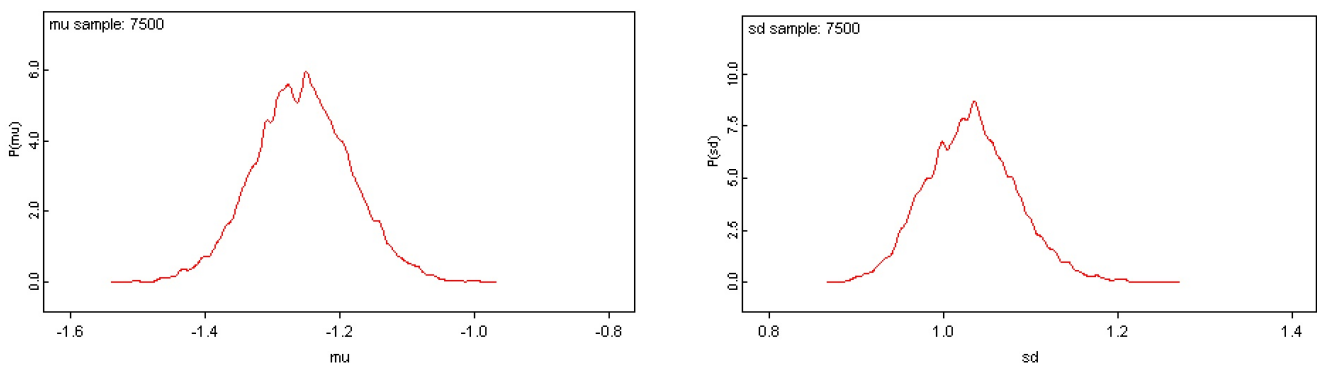
– The command *trace* provide a dynamic version.

## Trace Plot



– Looking at the plots the values of  $\mu$  and  $\sigma$  converge respectively around between  $[-1.2, -1.3]$  and  $[1.0, 1.1]$ .

## Density

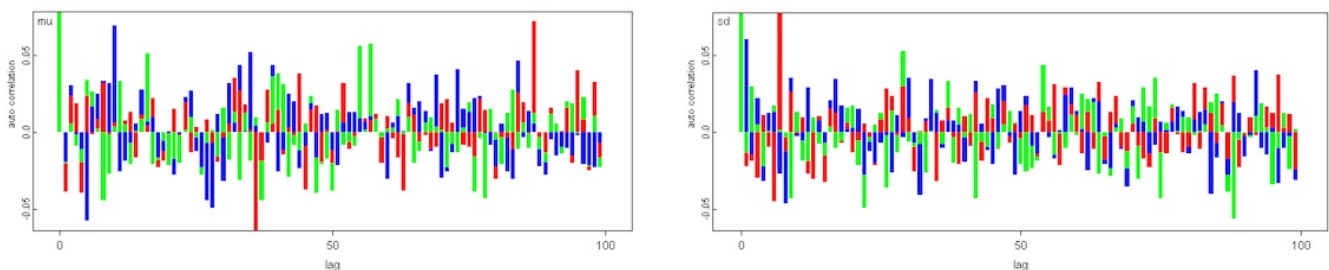


Our expectation is confirmed by the statistics table:

	mean	sd	MC_error	val2.5pc	median	val97.5pc	start	sample
$\mu$	-1.255	0.07144	8.309e-4	-1.395	-1.255	-1.111	501	7500
$\sigma$	1.033	0.05158	5.786e-4	0.9366	1.032	1.141	501	7500

Let's take a look to the autocorrelations:

## Autocorrelations



The autocorrelations also known as serial correlation or lagged correlation, explains the relationship between observations between the same variable over different periods of time. The value of autocorrelation can lie between +1 and -1. An autocorrelation of +1 represents perfect positive correlation. A value of -1 represents perfect negative correlation. The observations are said to be independent if autocorrelation is zero as in our plots.

---

### 3. JAGS: Inference on $\mu$ and $\sigma$

Let's see first the sample mean and variance. Since we have a lognormal distribution the mean and the variance are:

$$E(X) = e^{\mu + \frac{\sigma^2}{2}} \quad V(X) = e^{2\mu + \sigma^2} (e^{\sigma^2} - 1)$$

and so the sample mean and variance are:

$$\mu = \log \frac{E(X)}{\sqrt{1 + \frac{V(X)}{E(X)^2}}} \quad \sigma^2 = \ln\left(1 + \frac{V(X)}{E(X)^2}\right)$$

We can calculate this values in R:

```
mu_sample <- log( mean(x)/sqrt(1 + var(x)/ mean(x)^2))
sd_sample <- sqrt(log(1 + var(x)/mu_sample^2))
print(paste0("mu = ", round(mu_sample, 3), " and sd = ", round(sd_sample, 3)))

## [1] "mu = -1.078 and sd = 0.295"

library(R2jags)

model <- function() {
  # Prior
  mu ~ dnorm(0, 0.0001)
  sd ~ dgamma(0.0001, 0.0001)
  tau <- 1/pow(sd,2)
  for (i in 1:208) {
    # Likelihood
    x[i] ~ dlnorm(mu, tau) }
}

data <- list(x=x)

init <- function() list(mu=0, sd=1)

fit <- jags(
  model = model, #inits = init,
  parameters.to.save = c("mu", "sd"),
  data = data, n.iter=10000,
  n.chain = 3, n.burn=2000,
  DIC= FALSE)

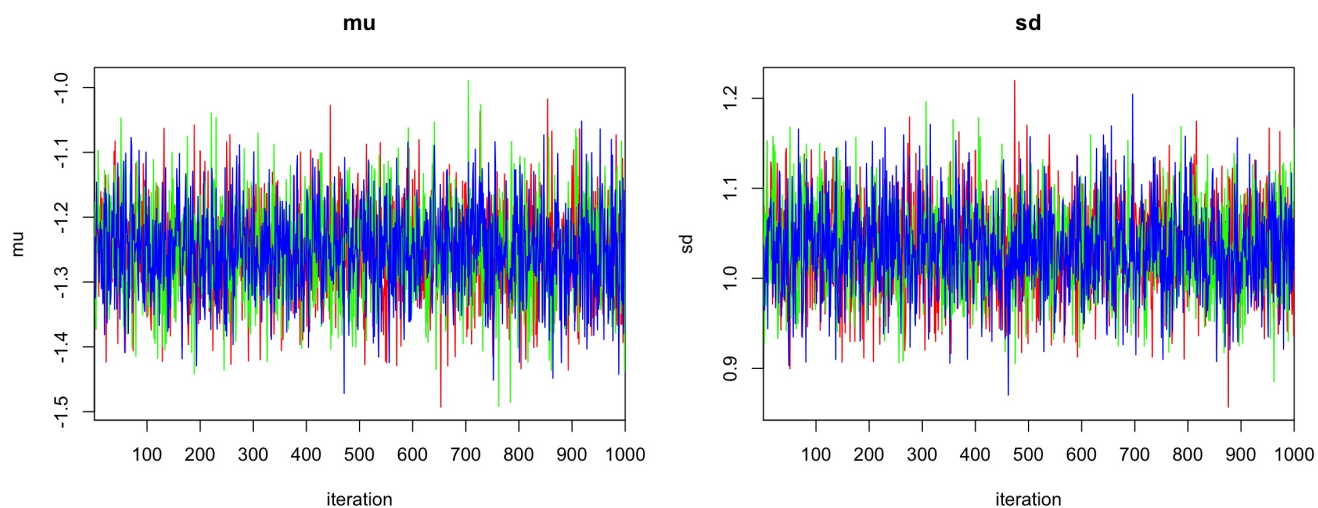
#Quick a look
fit

## Inference for Bugs model at "/var/folders/rr/6dg3xywn6dbd5bqm74411q240000gn/T//RtmpAMu0sa/modelcd0b46dd023
3.txt", fit using jags,
## 3 chains, each with 10000 iterations (first 2000 discarded), n.thin = 8
## n.sims = 3000 iterations saved
##      mu.vect sd.vect  2.5%    25%    50%    75%  97.5%  Rhat n.eff
## mu  -1.253   0.071 -1.390 -1.300 -1.254 -1.207 -1.112 1.001  3000
## sd   1.032   0.050  0.935  0.998  1.031  1.065  1.133 1.001  3000
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

As we can see the values are the same obtained in the table above with OpenBUGS.

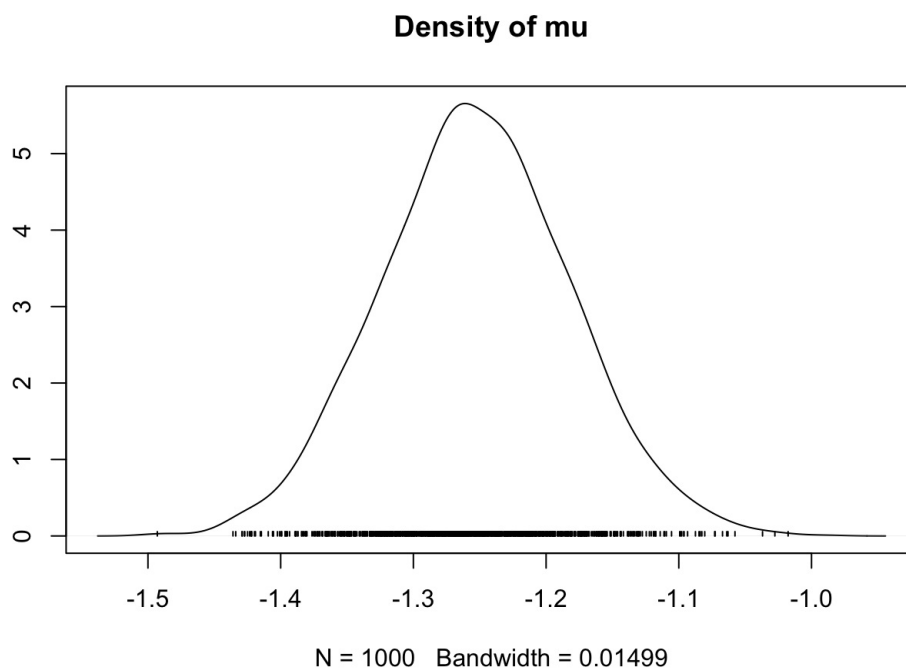
Also the traceplots look like the two done with OpenBUGS:

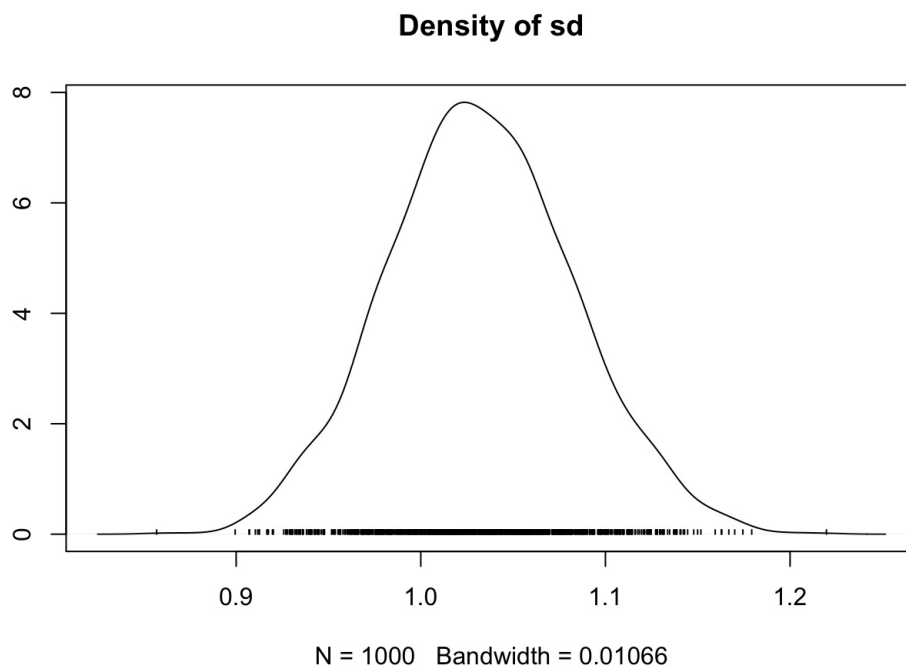
```
traceplot(fit, ask = FALSE, mfrow = c(1,2))
```



The following are the density plots:

```
mcfit <- as.mcmc(fit)
library(coda)
densplot(mcfit)
```





---

We can also focus on another measure called *Gelman-Rubin statistic* for looking at the convergence: it is analogous except that when we use it, we want its value to be close to 1.00 (no difference between the chains for a particular parameter). If a value is greater than 1.10 indicates inadequate mixing of the chains.

We can check this value with the R function:

```
gelman.diag(mcfit)

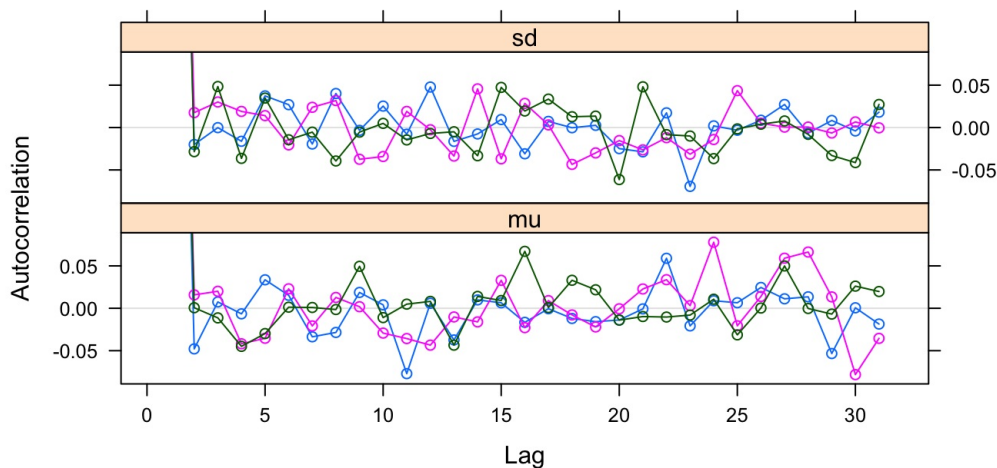
## Potential scale reduction factors:
##
##   Point est. Upper C.I.
## mu         1         1
## sd         1         1
##
## Multivariate psrf
##
## 1
```

---

The following is the autocorrelation according to the plots done with OpenBUGS:

```
acfplot(mcfit)
```





## 4. Change of the priors

Now let's see what happens changing the values of the priors when:

1.  $\mu \sim N(1, 0.001)$  and  $\sigma \sim \Gamma(0.001, 0.001)$

```
## Inference for Bugs model at "/var/folders/rr/6dg3xywn6dbd5bqm74411q240000gn/T//RtmpAMu0sa/modelcd0b220694d
9.txt", fit using jags,
## 3 chains, each with 10000 iterations (first 2000 discarded), n.thin = 8
## n.sims = 3000 iterations saved
## mu.vect sd.vect 2.5% 25% 50% 75% 97.5% Rhat n.eff
## mu -1.255 0.071 -1.394 -1.304 -1.256 -1.207 -1.115 1.001 3000
## sd 1.032 0.050 0.939 0.996 1.030 1.065 1.132 1.001 2900
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

2.  $\mu \sim N(1, 0.1)$  and  $\sigma \sim \Gamma(0.1, 0.1)$

```
## Inference for Bugs model at "/var/folders/rr/6dg3xywn6dbd5bqm74411q240000gn/T//RtmpAMu0sa/modelcd0b246abae
f.txt", fit using jags,
## 3 chains, each with 10000 iterations (first 2000 discarded), n.thin = 8
## n.sims = 3000 iterations saved
## mu.vect sd.vect 2.5% 25% 50% 75% 97.5% Rhat n.eff
## mu -1.254 0.072 -1.392 -1.302 -1.253 -1.207 -1.109 1.001 3000
## sd 1.035 0.053 0.940 0.998 1.032 1.067 1.145 1.002 1500
##
## For each parameter, n.eff is a crude measure of effective sample size,
## and Rhat is the potential scale reduction factor (at convergence, Rhat=1).
```

As we can see changing the values of the priors doesn't affect too much the outcomes, so it means that our results depend mostly on the choosing of the prior distributions and not from the parameters themselves.

We can conclude that the IMF is distributed as a lognormal:

$$IMF \sim \text{LogNorm}(-1.254, 1.033)$$

