Reading in the climate-model output.

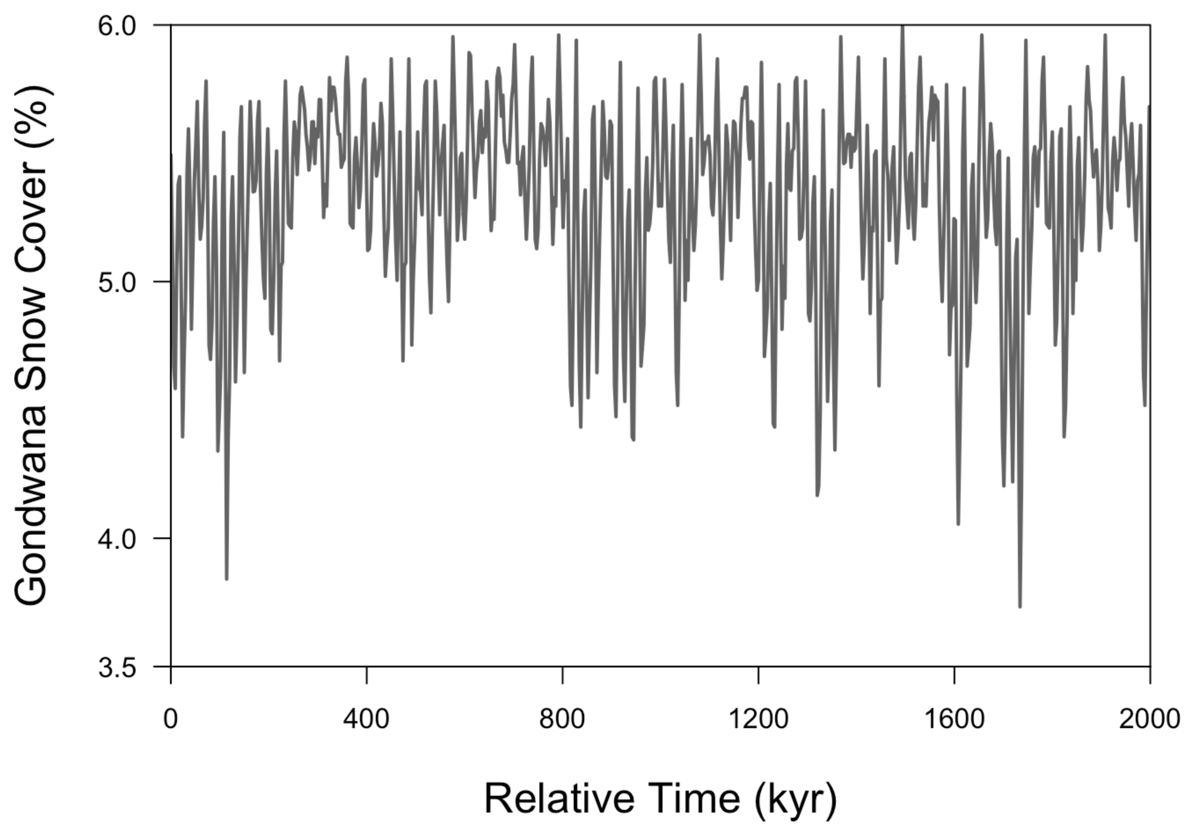
The climate output is 10 Myr long and has a 3 kyr time-resolution. The first 2 Myr of the series are shown in the figure. We add 10 000 kyr of relative time to get rid of negative values in the time column.

```
library(astrochron)
```

```
## Welcome to astrochron v0.9 (2019-01-08)
```

```
HadSM3_signal=read.csv(paste0(DIRECTORY,"Model_output_HadSM3_snowcover.csv"),header=F)
colnames(HadSM3_signal) <- c("Relative Time (kyr)", "Gondwana Snow cover (%)")
HadSM3_signal[,1]=HadSM3_signal[,1]+10000
```

```
par(mar=c(5,5,1,1))
plot(HadSM3_signal,type="l", ylim = c(3.5,6), xlim = c(0,2000), xaxs = "i", yaxs = "i", xaxt = "n", yaxt = "n", x
lab = "", ylab = "", col = "gray40", lwd = 2)
axis(1, at = c(0,400,800,1200,1600,2000), labels = c("0", "400", "800", "1200", "1600", "2000"), las = 1)
mtext("Relative Time (kyr)", side = 1, line = 3.5, cex = 1.5)
axis(2, at = c(3.5,4,5,6), las = 1)
mtext("Gondwana Snow Cover (%)", side = 2, line = 3.5, cex = 1.5)
```



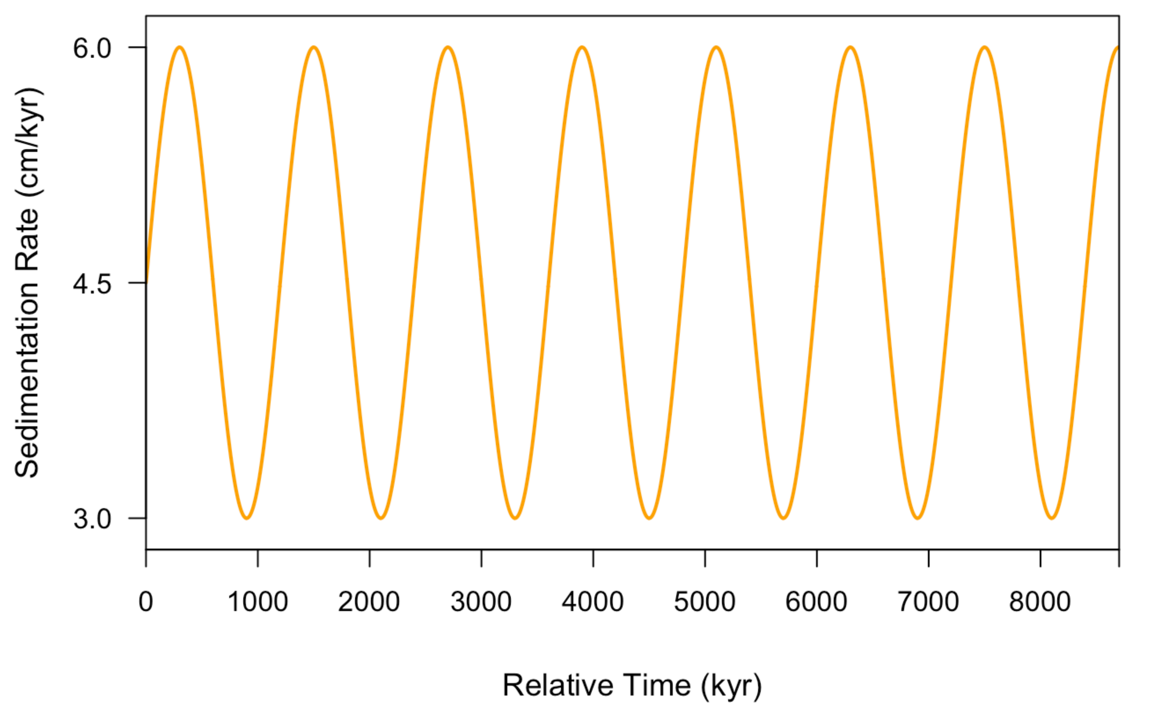
Distorting the time-domain signal to obtain a depth-domain signal

We will use a time-series that is 8.703 Myr in duration. To do that, we need the 2902 first datapoints of the input signal. Then, we define the sedimentation rate evolution, which oscilates between 3 and 6 cm/kyr with a period of 1.2 Myr

```
HadSM3_signal=HadSM3_signal[c(1:2902),]

sedrate=HadSM3_signal
colnames(sedrate)<-c("Relative Time (kyr)","Sed. rate (cm/kyr)")
sedrate[,2]=4.5+1.5*sin(2*pi*sedrate[,1]/1200)
```

```
plot(sedrate,type="l", ylim = c(2.8,6.2), xlim = c(0,8703), xaxs = "i", yaxs = "i", xaxt = "n", yaxt = "n", xlab
= "", ylab = "", col = "orange", lwd = 2)
axis(1, at = c(0,1000,2000,3000,4000,5000,6000,7000,8000,8703), las = 1)
mtext("Relative Time (kyr)", side = 1, line = 3.5, cex = 1.1)
axis(2, at = c(3,4.5,6), las = 1)
mtext("Sedimentation Rate (cm/kyr)", side = 2, line = 3, cex = 1.1)
```



Applying the sedimentation rate evolution to convert time to depth

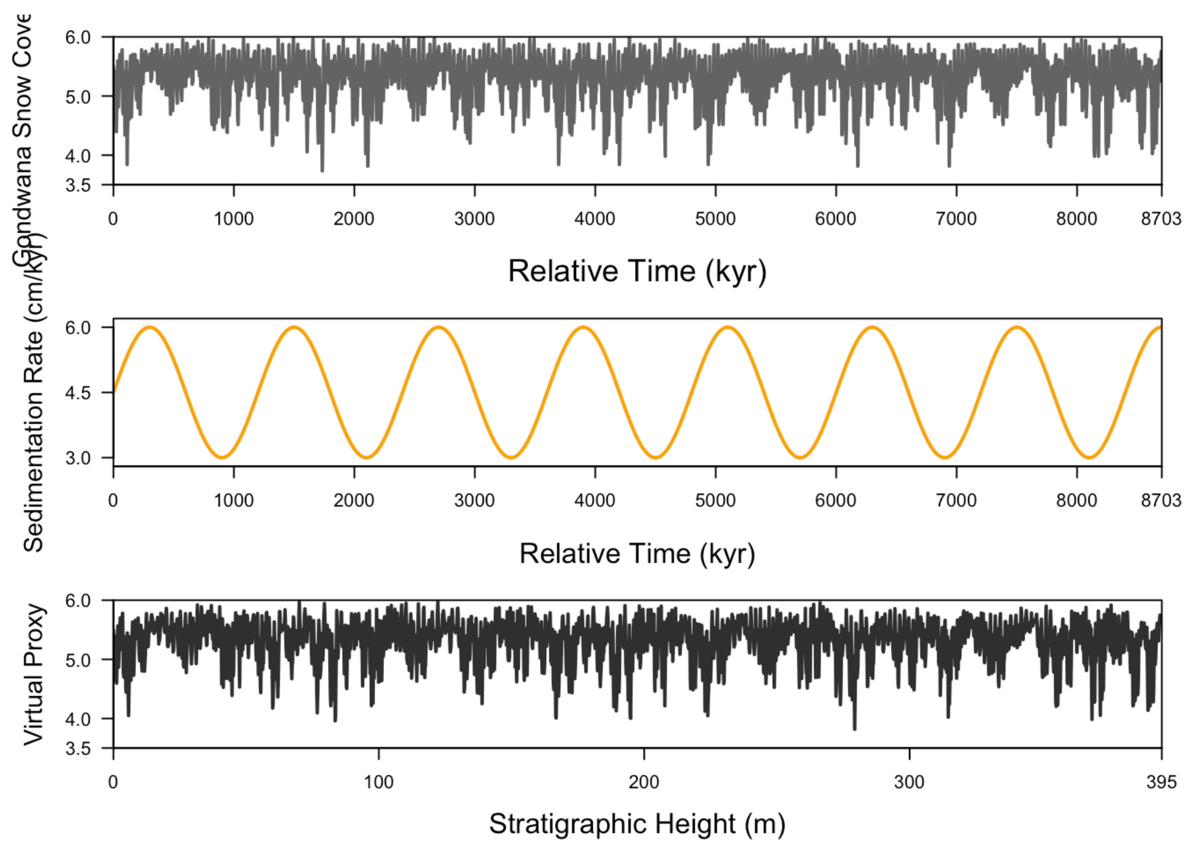
Subsequently, the depth-series "Case_3" is interpolated to an equally-spaced grid of 15 cm

```
Case_3=HadSM3_signal
colnames(Case_3)<-c("Stratigraphic Height (m)","Virtual Proxy")
for (i in 2:length(Case_3[,1])) {
  Case_3[i,1]=Case_3[i-1,1]+((HadSM3_signal[i,1]-HadSM3_signal[i-1,1])*sedrate[i,2]/100)
}
Case_3=linterp(Case_3,dt = 0.15, genplot = F, verbose = F)
```

```
par(mfrow=c(3,1), mar=c(5,5,1,1))
plot(HadSM3_signal,type="l", ylim = c(3.5,6), xlim = c(0,8703), xaxs = "i", yaxs = "i", xaxt = "n", yaxt = "n", x
lab = "", ylab = "", col = "gray40", lwd = 2)
axis(1, at = c(0,1000,2000,3000,4000,5000,6000,7000,8000,8703), las = 1)
mtext("Relative Time (kyr)", side = 1, line = 3.5, cex = 1.1)
axis(2, at = c(3.5,4,5,6), las = 1)
mtext("Gondwana Snow Cover (%)", side = 2, line = 3.5, cex = 0.9)

plot(sedrate,type="l", ylim = c(2.8,6.2), xlim = c(0,8703), xaxs = "i", yaxs = "i", xaxt = "n", yaxt = "n", xlab
= "", ylab = "", col = "orange", lwd = 2)
axis(1, at = c(0,1000,2000,3000,4000,5000,6000,7000,8000,8703), las = 1)
mtext("Relative Time (kyr)", side = 1, line = 3.5, cex = 1)
axis(2, at = c(3,4.5,6), las = 1)
mtext("Sedimentation Rate (cm/kyr)", side = 2, line = 3, cex = 0.9)

plot(Case_3, type="l", ylim = c(3.5,6), xlim = c(0,395), xaxs = "i", yaxs = "i", xaxt = "n", yaxt = "n", xlab = "
", ylab = "", col = "gray20", lwd = 2)
axis(1, at = c(0,100,200,300,395), las = 1)
mtext("Stratigraphic Height (m)", side = 1, line = 3, cex = 1)
axis(2, at = c(3.5,4,5,6), las = 1)
mtext("Virtual Proxy", side = 2, line = 3, cex = 0.9)
```



Adding red and white noise.

Red noise is auto-correlated noise that has high power in the low frequencies. It mimics the memory effect of the climate system. White noise is uncorrelated and completely independent between datapoints. It has high power at the high frequencies. It reflects many different sources of noise, including diagenetic and analytical noise.

```
red_noise=ar1(npts = 2631, dt = 0.15, mean = 0, sdev = 1, rho = 0.999, genplot = F, verbose = F)
white_noise=rnorm(n = 2631, mean = 0, sd = 0.5)
Case_3[,2]=Case_3[,2]+red_noise[,2]+white_noise
```

```
par(mfrow=c(3,1), mar=c(5,5,1,1))
plot(red_noise,type="l", ylim = c(-4,4), xlim = c(0,395), xaxs = "i", yaxs = "i", xaxt = "n", yaxt = "n", xlab =
"", ylab = "", col = "gray40", lwd = 2)
axis(1, at = c(0,100,200,300,395), las = 1)
mtext("Stratigraphic Height (m)", side = 1, line = 3.5, cex = 1.1)
axis(2, at = c(-4,0,4), las = 1)
mtext("Red Noise", side = 2, line = 3.5, cex = 1.1)

plot(Case_3[,1], white_noise, type="l", ylim = c(-4,4), xlim = c(0,395), xaxs = "i", yaxs = "i", xaxt = "n", yaxt
= "n", xlab = "", ylab = "", col = "gray40", lwd = 2)
axis(1, at = c(0,100,200,300,395), las = 1)
mtext("Stratigraphic Height (m)", side = 1, line = 3.5, cex = 1.1)
axis(2, at = c(-4,0,4), las = 1)
mtext("White Noise", side = 2, line = 3.5, cex = 1.1)

plot(Case_3, type="l", ylim = c(-5,15), xlim = c(0,395), xaxs = "i", yaxs = "i", xaxt = "n", yaxt = "n", xlab = "
", ylab = "", col = "gray20", lwd = 2)
axis(1, at = c(0,100,200,300,395), las = 1)
mtext("Stratigraphic Height (m)", side = 1, line = 3.5, cex = 1.1)
axis(2, at = c(0,5,10), las = 1)
mtext("Virtual Proxy", side = 2, line = 3.5, cex = 1.1)
```