

Chronology Building in dplR

Andy Bunn Mikko Korpela

Processed with dplR 1.6.7 in R version 3.4.3 (2017-11-30) on January 29, 2018

Abstract

In this vignette we cover methods for building mean-value chronologies using standardized ring widths.

Contents

1	Introduction	2
1.1	What Is Covered	2
1.2	Citing dplR and R	2
2	Data Sets	3
3	Building Chronologies	5
3.1	Traditional Chronology	5
3.2	Using a Cutoff	6
3.3	Chronology uncertainty	10
3.4	Stripping out series by EPS	11
4	Conclusion	12

1 Introduction

1.1 What Is Covered

The Dendrochronology Program Library in R (dplR) is a package for dendrochronologists to handle data processing and analysis. This document gives an illustration of chronology development in dplR.

1.2 Citing dplR and R

The creation of dplR is an act of love. We enjoy writing this software and helping users. However, neither of us is among the idle rich. Alas. We have jobs and occasionally have to answer to our betters. There is a nifty `citation` function in R that gives you information on how to best cite R and, in many cases, its packages. We ask that you please cite dplR and R appropriately in your work. This way when our department chairs and deans accuse us of being dilettantes we can point to the use of dplR as a partial excuse.

```
> citation()
```

To cite R in publications use:

```
R Core Team (2017). R: A language and environment  
for statistical computing. R Foundation for  
Statistical Computing, Vienna, Austria. URL  
https://www.R-project.org/.
```

A BibTeX entry for LaTeX users is

```
@Manual{,  
  title = {R: A Language and Environment for Statistical Computing},  
  author = {{R Core Team}},  
  organization = {R Foundation for Statistical Computing},  
  address = {Vienna, Austria},  
  year = {2017},  
  url = {https://www.R-project.org/},  
}
```

We have invested a lot of time and effort in creating R, please cite it when using it for data analysis. See also `'citation("pkgname")'` for citing R packages.

```
> citation("dplR")
```

Bunn AG (2008). "A dendrochronology program library in R (dplR)." *_Dendrochronologia_*, *26*(2), pp. 115-124. ISSN 1125-7865, doi: 10.1016/j.dendro.2008.01.002 (URL: <http://doi.org/10.1016/j.dendro.2008.01.002>).

Bunn AG (2010). "Statistical and visual crossdating in R using the dplR library." *_Dendrochronologia_*, *28*(4), pp. 251-258. ISSN 1125-7865, doi: 10.1016/j.dendro.2009.12.001 (URL: <http://doi.org/10.1016/j.dendro.2009.12.001>).

Andy Bunn, Mikko Korpela, Franco Biondi, Filipe Campelo, Pierre Mérian, Fares Qeadan, Christian Zang, Darwin Pucha-Cofrep and Jakob Wernicke (2018). dplR: Dendrochronology Program Library in R. R package version 1.6.7. <https://r-forge.r-project.org/projects/dplr/>

To see these entries in BibTeX format, use 'print(<citation>, bibtex=TRUE)', 'toBibtex(.)', or set 'options(citation.bibtex.max=999)'.

2 Data Sets

Throughout this vignette we will use the onboard data set `wa082` which gives the raw ring widths for Pacific silver fir *Abies amabilis* at Hurricane Ridge in Washington, USA. There are 23 series covering 286 years. The data are plotted in Figure 1. Note the summary stats below.

```
> library(dplR)
> data(wa082)
> wa082.sum <- summary(wa082)
> mean(wa082.sum$year)
```

```
[1] 197.2609
```

```
> mean(wa082.sum$stdev)
```

```
[1] 0.410913
```

```
> mean(wa082.sum$median)
```

```
[1] 0.9756522
```

```
> mean(wa082.sum$ar1)
```

```
[1] 0.7136957  
  
> mean(interseries.cor(wa082)[, 1])  
  
[1] 0.5637096  
  
> plot(wa082, plot.type="spag")
```

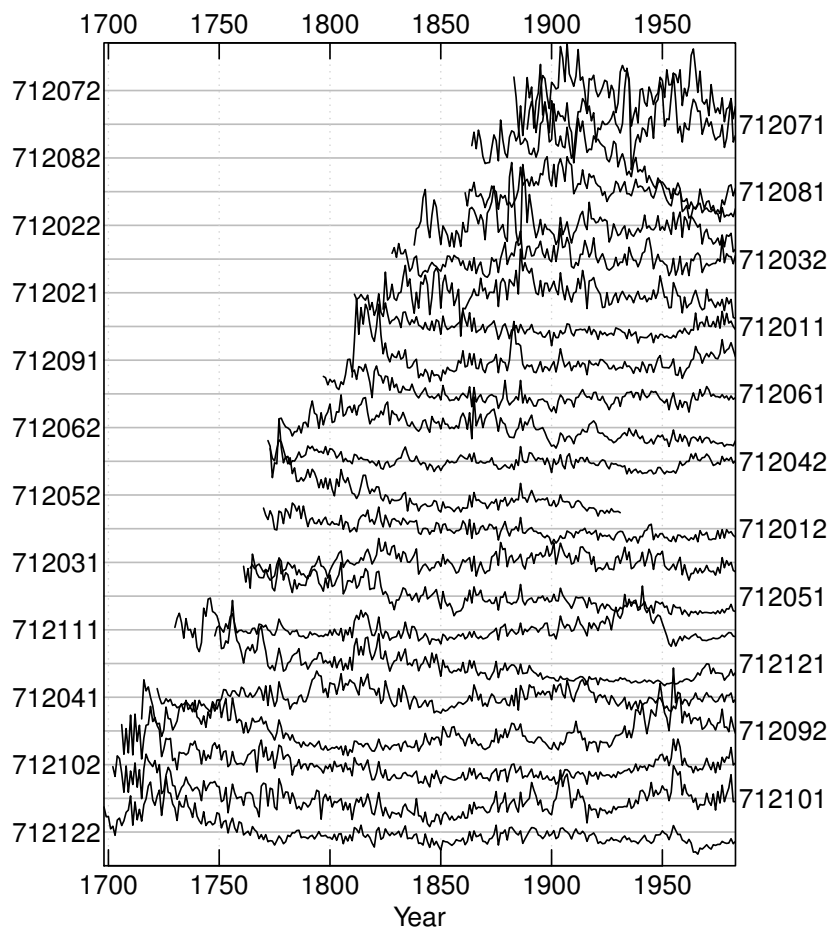


Figure 1: A spaghetti plot of the Pacific silver fir ring widths.

By the way, if this is all new to you – you should proceed immediately to a good primer on dendrochronology like Fritts (2001). This vignette is not intended to teach you about how to do tree-ring analysis. It is intended to teach you how to use the package.

3 Building Chronologies

Let us make a few chronologies from the `wa082` data after detrending each series with a spline that has a frequency response of 50% at a wavelength of 2/3 of each series's length. Detrending is an enormously complicated area unto itself. We use a spline here but could standardize by several other methods and likely get results that look quite different. There is a reason, after all, that dendrochronologists have been arguing about detrending for decades.

```
> wa082.rwi <- detrend(wa082, method="Spline")
```

3.1 Traditional Chronology

The simplest way to make a chronology in `dplR` is with the `crn` function which also has a `plot` method. This defaults to building a mean-value chronology by averaging the rows of the `rwi` data using Tukey's biweight robust mean (function `tbrm` in `dplR`). The result is plotted in Figure 2 with a 30-year smoothing spline for visualization.

```
> wa082.crn <- chron(wa082.rwi, prefix="HUR")
> tail(wa082.crn)
```

	HURstd	samp.depth
1978	0.8766400	22
1979	1.1043443	22
1980	1.0818157	22
1981	1.2447671	22
1982	0.8961536	22
1983	0.9714484	22

```
> plot(wa082.crn, add.spline=TRUE, nyrs=30)
```

Note that the `chron` function will also compute a residual chronology by “prewhitening” the series before averaging. If the `prewhiten` flag is set to `TRUE`, each series is whitened using `ar` prior to averaging. The residual chronology is thus white noise. Note that the `wa082.std.resid` object has two columns with chronologies as well as the sample depth in a third column. The result is plotted in Figure 3.

```
> wa082.std.resid <- chron(wa082.rwi, prefix="HUR", prewhiten = TRUE)
> tail(wa082.std.resid)
```

	HURstd	HURres	samp.depth
1978	0.8766400	0.8069522	22
1979	1.1043443	1.1348334	22

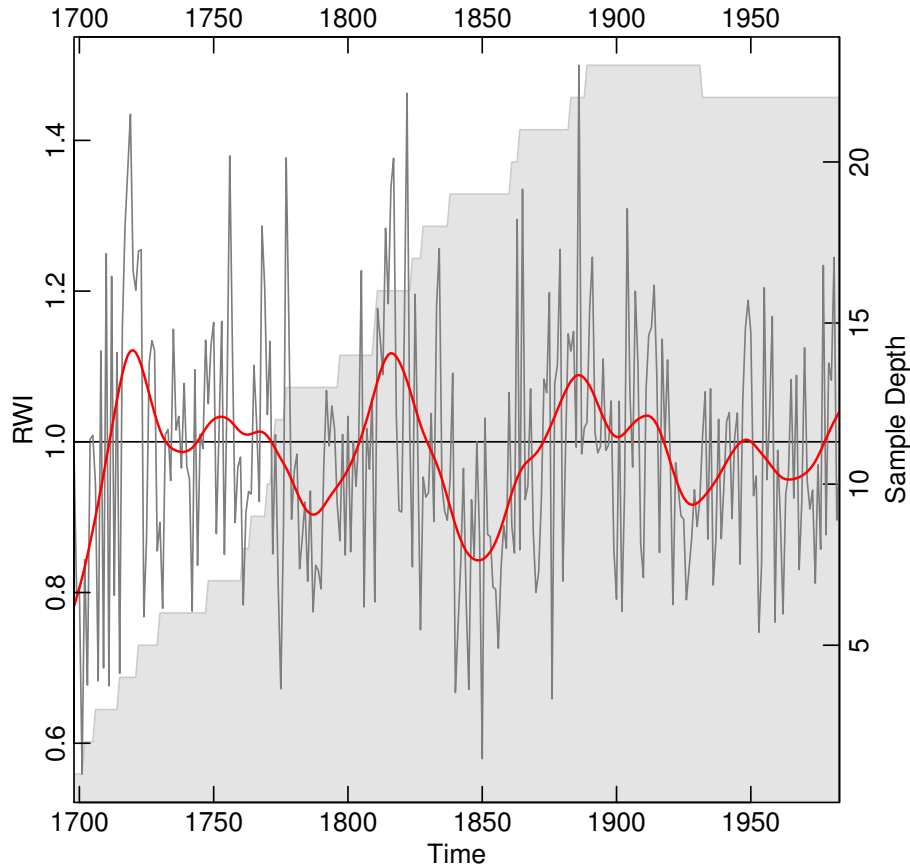


Figure 2: The wa082 chronology.

1980	1.0818157	1.0276662	22
1981	1.2447671	1.2000848	22
1982	0.8961536	0.7757018	22
1983	0.9714484	0.9871302	22

```
> plot(wa082.std.resid, add.spline=TRUE, nyrs=30)
```

3.2 Using a Cutoff

A relatively simple addition to the traditional chronology is to truncate the chronology when the sample depth gets to a certain threshold. The output from the `chron` function contains a column called `samp.depth` which shows the number of series that are average for a particular year. We can use the `subset` function to modify the chronology. The result is plotted in Figure 4.

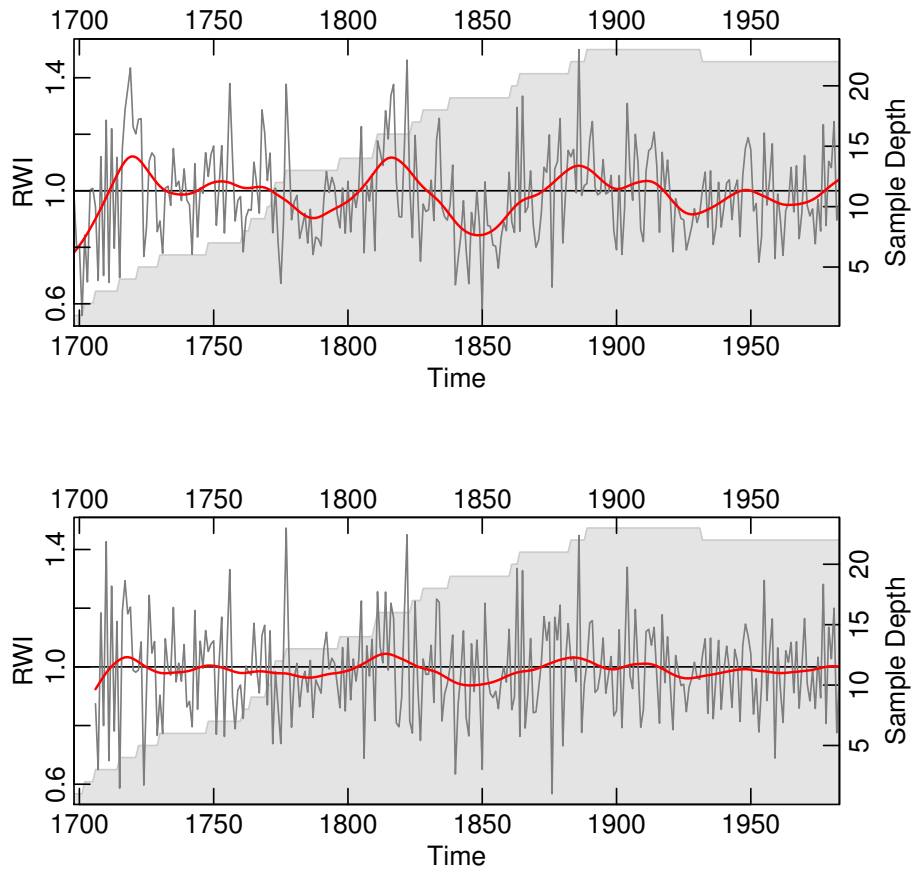


Figure 3: The wa082 chronology as the standard chronology and the residual chronology.

```
> head(wa082.crn)
```

	HURstd	samp.depth
1698	0.9864665	1
1699	0.7960214	1
1700	0.8070109	1
1701	0.5590470	1
1702	0.8436158	2
1703	0.6771966	2

```
> wa082.trunc <- subset(wa082.crn, samp.depth > 5)
> # and plot
> plot(wa082.trunc, add.spline=T, nyrs=30)
```

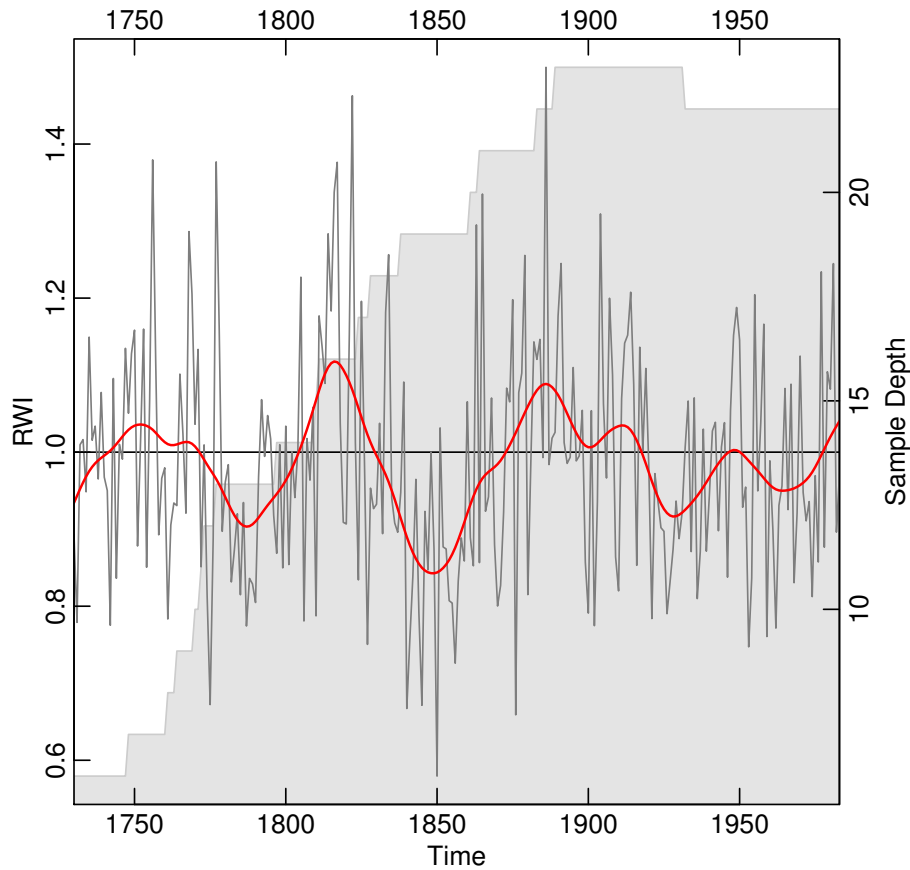


Figure 4: The wa082 chronology truncated by sample depth.

It would likely be more robust to recalculate the `rwi` object by truncating the `rwl` file and then making a chronology which could be done by nesting commands via:

```
> wa082.trunc2 <- chron(detrend(wa082[wa082.crn$samp.depth > 5,],
+                             method="Spline"), prefix="HUR")
```

The result in this case is likely to be virtually identical to truncating after calculating the chronology as in Figure 4.

A more interesting and likely more robust approach is to truncate via the expressed population signal (EPS). The result is plotted in Figure 5.

```
> wa082.ids <- autoread.ids(wa082)
> eps.cut <- 0.75 # An arbitrary EPS cutoff for demonstration
> wa082.rwi.stats <- rwi.stats.running(wa082.rwi, wa082.ids, window.length = 30)
```



```

> yrs <- as.numeric(rownames(wa082.crn))
> bar <- data.frame(yrs = c(min(yrs), wa082.rwi.stats$mid.year, max(yrs)),
+                   eps = c(NA, wa082.rwi.stats$eps, NA))
> op <- par(no.readonly=TRUE)
> par(mar = c(2, 2, 2, 2), mgp = c(1.1, 0.1, 0), tcl = 0.25,
+     mfcol = c(2, 1), xaxs='i')
> plot(yrs, wa082.crn[, 1], type = "n", xlab = "Year",
+      ylab = "RWI", axes=FALSE)
> cutoff <- max(bar$yrs[bar$eps < eps.cut], na.rm = TRUE)
> xx <- c(500, 500, cutoff, cutoff)
> yy <- c(-1, 3, 3, -1)
> polygon(xx, yy, col = "grey80")
> abline(h = 1, lwd = 1.5)
> lines(yrs, wa082.crn[, 1], col = "grey50")
> lines(yrs, ffc_saps(wa082.crn[, 1], nyrs = 30), col = "red", lwd = 2)
> axis(1); axis(2); axis(3);
> par(new = TRUE)
> ## Add EPS
> plot(bar$yrs, bar$eps, type = "b", xlab = "", ylab = "",
+     axes = FALSE, pch = 20, col = "blue")
> axis(4, at = pretty(wa082.rwi.stats$eps))
> mtext("EPS", side = 4, line = 1.1)
> box()
> ## Second plot is the chronology after the cutoff only
> ## Chronology is rebuilt using just years after cutoff but
> ## that difference is essentially nil.
> yr.mask <- yrs > cutoff
> yrs2 <- yrs[yr.mask]
> wa082.rwi2 <- detrend(wa082[yr.mask, ], method="Spline")
> wa082.crn.eps <- chron(wa082.rwi2)
> plot(yrs2, wa082.crn.eps[, 1], type = "n",
+     xlab = "Year", ylab = "RWI", axes=FALSE)
> abline(h = 1, lwd = 1.5)
> lines(yrs2, wa082.crn.eps[, 1], col = "grey50")
> lines(yrs2, ffc_saps(wa082.crn.eps[, 1], nyrs = 30),
+     col = "red", lwd = 2)
> axis(1); axis(2); axis(3); axis(4)
> box()
> par(op)

```

This method calculates a running value for EPS using the `rwi.stats.running` function with a 30-year window. Then we select a cutoff point where we truncate the chronology when the EPS value dips below a certain point. The cutoff method works in this case because the EPS dips as sample depth de-

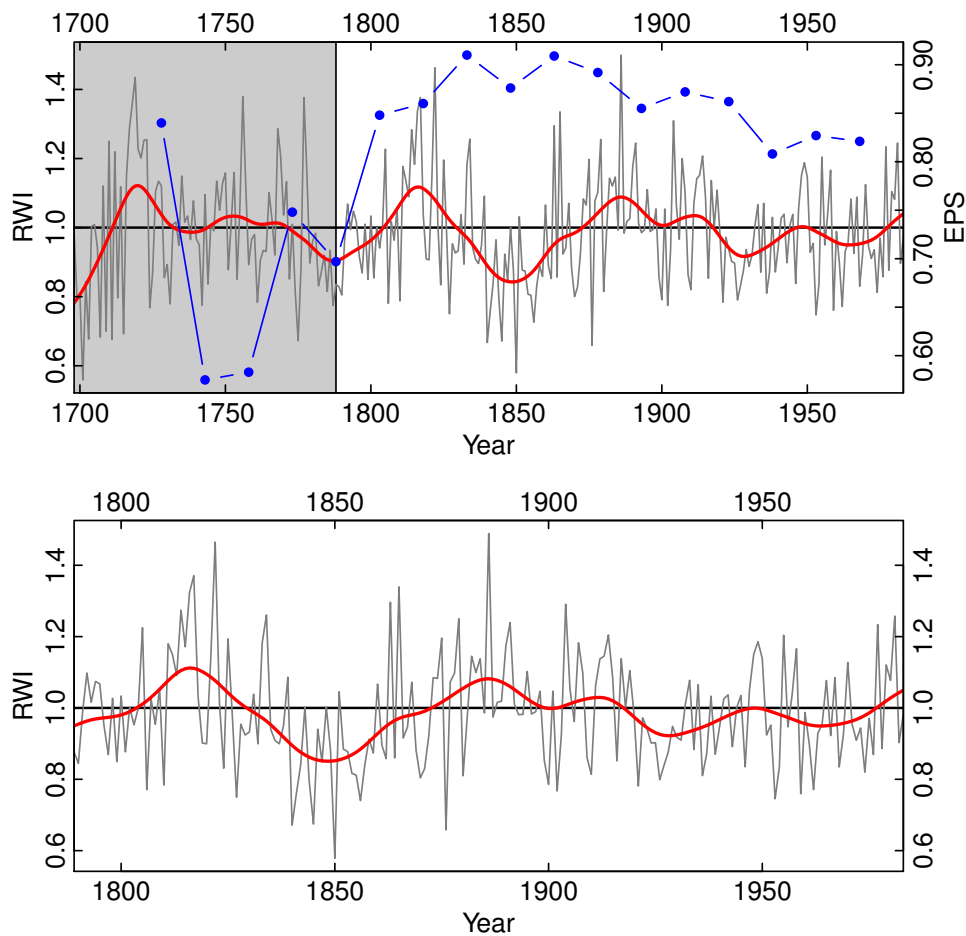


Figure 5: The wa082 chronology truncated by EPS.

clines. However, some feel this is a more objective way of setting a cutoff as opposed to just picking a minimum number of samples.

3.3 Chronology uncertainty

Typically we calculate a chronology by taking the average of each year from the `rwi` object. (And that is typically the biweight robust mean.) The function `chron` like pretty much all the functions in `dp1R` are relatively simple chunks of code that are used for convenience. We can make our own chronology and get the mean plus two standard errors of the yearly growth (Figure 6). Don't get stuck in just using the prepackaged functions in `dp1R`!

```
> wa082.avg <- apply(wa082.rwi2, 1, mean, na.rm=TRUE)
> se <- function(x){
```

```

+ x2 <- na.omit(x)
+ n <- length(x2)
+ sd(x2)/sqrt(n)
+ }
> wa082.se <- apply(wa082.rwi2,1,se)
> wa082.sd <- apply(wa082.rwi2,1,sd,na.rm=TRUE)
> par(mar = c(2, 2, 2, 2), mgp = c(1.1, 0.1, 0), tcl = 0.25, xaxs='i')
> plot(yrs2, wa082.avg, type = "n",ylim=c(0.5,1.6),
+       xlab = "Year", ylab = "RWI", axes=FALSE)
> abline(h = 1, lwd = 1.5)
> xx <- c(yrs2,rev(yrs2))
> yy <- c(wa082.avg+wa082.se*2,rev(wa082.avg-wa082.se*2))
> polygon(xx,yy,col="grey80",border = NA)
> lines(yrs2, wa082.avg, col = "black")
> lines(yrs2, ffcsaps(wa082.avg, nyrs = 30),
+       col = "red", lwd = 2)
> legend(x=1800,y=0.75,legend=c("Mean","2 SE", "30-yr Spline"),
+       lwd=c(1,10,1), col=c("black","grey","red"),
+       bg = "white")
> axis(1); axis(2); axis(3); axis(4)
> box()
> par(op)

```

3.4 Stripping out series by EPS

We want to introduce one other approach that doesn't deal explicitly with chronology building but can be used to build a better chronology. The `strip.rwl` function uses EPS-based chronology stripping Fowler and Boswijk (2003) where each series is assessed to see if its inclusion in the chronology improves the EPS. If it does not the series is dropped from the `rwl` object. As we will see in this example two series are excluded which causes a modest improvement in EPS (Figure 7).

```
> wa082.strip.rwl <- strip.rwl(wa082, ids = wa082.ids)
```

```
REMOVE -- Iteration 1: leaving series 712071 out.
EPS improved from 0.88 to 0.882.
```

```
REMOVE -- Iteration 2: leaving series 712072 out.
EPS improved from 0.882 to 0.883.
```

```
REMOVE -- Iteration 3: no improvement of EPS. Aborting...
REINSERT -- Iteration 1: no improvement of EPS. Aborting...
```

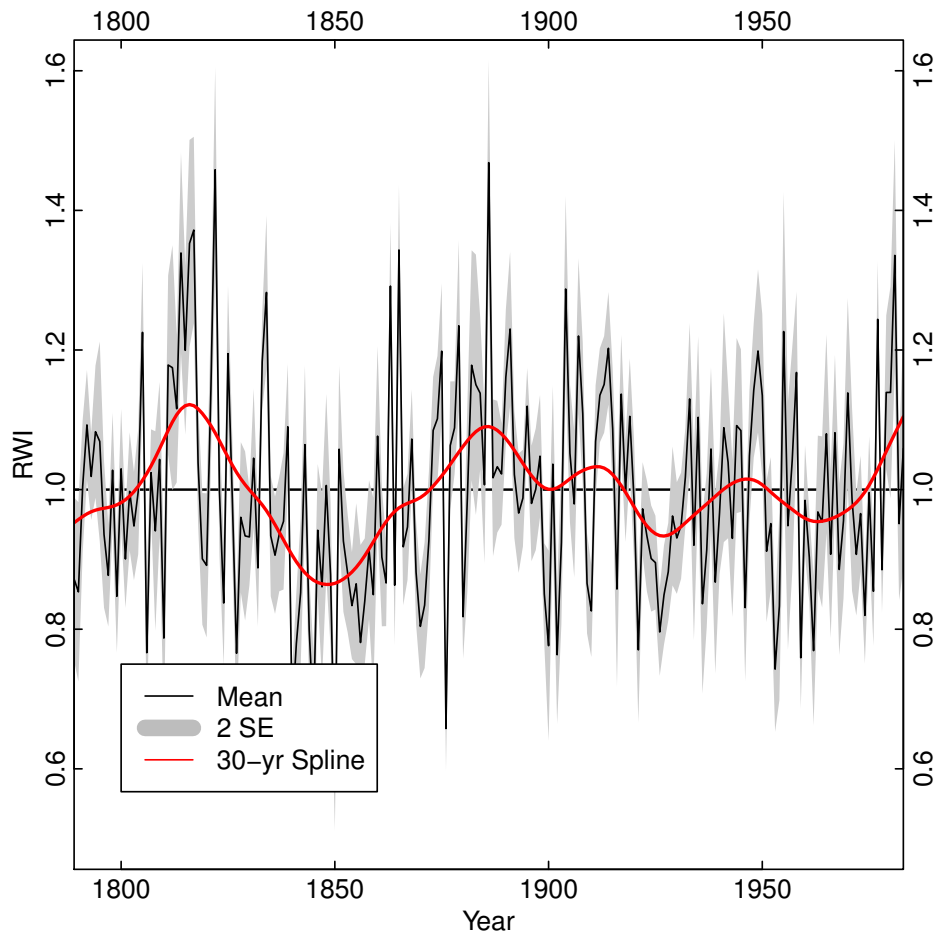


Figure 6: The wa082 chronology with two standard errors.

```

> wa082.rwi.strip <- detrend(wa082.strip.rwl, method="Spline")
> wa082.crn.strip <- chron(wa082.rwi.strip, prefix = "HUR")
> wa082.crn.strip <- subset(wa082.crn.strip, samp.depth > 5)
> plot(wa082.crn.strip, add.spline=TRUE, nyrs=30)

```

4 Conclusion

We have tried to introduce a few ways of building chronologies with `dplR` that are either typical (like truncating by sample depth) or less commonly used. In this vignette we aren't advocating any particular method but trying to get the users familiar with ways of interacting with the objects that `dplR` produces. Once the user understands the data structures the rest of R opens up.

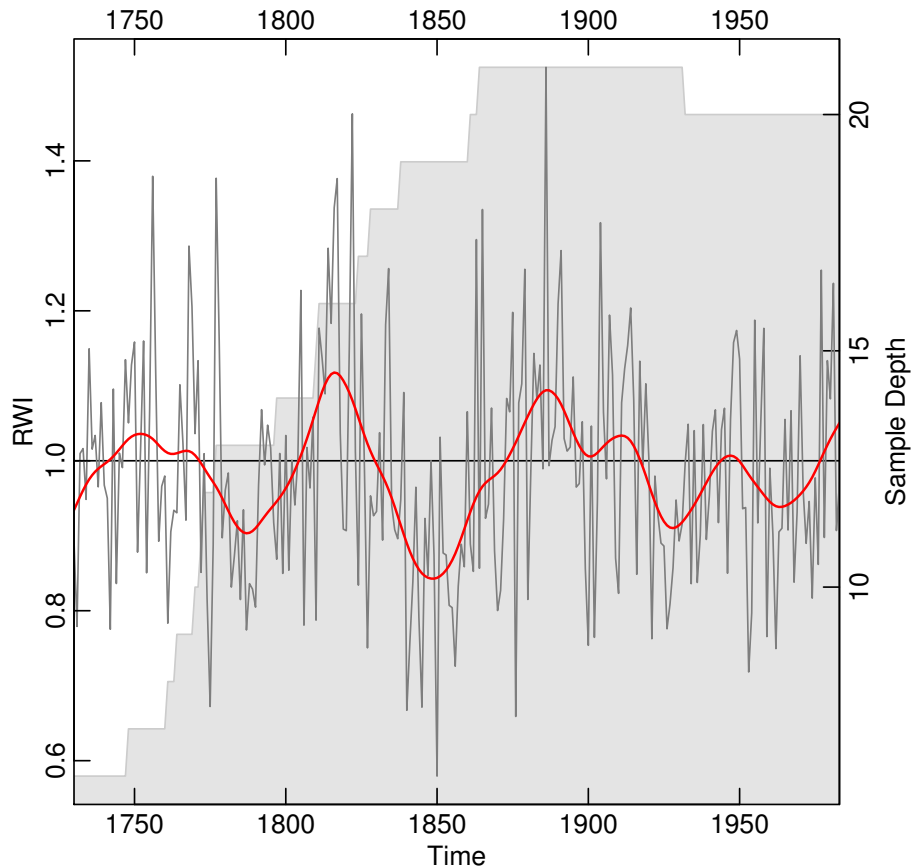


Figure 7: The wa082 chronology after stripping series with low EPS.

Again, we feel that it is important to reiterate that the advantage of using `dp1R` is that it gets the analyst to use `R` and thus have access to the essentially limitless tool that it provides. Go foRth!

References

- Fowler A, Boswijk G (2003). "Chronology stripping as a tool for enhancing the statistical quality of tree-ring chronologies." *Tree-Ring Research*, **59**(2), 53–62. URL <http://arizona.openrepository.com/arizona/handle/10150/262571?mode=full>.
- Fritts HC (2001). *Tree Rings and Climate*. The Blackburn Press. ISBN 1930665393. URL <http://www.amazon.com/Tree-Rings-Climate-H-Fritts/dp/1930665393>.