

Crossdating in dplR

Andy Bunn Mikko Korpela

Processed with dplR 1.7.1 in R version 3.6.2 (2019-12-12) on March 18, 2020

Abstract

In this vignette we cover basic crossdating techniques in dplR by deliberately misdating one of the series in a well-dated set of ring widths and tracking down the dating error. As with any dating enterprise, statistical crossdating is merely a tool and users should always rely on the wood to accurately date tree-ring data.

Contents

1 Introduction	2
1.1 What Is Covered	2
1.2 Citing dplR and R	2
2 Ruining a Perfectly Good Data Set	3
3 Crossdating	5

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 introduction of some of the crossdating functions in dplR. This vignette is essentially a rehashing of [Bunn \(2010\)](#). Please cite that paper if you use dplR for crossdating. There is more detailed information on all these functions in the help files.

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 (2019). 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 = {2019},
  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), 115-124. ISSN 1125-7865, doi: 10.1016/j.dendro.2008.01.002 (URL: <https://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), 251-258. ISSN 1125-7865, doi: 10.1016/j.dendro.2009.12.001 (URL: <https://doi.org/10.1016/j.dendro.2009.12.001>).

Andy Bunn, Mikko Korpela, Franco Biondi, Filipe Campelo, Pierre Mérian, Fares Qeadan and Christian Zang (2020). dplR: Dendrochronology Program Library in R. R package version 1.7.1.
<https://github.com/AndyBunn/dplR>

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

2 Ruining a Perfectly Good Data Set

Throughout this vignette we will use the onboard data set `co021` which gives the raw ring widths for Douglas fir *Pseudotsuga menziesii* at Mesa Verde in Colorado, USA. There are 35 series spanning 788 years.

We will rename the `co021` object to `dat` because we are going to mess around with it and it seems like good practice to rename it. It is a beautifully sensitive series with long segment lengths, high standard deviation (relative to ring widths), large first-order autocorrelation, and a high mean interseries correlation ($r \approx 0.84$). The data are plotted in Figure [1](#).

```
> library(dplR)
> data(co021)
> dat <- co021
> dat.sum <- summary(dat)
> mean(dat.sum$year)
```

```
[1] 564.9143
```

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

```
[1] 0.3231714
```

```

> mean(dat.sum$median)

[1] 0.3211429

> mean(dat.sum$ar1)

[1] 0.6038

> mean(interseries.cor(dat)[, 1])

[1] 0.8477981

> plot(dat, plot.type="spag")

```

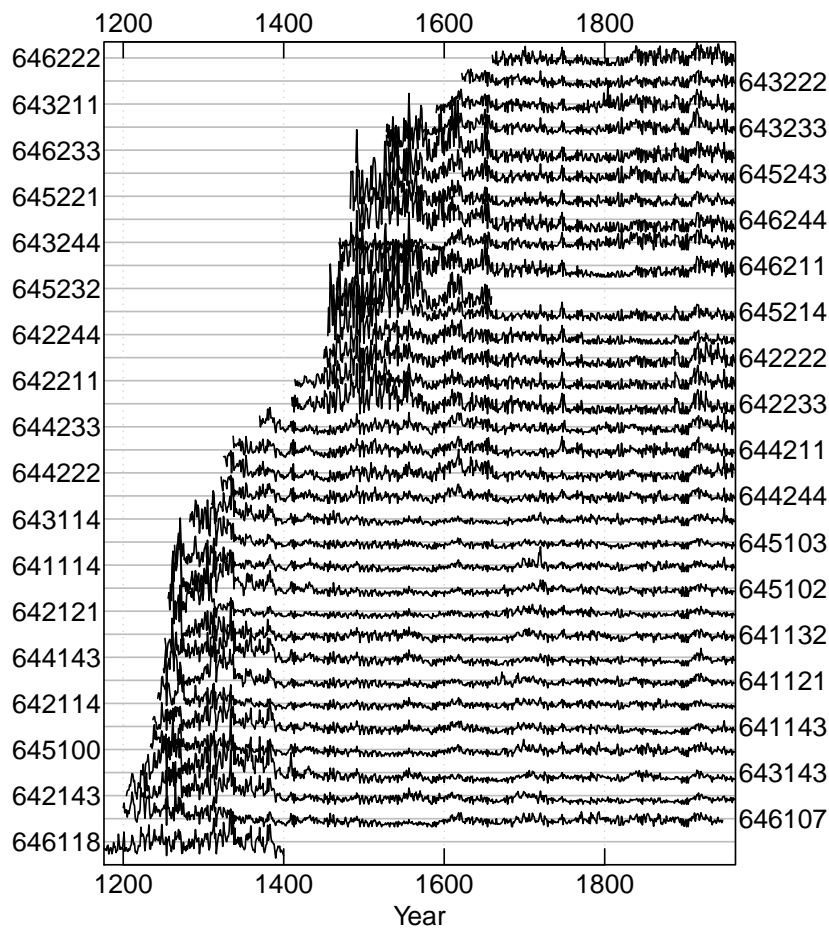


Figure 1: A spaghetti plot of the Mesa Verde ring widths.

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.

To demonstrate how crossdating works in `dplR`, we will take this perfectly lovely data set and corrupt the dating of one of the series. By doing so we will be able to reenact one of the most common tasks of the dendrochronologist: tracking down a misdated core. Here we will take a random series and remove one of the years of growth. This simulates a missing ring in the series. We will pick a random year in the core to give us a bit of a challenge in finding it.

```
> ## create a missing ring by deleting a random year of
> ## growth in a random series
> RNGversion("2.15.0")
> set.seed(4576)
> i <- sample(x=nrow(dat), size=1)
> j <- sample(x=ncol(dat), size=1)
> tmp <- dat[, j]
> tmp <- c(NA, tmp[-i])
> dat[, j] <- tmp
```

We have now deleted the i^{th} observation from the j^{th} core while making sure that `dat` still has the appropriate numbers of rows. By sticking the NA at the start of the series it is as if we missed a ring while measuring.

3 Crossdating

The primary function for looking the crossdating of a tree-ring data set in `dplR` is `corr.rwl.seg`. This function looks at the correlation between each tree-ring series and a master chronology built from all the other series in the `rwl` object (leave-one-out principle). These correlations are calculated on overlapping segments (e.g., 50-year segments would be overlapped by 25 years). By default, each of the series is filtered to remove low-frequency variation prior to the correlation analysis. The help file has abundant details. Here we will look at overlapping 60 year segments. A plot is produced by default with `corr.rwl.seg` (Figure 2). In the `corr.rwl.seg` plots each segment of each series is shown and colored by its correlation with the master. Each series is represented by two courses of lines with the bottom course adhering to the bottom axis timeline and the top course matching the upper axis timeline. Segments are colored according to the strength of the correlation between that segment and the master chronology. Blue correlates well (p-values less or equal to the user-set critical value) while potential dating problems are indicated by the red segments (p-values greater than the user-set critical value). Green lines show segments that do not completely overlap

the time period and thus have no correlations calculated. Our modified data set indicates one series with dating problems.

```
> rwl.60 <- corr.rwl.seg(dat, seg.length=60, pcrit=0.01)
```

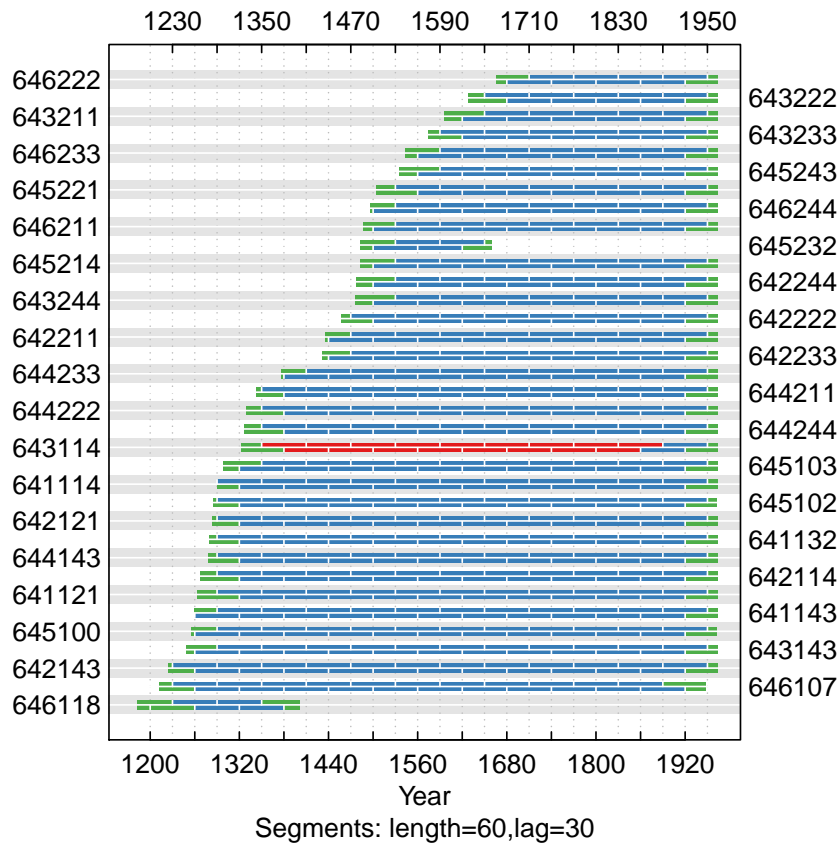


Figure 2: Each 60-year segment of each series in the modified Mesa Verde data set is shown and colored by its correlation with the master. Our modified data set indicates one series with dating problems.

The low correlation between series 643114 and the master indicates a dating problem (Figure 2). Now that we suspect a dating problem, let us take a closer look at this problem child. Figure 3 shows that series 643114 begins to lose correlation with the master at the end of the 19th century.

```
> ## look at this series with a running correlation
> seg.60 <- corr.series.seg(rwl=dat, series="643114",
+                           seg.length=60)
```

This figure strongly indicates that the dating in the series 643114 begins to deteriorate between 1850 and 1910. We can create a window of years to look more closely at this time period and compute a cross-correlation function to look at lagged correlations to see if we can spot the dating problem (Figure 4).

```
> win <- 1800:1960
> dat.yrs <- time(dat)
> dat.trunc <- dat[dat.yrs %in% win, ]
> ccf.30 <- ccf.series.rwl(rwl=dat.trunc, series="643114",
+                          seg.length=30, bin.floor=50)
```

NB: With `series.x = FALSE` (default), negative lags indicate missing rings in series

Figure 4 shows that 1865 to 1894 is the misdated part of this series. The lag of -1 over a lag of 0 indicates that the series 643114 is missing a ring as it better correlates to the master chronology with a one-year offset.¹

Using a smaller time window and shorter correlation segments we can try to further isolate the switch from correlation at lag 0 to lag -1. We will, of course, have to be very careful about using such short segments for correlation and be ready to adjust our expectations accordingly. Fortunately, in this case the trees are so exquisitely sensitive that we can look at 20-year segments with some confidence as in Figure 5.

```
> win <- 1850:1900
> dat.trunc <- dat[dat.yrs %in% win, ]
> ccf.20 <- ccf.series.rwl(rwl=dat.trunc, series="643114",
+                          seg.length=20, bin.floor=0)
```

NB: With `series.x = FALSE` (default), negative lags indicate missing rings in series

By 1879 the correlation between series 643114 and the master is solidly at lag -1 (Figure 5). The 1870 to 1889 correlation is marginal while the dating at 1880–1899 seems accurate (lag 0). This suggests that the dating error is between 1879 and 1889.

¹As of `dplR` version 1.7.0, the cross correlations in `ccf.series.rwl` can be calculated calling `ccf(x=series, y=master, lag.max=lag.max, plot=FALSE)` if `series.x = TRUE`. Otherwise, the function is called `ccf(x=master, y=series, lag.max=lag.max, plot=FALSE)`. Note that prior to `dplR` version 1.60 the behavior of the function was to have `series.x = FALSE` so that `master` was always set as `x` and the `series` as `y`. From versions 1.6.0 to 1.6.9 the `x` and `y` arguments were reversed to be more in line with (some) user expectations that a missing ring in a series would produce a positive lag in the plot rather than a negative lag. This structure of calling the function this way does put the plots at odds with Figure 3 in [Bunn \(2010\)](#) and with the text output from COFECHA. Thus, in true splitting the baby fashion, the argument `series.x` was introduced in version 1.7.0 making sure that everybody will be confused.

We have strong inference now that series 643114 is misdated somewhere in a ten year period around 1885. One final tool that dplR has is the ability to combine the visual style of crossdating using skeleton plots with the statistical approach of cross-correlation analysis. The `skel.ccf.plot` function does just this. Here we make a skeleton plot from the 40-year period around the suspected dating error (1885):

```
> xskel.ccf.plot(rwl=dat, series="643114",
+               win.start=1865, win.width=40)
```

Figure 6 confirms what we suspected. At this point the analyst would go to the wood and take a good look at the core and see what they could find out. There are more heroic efforts that one could go to to figure out exactly where the dating problem might be but nothing ever takes the place of looking at the sample!

We have strong inference now that series 643114 is misdated somewhere in a ten year period around 1885. We have still not revealed whether this is correct or not. Let us look at the values for `i` and `j` and see how we did:

```
> j
[1] 12
> colnames(co021)[j]
[1] "643114"
> i
[1] 709
> rownames(co021)[i]
[1] "1884"
```

Our sleuthing indicated that our dating error was around the year 1885. In fact, `i` was the year 1884. This is pretty spectacular!

References

- Bunn AG (2010). "Statistical and visual crossdating in R using the dplR library." *Dendrochronologia*, **28**(4), 251–258. ISSN 11257865. doi: 10.1016/j.dendro.2009.12.001. URL <http://linkinghub.elsevier.com/retrieve/pii/S1125786510000172>.
- Fritts HC (2001). *Tree Rings and Climate*. The Blackburn Press. ISBN 1930665393. URL <http://www.amazon.com/Tree-Rings-Climate-H-Fritts/dp/1930665393>.

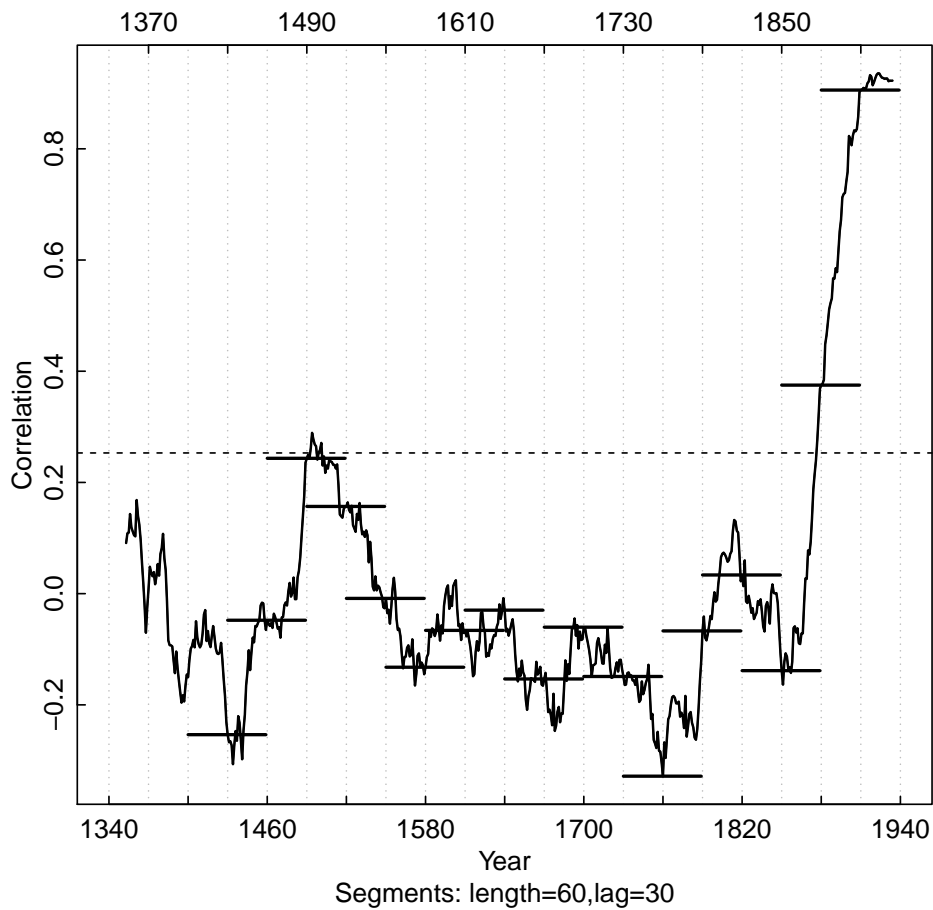


Figure 3: Correlations between series 643114 and the master chronology are shown with horizontal lines according (60-year segments lagged by 30 years). A centered running correlation with a length of 60 years complements the segment correlations. The critical level is shown with a dashed line.

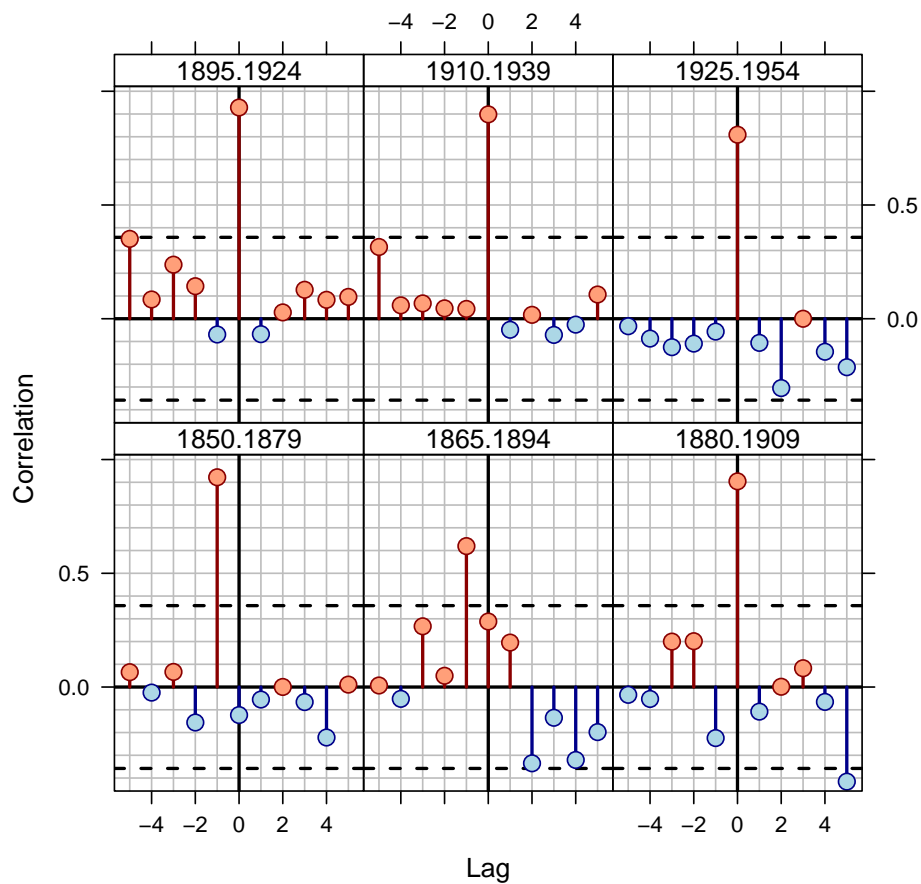
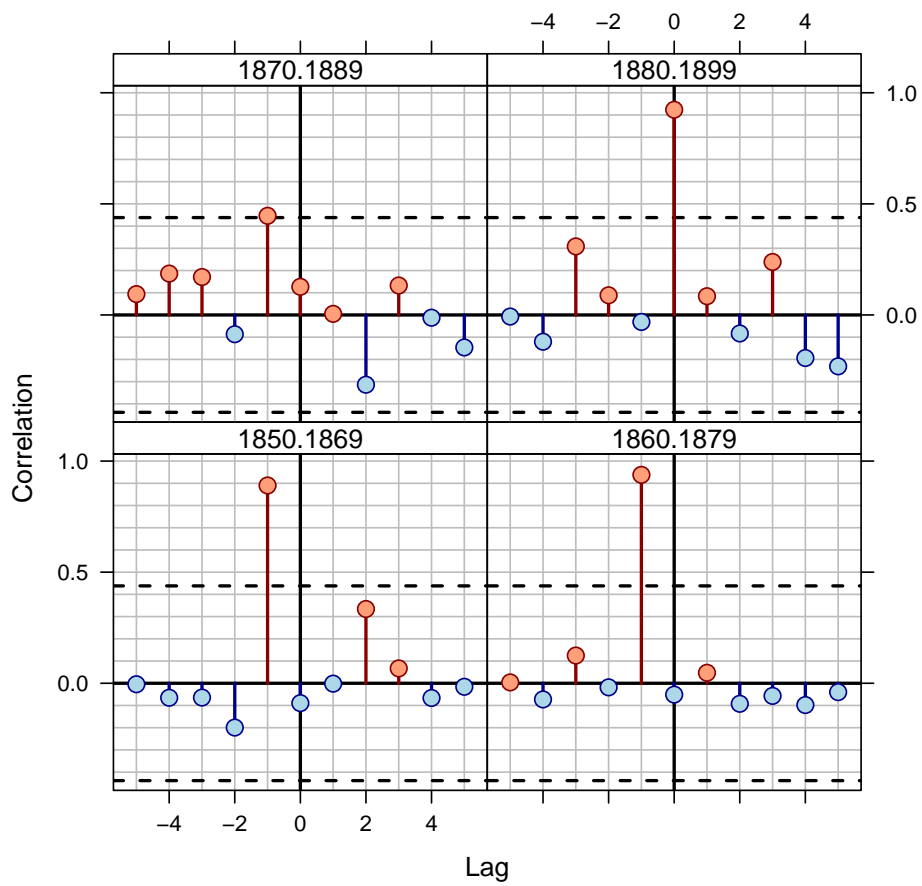
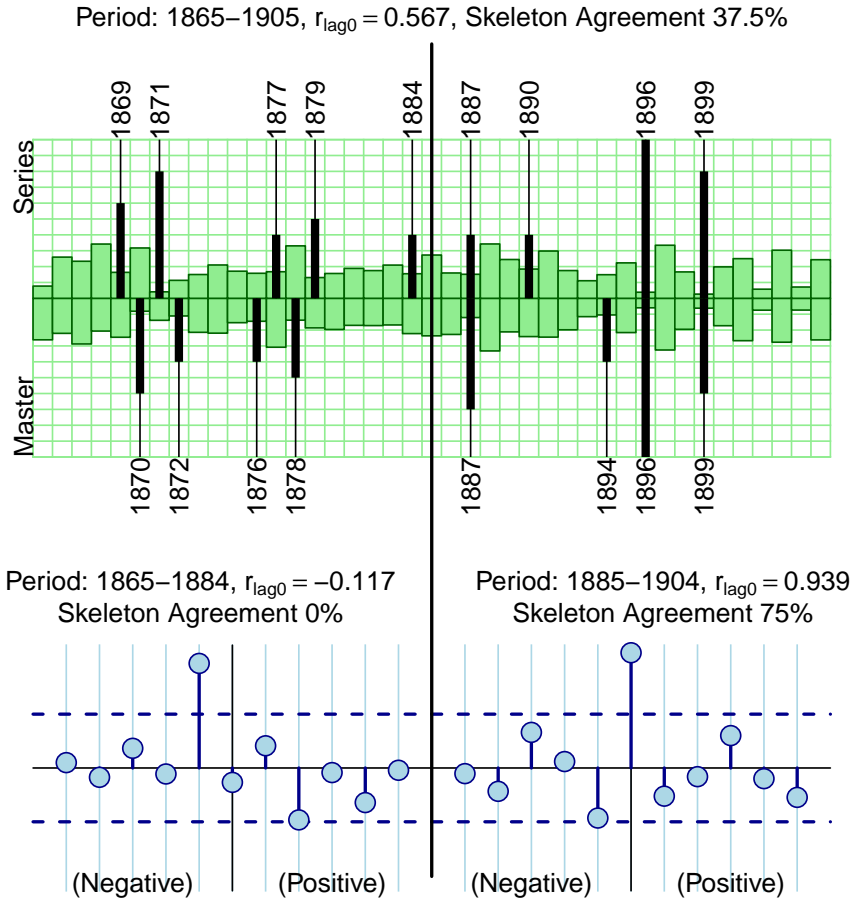


Figure 4: Cross-correlations between series 643114 and the master chronology are shown for each segment (30-year segments lagged by 15 years). The series correlates well at lag 0 until the 1865–1894 bin and then at lag -1 prior to 1865.



NB: With series.x = FALSE (default), negative lags indicate missing rings in series

Figure 5: Cross-correlations between series 643114 and the master chronology at 20-year segments lagged by 10 years over 1850–1900.



NB: With `series.x = FALSE` (default), negative lags indicate missing rings in series

Figure 6: The top panel shows the normalized values for the master chronology (bottom half) and the series 643114 (top half) in green. The values are relative. Similarly, the black lines are a skeleton plot for the master and series with the marker years annotated for the master on the bottom axis and series 643114 on the top. The text at the top of the figure gives the correlation between the series and master (green bars) as well as the percentage of agreement between the skeleton bars for the series and master. The bottom panels show cross correlations for the first half (left) and second half of the time series. In this case, the early period (1865–1884) shows a mismatch of the skeleton plot by one year coupled with a strong lag -1 correlation.