

Chapter 6

Time-Series Analysis



6.1 Taxonomy of Methods

Analysis of epidemic time series is a large endeavor because of the richness of dynamical patterns and plentitude of historical data (Rohani and King 2010). A wide range of tools are used, some of which are borrowed from mainstream statistics other of which are “custom made.” The classic “mainstream” methods belong to two categories: the so-called time-domain and frequency-domain methods. The autocorrelation function and ARIMA models belong to the former class and spectral analysis and the periodogram belong to the latter. Hybrid time/frequency methods have become increasingly prominent in the form of wavelet analysis because it allows the study of changes in disease dynamics through time (Grenfell et al. 2001). This chapter discusses a variety of “mainstream” methods using a variety of time-series data. Examples of “custom made” methods are mechanistic models such as the time-series SIR (TSIR) which is the focus of Chap. 7, semi-parametric models (Ellner et al. 1998) and nonparametric (“empirical dynamic”) models. An example of the latter is discussed in Sect. 10.8.

6.2 Time Domain: ACF and ARMA

The autocorrelation function (ACF) and the autoregressive-moving-average (ARMA) model are classic tools for describing serial dependence in time series in the time-domain. We first apply the ACF to the (weekly) time series of prevalence

This chapter uses the following R-packages: `forecast`, `Rwave`, `imputeTS`, `nlts`, and `plotrix`.

from the seasonally forced SEIR model. The ACF quantifies serial correlations at different time lags. Figure 6.1 shows the ACF for lags up to 3 years (=156 weeks):

```
times = seq(0, 100, by=1/52)
paras = c(mu = 1/50, N = 1, beta0 = 1000, beta1 = 0.2,
          sigma = 365/8, gamma = 365/5)
xstart = c(S=0.06, E=0, I=0.001, R = 0.939)
out = as.data.frame(ode(xstart, times, seirmod2, paras))

par(mfrow=c(1,2))
plot(times, out$I, ylab="Infected", xlab="Time",
      xlim=c(90,100), type="l")
acf(out$I, lag.max = 156, main="")
```

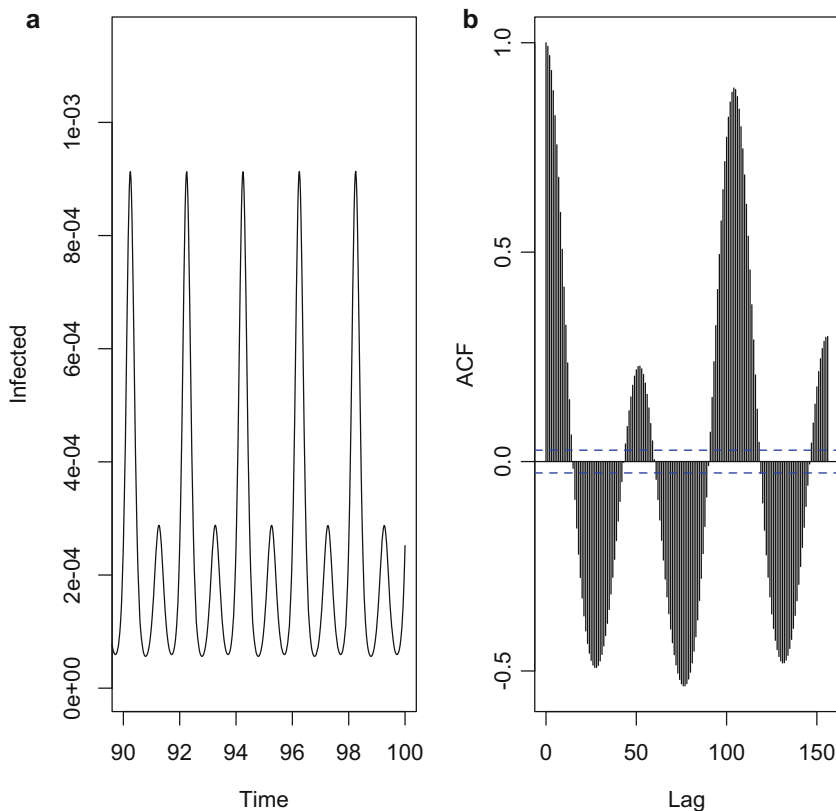


Fig. 6.1 The ACF of prevalence from the seasonally forced SEIR model. (a) time series, (b) ACF

The major peak in autocorrelation at 104 weeks reflects the dominant 2-year periodicity; the minor peak at 52 weeks reflects the subdominant annual periodicity.

6.2.1 ARMA

Autoregressive moving-average models have been used to forecast disease dynamics (e.g., influenza-like illness; Choi and Thacker 1981). The ARMA(p,q)-model assumes that the future incidence (Y_t) can be predicted according to $Y_t = a_1 Y_{t-1} + \dots + a_p Y_{t-p} + \varepsilon_t - b_1 \varepsilon_{t-1} - \dots - b_q \varepsilon_{t-q}$, where the ε 's represent stochasticity and the echo of past stochasticity.¹ We apply the ARMA model to monthly ILI incidence in Iceland using the `forecast`-package:

```
require(forecast)
data(Icelandflu)
```

We convert the data frame to a time-series `ts`-object and do a seasonal decomposition (Fig. 6.2). There is a slight trend through the data, but as expected the winter seasonality is the dominant feature of the time series. Because the epidemics are very peaky we consider square-root transformed numbers:

```
ilits=ts(sqrt(Icelandflu$ili), start=c(1980, 1),
         end=c(2009, 12), frequency=12)
plot(decompose(ilits))
```

We train a seasonal ARMA(2,1) model on for example 1990 through 2000 epidemics and do a 24-month forecast (Fig. 6.3):

```
wts=window(ilits, start=c(1990,6), end=c(2000,5))
fit = arima(sqrt(wts), order=c(2, 0, 1),
           list(order=c(1, 0, 0), period = 12))
coef(fit)

##          ar1          ar2          ma1          sar1
## 1.4460827 -0.7323795 -0.7819940  0.2026528
## intercept
## 2.4823415

fore = predict(fit, n.ahead=24)
```

¹ The ARMA model is usually considered a purely statistical model (i.e., not containing biological mechanism), though it can be shown that the linearized discrete-time SIR model with stochastic transmission can be rewritten as a ARMA(2,1) model (see Sect. 9.8).

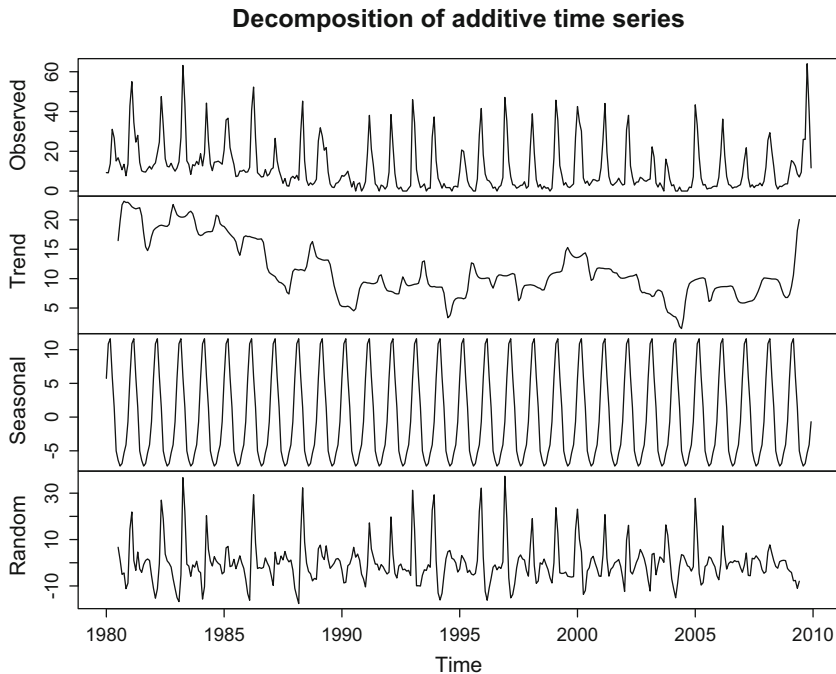


Fig. 6.2 A decomposition of the Iceland ILI time series

```
#Calculate approximate upper (U) and
#lower (L) prediction intervals
U = fore$pred + 2*fore$se
L = fore$pred - 2*fore$se
# plot observed and predicted values
ts.plot(sqrt(wts), fore$pred, U, L, col=c(1, 2, 4, 4),
        lty=c(1, 1, 2, 2), ylab="Sqrt(cases)")
legend("bottomleft", c("ILI", "Forecast",
        "95% Error Bounds"), col=c(1, 2, 4), lty=c(1, 1, 2))
```

While ARMA forecasting is useful in many disciplines and is an important part of the broad statistical toolbox, it suffers from lacking mechanism and can therefore not answer questions like “how are dynamics likely to change if we vaccinate 50% of susceptible children?” It furthermore assumes that time series are *stationary*, essentially meaning that dynamical patterns do not change radically over time. As we frequently see in infectious diseases, this is not a good assumption. In Chap. 7 we discuss how time-series methods that incorporate more biological mechanisms (like the “time-series SIR” model) are better able to capture/predict dynamic transitions.

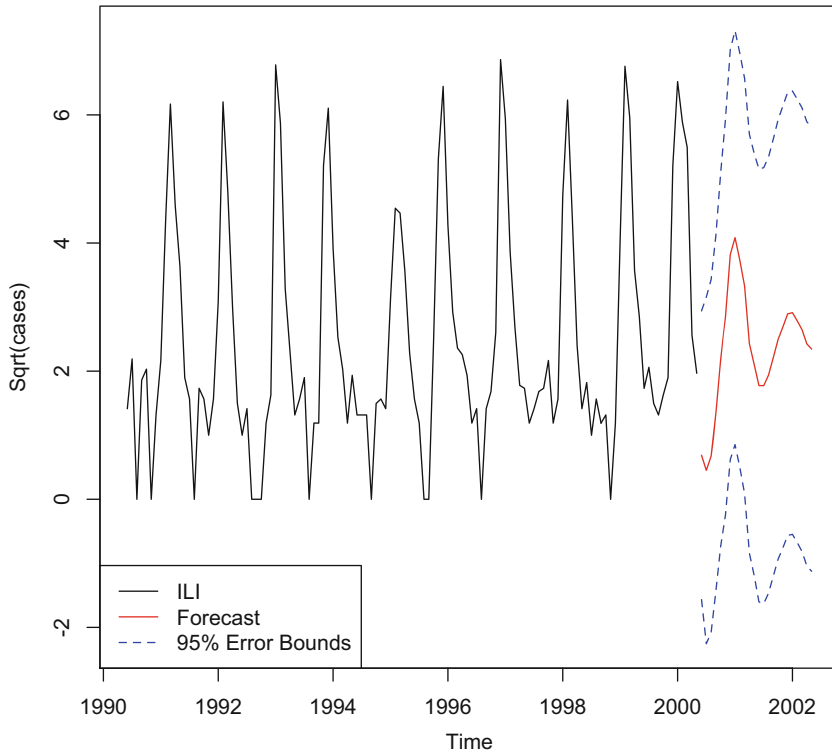


Fig. 6.3 Forecast of square-root transformed ILI incidence in Iceland for the 2001 and 2002 seasons using a seasonal ARMA(2,1) model

6.3 Frequency Domain

The Schuster periodogram is a direct way of estimating and testing for significant periodicity. The periodogram decomposes a time series into cycles of different frequencies (frequency = $1/\text{period}$). The importance of each frequency is measured by the spectral amplitude. We use the spectrum-function to calculate the periodogram for the time series from the seasonally forced SEIR model. The analysis clearly identifies the two superimposed periods (Fig. 6.4).

```
my.spec = spectrum(out$I, plot=FALSE)
par(mfrow=c(1,2))
#default plot (less default lables)
plot(my.spec, xlab="Frequency", ylab="Log-amplitude",
      main="", sub="")
#plot with period (rather than frequency)
plot(1/my.spec$freq/52, my.spec$spec, type="b", xlab=
      "Period (year)", ylab="Amplitude", xlim=c(0,5))
```

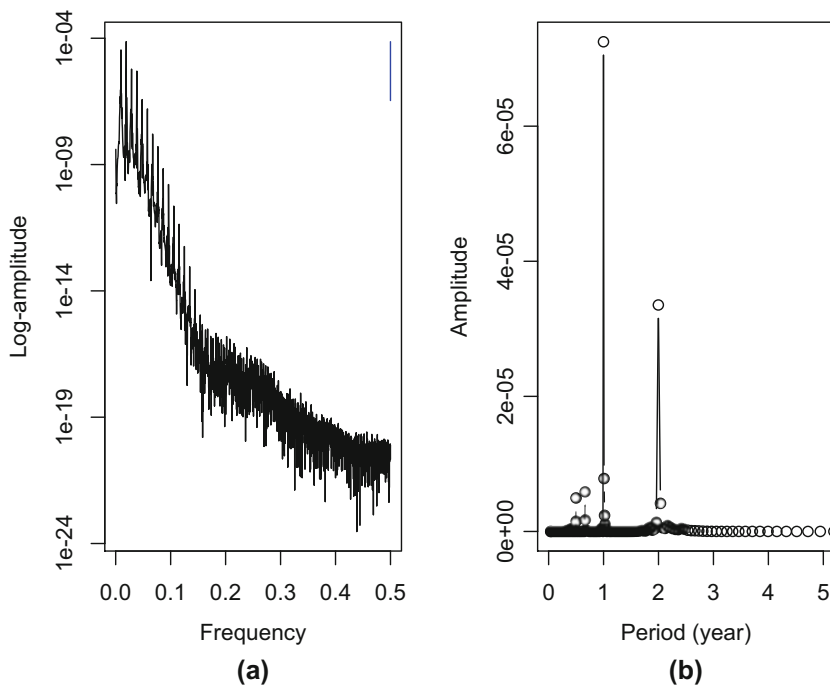


Fig. 6.4 The power spectrum of prevalence for the seasonally forced SEIR model. (a) Default plot of log-amplitude against frequency and (b) amplitude against period (in years)

Using the *fast Fourier transform* (FFT), the Schuster periodogram will automatically estimate the spectrum of a time series (of length T) at the following $T/2$ frequencies: $f = \{\frac{1}{T}, \frac{2}{T}, \dots, \frac{T/2}{T}\}$ (or equivalent periods: $\{T, \frac{T}{2}, \dots, 2\}$). An upside of using FFT is that it is fast. A downside is that the Schuster periodogram is not a *consistent* method, meaning that the estimated periodogram does not converge on the true power spectrum as the time series gets longer because the number of frequencies considered (and thus the number of parameters) increases linearly with time-series length. Numerous fixes of this have been developed, the most common is to smooth the periodogram (Priestley 1981), but nonparametric density estimation has also been proposed. We use Kooperberg et al.'s (1995) log-spline method in Sect. 9.7.

6.4 Wavelets

The wavelet spectrum is an extension of spectral analysis that allows an additional time axis and therefore to allow the study of changes in dynamics over time (Tor-

rence and Compo 1998). Unlike the periodogram, wavelets do not have “canonical” periods for decomposition. If we use the Morlet wavelet (which is provided by the `cwt`-function in the `Rwave`-package), we need to specify the periods we wish to consider through the number of *octaves*, `no`, and *voices*, `nv`. With 8 octaves the main periods will be $\{2^1, 2^2, \dots, 2^8\} = \{2, 4, \dots, 256\}$. The number of voices specifies how many subdivisions to estimate within each octave. With four voices the resultant periods will be $\{2^1, 2^{1.25}, 2^{1.5}, 2^{1.75}, 2^2, 2^{2.25}, \dots\}$. We first consider the simulated time series of prevalence for the unforced SEIR model (Fig. 6.5).

```
#Simulate and plot time series
times = seq(0, 25, by=1/52)
paras = c(mu = 1/50, N = 1, beta = 1000,
          sigma = 365/8, gamma = 365/5)
xstart = c(S=0.06, E=0, I=0.001, R = 0.939)
out2 = as.data.frame(ode(xstart, times, seirmod, paras))
par(mfrow = c(1, 2)) #Side-by-side plots
plot(times, out2$I, type="l", xlab="Time",
      ylab = "Infected")

#Wavelet analysis
require(Rwave)
#Set the number of "octaves" and "voices"
no = 8; nv = 32
#Calculate periods
a = 2^seq(1, no+1-1/nv, by = 1/nv)
#Do the continuous wavelet decomposition
wfit = cwt(out2$I, no, nv, plot=FALSE)
#Calculate the wavelet spectrum
wspec = Mod(wfit)

#Wavelet plot with contours
image(x=times, wspec, col=gray((12:32)/32), y=a/52,
      ylim=c(0,4), xlab="Time", ylab="Period")
contour(x=times, wspec, y=a/52, ylim=c(0,4),
        zlim=c(mean(wspec), max(wspec)), add=TRUE)
```

The initial inter-epidemic period at around 2.5 years is strong (recall that the dampening period of the SEIR with these parameters is 2.3 years with these parameters; Sect. 5.3), but then wanes as the system converges towards the stable endemic equilibrium. We see this clearly illustrated if we compare the wavelet spectrum at, for example, the beginning of year 2 and the beginning of year 10 (Fig. 6.6).

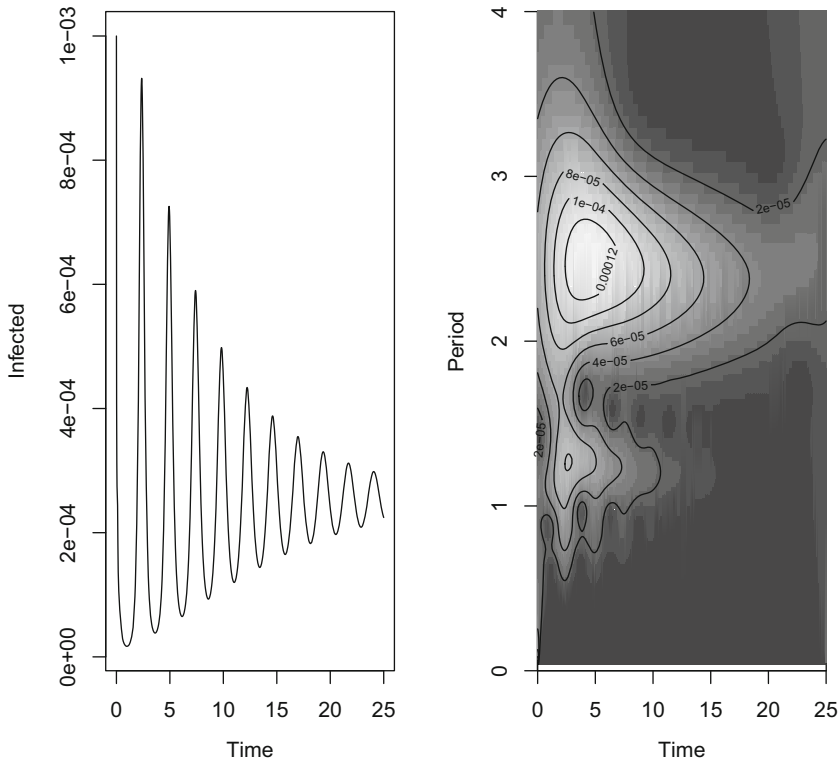


Fig. 6.5 Prevalence against time for the unforced SEIR model ($\mu = 1/50$, $N = 1$, $\beta = 1000$, $\sigma = 365/8$, $\gamma = 365/5$) with associated wavelet spectrum

```
plot(a/52, wspec[104,], type="l", ylab="Amplitude",
     xlab="Period")
lines(a/52, wspec[1040,], type="l",
      lty=2, col="red")
legend("topright", legend=c("Year 2", "Year 10"),
      lty=c(1,2), col=c("black", "red"))
```

6.5 Measles in London

The pre-vaccination incidence of measles shows interesting non-stationarities that have been traced back to changing susceptible recruitment due to the post-World War II baby boom (Fig. 6.7). The meas data set contains the biweekly incidence and births from 1944 and 1965.

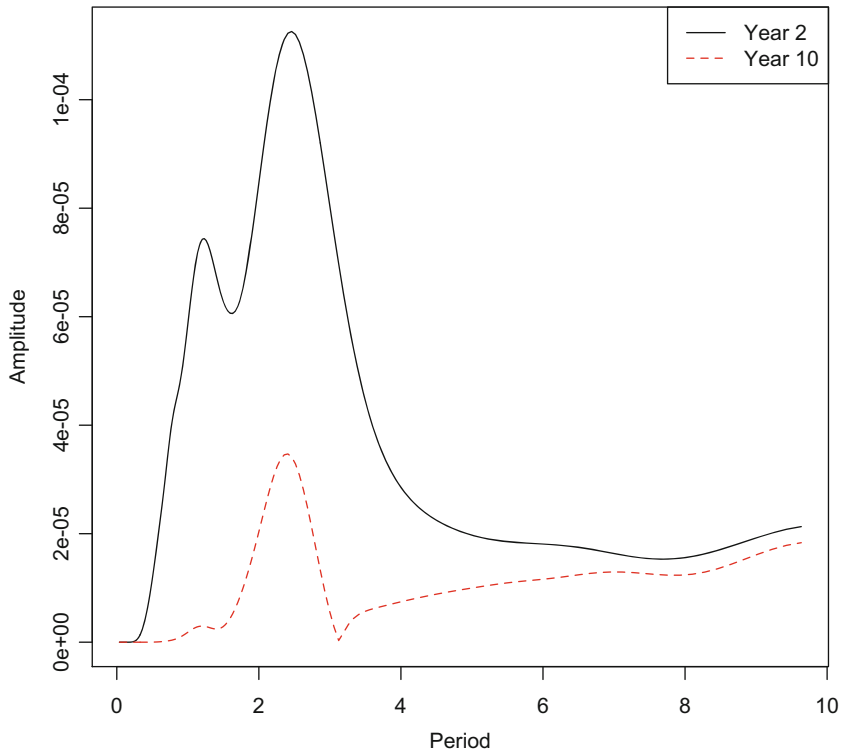


Fig. 6.6 The estimated wavelet spectrum at the first week of year 2 and year 10 for the unforced SEIR model

```

data(meas)
head(meas)

##   year week   time London    B
## 1   44   2 44.00000   180 1725
## 2   44   4 44.03846   271 1725
## 3   44   6 44.07692   423 1725
## 4   44   8 44.11538   465 1725
## 5   44  10 44.15385   523 1725
## 6   44  12 44.19231   649 1725

par(mar = c(5,5,2,5)) #Make room for two axes
plot(meas$time, meas$London, type="b", xlab="Week",
      ylab="Incidence", ylim=c(0,8000))
par(new=T) #Superimposed births plot
plot(meas$time, meas$B, type="l", col="red",
      axes=FALSE, xlab=NA, ylab=NA, ylim=c(1000, 2700))
axis(side = 4)

```

```
mtext(side = 4, line = 3, "Births")
legend("topright", legend=c("Cases", "Births"),
      lty=c(1,1), col=c("black", "red"))
```

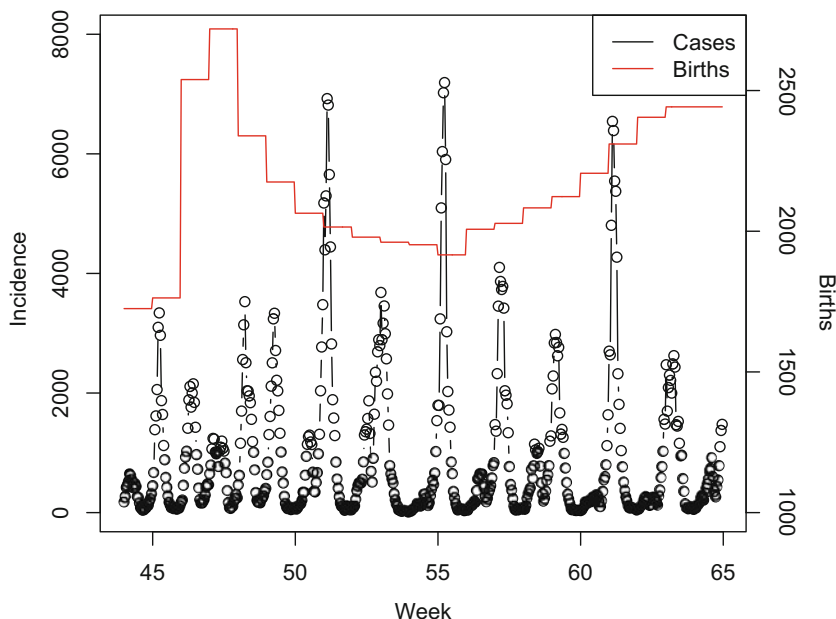


Fig. 6.7 Biweekly incidence of measles in London between 1944 and 1965 with susceptible recruitment (births) superimposed

We apply the wavelet analysis to the historical measles dynamics from London (Grenfell et al. 2001). In addition to providing a continuous wavelet transform, the Rwave-package has a “crazy climber” algorithm to highlight ridges in the wavelet spectrum (implemented with the `crc` and `cfamily`-functions). When applied to the London measles data, the crazy climber reveals the background annual rhythm and the punctuated appearance of the biennial cycle in the early 1950s (Fig. 6.8).

```
#Set octaves, voices and associated periods
no = 8; nv = 32
a = 2^seq(1,no+1-1/nv, by = 1/nv)
#Continous wavelet decomposition
wfit = cwt(meas$London, no, nv, plot=FALSE)
wspec = Mod(wfit)
#Crazy climber
crcinc<-crc(wspec, nbclimb=10, bstep=100)
```

```
frcrcinc<-cfamily(crcinc, ptile=0.5, nbchain=1000,
  bstep=10)

## There are 2 chains.

ridges<-frcrcinc[[1]]
ridges[which(ridges==0)]<-NA
#Wavelet plot with crazy-climber and contours
image(x=meas$time, wspec, col=gray((12:32)/32), y=a/26,
  ylim=c(0.1,3), ylab="Period", xlab="Year")
contour(x=meas$time, wspec, y=a/26, ylim=c(0,3),
  nlevels = 6, zlim=c(mean(wspec), max(wspec)),
  add=TRUE)
image(x=meas$time, y=a/26, z=ridges, add=TRUE,
  col=gray(0))
```

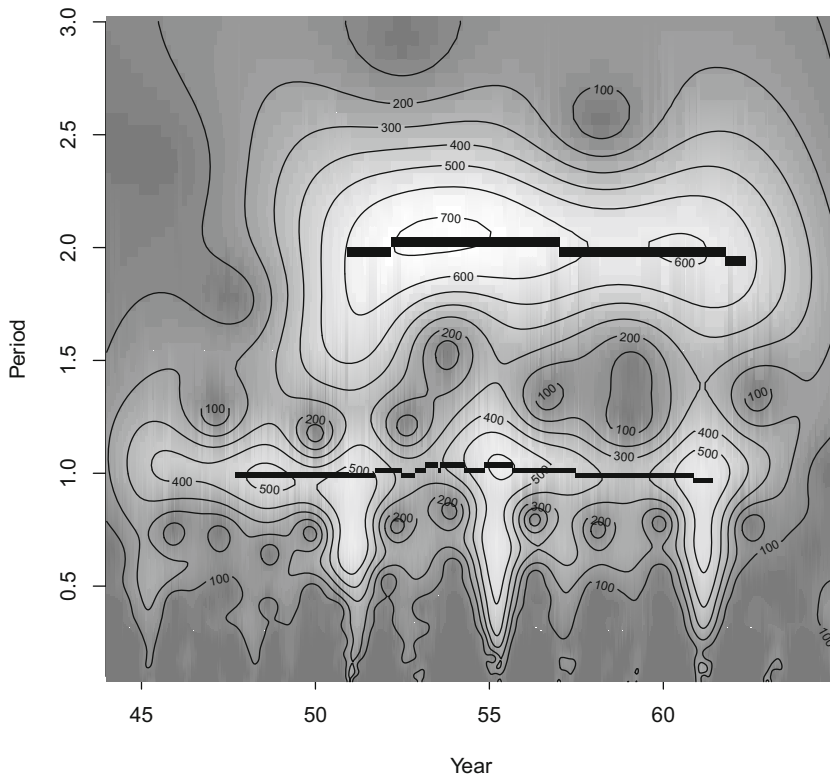


Fig. 6.8 The wavelet spectrum of the London measles incidence with “crazy-climber” ridges. The appearance of a significant biennial rhythm in the 1950s is conspicuous

We can contrast the spectrum of the first biweek of January 1945 and the first biweek of January 1954 (Fig. 6.9). The transition from a dominance of annual to biennial epidemics is conspicuous. Two-year cycles are pronounced when birth rates are around 20 per thousand per year; Annual epidemics are associated with higher birth rates. This transition, due to the post-World War II baby boom, is as predicted by the seasonally forced SEIR model with dropping birth rates (Earn et al. 2000b, Fig. 5.6).

```
plot(a/26,wspec[261,], type="l",xlim=c(0,3),
     xlab="period (years)", ylab="amplitude")
lines(a/26,wspec[27,], type="l", lty=2, col="red")
legend("topleft", legend=c("1945", "1954"),
      lty=c(2,1), col=c("red", "black"))
```

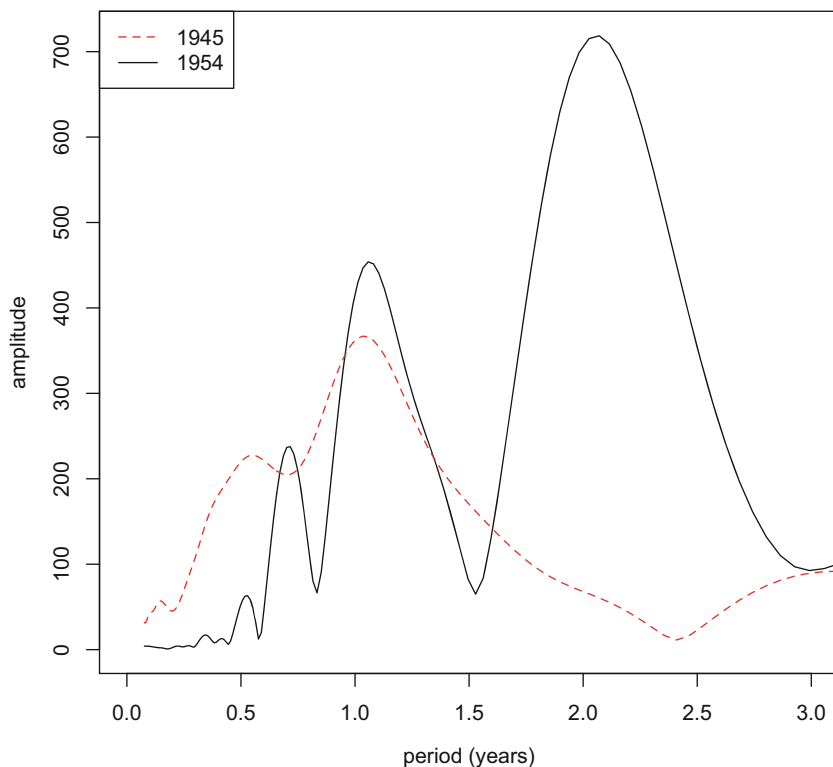


Fig. 6.9 The wavelet spectrum of the London measles in Jan 1945 vs Jan 1954

The above methods of time-series analysis require regularly spaced time series without any missing values. Lomb (1976) developed the Lomb periodogram for unequally spaced data. Furthermore, the classic spectral methods

cannot quantify rhythms in cruder “nonmetric” data such as presence/absence of infection. Legendre et al. (1981) developed the “contingency periodogram” for such situations. The `nltts`-package has the functions `spec.lomb` and `contingency.periodogram` to carry out such analyses. The `mvcbwt`-package can do wavelet analyses of time series with missing data.

6.6 Project Tycho

Project Tycho (<http://www.tycho.pitt.edu>) is a great resource for time series on historical disease incidence. The data used in Sect. 5.1 were downloaded from this database. Weekly data of whooping cough (1925–1947), diphtheria (1914–1947), and measles (1914–1947) in the city of Philadelphia are from Project Tycho and are saved in the `tywhooping`, `tydiphtheria`, and `tymeasles` data sets. These were all important causes of childhood mortality in the early twentieth century and were therefore “reportable infections” in the USA. Whooping cough is caused by bacterial colonization of the lower respiratory tract by congeneric species in the genus *Bordetella*, most notably *B. pertussis*, and cause violent coughing, vomiting, and pneumonia. Diphtheria is caused by infection by *Corynebacterium diphtheriae* which toxins cause a range of health complications. Measles is a severely immunocompromising paramyxovirus. We will use these time series to illustrate some additional aspects of disease dynamics/time-series analysis.

```
data(tywhooping)
tywhooping$TIME=tywhooping$YEAR+tywhooping$WEEK/52
tywhooping$TM=1:length(tywhooping$YEAR)
data(tydiphtheria)
data(tymeasles)
tydiphtheria$TIME=tymeasles$TIME=tymeasles$YEAR+
  tymeasles$WEEK/52
```

These time series have occasional weeks of missing data which we interpolate. We will use the `imputeTS`-package. But first we use the whooping cough data to illustrate the use of the Lomb periodogram for spectral analysis of unevenly spaced data.

6.7 Lomb Periodogram: Whooping Cough

There are 14 missing weeks in the `tywhooping` data set. For frequency-domain analyses of this data we either have to interpolate the missing weeks or use the Lomb periodogram. We compare the two approaches:

```

data(tywhooping)
whp=na.omit(tywhooping)

#data with missing values interpolated
require(imputeTS)
sum(is.na(tywhooping$PHILADELPHIA))

## [1] 14

tywhooping$PHILADELPHIA=
  na.interpolation(ts(tywhooping$PHILADELPHIA))

#Classic periodogram
my.spec = spectrum(sqrt(tywhooping$PHILADELPHIA))
#Lomb periodogram
require(nlts)
my.lomb=spec.lomb(x=whp$TM, y=sqrt(whp$PHILADELPHIA))

plot(1/my.spec$freq/52, my.spec$spec, type="b",
      xlab="Period (year)", ylab="Amplitude")
par(new=TRUE)
plot(1/my.lomb$freq/52, my.lomb$spec, axes=FALSE,
      type="b", col=2, xlab="", ylab="")
legend("topright", legend=c("Classic", "Lomb"),
      lty=c(1,1), pch=c(1,1), col=c("black", "red"))

```

With only 14 missing values in a 1000+ week long time series the shape of the Schuster periodogram (on interpolated data) and the Lomb periodogram are almost identical (Fig. 6.10).

6.8 Triennial Cycles: Philadelphia Measles

Like in London, pre-vaccination measles dynamics in Philadelphia exhibit interesting nonstationarities we can highlight with the wavelet analysis. There are 24 missing weeks we interpolate:

```

data(tymeasles)
sum(is.na(tymeasles$PHILADELPHIA))

## [1] 24

tymeasles$PHILADELPHIA=
  na.interpolation(ts(tymeasles$PHILADELPHIA))

```

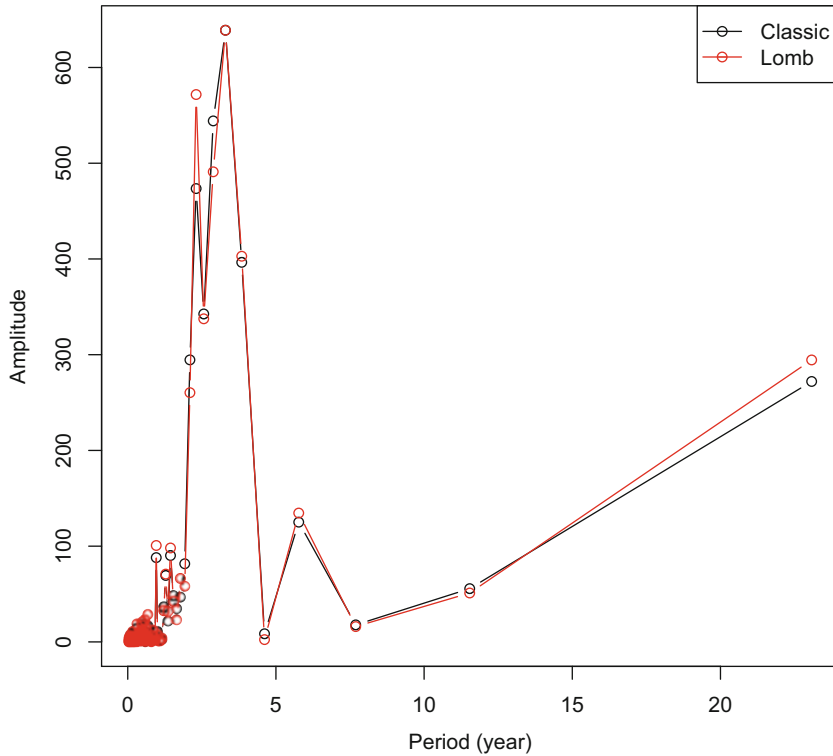


Fig. 6.10 The Lomb periodogram and the classic periodogram (on interpolated data) of the Philadelphia whooping cough time series

We twiddle with the graphics margins and layout using the `par` and `layout` functions to make a prettier compound graphic (Fig. 6.11).

```
par(mfrow=c(2,1), mar=c(2,4,2,1))
layout(matrix(c(1,1,2,2,2), ncol=1))
plot(tyemeasles$TIME, sqrt(tyemeasles$PHILADELPHIA),
     type="b", ylab="Sqrt(incidence)")
title("Measles 1914-47")

no = 8; nv = 16; a = 2^seq(1,no+1-1/nv, by = 1/nv)
wfit = cwt(sqrt(tyemeasles$PHILADELPHIA),
           no, nv, plot=FALSE)
wspec = Mod(wfit)
par(mar=c(1,4,0.25,1))
image(z=wspec, y=a/52, ylim=c(0,4), ylab="Period(year)",
      col=gray((12:32)/32), xaxt='n')
contour(z=wspec, y=a/52, ylim=c(0,4), nlevels = 6,
```

```
zlim=c(mean(wspec), max(wspec)), add=TRUE)
```

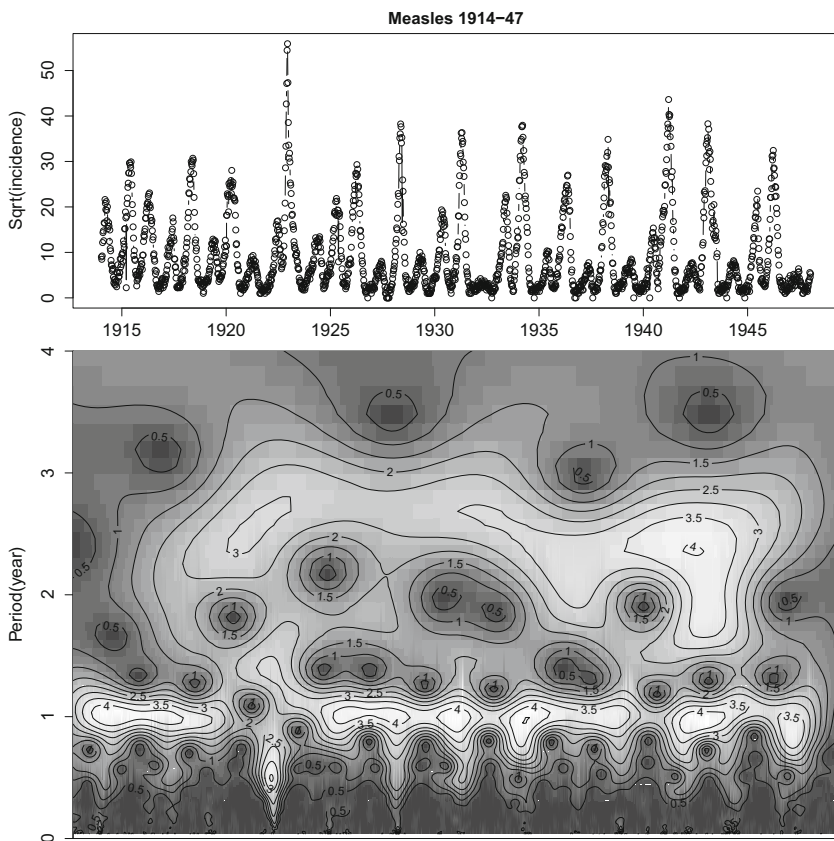


Fig. 6.11 Wavelet spectrum of measles in Philadelphia

The early annual epidemics give way to triennial epidemic cycles from 1920 onwards (Fig. 6.12). The tri-annual cycles are the hallmarks of chaotic epidemics (Dalziel et al. 2016) we discuss further in Sect. 10.2.

```
plot(a/52,wspec[54,], type="l", xlim=c(0,4),
     xlab="Period (years)", ylab="Amplitude",
     col="red", lty=2)
lines(a/52,wspec[1357,], type="l", xlim=c(0,4))
legend("topleft", legend=c("1915", "1940"),
       lty=c(2,1), col=c("red","black"))
```

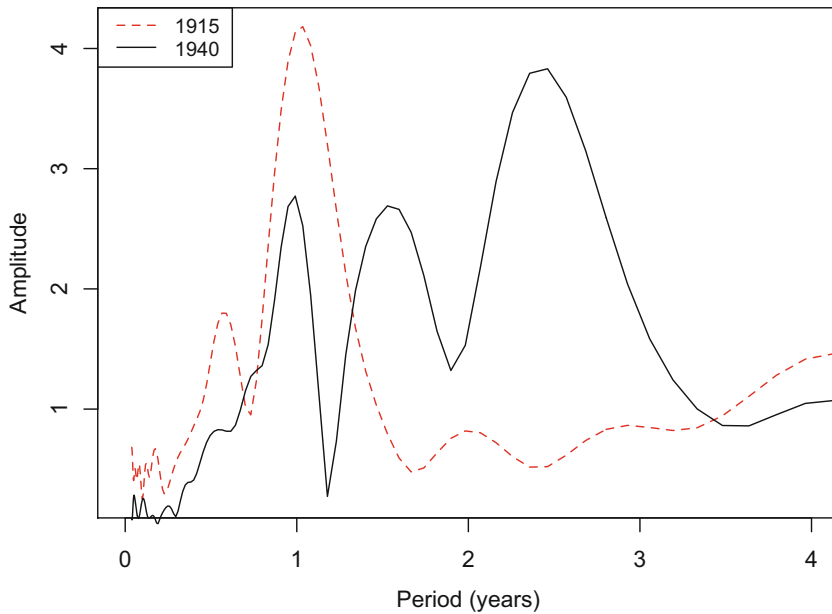



Fig. 6.12 The Jan 1915 versus Jan 1940 measles wavelet spectrum; Annual epidemics give way to triennial cycles

6.9 Wavelet Reconstruction and Wavelet Filter: Diphtheria

Diphtheria exhibited conspicuous annual cycles during the beginning of the twentieth century until the addition of an [adjuvant to the toxoid vaccine in 1926](#) led to a strong secular downward trend and effectively the elimination of the disease (Fig. 6.13). The wavelet lets us study how adjuvant-induced reduction in incidence is associated with a loss of periodicity and increase in high-frequency variability (“noise”) (Fig. 6.13). There are 18 missing values we interpolate prior to the analysis.

```
data(tydiphtheria)
sum(is.na(tydiphtheria$PHILADELPHIA))

## [1] 18

tydiphtheria$PHILADELPHIA=
  na.interpolation(ts(tydiphtheria$PHILADELPHIA))

par(mfrow=c(2,1), mar=c(2,4,2,1))
layout(matrix(c(1,1,2,2,2), ncol=1))
plot(tydiphtheria$TIME, sqrt(tydiphtheria$PHILADELPHIA),
```

```

type="b", ylab="Sqrt(incidence)")
title("Diphtheria 1914-47")

no = 8; nv = 16; a = 2^seq(1,no+1-1/nv, by = 1/nv)
wfit = cwt(sqrt(tydiphtheria$PHILADELPHIA),
           no, nv, plot=FALSE)
wspec = Mod(wfit)
par(mar=c(1,4,0.25,1))
image(z=wspec, y=a/52, ylim=c(0,3), ylab="Period(year)",
      col=gray((12:32)/32), xaxt='n')
contour(z=wspec, y=a/52, ylim=c(0,3), nlevels = 6,
       zlim=c(mean(wspect), max(wspect)), add=TRUE)

```

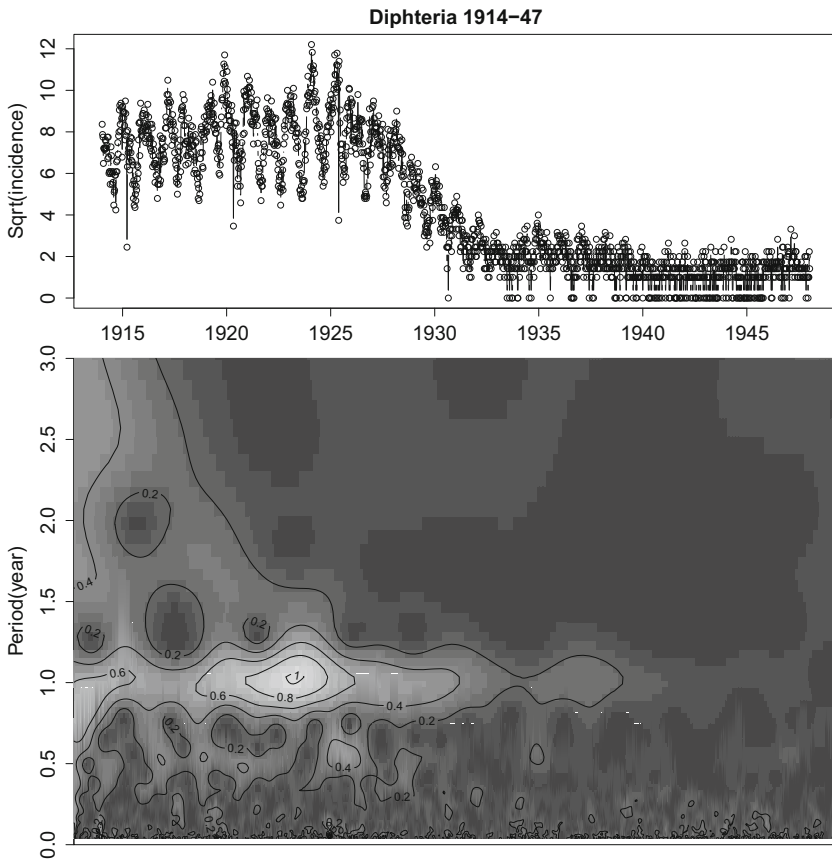


Fig. 6.13 Wavelet spectrum of diphtheria in Philadelphia

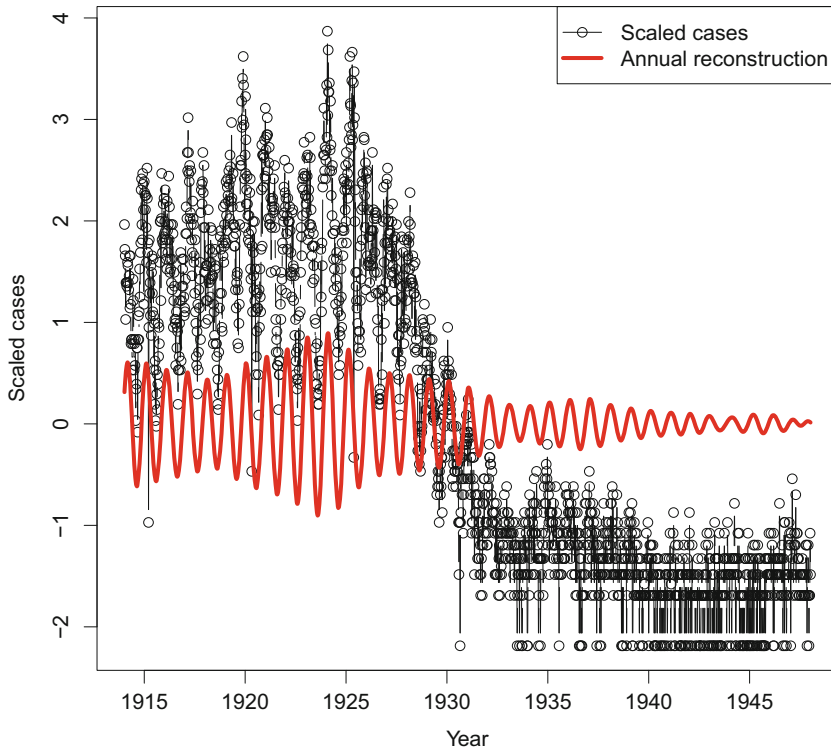


Fig. 6.14 Wavelet reconstructed variability in the 45–60 week range of diphtheria in Philadelphia

We are sometimes interested in using the wavelet as a “filter.” We may for example want to quantitate how the strength of the annual cycle of diphtheria (in the 45–60 week range, say) changes over time. To do this we use wavelet reconstruction around the relevant time scales (Fig. 6.14). For the Morlet wavelet the formula for reconstruction using the j 'th through $j + \text{sth}$ scales is provided by Torrence and Compo (1998). The mid-pass filter clearly illustrates the loss of annual signal over time (Fig. 6.14).

```
#midpass filter
sel=a>45 & a<60
rec=0.6*apply(Re(wfit[,sel])/sqrt(a[sel]), 1,
             sum)/(0.776*(pi^(-1/4)))
data=pi*scale(sqrt(tydiphtheria$PHILADELPHIA))/2
plot(tydiphtheria$TIME, data, type="b", xlab="Year",
     ylab="Scaled cases")
lines(tydiphtheria$TIME, rec, type="l", col=2, lwd=3)
legend("topright", legend=c("Scaled cases",
                             "Annual reconstruction"), pch=c(1, NA), lty=c(1,1),
      lwd=c(1,3), col=c("black", "red"))
```

6.10 Advanced: FFT and Reconstruction

One-hundred-and-twenty years ago, Arthur Schuster proposed the bold idea that any discrete time series can be decomposed and exactly reconstructed from a sum of trigonometric functions. Given its nonstationary transition from annual to biennial epidemics, the pre-vaccination 1944–1964 London measles time series (in the `meas` data set) offers a nice test-bed for this assertion.

The below code generates an animated visualization of the reconstruction. Section 11.6 discusses making in-line and permanent animations in more detail. A web-optimized animated gif can be found in <https://github.com/objornstad/epimdr/blob/master/mov/fftrecon.gif>.

If z is the fast Fourier transform of the time series, then the trigonometric “signal” of the k 'th observation is $\frac{1}{T}(\sum_f (Re(z) \cos(2\pi(k-1)f)) - Im(z) \sin(2\pi(k-1)f))$, where $Re()$ and $Im()$ represent real and imaginary parts. We first piece together relevant bits for the formula; we then do the reconstruction in the `rec2`-object where the contribution of each frequency is weighed:

```
# fft
x <- meas$London
p <- length(x)
z <- fft(x)
f <- seq(from = 0, length = p, by = 1/p)
a <- Re(z)
b <- Im(z)
# reconstruction
rec2 = matrix(NA, ncol = p, nrow = p)
for (k in 1:p) {
  rec2[, k] <- (a * cos(2 * pi * (k - 1) * f) - b *
    sin(2 * pi * (k - 1) * f)) / p
}
```

Finally we can visualize the convergence on the original signal using the sequence of frequencies order by amplitude (highest to lowest importance):

```
sim=rep(0, p)
n=0
samp=order(a^2+b^2, decreasing=TRUE)
for(g in samp){
n=n+1
par(mfrow=c(1,2))
plot(x, ylim=c(0,11000), ylab="Incidence",
  xlab="Biweek")
title(paste("nfreq = ", n))
sim=sim+rec2[g,]
lines(sim, col=2)
```

```
par (new=TRUE)
sc=scale((cos(2*pi*(0:(p-1))*f[g]) -
  sin(2*pi*(0:(p-1))*f[g]))/p)
plot(sc*(a^2+b^2)[g]/max(a^2+b^2), type="l", col=
  gray(.5), ylim=c(-8, 2), axes=FALSE, xlab="", ylab="")
plot(x, sim, ylab="Reconstructed", xlab="Observed",
  ylim=c(0,8000))
#Sys.sleep makes R wait a bit
Sys.sleep(.2)
}
```