10

TIME SERIES MODELS

"A big computer, a complex algorithm and a long time does not equal science."
—Robert Gentleman

In this chapter, we consider time series models. A time series is an ordered sequence of numbers with respect to time. In climatology, you encounter time-series data in a format given by

$$\{h\}_{t=1}^{T} = \{h_1, h_2, \ldots, h_T\} \tag{10.1}$$

where the time t is over a given season, month, week, or day and T is the time series length. The aim is to understand the underlying physical processes that produced the series. A trend is an example. Often by simply looking at a time series, you can pick out a trend that tells you that the process generating the data is changing.

A single time series gives you only one sample from the process. Yet under the ergodic hypothesis, a single time series of infinite length contains the same information (loosely speaking) as the collection of all possible series of finite length. In this case, you can use your series to learn about the nature of the process. This is analogous to spatial interpolation encountered by Chapter 9, where the variogram is computed under the assumption that the rainfall field is stationary.

Here we consider a selection of techniques and models for time series data. We begin by showing you how to overlay plots as a tool for exploratory analysis. This is done to compare the variation between two series qualitatively. We demonstrate large variation in hurricane counts arising from a constant rate process. We then show techniques for smoothing. We continue with a change-point model and techniques for decomposing a continuous-valued series. We conclude with a unique way to create a network graph from a time series of counts and suggest a new definition of a climate anomaly.

10.1 TIME SERIES OVERLAYS

A plot showing your variables on a common time axis is an informative exploratory graph. Values from two different series are scaled to have the same relative range so the covariation in the variables can be compared visually. Here you do this with hurricane counts and sea-surface temperature (SST). Begin by loading *annual.RData*. These data were assembled in Chapter 6. Subset the data for years starting with 1900 and rename the year column.

```
> load("annual.RData")
> dat = subset(annual, Year >= 1900)
> colnames(dat)[1] = "Yr"
```

Plot the basin-wide hurricane count by year, then overlay a plot of the North Atlantic SST. You do this by keeping the current graphics device open with the new=TRUE switch in the par function.

```
> par(las=1, mar=c(5, 4, 2, 4) + .1)
> plot(dat$Yr, dat$B.1, xlab="Year",
+   ylab="Hurricane Count", lab=c(10, 7, 20),
+   type="h", lwd=2)
> par(new=TRUE)
> plot(dat$Yr, dat$sst, type="l", col="red", xaxt="n",
+   yaxt="n", xlab="", ylab="", lwd=2)
> axis(4)
> mtext(expression(paste("SST [",degree,"C]")),
+   side=4, line=2.5, las=0)
> legend("topleft", col=c("black", "red"), lty=1,
+   legend=c("Hurricanes","SST"))
```

You turn off the axis labels in the second plot call and then add them using the axis function where 4 references the vertical axis on the right side of the graph. Axes are numbered clockwise starting from the bottom of the plot. The axis is labeled using the mtext function.

The plot is shown in Figure 10.1. The correspondence between the two series is clear. There tends to be more hurricanes in periods of higher SST and fewer hurricanes in periods of lower SST. You retain the distinction between the two series by using bars for the discrete counts and lines for the continuous SST values.

10.2 DISCRETE TIME SERIES

Your hurricane counts arise from a rate process that is described as Poisson. More precisely, the number of occurrences over an interval is quantified using a Poisson distribution with a rate parameter proportional to the time interval. The counts in nonoverlapping intervals are independent. Since the rate of hurricanes can change

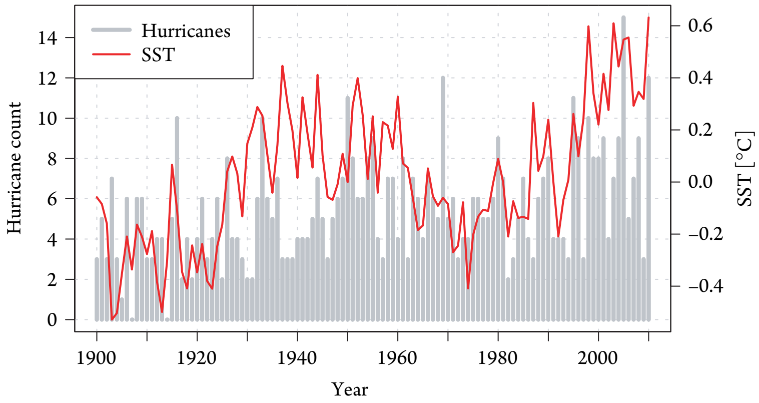

Figure 10.1 Hurricane counts and August–October SST anomalies.

from day to day and from year to year, you assume that the process has a rate that is a function of time ($\lambda(t)$).

Note that if you are interested in yearly counts, you focus on modeling the underlying yearly rate (more precisely, the integral of the underlying instantaneous rate over a year). You can integrate the rate over any time period and obtain the hurricane count over that period. For example, you can integrate to find the expected number of hurricanes after September 15th.

Here you examine methods for estimating the rate process. You first consider running averages to get a smoothed estimate of the annual rate. You then consider a change-point model where the period rate is constant but changes abruptly between periods. Running averages and change-point models provide a description of your series, but they are not very useful for predictions. You begin with a look at interannual count variability.

10.2.1 Count Variability

The time series of hurricane counts appears to have large interannual variable as seen in Figure 10.1. But this might simply be a consequence of the randomness in the counts given the rate. In fact, large variations in small-count processes are often misdiagnosed as physically significant. As an example, consider hurricane counts over a sequence of `N` years with a constant annual Poisson rate `lambda`. What is the probability that you will find at least `M` of these years with a count less than `X` (described as an inactive season) or a count greater than `Y` (described as an active season)?

Here we write it out in steps using R notation.

1. Assign the probability of the count `h` less than `X` or greater than `Y` as `PXY = 1 - ppois(Y) + ppois(X - 1))`. In other words, assign it as one minus the probability that `h` lies between `X` and `Y`, inclusive.
2. Assign an indicator `I = 1` for each year with `h < X` or `h > Y`.

3. Let the sum of `I` have a binomial distribution (Chapter 3) with probability `PXY` and `N`.
4. Let the probability of observing at least `M` of these years be given as
   `PM = 1 - pbinom(M - 1, N, PXY)`

You create the following function to perform these computations.

```
> PM = function(X, Y, lambda, N, M){
+   PXY = 1 - diff(ppois(c(X - 1, Y), lambda))
+   return(1 - pbinom(M - 1, N, PXY))
+   }
```

Arguments for `ppois` are `q` (quantile) and `lambda` (rate) and the arguments for `pbinom` are `q`, `size`, and `prob`.

You use your function to answer the following question. Given an annual rate of 6 hurricanes per year (`lambda`), what is the probability that in a random sequence of 10 years (`N`) you will find at least 2 years (`M`) with a hurricane count less than 3 (`X`) or greater than 9 (`Y`)?

```
> PM(X=3, Y=9, lambda=6, N=10, M=2)
[1] 0.441
```

Thus, you find a 44 percent chance of having 2 years with large departures from the mean rate.

Your function is handy. It protects you against getting fooled by randomness. Indeed, the probability that at least 1 year in 10 falls outside the range of ± 2 standard deviations from the mean is 80 percent. This compares to 37 percent for a set of variables described by a normal distribution and underscores the limitation of using a concept that is relevant for continuous distributions on count data.

By contrast, if you consider the annual global counts over the period 1981–2006[1], you find a mean of 80.7 tropical cyclones per year with a range between 66 and 95. Assuming the global counts are Poisson, you use your function to determine the probability that no years have less than 66 or more than 95 tropical cyclones in the 26-year sample.

```
> 1 - PM(X=66, Y=95, lambda=80.7, N=26, M=1)
[1] 0.0757
```

This low probability provides suggestive evidence to support the notion that the physical processes governing global hurricane activity is more regular than Poisson. The regularity could be due to feedbacks in the climate system. For example, the cumulative effect of many hurricanes over a particular basin might make the atmosphere less conducive for activity in other basins. Or it might be related to a few governing mechanism like the North Atlantic Oscillation (Elsner and Kocher, 2000).

[1] From Elsner et al. (2008b).

10.2.2 Moving Average

A moving average removes year-to-year fluctuation in counts. The assumption is that of a smoothly varying rate process. You use the `filter` function to compute running averages. The first argument in the function is a univariate or multivariate time series and the second is the filter as a vector of coefficients in reverse time order.

For a moving average of length N, the coefficients all have the same value of $1/N$. For example, to compute the 5-year running average of the basin-wide hurricane counts, type

```
> ma = filter(dat$B.1, rep(1, 5)/5)
> str(ma, strict.width="cut")
 Time-Series [1:111] from 1 to 111: NA NA 4.2 3.8 4..
```

The output is an object of class `ts` (time series). Note, the filtering is not performed on values at the ends of the time series, so `NA`s are returned. If you use an odd number of years, then the number of values missing at the start of the filtered series matches the number of values missing at the end of the series.

Here you create a new function called `moveavg` and use it to compute the moving averages of basin counts over 5, 11, and 21 years.

```
> moveavg = function(X, N){filter(X, rep(1, N)/N)}
> h.5 = moveavg(dat$B.1, 5)
> h.11 = moveavg(dat$B.1, 11)
> h.21 = moveavg(dat$B.1, 21)
```

Then plot the moving averages on top of the observed counts.

```
> plot(dat$Yr, dat$B.1, ylab="Hurricane Count/Rate",
+    xlab="Year", col="grey", type="h", lwd=1)
> cls = c("grey", "red", "blue", "green")
> lg = c("Count", "5-Yr Rate", "11-Yr Rate",
+   "21-Yr Rate")
> lines(dat$Yr, h.5, col="red", lwd=2)
> lines(dat$Yr, h.11, col="blue", lwd=2)
> lines(dat$Yr, h.21, col="green", lwd=2)
> legend("topleft", lty=1, lwd=2, col=cls, legend=lg)
```

Figure 10.2 shows the results. Note the reduction in the year-to-year variability as the length of the moving average increases. Note also that the low-frequency variation is not affected. Check this yourself by comparing the means (the mean is the zero frequency) of the moving averages. Thus a moving average is a low-pass "boxcar" filter.

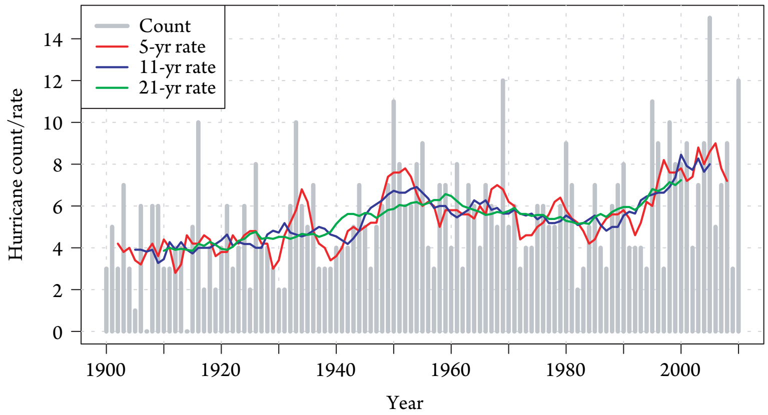

Figure 10.2 Hurricane counts and rates.

10.2.3 Seasonality

Hurricanes are seasonal. Very few occur before July, September is the most active month, and the season is typically over by November. In general, the ocean is too cool and the wind shear too strong during the months of January through May and from November through December. Ocean warmth peaks in early September. Seasonality is evident in plots showing the historical number of hurricanes that have occurred on each day of the year. Here we show you how to model this seasonality to produce a probability of hurricane occurrence as a function of the day of year.

You use the hourly interpolated best-track data described in Chapter 6 and saved in *best.use.RData*. The data span the years from 1851 to 2010. Import the data frame and subset on hurricane-force wind speeds.

```
> load("best.use.RData")
> H.df = subset(best.use, WmaxS >= 64)
> head(H.df)
    Sid Sn SYear         name   Yr Mo Da hr   lon lat
1     1  1  1851 NOT NAMED  1851  6 25  0 -94.8  28
1.1   1  1  1851 NOT NAMED  1851  6 25  1 -94.9  28
1.2   1  1  1851 NOT NAMED  1851  6 25  2 -95.0  28
1.3   1  1  1851 NOT NAMED  1851  6 25  3 -95.1  28
1.4   1  1  1851 NOT NAMED  1851  6 25  4 -95.2  28
1.5   1  1  1851 NOT NAMED  1851  6 25  5 -95.3  28
    Wmax WmaxS DWmaxDt Type Shour maguv diruv  jd
1   80.0  79.8  0.0860    *     0  5.24   271 175
1.1 80.0  79.9  0.0996    *     1  5.25   271 175
1.2 80.1  80.0  0.1114    *     2  5.26   271 175
1.3 80.1  80.2  0.1197    *     3  5.29   270 175
1.4 80.1  80.3  0.1227    *     4  5.32   270 175
1.5 80.0  80.4  0.1187    *     5  5.37   269 175
```

```
        M
1   FALSE
1.1 FALSE
1.2 FALSE
1.3 FALSE
1.4 FALSE
1.5 FALSE
```

Next, create a factor variable from the day-of-year column (`jd`). The day of year starts on the first of January. You use only the integer portion as the rows correspond to separate hours.

```
> jdf = factor(trunc(H.df$jd), levels=1:365)
```

The vector contains the day of year (1 through 365) for all 83,151 hurricane hours in the data set. You could use 366, but there are no hurricanes on December 31 during any leap year over the period of record.

Next, use the `table` function on the vector to obtain total hurricane hours by day of year and create a count of hurricane days by dividing the number of hours and rounding to the nearest integer.

```
> Hhrs = as.numeric(table(jdf))
> Hd = round(Hhrs/24, 0)
```

The vector `Hd` contains the number of hurricane days for each day of the year.

A plot of the raw counts shows that the variation from day to day is large. Here you create a model that smooths these variations. This is done with the `gamlss` function (see Chapter 8) in the **gamlss** package (Rigby and Stasinopoulos, 2005). You model your counts using a Poisson distribution with the logarithmic link as a function of day of year.

```
> require(gamlss)
> julian = 1:365
> sm = gamlss(Hd ~ pb(julian), family=PO, trace=FALSE)
```

Here you use a nonparametric smoothing function on the Julian day. The function is a penalized B-spline (Eilers and Marx, 1996) and is indicated as `pb()` in the model formula. The penalized B-spline is an extension of a Poisson regression that conserves the mean and variance of the daily hurricane counts and that has a polynomial curve as the limit. The Poisson distribution is specified in the `family` argument with `PO`.

Although there are days with hurricanes outside the main season, your interest centers on the months of June through November. Here you create a sequence of Julian days defining the hurricane season and convert them to dates.

```
> hs = 150:350
> doy = as.Date("1970-12-31") + hs
```

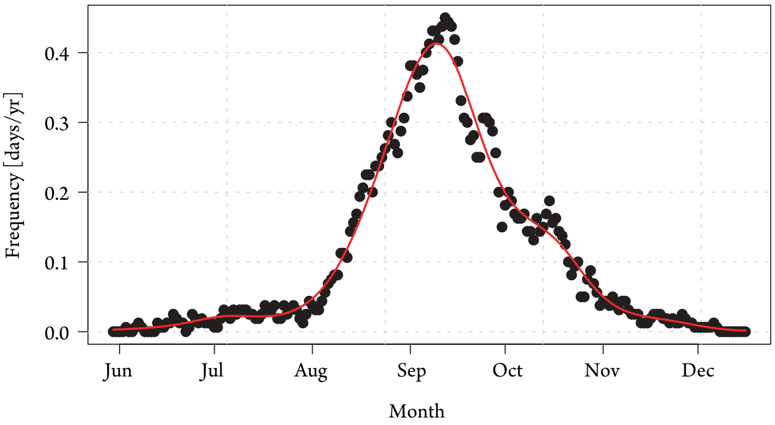

Figure 10.3 Seasonal occurrence of hurricanes.

You then convert the hurricane days to a relative frequency to allow for a probabilistic interpretation. This is done for the actual counts and the smoothed modeled counts.

```
> ny = (2010 - 1851) + 1
> Hdm = Hd[hs]/ny
> smf = fitted(sm)[hs]/ny
```

Finally, you plot the modeled and actual daily frequencies by typing

```
> plot(doy, Hdm, pch=16, xlab="",
+   ylab="Frequency (days/yr)")
> lines(doy, smf, lwd=2, col="red")
```

The results are shown in Figure 10.3. Points show the observed relative frequency of hurricanes by day of year. The red line is the fitted values of the model for the frequencies. Horizontal tic marks indicate the first day of the month.

On average, hurricane activity increases slowly until the beginning of August as the ocean warms and wind shear subsides. The increase is more pronounced starting in early August and peaks around the first or second week in September. The decline starting in mid-September is somewhat less pronounced than the increase and is associated with ocean cooling. There is a minor secondary peak during the middle of October related to hurricane genesis over the western Caribbean Sea. The climate processes that make this part of the basin relatively active are likely somewhat different than the processes occurring during the peak of the season.

10.3 CHANGE POINTS

Hurricane activity can change from inactive to active abruptly. In this case, a change-point model is appropriate for describing the time series. Here a change point refers to a jump in the rate of activity from one time to the next. The underlying assumption is a discontinuity in the rates. For example, suppose hurricanes suddenly become more frequent in the years 1934 and 1990, then the model would still be Poisson, but with different rates in the periods (epochs) 1900–1933, 1934–1989, and 1990–2010. Here you use the annual data loaded in §10.1 to build a change-point model.

10.3.1 Counts

The simplest model is one with a single change point. For instance, you check to see whether a model that has a rate change during a given year is better than a model that does not have a change during that year. Thus you have two models: one with a change point and the other without one. To make a choice, you check to see which model has the lower Schwarz Bayesian Criterion (SBC).

The SBC is proportional to $-2\log[p(\text{data}|\text{model})]$, where $p(\text{data}|\text{model})$ is the probability of the data given the model (see Chapter 4). This is done using the `gamlss` function in the **gamlss** package. Make the package available and obtain the SBC value for each of three models by typing

```
> require(gamlss, quiet=TRUE)
> gamlss(B.1 ~ 1, family=PO, data=dat,
+   trace=FALSE)$sbc
[1] 529
> gamlss(B.1 ~ I(Yr >= 1910), family=PO, data=dat,
+   trace=FALSE)$sbc
[1] 529
> gamlss(B.1 ~ I(Yr >= 1940), family=PO, data=dat,
+   trace=FALSE)$sbc
[1] 515
```

Here the Poisson family is given as `PO` with the logarithm of the rate as the default link (Stasinopoulos and Rigby, 2007). The first model is one with no change point. The next two are change-point models with the first having a change point in the year 1910 and the second having a change point in 1940. The change-point models use the indictor function `I` to assign a `TRUE` or `FALSE` to each year based on logical expression involving the variable `Yr`.

The SBC value is 528.5 for the model with no change points. This compares with an SBC value of 528.7 for the change-point model where the change occurs in 1910 and a value of 514.8 for the change-point model where the change occurs in 1940. Since the SBC is lower in the latter case, 1940 is a candidate year for a change point.

You apply this procedure successively where each year gets considered in turn as a possible change point. You then plot the SBC as a function of year (Fig. 10.4). The horizontal line is the SBC for a model with no change points and tick marks are local minimum of SBC. Here the SBC for the model without a change point is adjusted by adding $2\log(20)$ to account for the prior possibility of five or six equally likely change points over the period of record. Here you find four candidate change points based on local minima of the SBC. The years are 1995, 1948, 1944, and 1932.

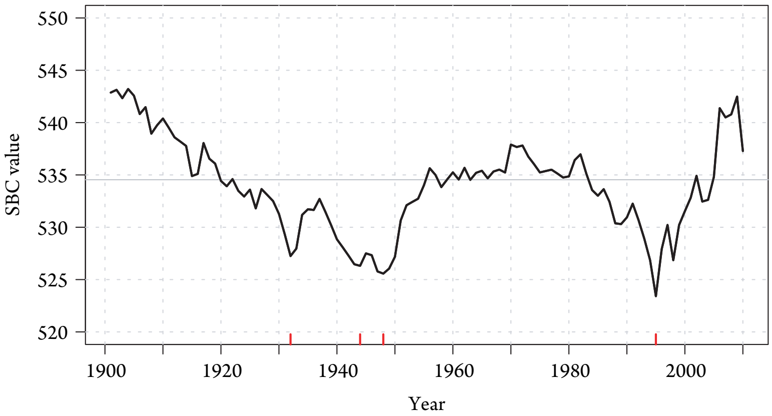

Figure 10.4 Schwarz Bayesian criterion (SBC) for change points.

You assume a priori that there is at most one change point per decade and that the posterior probability of the intercept model is 20 times that of the change-point model. This gives you 12 possible models (1995 only, 1995 & 1948, 1995 & 1948 & 1932, etc.) including the intercept-only model but excludes models with both 1944 and 1948 as the changes occur too close in time.

Next, you estimate the posterior probabilities for each of the 12 models using

$$\Pr(M_i|\text{data}) = \frac{\exp(-.5 \cdot \text{SBC}(M_i))}{\sum_{j=1}^{12} \exp(-.5 \cdot \text{SBC}(M_j))} \tag{10.2}$$

where the models are given by M_i, for $i = 1, \ldots, 12$. The results are shown in Table 10.1. The top three models have a total posterior probability of 80%. These

Table 10.1 Model posterior probabilities from most (top) to least probable.

	Formula	*Probability*
X10	B.1~I(Yr>=1995) + I(Yr>=1932)	0.43
X6	B.1~I(Yr>=1995) + I(Yr>=1944)	0.20
X4	B.1~I(Yr>=1995) + I(Yr>=1948)	0.18
X12	B.1~I(Yr>=1995) + I(Yr>=1948) + I(Yr>=1932)	0.07
X14	B.1~I(Yr>=1995) + I(Yr>=1944) + I(Yr>=1932)	0.06
X3	B.1~I(Yr>=1948)	0.02
X5	B.1~I(Yr>=1944)	0.01
X2	B.1~I(Yr>=1995)	0.01
X9	B.1~I(Yr>=1932)	0.01
X11	B.1~I(Yr>=1948) + I(Yr>=1932)	0.01
X13	B.1~I(Yr>=1944) + I(Yr>=1932)	0.00
X1	B.1~1	0.00

Table 10.2 Best model coefficients and standard errors.

	Estimate	*Std. Error*	*t value*	*Pr(>|t|)*
(Intercept)	3.9063	0.4101	9.53	0.0000
I(Yr >= 1995)TRUE	2.5069	0.6494	3.86	0.0002
I(Yr >= 1932)TRUE	1.6493	0.5036	3.28	0.0014

models all include 1995 with 1932, 1944, and 1948 competing as the second most important change-point year. You can select any of the models, but it makes sense to choose one with a high posterior probability. Note the weaker support for the single change-point models and even less support for the no change-point model.

The single best model has change points in 1932 and 1995. The coefficients of this model are shown in Table 10.2. The model predicts a rate of 3.9 hur/yr in the period 1900–1931. The rate jumps to 6.4 hur/yr in the period 1931–1994 and jumps again to 8.1 in the period 1995–2010.

10.3.2 Covariates

To better understand what might be causing the shifts in hurricane activity, here you include known covariates in the model. The idea is that if the shift is no longer significant after adding a covariate, then you conclude that the likely causal mechanism is a change in climate.

The two important covariates for annual basin-wide hurricane frequency are SST and the SOI as used throughout this book. You first fit and summarize a model using the two change points and the two covariates.

```
> model1 = gamlss(B.1 ~ I(Yr >= 1932) + I(Yr >= 1995) +
+   sst + soi, family=PO, data=dat, trace=FALSE)
> summary(model1)
```

You find that the change point at 1995 has the largest p-value among the variables. You also note that the model has an SBC of 498.5.

You consider whether the model can be improved by removing the change point at 1995, so you remove it and refit the model.

```
> model2 = gamlss(B.1 ~ I(Yr >= 1932)
+    + sst + soi, family=PO, data=dat, trace=FALSE)
> summary(model2)
```

With the reduced model, you find all variables statistically significant (p-value less than 0.1) and the model has an SBC of 496.3, which is lower than the SBC of your first model that includes 1995 as a change point.

Thus you conclude that the shift in the rate at 1995 is more likely the result of a synchronization (Tsonis et al., 2006) of the effects of SST and ENSO on hurricane activity than is the shift in 1932. The shift in 1932 is important after including SST and ENSO influences providing evidence that the increase in activity at this time is likely due, at least in part, to improvements in observing technologies.

A change-point model is useful for detecting rate shifts caused by climate and observational improvements. When used together with climate covariates, it can help you differentiate between these two possibilities. However, change-point models are not particularly useful for predicting when the next change will occur.

10.4 CONTINUOUS TIME SERIES

The SST temperature, the SOI, and the NAO are continuous time series. Values fluctuate over a range of scales often without abrupt changes. In this case, it can be useful to split the series into a few components where each component has a smaller range of scales.

Here your goal is to decompose the SST time series as an initial step in creating a time-series model. The model can be used to make predictions of future SST values. Predicted SST values are subsequently used in your hurricane frequency model to forecast the probability of hurricanes (Elsner et al., 2008a).

You return to your monthly SST values over the period 1856–2010. As you did with the NAO values in Chapter 5, you input the data and create a continuous-valued time series object (sst.ts) containing monthly SST values beginning with January 1856.

```
> SST = read.table("SST.txt", header=TRUE)
> sst.m = as.matrix(SST[6:160, 2:13])
> sst.v = as.vector(t(sst.m))
> sst.ts = ts(sst.v, frequency=12, start=c(1856, 1))
```

First you plot your SST time series by typing

```
> plot(sst.ts, ylab="SST (C)")
```

The graph shows that the SST is dominated by interannual variability (Fig. 10.5). The ocean is coldest in February and March and warmest in August and September. The average temperature during March is 18.6°C and during August is 23.1°C. There also appears to be a trend toward greater warmth, although it is difficult to see because of the larger interannual variations.

The SST times series can be decomposed into components using the stl function. The function accepts a time series object as its first argument and the type of smoothing window is specified through the s.window argument.

```
> sdts = stl(sst.ts, s.window="periodic")
```

The seasonal component is found by a local regression smoothing of the monthly means. The seasonal values are then subtracted, and the remainder of the series smoothed to find the trend. The overall time-series mean value is removed from the seasonal component and added to the trend component. The process is iterated a few times. What remains is the difference between the actual monthly values and the sum

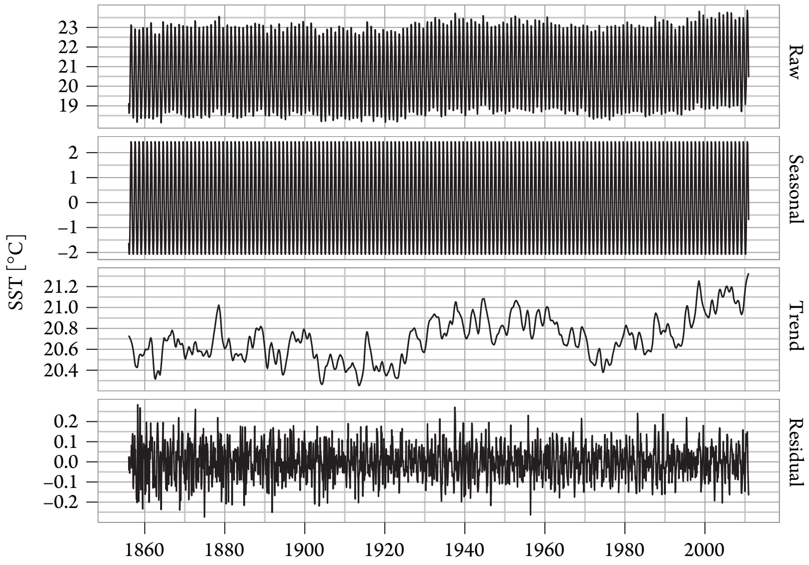

Figure 10.5 Monthly raw and component SST values.

of the seasonal and trend components. Note that if you have change points in your time series, you can use the **bfast** package and the `bfast` function to decompose your time series. In this case, the trend component has the change points.

The raw and component series are plotted in Figure 10.5. The data are prepared as follows. First a vector of dates is constructed using the `seq.dates` function from the **chron** package. This allows you to display the graphs using a scale that corresponds to real dates.

```
> require(chron)
> date = seq.dates(from="01/01/1856", to="12/31/2010",
+   by="months")
> #dates = chron(dates, origin=c(1, 1, 1856))
```

Next, a data frame is constructed that contains the vector of dates, the raw monthly SST time series, and the corresponding components from the seasonal decomposition.

```
> datw = data.frame(Date=as.Date(date),
+   Raw=as.numeric(sst.ts),
+   Seasonal=as.numeric(sdts$time.series[, 1]),
+   Trend=as.numeric(sdts$time.series[, 2]),
+   Residual=as.numeric(sdts$time.series[, 3]))
> head(datw)
        Date  Raw Seasonal Trend Residual
1 1856-01-01 19.1   -1.621  20.7  0.02587
2 1856-02-01 18.6   -2.060  20.7 -0.04036
```

```
3 1856-03-01 18.7   -2.068  20.7  0.02453
4 1856-04-01 19.0   -1.640  20.7 -0.05631
5 1856-05-01 19.9   -0.756  20.7 -0.01318
6 1856-06-01 21.2    0.478  20.7  0.00159
```

Here the data are in the “wide” form like a spreadsheet. To make them easier to plot as separate time-series graphs, you create a “long” form of the data frame with the `melt` function in the **reshape** package. The function melds your data frame into a form suitable for casting (Wickham, 2007). You specify the data frame and your `Date` column as your id variable. The function assumes that remaining variables are measure variables (non id variables) with the column names turned into a vector of factors.

```
> require(reshape)
> datl = melt(datw, id="Date")
> head(datl); tail(datl)
        Date variable value
1 1856-01-01      Raw  19.1
2 1856-02-01      Raw  18.6
3 1856-03-01      Raw  18.7
4 1856-04-01      Raw  19.0
5 1856-05-01      Raw  19.9
6 1856-06-01      Raw  21.2
           Date variable   value
7435 2010-07-01 Residual  0.0807
7436 2010-08-01 Residual  0.1489
7437 2010-09-01 Residual  0.0601
7438 2010-10-01 Residual -0.0501
7439 2010-11-01 Residual -0.1157
7440 2010-12-01 Residual -0.1666
```

Here you make use of the **ggplot2** package (see Chapter 5) to create a facet grid to display your time-series plots with the same time axis. The `qplot` function graphs the decomposed time-series values grouped by variable. The argument `scale="free_y"` allows the y axes to have different scales. This is important as the decomposition results in a large seasonal component centered on zero, while the trend component is smaller.

```
> require(ggplot2)
> qplot(Date, value, data=datl, geom="line",
+   group=variable) + facet_grid(variable ~.,
+   scale="free_y")
```

The monthly time-series components are shown in Figure 10.5. The observed (raw) values are shown in the top panel. The seasonal component, trend component, and residuals are also shown in separate panels on the same time-series axis. Temperatures increase by more than 0.5°C over the past 100 years. But the trend is not monotonic. The residuals show year-to-year variation generally between −0.15 and +0.15°C with somewhat larger variation before about 1871.

You can build separate time series models for each component. For example, an autoregressive moving average (ARMA) model can be built for the residual component (R_t) be used. An ARMA model with p autoregressive terms and q moving average terms [ARMA(p, q)] is given by

$$R_t = \sum_{i=1}^{p} \phi_i R_{t-i} + \sum_{i=1}^{q} \theta_i \varepsilon_{t-i} + \varepsilon_t \tag{10.3}$$

where the ϕ_i's and the θ_i's are the parameters of the autoregressive and moving average terms respectively and ε_t's are random white noise assumed to be described by independent normal distributions with zero mean and variance σ^2. For the trend component, an ARIMA model is more appropriate. An ARIMA model generalizes the ARMA model by removing the nonstationarity through an initial differencing step (the "integrated" part of the model).

Here you use the `ar` function to determine the autoregressive portion of the series using the AIC.

```
> ar(datw$Trend)
Call:
ar(x = datw$Trend)

Coefficients:
     1       2       3       4       5       6
 1.279  -0.058  -0.101  -0.062  -0.045  -0.032
     7       8       9      10      11
-0.008  -0.023   0.010  -0.034   0.067

Order selected 11  sigma^2 estimated as  0.000298
```

Result shows an autoregressive order of 11 months. Continuing, you assume that the integrated and moving average orders are both one.

```
> model = arima(datw$Trend, order=c(11, 1, 1))
```

You then use the model to make monthly forecasts out to 36 months using the `predict` method. Predictions are made at times specified by the `newxreg` argument.

```
> nfcs = 36
> fcst = predict(model, n.ahead=nfcs)
```

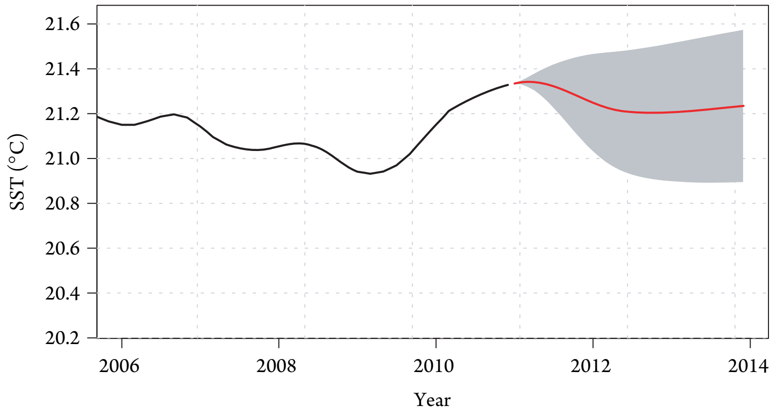

Figure 10.6 Observed and forecast SST trend component.

You plot the forecasts along the corresponding date axis by typing

```
> newdate = seq.dates(from="01/01/2011",
+   to="12/01/2013", by="months")
> plot(newdate, fcst$pred, type="l", ylim=c(21, 21.5))
```

The last 5 years of SST trend and the 36-month forecast are shown in Figure 10.6. The observed values are in black and the forecast values are in red. A 95 percent confidence band is shown in gray. Here you use the same scale. The band is quite large after a few months. A forecast of the actual SST must include forecasts for the seasonal and residual components as well.

10.5 TIME-SERIES NETWORK

Here, we show you an interesting new way to characterize a time series. You first map the series to a network using geometry and then employ tools from graph theory to get a unique perspective on your data.

Network analysis is the application of graph theory. Graph theory is the study of mathematical structures used to model relations between objects. Objects and relations can be many things with the most familiar being people and friendships.

Network analysis was introduced into climatology by Tsonis and Roebber (2004). They used values of geopotential height on a spatial grid and the relations were based on correlation. Here you use network analysis to examine year-to-year relations in hurricane activity. The idea is new and requires mapping a time series to a network (Lacasa et al., 2008). The presentation here follows the work of Elsner et al. (2009).

10.5.1 Time Series Visibility

How can a time series of hurricane counts be represented as a network? Consider the plot in Figure 10.7. The time series of U.S. hurricane counts forms a discrete land-

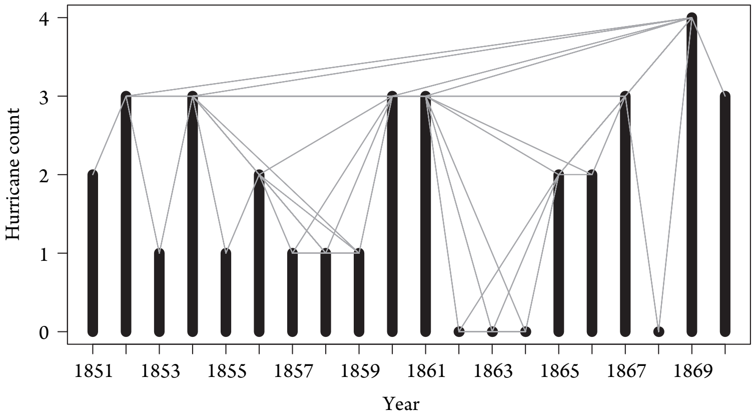

Figure 10.7 Visibility landscape for hurricane counts.

scape. A bar is connected to another bar if there is a line of sight (visibility line) that allows the bars to "see" each other. Here visibility lines are drawn for all 10 bars. It is clear that 1869 by virtue of its high hurricane count (4) can see 1852, 1854, 1860, 1861, 1867, 1868, and 1870, while 1868 with its zero count can see only 1867 and 1869. Lines do not cut through bars. In this way, each year in the time series is linked in a network. The nodes are the years and the links (edges) are the visibility lines.

More formally, let h_a be the hurricane count for year t_a and h_b the count for year t_b, then 2 years are linked if for any other year t_i with count h_i

$$h_i \leq h_b + (h_a - h_b)\frac{t_b - t_i}{t_b - t_a} \tag{10.4}$$

By this definition, each year is visible to at least its nearest neighbors (the year before and the year after), but not itself. The network is invariant under rescaling the horizontal or vertical axes of the time series as well as under horizontal and vertical translations (Lacasa et al., 2008).

In network parlance, years are nodes and the visibility lines are the links (or edges). The network arises by releasing the years from chronological order and treating them as nodes linked by visibility lines. Here we see that 1869 is well connected, while 1853 is not. Years featuring many hurricanes generally result in more links, especially if neighboring years have relatively few hurricanes. This can be seen by comparing 1853 with 1858. Both years have only a single hurricane, but 1858 is adjacent to years that also have a single hurricane so it is linked to four other years. By contrast, 1853 is next to 2 years each with three hurricanes so it has the minimum number of two links. The degree of a node is the number of links connected to it.

The function `get.visibility` available in **get.visibility.R** computes the visibility lines. It takes a vector of counts as input and returns three lists: one containing the incidence matrix (`sm`), another a set of node edges (`node`), and the third a degree distribution (`pk`), which indicate the number of years with k number of edges. Source the code and compute the visibility lines by typing,

```
> source("get.visibility.R")
> vis = get.visibility(annual$US.1)
```

10.5.2 Network Plot

You use the `network` function from the **network** package (Butts et al., 2011) to create a network object from the incidence matrix by typing

```
> require(network)
> net = network(vis$sm, directed=FALSE)
```

Then use the plot method for network objects to graph the network.

```
> plot(net, label=1851:2010, label.cex=.6,
+   vertex.cex=1.5, label.pos=5, edge.col="grey")
```

The results are shown in Figure 10.8. Node color indicates the number of links (degree) going from light purple (few) to red. Here the placement of years on the network plot is based on simulated annealing (Kamada and Kawai, (1989)), and the nodes are colored, based on the number of edges. Years with the largest number of edges are more likely to be found in dense sections of the network and are colored dark red. Years with fewer edges are found near the perimeter of the network and are colored light purple.

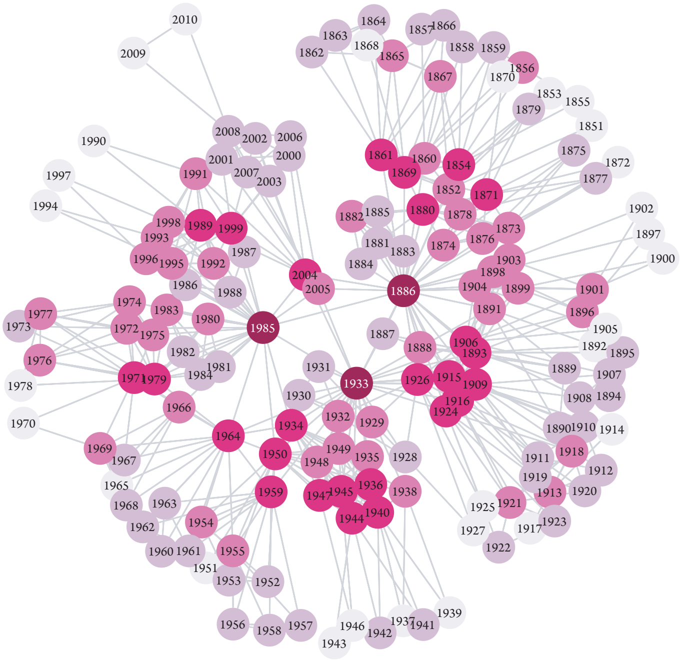

Figure 10.8 Visibility network of U.S hurricanes.

The **sna** package (Butts, 2010) contains functions for computing properties of your network. First create a square adjacency matrix where the number of rows is the number of years and each element is a zero or one depending on whether the years are linked and with zeros along the diagonal (a year is not linked with itself). Then compute the degree of each year indicating the number of years it can see and find which years can see the farthest.

The year with the highest degree is 1886 with 34 links. Two other years with high degree include 1933 with 31 links and 1985 with 28 links. Other relatively highly connected years are 1893, 1950 1964, and 1906, in that order. The average degree is 6.6, but the degree distribution is skewed so this number says little about a typical year.

10.5.3 Degree Distribution and Anomalous Years

The total number of links in the network (sum of the links over all nodes) is 1,054. There are 160 nodes, so 20 percent of the network consists of 32 of them. If you rank the nodes by the number of links, you find that the top 20 percent account for 40 percent of the links.

You plot the degree distribution of your network by typing

```
> plot(vis$pk$k, cumsum(vis$pk$P), pch=16, log="x",
+   ylab="Proportion of Years With k or Fewer Links",
+   xlab="Number of Links (k)")
```

The distribution (Fig. 10.9) is the cumulative percentage of years with *k* or fewer links as a function of the number of links. The horizontal axis is plotted using a log scale. Just over 80 percent of all years have 10 or fewer links, and over 50 percent have 5 or fewer. Although the degree distribution is skewed to the right, it does not appear to represent a small-world network (power-law distribution).

We perform a Monte Carlo (MC) simulation by randomly drawing counts from a Poisson distribution with the same number of years and the same hurricane rate

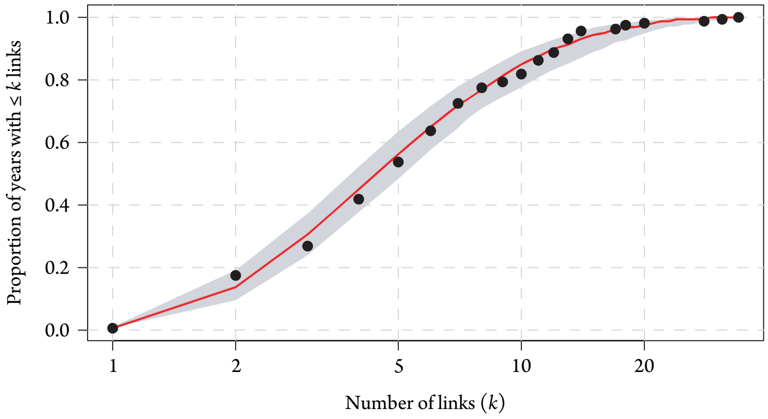

Figure 10.9 Degree distribution of the visibility network.

as the observations. A visibility network is constructed from the random counts and the degree distribution computed as before. The process is repeated 1,000 times after which the median and quantile values of the degree distributions are obtained. The median distribution is shown as a red line in Figure 10.9 and the 95 percent confidence interval shown as a gray band. Results indicate that the degree distribution of your hurricane count data does not deviate significantly from the degree distribution of a Poisson random time series.

However, it does suggest a new way to think about anomalous years. Years are anomalous not in a statistical sense of violating a Poisson assumption, but in the sense that the temporal ordering of the counts identifies a year that is unique in that it has a large count but is surrounded before and after by years with low counts. Thus we contend that node degree is a useful indicator of climatologically anomalous year. That is, a year that stands above most of the other years, but particularly above its "neighboring" years represents more of an anomaly in a physical sense than does a year that is simply well above the average. Node degree captures information about the frequency of hurricanes for a given year and information about the relationship of that frequency to the frequencies over the given year's recent history and near future.

The relationship between node degree and the annual hurricane count is tight but not exact. Years with a low number of hurricanes are ones that are not well connected to other years, while years with an above-normal number are ones that are more connected on average. The Spearman rank correlation between year degree and year count is 0.73. But this is largely a result of low count years. The correlation drops to 0.48 when considering only years with more than two hurricanes. Thus high count is necessary but not sufficient for characterizing the year as anomalous, as perhaps it should be.

10.5.4 Global Metrics

Global network metrics are used to compare different data sets. One example is the diameter of the network as the length of the longest geodesic path between any two years for which a path exists. A geodesic path (shortest path) is a path between two years such that no shorter path exists. For instance, in Figure 10.7, you see that 1861 is connected to 1865 directly and through a connection with 1862. The direct connection is a path of length one while the connection through 1862 is a path of length two.

The **igraph** package (Csardi and Nepusz, 2006) contains functions for computing network analytics. To find the diameter of your visibility network, load the package, create the network (graph) from the list of edges, then use the `diameter` function. Prefix the function name with the package name and two colons to avoid a conflict with the same name from another loaded package.

```
> require(igraph)
> vis = get.visibility(annual$US.1)
> g = graph.edgelist(vis$sm, directed=FALSE)
> igraph::diameter(g)
[1] 5
```

The result indicates that any 2 years are separated by at most 5 links, although there is more than one such geodesic.

Transitivity measures the probability that the adjacent nodes are themselves connected. Given that year *i* can see years *j* and *k*, what is the probability that year *j* can see year *k*? In a social network, transivity indicates the likelihood that any two of your friends are themselves friends. To compute the transitivity for your visibility network, type

```
> tran = transitivity(g)
> round(tran, 3)
[1] 0.468
```

The transitivity tells you that there is a 46.8 percent chance that two adjacent nodes of a given node are connected. The higher the probability, the greater the network density. The visibility network constructed from Gulf hurricane counts has a transitivity of 0.563, which compares with a transitivity of 0.479 for the network constructed from Florida counts. The network density is inversely related to interannual variance, but this rather large difference provides some evidence to support clustering of hurricanes in the vicinity of Florida relative to the Gulf coast region (see Chapter 11). A-nMC simulation would help you interpret the difference against the backdrop of random variations.

Another global property is the minimum spanning tree. A tree is a connected network that contains no closed loops. By "connected," we mean that every year in the network is reachable from every other year via some path through the network (Newman, 2010). A tree is said to span if it connects all the years together. A network may have more than one spanning tree. The minimum spanning tree is the one with the fewest number of edges. A network may contain more than one minimum spanning tree. You compute the minimum spanning tree by typing

```
> mst = minimum.spanning.tree(g)
> net = network(get.edgelist(mst))
```

The result is an object of class `igraph`. This is converted to a network object by specifying the edge list in the `network` function. You plot the network tree by typing

```
> plot(net)
```

The graph is shown in Figure 10.10, where the nodes are labeled with their corresponding years and are colored according to the level of "betweenness." Arrows point toward later years. The node betweenness (or betweenness centrality) is the number

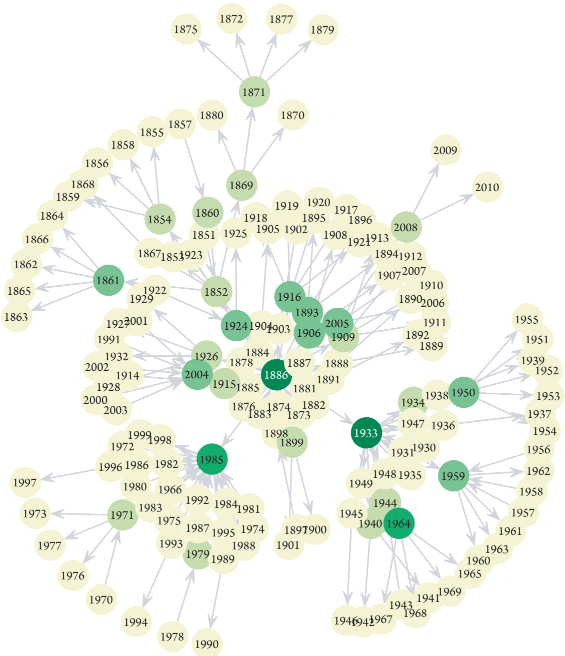

Figure 10.10 Minimum spanning tree of the hurricane visibility network.

of geodesics (shortest paths) going through it. By definition, the minimum spanning tree must have a transitivity of zero. You check this by typing

```
> transitivity(mst)
[1] 0
```

In summary, the visibility network is the set of years as nodes together with links defined by a sight line on the time series graph such that the line does not intersect a count bar. Analysis of the topological properties of the network, like betweenness, provide new insights into the relationship between hurricanes and climate.

This chapter showed you some methods and models for working with time-series data. We began by showing you how to overlay time-series plots. We then discussed the nature of discrete time series and showed how to compute moving averages and how to create a model for describing the day-to-day variation in hurricane activity. Next we showed how to analyze and model count data for change points and how to interpret the shifts in light of climate variability confounded by technological advances. We then looked at ways to analyze and model continuous time series. We showed how to decompose a series into its component parts and how to model the nonseasonal part with an ARMA model. We finished with a novel way to construct a network from time-series counts. We showed how metrics from the network provide insight into hurricane climatology. In the next chapter, we consider ways to analyze and model hurricane clusters.