Chapter 25

Time Series

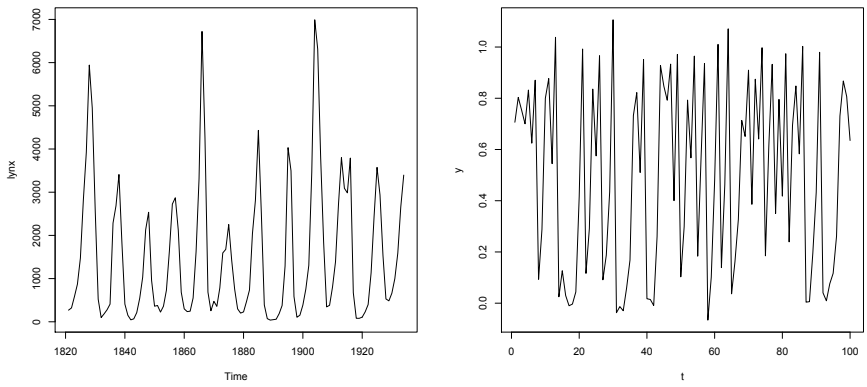
So far, we have assumed that all data points are pretty much independent of each other. In the chapters on regression, we assumed that each Y_i was independent of every other, given its X_i, and we often assumed that the X_i were themselves independent. In the chapters on multivariate distributions and even on causal inference, we allowed for arbitrarily complicated dependence between the *variables*, but each *data-point* was assumed to be generated independently. We will now relax this assumption, and see what sense we can make of dependent data.

25.1 Time Series, What They Are

The simplest form of dependent data are time series, which are just what they sound like: a series of values recorded over time. The most common version of this, in statistical applications, is to have measurements of a variable or variables X at equally-spaced time-points starting from t, written say $X_t, X_{t+h}, X_{t+2h}, \ldots$, or $X(t), X(t+h), X(t+2h), \ldots$. Here h, the amount of time between observations, is called the "sampling interval", and $1/h$ is the "sampling frequency" or "sampling rate".

Figure 25.1 shows two fairly typical time series. One of them is actual data (the number of lynxes trapped each year in a particular region of Canada); the other is the output of a purely artificial model. (Without the labels, it might not be obvious which one was which.) The basic idea of all of time series analysis is one which we're already familiar with from the rest of statistics: we regard the actual time series we see as one realization of some underlying, partially-random ("stochastic") process, which generated the data. We use the data to make guesses ("inferences") about the process, and want to make *reliable* guesses while being clear about the uncertainty involved. The complication is that each observation is dependent on all the other observations; in fact it's usually this dependence that we want to draw inferences about.

25.1.0.0.2 Other kinds of time series One sometimes encounters irregularly-sampled time series, $X(t_1), X(t_2), \ldots$, where $t_i - t_{i-1} \neq t_{i+1} - t_i$. This is mostly an annoyance, unless the observation times are somehow dependent on the values.




```
data(lynx); plot(lynx)
```

Figure 25.1: Left: annual number of trapped lynxes in the Mackenzie River region of Canada. Right: a toy dynamical model. (See code online for the toy.)

Continuously-observed processes are rarer — especially now that digital sampling has replaced analog measurement in so many applications. (It is more common to model the process as evolving continuously in time, but observe it at discrete times.) We skip both of these in the interest of space.

Regular, irregular or continuous time series all record the same variable at every moment of time. An alternative is to just record the sequence of times at which some event happened; this is called a "point process". More refined data might record the time of each event and its type — a "marked point process". Point processes include very important kinds of data (e.g., earthquakes), but they need special techniques, and we'll skip them.

25.1.0.0.3 Notation For a regularly-sampled time series, it's convenient not to have to keep writing the actual time, but just the position in the series, as $X_1, X_2, \ldots$, or $X(1), X(2), \ldots$. This leads to a useful short-hand, that $X_i^j = (X_i, X_{i+1}, \ldots X_{j-1}, X_j)$, a whole block of time; some people write $X_{i:j}$ with the same meaning.

25.2 Stationarity

In our old IID world, the distribution of each observation is the same as the distribution of every other data point. It would be nice to have something like this for time series. The property is called **stationarity**, which doesn't mean that the time series never changes, but that its *distribution* doesn't.

More precisely, a time series is **strictly stationary** or **strongly stationary** when X_1^k and X_t^{t+k-1} have the same distribution, for all k and t — the distribution of

blocks of length k is **time-invariant**. Again, this doesn't mean that every block of length k has the same value, just that it has the same distribution of values.

If there is strong or strict stationarity, there should be **weak** or **loose** (or **wide-sense**) stationarity, and there is. All it requires is that $\mathbf{E}\left[X_1\right] = \mathbf{E}\left[X_t\right]$, and that $\mathrm{Cov}\left[X_1, X_k\right] = \mathrm{Cov}\left[X_t, X_{t+k-1}\right]$. (Notice that it's not dealing with whole blocks of time any more, just single time-points.) Clearly (exercise!), strong stationarity implies weak stationarity, but not, in general, the other way around, hence the names. It may not surprise you to learn that strong and weak stationarity coincide when X_t is a Gaussian process, but not,in general, otherwise.

You should convince yourself that an IID sequence is strongly stationary.

25.2.1 Autocorrelation

Time series are **serially dependent**: X_t is in general dependent on all earlier values in time, and on all later ones. Typically, however, there is **decay of dependence** (sometimes called **decay of correlations**): X_t and X_{t+h} become more and more nearly independent as $h \to \infty$. The oldest way of measuring this is the **autocovariance**,

$$\gamma(h) = \mathrm{Cov}\left[X_t, X_{t+h}\right] \tag{25.1}$$

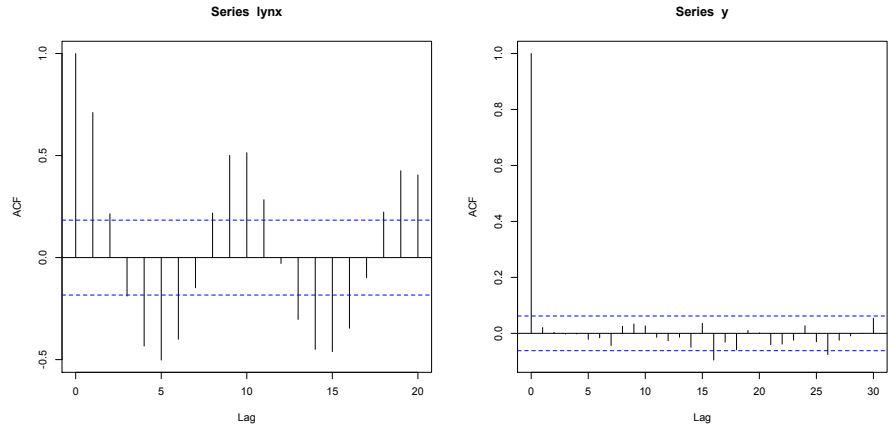
which is well-defined just when the process is weakly stationary. We could equally well use the **autocorrelation**,

$$\rho(h) = \frac{\mathrm{Cov}\left[X_t, X_{t+h}\right]}{\mathrm{Var}\left[X_t\right]} = \frac{\gamma(h)}{\gamma(0)} \tag{25.2}$$

again using stationarity to simplify the denominator.

As I said, for most time series $\gamma(h) \to 0$ as h grows. Of course, $\gamma(h)$ could be exactly zero while X_t and X_{t+h} are strongly dependent. Figure 25.2 shows the autocorrelation functions (ACFs) of the lynx data and the simulation model; the correlation for the latter is basically never distinguishable from zero, which doesn't accord at all with the visual impression of the series. Indeed, we can confirm that *something* is going on the series by the simple device of plotting X_{t+1} against X_t (Figure 25.3). More general measures of dependence would include looking at the Spearman rank-correlation of X_t and X_{t+h}, or quantities like mutual information.

Autocorrelation is important for four reasons, however. First, because it is the oldest measure of serial dependence, it has a "large installed base": everybody knows about it, they use it to communicate, and they'll ask you about it. Second, in the rather special case of Gaussian processes, it really does tell us everything we need to know. Third, in the somewhat less special case of linear prediction, it tells us everything we need to know. Fourth and finally, it plays an important role in a crucial theoretical result, which we'll go over next.




```
acf(lynx); acf(y)
```

Figure 25.2: Autocorrelation functions of the lynx data (above) and the simulation (below). The `acf` function plots the autocorrelation function as an automatic side-effect; it actually returns the actual value of the autocorrelations, which you can capture. The 95% confidence interval around zero is computed under Gaussian assumptions which shouldn't be taken too seriously, unless the sample size is quite large, but are useful as guides to the eye.

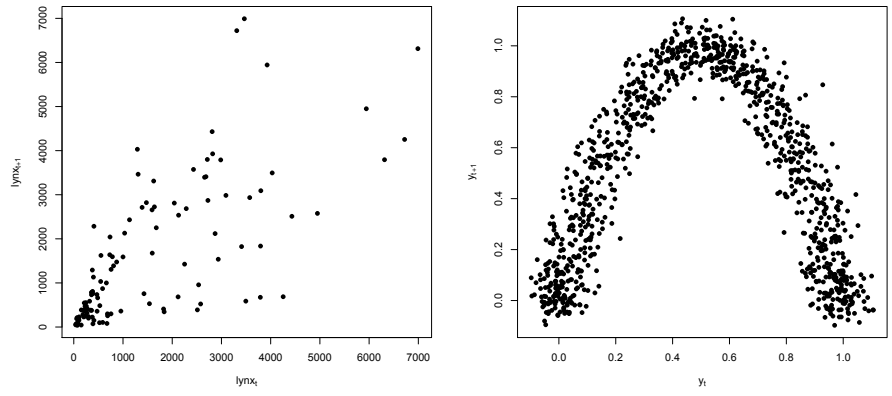


Figure 25.3: Plots of X_{t+1} versus X_t, for the lynx (above) and the simulation (below). (See code online.) Note that even though the correlation between successive iterates is next to zero for the simulation, there is clearly a lot of dependence.

25.2.2 The Ergodic Theorem

With IID data, the ultimate basis of all our statistical inference is the law of large numbers, which told us that

$$\frac{1}{n}\sum_{i=1}^{n} X_i \to \mathbf{E}[X_1] \tag{25.3}$$

For complicated historical reasons, the corresponding result for time series is called the **ergodic theorem**[1]. The most general and powerful versions of it are quite formidable, and have very subtle proofs, but there is a simple version which gives the flavor of them all, and is often useful enough.

25.2.2.1 The World's Simplest Ergodic Theorem

Suppose X_t is weakly stationary, and that

$$\sum_{h=0}^{\infty} |\gamma(h)| = \gamma(0)\tau < \infty \tag{25.4}$$

(Remember that $\gamma(0) = \mathrm{Var}\left[X_t\right]$.) The quantity τ is called the **correlation time**, or **integrated autocorrelation time**.

Now consider the average of the first n observations,

$$\overline{X}_n = \frac{1}{n}\sum_{t=1}^{n} X_t \tag{25.5}$$

This **time average** is a random variable. Its expectation value is

$$\mathbf{E}\left[\overline{X}_n\right] = \frac{1}{n}\sum_{t=1}^{n} \mathbf{E}\left[X_t\right] = \mathbf{E}[X_1] \tag{25.6}$$

[1] In the late 1800s, the physicist Ludwig Boltzmann needed a word to express the idea that if you took an isolated system at constant energy and let it run, any one trajectory, continued long enough, would be representative of the system as a whole. Being a highly-educated nineteenth century German-speaker, Boltzmann knew far too much ancient Greek, so he called this the "ergodic property", from *ergon* "energy, work" and *hodos* "way, path". The name stuck.

because the mean is stationary. What about its variance?

$$\begin{aligned}
\mathrm{Var}\left[\overline{X}_n\right] &= \mathrm{Var}\left[\frac{1}{n}\sum_{t=1}^{n} X_t\right] && (25.7)\\
&= \frac{1}{n^2}\left[\sum_{t=1}^{n}\mathrm{Var}\left[X_t\right] + 2\sum_{t=1}^{n}\sum_{s=t+1}^{n}\mathrm{Cov}\left[X_t, X_s\right]\right] && (25.8)\\
&= \frac{1}{n^2}\left[n\mathrm{Var}\left[X_1\right] + 2\sum_{t=1}^{n}\sum_{s=t+1}^{n}\gamma(s-t)\right] && (25.9)\\
&\leq \frac{1}{n^2}\left[n\gamma(0) + 2\sum_{t=1}^{n}\sum_{s=t+1}^{n}|\gamma(s-t)|\right] && (25.10)\\
&\leq \frac{1}{n^2}\left[n\gamma(0) + 2\sum_{t=1}^{n}\sum_{h=1}^{n}|\gamma(h)|\right] && (25.11)\\
&\leq \frac{1}{n^2}\left[n\gamma(0) + 2\sum_{t=1}^{n}\sum_{h=1}^{\infty}|\gamma(h)|\right] && (25.12)\\
&= \frac{n\gamma(0)(1+2\tau)}{n^2} && (25.13)\\
&= \frac{\gamma(0)(1+2\tau)}{n} && (25.14)
\end{aligned}$$

Eq. 25.9 uses stationarity again, and then Eq. 25.13 uses the assumption that the correlation time τ is finite.

Since $\mathbf{E}\left[\overline{X_n}\right] = \mathbf{E}\left[X_1\right]$, and $\mathrm{Var}\left[\overline{X}_n\right] \to 0$, we have that $\overline{X_n} \to \mathbf{E}\left[X_1\right]$, exactly as in the IID case. ("Time averages converge on expected values.") In fact, we can say a bit more. Remember Chebyshev's inequality: for any random variable Z,

$$\Pr(|Z - \mathbf{E}\left[Z\right]| > \epsilon) \leq \frac{\mathrm{Var}\left[Z\right]}{\epsilon^2} \tag{25.15}$$

so

$$\Pr\left(|\overline{X}_n - \mathbf{E}\left[X_1\right]| > \epsilon\right) \leq \frac{\gamma(0)(1+2\tau)}{n\epsilon^2} \tag{25.16}$$

which goes to zero as n grows for any given ϵ.

You may wonder whether the condition that $\sum_{h=0}^{\infty}|\gamma(h)| < \infty$ is as weak as possible. It turns out that it can in fact be weakened to just $\lim_{n\to\infty}\frac{1}{n}\sum_{h=0}^{n}\gamma(h) = 0$, as indeed the proof above might suggest.

25.2.2.2 Rate of Convergence

If the X_t were all IID, or even just uncorrelated, we would have $\mathrm{Var}\left[\overline{X}_n\right] = \gamma(0)/n$ exactly. Our bound on the variance is larger by a factor of $(1+2\tau)$, which reflects the influence of the correlations. Said another way, we can more or less pretend

that instead of having n correlated data points, we have $n/(1+2\tau)$ independent data points, that $n/(1+2\tau)$ is our **effective sample size**[2]

Generally speaking, dependence between observations reduces the effective sample size, and the stronger the dependence, the greater the reduction. (For an extreme, consider the situation where X_1 is randomly drawn, but thereafter $X_{t+1} = X_t$.) In more complicated situations, finding the effective sample size is itself a tricky undertaking, but it's often got this general flavor.

25.2.2.3 Why Ergodicity Matters

The ergodic theorem is important, because it tells us that a single long time series becomes representative of the whole data-generating process, just the same way that a large IID sample becomes representative of the whole population or distribution. We can therefore actually learn about the process from empirical data.

Strictly speaking, we have established that time-averages converge on expectations only for X_t itself, not even for $f(X_t)$ where the function f is non-linear. It might be that $f(X_t)$ doesn't have a finite correlation time even though X_t does, or indeed vice versa. This is annoying; we don't want to have to go through the analysis of the last section for every different function we might want to calculate.

When people say that the whole process is **ergodic**, they roughly speaking mean that

$$\frac{1}{n}\sum_{t=1}^{n} f(X_t^{t+k-1}) \rightarrow \mathbf{E}\left[f(X_1^k)\right] \tag{25.17}$$

for any reasonable function f. This is (again very roughly) equivalent to

$$\frac{1}{n}\sum_{t=1}^{n} \Pr\left(X_1^k \in A, X_t^{t+l-1} \in B\right) \rightarrow \Pr\left(X_1^k \in A\right)\Pr\left(X_1^l \in B\right) \tag{25.18}$$

which is a kind of asymptotic independence-on-average[3]

If our data source is ergodic, then what Eq. 25.17 tells us is that sample averages of any reasonable function are representative of expectation values, which is what we need to be in business statistically. This in turn is basically implied by stationarity.[4] What does this let us do?

[2]Some people like to define the correlation time as, in this notation, $1+2\tau$ for just this reason.

[3]It's worth sketching a less rough statement. Instead of working with X_t, work with the whole future trajectory $Y_t = (X_t, X_{t+1}, X_{t+2}, \ldots)$. Now the dynamics, the rule which moves us into the future, can be summed up in a very simple, and deterministic, operation T: $Y_{t+1} = TY_t = (X_{t+1}, X_{t+2}, X_{t+3}, \ldots)$. A set of trajectories is **invariant** if it is left unchanged by T: for every $y \in A$, there is another y' in A where $Ty' = y$. A process is **ergodic** if every invariant set either has probability 0 or probability 1. What this means is that (almost) all trajectories generated by an ergodic process belong to a single invariant set, and they all wander from every part of that set to every other part — they are **metrically transitive**. (Because: no smaller set with any probability is invariant.) Metric transitivity, in turn, is equivalent, assuming stationarity, to $n^{-1}\sum_{t=0}^{n-1}\Pr(Y \in A, T^t Y \in B) \rightarrow \Pr(Y \in A)\Pr(Y \in B)$. From metric transitivity follows Birkhoff's "individual" ergodic theorem, that $n^{-1}\sum_{t=0}^{n-1} f(T^t Y) \rightarrow \mathbf{E}[f(Y)]$, with probability 1. Since a function of the trajectory can be a function of a block of length k, we get Eq. 25.17.

[4]Again, a sketch of a less rough statement. Use Y again for whole trajectories. Every stationary distribution for Y can be written as a mixture of stationary and ergodic distributions, rather as we wrote complicated distributions as mixtures of simple Gaussians in Chapter 19. (This is called the "ergodic

25.3 Markov Models

For this section, we'll assume that X_t comes from a stationary, ergodic time series. So for any reasonable function f, the time-average of $f(X_t)$ converges on $\mathbf{E}[f(X_1)]$. Among the "reasonable" functions are the indicators, so

$$\frac{1}{n}\sum_{t=1}^{n}\mathbf{1}_A(X_t) \to \Pr(X_1 \in A) \tag{25.19}$$

Since this also applies to functions of blocks,

$$\frac{1}{n}\sum_{t=1}^{n}\mathbf{1}_{A,B}(X_t, X_{t+1}) \to \Pr(X_1 \in A, X_2 \in B) \tag{25.20}$$

and so on. If we can learn joint and marginal probabilities, and we remember how to divide, then we can learn conditional probabilities.

It turns out that pretty much any density estimation method which works for IID data will also work for getting the marginal and conditional distributions of time series (though, again, the effective sample size depends on how quickly dependence decays). So if we want to know $p(x_t)$, or $p(x_{t+1}|x_t)$, we can estimate it just as we learned how to do in Chapter 15.

Now, the conditional pdf $p(x_{t+1}|x_t)$ always exists, and we can always estimate it. But why stop just one step back into the past? Why not look at $p(x_{t+1}|x_t, x_{t-1})$, or for that matter $p(x_{t+1}|x_{t-999}^t)$? There are three reasons, in decreasing order of pragmatism.

- Estimating $p(x_{t+1}|x_{t-999}^t)$ means estimating a thousand-and-one-dimensional distribution. The curse of dimensionality will crush us.
- Because of the decay of dependence, there shouldn't be much difference, much of the time, between $p(x_{t+1}|x_{t-999}^t)$ and $p(x_{t+1}|x_{t-998}^t)$, etc. Even if we could go very far back into the past, it shouldn't, usually, change our predictions very much.
- Sometimes, a finite, short block of the past completely screens off the remote past.

You will remember the Markov property from your previous probability classes:

$$X_{t+1} \perp\!\!\!\perp X_1^{t-1} | X_t \tag{25.21}$$

decomposition" of the process.) We can think of this as first picking an ergodic process according to some fixed distribution, and then generating Y from that process. Time averages computed along any one trajectory thus converge according to Eq. 25.17. If we have only a single trajectory, it looks just like a stationary and ergodic process. If we have multiple trajectories from the same source, each one may be converging to a different ergodic component. It is common, and only rarely a problem, to assume that the data source is not only stationary but also ergodic.

When the Markov property holds, there is simply no point in looking at $p(x_{t+1}|x_t, x_{t-1})$, because it's got to be just the same as $p(x_{t+1}|x_t)$. If the process isn't a simple Markov chain but has a higher-order Markov property,

$$X_{t+1} \perp\!\!\!\perp X_1^{t-k} | X_{t-k+1}^t \tag{25.22}$$

then we never have to condition on more than the last k steps to learn all that there is to know. The Markov property means that the current state screens off the future from the past.

It is *always* an option to model X_t as a Markov process, or a higher-order Markov process. If it isn't exactly Markov, if there's really some dependence between the past and the future even given the current state, then we're introducing some bias, but it can be small, and dominated by the reduced variance of not having to worry about higher-order dependencies.

25.3.1 Meaning of the Markov Property

The Markov property is a weakening of both being strictly IID and being strictly deterministic.

That being Markov is weaker than being IID is obvious: an IID sequence satisfies the Markov property, because everything is independent of everything else no matter what we condition on.

In a deterministic dynamical system, on the other hand, we have $X_{t+1} = g(X_t)$ for some fixed function g. Iterating this equation, the current state X_t fixes the whole future trajectory $X_{t+1}, X_{t+2}, \ldots$. In a Markov chain, we weaken this to $X_{t+1} = g(X_t, U_t)$, where the U_t are IID noise variables (which we can take to be uniform for simplicity). The current state of a Markov chain doesn't fix the exact future trajectory, but it does fix the *distribution* over trajectories.

The real meaning of the Markov property, then, is about information flow: the current state is the only channel through which the past can affect the future.

t	x
1821	269
1822	321
1823	585
1824	871
1825	1475
1826	2821
1827	3928
1828	5943
1829	4950
...	

$\Rightarrow$

lag0	lag1	lag2	lag3
871	585	321	269
1475	871	585	321
2821	1475	871	585
3928	2821	1475	871
5943	3928	2821	1475
4950	5943	3928	2821
...			

Figure 25.4: Turning a time series (here, the beginning of `lynx`) into a regression-suitable matrix.

```
design.matrix.from.ts <- function(ts,order) {
  n <- length(ts)
  x <- ts[(order+1):n]
  for (lag in 1:order) {
    x <- cbind(x,ts[(order+1-lag):(n-lag)])
  }
  colnames(x) <- c("lag0",paste("lag",1:order,sep=""))
  return(as.data.frame(x))
}
```

Code Example 32: Example code for turning a time series into a design matrix, suitable for regression.

25.4 Autoregressive Models

Instead of trying to estimate the whole conditional distribution of X_t, we can just look at its conditional expectation. This is a regression problem, but since we are regressing X_t on earlier values of the series, it's called an **autoregression**:

$$\mathbf{E}\left[X_t | X_{t-p}^{t-1} = x_1^p\right] = r(x_1^p) \tag{25.23}$$

If we think the process is Markov of order p, then of course there is no point in conditioning on more than p steps of the past when doing an autoregression. But even if we don't think the process is Markov, the same reasons which inclined us towards Markov approximations also make limited-order autoregressions attractive.

Since this is a regression problem, we can employ all the tools we know for regression analysis: linear models, kernel regression, spline smoothing, additive models, etc., mixtures of regressions, etc. Since we are regressing X_t on earlier values from the same series, it is useful to have tools for turning a time series into a regression-style design matrix (as in Figure 25.4); see Code Example 32.

Suppose $p = 1$. Then we essentially want to draw regression curves through plots like those in Figure 25.3. Figure 25.5 shows an example for the artificial series.

25.4.1 Autoregressions with Covariates

Nothing keeps us from adding a variable other than the past of X_t to the regression:

$$\mathbf{E}\left[X_{t+1}|X_{t-k+1}^{t}, Z\right] \tag{25.24}$$

or even another time series:

$$\mathbf{E}\left[X_{t+1}|X_{t-k+1}^{t}, Z_{t-l+1}^{t}\right] \tag{25.25}$$

These are perfectly well-defined conditional expectations, and quite estimable in principle. Of course, adding more variables to a regression means having to estimate more, so again the curse of dimensionality comes up, but our methods are very much the same as in the basic regression analyses.

25.4.2 Additive Autoregressions

As before, if we want some of the flexibility of non-parametric smoothing, without the curse of dimensionality, we can try to approximate the conditional expectation as an additive function:

$$\mathbf{E}\left[X_t|X_{t-p}^{t-1}\right] \approx \alpha_0 + \sum_{j=1}^{p} g_j(X_{t-j}) \tag{25.26}$$

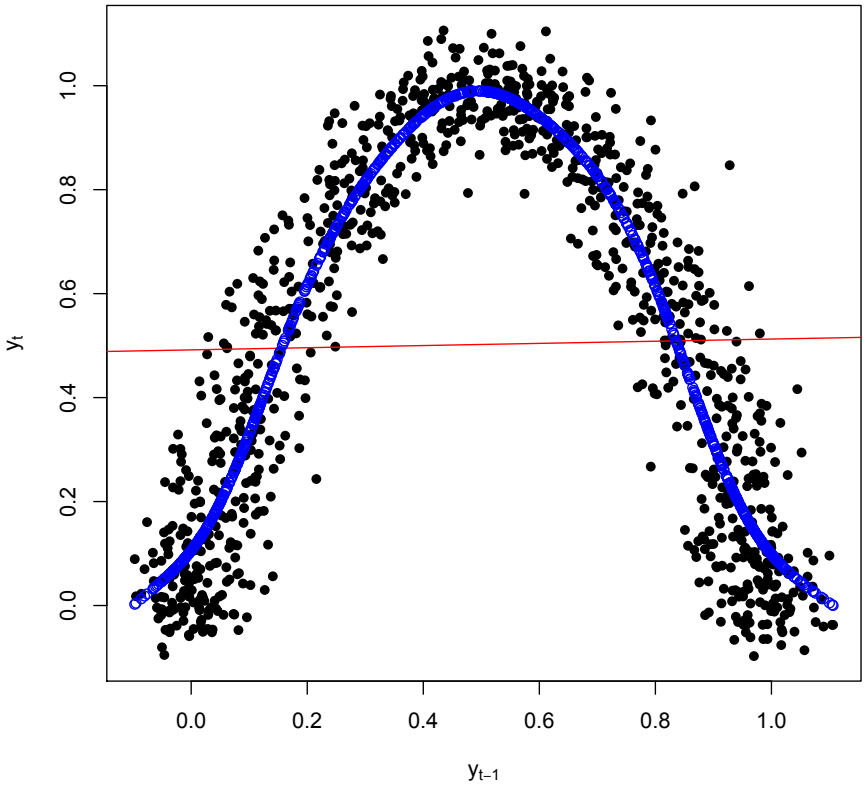
25.4.2.0.1 Example: The lynx Let's try fitting an additive model for the lynx. Code Example 33 shows some code for doing this. (Most of the work is re-shaping the time series into a data frame, and then automatically generating the right formula for `gam`.) Let's try out $p = 2$.

```
lynx.aar2 <- aar(lynx,2)
```

This inherits everything we can do with a GAM, so we can do things like plot the partial response functions (Figure 25.6), plot the fitted values against the actual (Figure 25.7), etc. To get a sense of how well it can actually extrapolate, Figure 25.8 re-fits the model to just the first 80 data points, and then predicts the remaining 34.

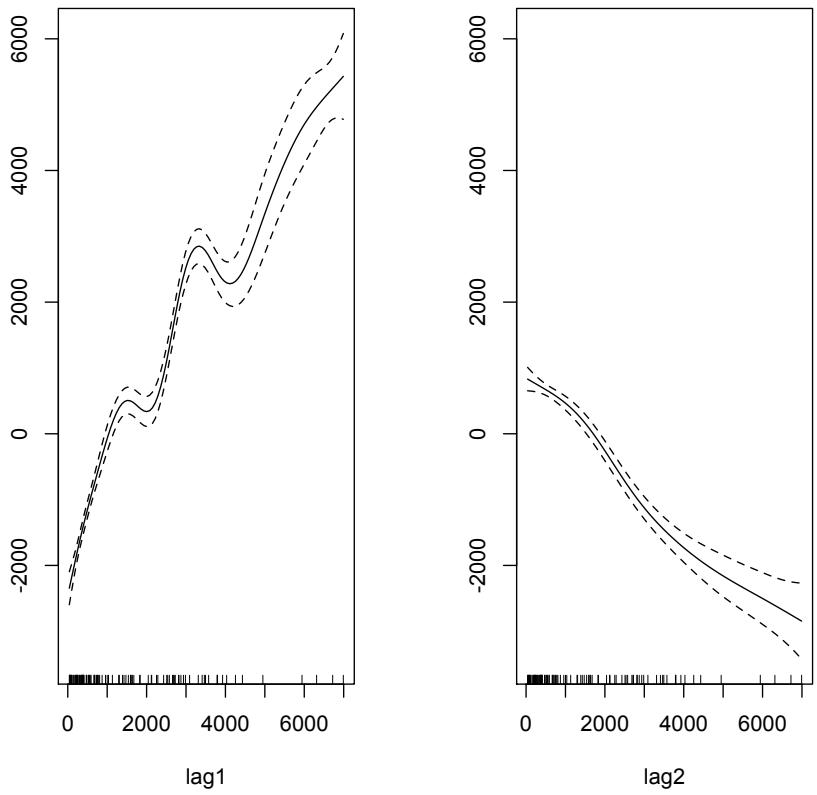
25.4.3 Linear Autoregression

When people talk about autoregressive models, they usually (alas) just mean linear autoregressions. There is almost never any justification in scientific theory for this preference, but we can always ask for the best linear approximation to the true autoregression, if only because it's fast to compute and fast to converge.



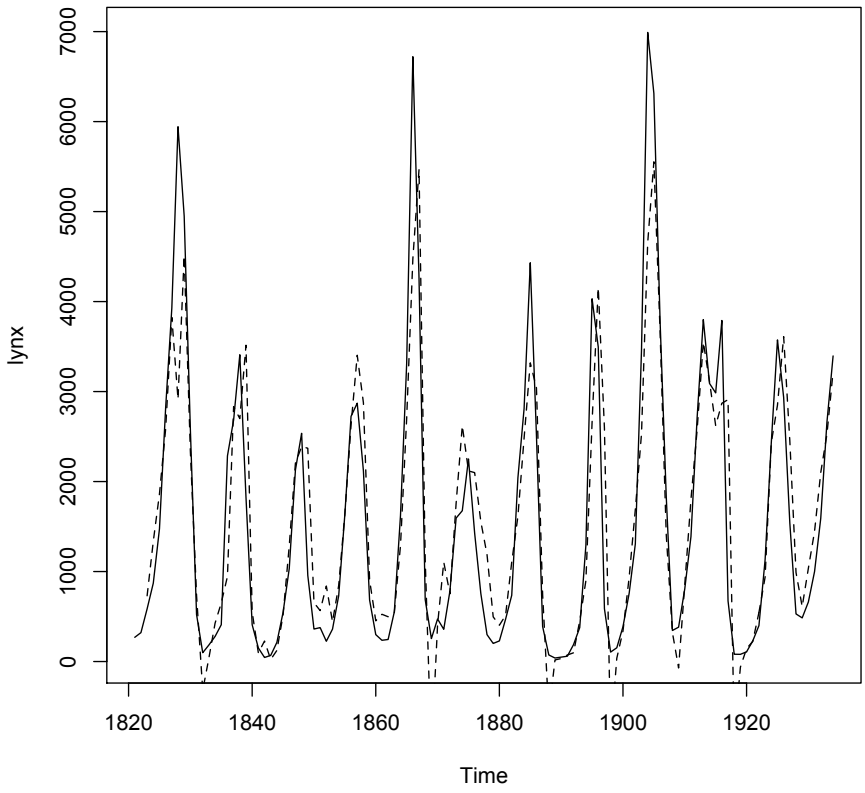

```
plot(lag0 ~ lag1,data=design.matrix.from.ts(y,1),xlab=expression(y[t-1]),
  ylab=expression(y[t]),pch=16)
abline(lm(lag0~lag1,data=design.matrix.from.ts(y,1)),col="red")
yaar1 <- aar(y,order=1)
points(y[-length(y)],fitted(yaar1),col="blue")
```

Figure 25.5: Plotting successive values of the artificial time series against each other, along with the linear regression, and a spline curve (see below for the `aar` function, which fits additive autoregressive models; with `order=1`, it just fits a spline.



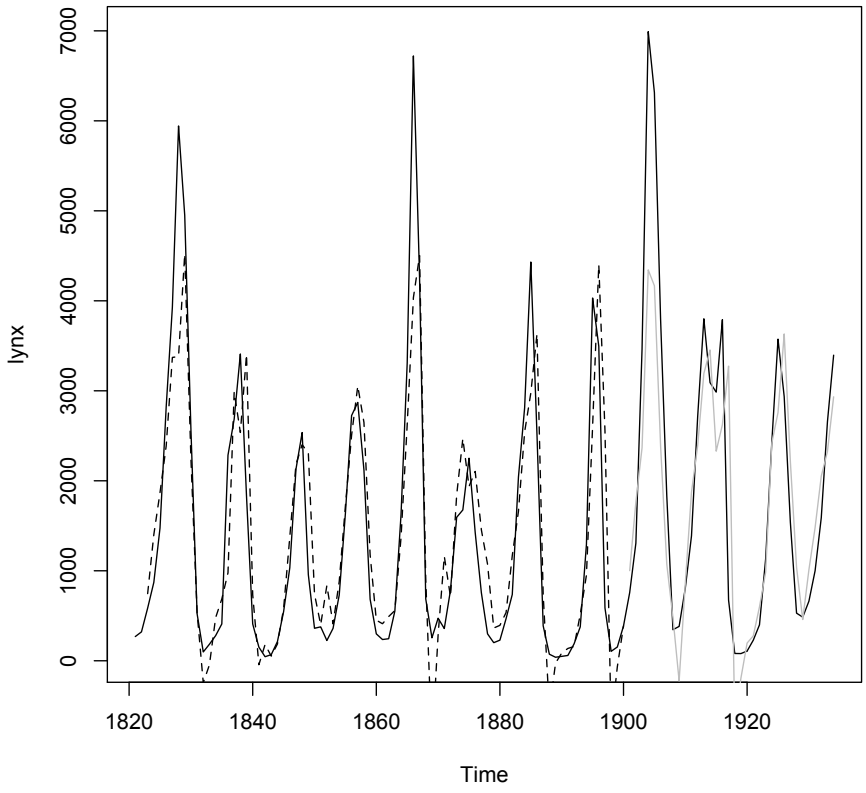

```
plot(lynx.aar2,pages=1)
```

Figure 25.6: Partial response functions for the second-order additive autoregression model of the lynx. Notice that a high count last year predicts a higher count this year, but a high count *two* years ago predicts a lower count this year. This is the sort of alternation which will tend to drive oscillations.




```
plot(lynx)
lines(1823:1934,fitted(lynx.aar2),lty="dashed")
```

Figure 25.7: Actual time series (solid line) and predicted values (dashed) for the second-order additive autoregression model of the lynx. The match is quite good, but of course every one of these points was used to learn the model, so it's not quite as impressive as all that. (Also, the occasional prediction of a negative number of lynxes is less than ideal.)




```
lynx.aar2b <- aar(lynx[1:80],2)
out.of.sample <- design.matrix.from.ts(lynx[-(1:78)],2)
lynx.preds <- predict(lynx.aar2b,newdata=out.of.sample)
plot(lynx)
lines(1823:1900,fitted(lynx.aar2b),lty="dashed")
lines(1901:1934,lynx.preds,col="grey")
```

Figure 25.8: Out-of-sample forecasting. The same model specification as before is estimated on the first 80 years of the lynx data, then used to predict the remaining 34 years. Solid black line, data; dashed line, the in-sample prediction on the training data; grey lines, predictions on the testing data. The RMS errors are 723 lynxes/year in-sample, 922 lynxes/year out-of-sample.

```
aar <- function(ts,order) {
  stopifnot(require(mgcv))
  fit <- gam(as.formula(auto.formula(order)),
    data=design.matrix.from.ts(ts,order))
  return(fit)
}

auto.formula <- function(order) {
  inputs <- paste("s(lag",1:order,")",sep="",collapse="+")
    form <- paste("lag0 ~ ",inputs)
    return(form)
}
```

Code Example 33: Fitting an additive autoregression of arbitrary order to a time series. See online for comments.

The analysis we did in Chapter 2 of how to find the optimal linear predictor carries over with no change whatsoever. If we want to predict X_t as a linear combination of the last k observations, $X_{t-1}, X_{t-2}, \ldots X_{t-p}$, then the ideal coefficients β are

$$\beta = \left(\mathrm{Var}\left[X_{t-p}^{t-1}\right]\right)^{-1} \mathrm{Cov}\left[X_{t-p}^{t-1}, X_t\right] \tag{25.27}$$

where $\mathrm{Var}\left[X_{t-p}^{t}\right]$ is the variance-covariance matrix of $(X_{t-1}, \ldots X_{t-p})$ and similarly $\mathrm{Cov}\left[X_{t-p}^{t-1}, X_t\right]$ is a vector of covariances. Assume stationarity, $\mathrm{Var}\left[X_t\right]$ is constant in t, and so the common factor of the over-all variance goes away, and β could be written entirely in terms of the correlation function ρ. Stationarity also lets us estimate these covariances, by taking time-averages.

A huge amount of effort is given over to using linear AR models, which in my opinion is out of all proportion to their utility — but very reflective of what was computationally feasible up to about 1980. My experience is that results like Figure 25.9 is pretty typical.

25.4.3.1 "Unit Roots" and Stationary Solutions

Suppose we really believed a first-order linear autoregression,

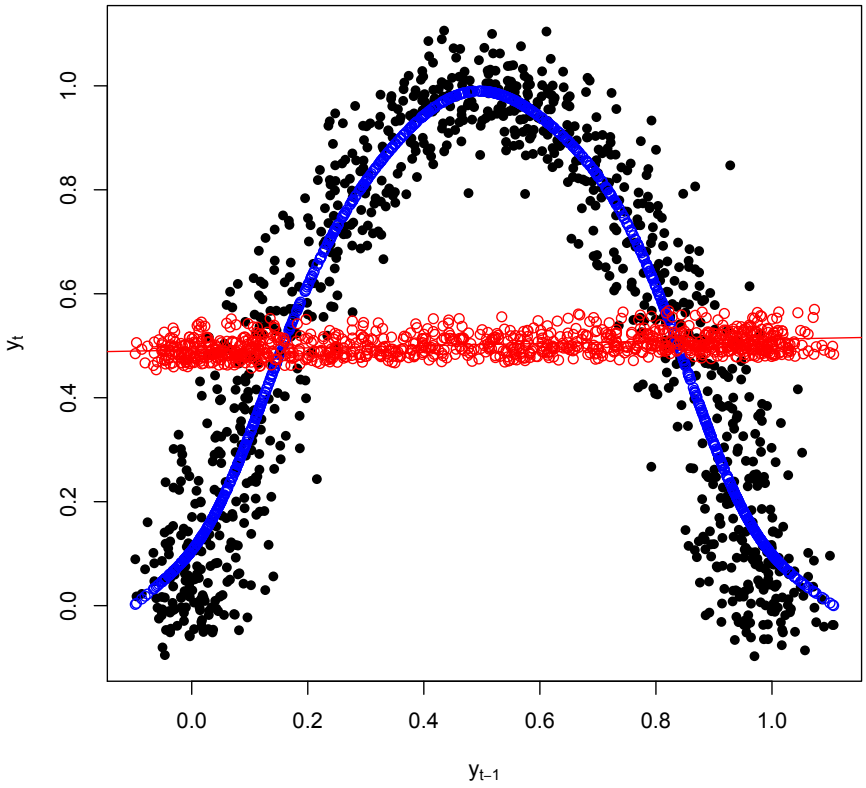
$$X_{t+1} = \alpha + \beta X_t + \epsilon_t \tag{25.28}$$

with ϵ_t some IID noise sequence. Let's suppose that the mean is zero for simplicity, so $\alpha = 0$. Then

$$X_{t+2} = \beta^2 X_t + \beta \epsilon_t + \epsilon_{t+1} \tag{25.29}$$

$$X_{t+3} = \beta^3 X_t + \beta^2 \epsilon_t + \beta \epsilon_{t+1} + \epsilon_{t+2}\,, \tag{25.30}$$

etc. If this is going to be stationary, it'd better be the case that what happened at time t doesn't go on to dominate what happens at all later times, but clearly that




```
library(tseries)
yar8 <- arma(y,order=c(8,0))
points(y[-length(y)],fitted(yar8)[-1],col="red")
```

Figure 25.9: Adding the predictions of an eighth-order linear AR model (red dots) to Figure 25.5. We will see the `arma` function in more detail next time; for now, it's enough to know that when the second component of its `order` argument is 0, it estimates and fits a linear AR model.

will happen if $|\beta| > 1$, whereas if $|\beta| < 1$, eventually all memory of X_t (and ϵ_t) fades away. The linear AR(1) model in fact can only produce stationary distributions when $|\beta| < 1$.

For higher-order linear AR models, with parameters $\beta_1, \beta_2, \ldots \beta_p$, the corresponding condition is that all the roots of the polynomial

$$\sum_{j=1}^{p} \beta_j z^j - 1 \tag{25.31}$$

must be outside the unit circle. When this fails, when there is a "unit root", the linear AR model cannot generate a stationary process.

There is a fairly elaborate machinery for testing for unit roots, which is sometimes also used to test for non-stationarity. It is not clear how much this really matters. A non-stationary but truly linear AR model can certainly be estimated[5]; a linear AR model can be non-stationary even if it has no unit roots[6]; and if the linear model is just an approximation to a non-linear one, the unit-root criterion doesn't apply to the true model anyway.

25.4.4 Conditional Variance

Having estimated the conditional expectation, we can estimate the conditional variance $\mathrm{Var}\left[X_t | X_{t-p}^{t-1}\right]$ just as we estimated other conditional variances, in Chapter 7.

25.4.4.0.1 Example: lynx The lynx series seems ripe for fitting conditional variance functions — presumably when there are a few thousand lynxes, the noise is going to be larger than when there are only a few hundred.

```
sq.res <- residuals(lynx.aar2)^2
lynx.condvar1 <- gam(sq.res ~ s(lynx[-(1:2)]))
lynx.condvar2 <- gam(sq.res ~ s(lag1)+s(lag2),
  data=design.matrix.from.ts(lynx,2))
```

I have fit two different models for the conditional variance here, just because. Figure 25.10 shows the data, and the predictions of the second-order additive AR model, but with just the standard deviation bands corresponding to the first of these two models; you can try making the analogous plot for `lynx.condvar2`.

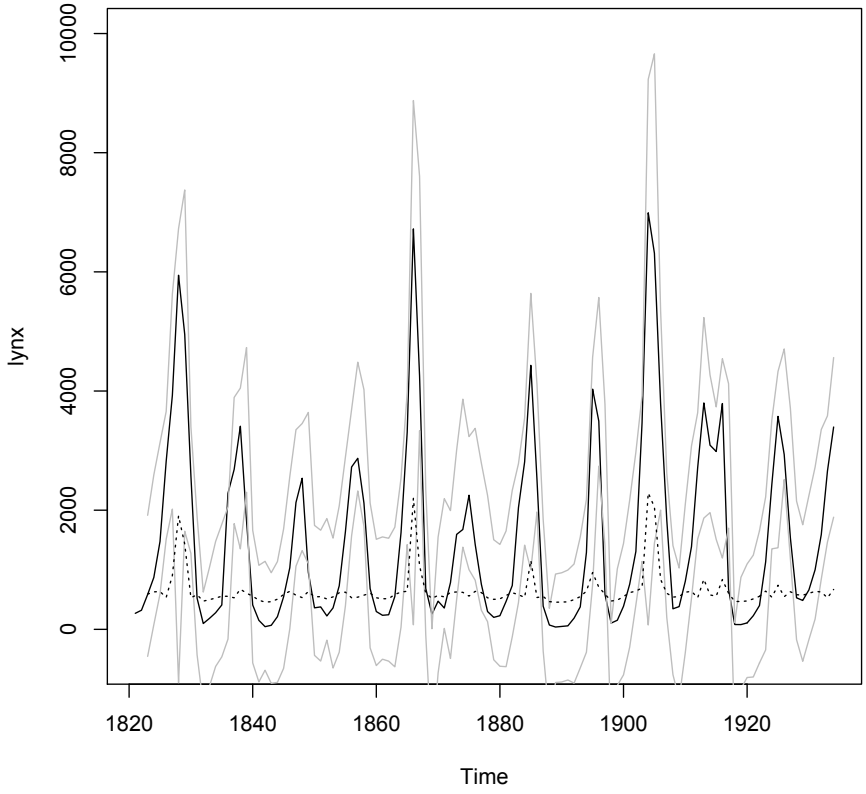
25.4.5 Regression with Correlated Noise; Generalized Least Squares

Suppose we have an old-fashioned regression problem

$$Y_t = r(X_t) + \epsilon_t \tag{25.32}$$

[5] Because the correlation structure stays the same, even as the means and variances can change. Consider $X_t = X_{t-1} + \epsilon_t$, with ϵ_t IID.

[6] Start it with X_1 very far from the expected value.




```
plot(lynx,ylim=c(-500,10000))
sd1 <- sqrt(fitted(lynx.condvar1))
lines(1823:1934,fitted(lynx.aar2)+2*sd1,col="grey")
lines(1823:1934,fitted(lynx.aar2)-2*sd1,col="grey")
lines(1823:1934,sd1,lty="dotted")
```

Figure 25.10: The lynx data (black line), together with the predictions of the additive autoregression ± 2 conditional standard deviations. The dotted line shows how the conditional standard deviation changes over time; notice how it ticks upwards around the big spikes in population.

only now the noise terms ϵ_t are themselves a dependent time series. Ignoring this dependence, and trying to estimate m by minimizing the mean squared error, is very much like ignoring heteroskedasticity. (In fact, heteroskedastic ϵ_t are a special case.) What we saw in Chapter 7 is that ignoring heteroskedasticity doesn't lead to bias, but it does mess up our understanding of the uncertainty of our estimates, and is generally inefficient. The solution was to weight observations, with weights inversely proportional to the variance of the noise.

With correlated noise, we do something very similar. Suppose we knew the covariance function $\gamma(h)$ of the noise. From this , we could construct the variance-covariance matrix $\mathbf{\Gamma}$ of the ϵ_t (since $\Gamma_{ij} = \gamma(i-j)$, of course).

We can use this as follows. Say that our guess about the regression function is m. Stacking $y_1, y_2, \ldots y_n$ into a matrix $\mathbf{y}$ as usual in regression, and likewise creating $\mathbf{m}(x)$, the Gauss-Markov theorem (Appendix D) tells us that the most efficient estimate is the solution to the **generalized least squares** problem,

$$\widehat{m}_{GLS} = \operatorname*{argmin}_m \frac{1}{n}(\mathbf{y} - \mathbf{m}(\mathbf{x}))^T \Gamma^{-1} (\mathbf{y} - \mathbf{m}(\mathbf{x})) \tag{25.33}$$

as opposed to just minimizing the mean-squared error,

$$\widehat{m}_{OLS} = \operatorname*{argmin}_m \frac{1}{n}(\mathbf{y} - \mathbf{m}(\mathbf{x}))^T (\mathbf{y} - \mathbf{m}(\mathbf{x})) \tag{25.34}$$

Multiplying by the inverse of Γ appropriately discounts for observations which are very noisy, *and* discounts for correlations between observations introduced by the noise.[7]

This raises the question of how to get $\gamma(h)$ in the first place. If we knew the true regression function r, we could use the covariance of $Y_t - r(X_t)$ across different t. Since we don't know r, but have only an estimate $\hat{m}$, we can try alternating between using a guess at γ to estimate $\hat{m}$, and using $\hat{m}$ to improve our guess at γ. We used this sort of iterative approximation for weighted least squares, and it can work here, too.

[7] If you want to use a linear model for m, this can be carried through to an explicit modification of the usual ordinary-least-squares estimate — Exercise 1.

25.5 Bootstrapping Time Series

The big picture of bootstrapping doesn't change: simulate a distribution which is close to the true one, repeat our estimate (or test or whatever) on the simulation, and then look at the distribution of this statistic over many simulations. The catch is that the surrogate data from the simulation has to have the same sort of dependence as the original time series. This means that simple resampling is just wrong (unless the data are independent), and our simulations will have to be more complicated.

25.5.1 Parametric or Model-Based Bootstrap

Conceptually, the simplest situation is when we fit a full, generative model — something which we could step through to generate a new time series. If we are confident in the model specification, then we can bootstrap by, in fact, simulating from the fitted model. This is the parametric bootstrap we saw in Chapter 6.

25.5.2 Block Bootstraps

Simple resampling won't work, because it destroys the dependence between successive values in the time series. There is, however, a clever trick which does work, and is almost as simple. Take the full time series x_1^n and divide it up into overlapping blocks of length k, so $x_1^k, x_2^{k+1}, \ldots x_{n-k+1}^n$. Now draw $m = n/k$ of these *blocks* with replacement[8], and set them down in order. Call the new time series $\tilde{x}_1^n$.

Within each block, we have preserved *all* of the dependence between observations. It's true that successive observations are now completely independent, which generally wasn't true of the original data, so we're introducing some inaccuracy, but we're certainly coming closer than just resampling individual observations (which would be $k = 1$). Moreover, we can make this inaccuracy smaller and smaller by letting k grow as n grows. One can show[9] that the optimal $k = O(n^{1/3})$; this gives a growing number ($O(n^{2/3})$) of increasingly long blocks, capturing more and more of the dependence. (We will consider how exactly to pick k in the next chapter.)

The block bootstrap scheme is extremely clever, and has led to a great many variants. Three in particular are worth mentioning.

1. In the **circular block bootstrap** (or **circular bootstrap**), we "wrap the time series around a circle", so that it goes $x_1, x_2, \ldots x_{n_1}, x_n, x_1, x_2, \ldots$. We then sample the n blocks of length k this gives us, rather than the merely $n - k$ blocks of the simple block bootstrap. This makes better use of the information we have about dependence on distances $< k$.

2. In the **block-of-blocks bootstrap**, we first divide the series into blocks of length k_2, and then subdivide each of those into sub-blocks of length $k_1 < k_2$. To generate a new series, we sample blocks with replacement, and then sample

[8] If n/k isn't a whole number, round.
[9] I.e., I will not show.

t	x
1821	269
1822	321
1823	585
1824	871
1825	1475
1826	2821
1827	3928
1828	5943

$\Rightarrow$

lag2	lag1	lag0
269	321	585
321	585	871
585	871	1475
871	1475	2821
1475	2821	3928
2821	3928	5943

$\Rightarrow$

lag2	lag1	lag0
269	321	585
871	1475	2821
585	871	1475

$\Rightarrow$

t	$\tilde{x}$
1821	269
1822	321
1823	585
1824	871
1825	1475
1826	2821
1827	585
1828	871

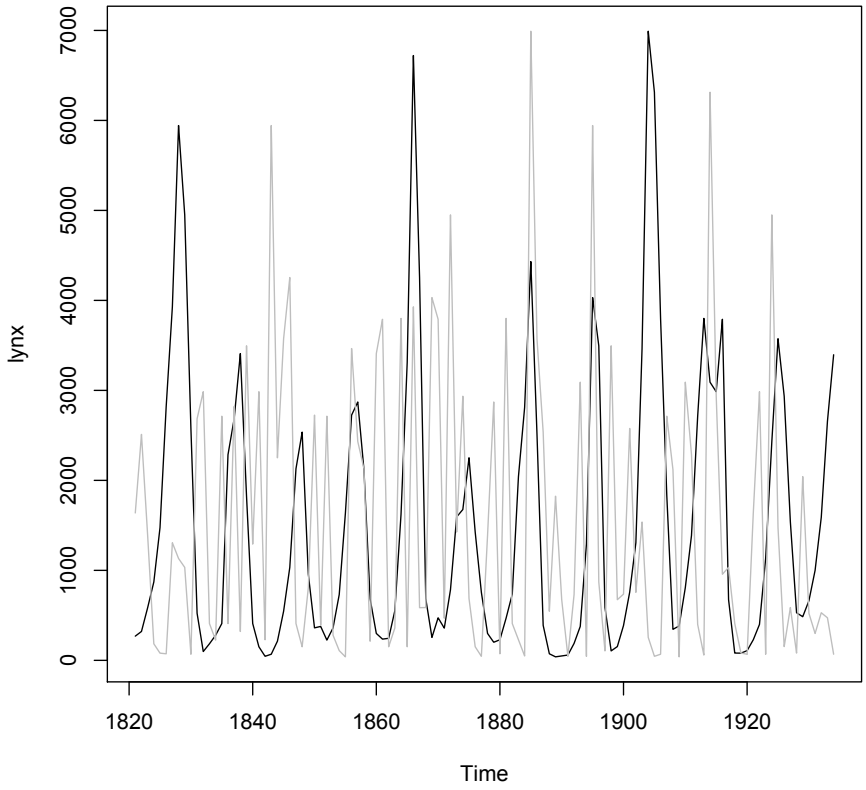
Figure 25.11: Scheme for block bootstrapping: turn the time series (here, the first eight years of `lynx`) into blocks of consecutive values; randomly resample enough of these blocks to get a series as long as the original; then string the blocks together in order. See `rblockboot` online for code.

sub-blocks within each block with replacement. This gives a somewhat better idea of longer-range dependence, though we have to pick two block-lengths.

3. In the **stationary bootstrap**, the length of each block is random, chosen from a geometric distribution of mean k. Once we have chosen a sequence of block lengths, we sample the appropriate blocks with replacement. The advantage of this is that the ordinary block bootstrap doesn't quite give us a stationary time series. (The distribution of X_{k-1}^{k} is not the same as the distribution of X_{k}^{k+1}.) Averaging over the random choices of block lengths, the stationary bootstrap does. It tends to be slightly slower to converge that the block or circular bootstrap, but there are some applications where the surrogate data really needs to be strictly stationary.

25.5.3 Sieve Bootstrap

A compromise between model-based and non-parametric bootstraps is to use a **sieve bootstrap**. This also simulates from models, but we don't really believe in them; rather, we just want them to be reasonable easy to fit and simulate, yet flexible enough that they can capture a wide range of processes if we just give them enough capacity.




```
plot(lynx)
lines(1821:1934, rblockboot(lynx,4),col="grey")
```

Figure 25.12: The lynx time series, and one run of resampling it with a block bootstrap, block length = 4. (See online for the code to `rblockboot`.)

We then (slowly) let them get more complicated as we get more data[10]. One popular choice is to use linear AR(p) models, and let p grow with n — but there is nothing special about linear AR models, other than that they are very easy to fit and simulate from. Additive autoregressive models, for instance, would often work at least as well.

25.6 Trends and De-Trending

The sad fact is that a lot of important real time series are not even approximately stationary. For instance, Figure 25.13 shows US national income per person (adjusted for inflation) over the period from 1952 (when the data begins) until now. It is *possible* that this is sample from a stationary process. But in that case, the correlation time is evidently much longer than 50 years, on the order of centuries, and so the theoretical stationarity is irrelevant for anyone but a very ambitious quantitative historian.

More sensibly, we should try to treat data like this as a non-stationary time series. The conventional approach is o try to separate time series like this into a persistent **trend**, and stationary **fluctuations** (or **deviations**) around the trend,

$$\begin{aligned} Y_t &= X_t + Z_t \\ \text{series} &= \text{fluctuations} + \text{trend} \end{aligned} \tag{25.35}$$

Since we could add or subtract a constant to each X_t without changing whether they are stationary, we'll stipulate that $\mathbf{E}\left[X_t\right] = 0$, so $\mathbf{E}\left[Y_t\right] = \mathbf{E}\left[Z_t\right]$. (In other situations, the decomposition might be multiplicative instead of additive, etc.) How might we find such a decomposition?

If we have multiple independent realizations $Y_{i,t}$ of the same process, say m of them, and they all have the same trend Z_t, then we can try to find the common trend by averaging the time series:

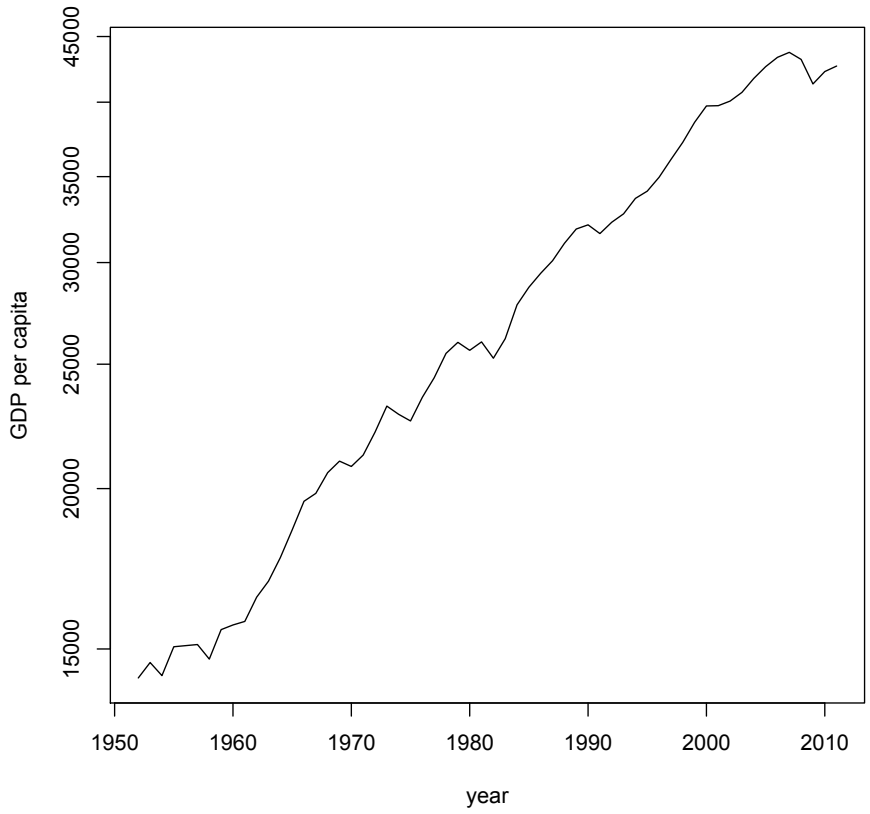
$$Z_t = \mathbf{E}\left[Y_{\cdot,t}\right] \approx \sum_{i=1}^{m} Y_{i,t} \tag{25.36}$$

Multiple time series with the same trend do exist, especially in the experimental sciences. $Y_{i,t}$ might be the measurement of some chemical in a reactor at time t in the i^{th} repetition of the experiment, and then it would make sense to average the $Y_{i,t}$ to get the common Z_t trend, the average trajectory of the chemical concentration. One can tell similar stories about experiments in biology or even psychology, though those are complicated by the tendency of animals to get tired and to learn[11].

For better or for worse, however, we have only one realization of the post-WWII US economy, so we can't average multiple runs of the experiment together. If we have a theoretical model of the trend, we can try to fit that model. For instance,

[10] This is where the metaphor of the "sieve" comes in: the idea is that the mesh of the sieve gets finer and finer, catching more and more subtle features of the data.

[11] Even if we do have multiple independent experimental runs, it is very important to get them aligned in time, so that $Y_{i,t}$ and $Y_{j,t}$ refer to the *same* point in time relative to the start of the experiment; otherwise, averaging them is just mush. It can also be important to ensure that the initial state, before the experiment, is the same for every run.




```
gdppc <- read.csv("gdp-pc.csv")
gdppc$y <- gdppc$y*1e6
plot(gdppc,log="y",type="l",ylab="GDP per capita")
```

Figure 25.13: US GDP per capita, constant dollars (consumer price index deflator). Note logarithmic scale of vertical axis. (The values were initially recorded in the file in millions of dollars per person per year, hence the correction.)

some (simple) models of economic growth predict that series like the one in Figure 25.13 should, on average, grow at a steady exponential rate[12]. We could then estimate Z_t by fitting a model to Y_t of the form $\beta_0 e^{\beta t}$, or even by doing a linear regression of $\log Y_t$ on t. The fluctuations X_t are then taken to be the residuals of this model.

If we only have one time series (no replicates), and we don't have a good theory which tells us what the trend should be, we fall back on curve fitting. In other words, we regress Y_t on t, call the fitted values Z_t, and call the residuals X_t. This is frankly rests more on hope than on theorems. The hope is that the characteristic time-scale for the fluctuations X_t (say, their correlation time τ) is short compared to the characteristic time-scale for the trend Z_t[13]. Then if we average Y_t over a band-width which is large compared to τ, but small compared to the scale of Z_t, we should get something which is mostly Z_t — there won't be too much bias from averaging, and the fluctuations should mostly cancel out.

Once we have the fluctuations, and are reasonably satisfied that they're stationary, we can model them like any other stationary time series. Of course, to actually make predictions, we need to extrapolate the trend, which is a harder business.

25.6.1 Forecasting Trends

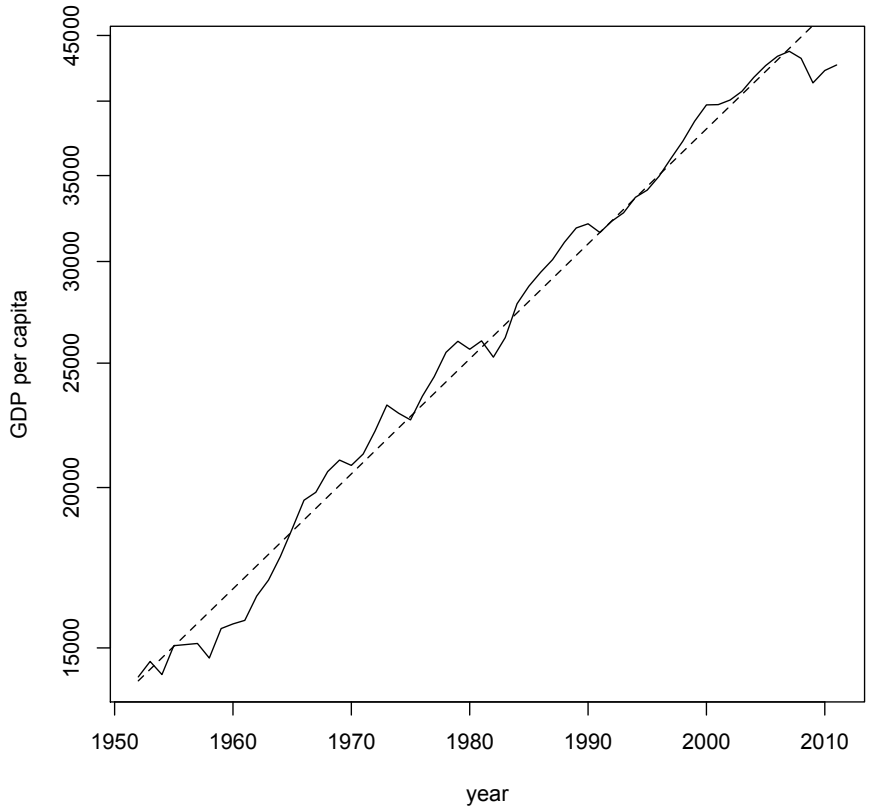
The problem with making predictions when there is a substantial trend is that it is usually hard to know how to continue or extrapolate the trend beyond the last data point. If we are in the situation where we have multiple runs of the same process, we can at least extrapolate up to the limits of the different runs. If we have an actual model which tells us that the trend should follow a certain functional form, and we've estimated that model, we can use it to extrapolate. But if we have found the trend purely through curve-fitting, we have a problem.

Suppose that we've found the trend by spline smoothing, as in Figure 25.16. The fitted spline model will cheerfully make predictions for the what the trend of GDP per capita will be in, say, 2252, far outside the data. This will be a simple linear extrapolation, because splines are always linear outside the data range (Chapter 8, p. 178). This is just because of the way splines are set up, not because linear extrapolation is such a good idea. Had we used kernel regression, or any of many other ways of fitting the curve, we'd get different extrapolations. People in 2252 could look back and see whether the spline had fit well, or some other curve would have done better. (But why would they want to?) Right now, if *all* we have is curve-fitting, we are in a dubious position even as regards 2013, never mind 2252[14]

[12] This is not *quite* what is claimed by Solow (1970), which you should read anyway if this kind of question is at all interesting to you.

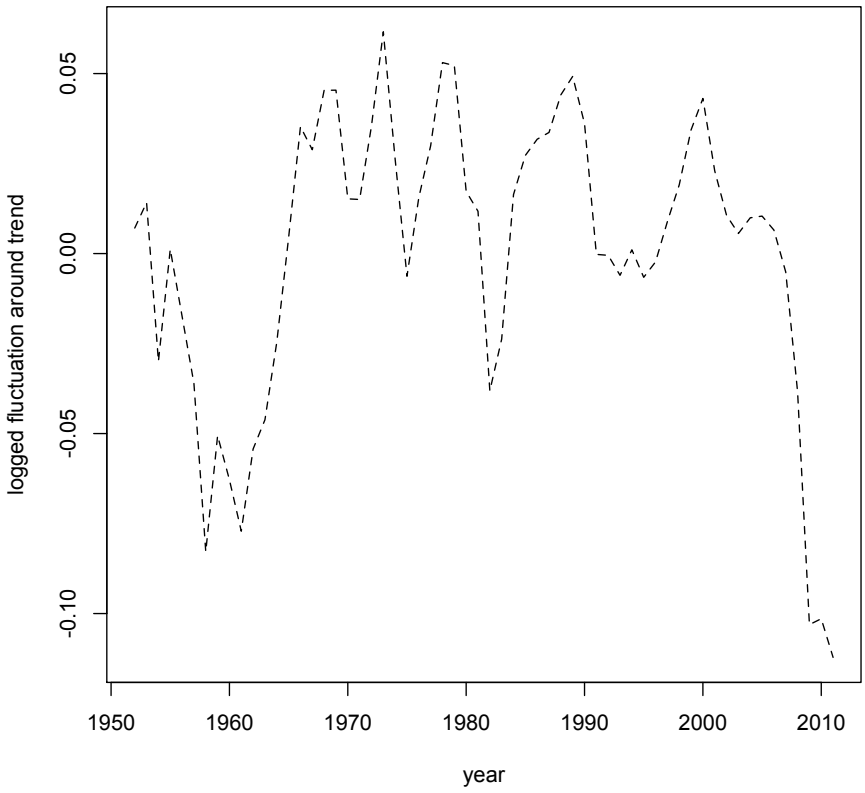
[13] I am being deliberately vague about what "the characteristic time scale of Z_t" means. Intuitively, it's the amount of time required for Z_t to change substantially. You might think of it as something like $n^{-1}\sum_{t=1}^{n-1} 1/|Z_{t+1} - Z_t|$, if you promise not to treat that too seriously. Trying to get an exact statement of what's involved in identifying trends requires being very precise, and getting into topics at the intersection of statistics and functional analysis which are beyond the scope of this class.

[14] Yet again, we hit a basic philosophical obstacle, which is the problem of induction. We have so far evaded it, by assuming that we're dealing with IID or a stationary probability distribution; these assumptions let us *deductively* extrapolate from past data to future observations, with more or less confidence. (For more on this line of thought, see Hacking (2001); Spanos (2011); Gelman and Shalizi (2013).) If we



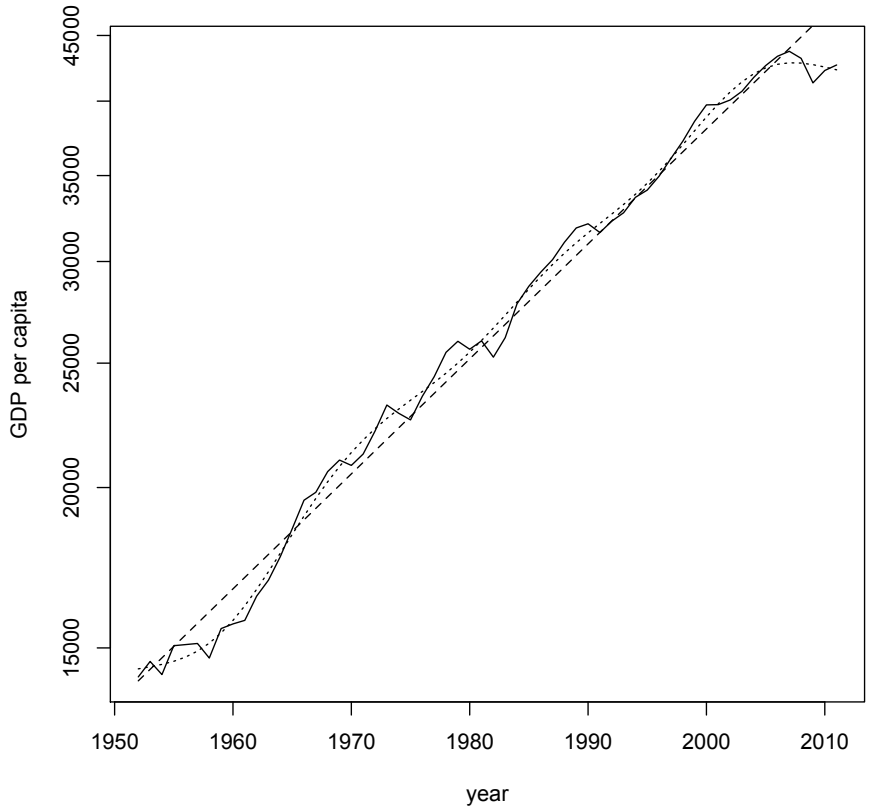

```
gdppc.exp <- lm(log(y) ~ year,data=gdppc)
beta0 <- exp(coefficients(gdppc.exp)[1])
beta <- coefficients(gdppc.exp)[2]
curve(beta0*exp(beta*x),lty="dashed",add=TRUE)
```

Figure 25.14: As in Figure 25.13, but with an exponential trend fitted.



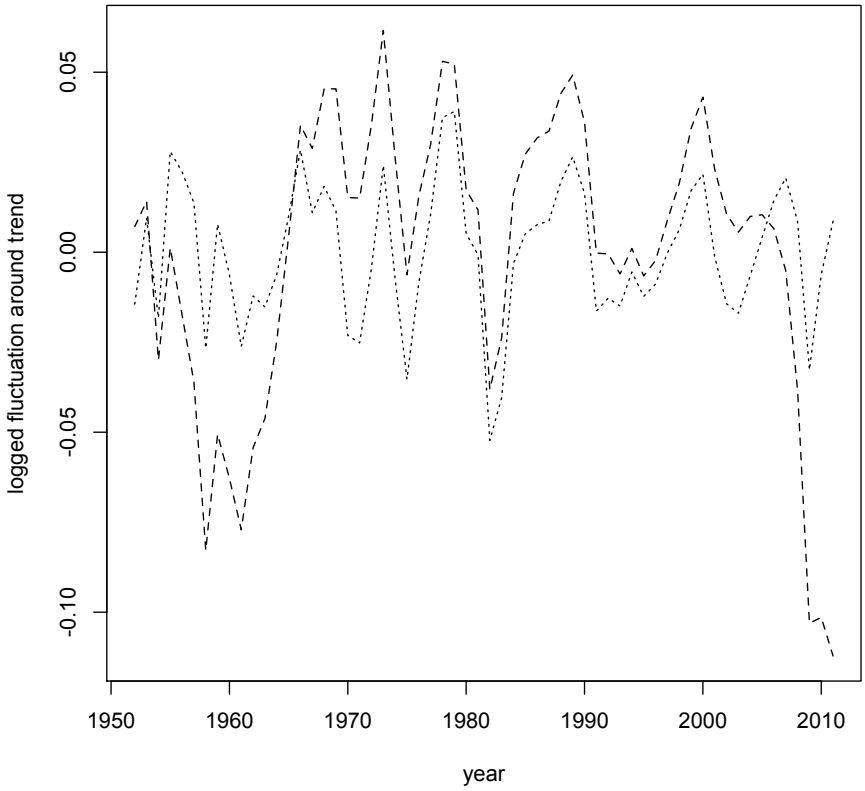

```
plot(gdppc$year,residuals(gdppc.exp),xlab="year",
  ylab="logged fluctuation around trend",type="l",lty="dashed")
```

Figure 25.15: The hopefully-stationary fluctuations around the exponential growth trend in Figure 25.14. Note that these are $\log \frac{Y_t}{\hat{\beta}_0 e^{\hat{\beta}_1 t}}$, and so unitless.




```
gdp.spline <- fitted(gam(y~s(year),data=gdppc))
lines(gdppc$year,gdp.spline,lty="dotted")
```

Figure 25.16: Figure 25.14, but with the addition of a spline curve for the time trend (dotted line). This is, perhaps unsurprisingly, not all *that* different from the simple exponential-growth trend.




```
lines(gdppc$year,log(gdppc$y/gdp.spline),xlab="year",
  ylab="logged fluctuations around trend",lty="dotted")
```

Figure 25.17: Adding the logged deviations from the spline trend (dotted) to Figure 25.15.

25.6.2 Seasonal Components

Sometimes we know that time series contain components which repeat, pretty exactly, over regular periods. These are called **seasonal** components, after the obvious example of trends which cycle each year with the season. But they could cycle over months, weeks, days, etc.

The decomposition of the process is thus

$$Y_t = X_t + Z_t + S_t \tag{25.37}$$

where X_t handles the stationary fluctuations, Z_t the long-term trends, and S_t the repeating seasonal component.

If $Z_t = 0$, or equivalently if we have a good estimate of it and can subtract it out, we can find S_t by averaging over multiple cycles of the seasonal trend. Suppose that we know the period of the cycle is T, and we can observe $m = n/T$ full cycles. Then

$$S_t \approx \frac{1}{m}\sum_{j=0}^{m-1} Y_{t+jT} \tag{25.38}$$

This works because, with Z_t out of the picture, $Y_t = X_t + S_t$, and S_t is periodic, $S_t = S_{t+T}$. Averaging over multiple cycles, the stationary fluctuations tend to cancel out (by the ergodic theorem), but the seasonal component does not.

For this trick to work, we need to know the period. If the true $T = 355$, but we use $T = 365$ without thinking[15], we can get mush.

We also need to know the over-all trend. Of course, if there are seasonal components, we really ought to subtract them out before trying to find Z_t. So we have yet another vicious cycle, or, more optimistically, another case for iterative approximation.

25.6.3 Detrending by Differencing

Suppose that Y_t has a linear time trend:

$$Y_t = \beta_0 + \beta t + X_t \tag{25.39}$$

with X_t stationary. Then if we take the difference between successive values of Y_t, the trend goes away:

$$Y_t - Y_{t-1} = \beta + X_t - X_{t-1} \tag{25.40}$$

Since X_t is stationary, $\beta + X_t - X_{t-1}$ is also stationary. Taking differences has removed the trend.

Differencing will not only get rid of linear time trends. Suppose that

$$Z_t = Z_{t-1} + \epsilon_t \tag{25.41}$$

assume a certain form or model for the trend, then again we can deduce future behavior on that basis. But if we have neither probabilistic nor mechanistic assumptions, we are, to use a technical term, stuck with induction. Whether there is some principle which might help — perhaps a form of Occam's Razor (Kelly, 2007)? — is a nice question.

[15] Exercise: come up with an example of a time series where the periodicity *should* be 355 days.

where the "innovations" or "shocks" ϵ_t are IID, and that

$$Y_t = Z_t + X_t \tag{25.42}$$

with X_t stationary, and independent of the ϵ_t. It is easy to check that (i) Z_t is not stationary (Exercise 2), but that (ii) the first difference

$$Y_t - Y_{t-1} = \epsilon_t + X_t - X_{t-1} \tag{25.43}$$

is stationary. So differencing can get rid of trends which are built out of the summation of *persistent* random shocks.

This gives us another way of making a time series stationary: instead of trying to model the time trend, take the difference between successive values, and see if that is stationary. (The `diff()` function in R does this; see Figure 25.18.) If such "first differences" don't look stationary, take differences among differences, third differences, etc., until you have something satisfying.

Notice that now we can continue to the trend: once we predict $Y_{t+1} - Y_t$, we add it on to Y_t (which we observed) to get Y_{t+1}.

Differencing is like taking the discrete version of a derivative. It will eventually get rid of trends if they correspond to curves (e.g., polynomials) with only finitely many non-zero derivatives. It fails for trends which aren't like that, like exponentials or sinusoids, though you can hope that eventually the higher differences are small enough that they don't matter much.

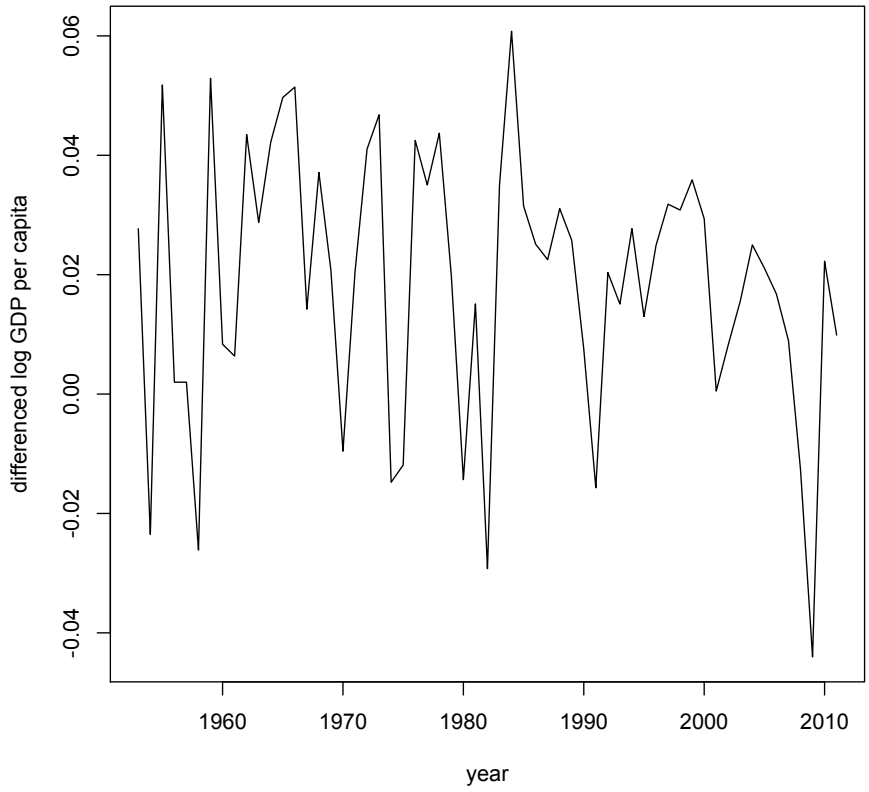
25.6.4 Bootstrapping with Trends

All the bootstraps discussed in §25.5 work primarily for stationary time series. (Parametric bootstraps are an exception, since we *could* include trends in the model.) If we have done extensive de-trending, the reasonable thing to do is to use a bootstrap to generate a series of fluctuations, add it to the estimated trend, and then repeat the *whole* analysis on the new, non-stationary surrogate series, including the de-trending. This works on the same sort of principle as resampling residuals in regressions (§6.4, especially 6.4.3).

25.7 Further Reading

Shumway and Stoffer (2000) is a good introduction to conventional time series analysis, covering R practicalities. Lindsey (2004) surveys a broader range of situations in less depth; it is readable, but opinionated, and I don't always agree with the opinions. Fan and Yao (2003) is a deservedly-standard reference on nonparametric time series models. The theoretical portions would be challenging for most readers of this book, but the methodology isn't, and it devotes about the right amount of space (no more than a quarter of the book) to the usual linear-model theory.

The block bootstrap was introduced by Künsch (1989). Davison and Hinkley (1997, §8.2) has a characteristically-clear treatment of the main flavors of bootstrap for time series; Lahiri (2003) is a thorough but theoretical. Bühlmann (2002) is also useful.




```
plot(gdppc$year[-1],diff(log(gdppc$y)),type="l",xlab="year",
  ylab="differenced log GDP per capita")
```

Figure 25.18: First differences of log GDP per capita, i.e., the year-to-year growth rate of GDP per capita.

The best introduction to stochastic processes I know of, by a very wide margin, is Grimmett and Stirzaker (1992). However, like most textbooks on stochastic processes, it says next to nothing about how to use them as models of data. Two exceptions I can recommend are the old but insightful Bartlett (1955), and the excellent Guttorp (1995).

The basic ergodic theorem in §25.2.2.1 follows a continuous-time argument in Frisch (1995). My general treatment of ergodicity is heavily shaped by Gray (1988) and Shields (1996).

In parallel to the treatment of time series by statisticians, physicists and mathematicians developed their own tradition of time-series analysis (Packard *et al.*, 1980), where the basic models are not stochastic processes but deterministic, yet unstable, dynamical systems. Perhaps the best treatment of this are Abarbanel (1996); Kantz and Schreiber (2004). There are in fact very deep connections between this approach and the question of why probability theory works in the first place (Ruelle, 1991), but that's not a subject for *data analysis*.

25.8 Exercises

1. In Eq. 25.33, assume that $m(x)$ has to be a linear function, $m(x) = \beta \cdot x$. Solve for the optimal β in terms of $\mathbf{y}$, $\mathbf{x}$, and $\mathbf{\Gamma}$. This "generalized least squares" (GLS) solution should reduce to ordinary least squares when $\Gamma = \sigma^2 \mathbf{I}$.

2. If $Z_t = Z_{t-1} + \epsilon_t$, with ϵ_t IID, prove that Z_t is not stationary. *Hint:* consider $\mathrm{Var}\left[Z_t\right]$.