Analysing Seasonal Data

by Adrian G Barnett, Peter Baker and Annette J Dobson

Abstract Many common diseases, such as the flu and cardiovascular disease, increase markedly in winter and dip in summer. These seasonal patterns have been part of life for millennia and were first noted in ancient Greece by both Hippocrates and Herodotus. Recent interest has focused on climate change, and the concern that seasons will become more extreme with harsher winter and summer weather. We describe a set of R functions designed to model seasonal patterns in disease. We illustrate some simple descriptive and graphical methods, a more complex method that is able to model non-stationary patterns, and the case-crossover to control for seasonal confounding.

In this paper we illustrate some of the functions of the **season** package (Barnett et al., 2012), which contains a range of functions for analysing seasonal health data. We were motivated by the great interest in seasonality found in the health literature, and the relatively small number of seasonal tools in R (or other software packages). The existing seasonal tools in R are:

- the `baysea` function of the **timsac** package and the `decompose` and `stl` functions of the **stats** package for decomposing a time series into a trend and season;
- the `dynlm` function of the **dynlm** package and the `ssm` function of the **sspir** package for fitting dynamic linear models with optional seasonal components;
- the `arima` function of the **stats** package and the `Arima` function of the **forecast** package for fitting seasonal components as part of an autoregressive integrated moving average (ARIMA) model; and
- the **bfast** package for detecting breaks in a seasonal pattern.

These tools are all useful, but most concern decomposing equally spaced time series data. Our package includes models that can be applied to seasonal patterns in unequally spaced data. Such data are common in observational studies when the timing of responses cannot be controlled (e.g. for a postal survey).

In the health literature much of the analysis of seasonal data uses simple methods such as comparing rates of disease by month or using a cosinor regression model, which assumes a sinusoidal seasonal pattern. We have created functions for these simple, but often very effective analyses, as we describe below.

More complex seasonal analyses examine non-stationary seasonal patterns that change over time. Changing seasonal patterns in health are currently of great interest as global warming is predicted to make seasonal changes in the weather more extreme. Hence there is a need for statistical tools that can estimate whether a seasonal pattern has become more extreme over time or whether its phase has changed.

Ours is also the first R package that includes the case-crossover, a useful method for controlling for seasonality.

This paper illustrates just some of the functions of the **season** package. We show some descriptive functions that give simple means or plots, and functions whose goal is inference based on generalised linear models. The package was written as a companion to a book on seasonal analysis by Barnett and Dobson (2010), which contains further details on the statistical methods and R code.

Analysing monthly seasonal patterns

Seasonal time series are often based on data collected every month. An example that we use here is the monthly number of cardiovascular disease deaths in people aged ≥ 75 years in Los Angeles for the years 1987–2000 (Samet et al., 2000). Before we examine or plot the monthly death rates we need to make them more comparable by adjusting them to a common month length (Barnett and Dobson, 2010, Section 2.2.1). Otherwise January (with 31 days) will likely have more deaths than February (with 28 or 29).

In the example below the `monthmean` function is used to create the variable `mmean` which is the monthly average rate of cardiovascular disease deaths standardised to a month length of 30 days. As the data set contains the population size (`pop`) we can also standardise the rates to the number of deaths per 100,000 people. The highest death rate is in January (397 per 100,000) and the lowest in July (278 per 100,000).

```
> data(CVD)
> mmean = monthmean(data = CVD,
  resp = CVD$cvd, adjmonth = "thirty",
  pop = pop/100000)
> mmean
      Month  Mean
    January 396.8
   February 360.8
      March 327.3
      April 311.9
```

```
      May 294.9
     June 284.5
     July 277.8
   August 279.2
September 279.1
  October 292.3
 November 313.3
 December 368.5
```

Plotting monthly data

We can plot these standardised means in a circular plot using the `plotCircular` function:

```
> plotCircular(area1 = mmean$mean,
    dp = 1, labels = month.abb,
    scale = 0.7)
```

This produces the circular plot shown in Figure 1. The numbers under each month are the adjusted averages, and the area of each segment is proportional to this average.

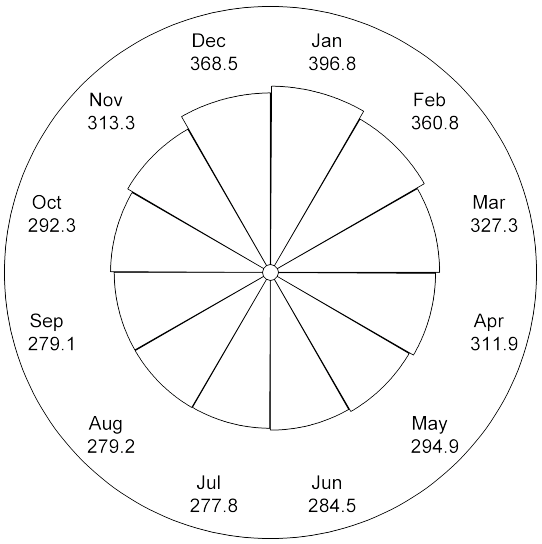

Figure 1: A circular plot of the adjusted monthly mean number of cardiovascular deaths in Los Angeles in people aged $\geq$ 75, 1987–2000.

The peak in the average number of deaths is in January, and the low is six months later in July indicating an annual seasonal pattern. If there were no seasonal pattern we would expect the averages in each month to be equal, and so the plot would be perfectly circular. The seasonal pattern is somewhat non-symmetric, as the decrease in deaths from January to July does not mirror the seasonal increase from July to January. This is because the increase in deaths does not start in earnest until October.

Circular plots are also useful when we have an observed and expected number of observations in each month. As an example, Figure 2 shows the number of Australian Football League players by their month of birth (for the 2009 football season) and the expected number of births per month based on national data. For this example we did not adjust for the unequal number of days in the months because we can compare the observed numbers to the expected (which are also based on unequal month lengths). Using the expected numbers also shows any seasonal pattern in the national birth numbers. In this example there is a very slight decrease in births in November and December.

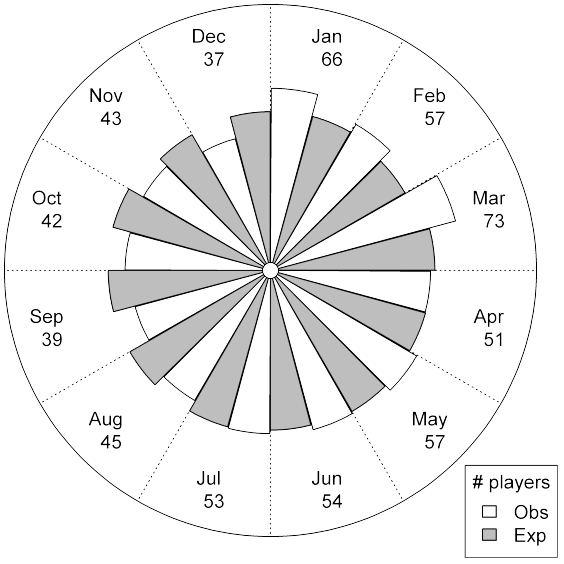

Figure 2: A circular plot of the monthly number of Australian Football League players by their month of birth (*white segments*) and the expected numbers based on national data for men born in the same period (*grey segments*). Australian born players in the 2009 football season.

The figure shows the greater than expected number of players born in January to March, and the fewer than expected born in August to December. The numbers around the outside are the observed number of players. The code to create this plot is:

```
> data(AFL)
> plotCircular(area1 = AFL$players,
  area2 = AFL$expected, scale = 0.72,
  labels = month.abb, dp = 0, lines = TRUE,
  auto.legend = list(
    labels = c("Obs", "Exp"),
    title = "# players"))
```

The key difference from the code to create the previous circular plot is that we have given values for both `area1` and `area2`. The '`lines = TRUE`' option added the dotted lines between the months. We have also included a legend.

As well as a circular plot we also recommend a time series plot for monthly data, as these plots are useful for highlighting the consistency in the seasonal pattern and possibly also the secular trend and

any unusual observations. For the cardiovascular example data a time series plot is created using

```
> plot(CVD$yrmon, CVD$cvd, type = 'o',
  pch = 19,
  ylab = 'Number of CVD deaths per month',
  xlab = 'Time')
```

The result is shown in Figure 3. The January peak in CVD was clearly larger in 1992 and 1994 compared with 1991, 1993 and 1995. There also appears to be a slight downward trend from 1987 to 1992.

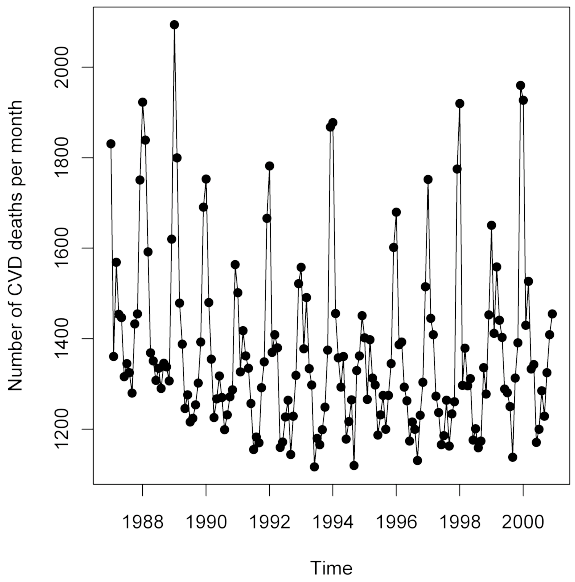

Figure 3: Monthly number of cardiovascular deaths in Los Angeles for people aged ≥ 75, 1987–2000.

Modelling monthly data

A simple and popular statistical model for examining seasonal patterns in monthly data is to use a simple linear regression model (or generalised linear model) with a categorical variable of month. The code below fits just such a model to the cardiovascular disease data and then plots the rate ratios (Figure 4).

```
> mmodel = monthglm(formula = cvd ~ 1,
  data = CVD, family = poisson(),
  offsetpop = pop/100000,
  offsetmonth = TRUE, refmonth = 7)
> plot(mmodel)
```

As the data are counts we used a Poisson model. We adjusted for the unequal number of days in the month by using an offset (`offsetmonth = TRUE`), which divides the number of deaths in each month by the number of days in each month to give a daily rate. The reference month was set to July (`refmonth = 7`). We could have added other variables to the model, by adding them to the right hand side of the equation (e.g. '`formula = cvd ~ year`' to include a linear trend for year).

The plot in Figure 4 shows the mean rate ratios and 95% confidence intervals. The dotted horizontal reference line is at the rate ratio of 1. The mean rate of deaths in January is 1.43 times the rate in July. The rates in August and September are not statistically significantly different to the rates in July, as the confidence intervals in these months both cross 1.

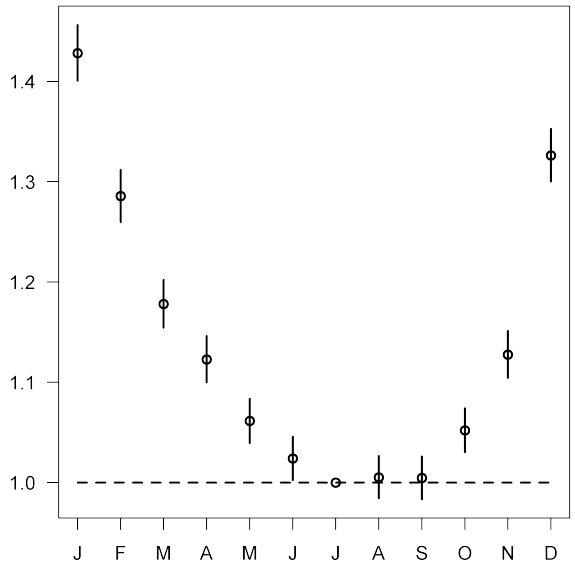

Figure 4: Mean rate ratios and 95% confidence intervals of cardiovascular disease deaths using July as a reference month.

Cosinor

The previous model assumed that the rate of cardiovascular disease varied arbitrarily in each month with no smoothing of or correlation between neighbouring months. This is an unlikely assumption for this seasonal pattern (Figure 4). The advantage of using arbitrary estimates is that it does not constrain the shape of the seasonal pattern. The disadvantage is a potential loss of statistical power. Models that assume some parametric seasonal pattern will have a greater power when the parametric model is correct. A popular parametric seasonal model is the cosinor model (Barnett and Dobson, 2010, Chapter 3), which is based on a sinusoidal pattern,

$$s_t = A\cos\left(\frac{2\pi t}{c} - P\right), \qquad t = 1,\ldots,n,$$

where A is the amplitude of the sinusoid and P is its phase, c is the length of the seasonal cycle (e.g. $c = 12$ for monthly data with an annual seasonal pattern), t is the time of each observation and n is the total number of times observed. The amplitude tells us the size of the seasonal change and the phase tells us where it peaks. The sinusoid assumes a smooth seasonal pattern that is symmetric about its peak (so the rate of the seasonal increase in disease is equal to the decrease). We fit the Cosinor as part of a generalised linear model.

The example code below fits a cosinor model to the cardiovascular disease data. The results are for each month, so we used the 'type = 'monthly'' option with 'date = month'.

```
> res = cosinor(cvd ~ 1, date = month,
  data = CVD, type = 'monthly',
  family = poisson(), offsetmonth = TRUE)
> summary(res)
Cosinor test
Number of observations = 168
Amplitude = 232.34 (absolute scale)
Phase: Month = 1.3
Low point: Month = 7.3
Significant seasonality based on adjusted
significance level of 0.025  =  TRUE
```

We again adjusted for the unequal number of days in the months using an offset (offsetmonth = TRUE). The amplitude is 232 deaths which has been given on the absolute scale and the phase is estimated as 1.27 months (early January).

An advantage of these cosinor models is that they can be fitted to unequally spaced data. The example code below fits a cosinor model to data from a randomised controlled trial of physical activity with data on body mass index (BMI) at baseline (Eakin et al., 2009). Subjects were recruited as they became available and so the measurement dates are not equally spaced. In the example below we test for a sinusoidal seasonal pattern in BMI.

```
> data(exercise)
> res = cosinor(bmi ~ 1, date = date,
  type = 'daily', data = exercise,
  family = gaussian())
> summary(res)
Cosinor test
Number of observations = 1152
Amplitude = 0.3765669
Phase: Month = November , day = 18
Low point: Month = May , day = 19
Significant seasonality based on adjusted
significance level of 0.025  =  FALSE
```

Body mass index has an amplitude of 0.38 kg/m^2 which peaks on 18 November, but this increase is not statistically significant. In this example we used 'type = 'daily'' as subjects' results related to a specific date ('date = date' specifies the day when they were measured). Thus the phase for body mass index is given on a scale of days, whereas the phase for cardiovascular death was given on a scale of months.

Non-stationary cosinor

The models illustrated so far have all assumed a stationary seasonal pattern, meaning a pattern that does not change from year to year. However, seasonal patterns in disease may gradually change because of changes in an important exposure. For example, improvements in housing over the 20th century are part of the reason for a decline in the winter peak in mortality in London (Carson et al., 2006).

To fit a non-stationary cosinor we expand the previous sinusoidal equation thus

$$s_t = A_t \cos\left(\frac{2\pi t}{c} - P_t\right), \qquad t = 1, \ldots, n$$

so that both the amplitude and phase of the cosinor are now dependent on time. The key unknown is the extent to which these parameters will change over time. Using our nscosinor function the user has some control over the amount of change and a number of different models can be tested assuming different levels of change. The final model should be chosen using model fit diagnostics and residual checks (available in the seasrescheck function).

The nscosinor function uses the Kalman filter to decompose the time series into a trend and seasonal components (West and Harrison, 1997, Chapter 8), so can only be applied to equally spaced time series data. The code below fits a non-stationary sinusoidal model to the cardiovascular disease data (using the counts adjusted to the average month length, adj).

```
> nsmodel = nscosinor(data = CVD,
  response = adj, cycles = 12, niters = 5000,
  burnin = 1000, tau = c(10, 500), inits = 1)
```

The model uses Markov chain Monte Carlo (MCMC) sampling, so we needed to specify the number of iterations (niters), the number discarded as a burn-in (burnin), and an initial value for each seasonal component (inits). The cycles gives the frequency of the sinusoid in units of time, in this case a seasonal pattern that completes a cycle in 12 months. We can fit multiple seasonal components, for example 6 and 12 month seasonal patterns would be fitted using 'cycles = c(6,12)'. The tau are smoothing parameters, with tau[1] for the trend, tau[2] for the first seasonal parameter, tau[3] for the second seasonal parameter. They are fixed values that scale the time between observations. Larger values allow more time between observations and hence create a more flexible spline. The ideal values for tau should be chosen using residual checking and trial and error.

The estimated seasonal pattern is shown in Figure 5. The mean amplitude varies from around 230 deaths (winter 1989) to around 180 deaths (winter 1995), so some winters were worse than others. Importantly the results did not show a steady decline in amplitude, so over this period seasonal deaths continued to be a problem despite any improvements in health care or housing. However, the residuals from this model do show a significant seasonal pattern (checked using the seasrescheck function). This residual seasonal pattern is caused because the

seasonal pattern in cardiovascular deaths is non-sinusoidal (as shown in Figure 1) with a sharper increase in deaths than decline. The model assumed a sinusoidal pattern, albeit a non-stationary one. A better fit might be achieved by adding a second seasonal cycle at a shorter frequency, such as 6 months.

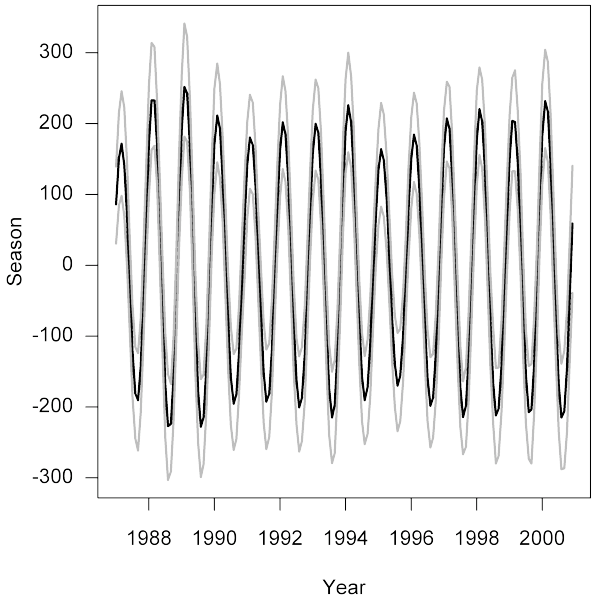

Figure 5: Estimated non-stationary seasonal pattern in cardiovascular disease deaths for Los Angeles, 1987–2000. Mean (*black line*) and 95% confidence interval (*grey lines*).

Case-crossover

In some circumstances seasonality is not the focus of investigation, but is important because its effects need to be taken into account. This could be because both the outcome and the exposure have an annual seasonal pattern, but we are interested in associations at a different frequency (e.g. daily).

The case-crossover can be used for individual-level data, e.g. when the data are individual cases with their date of heart attack and their recent exposure. However, we are concerned with regularly spaced time-series data, where the data are grouped, e.g. the number of heart attacks on each day in a year.

The case-crossover is a useful time series method for controlling for seasonality (Maclure, 1991). It is similar to the matched case-control design, where the exposure of cases with the disease are compared with one or more matched controls without the disease. In the case-crossover, cases act as their own control, since exposures are compared on case and control days (also known as index and referent days). The case day will be the day on which an event occurred (e.g. death), and the control days will be nearby days in the same season as the exposure but with a possibly different exposure. This means the cases and controls are matched for season, but not for some other short-term change in exposure such as air pollution or temperature. A number of different case-crossover designs for time-series data have been proposed. We used the time-stratified method as it is a localisable and ignorable design that is free from overlap bias while other referent window designs that are commonly used in the literature (e.g. symmetric bi-directional) are not (Janes et al., 2005). Using this design the data in broken into a number of fixed strata (e.g. 28 days or the months of the year) and the case and control days are compared within the same strata.

The code below applies a case-crossover model to the cardiovascular disease data. In this case we use the daily cardiovascular disease (with the number of deaths on every day) rather than the data used above which used the number of cardiovascular deaths in each month. The independent variables are mean daily ozone (`o3mean`, which we first scale to a 10 unit increase) and temperature (`tmpd`). We also control for day of the week (using Sunday as the reference category). For this model we are interested in the effect of day-to-day changes in ozone on the day-to-day changes in mortality.

```
> data(CVDdaily)
> CVDdaily$o3mean = CVDdaily$o3mean / 10
> cmodel = casecross(cvd ~ o3mean + tmpd +
  Mon + Tue + Wed + Thu + Fri + Sat,
  data = CVDdaily)
> summary(cmodel, digits = 2)
Time-stratified case-crossover with a stratum
length of 28 days
Total number of cases 230695
Number of case days with available control
days 5114
Average number of control days per case day 23.2

Parameter Estimates:
          coef exp(coef) se(coef)     z Pr(>|z|)
o3mean -0.0072      0.99  0.00362 -1.98  4.7e-02
tmpd    0.0024      1.00  0.00059  4.09  4.3e-05
Mon     0.0323      1.03  0.00800  4.04  5.3e-05
Tue     0.0144      1.01  0.00808  1.78  7.5e-02
Wed    -0.0146      0.99  0.00807 -1.81  7.0e-02
Thu    -0.0118      0.99  0.00805 -1.46  1.4e-01
Fri     0.0065      1.01  0.00806  0.81  4.2e-01
Sat     0.0136      1.01  0.00788  1.73  8.4e-02
```

The default stratum length is 28, which means that cases and controls are compared in blocks of 28 days. This stratum length should be short enough to remove any seasonal pattern in ozone and temperature. Ozone is formed by a reaction between other air pollutants and sunlight and so is strongly seasonal with a peak in summer. Cardiovascular mortality is at its lowest in summer as warmer temperatures lower blood pressures and prevent flu outbreaks. So without removing these seasonal patterns we might find a significant negative association between ozone and mortality. The above results suggest a marginally significant negative association be-

tween ozone and mortality, as the odds ratio for a ten unit increase in ozone is $\exp(-0.0072) = 0.993$ (p-value = 0.047). This may indicate that we have not sufficiently controlled for season and so should reduce the stratum length using the `stratalength` option.

As well as matching cases and controls by stratum, it is also possible to match on another confounder. The code below shows a case-crossover model that matched case and control days by a mean temperature of ± 1 degrees Fahrenheit.

```
> mmodel = casecross(cvd ~ o3mean +
  Mon + Tue + Wed + Thu + Fri + Sat,
  matchconf = 'tmpd', confrange = 1,
  data = CVDdaily)
> summary(mmodel, digits = 2)
Time-stratified case-crossover with a stratum
length of 28 days
Total number of cases 205612
Matched on tmpd plus/minus 1
Number of case days with available control
days 4581
Average number of control days per case day 5.6

Parameter Estimates:
         coef exp(coef) se(coef)    z Pr(>|z|)
o3mean 0.0046         1   0.0043 1.07  2.8e-01
Mon    0.0461         1   0.0094 4.93  8.1e-07
Tue    0.0324         1   0.0095 3.40  6.9e-04
Wed    0.0103         1   0.0094 1.10  2.7e-01
Thu    0.0034         1   0.0093 0.36  7.2e-01
Fri    0.0229         1   0.0094 2.45  1.4e-02
Sat    0.0224         1   0.0092 2.45  1.4e-02
```

By matching on temperature we have restricted the number of available control days, so there are now only an average of 5.6 control days per case, compared with 23.2 days in the previous example. Also there are now only 4581 case days with at least one control day available compared with 5114 days for the previous analysis. So 533 days have been lost (and 25,083 cases), and these are most likely the days with unusual temperatures that could not be matched to any other days in the same stratum. We did not use temperature as an independent variable in this model, as it has been controlled for by the matching. The odds ratio for a ten unit increase in ozone is now positive (OR = $\exp(0.0046) = 1.005$) although not statistically significant (p-value = 0.28).

It is also possible to match cases and control days by the day of the week using the '`matchdow = TRUE`' option.

Bibliography

A. G. Barnett, P. Baker, and A. J. Dobson. *season: Seasonal analysis of health data*, 2012. URL http://CRAN.R-project.org/package=season. R package version 0.3-1.

A. G. Barnett and A. J. Dobson. *Analysing Seasonal Health Data*. Springer, 2010.

C. Carson, S. Hajat, B. Armstrong, and P. Wilkinson. Declining vulnerability to temperature-related mortality in London over the 20th Century. *Am J Epidemiol*, 164(1):77–84, 2006.

E. Eakin, M. Reeves, S. Lawler, N. Graves, B. Oldenburg, C. DelMar, K. Wilke, E. Winkler, and A. Barnett. Telephone counseling for physical activity and diet in primary care patients. *Am J of Prev Med*, 36(2):142–149, 2009.

H. Janes, L. Sheppard, and T. Lumley. Case-crossover analyses of air pollution exposure data: Referent selection strategies and their implications for bias. *Epidemiology*, 16(6):717–726, 2005.

M. Maclure. The case-crossover design: a method for studying transient effects on the risk of acute events. *Am J Epidemiol*, 133(2):144–153, 1991.

J. Samet, F. Dominici, S. Zeger, J. Schwartz, and D. Dockery. The national morbidity, mortality, and air pollution study, part I: Methods and methodologic issues. 2000.

M. West and J. Harrison. *Bayesian Forecasting and Dynamic Models*. Springer Series in Statistics. Springer, New York; Berlin, 2nd edition, 1997.

Adrian Barnett
School of Public Health
Queensland University of Technology
Australia
a.barnett@qut.edu.au

Peter Baker
School of Population Health
University of Queensland
Australia

Annette Dobson
School of Population Health
University of Queensland
Australia