Time Series Analysis

Objectives

- Introduction
- Identifying trend, seasonality, and random errors
- Fitting a model and relevant diagnostics

What are Time Series Data?

Time series data are vectors of numbers, typically regularly spaced in time. Yearly counts of animals, daily prices of shares, monthly means of temperature, and minute-by-minute details of blood pressure are all examples of time series, but they are measured on different time scales. Sometimes the interest is in the time series itself (e.g., whether or not it is cyclic, or how well the data fit a particular theoretical model), and sometimes the time series is incidental to a designed experiment (e.g., repeated measures). The three key concepts in time series analysis are

- trend
- serial dependence
- stationarity

Most time series analyses assume that the data are untrended. If they do show a consistent upward or downward trend, then they can be detrended before analysis (e.g., by differencing). Serial dependence arises because the values of adjacent members of a time series may well be correlated. Stationarity is a technical concept, but it can be thought of simply as meaning that the time series has the same properties wherever you start looking at it (e.g., white noise is a sequence of mutually independent random variables each with mean zero and variance $\sigma^2 > 0$)

Ex 1. The following sample data `airpass` are from the `faraway` library. Study the use of "lag," `lag.plot`, fitting a model with "lags" and predicting.

```
> library(faraway)
> dim(airpass)
[1] 144   2
> head(airpass)
  pass     year
1  112 49.08333
2  118 49.16667
3  132 49.25000
4  129 49.33333
5  121 49.41667
6  135 49.50000
> attach(airpass)
> plot(pass~year,type="l",ylab="Passengers")   # Fig. 1
> # plot.ts(pass,type="l",ylab="Passengers")
```

```
> library(graphics)
> model1 <- ts(airpass[,1],start=1949,freq=12)
> plot(stl(model1,s.window="periodic"))   # Fig. 2

> model2 <- lm(log(pass)~year)
> plot(pass~year,type="l",ylab="Passengers")
> lines(exp(predict(model2))~year,col=2,lwd=2)   # Fig. 1
> diff(pass)   # Calculates successive differences, i.e., y2-y1, y3-y2, etc
  [1]    6   14   -3   -8   14   13    0  -12  -17  -15   14   -3   11   15   -6  -10
.....

> lag.plot(pass,13,col=4) # Because it's yearly data, we are checking up to "lag13"; See Fig. 3
> lag.df <- embed(log(pass),14) # Checks up to "lag14" of "log(y)"
# lag.df
# (lag.df1 <- embed(pass,4)) # Study this to learns how "embed" works.
> colnames(lag.df) <- c("Y",paste0("lag",1:13))
> lag.df <- data.frame(lag.df)

> model3 <- lm(Y~lag1+lag12+lag13,data=lag.df)
> summary(model3)

Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept)  0.13848    0.05361   2.583   0.0109 *
lag1         0.69231    0.06186  11.192  < 2e-16 ***
lag12        0.92152    0.03473  26.532  < 2e-16 ***
lag13       -0.63214    0.06768  -9.340 4.16e-16 ***
---
Residual standard error: 0.04164 on 127 degrees of freedom
Multiple R-squared:  0.9892,    Adjusted R-squared:  0.989
F-statistic:  3892 on 3 and 127 DF,  p-value: < 2.2e-16

> model4 <- lm(Y~lag1+lag2+lag3+lag12+lag13,data=lag.df)
> summary(model4)
> plot(pass~year,type="l")
> lines(airpass$year[14:144],exp(predict(model3)),col=2,lwd=2)   # Fig. 4
> lag.df[nrow(lag.df),] # To predict the next "future" value
> predict(model3,list(lag1=6.068426,lag12=6.033086,lag13=6.003887),se=T,interval="p")
```

Durbin-Watson Test for Autocorrelation

Autocorrelation refers to the correlation of a time series with its own past and future values. Autocorrelation is also sometimes called lagged correlation or serial correlation, which refers to the correlation between members of a series of numbers arranged in time. Positive autocorrelation might be considered a specific form of persistence, a tendency for a system to remain in the same state from one observation to the next. For example, the likelihood of tomorrow being rainy is greater if today is rainy than if today is dry. Geophysical time series are frequently autocorrelated because of inertia or carryover processes in the physical system. For example, the slowly evolving and moving low pressure systems in the atmosphere might impart persistence to daily rainfall. Or the slow drainage of groundwater reserves might impart correlation to successive annual flows of a river. Or stored photosynthates might impart correlation to successive annual values of tree-ring indices.

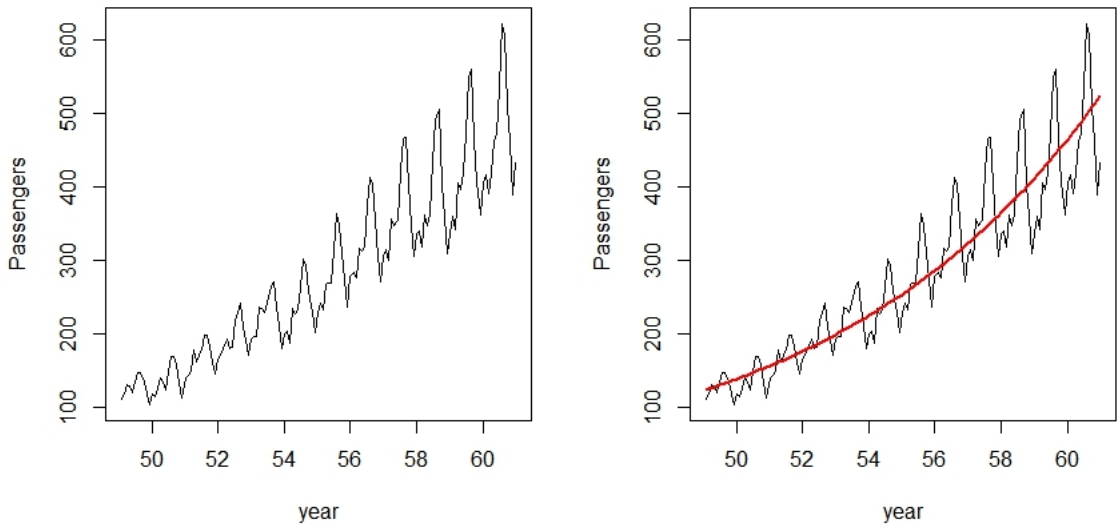

Figure 1: **(Left)** Time series plot of the sample data, **(Right)** Data plot overlaid with model2: $\log(y)$ vs x

Autocorrelation complicates the application of statistical tests by reducing the number of independent observations. Autocorrelation can also complicate the identification of significant covariance or correlation between time series (e.g., precipitation with a tree-ring series). Autocorrelation can be exploited for predictions: an autocorrelated time series is predictable, probabilistically, because future values depend on current and past values. Three tools for assessing the autocorrelation of a time series are (1) the time series plot, (2) the lagged scatterplot, and (3) the autocorrelation function.

Ex 2. The Australian ecologist, A.J. Nicholson, reared blowfly larvae on pieces of liver in laboratory cultures that his technicians kept running continuously for almost 7 years (361 weeks, to be exact). To view and play with this sample dataset, go to `http://users.humboldt.edu/ygkim`, first click "Sample Datasets for R Book," then click "download text files." I have saved "blowfly" dataset at `U:STAT510`, that's "U:" drive under a subdirectory "STAT510."

```
> blowfly <- read.table("U:\\STAT510\\blowfly.txt",header=T)
> dim(blowfly)
[1] 361   1
> attach(blowfly)
> names(blowfly)
> flies <- ts(flies)    #You need this to make the variable into a time series object in R.
> flies
Time Series:
Start = 1
End = 361
Frequency = 1
```

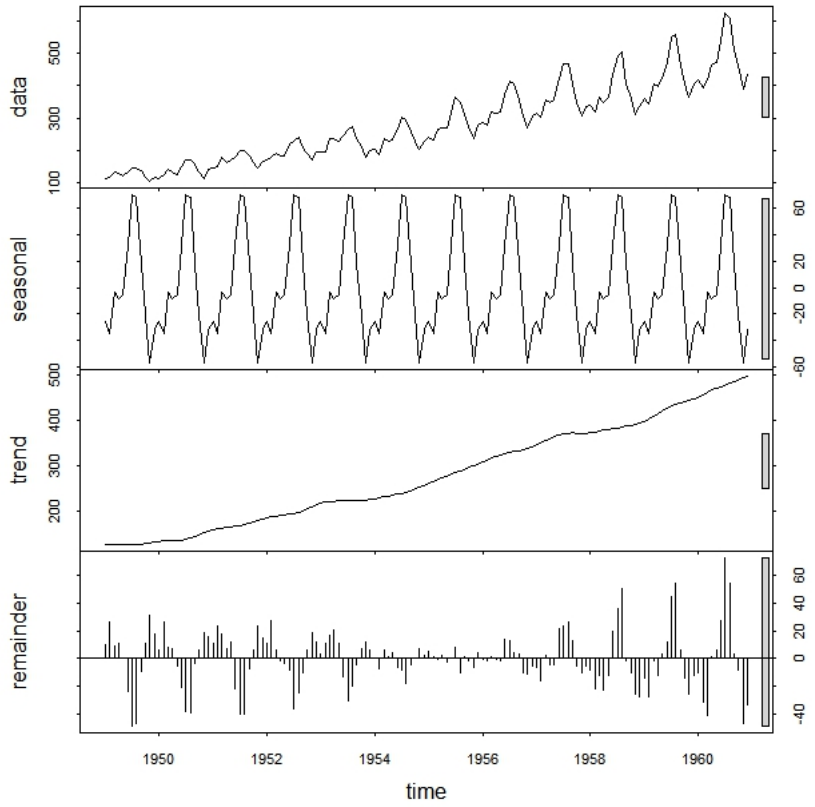

Figure 2: R's built-in plot

```
  [1]   948   942   911   858   801   676   504   397   248   146  1801  6235  5974
 << omitted to save paper >>>
> plot(flies)   #ts.plot(flies) produces the same plot. See Fig. 3
```

This classic time series has two clear features:

- For the first 200 weeks the system exhibits beautifully regular cycles.
- After week 200, things change (perhaps a genetic mutation had arisen); the cycles become much less clear-cut, and the population begins a pronounced upward "trend."

There are two important ideas to understand in time series analysis: **autocorrelation** and **partial autocorrelation**. The first describes how this week's population is related to last week's population. This is the autocorrelation at lag 1. The second describes the relationship between this week's population and the population at lag t once we have controlled for the correlations between all of the successive weeks between this week and week t. This should become clear if we draw the scatterplots from which the first four autocorrelation terms are calculated (lag 1 to lag 4). There is a snag, however. The vector of flies at lag 1 is shorter (by one) than the original vector because the first element of the lagged vector is the second element of flies. The coordinates of the first data point to be drawn on the scatterplot are (`flies[1],flies[2]`) and the coordinates of the last plot

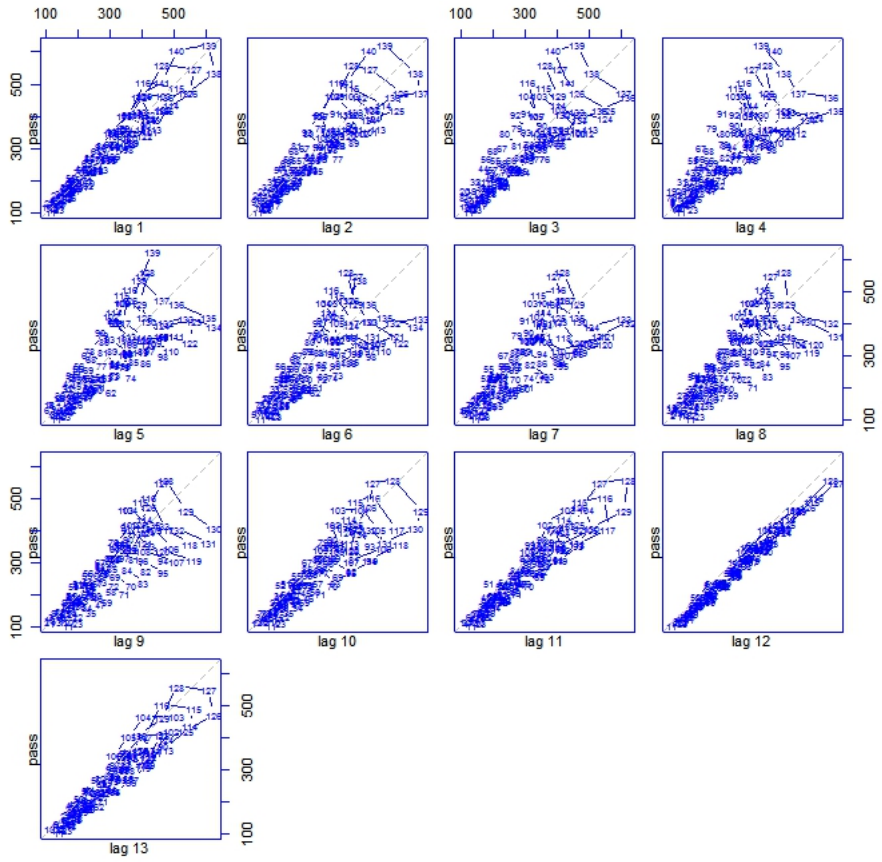

Figure 3: y's are most strongly autocorrelated with lag 12 & lag 1

that can be drawn are (`flies[360], flies[361]`) because the original vector is 361 element long. First, Durbin-Watson test looks like this:

```
> library(car)
> durbinWatsonTest(lm(flies[-c(361)]~flies[-c(1)]))
 lag Autocorrelation D-W Statistic p-value
   1       0.1394312      1.714666   0.002
 Alternative hypothesis: rho != 0
> library(lmtest)
> dwtest(lm(flies[-c(361)]~flies[-c(1)]))

        Durbin-Watson test

data:  lm(flies[-c(361)] ~ flies[-c(1)])
DW = 1.7147, p-value = 0.002878
alternative hypothesis: true autocorrelation is greater than 0
```

It says there is a significant autocorrelation. To investiaget more, we then produce the four plots for lags 1 to 4 uisng a function like this:

```
> par(mfrow=c(2,2))
> sapply(1:4,function(x) plot(flies[-c(361:(361-x+1))],flies[-c(1:x)]))
```

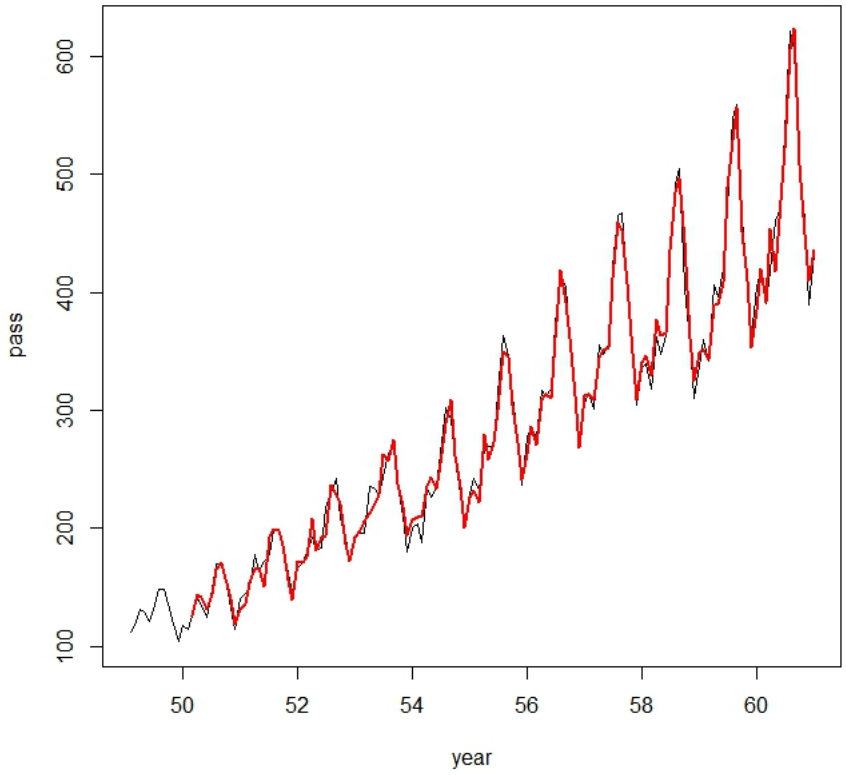

Figure 4: Plot of data overlaid with model3: y vs. lag 1, lag 12, & lag13

(See Fig. 8) The correlation is very strong at lag 1, but notice how the variance increases with population size: small populations this week are invariably correlated with small populations next week, but large populations this week may be associated with large or small populations next week. The striking pattern here is the way that the correlation fades away as the size of the lag increases. Because the population is cyclic, the correlation goes to zero, then becomes weakly negative and then becomes strongly negative. This occurs at lags that are half the cycle length. Looking back at the time series, the cycles look to be about 20 weeks in length. So lets repeat the scatterplots at lags of 7, 8, 9 and 10 weeks:

```
> par(mfrow=c(2,2))
> sapply(7:10,function(x) plot(flies[-c(361:(361-x+1))],flies[-c(1:x)]))
```

(See Fig. 9) The negative correlation at lag 10 gradually emerges from the fog of no correlation at lag 7.

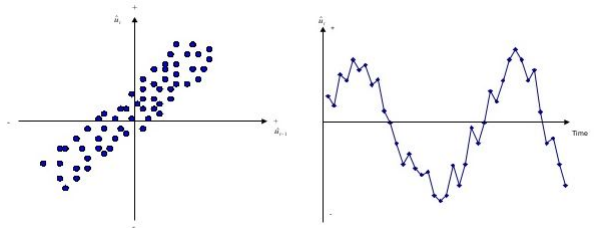

Figure 5: Positive autocorrelation in indicated by a cyclical residual plot over time.

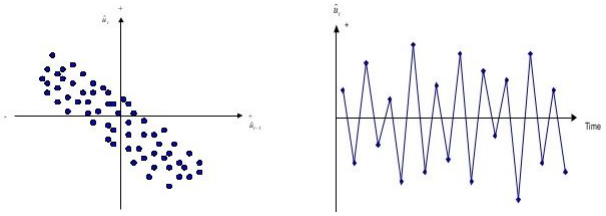

Figure 6: Negative autocorrelation in indicated by an alternating pattern where the residuals cross time axis more frequently than if they were distributed randomly.

More formally, the autocorrelation function $\rho(k)$ at lag k is

$$\rho(k) = \frac{\gamma(k)}{\gamma(0)},$$

where $\rho(k)$ is the autocovariance function at lag k of a stationary random function $\{Y(t)\}$ given by

$$\gamma(k) = cov\{Y(t), Y(t-k)\}.$$

The most important properties of the autocorrelation coefficient are:

- They are symmetric backwards and forwards, so $\rho(k) = \rho(-k)$.
- The limits are $-1 \leq \rho(k) \leq 1$.
- When $Y(t)$ and $Y(t-k)$ are independent, then $\rho(k) = 0$.
- The converse of this is not true, so that $\rho(k) = 0$ does not imply that $Y(t)$ and $Y(t-k)$ are independent.

Autocorrelation, Partial Autocorrelation

A **first-order autoregressive process** is written as

$$Y_t = \alpha Y_{t-1} + Z_t$$

This means that this week's population is α times last week's population plus a random term Z_t. The randomness is white noise; the values of Z's are serially independent with a mean of zero, and finite variance σ^2.

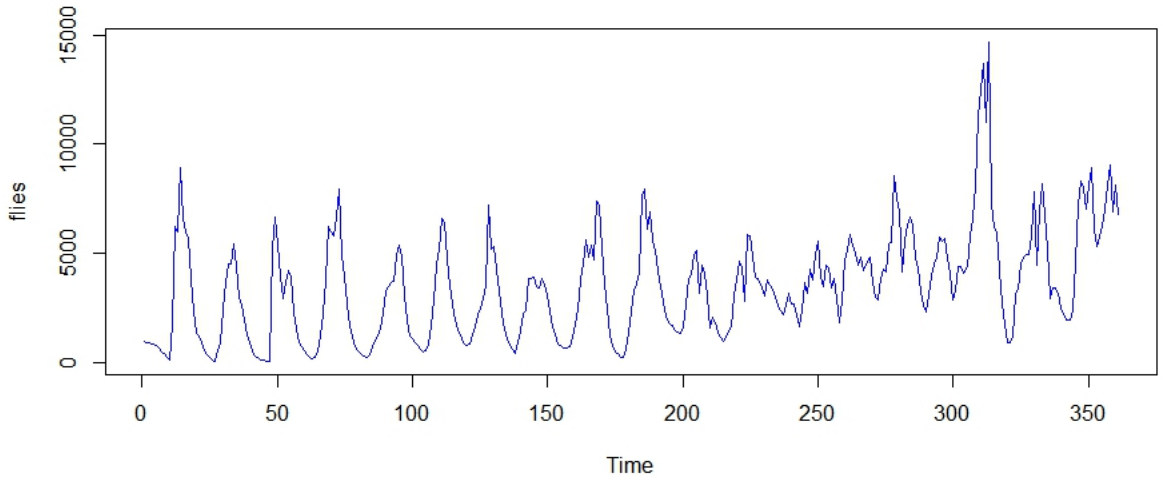

Figure 7: Time series plot of the sample dataset `blowfly`

In a stationary times series, $-1 < \alpha < 1$. In general, then, the autocorrelation function of $\{Y(t)\}$ is

$$\rho(k) = \alpha^k, \qquad k = 0, 1, 2, \ldots$$

Partial autocorrelation is the relationship between this week's population and the population at lag k when we have controlled for the correlations between all of the successive weeks between this week and week k. That is to say, the partial autocorrelation is the correlation between $Y(t)$ and $Y(t+k)$ after regression of $Y(t)$ on $Y(t+1), Y(t+2), \ldots, Y(t+k-1)$. It is obtained by solving the Yule-Walker equation

$$\rho_k = \sum_{i=1}^{p} \alpha_i \rho_{k-i}, \qquad k > 0$$

with ρ replaced by r (correlation coefficients estimated from the data). For example, the partial autocorrelation between time 1 and time 3 is calculated by

$$r_{13.2} = \frac{r_{13} - r_{12} r_{23}}{\sqrt{\left(1 - r_{12}^2\right)\left(1 - r_{23}^2\right)}}$$

Let's look at the correlation structure of the `blowfly` data. The R function for calculating autocorrelations and partial autocorrelations is acf (the "autocorrelation function").

```
> acf(flies,col=2,main="autocorrelation plot")  # See Fig. 6

> acf(flies,col=2,type="p",main="partial autocorrelation plot") # See Fig. 7
```

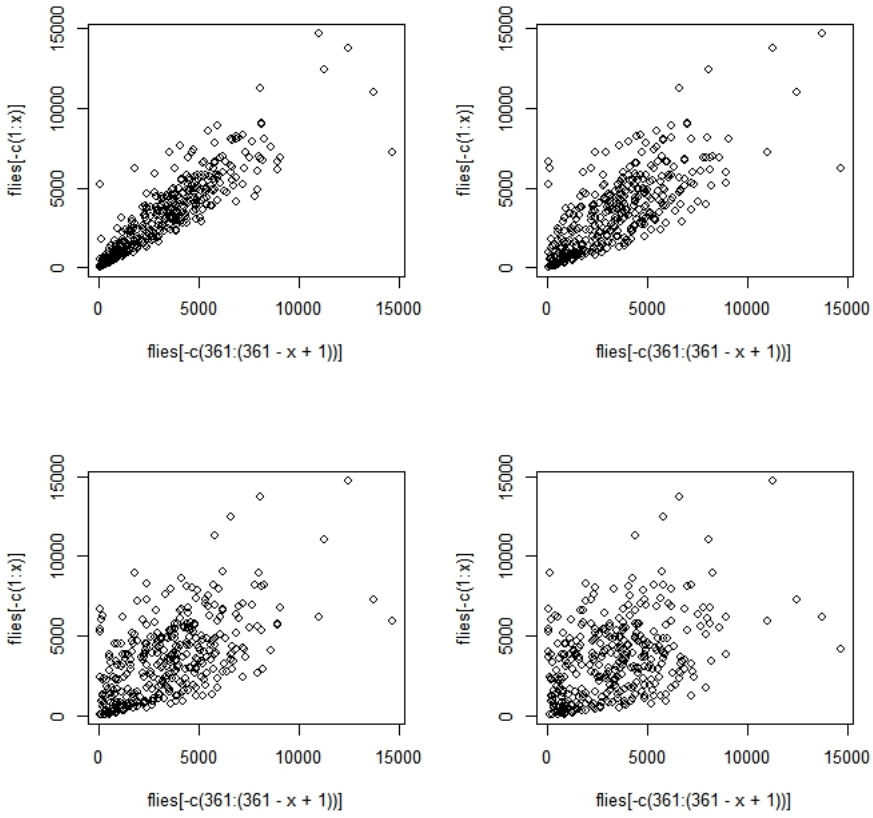

Figure 8: Scatterplots of `blowfly` with lags at 1, 2, 3 and 4 weeks

Detrending

We now investigate the behaviour of the second half of the time series separately. Let's say it is from week 201 onwards. In order to check the upward "trend" we have seen earlier, we fit a linear model with `Index` as an explanatory variable.

```
> second <- flies[201:361]
> length(second)
[1] 161
> summary(lm(second~I(1:161)))
Coefficients:
            Estimate Std. Error t value Pr(>|t|)
(Intercept) 2827.531    336.661   8.399 2.37e-14 ***
I(1:161)      21.945      3.605   6.087 8.29e-09 ***
---
Residual standard error: 2126 on 159 degrees of freedom
Multiple R-squared:  0.189,     Adjusted R-squared:  0.1839
F-statistic: 37.05 on 1 and 159 DF,  p-value: 8.289e-09
```

This shows that there is a highly significant upward trend of about 22 extra flies on average each week in the second half of time series. We can detrend the data by subtracting the fitted values

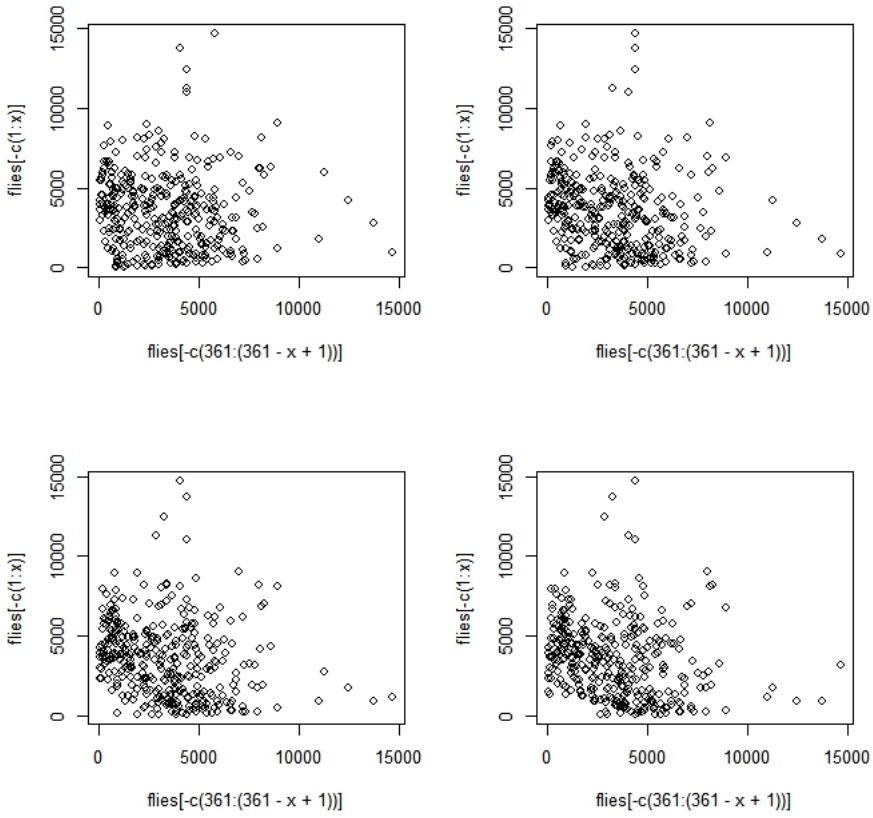

Figure 9: Scatterplots of `blowfly` with lags at 7, 8, 9 and 10 weeks

from the linear regression of second on day number:

```
> second <- flies[201:361]
> detrended <- second - predict(lm(second~I(1:161)))
> par(mfrow=c(1,3))
> ts.plot(detrended,col=2)   # See Fig. 8
> acf(detrended,main="",col=2)
> acf(detrended,main="",type="p",col=2)
```

Moving Average

Moving average is a simple way of seeing pattern in time series data. For example, a 3-point moving average is calculated by

$$y_i^{'} = \frac{y_{i-1} + y_i + y_{i+1}}{3}.$$

We can create a simple function to calculate this and overlay them on the data.

```
> ma3 <- function(x) {
+     y <- numeric(length(x)-2)
+     for(i in 2:(length(x)-1)) {
```

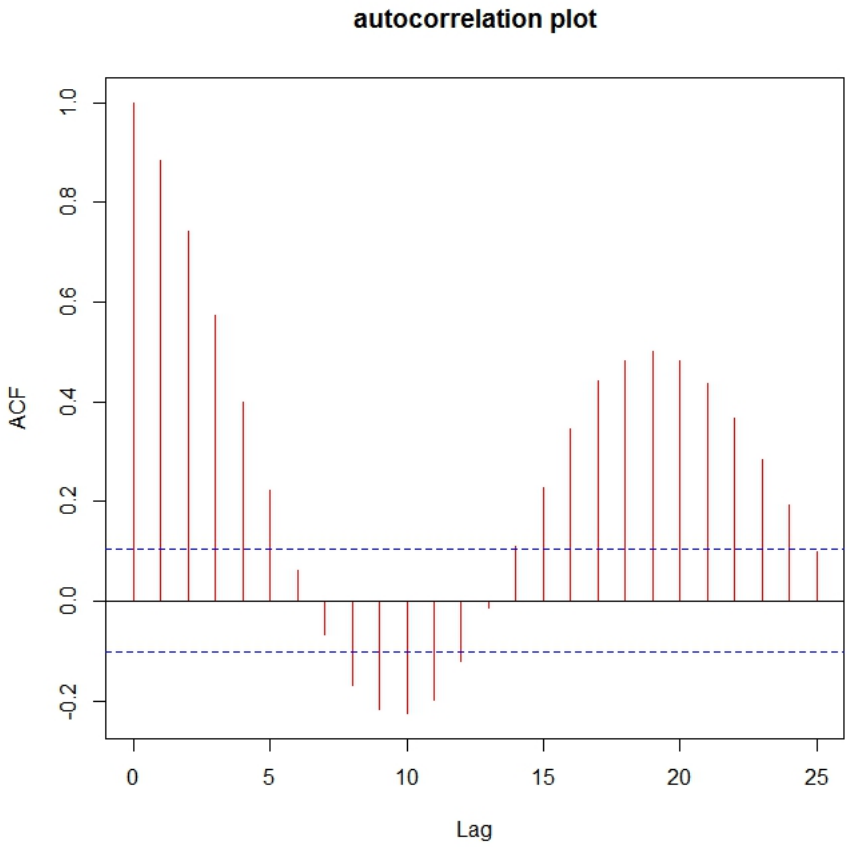

Figure 10: Autocorrelation plot by `acf` is used to identify "period" of cycles. In this plot, regular cycles have a period of 19 weeks. Autocorrelation is 1 at "lag=0" (as it should!), 0.8863 (at lag=1), and 0.7502 (at lag=2). These can also be found by `> cor(flies[-c(361)],flies[-c(1)])` and `> cor(flies[-c(361,360)],flies[-c(1,2)])`

```
+     y[i] <- (x[i-1]+x[i]+x[i+1])/3  }
+ y }
> flies_ma <- ma3(blowfly$flies)
> plot(flies) # See Fig. 9
> lines(flies_ma,lty=1,col=2,lwd=2)
```

Seasonality

Many time series data exhibit seasonal cycles. The commonest applications involve weather data: here are daily maximum and minimum temperatures from Silwood Park in south-east England over the period 1987-2005 inclusive.

```
> weather <- read.table("U:\\STAT510\\SilwoodWeather.txt",header=T)
> dim(weather)
[1] 6940    5
> head(weather)
  upper lower rain month   yr
```

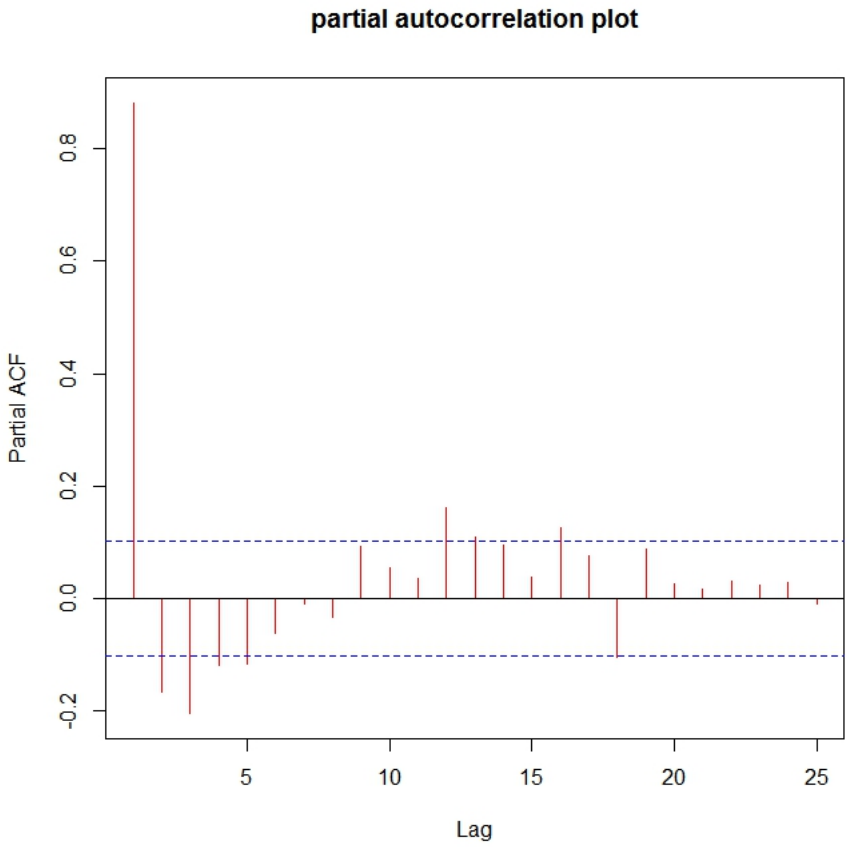

Figure 11: Partial autocorrelation plot by `acf` is used to identify proper number of "lags." In this plot, lags of 1, 2 and 3 are significant, lags of 4 and 5 are marginally significant. These lags reflect the duration of the larval and pupal period (1 and 2 periods, respectively). The cycles are clearly caused by overcompensating density dependence, resulting from intraspecific competition between the larvae for food (what Nicholson called "scramble competition"). There is a curious positive feedback at a lag of 12 weeks (12−16 weeks, in fact).

```
1  10.8   6.5 12.2     1 1987
2  10.5   4.5  1.3     1 1987
3   7.5  -1.0  0.1     1 1987
4   6.5  -3.3  1.1     1 1987
5  10.0   5.0  3.5     1 1987
6   8.0   3.0  0.1     1 1987
> index <- 1:6940
> 6940/19
[1] 365.2632
> time <- index/365.2632 #time goes from about 0 to about 19.
> model5 <- lm(weather$upper~sin(time*2*pi)+cos(time*2*pi)) #Model for the seasonal cycle.
> summary(model5)

Coefficients:
                  Estimate Std. Error t value Pr(>|t|)
(Intercept)        14.95647    0.04088  365.86   <2e-16 ***
sin(time * 2 * pi) -2.53883    0.05781  -43.91   <2e-16 ***
cos(time * 2 * pi) -7.24017    0.05781 -125.23   <2e-16 ***
```

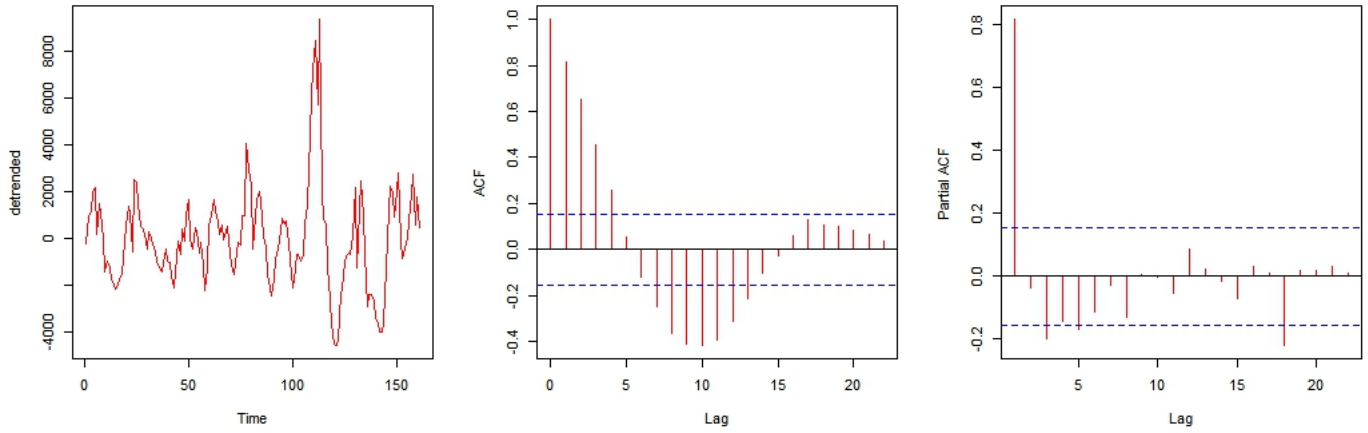

Figure 12: Plots using "detrended" data of the last 161 observations. "acf" plot shows there are still cycles, but they are weaker and less regular. They look more like damped oscillations than repeated cycles. "pacf" plot shows there are still significant negative partial autocorrelations at lags 3 and 5, and now there is a curious extra negative partial at lag 18. It looks, therefore, as if the main features of the ecology are the same (a scramble contest for food between the larvae, leading to negative partials at 3 and 5 weeks after 1 and 2 generation lags), but population size is drifting upwards and the cycles are showing a tendency to dampen out.

```
Residual standard error: 3.406 on 6937 degrees of freedom
Multiple R-squared:  0.7174,    Adjusted R-squared:  0.7173
F-statistic:  8806 on 2 and 6937 DF,  p-value: < 2.2e-16
```

```
> plot(time,weather$upper,pch=".")  # See Fig. 10
> lines(time,predict(model5),col=2,lwd=2)
```

We can look at some of the residual plots to check for any patterns (e.g., trends in the mean, or autocorrelation structure).

```
> par(mfrow=c(1,3))
> plot(model5$residuals,pch=".")  # See Fig. 11
> acf(model5$residuals)
> acf(model5$residuals,type="p",col=2)
```

Decompositions

The function `stl` performs decomposition of a time series into seasonal, trend and irregular components using `loess` (locally weighted scatterplot smoothing). First, we make a time series object, specifying the start date and the frequency, then use `stl` to decompose the series, then plot.

```
> myFlies <- ts(blowfly$flies,start=c(2001,1),frequency=52)
> plot(stl(myFlies,"periodic"))  # See Fig. 12
```

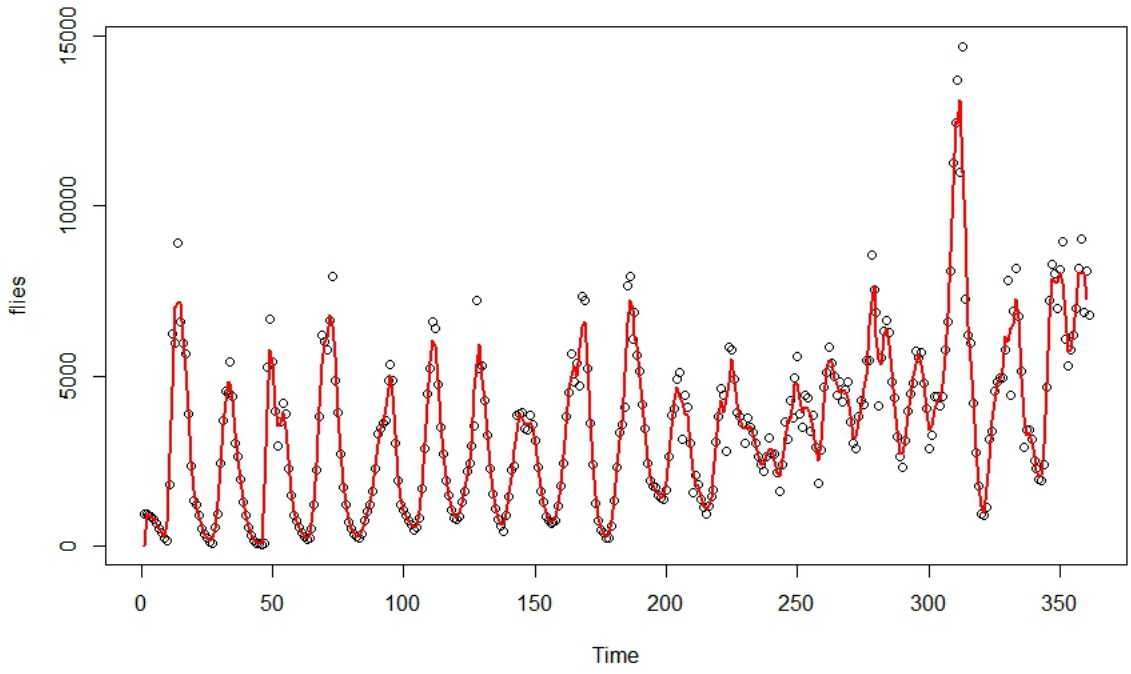

Figure 13: Data overlaid with 3-point moving averages. Note that a moving average can never capture the maxima or minima of a series (because they are averaged away).

Testing for a Trend

The trend part of the data indicates a fluctuating increase, but is it significant? We cannot test for a trend with a simple linear regression because of the massive temporal pseudoreplication. With so much temporal pseudoreplication we should use a mixed model (`lmer`). Because we want to compare two models with different fixed effects we use the method of maximum likelihood ("ML" rather than "REML"). The explanatory variable for any trend is index, and we fit the model with and without this variable, allowing for different intercepts for the different years as a random effect.

```
> myIndex <- 1:361
> myTime <- myIndex/51.57143  #51.57143 came from 361 divided by 7
> blowfly$myYear <- c(rep(c(2001:2006),each=52),rep(2007,49))
> model8 <- lmer(blowfly$flies~myIndex+sin(myTime*2*pi)+cos(myTime*2*pi)+(1|factor(blowfly$myYear)),REML=F)
> model9 <- lmer(blowfly$flies~sin(myTime*2*pi)+cos(myTime*2*pi)+(1|factor(blowfly$myYear)),REML=F)
> model7 <- lm(blowfly$flies~myIndex+sin(myTime*2*pi)+cos(myTime*2*pi))
> anova(model8,model9)
Data: NULL
Models:
model9: blowfly$flies ~ sin(myTime * 2 * pi) + cos(myTime * 2 * pi) + (1 | factor(blowfly$myYear))
model8: blowfly$flies ~ myIndex + sin(myTime * 2 * pi) + cos(myTime * 2 * pi) + (1 | factor(blowfly$myYear))
       Df    AIC    BIC  logLik deviance  Chisq Chi Df Pr(>Chisq)
model9  5 6582.1 6601.5 -3286.0   6572.1
model8  6 6565.4 6588.7 -3276.7   6553.4 18.675      1   1.55e-05 ***
```

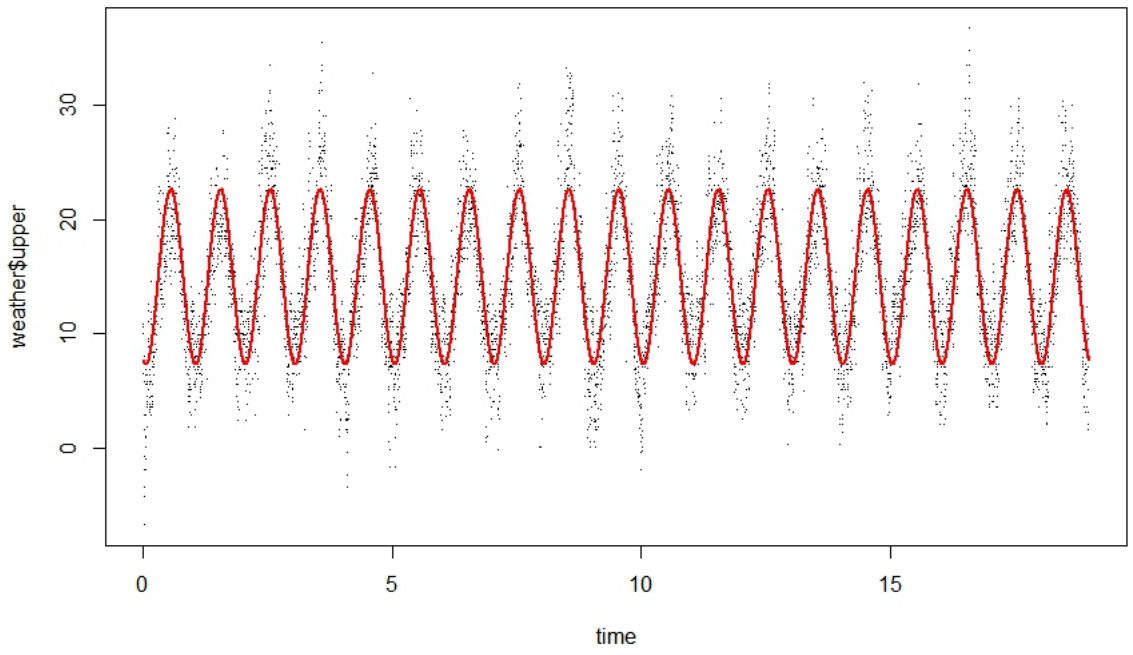

Figure 14: Plot to check seasonality. All three model terms are highly significant, thus provides a good fit.

It says the trend is highly significant.

Spectral Analysis

Another way to find the "cycle" period of a time series is by the **periodogram**, the fundamental tool of spectral analysis. Frequency is the reciprocal of cycle period. This is based on the squared correlation between the time series and sine/cosine waves of frequency ω, and it conveys exactly the same information as the autocovariance function. It sometimes make the information easier to interpret. We use the fact that "frequency" is the reciprocal of cycle period. For example, ten-year cycles would have a frequency 0.1 per year.

```
> plot.ts(blowfly$flies)
> spectrum(blowfly$flies,col=2)  # See Fig. 13
```

Fitting a Model and Diagnostics

Time series models come in three kinds (due to Box and Jenkins).

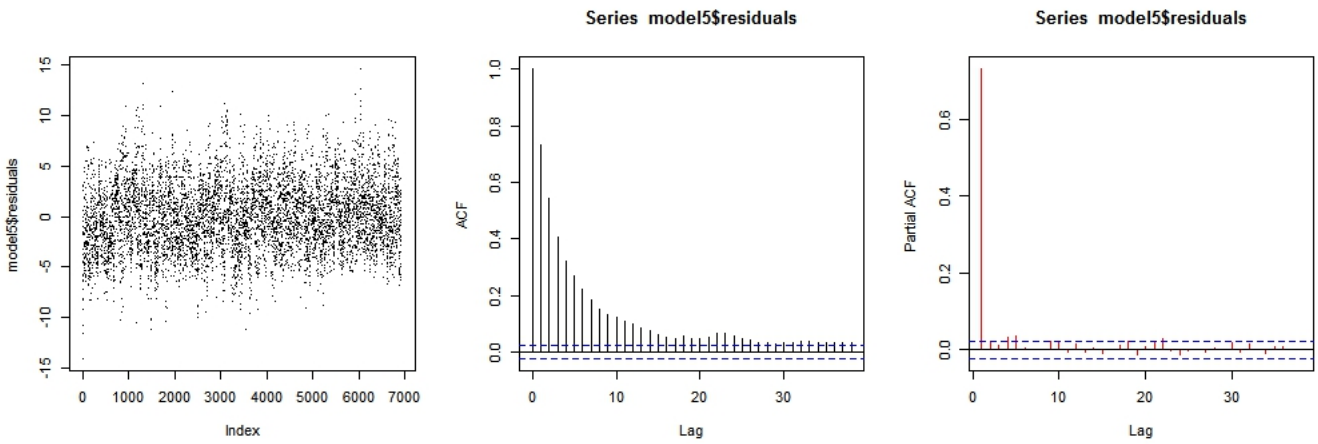

Figure 15: (1) The first residual plot shows some periodicity, but no obvious trends. (2) There is very strong serial correlation in the residuals, and this drops off fast with increasing lag (acf plot). (3) The partial autocorrelation at lag 1 is very large (0.7317), but the correlations at higher lags are much smaller. This suggests that an AR(1) model (autoregressive model with order 1) might be appropriate. This is the statistical justification behind the old joke about the weather for tomorrow would be like today's.

- autoregressive (AR) models: $\hat{y}_t = \sum_{i=1}^{p} \phi_i Y_{t-i} + \varepsilon_t$
- moving average (MA) models: $\hat{y}_t = \sum_{j=0}^{q} \theta_j \varepsilon_{t-j}$
- autoregressive moving average (ARMA) models: $\hat{y}_t = \sum_{i=1}^{p} \phi_i Y_{t-i} + \sum_{j=0}^{q} \theta_j \varepsilon_{t-j}$

A moving average of order q averages the random variation over the last q time periods. An autoregressive model of order p computes $\hat{y}_t$ as a function of the last p values of Y, so, for a second-order autoregressive process, we would use

$$\hat{y}_t = \phi_1 Y_{t-1} + \phi_2 Y_{t-2} + \varepsilon_t$$

Models are fitted using the `arima` function, and their performances are compared by AIC. The most important component of the model is "order." This is a vector of length 3 specifying the order of the autoregressive operators, the number of differences, and the order of moving average operators. For example, `order=c(1,3,2)` is based on a first-order autoregressive process, three differences, and a second-order moving average.

The order vector specifies the non-seasonal part of the ARIMA model: the three components (p, d, q) are the AR order, the degree of differencing, and the MA order. We start by investigating the effects of AR order with no differencing and no moving average terms, comparing models on

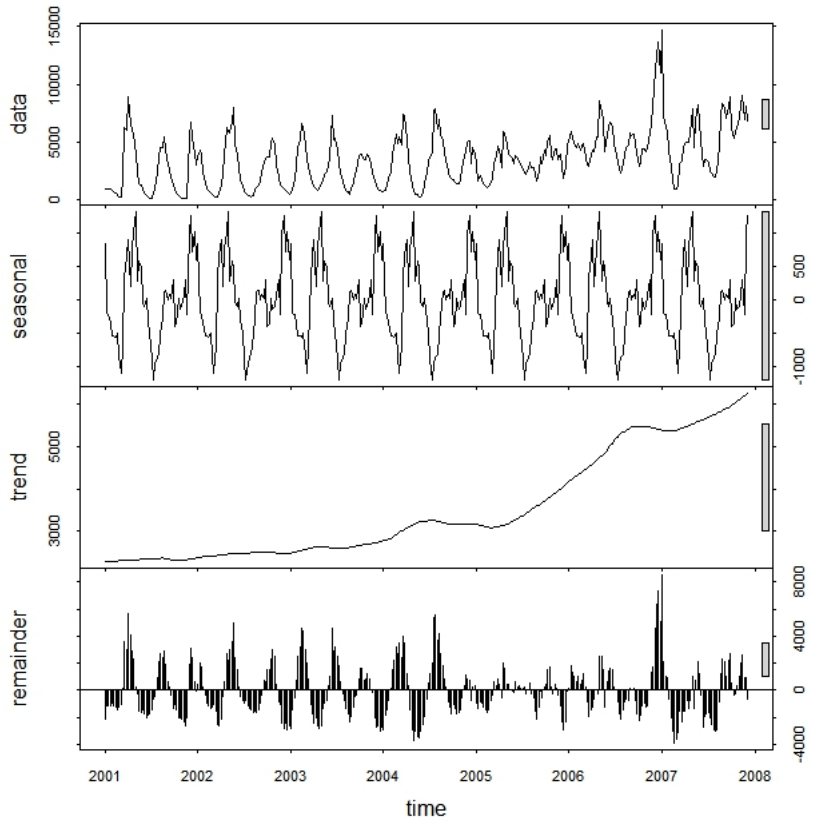

Figure 16: R's built-in decompositions

the basis of AIC.

The acronym **ARIMA** stands for "Auto-Regressive Integrated Moving Average." Lags of the stationarized series in the forecasting equation are called "autoregressive" terms, lags of the forecast errors are called "moving average" terms, and a time series which needs to be differenced to be made stationary is said to be an "integrated" version of a stationary series. Random-walk and random-trend models, autoregressive models, and exponential smoothing models are all special cases of ARIMA models.

A nonseasonal ARIMA model is written as an "ARIMA (p, d, q)" model, where:

- p is the number of autoregressive terms,
- d is the number of differences needed for stationarity, and
- q is the number of lagged forecast errors in the prediction equation.

The forecasting equation is constructed as follows. First, let y denote the dth difference of Y, which means:

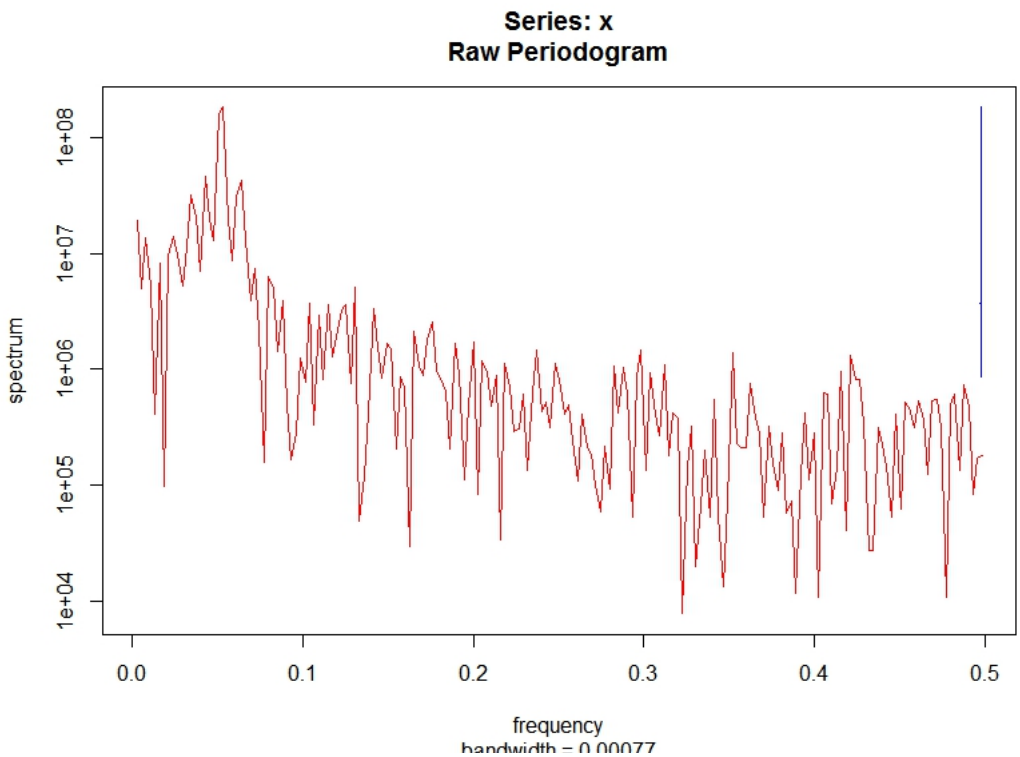

Figure 17: The plot is on a log scale, in units of decibels, and the sub-title shows the bandwidth, while the 95% confidence interval in decibels is shown by the vertical bar in the top right-hand corner. The figure shows strong cycles with a frequency of about 0.05, where the maximum value of spectrum occurs. That means cycles with a period of 1/0.05 (i.e., about 20). There is a hint of some shorter-term cycles.

- If $d = 0 : y_t = Y_t$
- If $d = 1 : y_t = Y_t - Y_{t-1}$
- If $d = 2 : y_t = (Y_t - Y_{t-1}) - (Y_{t-1} - Y_{t-2}) = Y_t - 2Y_{t-1} + Y_{t-2}$

Note that the second difference of Y (the $d = 2$ case) is not the difference from 2 periods ago. Rather, it is *the first-difference-of-the-first difference*, which is the discrete analog of a second derivative, i.e., the local acceleration of the series rather than its local trend.

In terms of y, the general forecasting equation is:

$$\hat{y}_t = \mu + \phi_1 Y_{t-1} + \ldots + \phi_p Y_{t-p} + \theta_1 \varepsilon_{t-1} + \ldots + \theta_q \varepsilon_{t-q}$$

ARIMA $(1, 0, 0)$ = first-order autoregressive model: if the series is stationary and autocorrelated, it can be predicted as a multiple of its own previous value, plus a constant. The forecasting equation in this case is $\hat{y}_t = \mu + \phi_1 Y_{t-1}$, which is Y regressed on itself lagged by one period. This is an "ARIMA $(1, 0, 0)$+ constant" model. If the mean of Y is zero, then the constant term would not be included. In a second-order autoregressive model ARIMA $(2, 0, 0)$, there would be a Y_{t-2} term

on the right as well, and so on.

ARIMA $(0,1,0)$ = random walk: If the series Y is not stationary, the simplest possible model for it is a random walk model. The prediction equation for this model is written as $\hat{y}_t - Y_{t-1} = \mu$ or equivalently $\hat{y}_t = \mu + Y_{t-1}$, which can be considered as a limiting case of an AR(1) model in which the autoregressive coefficient is equal to 1. Since it includes (only) a nonseasonal difference and a constant term, it is classified as an "ARIMA $(0,1,0)$ model with constant."

ARIMA $(1,1,0)$ = differenced first-order autoregressive model: If the errors of a random walk model are autocorrelated, perhaps the problem can be fixed by adding one lag of the dependent variable to the prediction equation, i.e., by regressing the first difference of Y on itself lagged by one period. This would yield the following prediction equation: $\hat{y}_t - Y_{t-1} = \mu + \phi_1 (Y_{t-1} - Y_{t-2})$, which can be rearranged to $\hat{y}_t = \mu + Y_{t-1} + \phi_1 (Y_{t-1} - Y_{t-2})$.

We now demonstrate how to fit an ARIMA model to the sample data. First, try with AR models.

```
> model10 <- arima(blowfly$flies,order=c(1,0,0))
> model20 <- arima(blowfly$flies,order=c(2,0,0))
> model30 <- arima(blowfly$flies,order=c(3,0,0))
> model40 <- arima(blowfly$flies,order=c(4,0,0))
> model50 <- arima(blowfly$flies,order=c(5,0,0))
> model60 <- arima(blowfly$flies,order=c(6,0,0))
> AIC(model10,model20,model30,model40,model50,model60)
        df      AIC
model10  3 6104.922
model20  4 6097.490
model30  5 6082.210
model40  6 6080.320
model50  7 6076.922
model60  8 6077.192
```

Among these AR models, it appears that order 5 is best (smallest $AIC = 6076.922$). Next, we check various MA models.

```
> model51 <- arima(blowfly$flies,order=c(5,0,1))
> model52 <- arima(blowfly$flies,order=c(5,0,2))
> model53 <- arima(blowfly$flies,order=c(5,0,3))
> model54 <- arima(blowfly$flies,order=c(5,0,4))
> model55 <- arima(blowfly$flies,order=c(5,0,5))
> model56 <- arima(blowfly$flies,order=c(5,0,6))
> model57 <- arima(blowfly$flies,order=c(5,0,7))
> AIC(model51,model52,model53,model54,model55,model56,model57)
        df      AIC
model51  8 6077.506
model52  9 6053.722
model53 10 6075.154
model54 11 6067.374
model55 12 6077.402
model56 13 6070.184
```

```
model57 14 6071.244
```

Among these ARMA models, it appears that order 2 is best (smallest $AIC = 6053.722$). Finally, we check differencing.

```
> model502 <- arima(blowfly$flies,order=c(5,0,2))
> model512 <- arima(blowfly$flies,order=c(5,1,2))
> model522 <- arima(blowfly$flies,order=c(5,2,2))
> model532 <- arima(blowfly$flies,order=c(5,3,2))
> model542 <- arima(blowfly$flies,order=c(5,4,2))
> AIC(model502,model512,model522,model532,model542)
         df      AIC
model502  9 6053.722
model512  8 6032.167
model522  8 6035.608
model532  8 6092.950
model542  8 6154.759
```

Among these models, the model with "1" differencing performs best (smallest $AIC = 6032.167$). Having decided on a model, we look at various diagnostic statistics and plots.

```
> predicted = blowfly$flies-model512$residuals
> plot(blowfly$flies,type="l")
> lines(predicted,type="l",lty=2,lwd=2,col=2) # See Fig. 14
> tsdiag(model512) # See Fig. 15
> Box.test(model512$residuals) #Box$-$Pierce or Ljung$$Box test
> # Test for the null hypothesis of independence in a given time series.
        Box-Pierce test

data:  model512$residuals
X-squared = 0.040833, df = 1, p-value = 0.8399
> model512

Call:
arima(x = blowfly$flies, order = c(5, 1, 2))

Coefficients:
         ar1      ar2      ar3     ar4      ar5      ma1     ma2
      1.5710  -0.5725  -0.1423  0.1082  -0.1012  -1.6984  0.7134
s.e.  0.0948   0.1196   0.1029  0.0997   0.0622   0.0823  0.0797

sigma^2 estimated as 1051719:  log likelihood = -3008.08,  aic = 6032.17
> predicted = blowfly$flies-model512$residuals
> par(mfrow=c(1,2))
> plot(model512$residuals~predicted) #Plot of residuals vs. predicted
> qqnorm(model512$residuals); qqline(model512$residuals) #normal QQ plot of residuals
```

From the estimated coefficients and their standard errors, only the first two AR terms and MA terms seem significant, which means appropriate $p = 2$ and $q = 2$. From the principle of model parimony, we can compare ARIMA $(2, 1, 2)$ and $(2, 0, 2)$ models with model512 above, i.e., ARIMA $(5, 1, 2)$ before finalizing our choice.

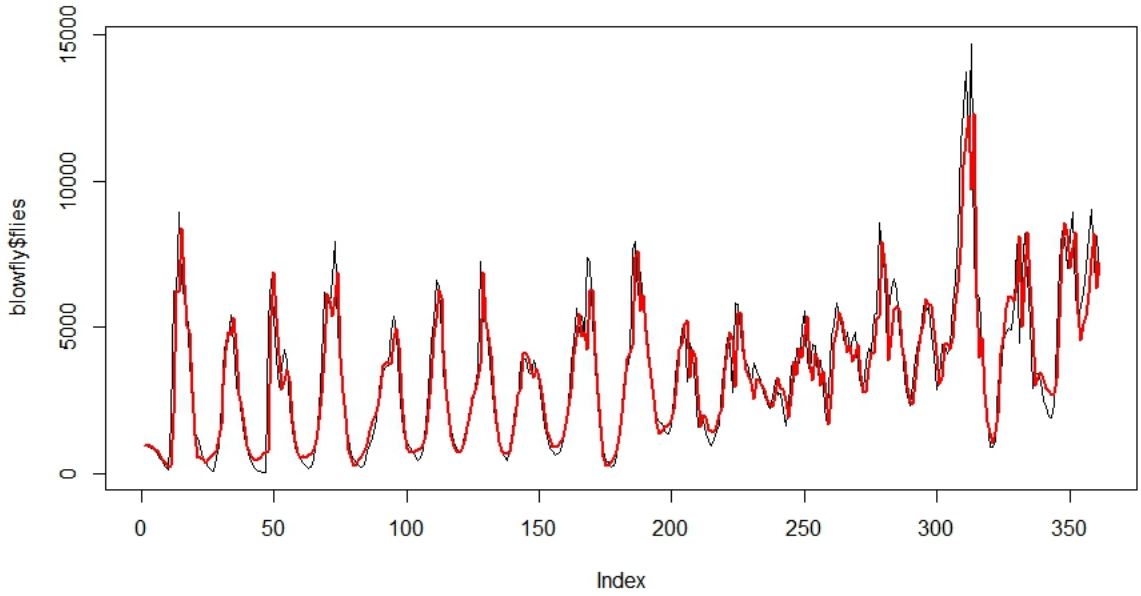

Figure 18: Plot of raw data overlaid with ARIMA(5,1,2) model.

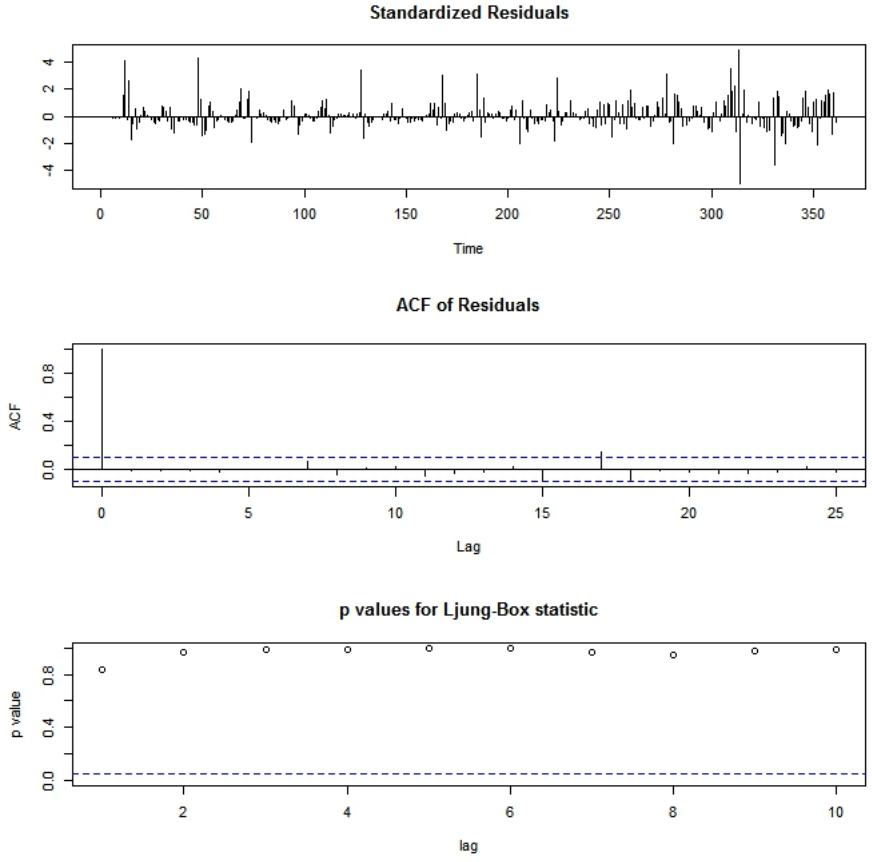

Figure 19: Diagnostic plots of residuals from ARIMA(5,1,2) model.