Mathematical Modeling of Infectious Diseases, UCSF

Lecture 4

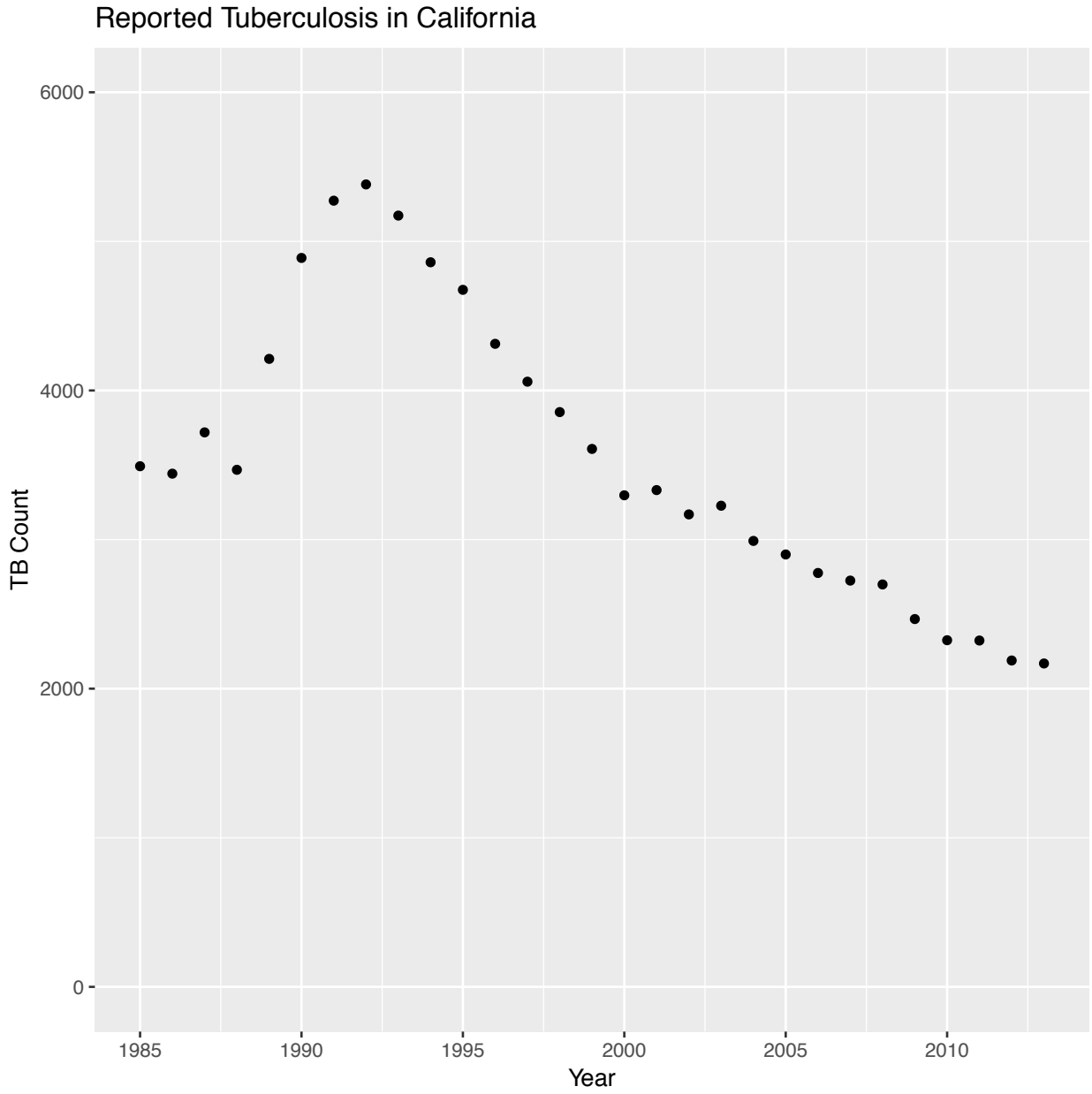
Version 1.1—modified 7 Feb 2017

This week, we're going to take a step back and use some of what we learned to look at data. At the end we'll do a little more review and preparation for next week's model: the Galton-Watson process.

Holt-Winters forecasting

The following data are the reported yearly case counts for Tuberculosis in California since 1985.

```
qplot(tbh[,1],tbh[,2],ylim=c(0,6000),xlab="Year",ylab="TB Count",main="Reported Tuberculosis in Califor
```



We'd like to produce some forecasts for the next year. One idea is that recent events are more important or relevant than things in the far past. Obviously this isn't true in general. It's more a rule of thumb that's often helpful.

Our goal is to look at some simple ways to predict next year's case count, based on the time series.

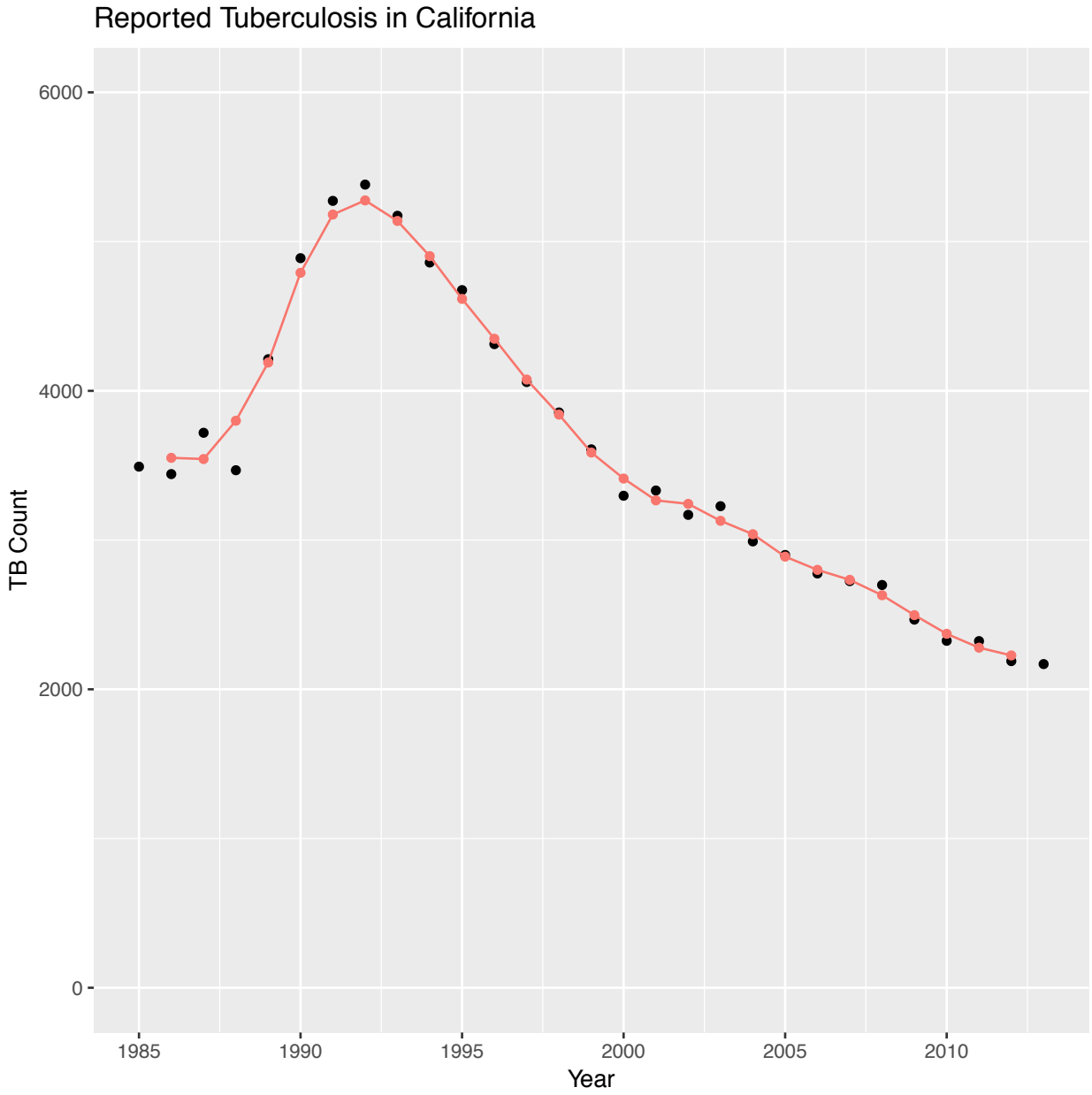
The first, and most naive thing we could do, would be to just say next year's will be equal to this year's, or to the average of the last few years. Of course that ignores the trend. So we might take a couple of recent

points (how many?) and draw a line thru them, and just extrapolate forward. We might feel this doesn't make good use of the available information.

We could take moving averages; R provides a function called `filter` for this. For a three point centered moving average, we do this. We show the centered moving average in red.

$$y_i = \frac{x_{i-1} + x_i + x_{i+1}}{3}$$

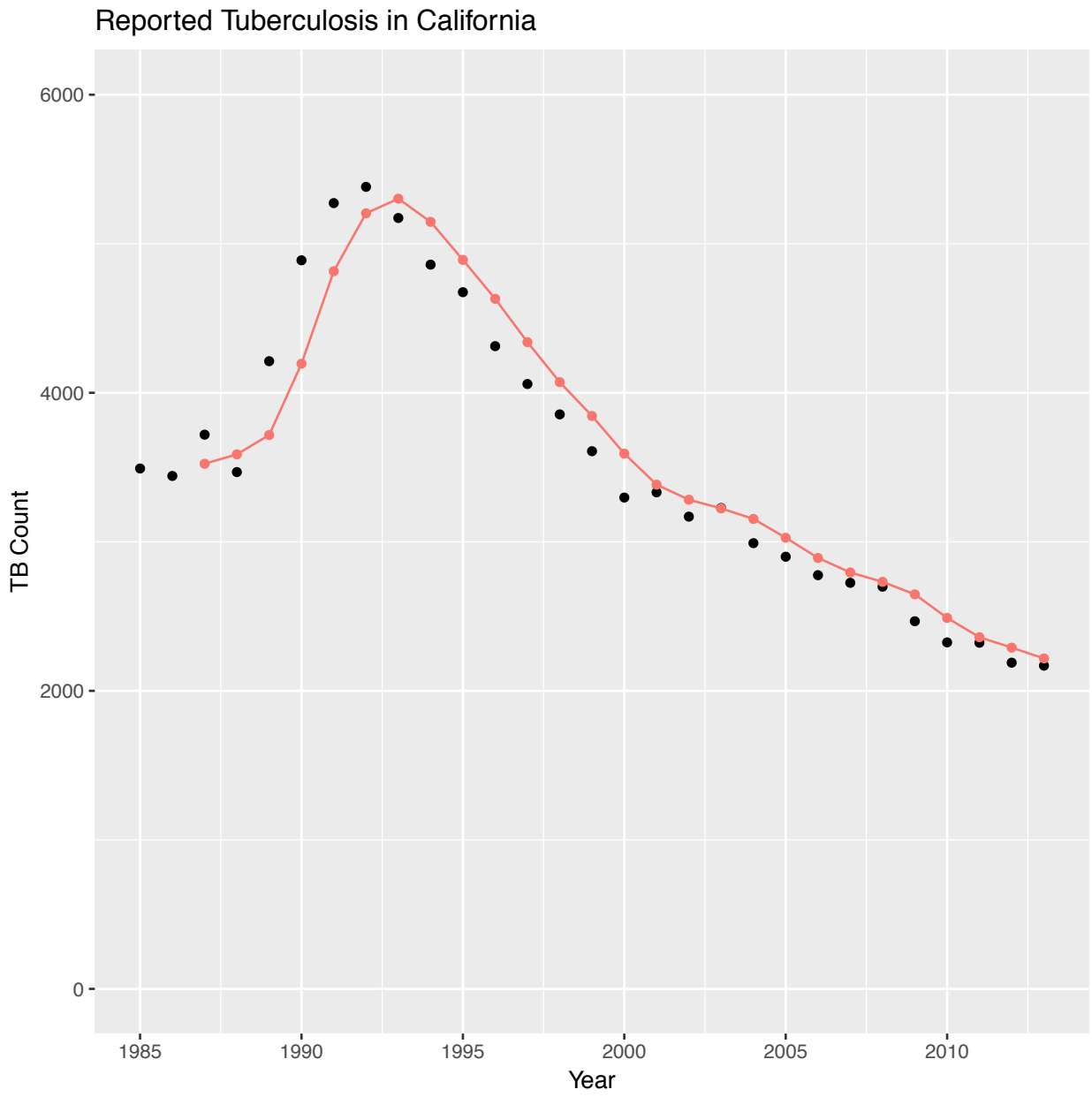
```
qplot(tbh[,1],tbh[,2],ylim=c(0,6000),xlab="Year",ylab="TB Count",main="Reported Tuberculosis in Califor
```



This does smooth out irregularities in the sequence and is useful for some things.

Other kinds of centered moving averages can be conducted as well. For instance, we could keep our window of 3, say, but pick different weights. We could, for instance, pick weights of 1/4, 1/2, and 1/4. This way, we weight the point at the "cursor" so to speak more heavily than those surrounding it. Many choices of smoothing filter are possible.

```
qplot(tbh[,1],tbh[,2],ylim=c(0,6000),xlab="Year",ylab="TB Count",main="Reported Tuberculosis in Califor
```



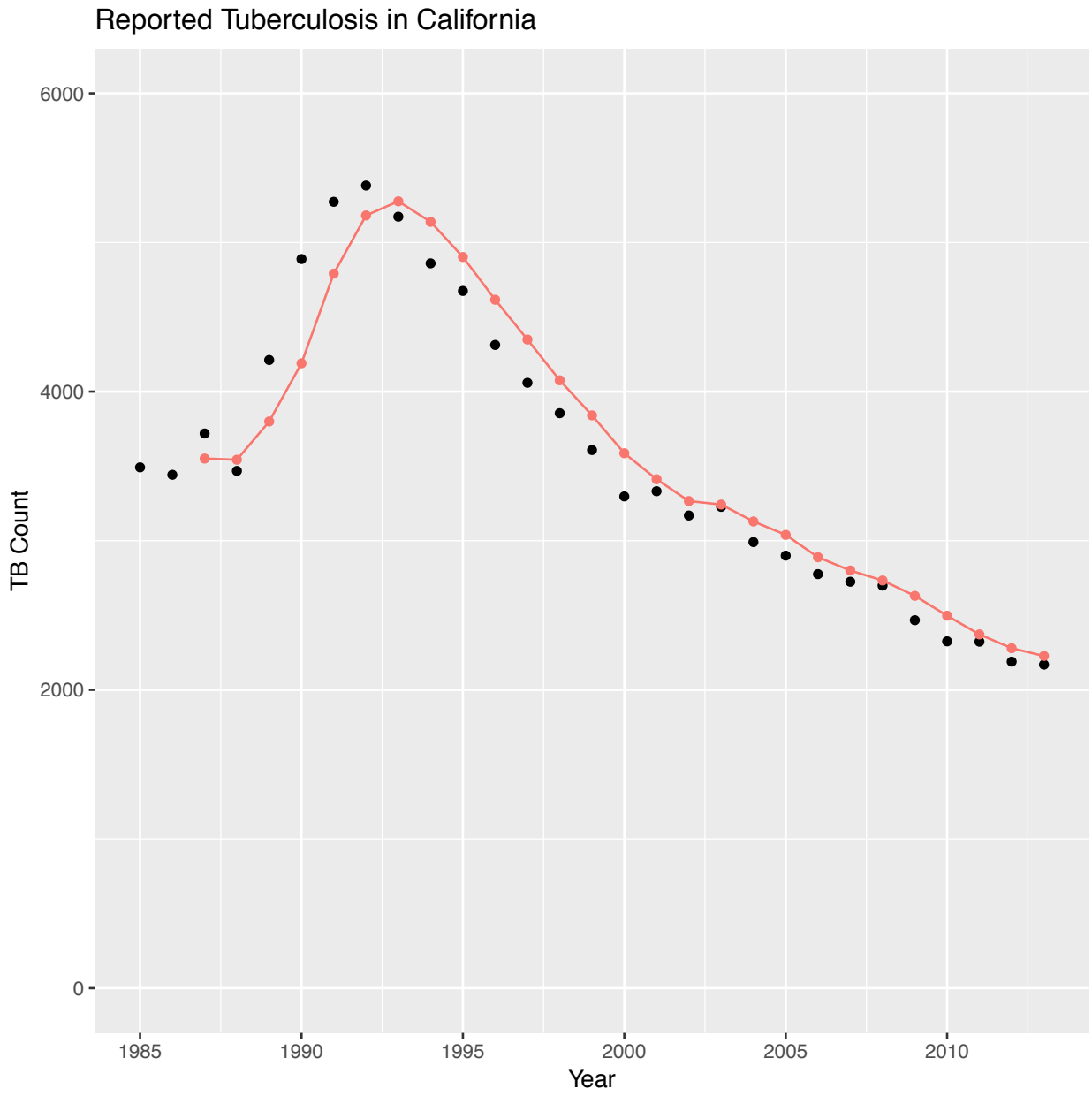
In R, these are implemented using the `filter` command.

But it isn't so great for forecasting since it isn't, strictly speaking, available at the ends. We could work around that.

One idea is to just use the last three. This is a moving average again; at each time point, we use the

average of the current value and the previous two.

```
qplot(tbh[,1],tbh[,2],ylim=c(0,6000),xlab="Year",ylab="TB Count",main="Reported Tuberculosis in Califor
```



Now, when it's rising consistently, the data are above their three year moving average; when it's falling, the data are below the moving average.

I'm going to follow the discussion in Harvey closely.

Exponentially weighted moving average. One idea that tries to use more of the data but still to weight recent data more heavily is called the exponentially weighted moving average. Remember our geometric series:

$$\sum_{i=0}^{\infty} a^i = \frac{1}{1-a}$$

If we use weights $(1-a)a^i$, they will actually add up to 1 if we had an infinite amount of data going back into past eternity. But in practice we can often go far enough back to where this is so close to 1 the difference is irrelevant to anything. Let $\lambda = 1 - a$.

Then we have a moving average estimate of the current "true value" or level:

$$\mu_\tau = \lambda x_\tau + \lambda(1-\lambda)x_{\tau-1} + \lambda(1-\lambda)^2 x_{\tau-2} + \cdots = \lambda \sum_{i=0}^{T} (1-\lambda)^i x_{\tau-i}.$$

If T is big enough those weights will sum to nearly 1 and be good enough.

One nice thing about this is that we can construct a recursion. let's start with this:

$$\mu_\tau = \lambda x_\tau + \lambda \sum_{i=1}^{T} (1-\lambda)^i x_{\tau-i}.$$

Let's factor out a $(1-\lambda)$ from the summation:

$$\mu_\tau \lambda x_\tau + \lambda(1-\lambda) \sum_{i=1}^{i=T} (1-\lambda)^{(i-1)} x_{\tau-i}.$$

Then, let's make this substitution: $j = i - 1$, so we can start the summation at 0. If $i = 1$, $j = 0$, so we start at 0 as advertised. We end at $i = T$, so this is $j + 1 = T$, which means j goes to $T - 1$:

$$\lambda x_\tau + (1-\lambda)\lambda \sum_{j=0}^{T-1} (1-\lambda)^j x_{\tau-(j+1)}.$$

Look at this piece:

$$\sum_{j=0}^{T-1} (1-\lambda)^j x_{\tau-(j+1)}.$$

This is $\mu_{\tau-1}$ (remember T is supposed to be big, so big that $T - 1$ is basically T). So what we have is

$$\mu_\tau = \lambda x_\tau + (1-\lambda)\mu_{\tau-1}.$$

In exponential smoothing, we have our running estimate of the "level" or true value, μ_τ. As new data come in, we take a weighted average of the new data with the current running average.

Let's watch this in a simple example. Say we have this simple time series: 12, 0, 0, 0, 0, 0, 0, and so on. Start with $\mu_0 = x_0 = 12$, and let's use $\lambda = 1/3$ say (in practice the choice of λ depends on data, experience, and so forth—in a formal time series class you would learn more about it). What is μ_1?

$$\mu_1 = 1/3 \times 0 + (2/3) \times 12 = 8$$

Now on to the next:

$$\mu_2 = 1/3 \times 0 + (2/3) \times 8 = 5.33$$

And

$$\mu_2 = 1/3 \times 0 + (2/3) \times (16/3) = 3.56$$

It is gradually forgetting about the 12, the further back in time it becomes (so to speak). I emphasize that normally this is the sort of thing that makes sense in a longer time series.

Let's do a more realistic worked example. Suppose we start with the CA tuberculosis data for year 2003 and after. The successive values are 3227, 2991, 2900, 2776, 2725, and on (based on the 2015 reports). Here, we will let τ range from 1 to 5, and so $x_1 = 3227$, $x_2 = 2991$, and so on. We will take our starting estimate of the level to be the first observed value, which here is 3227. So

$$\mu_1 = 3227.$$

Now, let's update:

$$\mu_2 = \lambda x_2 + (1 - \lambda)\mu_1.$$

What is λ? That is our smoothing parameter; we can pick anything between 0 and 1. Say we pick 1. If we do that, $\mu_2 = x_2$; that's kind of dull. If $\lambda = 1$, we don't care about the past at all; we're just tracking the current observed value. Say we pick 0 for λ. Now, $\mu_2 = \mu_1$; if we pick $\lambda = 0$, we never learn anything new—the level never changes from the starting value. We want something in between, and the best value might require experience, data, and not be obvious. Let us pick the value 1/3 say. Now,

$$\mu_2 = (1/3) \times 2991 + (2/3) \times 3227 = 3148.333$$

(to three decimals; I'm going to just report everything to 3 digits. I'm doing the calculations in double precision and only rounding off when reporting them, so if you're following along, you might get slightly different results at each step).

The next step is then

$$\mu_3 = (1/3) \times 2900 + (2/3) \times 3148.333$$

which is 3065.556.

If we were forecasting, we could just use μ_τ, our estimated level, as the forecast for the next observation. But if the data were consistently rising, what is going to happen is we're going to be undercalling it. If there is a steadily rising trend, we expect next year's result to be a bit bigger than this year's. What we really need is a way to take that into account.

Holt-Winters. Again we follow the notation in Harvey. Suppose we extend the simple method above by having an estimated level μ_τ and an estimated slope b_τ at each time. The idea is to forecast the future as a linear trend:

$$\hat{x}_{\tau+1|\tau} = \mu_\tau + b_\tau.$$

We update and get the new estimated level for time τ by taking the last forecast of it (from one step ago) and making a weighted average of this forecast and the new data:

$$\mu_\tau = \lambda_0 x_\tau + (1-\lambda_0)\hat{x}_{\tau|\tau-1}.$$

Here,

$$\hat{x}_{\tau|\tau-1} = \mu_{\tau-1} + b_{\tau-1}.$$

This is a little different in detail from the simple exponential smoothing example, but the same in spirit: we are updating our level estimate μ_τ by taking a weighted combination of an old estimate and the new data. At time $\tau-1$, I had an estimate of the level at time $\tau-1$ and the rate of change, and my estimate *then* of what the level would be at time τ was $\mu_\tau + b_\tau$, and *that* is what I use in the average.

But we should also be learning new things about the slope. Shouldn't we update the slope also, as new data come in? To update the slope, take the past estimate $b_{\tau-1}$ and make a different weighted average of this slope with the change in the level:

$$b_\tau = \lambda_1(\mu_\tau - \mu_{\tau-1}) + (1-\lambda_1)b_{\tau-1}.$$

Traditionally, we take $\mu_2 = x_2$, $b_2 = x_2 - x_1$. Here λ_0 and λ_1 are smoothing constants. This is known as the *Holt-Winters* procedure (discussion follows that in Harvey). Thus, the Holt-Winters forecast method is based on simple linear recursions for the mean and slope.

Let's go back to the same TB data and try it. Before we did simple exponential smoothing, with just a λ. But this is different; we need two smoothing parameters. Let's pick $\lambda_0 = 1/4$ and $\lambda_1 = 0.4$. We will take $\mu_2 = 2991$, and $b_2 = 2991 - 3227 = -236$. When the recursions start, we have a level of 2991 (from our second observation), and a starting trend guess of -236 (down 236 every year). So what is our first forecast?

$$\hat{x_{3|2}} = \mu_2 + b_2$$

This is going to just be 2991 - 236 = 2755. At time 2, I take what we think the level is and what we think the slope is, and we project forward.

Now, let's do the recurrence. First we update the level by taking the weighted average of our current *forecast* and the new data, $x_3 = 2900$:

$$\mu_3 = (1/4) \times 2900 + (3/4) \times 2755 = 2791.25$$

Then, we update the slope. The new information about the slope is the difference between the levels, so we have 2791.25 - 2755 = 36.25 . So:

$$b_3 = 0.4 \times (2791.25 - 2755) + 0.6 \times (-236) = -127.1$$

Early on, our starting values play a large role, but after a while, they start to disappear into the past and to no longer make a big difference. One does not really expect this procedure to work well in a very short series. The R built-in `HoltWinters` chooses the smoothers for you to minimize a sum of squared error, but in practice you may need to experiment with it in various ways. Formal time series procedures are beyond our scope in this class.

Exercise 1. Let $\hat{e}_\tau$ be the one-step forecast error:

$$\hat{e}_\tau = x_\tau - \hat{x}_{\tau|\tau-1}.$$

Show the forecast can be conducted using these equations:

$$\mu_\tau = \mu_{\tau-1} + b_{\tau-1} + \lambda_0 \hat{e}_\tau$$

$$b_\tau = b_{\tau-1} + \lambda_0 \lambda_1 \hat{e}_\tau$$

∎

Exercise 2. Show that if the data x_t lie along a straight line, the Holt-Winters forecasts will follow the line perfectly. ∎

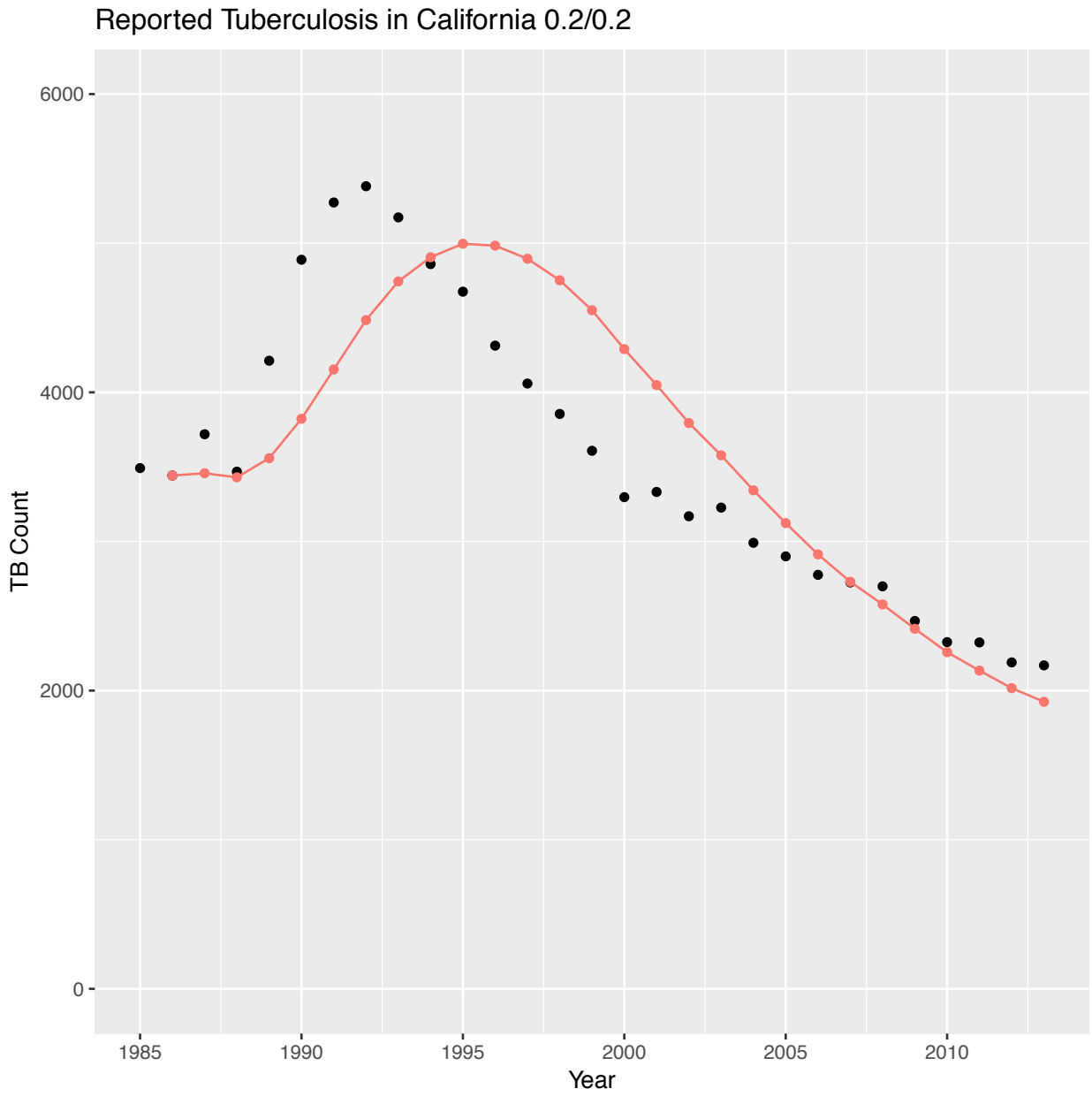
Let's run the Holt-Winters filter on the tuberculosis data and see what we get. R has a builtin function called `HoltWinters` which we want to think about, but let's do it by hand first.

```
hw.filter <- function(xs,lam0,lam1,forecasttime=NA) {
  lx <- length(xs)
  if (lx < 3) {
    stop("xs too short")
  }
```

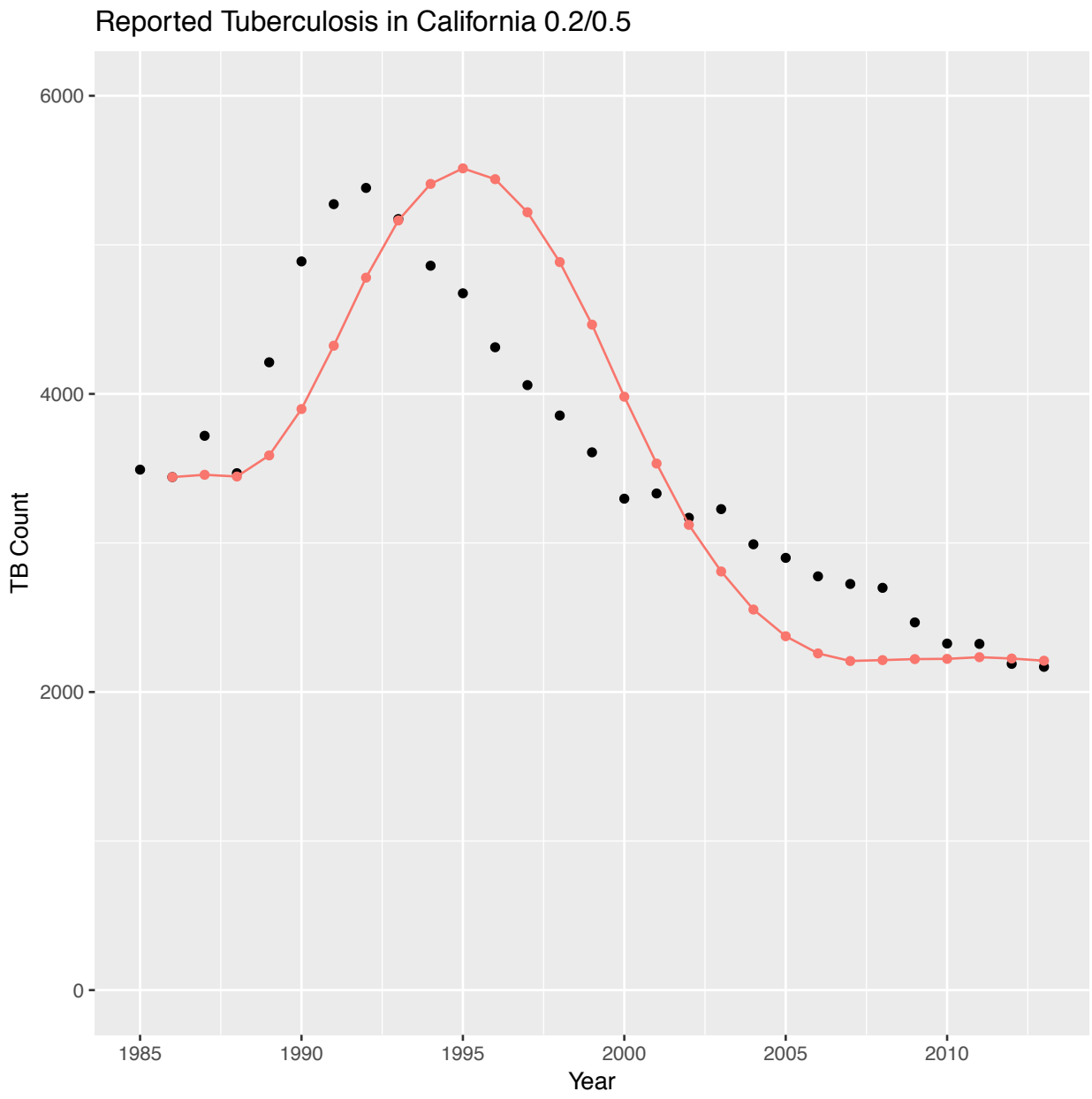
```
  mu <- rep(NA,lx)
  bb <- rep(NA,lx)
  mu[2] <- xs[2]
  bb[2] <- xs[2]-xs[1]
  for (ii in 3:lx) {
    ehat <- xs[ii] - mu[ii-1] - bb[ii-1]
    mu[ii] <- mu[ii-1]+bb[ii-1]+lam0*ehat
    bb[ii] <- bb[ii-1]+lam0*lam1*ehat
  }
  if (is.na(forecasttime)) {
    mu
  } else {
    mu[lx]+forecasttime*bb[lx]
  }
}
```

Let's try a few.

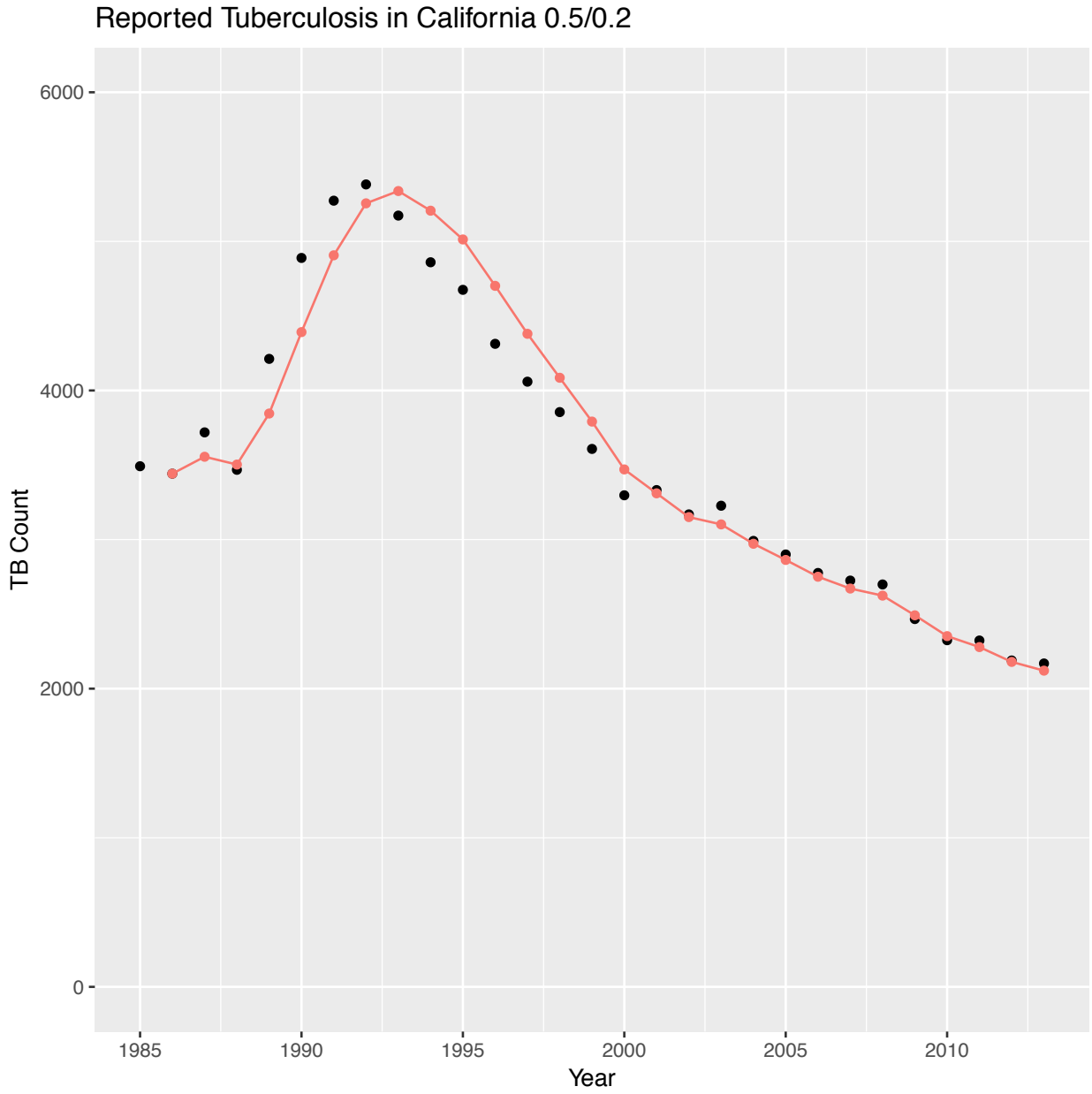
```
tbh$hw22 <- hw.filter(tbh[,2],0.2,0.2)
tbh$hw55 <- hw.filter(tbh[,2],0.5,0.5)
tbh$hw88 <- hw.filter(tbh[,2],0.8,0.8)
tbh$hw25 <- hw.filter(tbh[,2],0.2,0.5)
tbh$hw58 <- hw.filter(tbh[,2],0.5,0.8)
tbh$hw82 <- hw.filter(tbh[,2],0.8,0.2)
tbh$hw28 <- hw.filter(tbh[,2],0.2,0.8)
tbh$hw52 <- hw.filter(tbh[,2],0.5,0.2)
tbh$hw85 <- hw.filter(tbh[,2],0.8,0.5)
qplot(tbh[,1],tbh[,2],ylim=c(0,6000),xlab="Year",ylab="TB Count",main="Reported Tuberculosis in Califor
```



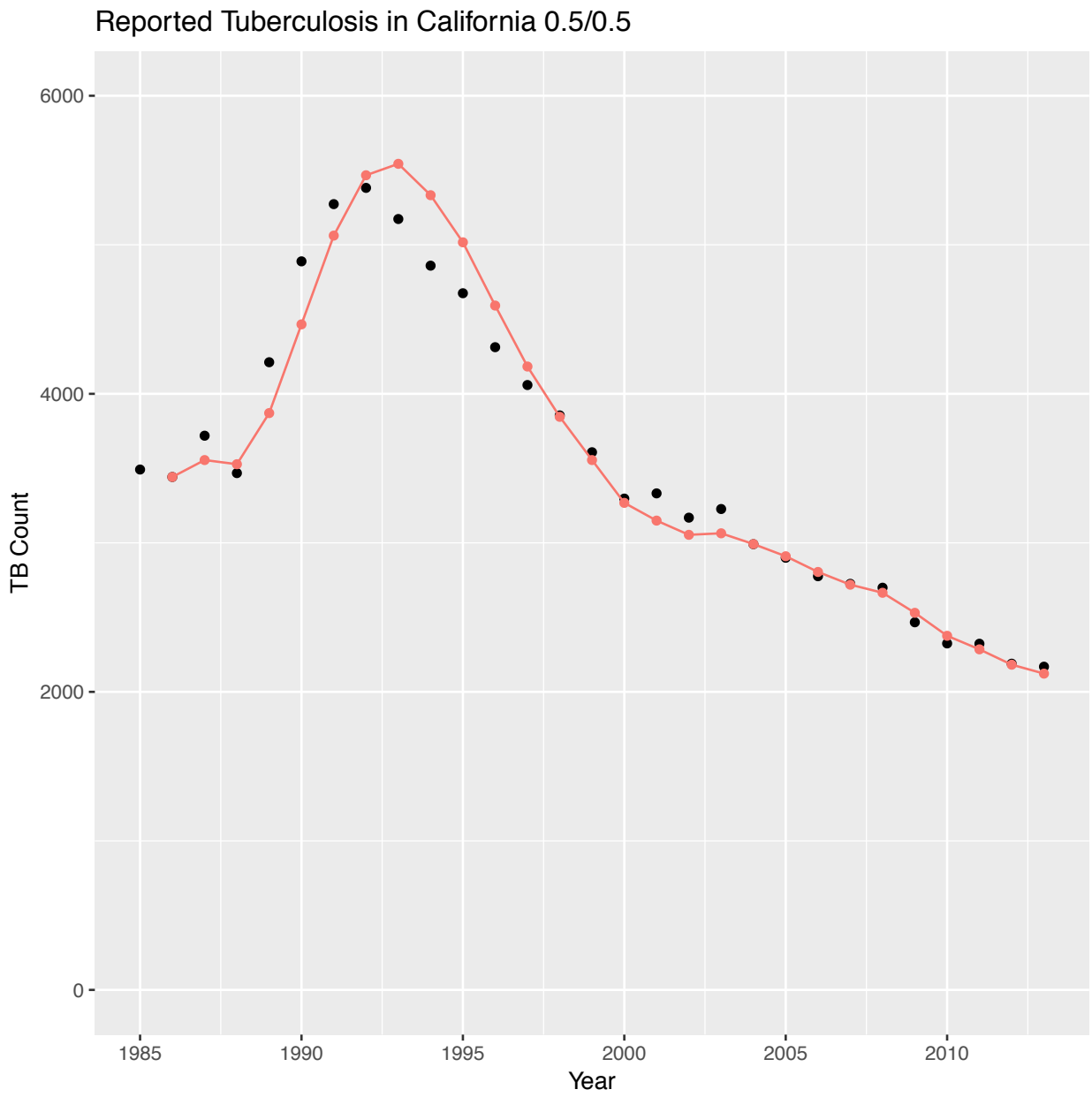

```
qplot(tbh[,1],tbh[,2],ylim=c(0,6000),xlab="Year",ylab="TB Count",main="Reported Tuberculosis in Califor
```




```
qplot(tbh[,1],tbh[,2],ylim=c(0,6000),xlab="Year",ylab="TB Count",main="Reported Tuberculosis in Califor
```



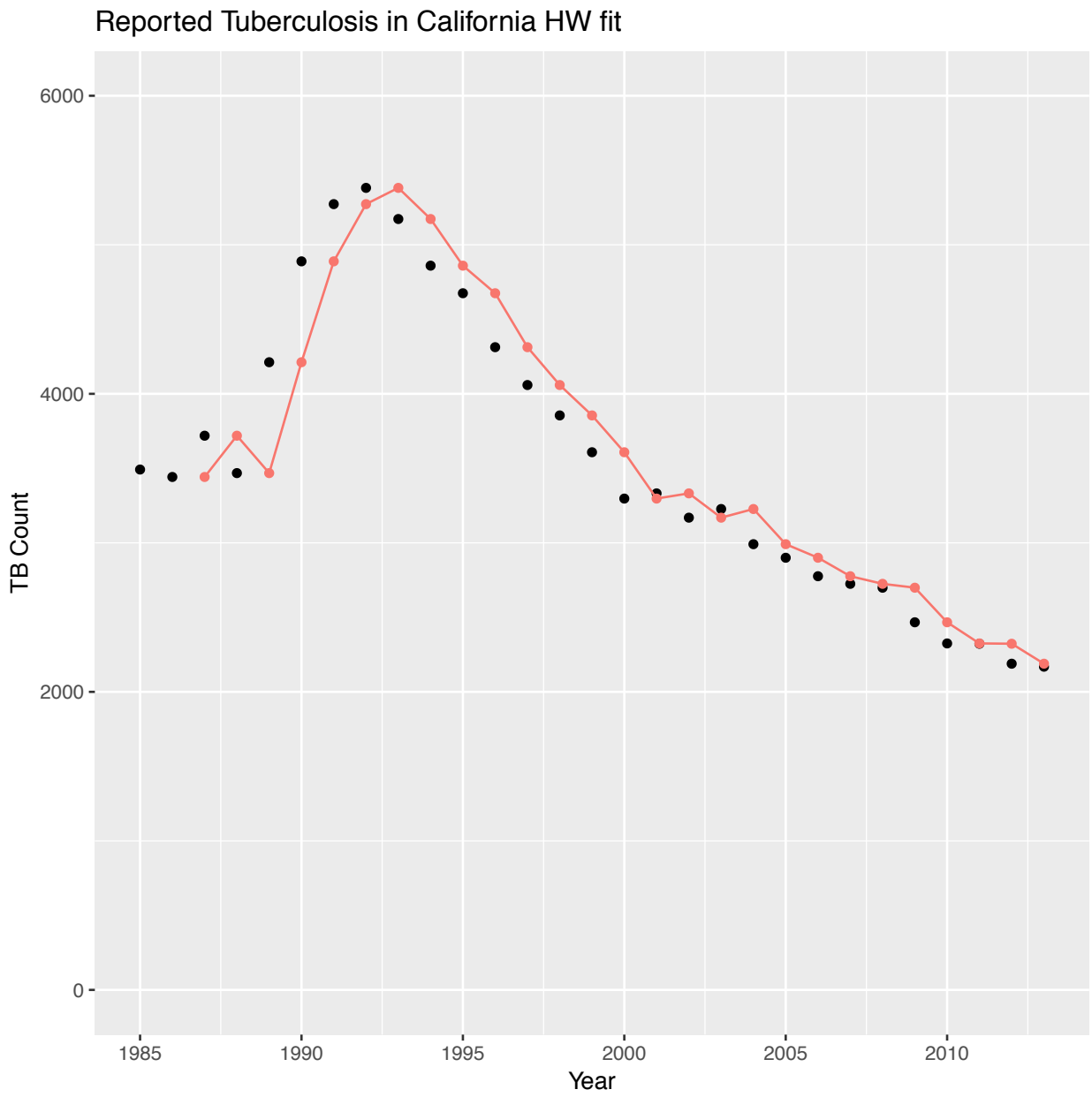

```
qplot(tbh[,1],tbh[,2],ylim=c(0,6000),xlab="Year",ylab="TB Count",main="Reported Tuberculosis in Califor
```



Let's see how the builtin function works. As alluded do above, it chooses what it thinks the best values are for you.

```
hwf <- HoltWinters(tbh[,2],gamma=FALSE)
hwf

## Holt-Winters exponential smoothing with trend and without seasonal component.
```

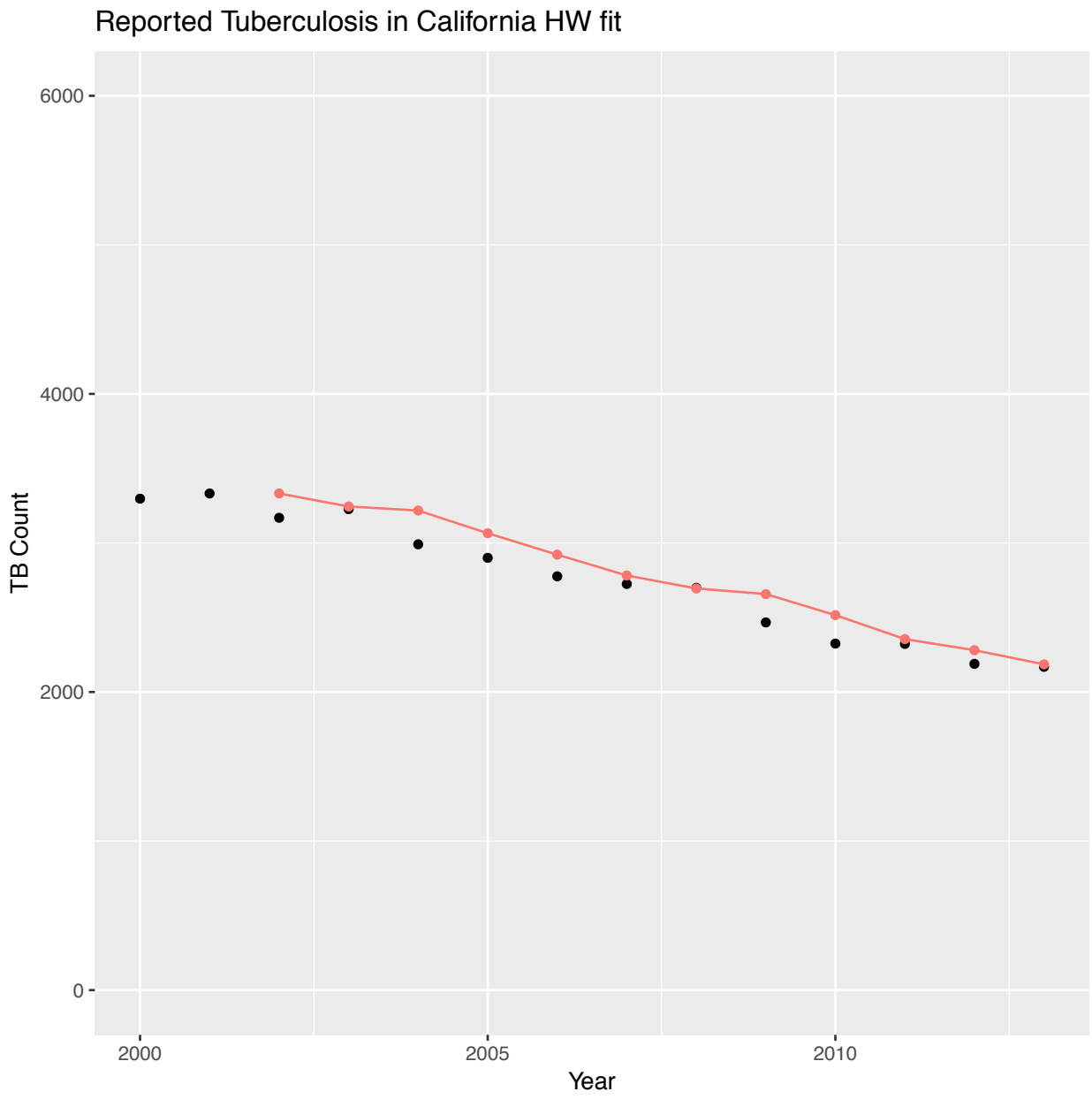
```
##
## Call:
## HoltWinters(x = tbh[, 2], gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 1
##  beta : 0
##  gamma: FALSE
##
## Coefficients:
##   [,1]
## a 2169
## b  -50

tmp <- cbind(tbh[,1],tbh[,2],tbh[,1])
tmp[1:2,3] <- NA
tmp[3:dim(tmp)[1],3] <- hwf$fitted[,"level"]
tmp <- as.data.frame(tmp)
qplot(tmp[,1],tmp[,2],ylim=c(0,6000),xlab="Year",ylab="TB Count",main="Reported Tuberculosis in Califor
```



What if we fit using only since the year 2000?

```
hwf2 <- HoltWinters(tbh[16:29,2],gamma=FALSE)
hwf2

## Holt-Winters exponential smoothing with trend and without seasonal component.
##
```

```
## Call:
## HoltWinters(x = tbh[16:29, 2], gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.6179972
##  beta : 0.6317919
##  gamma: FALSE
##
## Coefficients:
##         [,1]
## a 2138.61761
## b  -65.42894

tmp <- cbind(tbh[16:29,1],tbh[16:29,2],tbh[16:29,1])
tmp[1:2,3] <- NA
tmp[3:dim(tmp)[1],3] <- hwf2$fitted[,"level"]
tmp <- as.data.frame(tmp)
qplot(tmp[,1],tmp[,2],ylim=c(0,6000),xlab="Year",ylab="TB Count",main="Reported Tuberculosis in Califor
```

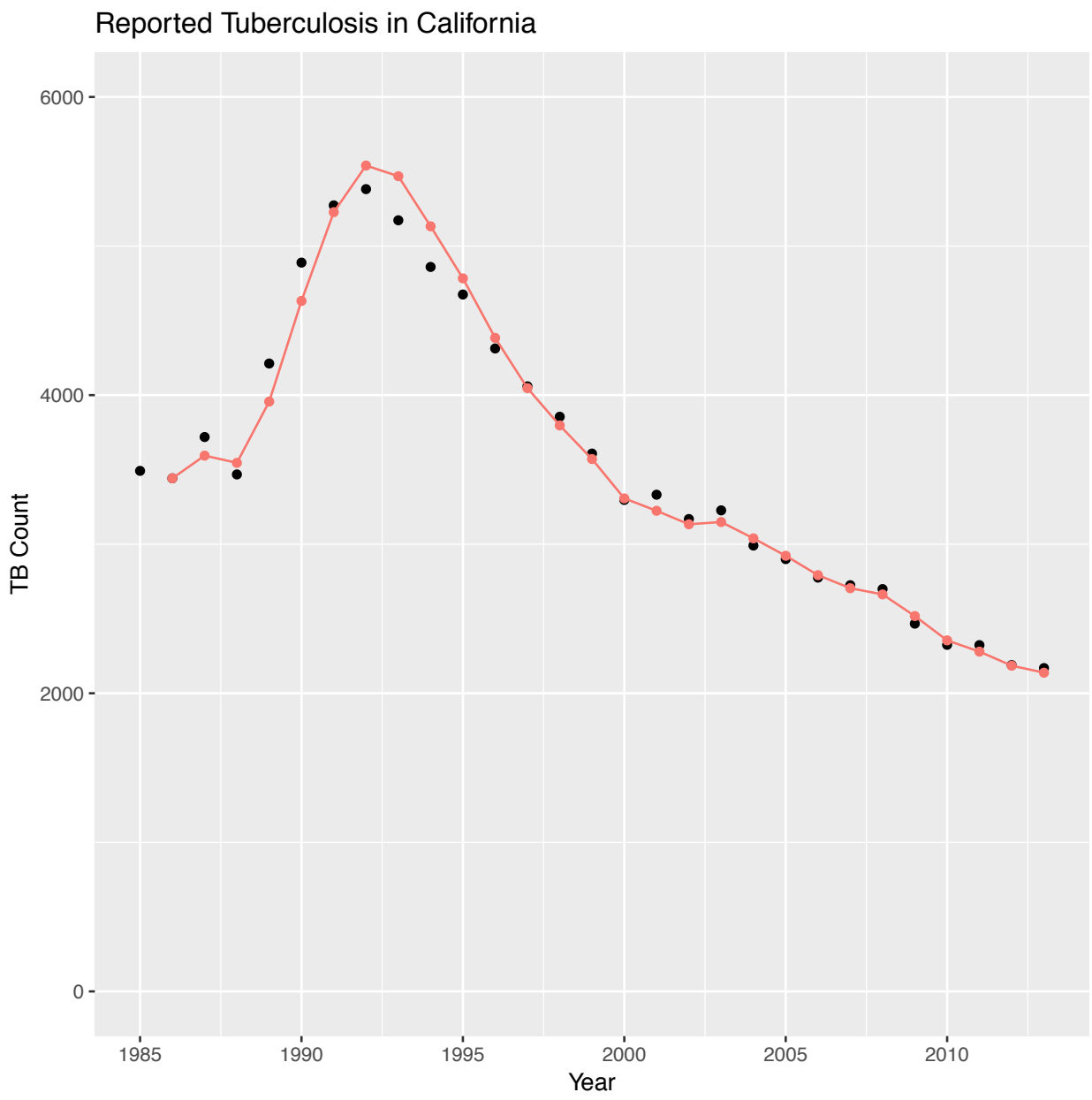


According to R, their coefficients are interpreted according to these equations (I've redacted the seasonal stuff):

```
Yhat[t+h] = a[t] + h * b[t],
where a[t], b[t] are given by
a[t] =  Y[t] + (1-) (a[t-1] + b[t-1])
b[t] =  (a[t] - a[t-1]) + (1-) b[t-1]
```

So with $a = \mu$, $b = b$, $\alpha = \lambda_0$, $\beta = \lambda_1$, $Y = x$, $t = \tau$, this is the same as above. Let's use the estimated parameters from the last part of the series, and use them on the whole series:

```
tbh$hwx <- hw.filter(tbh[,2],hwf2$alpha,hwf2$beta)
qplot(tbh[,1],tbh[,2],ylim=c(0,6000),xlab="Year",ylab="TB Count",main="Reported Tuberculosis in Califor
```



So apparently one issue was just that we were not starting the recurrences at the right place when we

looked at the short series.

What is our forecast for 2014?

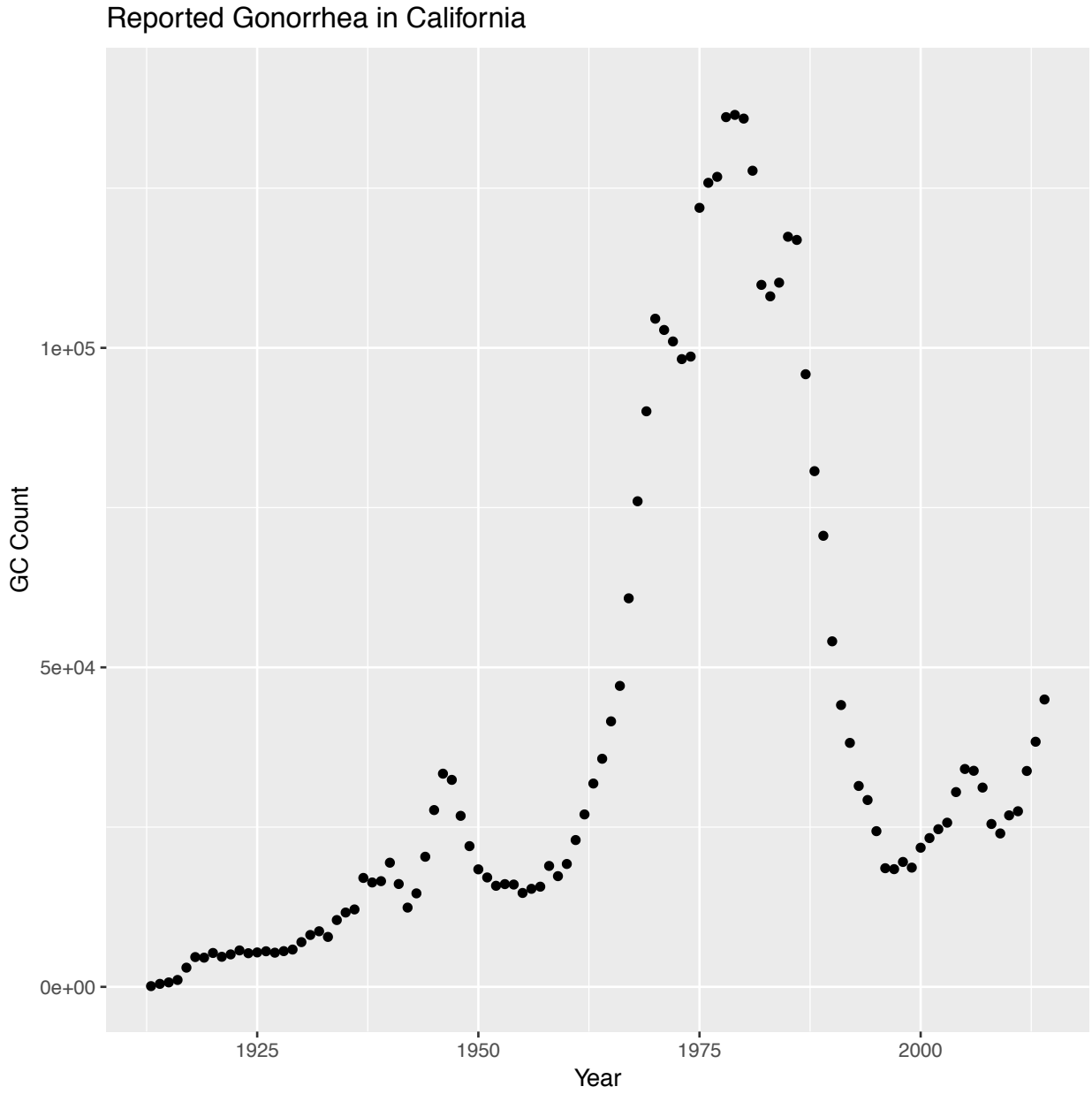
```
hw.filter(tbh[,2],hwf2$alpha,hwf2$beta,1)

## [1] 2072.733
```

The true answer, from the California DPH web site, was 2147.

Note that the Holt-Winters procedure is actually designed for seasonal data as well, and the R package delivers this as well. We won't go into it in detail.

Incidence of gonorrhea per year in California (source: CDPH).

```
cag <- structure(list(year = 1913:2014, gono = c(117L, 467L, 695L, 1083L,
3006L, 4665L, 4570L, 5305L, 4709L, 5080L, 5704L, 5265L, 5391L,
5570L, 5348L, 5593L, 5842L, 7001L, 8123L, 8702L, 7817L, 10459L,
11634L, 12118L, 17051L, 16336L, 16542L, 19433L, 16098L, 12408L,
14632L, 20365L, 27668L, 33364L, 32396L, 26767L, 22027L, 18394L,
17122L, 15821L, 16081L, 16012L, 14697L, 15346L, 15679L, 18928L,
17327L, 19236L, 22979L, 26987L, 31825L, 35700L, 41551L, 47099L,
60810L, 75998L, 90073L, 104568L, 102804L, 101006L, 98242L, 98639L,
121919L, 125833L, 126768L, 136109L, 136463L, 135885L, 127723L,
109860L, 108066L, 110208L, 117392L, 116895L, 95877L, 80708L,
70596L, 54076L, 44104L, 38182L, 31443L, 29241L, 24369L, 18570L,
18424L, 19550L, 18662L, 21778L, 23285L, 24673L, 25693L, 30484L,
34098L, 33817L, 31191L, 25493L, 24009L, 26842L, 27483L, 33788L,
38364L, 44974L)), .Names = c("year", "gono"), class = "data.frame", row.names = c(NA,
-102L))
qplot(cag$year,cag$gono,ylim=c(0,140000),xlab="Year",ylab="GC Count",main="Reported Gonorrhea in Califo
```



Exercise 3. Use the California state gonorrhea counts to produce Holt-Winters forecasts for next year's case count. Please consider log transforming the data before analysis; you may wish to omit the first five years. Why do you think the peaks and troughs occurred? Do you think the Holt-Winters procedure is appropriate for the data? There is no clear right answer; please feel free to be informal and thoughtful. ∎

More about time series

Another interesting process is derived from this recurrence:

$$X_{\tau+1} = \alpha X_\tau + \epsilon_\tau$$

where we assume for now $|alpha| < 1$ and $E[\epsilon_\tau] = 0$ for all τ. We need to assume more though about the process ϵ_τ. One simple assumption is that $E[\epsilon_\tau \epsilon_{\tau'}] = E[\epsilon_\tau]E[\epsilon_{\tau'}]$ for $\tau \neq \tau'$. Such as series is *uncorrelated*; an uncorrelated series of random variables is called *white noise*.

We will assume the series ϵ_τ is uncorrelated, and also that ϵ_τ is uncorrelated with every value of X_i for $i = 1, 2, \ldots, \tau$.

So now

$$E[X_{\tau+1}] = E[\alpha X_\tau + \epsilon_\tau]$$

Apply linearity:

$$E[X_{\tau+1}] = E[\alpha X_\tau] + E[\epsilon_\tau]$$

Again:

$$E[X_{\tau+1}] = E[\alpha X_\tau] + E[\epsilon_\tau]$$

$$E[X_{\tau+1}] = \alpha E[X_\tau] + E[\epsilon_\tau]$$

The epsilons have mean zero:

$$E[X_{\tau+1}] = \alpha E[X_\tau] + E[\epsilon_\tau]$$

$$E[X_{\tau+1}] = \alpha E[X_\tau] + 0$$

So

$$E[X_{\tau+1}] = \alpha E[X_\tau].$$

Let $\mu_\tau = E[X_\tau]$. Therefore $\mu_{\tau+1} = \alpha \mu_\tau$. We can see that

$$\mu_{\tau+\ell} = \alpha^\ell \mu_\tau$$

in general, and

$$\mu_T = \alpha^T \mu_0.$$

Optional Exercise 4. What is the limit of μ_T for large T? ∎

The following material is optional.

Let's compute the variance in the far future.

$$\mathrm{var}(X_{\tau+1}) = \mathrm{var}(\alpha X_\tau + \epsilon_\tau) = \alpha^2 \mathrm{var}(X_\tau) + \sigma^2.$$

This happens to be another linear recurrence (difference equation); the equilibrium is

$$\sigma'^2 = \frac{\sigma^2}{1-\alpha^2}.$$

Let's compute the correlation coefficient of two values, one time apart.

$$E[X_\tau X_{\tau+1}] = E[X_\tau(\alpha X_\tau + \epsilon_\tau)] = E[\alpha X_\tau^2] + E[X_\tau \epsilon_\tau]$$

We assumed the noise ϵ_τ was uncorrelated with the value, so the second term is zero. So

$$E[X_\tau X_{\tau+1}] = \alpha E[X_\tau^2].$$

Similarly, we have

$$E[X_\tau]E[X_{\tau+1}] = E[X_\tau](E[\alpha X_\tau + \epsilon_\tau]) = E[X_\tau]\alpha(E[X_\tau]) = \alpha(E[X_\tau])^2$$

The correlation coefficient of X and Y is

$$\rho = \frac{E[XY] - EXEY}{\sqrt{\mathrm{var}(X)\mathrm{var}(Y)}}.$$

So our numerator is $\alpha \mathrm{var}(X_\tau)$.

What is the variance of $X_{\tau+1}$? It must be $\mathrm{var}(\alpha X_\tau + \epsilon_\tau)$. This is going to be $\alpha^2 \mathrm{var}(X_\tau) + \sigma^2$.

So

$$\rho = \frac{\alpha \mathrm{var}(X_\tau)}{\sqrt{\mathrm{var}(X_\tau)(\alpha^2 \mathrm{var}(X_\tau) + \sigma^2)}}$$

Let's go ahead and use $\mathrm{var}(X_\tau) = \sigma^2/(1-\alpha^2)$ for large times:

$$\rho = \frac{\alpha \frac{\sigma^2}{1-\alpha^2}}{\sqrt{\frac{\sigma^2}{1-\alpha^2}\left(\alpha^2 \frac{\sigma^2}{1-\alpha^2} + \sigma^2\right)}}$$

Let's clean up this mess:

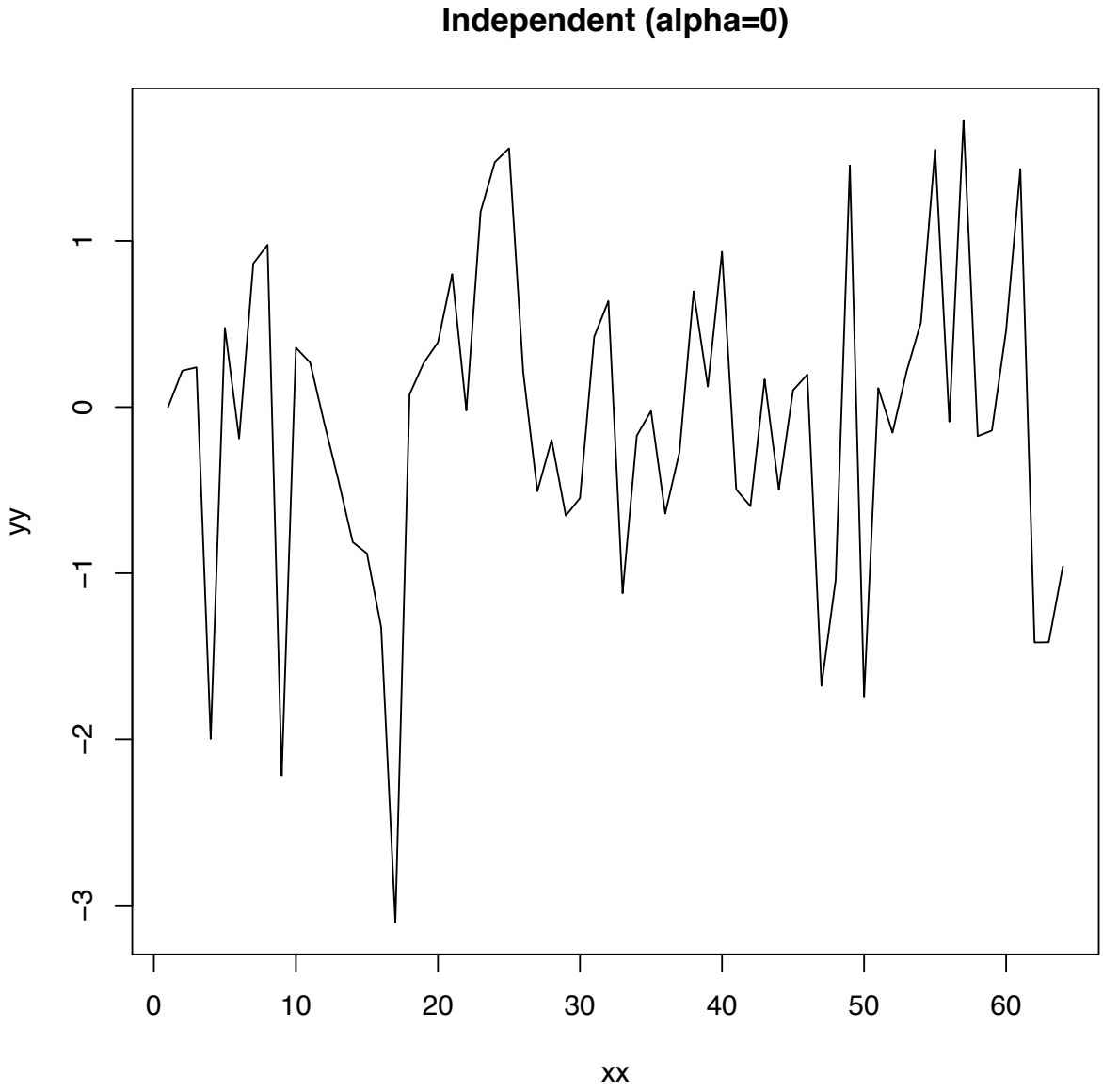
$$\rho = \frac{\alpha \frac{1}{1-\alpha^2}}{\sqrt{\frac{1}{1-\alpha^2}\left(\alpha^2 \frac{1}{1-\alpha^2} + \frac{1-\alpha^2}{1-\alpha^2}\right)}}$$

$$\rho = \frac{\alpha \frac{1}{1-\alpha^2}}{\sqrt{\frac{1}{1-\alpha^2}\frac{1}{1-\alpha^2}}} = \alpha$$

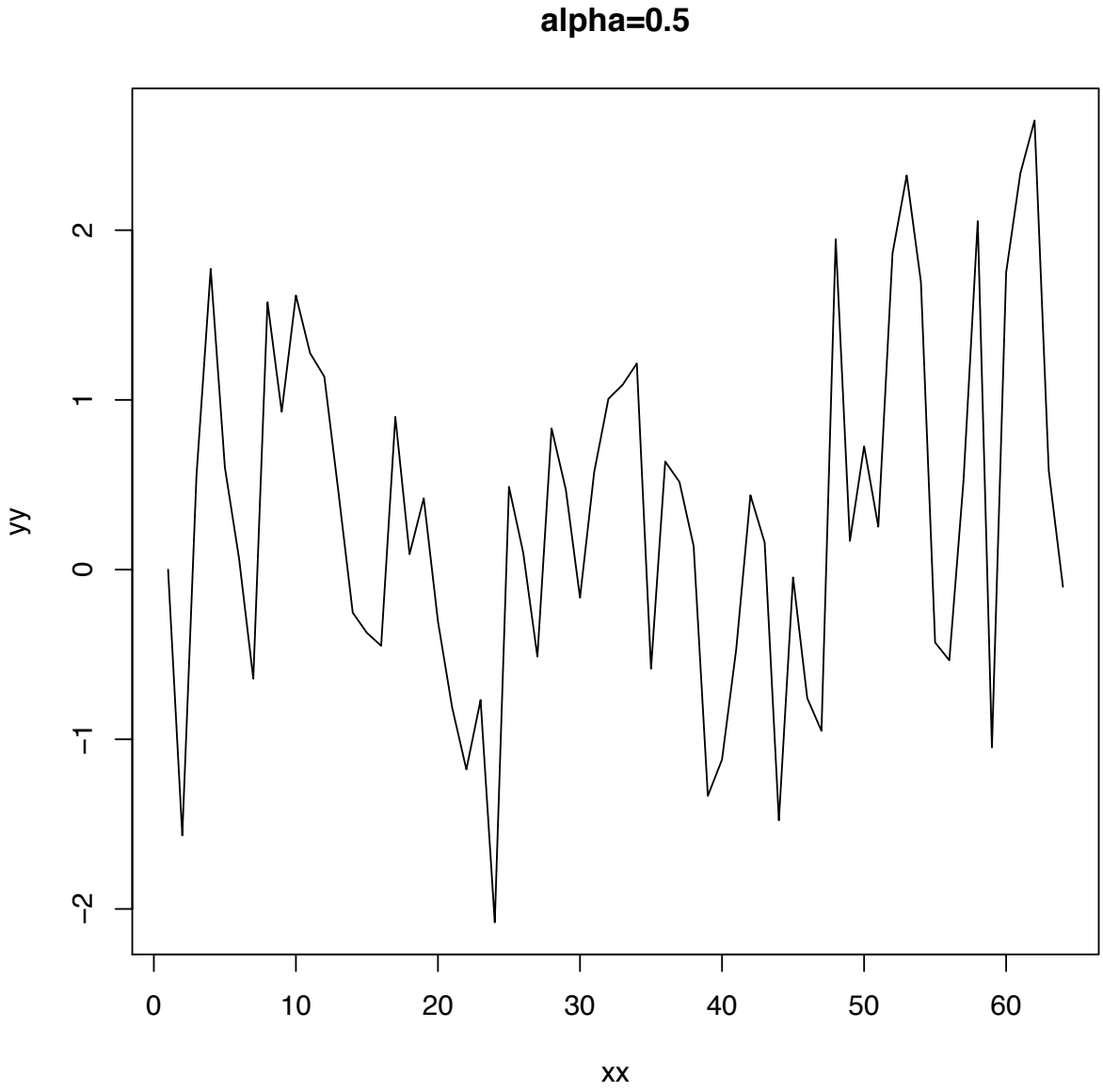
Since all we need here is $|\alpha| < 1$, this process can have negative autocorrelation as well as positive. This is called an AR(1) process and is important in time series and repeated measures analysis.

Let's look at some examples; I'm going to assume independent normal (Gaussian) white noise with variance 1. Here is R code for the simulation.

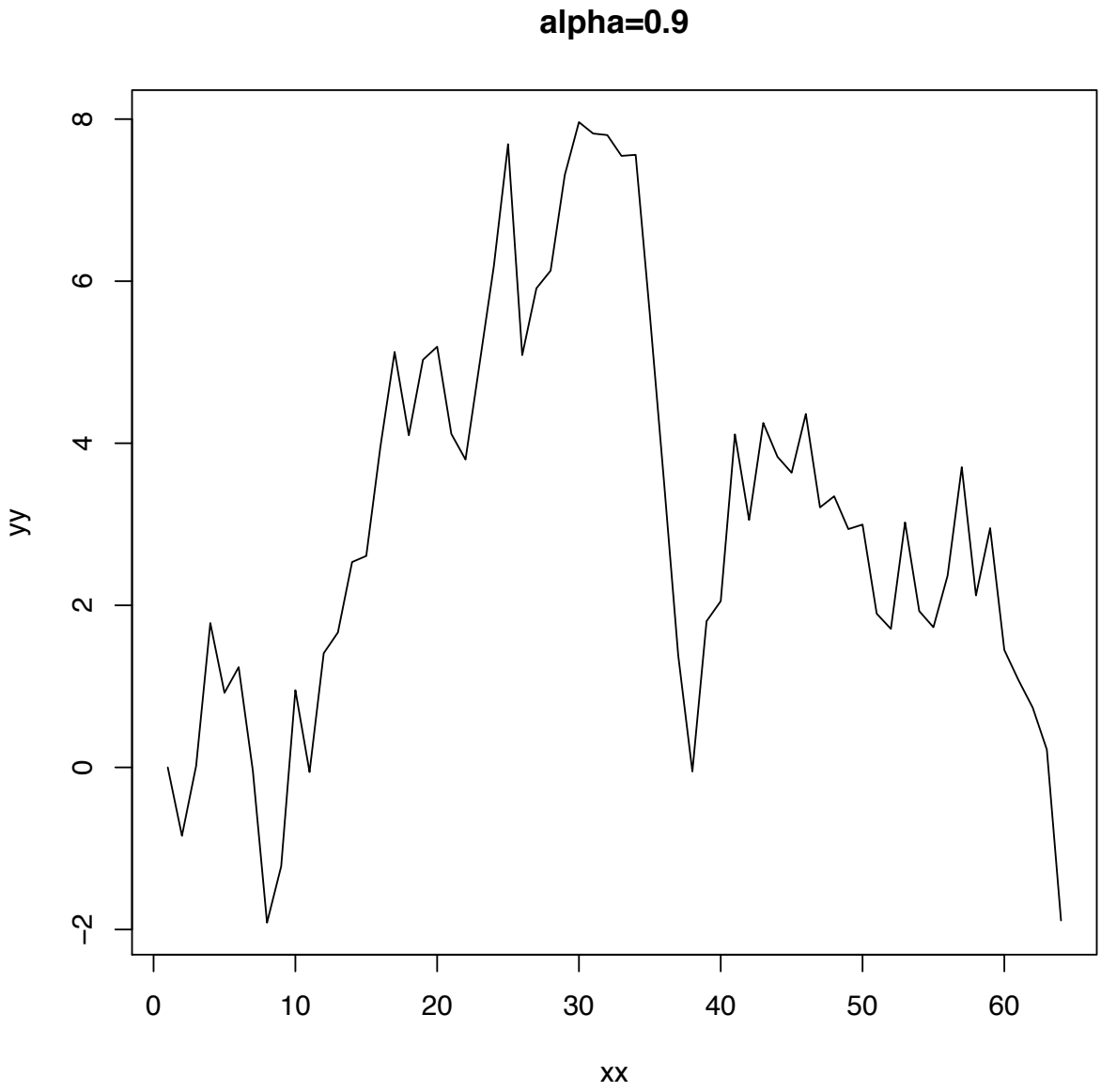
```r
ar1sim <- function(len,alpha,starting=0) {
  answer <- rep(NA,len)
  answer[1] <- starting
  cur <- starting
  for (ii in 2:len) {
    cur <- alpha*cur + rnorm(1,mean=0,sd=1)
    answer[ii] <- cur
  }
  answer
}
xx <- 1:64
yy <- ar1sim(length(xx),0)
plot(xx,yy,main="Independent (alpha=0)",type="l")
```




```
yy <- ar1sim(length(xx),0.5)
plot(xx,yy,main="alpha=0.5",type="l")
```




```
yy <- ar1sim(length(xx),0.9)
plot(xx,yy,main="alpha=0.9",type="l")
```



Optional Exercise 5. Using R or some other language of your choice, plot simulated AR(1) time series for the normal model with unit variance. Choose $\alpha = 0.99, 0.8, 0.5, 0.2, 0.1, -0.1, -0.2, -0.5, -0.8, -0.99$. Run series longer than 64; run replications over and over again. (You only need to turn in one replication.) Please comment on the differences and similarities. ∎

What I want you to understand from this section is the idea of a stochastic recurrence, and in particular the AR(1) process. You should know that for $|\alpha| < 1$, this parameter α is the correlation from one time to

the next.

Review and preparation for the Galton-Watson process

Suppose we parameterize the geometric distribution in such a way that we ask how many failures (not how many trials: how many failures) before we have our first success. Then we get $P(Y = i) = p(1-p)^i$ for $i = 0, 1, 2, \ldots$. This is still called the geometric; it's just defined from 0 to infinity. You see both formulations in the literature; R, for example, uses this one. The generating function in this case is

$$E[u^Y] = \sum_{y=0}^{\infty} u^y P(Y = y) = \sum_{y=0}^{\infty} u^y p(1-p)^y = p\frac{1}{1-u(1-p)}.$$

Note something interesting when comparing this distribution to the version from before—the probability generating function of the version shifted 1 to the right is just this one multiplied by u.

Let's ask the question how many failures before we get 2 successes. That's a geometric waiting time for the first success, and another for the second. It's the sum of two independent geometrics (with the same p). Let the first one be Y_1, the second Y_2:

$$E[u^{Y_1+Y_2}] = E[u_{Y_1}]E[u_{Y_2}] = p\frac{1}{1-u(1-p)}p\frac{1}{1-u(1-p)} = \frac{p^2}{(1-u(1-p))^2}$$

If we are waiting for k successes, we have to add up k independent identical geometrics, and we get

$$E[u^{Y_1+\cdots+Y_k}] = \frac{p^k}{(1-u(1-p))^k}$$

a distribution known as the *negative binomial* distribution.

Even though we defined it as a waiting time until k successes, in fact there's nothing stopping us from using the generating function we just derived, but with k any positive number, integer or not. This distribution is widely used in modeling and applied statistics for all sorts of things having nothing to do with waiting times, though it can be used for those of course.

Let's get the mean of a negative binomial Y with parameters p and k:

$$E[Y] = g'(1) = \frac{d}{du}\frac{p^k}{(1-u(1-p))^k} = p^k(-k)(1-u(1-p))^{-k-1}(-(1-p))|_{u=1}.$$

This gives us

$$E[Y] = \frac{p^k k(1-p)}{p^{k+1}} = \frac{k(1-p)}{p}.$$

Of course—this was from adding up independent geometrics (geometrics starting from 0).

Exercise 6. Find the variance of the negative binomial. ∎

It turns out that if you take $k \to \infty$ holding the mean constant, you get a Poisson. Just as the binomial has a Poisson limit, so does the negative binomial.

Reference:

Harvey, AC. Forecasting, structural time series, and the Kalman filter.