Lecture 7: Exponential Smoothing With Trend and Seasonality

And a Brief Tour of Other Popular Forecasting Methods

Introduction to Time Series, Fall 2023

Ryan Tibshirani

Related reading: Chapters 8, 9.10, 12.2, and 12.4 of Hyndman and Athanasopoulos (HA).

1 Simple exponential smoothing

- Exponential smoothing is arguably the other—outside of ARIMA—most popular basic framework for forecasting in time series. These two frameworks bear a neat connection, which you saw at the end of the last lecture on ARIMA, and which we'll revisit a bit later in this lecture
- We'll begin with the simplest possible exponential smoother, called (unsurprisingly?) *simple exponential smoothing* (SES). This constructs a 1-step ahead forecast via

$$\hat{x}_{t+1|t} = \alpha x_t + (1-\alpha)\hat{x}_{t|t-1} \tag{1}$$

  where $\alpha \in [0,1]$ is a parameter to be estimated
- In other words, the SES forecast (1) is a weighted combination of the current observation x_t and the previous forecast $\hat{x}_{t|t-1}$
- By unraveling the iteration, which is basically the same calculation that we did in the ARIMA lecture (but now in the opposite direction), this can also be written as

$$\hat{x}_{t+1|t} = \alpha x_t + \alpha(1-\alpha)x_{t-1} + \alpha(1-\alpha)^2 x_{t-2} + \ldots \tag{2}$$

  This explains its name, since observations x_{t-k} that are k steps into the past are exponentially-downweighted, with weight $(1-\alpha)^k$
- (Note: we are being intentionally vague here about the boundary condition. In ARIMA, to develop the theory cleanly, we let time extend back to $-\infty$. In exponential smoothing, we instead usually index time starting at $t = 0$, in which case the right-hand side in (2) would end with $\alpha(1-\alpha)^t x_0$)
- To make h-step ahead forecasts, we iterate (1), where (as usual) we replace future observations by their forecasts. This simply yields $\hat{x}_{t+2|t} = \alpha\hat{x}_{t+1|t} + (1-\alpha)\hat{x}_{t+1|t} = \hat{x}_{t+1|t}$, and in general,

$$\hat{x}_{t+h|t} = \alpha x_t + (1-\alpha)\hat{x}_{t|t-1} \tag{3}$$

  for all horizons $h \geq 1$. That is, SES generates *flat* forecast trajectories. We'll see how to extend this to accomodate a trend, shortly
- While SES smoothing is already very intuitive, we can motivate it in different way, as follows. The *naive flatline forecaster* produces forecasts via

$$\hat{x}_{t+h|t} = x_t \tag{4}$$

  i.e., it just propogates the last observation forward. Meanwhile, the *naive average forecaster* produces forecasts via

$$\hat{x}_{t+h|t} = \frac{1}{t}\sum_{i=1}^{t} x_i \tag{5}$$

  Often we want something in between these two extremes, and that something is given to us by exponential smoothing, recalling the form in (2)

1.1 Component form

- For the developments that follow, it is helpful to rewrite the SES forecast (3) in what is known as *component form*
- Specifically, we think of a "hidden" level ℓ_t that we are tracking over time, that base our forecasts on:

$$\begin{aligned} \hat{x}_{t+h|t} &= \ell_t \\ \ell_t &= \alpha x_t + (1-\alpha)\ell_{t-1} \end{aligned} \tag{6}$$

- The representation in (6) may appear as kind of a trivial rewriting of (3), where we replace $\hat{x}_{t+1|t}$ by ℓ_t, and $\hat{x}_{t|t-1}$ by ℓ_{t-1}. Nonetheless, it serve as a useful jumping off point to extend the model in the next section
- Before moving on, we give a brief example of SES from HA, to forecast internet useage per minute. The data and SES forecast are shown in Figure 1, top row. In order to carry out the forecast, we have to estimate the smoothing parameter α in (6). This is typically done by maximum likelihood (but where does the probabilistic model come from? more later ...), and is what is implemented as the default in the `ETS()` function in the `fable` package
- The forecast from SES is not very impressive, and honestly, in general, SES should probably only be viewed as a small step up from the naive forecasters (4), (5)
- The forecast trajectory from SES is flat, by construction (as previously noted). Next we'll see how to extend the method to accommodate a linear trend

2 Trend extensions

- An extension of the SES forecaster in (6) is *Holt's linear trend* method. This changes both the forecast equation (first line) and the level equation (second line) to accomodate an estimate of the slope b_t of the series at time t. We add a trend equation to evolve the slope component
- Precisely, Holt's linear trend method in component form (which we will stick to henceforth) is:

$$\begin{aligned} \hat{x}_{t+h|t} &= \ell_t + b_t h \\ \ell_t &= \alpha x_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta) b_{t-1} \end{aligned} \tag{7}$$

  Now β is an additional parameter to be estimated, where both $\alpha, \beta \in [0,1]$
- As before, the level equation updates ℓ_t as an α-weighted combination of the current observation x_t and the previous 1-step ahead forecast $\hat{x}_{t|t-1} = \ell_{t-1} + b_{t-1}$
- The trend equation updates b_t as a β-weighted combination of the current trend $\ell_t - \ell_{t-1}$ and the previous trend b_{t-1}
- Critically, the forecast trajectory from Holt's linear trend method is no longer flat but (as the name suggests, and as is apparent from (7)) a linear function, with slope b_t
- The middle row of Figure 1 shows the forecast from Holt's linear trend method on the internet useage today. To be clear, now both α, β have been estimated from the data. We can see that it predicts a downward trajectory, since b_t at the last time t appears to be negative. However, its prediction intervals are very wide, suggesting that the model is highly uncertain of trend directionality

2.1 Damped trends

- Linear trend forecasts at long horizons can be somewhat erratic; we've already seen that the forecast variance is quite high in the example in Figure 1 (as evidenced by the wide prediction intervals) and this wasn't even a super long horizon ...

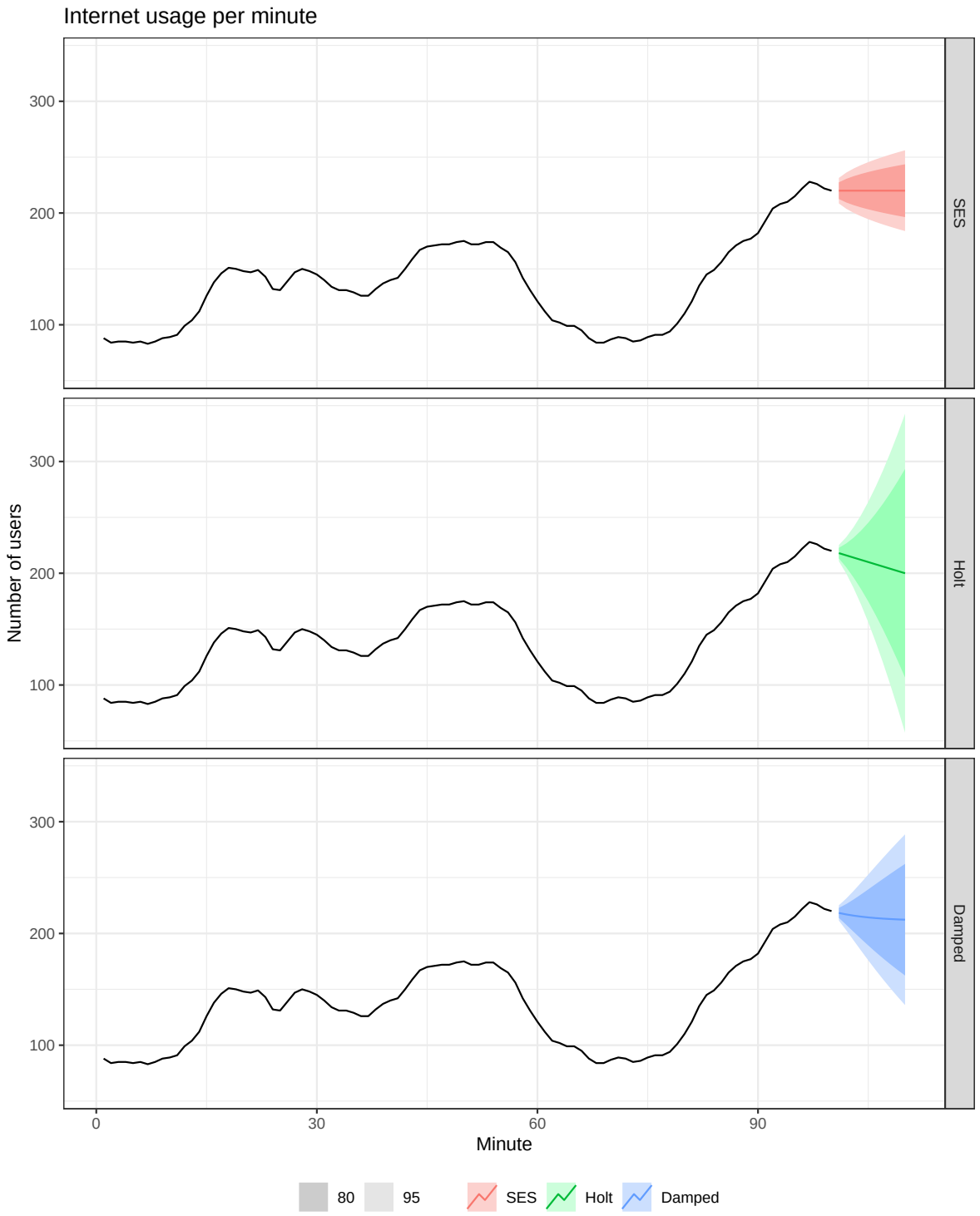

Figure 1: Forecasts on internet useage data (from HA) at 10-steps ahead, from three different exponential smoothing models: simple exponential smoothing (top), Holt's linear trend (middle), and damped linear trend (bottom).

- As a kind of regularization, we can *dampen* the forecasts from Holt's linear trend method. This is often called *damped Holt's* or the *damped linear trend* method
- In addition to $\alpha, \beta \in [0,1]$ as in (7), we introduce a third parameter $\phi \in [0,1]$, to dampen the forecast trajectory:

$$\begin{aligned} \hat{x}_{t+h|t} &= \ell_t + b_t(\phi + \phi^2 + \cdots + \phi^h) \\ \ell_t &= \alpha x_t + (1-\alpha)(\ell_{t-1} + b_{t-1}\phi) \\ b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1}\phi \end{aligned} \tag{8}$$

  Note that when $\phi = 1$, this reduces to Holt's linear trend method in (7)
- The interpretation in the damped method (8) is mostly the same as in Holt's method (7), but you can think of the modification like this: the contribution of a given slope to a forecast in the future diminishes at each step into the future, by a multiplicative factor of ϕ
- As $h \to \infty$, the forecasts from the damped linear trend method approach a particular constant (finite) level, namely

$$\hat{x}_{t+h|t} \to \ell_t + b_t \sum_{j=1}^{\infty} \phi^j = \ell_t + b_t \frac{\phi}{1-\phi}$$

  For example, when $\phi = 0.9$, this limit is $\ell_t + 9b_t$
- HA say that a practical range for ϕ is usually 0.8 to 0.98; when ϕ is below 0.8, the dampening is too strong (and short-term forecasts are not "trended" enough); when ϕ is above 0.98, it is too weak (and you cannot distinguish the forecasts from Holt's linear trend method for reasonably horizons). In fact, the `ETS()` function limits the range of ϕ to be $[0.8, 0.98]$, by default
- The bottom row of Figure 1 shows the forecasts from the damped linear trend method on the internet useage data. To be clear, all of α, β, ϕ are estimated from the data. We can see a weak downward trend, that is quickly attenuated. The prediction intervals are also much narrower. Inspection of the fitted model (see the R notebook) shows that we the dampening coeffcient estimate is $\hat{\phi} = 0.81$, so it has quite a pronounced effect here

3 Seasonality extensions

- To account for seasonality, on top of trend, we can use what is called the *Holt-Winters* method. This changes the forecast and level equations in (7) in order to adjust for a seasonal effect, but the trend equation stays the same. We add a seasonality equation to evolve the seasonal component
- We assume a known seasonal period m. That is, observations occurring every m time points share a common (but unknown) seasonal effect. The Holt-Winters method is then:

$$\begin{aligned} \hat{x}_{t+h|t} &= \ell_t + b_t h + s_{t+h-mk} \\ \ell_t &= \alpha(x_t - s_t) + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \\ s_t &= \gamma(x_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m} \end{aligned} \tag{9}$$

  The parameters of the model are $\alpha, \beta \in [0,1]$ and $\gamma \in [0, 1-\alpha]$
- In the forecast equation (first line), we define $k \geq 0$ to be the integer such that $t + h - mk \in [t-m, t]$: the seasonal component we are using to adjust the forecast should be the latest one (whatever was available in the last period). This is equivalent to seeking $mk \in [h, h+m]$. You can check that this is accomplished by setting $k = \lceil h/m \rceil$
- The interpretation of Holt-Winters (9) is similar to Holt's method (7), except that we *seasonally-adjust* the observations in the level and trend equations. The seasonal equation updates s_t as a γ-weighted combination of $x_t - \ell_{t-1} - b_{t-1}$ and the last relevant seasonal component s_{t-m}

- We can view $x_t - \ell_{t-1} - b_{t-1}$ as the result of solving for s^* in the equation:

$$x_t = \ell_{t-1} + b_{t-1} + s^*$$

  This is like the 1-step ahead forecast equation a time $t-1$, but where we replace the forecast $\hat{x}_{t|t-1}$ by the observation x_t

- Note: we can also introduce dampening into (9), which is just as in (8), but do not write this out for brevity

- Figure 2 shows an example of Holt-Winters in action, on Australian holiday travel data from HA. Its ability to pick up (and evolve!) trend and seasonality appears impressive

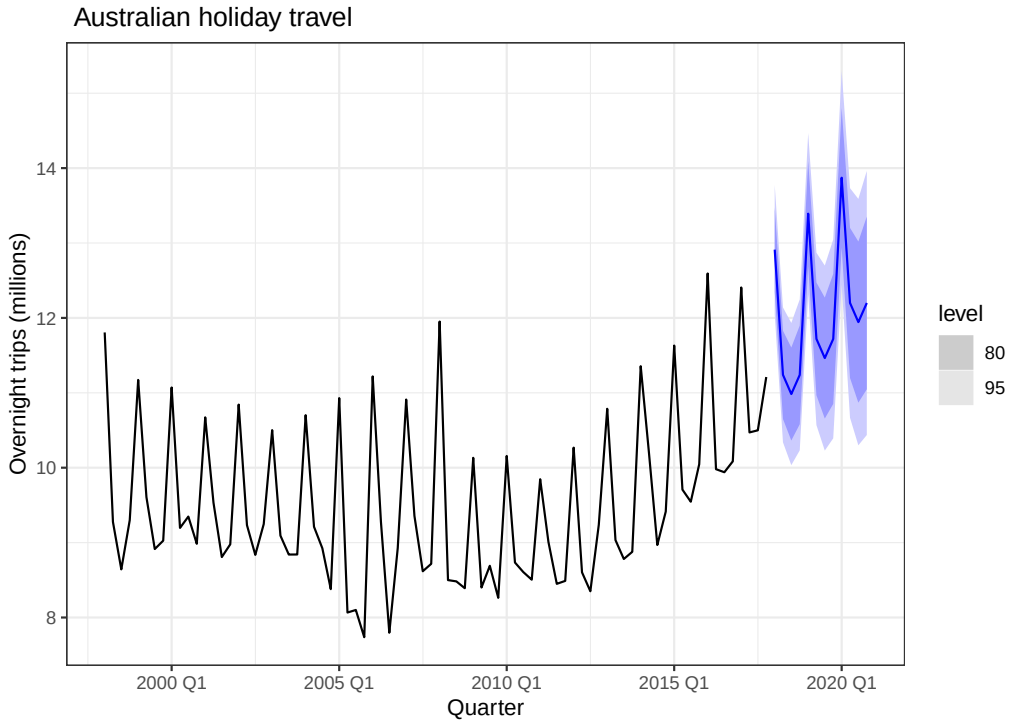

Figure 2: Forecasts on Australian holiday travel data (from HA) at 12-steps ahead, from the Holt-Winters method.

3.1 Multiplicative seasonality

- The seasonal effect in the Holt-Winters method (9) is *additive*. If we look at the forecast equation, in the first line, then we see that the seasonal component is being added to the level and trend components in order to produce the forecast

- A version of Holt-Winters with *multiplicative seasonality* is also possible:

$$\begin{aligned} \hat{x}_{t+h|t} &= (\ell_t + b_t h) s_{t+h-mk} \\ \ell_t &= \alpha \frac{x_t}{s_t} + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\ b_t &= \beta(\ell_t - \ell_{t-1}) + (1-\beta) b_{t-1} \\ s_t &= \gamma \frac{x_t}{\ell_{t-1} - b_{t-1}} + (1-\gamma) s_{t-m} \end{aligned} \tag{10}$$

  The parameters are again $\alpha, \beta \in [0,1]$ and $\gamma \in [0, 1-\alpha]$

- The equations in (10) are motivated and interpreted just as in (9), except that the contribution of the seasonal component is multiplicative. We can see this in the forecast equation, in the first line: we take the non-seasonal forecast (level and trend) and multiply it by the seasonal component
- We can view $x_t/(\ell_{t-1} + b_{t-1})$, in the seasonal equation, as the result of solving for s^* in the equation:

$$x_t = (\ell_{t-1} + b_{t-1})s^*$$

  This is again like the 1-step ahead forecast equation a time $t-1$, but where we replace the forecast $\hat{x}_{t|t-1}$ by the observation x_t
- In practice, the additive and multiplicative seasonal models can sometimes result in fairly similar component estimates—this happens with the holiday travel data, for example (the next subsection gives evidence of this). But in other problems they can result in genuine differences, so it is worth thinking about whether the seasonal effect in the problem at hand could *plausibly* be multiplicative, and if so, worth trying out and evaluating this formulation as well

3.2 Time series decomposition

- We've talked about decompositions of time series at various points in the past, and touched on multiple approaches for fitting them
- The Holt-Winters method, either in additive or multiplicative form, is not just a forecaster but also provides us with a decomposition of the given time series by extracting the fitted level ℓ_t, trend b_t, and seasonal s_t sequences
- There is a bit of an unfortunate clash of nomenclature here: previously we talked about seasonal-trend decompositions. And the fitted level component from Holt-Winters actually provides what we called the trend component previously! Meanwhile, the fitted trend component from Holt-Winters does not have a correspondence to anything we talked about previously. It reflects "where the time series is heading"
- Figure 3 shows an example on the holiday travel data. Both the additive and multiplicative methods return fairly similar component estimates. The seasonal pattern also appears to be unchanging over time, which is a consequence of the fact that the estimate of γ in both models is tiny (around 0.0001 in both models, as can be seen in the R notebook)

4 ETS models

- *Exponential smoothing with trend and seasonality* (ETS) models are a class that includes everything we've seen thus far: simple exponential smoothing, Holt's linear trend method, Holt-Winters method with additive or multiplicative seasonality, and all of their dampened trend versions
- In fact, ETS includes more: we can also change the model to accomodate a *multiplicative error* component (rather than an additive error component, as was previously introduced). This will be made more precise when we talk about the state space representation, below
- (While mathematically possible, HA do not recommend allowing for a multiplicative trend, since they say that it can often behave poorly in practice)
- Thus we an ETS model is written ETS(x, y, z), where
  - $x \in \{\text{A}, \text{M}\}$
  - $y \in \{\text{N}, \text{A}, \text{Ad}\}$
  - $z \in \{\text{N}, \text{A}, \text{M}\}$

  here N stands for "none" (no trend component or no seasonality component), A stands for "additive", Ad stands for "additive-dampened", and M stands for multiplicative

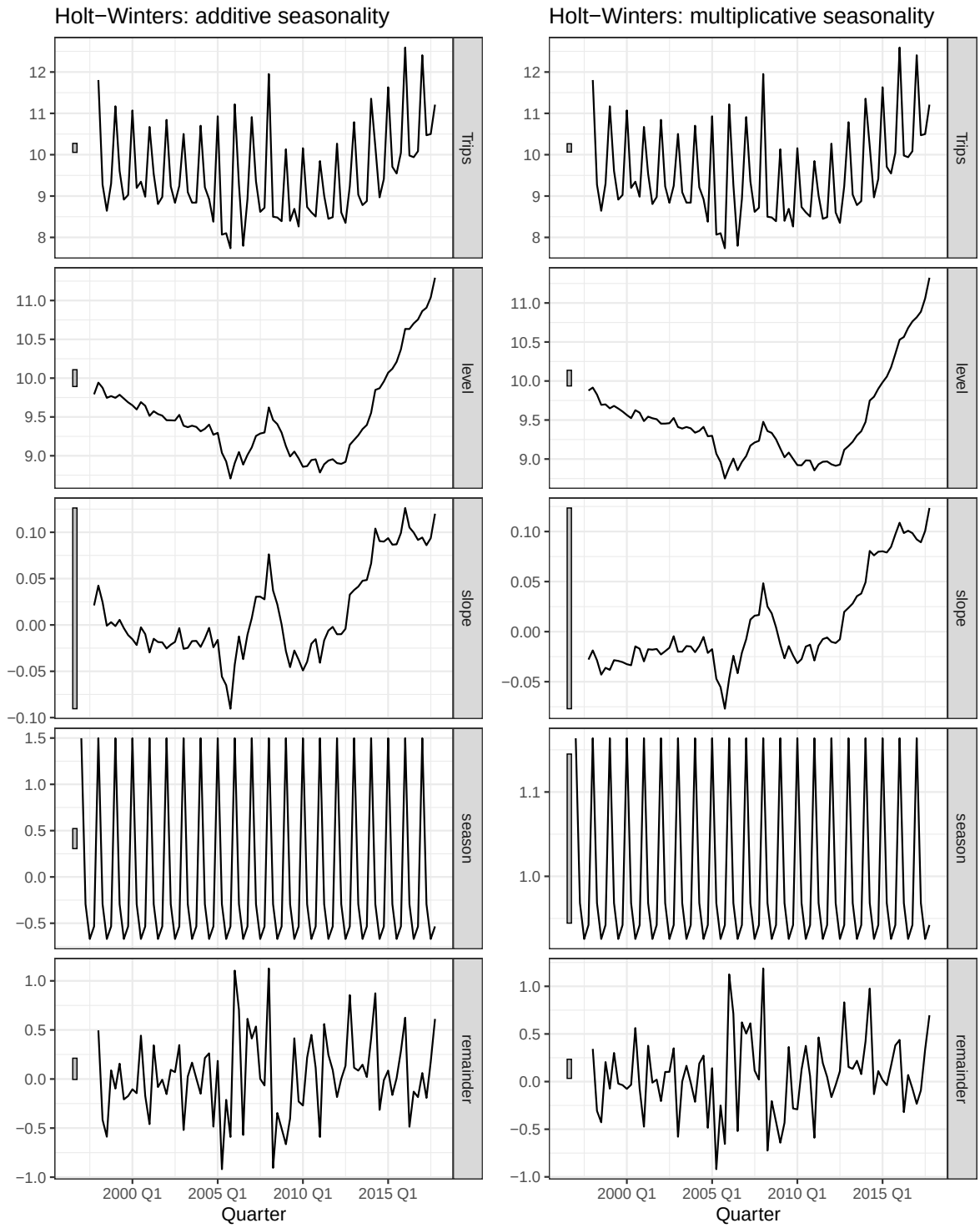

Figure 3: Decomposition of the Australian holiday travel data (from HA) from the Holt-Winters method in additive and multiplication forms.

- So to be clear:
  - ETS(A,N,N) is simple exponential smoothing
  - ETS(A,A,N) is Holt's linear trend method
  - ETS(A,Ad,N) is the dampened linear trend method
  - ETS(A,A,A) is Holt-Winters with additive seasonality
  - ETS(A,A,M) is Holt-Winters with multiplicative seasonality
- Note: HA do not recommend combining additive errors with multiplicative seasonality, saying that this can lead to numerical instability when estimating parameters. Thus they say that ETS(A,*,M) should generally be ditched in favor of ETS(M,*,M)

4.1 State space representation

- A useful way to represent (and think about) ETS models is what is called a *state space* formulation. This gives equivalent point forecasts $\hat{x}_{t+h|t}$, but moreover provides a probabilistic framework for ETS models, which informs us how to carry out maximum likelihood and compute prediction intervals
- We'll start by writing ETS(A,N,N) or SES (6) in state space form. This is:

$$\begin{aligned} x_t &= \ell_{t-1} + \epsilon_t \\ \ell_t &= \ell_{t-1} + \alpha\epsilon_t \end{aligned} \tag{11}$$

  The first equation is typically called the observation (or measurement) equation, and the second is called the state equation
- In (11), and all state space formulations henceforth, each error ϵ_t is i.i.d. taken to be $N(0, \sigma^2)$. Thus the joint distribution of the observations $x_1, \ldots, x_t$ is fully specified by (11) and we can do maximum likelihood in order to estimate the unknown parameters: here, α and ℓ_0
- To see (11) is equivalent to (6), we have to understand how point forecasts are generated. This will explained in more detail later, but it is really the exact same story as in ARIMA: we replace past errors by their residuals, and future errors by zero. Thus in (11), the state equation at time t becomes

$$\ell_t = \ell_{t-1} + \alpha(x_t - \ell_{t-1})$$

  and the observation equation at time $t+1$ gives us the forecast:

$$\hat{x}_{t+1|t} = \ell_t + 0$$

  Clearly, the last two equations together recreate SES as defined in (6)
- As another example, ETS(A,A,N) or Holt's linear trend (7) in state space form is:

$$\begin{aligned} x_t &= \ell_{t-1} + b_{t-1} + \epsilon_t \\ \ell_t &= \ell_{t-1} + b_{t-1} + \alpha\epsilon_t \\ b_t &= b_{t-1} + \beta\epsilon_t \end{aligned} \tag{12}$$

  For convenience, we have redefined the product $\alpha\beta$ (with β as originally defined in (7)) as simply β in (12), which is commonly done in the literature
- As another example, ETS(M,A,N) in state space form is:

$$\begin{aligned} x_t &= (\ell_{t-1} + b_{t-1})(1 + \epsilon_t) \\ \ell_t &= (\ell_{t-1} + b_{t-1})(1 + \alpha\epsilon_t) \\ b_t &= b_{t-1} + \beta(\ell_{t-1} + b_{t-1})\epsilon_t \end{aligned} \tag{13}$$

  Note that the contribution of errors in (13) are multiplicative
- To read the details of the other state space representations, see Chapter 8.5 of HA

4.2 Estimation and selection

- On estimation and selection in ETS, we will say even less than we did with ARIMA, but the story is largely the same ...
- To estimate model parameters ($\alpha, \beta, \phi, \gamma, \ell_0, b_0, s_0, s_{-1}, \ldots$, or some subset thereof), we use the state space representation, form the likelihood—assuming normal errors in the state space representation, and then maximize the likelihood. We could also estimate parameters by minimizing least squares, but that is not equivalent for ETS models in general
- In R, the `ETS()` function in the `fable` package allows us to specify and fit any model in the ETS family
- Selecting a particular ETS model (additive or multiplicative errors? presence of trend? dampened trend? seasonality? etc.?) is pretty difficult, in one sense, just like order selection in ARIMA is difficult. Here, we refer to difficulty in the sense of model identification: supposing there were one true data generating model among the ETS state space family, it would be hard to identify it reliably
- There are automated algorithms to select an ETS model, and these are implemented in `ETS()`, but you have to be careful using them. They can sometimes identify some aspects well (like the period, if seasonality is obvious), but can also introduce a lot of variance
- We will adopt our same perspective with ARIMA, and forecasting models in general: *an ETS model is useful is it predicts well*, and will appeal to time series CV to help us decide what to use

4.3 Forecasting

- Point forecasts with ETS are defined directly with their original formulations: (6), (7), (8), (9), etc.
- The state space representation provides an equivalent view, and also allows us to produce prediction intervals. The general approach to forecasting is analogous to that in ARIMA
- Obtaining the forecast $\hat{x}_{t+h|t}$ from an ETS model can be done by iterating the following steps:
  1. Start with the ETS state space representation and rewrite the equation by replacing t with $t+h$
  2. Replace future observations (x_{t+k}, $k \geq 1$) with their forecasts, future errors (w_{t+k}, $k \geq 1$) with zero, and past errors (w_{t-k}, $k \geq 0$) with their ETS residuals
- Obtaining prediction intervals can be done by first computing an estimate $\hat{\sigma}_h^2$ of the variance of the "h-step ahead forecast distribution" from ETS (in quotes because we have not precisely defined this, but see Section 8.7 in HA for details and for precise formulae). Then we could use

  $$N\big(\hat{x}_{t+h|t}, \hat{\sigma}_h^2\big)$$

  as our model for the h-step ahead forecast distribution at time t, and compute prediction intervals accordingly
- We could also simulate future forecast paths via the bootstrap, as implemented in the `fable` package's `forecast()` function when `bootstrap = TRUE`. The idea is just as before (as in ARIMA): in step 2, instead of replacing future errors (w_{t+k}, $k \geq 0$) by zero, we replace them by a bootstrap draw (i.e., a uniform sample with replacement) from the empirical distribution of past residuals. Then, post simulation, we read off sample quantiles at $t+h$ for a prediction interval
- Neither of the above methods (nor any traditional methods) actually guarantee coverage in practice. We will need to run calibration methods on top if we want long-run coverage guarantees in general, which we'll cover in the next (and final) lecture

4.4 ARIMA versus ETS

- ARIMA and ETS models bear interesting connections, and it is worth discussing their similarities and differences

- Recall that there are 18 ETS models in total: 2 choices for the errors (A, M) $\times$ 3 choices for trend (N, A, Ad) $\times$ 3 choices for seasonality (N, A, M)
- Interesting fact: *the 6 fully additive ETS models: ETS(A,N,N), ETS(A,A,N), ETS(A,N,A), ETS(A,A,A), ETS(A,Ad,N), and ETS(A,Ad,A), are each special cases of ARIMA models*
- We showed at the end of the last lecture that ARIMA(0,1,1) was actually the same as SES. Similar proofs are possible for the other fully additive ETS models. An important note: in each case here, the I component in the equivalent ARIMA model is nontrivial ($d \neq 0$), which means that all ETS models are nonstationary
- Meanwhile, each of the other 12 ETS models with a multiplicative component is not a special case of ARIMA
- And lastly, some ARIMA models are stationary: precisely, ARMA models ($d = 0$), whereas all ETS models are nonstationary, as just noted above
- This is all nicely summarized by Figure 4 which is taken from Chapter 9.10 of HA

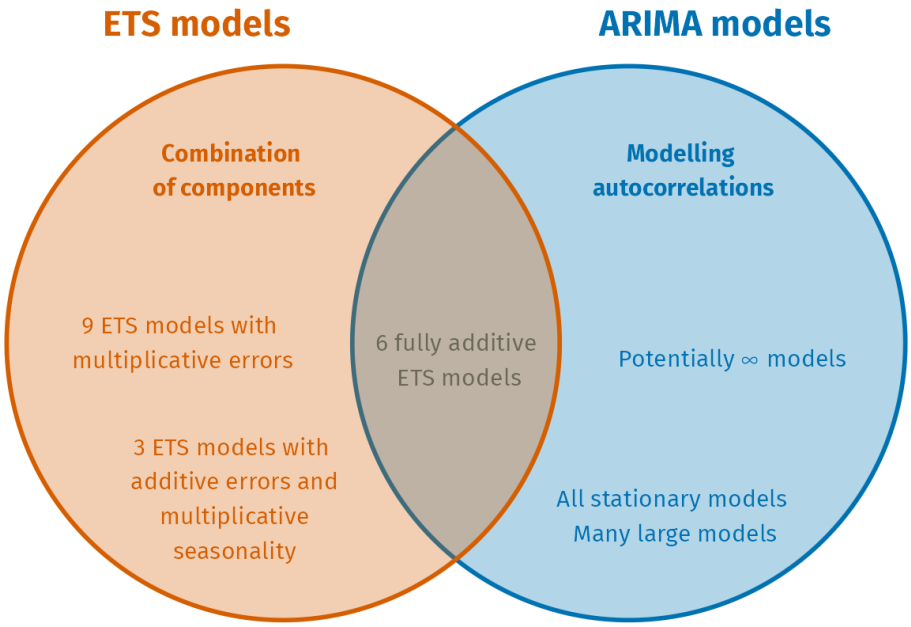

Figure 4: Summary of similarities and differences between ARIMA and ETS (from HA).

- Conceptually, ARIMA and ETS are born out of different lines of motivation: ARIMA stems from modeling autocorrelations, in either the process itself (AR) or the errors (MA), whereas ETS stems from a modeling components of the time series (level, trend, seasonal)
- Both classes can lead to useful forecasters, and we will typically fit one or more models from each class and use time series CV to compare

5 Theta model

6 Prophet model

7 Neural network AR