Regression Methods

ECON 5753 — University of Arkansas

Prof. Kyle Butts

April 2025

Time-series

Time-series data is a set of observations y_t that occur for a single unit measured over the course of time

→ In general, we call t the 'period'

→ If the time-series is spaced evenly over time without missing, it is called regular.

- Some methods require regular time intervals, and will do weird things if you give it irregular time series

→ If you observe many units' time-series, this is called panel data

- Panel data that is regular is called a balanced panel

Examples

Examples include:

→ Annual data on the DGP of a country

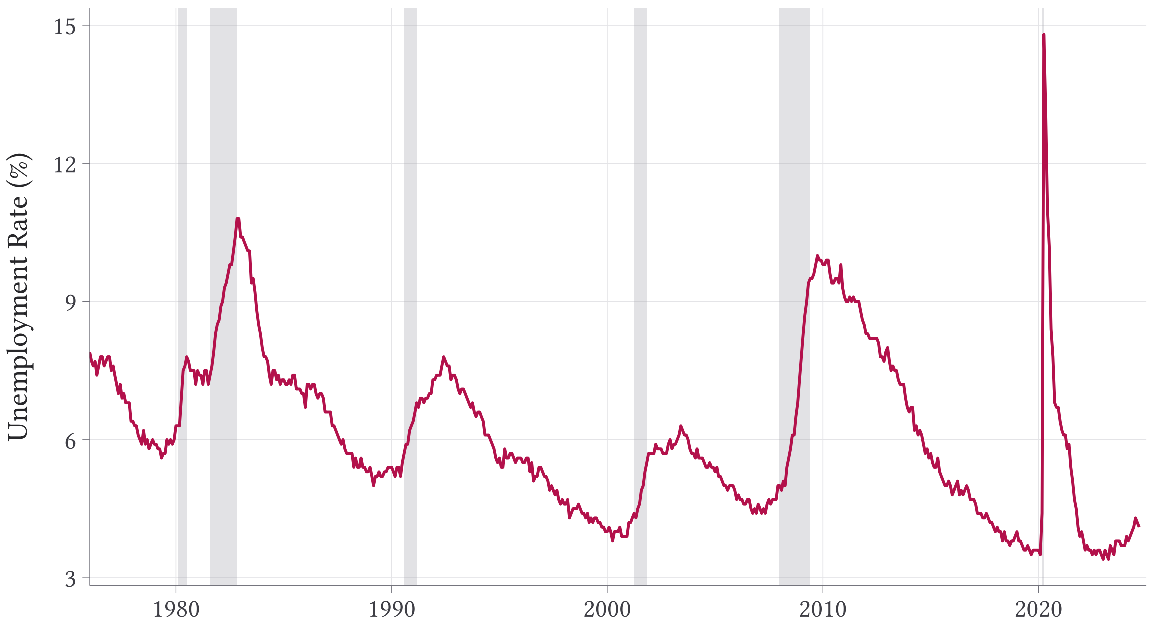
→ Hourly stock price for a company

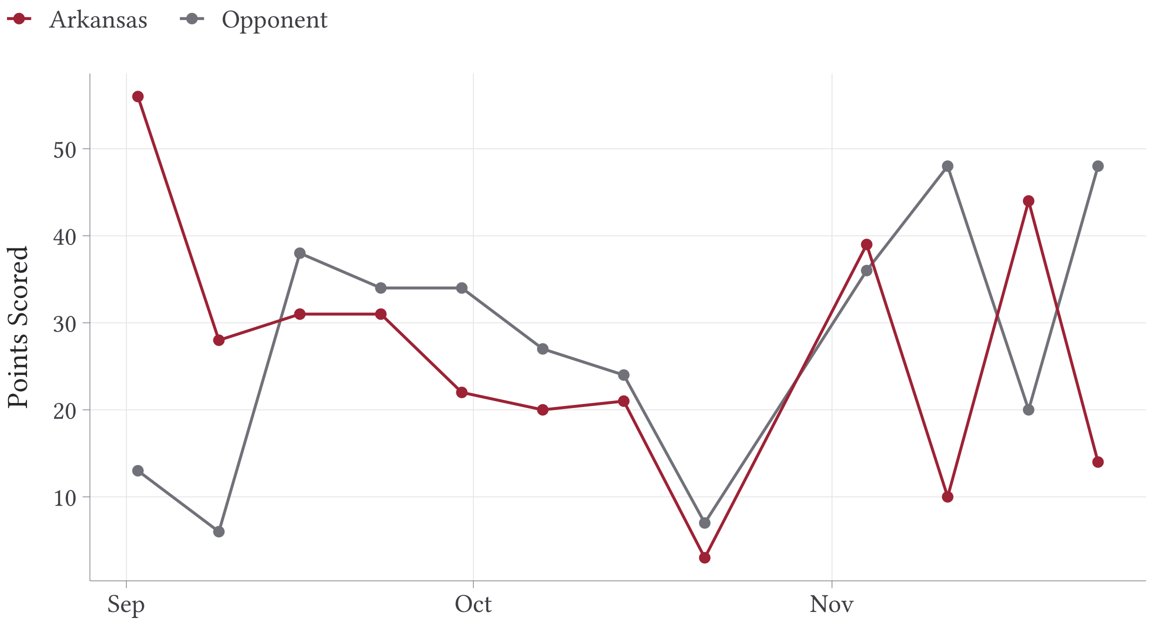
→ Annual data on cigarette consumption per capita in a state

→ A sport's teams number of points scored in games (unequally spaced)

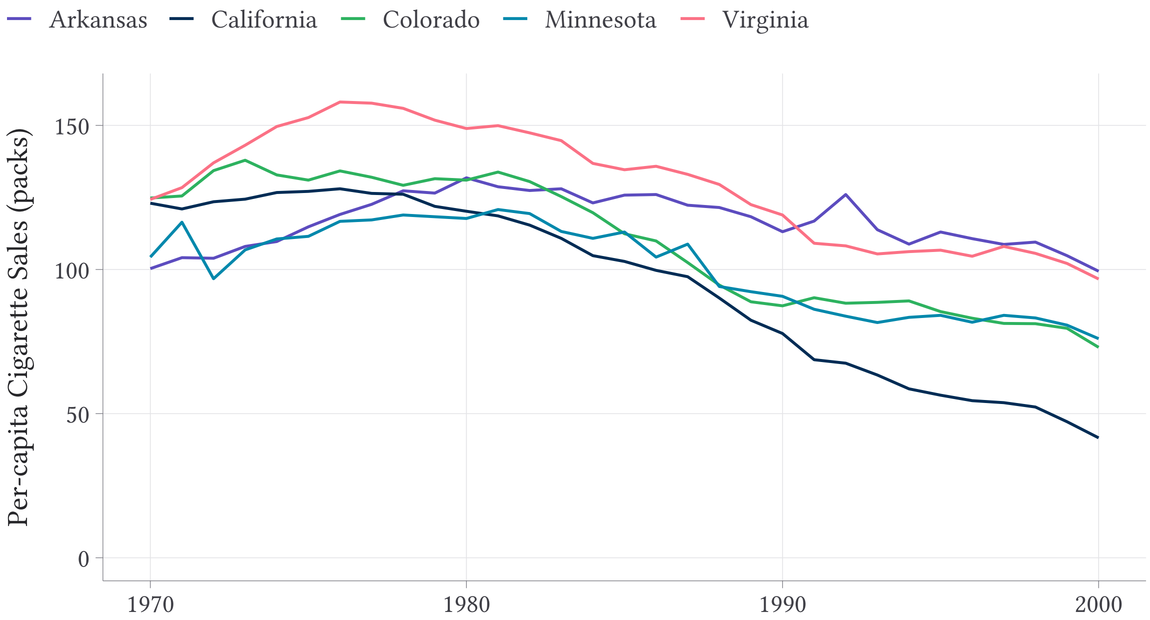
For each example, think through:

1. Is this a regular time-series?
2. Is this a panel dataset?


Unemployment Rate (%)
15
12
9
6
3
1980
1990
2000
2010
2020


Arkansas
Opponent
Points Scored
50
40
30
20
10
Sep
Oct
Nov


Apple Stock Price ($)
0
50
100
150
200
2010
2015
2020
2025


Arkansas
California
Colorado
Minnesota
Virginia
Per-capita Cigarette Sales (packs)
150
100
50
0
1970
1980
1990
2000

What is special about time-series?

In our previous topics, we have been thinking about cross-sectional data

→ Each person in your dataset is an independent draw that provides us with unique information

In cross-sectional data, knowing about one indivdiual does not really tell me much information about another

→ This is not *entirely true*; e.g. worker's in same firm have common experiences, kids in same school have same teacher quality, etc. (hence why we might cluster our standard errors)

What is special about time-series?

In time-series data, knowing last period's value of y_{t-1} is often very useful for this period's value of y_t

$\rightarrow$ This property is essential in forecasting; following a variable over time might let us predict future values

$\rightarrow$ Shocks that happened last period probably still impact me today(!)

What is special about time-series?

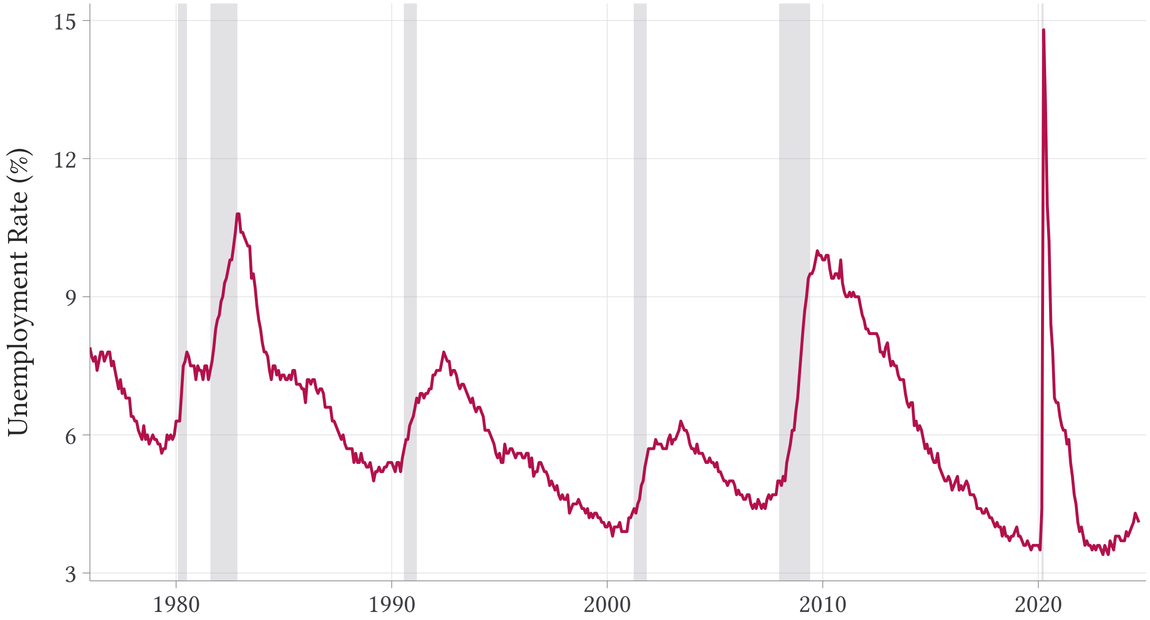
In time-series data, knowing last period's value of y_{t-1} is often very useful for this period's value of y_t

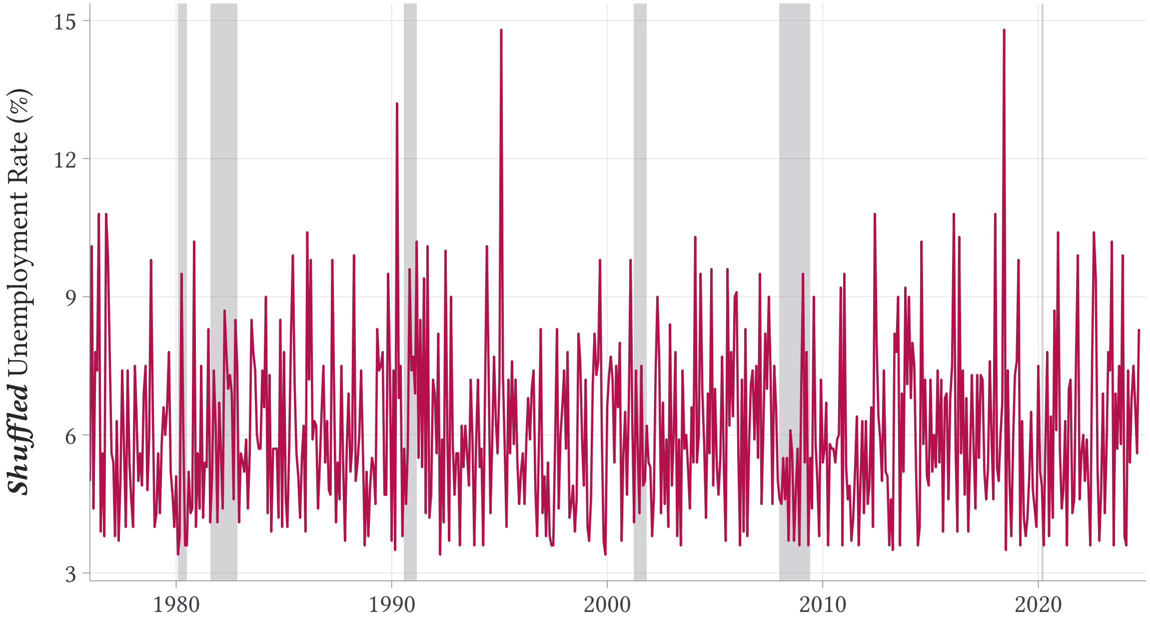
$\rightarrow$ This property is essential in forecasting; following a variable over time might let us predict future values

$\rightarrow$ Shocks that happened last period probably still impact me today(!)

Another way of saying this, is if we randomly shuffled time-series data, we would lose information!

$\rightarrow$ This is not true of a cross-sectional dataset; we can reshuffle rows without problem


Unemployment Rate (%)
15
12
9
6
3
1980
1990
2000
2010
2020



Shuffled Unemployment Rate (%)
15
12
9
6
3
1980
1990
2000
2010
2020

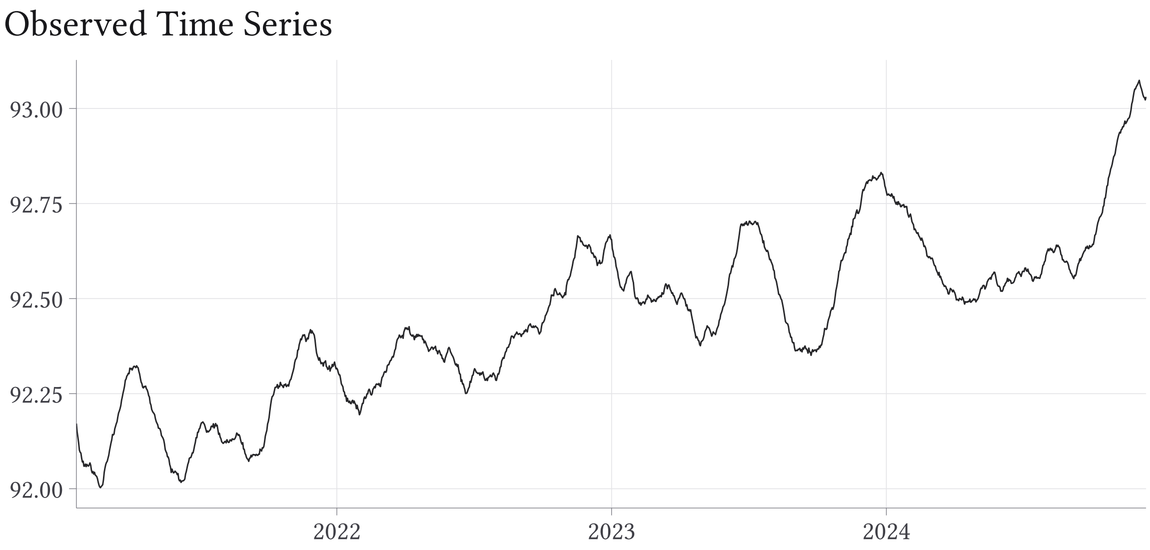
Thinking about inference with time-series

We have a bit of a problem with our time-series:

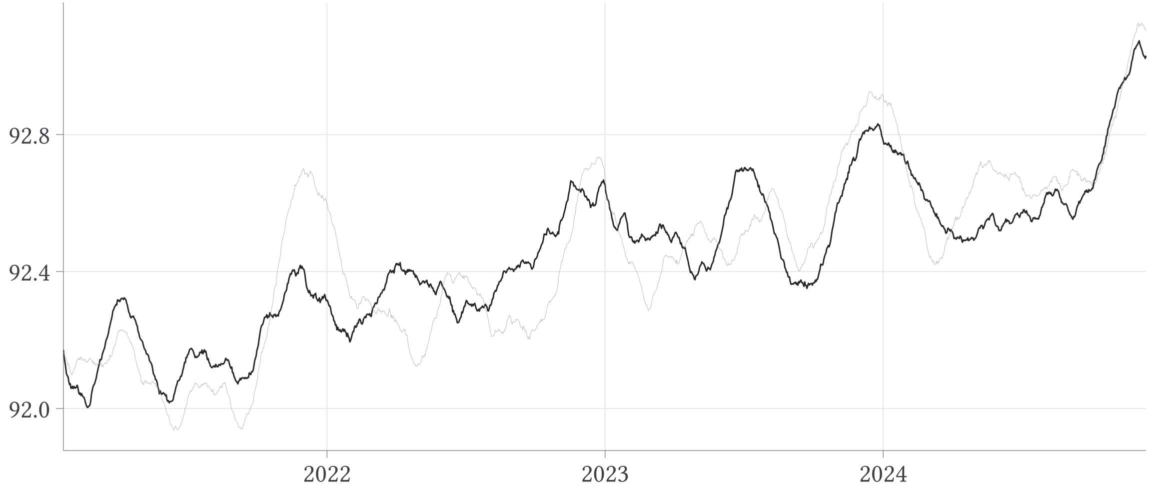
$$y_1, y_2, \ldots, y_T$$

- $\rightarrow$ y_1 is related to y_2
- $\rightarrow$ y_2 is related to y_3
- $\rightarrow$ and so on...

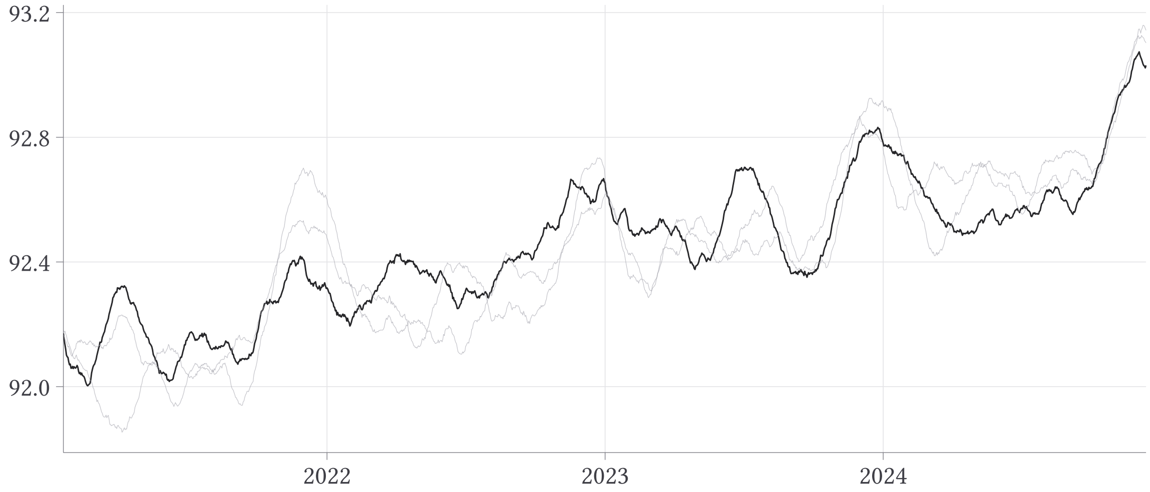
In some sense, we have only a 'single' observation


Observed Time Series
93.00
92.75
92.50
92.25
92.00
2022
2023
2024

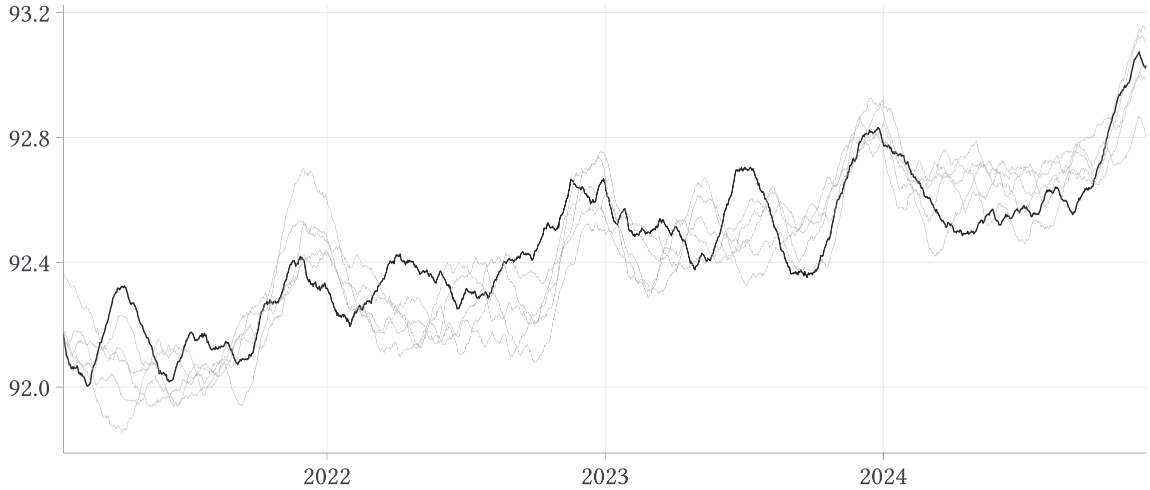
Observed Time Series + 1 Extra Sample



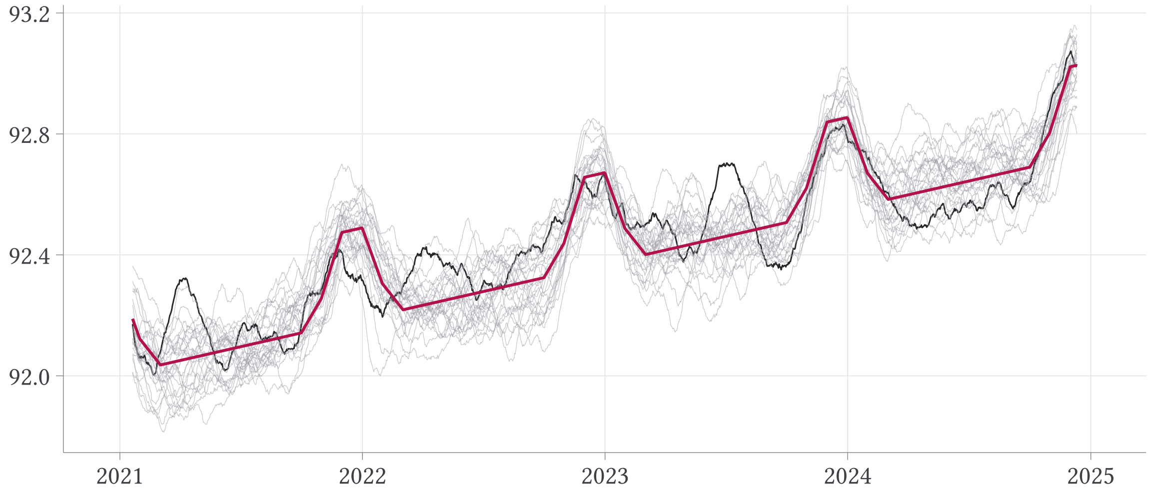
Observed Time Series + 2 Extra Samples



Observed Time Series + 5 Extra Samples



Observed Time Series + 25 Extra Samples; Systematic component: μ_t



Thinking about inference with time-series

$$y_1, y_2, \dots, y_T$$

While every observation might be related to one another, we are typically willing to assume that as you move away in time, observations become less and less correlated.

> “”

Thinking about inference with time-series

$$y_1, y_2, \cdots, y_T$$

Hence, statistical inference is quite a bit more challenging in time-series and requires weak dependency central limit theorems

$\rightarrow$ The intuition is that as you have larger and larger T, the information you have increases

Thinking about inference with time-series

$$y_1, y_2, \cdots, y_T$$

Hence, statistical inference is quite a bit more challenging in time-series and requires weak dependency central limit theorems

$\rightarrow$ The intuition is that as you have larger and larger T, the information you have increases

But, we will not spend much time discussing that in this class:

$\rightarrow$ If you are running time-series regressions, default to Newey West standard errors; in `fixest`, use `vcov = NW(lag = #)`

Introduction to Time-Series

Learning from Time-Series

Time-series Statistics

Smoothing Methods

- Moving Averages
- Trends and Seasonality

Exponential Smoothing

- Simple Exponential Smoothing
- Trends and Seasonality with SES

What we can gain from using time-series

Time-series forecasting can be useful to:

- $\rightarrow$ Predict future values based on past data
- $\rightarrow$ Inform decision-making by anticipating changes over time
- $\rightarrow$ Identify patterns like trends or seasonality

Two goals of time-series

There are two possible goals that we can tackle when working with time-series data:

1. Learn about *persistent* patterns in how y_t evolves over time while ignoring random fluctuations (inference)
   - E.g. learn about seasonality, trends, etc.

2. Predict future values of y_t (forecasting)
   - The above step might be useful in predicting future y, but not necessary (only care about prediction)

Will try to clarify when we are discussing forecasting vs. describing time-series patterns (inference)

Learning from time-series

We observe a set of time-series observations y_t. Think of the observed y as being generated by

$$y_t = \mu_t + \varepsilon_t$$

- $\rightarrow$ μ_t is the 'typical' or 'systematic' value of y at time t
- $\rightarrow$ ε_t is a random fluctuation

Of course, we do not know which fluctuations are due to μ_t changing over time or ε_t changing over time

- $\rightarrow$ Without any more structure, this is is an impossible task

Learning from time-series

$$y_t = \mu_t + \varepsilon_t$$

Say we assume $\mathbb{E}[\varepsilon_t] = 0$

$\rightarrow$ On average over different draws of the time-series, the error term is on average 0

But, our observed time-series is a single draw, so it's not obvious that the noise will 'average away'

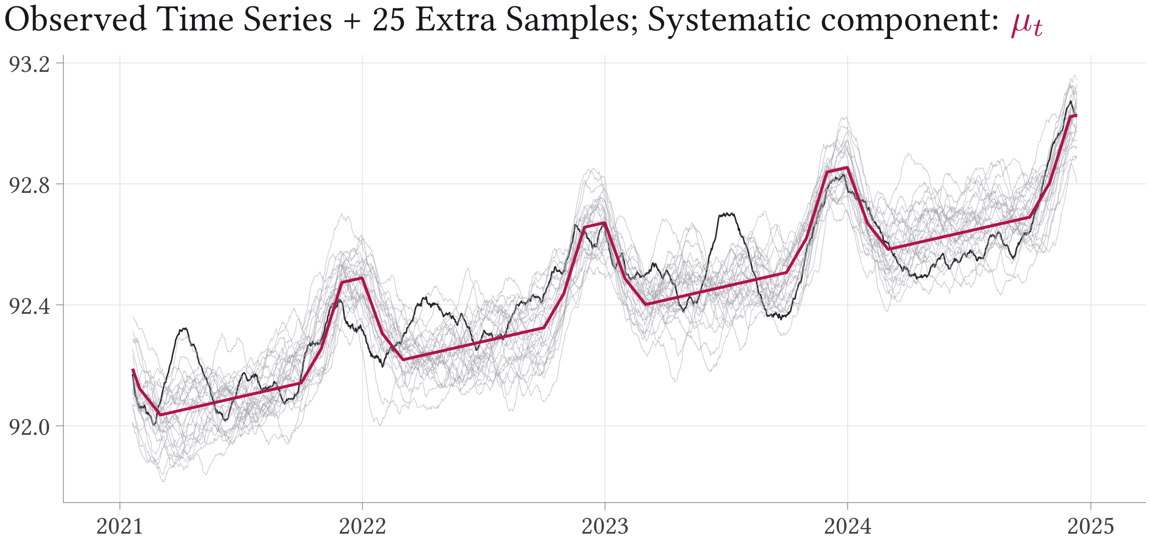
$\rightarrow$ Is the bump in the time-series just a shock that affected the unit for multiple periods or a systematic component

Learning from time-series

$$y_t = \mu_t + \varepsilon_t$$

Two options for solving this:

1. Assume that ε_t is not too persistent, and use some kind of 'smoothing' method to smooth out noise
2. Rely on functional form assumptions and run a time-series regression
   - By pooling over time, we are averaging out noise (but requires us to model μ_t well)


Observed Time Series + 25 Extra Samples; Systematic component: μ_t
93.2
92.8
92.4
92.0
2021
2022
2023
2024
2025

$$y_t = \mu_t + \varepsilon_t$$

Here are some examples of what we can hope to learn using time-series data:

1. Identify seasonality in data
   - Does the change in μ_t over the year follow a standard pattern?
   - E.g. retail sales increasing in December
2. Detect long-term trends and short-term shocks
   - How does μ_t change over time?
   - E.g. trends in GDP changing over time?
   - E.g. recessions
3. Assess how strongly autocorrelated the data is
   - How 'sticky' shocks are from past periods are

Key insight in time-series forecasting

Key Insight: By analyzing the changes across time, we reveal structure and patterns that help in making better predictions. For example:

- → Does yesterday's sales help us learn about what products people will buy today?
- → Do we see an up-swing in jacket sales every October?

Key insight in time-series forecasting

Key Insight: By analyzing the changes across time, we reveal structure and patterns that help in making better predictions. For example:

- → Does yesterday's sales help us learn about what products people will buy today?
- → Do we see an up-swing in jacket sales every October?

Of course, this can fail if the underlying structure of the world changes over time

- → If we are using data from early 2000s on homes, we will surely fail at forecasting during the Great Recession
- → Assumptions on the stability of the time-series is called stationarity

Evaluating forecasting methods

As usual, we can use the mean-squared prediction error to evaluate our models:

$$\mathsf{MSE} = \frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2$$

$\rightarrow$ Typically, will evaluate on the time-series data you do observe

Plotting residuals

It is also common in time-series methods to plot the residuals over time: t on x-axis and $y_t - \hat{y}_t$ on y-axis.

$\rightarrow$ If your forecast is doing a good job, then you should see no pattern in the residuals ("eyeball test")

Beware!! Just because there's no remaining patterns in the residuals, does not mean your model is well fit. You could be overfitted!!

Evaluating forecasting methods

Time-series forecasting is particularly difficult to evaluate

→ Our training data is past-values up until today

→ Our testing data is values in the future

If the structure of the world changes over time, then our testing data *can* look fundamentally different over time

→ Consumer preferences change over time can make predicting future sales hard

Over-fitting

For this reason, we have to be *very* careful when using forecasting methods on time-series

$\rightarrow$ Over-fitting the past data makes us learn 'false' time-series relationships

Statistics of Time-series

For the next few slides, we will discuss some statistics of time-series data that we might be interested in

To review, in cross-sectional data, we mainly cared about:

→ the mean and the variance of a single variable, and

→ the correlation between two variables

Autocovariance

Autocovariance measures the covariance between a variable and a lagged version of itself over successive time periods.

In formal terms, the autocovariance at lag k is defined as:

$$\gamma_k = \text{Cov}(y_t, y_{t-k}) = \mathbb{E}[(y_t - \mu)(y_{t-k} - \mu)]$$

where:

→ μ is the mean of y_t,

→ $\text{Cov}(y_t, y_{t-k})$ is the covariance between y_t and y_{t-k}.

Autocovariance

$$\gamma_k = \text{Cov}(y_t, y_{t-k}) = \mathbb{E}[(y_t - \mu)(y_{t-k} - \mu)]$$

Intuition: Autocovariance helps quantify how much the past values of y move together with its current value.

$\rightarrow$ When y_{t-k} was above the mean, was y_t typically above it's mean?

Autocovariance

$$\gamma_k = \text{Cov}(y_t, y_{t-k}) = \mathbb{E}[(y_t - \mu)(y_{t-k} - \mu)]$$

Intuition: Autocovariance helps quantify how much the past values of y move together with its current value.

$\rightarrow$ When y_{t-k} was above the mean, was y_t typically above it's mean?

In most settings, it is likely that $\gamma_1 \geq \gamma_2 \geq \dots$

$\rightarrow$ More-recent 'shocks' (in say $t-1$) tend to persist for a little and then fade-out

Autocovariance

$$\gamma_k = \text{Cov}(y_t, y_{t-k}) = \mathbb{E}[(y_t - \mu)(y_{t-k} - \mu)]$$

As an aside, note that when $k = 0$,

$$\gamma_0 = \text{Cov}(y_t, y_t) = \text{Var}(y_t)$$

Autocorrelation

Autocorrelation is the normalized version of autocovariance. It measures the correlation of a variable with its lagged values.

The autocorrelation at lag k is defined as:

$$\rho_k = \frac{\gamma_k}{\text{Var}(y_t)} = \frac{\text{Cov}(y_t, y_{t-k})}{\text{Var}(y_t)}$$

where:

- $\rightarrow$ γ_k is the autocovariance at lag k,
- $\rightarrow$ γ_0 is the variance of y_t (i.e., autocovariance at lag 0).

Autocorrelation

$$\rho_k = \frac{\gamma_k}{\text{Var}(y_t)} = \frac{\text{Cov}(y_t, y_{t-k})}{\text{Var}(y_t)}$$

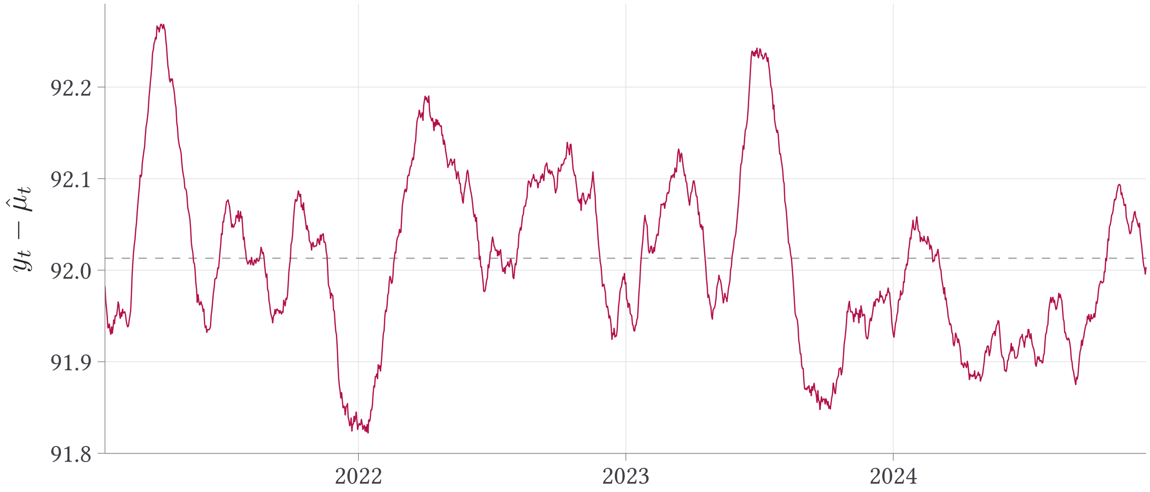
Intuition: Autocorrelation tells us the strength of the relationship between y_t and its past values. It ranges between -1 and 1.

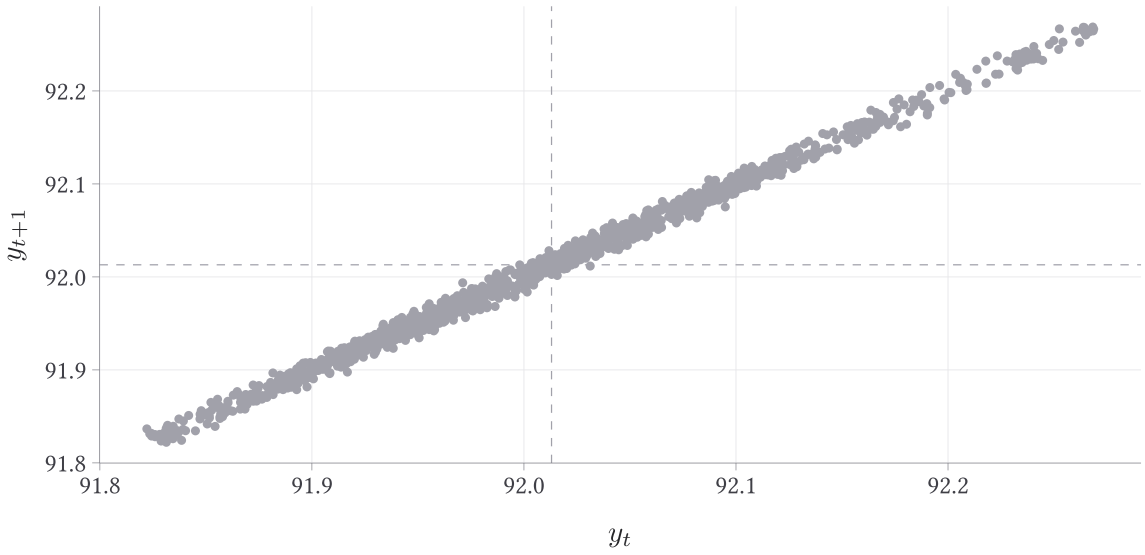
Examples of Covariance

Let's give two examples to help build intuition:

1. The first time-series will have very significant autocorrelation
2. The second time-series will have near zero autocorrelation

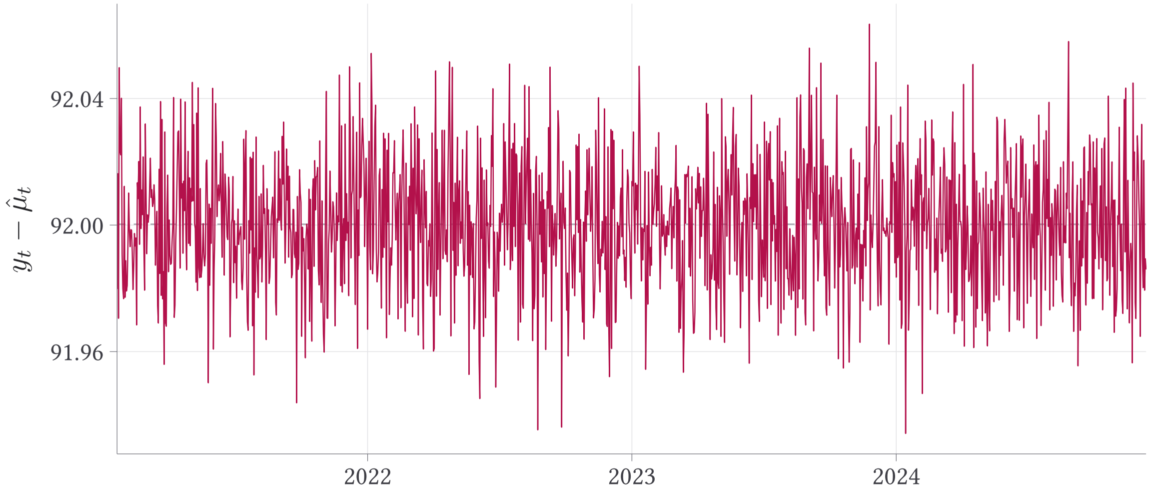
Observed Time Series

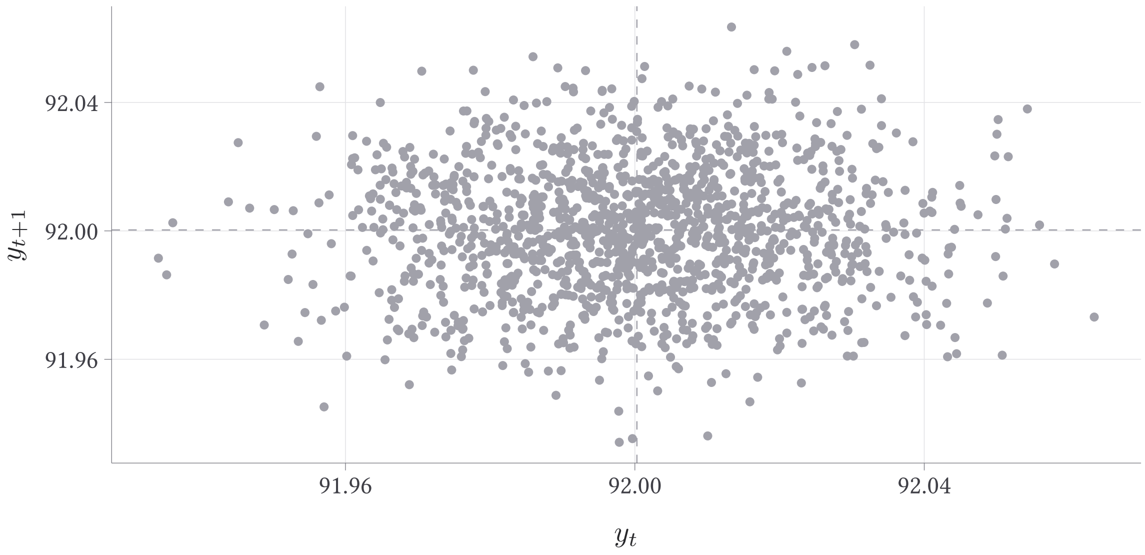




y_{t+1}
92.2
92.1
92.0
91.9
91.8
91.8
91.9
92.0
92.1
92.2
y_t

Observed Time Series





92.04
92.00
91.96
y_{t+1}
91.96
92.00
92.04
y_t

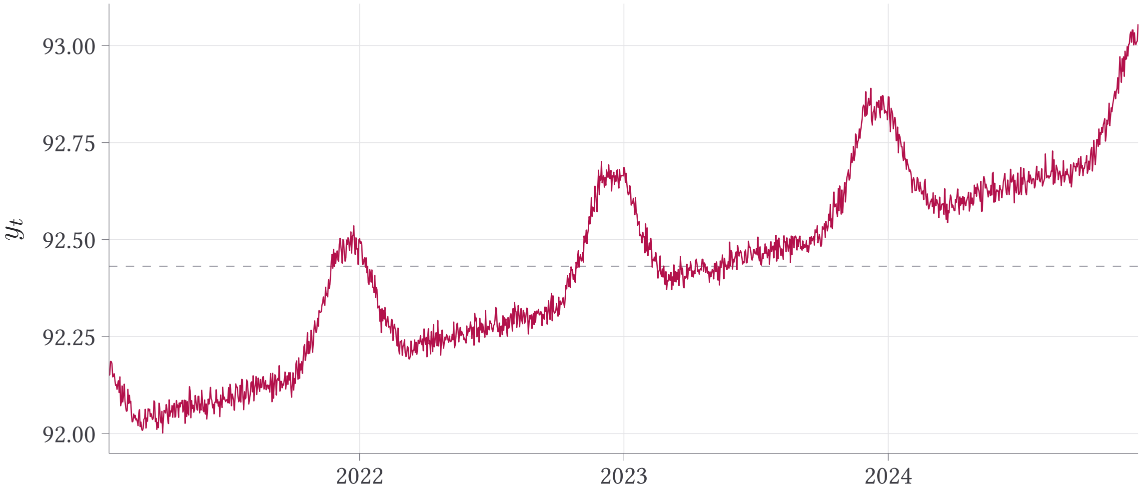
Autocorrelation with trends

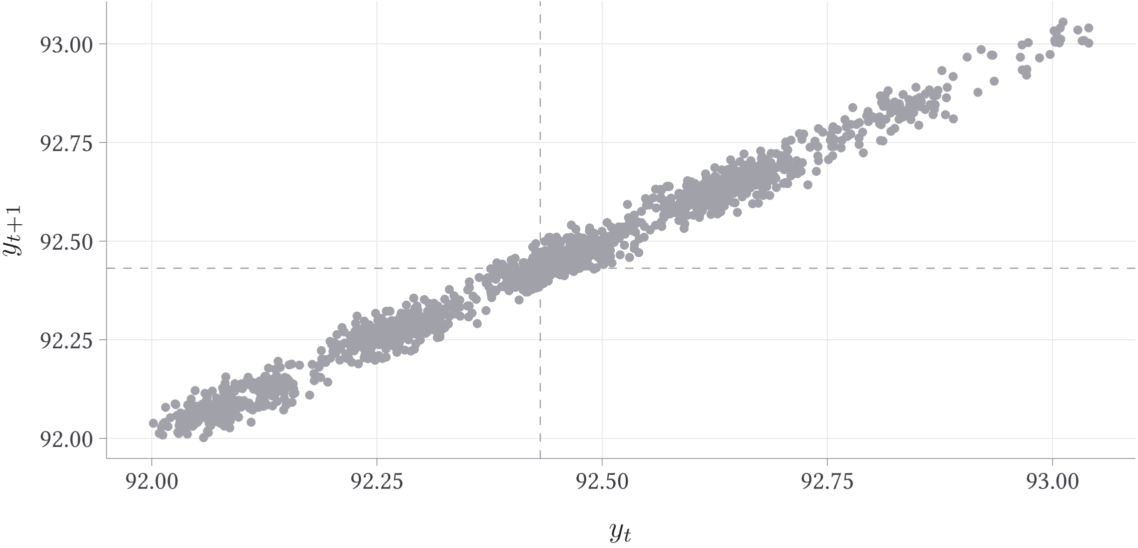
When the data is trending in a direction (e.g. up over time), the data will exhibit a strong autocorrelation.

E.g. we generate data with a trend and a winter seasonal effect plus an error term ε_t that is normal and independent in each period

$\rightarrow$ The error term ε_t exhibitis zero autocorrelation

Observed Time Series




93.00
92.75
92.50
92.25
92.00
y_{t+1}
92.00
92.25
92.50
92.75
93.00
y_t

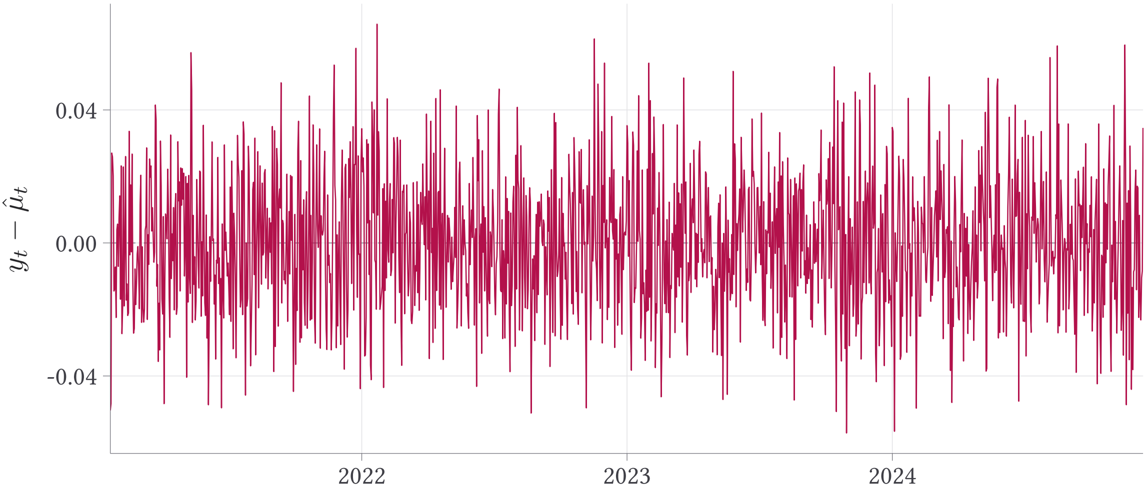
Autocorrelation with trends

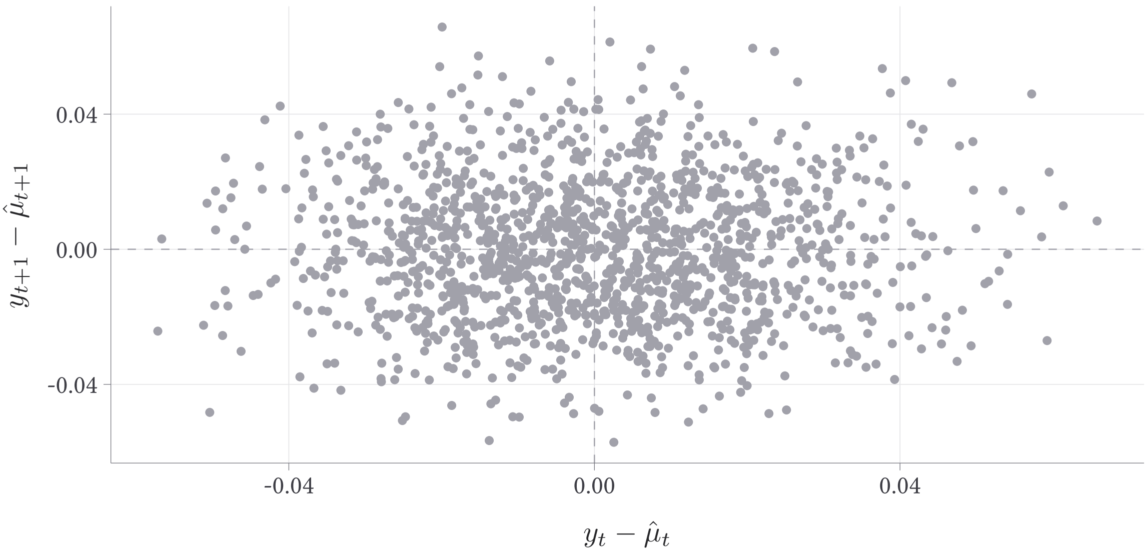
Now, let's estimate a time-series regression (we will see how in the future) that estimates the trend and seasonal effects and subtracts them off

Then, we can evaluate the autocorrelation of the "de-trended data": $y_t - \hat{y}_t$

$\rightarrow$ In our example, we generated ε_t with zero autocorrelation, so let's see how that looks

Residualized Time Series




0.04
0.00
-0.04
-0.04
0.00
0.04
$y_{t+1} - \hat{\mu}_{t+1}$
$y_t - \hat{\mu}_t$

Unemployment Rate Example

In the unemployment example, the time-series

$$\hat{\gamma}_1 = \text{ Cov}(y_t, y_{t-1}) = 2.968 \quad \text{and} \quad \hat{\rho}_1 = 0.961$$

$\rightarrow$ Unsurprisingly the correlation of unemployment from 1-month to the next is very strong

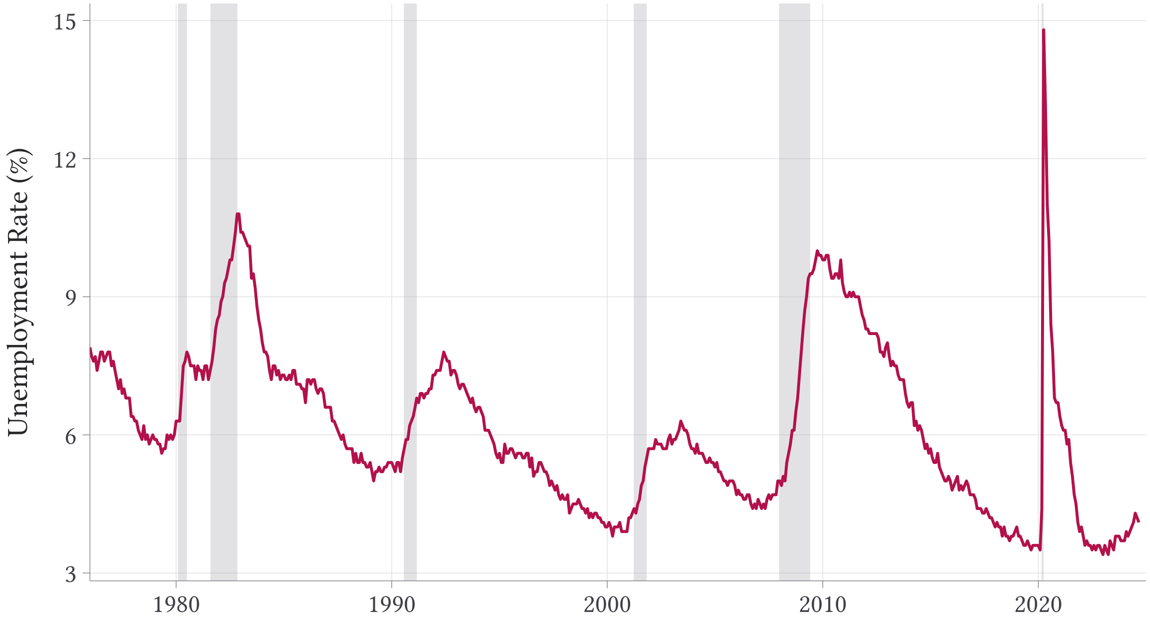
Unemployment Rate Example

In the unemployment example, the time-series

$$\hat{\gamma}_1 = \text{ Cov}(y_t, y_{t-1}) = 2.968 \quad \text{and} \quad \hat{\rho}_1 = 0.961$$

$\rightarrow$ Unsurprisingly the correlation of unemployment from 1-month to the next is very strong

This is useful for forecasting; a very strong autocorrelation tells us that recent values of y should be useful for predicting future values of y



Unemployment Rate (%)
15
12
9
6
3
1980
1990
2000
2010
2020

Unemployment Rate Example

Let's look at the correlation unemployment over 12 periods (year to year)

$$\hat{\rho}_{12} = 0.659$$

$\rightarrow$ Shocks to last year's unemployment seem to 'persist' into the current period

Unemployment Rate Example

If we use the reshuffled gdp data, what do we think the autocorrelation may be?

Unemployment Rate Example

If we use the reshuffled gdp data, what do we think the autocorrelation may be?

$$\hat{\rho}_{1,\texttt{reshuffled}} = -0.03081183$$

When we completely randomly shuffled the data, we have destroyed any autocorrelation!

$\rightarrow$ This makes sense. If I reshuffled the data, knowing last month's (reshuffled) unemployment is no longer useful for predicting this month's (reshuffled) unemployment rate

How to calculate in R

The first thing we need to do is calculate the sample mean $\bar{y} = \frac{1}{T}\sum_{t=1}^{T} y_t$ using `mean(y)`.

We want two vectors

$$\begin{bmatrix} y_1 \\ \vdots \\ y_{T-1} \\ y_T \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} y_2 \\ \vdots \\ y_T \\ \texttt{NA} \end{bmatrix}$$

The first one is our original vector `y` and we need `L1_y` ("lag 1 y").

Then, calculate $\frac{1}{T}\sum_{t=2}^{T}(y_t - \bar{y})(y_{t-1} - \bar{y})$

How to calculate in R

First, we will do it by hand to make sure we follow all the steps

$\rightarrow$ Note this requires the time-series to be sorted in order!

```
y <- 1:10
T <- length(y)
y_dm <- y - mean(y)
sum(y_dm[1:(T - 1)] * y_dm[2:T]) / T
#> [1] 5.775
```

How to calculate in R

Or we can use the `acf` function to make things way easier

```
# Or, using a function
acf(y, lag.max = 1, type = "covariance", plot = FALSE)
#>    0    1
#> 8.25 5.78

acf(y, lag.max = 1, plot = FALSE)
#> Autocorrelations of series 'y', by lag
#>
#>   0   1
#> 1.0 0.7
```

Introduction to Time-Series

Learning from Time-Series

Time-series Statistics

Smoothing Methods

- Moving Averages
- Trends and Seasonality

Exponential Smoothing

- Simple Exponential Smoothing
- Trends and Seasonality with SES

Regular Time Series

Before we begin, it is *very important* to note that all of these methods are going to rely on *regular* time series

$\rightarrow$ Assuming that each observation is equally spaced apart and there is no missingness

Use these method with irregular time series, you will get weird results

Smoothing Methods

Recall we said y_t was generated by

$$y_t = \mu_t + \varepsilon_t$$

$\rightarrow$ μ_t is the 'typical' or 'systematic' value of y at time t

The idea of smoothing methods is to use time periods right around period t to estimate a smoothed value at period t

$\rightarrow$ Want to "smooth" over random fluctuations

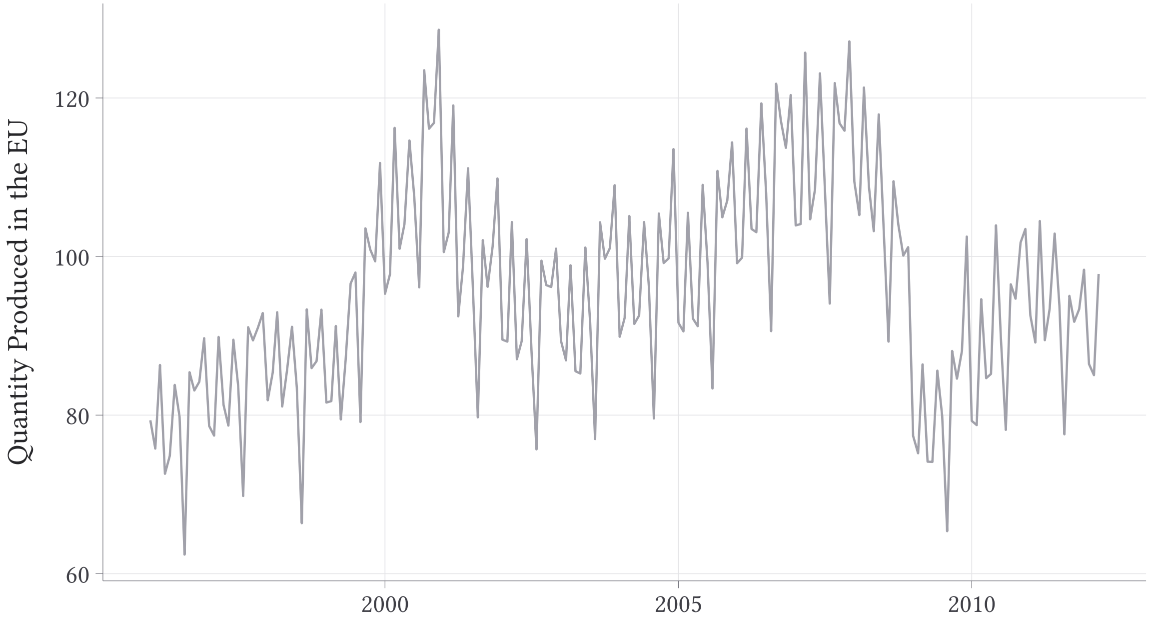
Example Electrical Manufacturing in the EU

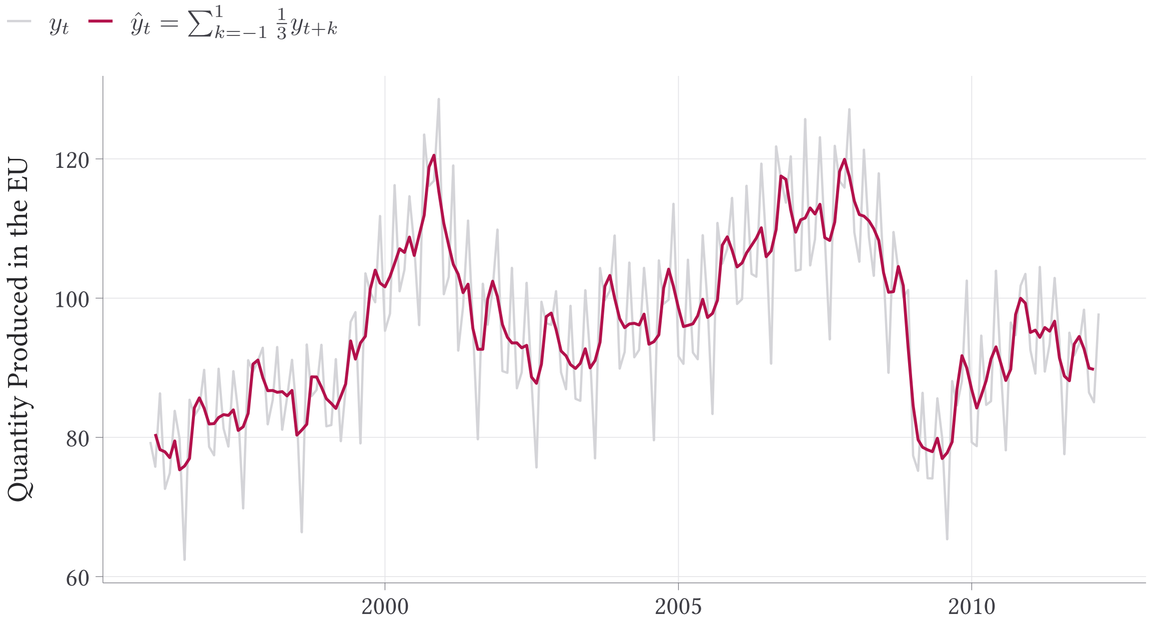
On the following slide I'm going to show you production figures across the European Union

→ Time-series data is on electircal manufactuing (computers and other technology) and is from EUROSTAT

When looking at this figure, try to imagine the 'systematic' component versus the random fluctuation ε_t

$$y_t = \mu_t + \varepsilon_t$$


Quantity Produced in the EU
120
100
80
60
2000
2005
2010



y_t
$\hat{y}_t = \sum_{k=-1}^{1} \frac{1}{3} y_{t+k}$
Quantity Produced in the EU
120
100
80
60
2000
2005
2010

In the previous figure, I created a moving average where I estimated the μ_t as being an average of $y_{t-2}, y_{t-1}, y_t, y_{t+1}, y_{t+2}$

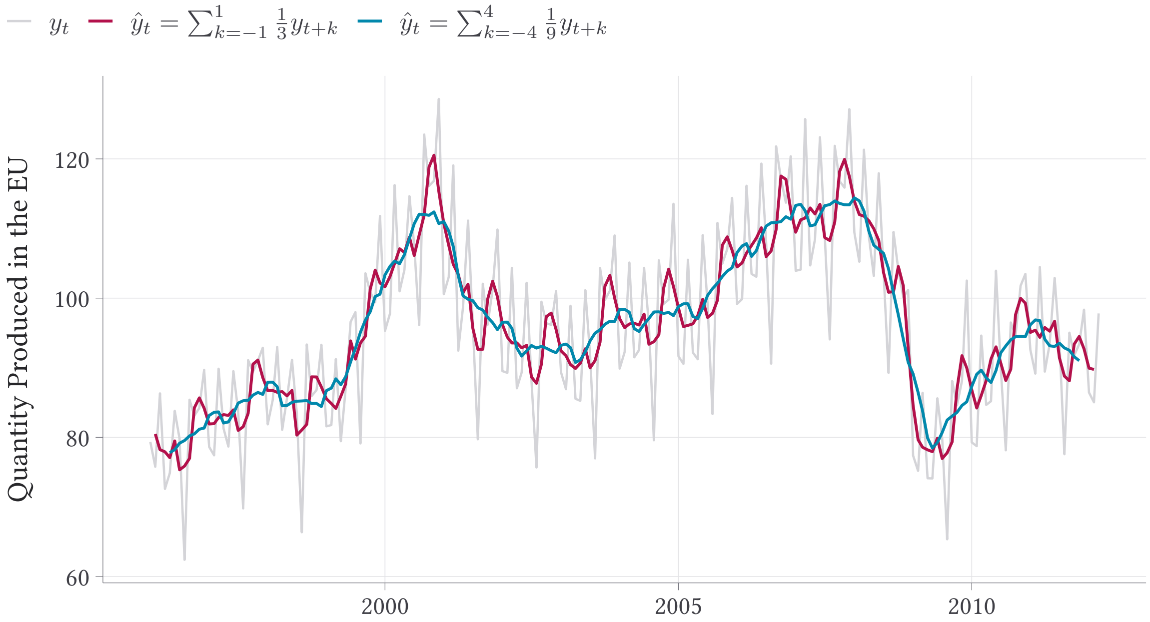
→ This helped to smooth out some of the random fluctuations, perhaps better isolating systematic trends in y_t

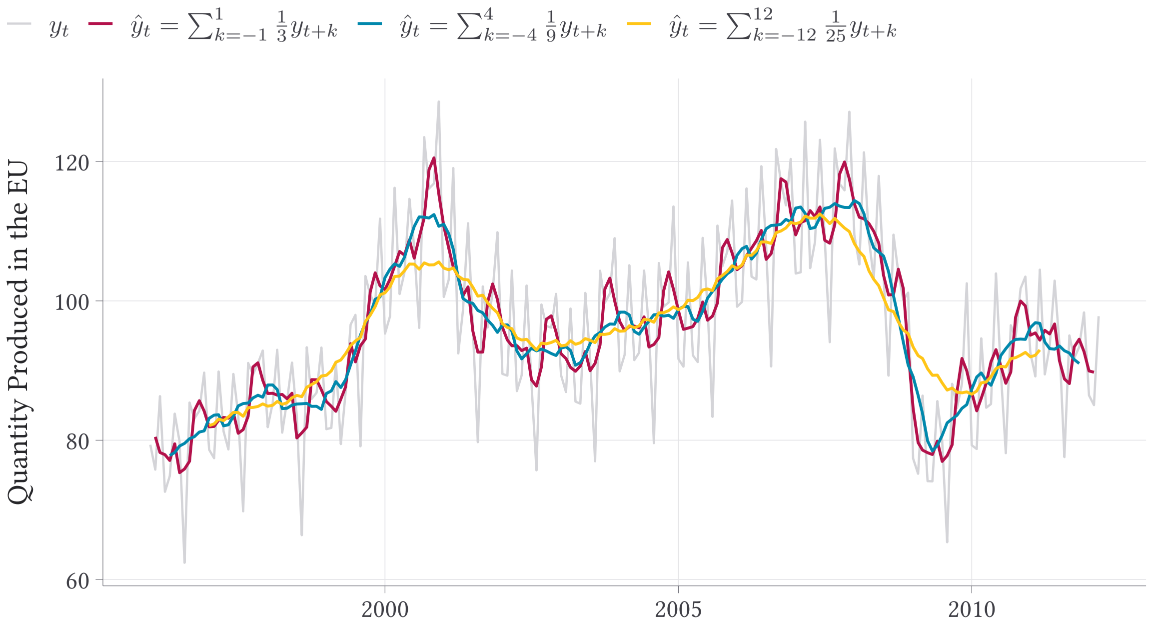
Smoothing methods

In the previous figure, I created a moving average where I estimated the μ_t as being an average of $y_{t-2}, y_{t-1}, y_t, y_{t+1}, y_{t+2}$

$\rightarrow$ This helped to smooth out some of the random fluctuations, perhaps better isolating systematic trends in y_t

What happens if I average a bit more over time?


y_t
$\hat{y}_t = \sum_{k=-1}^{1} \frac{1}{3} y_{t+k}$
$\hat{y}_t = \sum_{k=-4}^{4} \frac{1}{9} y_{t+k}$
Quantity Produced in the EU
120
100
80
60
2000
2005
2010



y_t
$\hat{y}_t = \sum_{k=-1}^{1} \frac{1}{3} y_{t+k}$
$\hat{y}_t = \sum_{k=-4}^{4} \frac{1}{9} y_{t+k}$
$\hat{y}_t = \sum_{k=-12}^{12} \frac{1}{25} y_{t+k}$
Quantity Produced in the EU
120
100
80
60
2000
2005
2010

Moving average

In general, our moving average can be calculated as follows:

$$\hat{y}_t = \sum_{k=-K}^{K} \frac{1}{2K+1} y_{t+k}$$

This is just the sample mean using observations within $\pm K$ periods of t

$\rightarrow$ K is the number of observations on each side of y_t we include

$\rightarrow$ $2K + 1$ is the number of observations. Note $+1$ because we include y_t

Moving average

In general, our moving average can be calculated as follows:

$$\hat{y}_t = \sum_{k=-K}^{K} \frac{1}{2K+1} y_{t+k}$$

This is just the sample mean using observations within $\pm K$ periods of t

$\rightarrow$ K is the number of observations on each side of y_t we include

$\rightarrow$ $2K + 1$ is the number of observations. Note $+1$ because we include y_t

I will show you how to do this using the `slider` package in `R`

What happened to the end points?

$$\hat{y}_t = \sum_{k=-K}^{K} \frac{1}{2K+1} y_{t+k}$$

Note when calculating a rolling-average, we will face problems on either end of our observed time-series

$\rightarrow$ E.g. for my first observation, I do not have the y from the period before

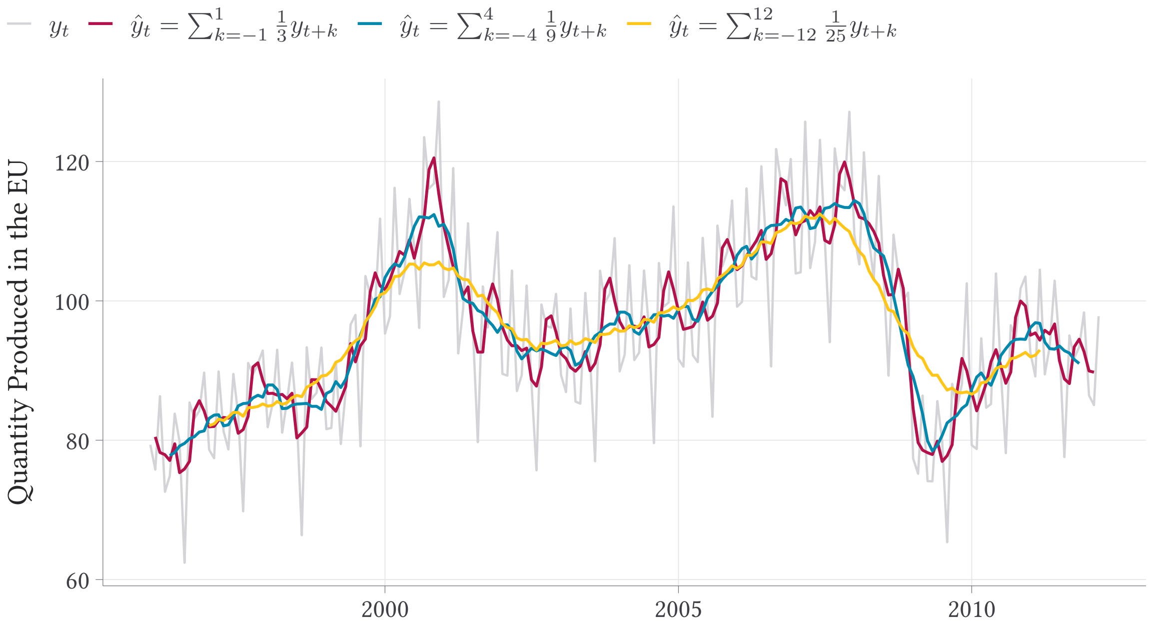
What happened to the end points?

$$\hat{y}_t = \sum_{k=-K}^{K} \frac{1}{2K+1} y_{t+k}$$

Note when calculating a rolling-average, we will face problems on either end of our observed time-series

$\rightarrow$ E.g. for my first observation, I do not have the y from the period before

That is what causes the truncated ends of the smoothed time-series graph



y_t
$\hat{y}_t = \sum_{k=-1}^{1} \frac{1}{3} y_{t+k}$
$\hat{y}_t = \sum_{k=-4}^{4} \frac{1}{9} y_{t+k}$
$\hat{y}_t = \sum_{k=-12}^{12} \frac{1}{25} y_{t+k}$
Quantity Produced in the EU
120
100
80
60
2000
2005
2010

Problems with moving averages

"Over-smoothing"

When K is large, we are using observations quite far away from the current period (e.g. using data from 12 months ago)

$\rightarrow$ This prevents $\hat{y}_t$ from being driven too much by the current period's observation (for better or worse!)

Problems with moving averages

"Over-smoothing"

When K is large, we are using observations quite far away from the current period (e.g. using data from 12 months ago)

$\rightarrow$ This prevents $\hat{y}_t$ from being driven too much by the current period's observation (for better or worse!)

When we have a high-degree of smoothing, our smoothed time-series misses out on true shocks to μ_t that are short-lived

$\rightarrow$ In our previous example, the overly-smoothed version misses the short jump in manufacturing in the early 2000s

Problems with moving averages

Seasonality

Say you had time-series data on candy sales over the course of the last decade

$\rightarrow$ You would see a bump every October for Halloween (i.e. it is part of μ_t)

Even a moderately small $K = 1$ would make $\hat{y}_t$ be too small in October

$\rightarrow$ Temporary seasonal swings in y (i.e. last only a period or two) are going to be lost

Selecting K

There is a trade-off at play

$\rightarrow$ Using a small K only uses the most recent information (perhaps better picking up on recent shocks)

$\rightarrow$ Using a larger K helps average over non-persistant random noise

Selecting K

There is a trade-off at play

- $\rightarrow$ Using a small K only uses the most recent information (perhaps better picking up on recent shocks)
- $\rightarrow$ Using a larger K helps average over non-persistant random noise

This is an example of a *bias-variance tradeoff*

- $\rightarrow$ Smaller K lowers bias, but increases variance

Mean-squared prediction error

Say we wanted to use data to tell us the 'best' K to use for forming $\hat{y}_t$

We could search over $K = 0, 1, 2, 3, \ldots$ and see which gives us the smallest mean-squared prediction error:

$$\mathsf{MSE} = \frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2$$

Mean-squared prediction error

$$\text{MSE} = \frac{1}{T}\sum_{t=1}^{T}(y_t - \hat{y}_t)^2$$

For smoothing averages, when $K = 0$, we just use $\hat{y}_t = y_t$ and we have MSE of 0

$\rightarrow$ As K increases, the MSE necessarily grows

Trying to select K this way fails utterly because we are using our training data as our testing data!

Seasonality, Trends, and Shocks

It is often desirable to break up μ_t into two components:

$$y_t = T_t + S_t + \varepsilon_t$$

- → S_t is the seasonality term (e.g. year over year)
- → T_t is the trend-term
- → and ε_t is the remaining noise (random fluctuations)

Let's look into how we can try to separate trends from seasonality

- → This section will cover the 'classical' decomposition (see 3.4 in Forecasting: Principles and Practices)

Moving average to remove seasonality, S_t

It turns out, there is a particular moving average that can remove seasonality from the data

→ For this example, we will think of monthly data and try to remove annual trend (you can similarly do this with quarterly data)

Remember we can write our general moving average as

$$\hat{y}_t = \sum_{k=-K}^{K} w_k y_{t+k}$$

→ If we choose K and w_k right, we will try to remove seasonality

Moving average to remove seasonality, S_t

$$
\begin{aligned}
\hat{y}_t = &\frac{1}{24}y_{t-6} + \frac{1}{12}y_{t-5} + \frac{1}{12}y_{t-4} + \frac{1}{12}y_{t-3} + \frac{1}{12}y_{t-2} + \frac{1}{12}y_{t-1} \\
&+ \frac{1}{12}y_t \\
&+ \frac{1}{12}y_{t+1} + \frac{1}{12}y_{t+2} + \frac{1}{12}y_{t3+3}\frac{1}{12}y_{t+4} + \frac{1}{12}y_{t+5} + \frac{1}{24}y_{t+6}
\end{aligned}
$$

Basically a $\pm K$ smoothing average, but first and last get a half the weight

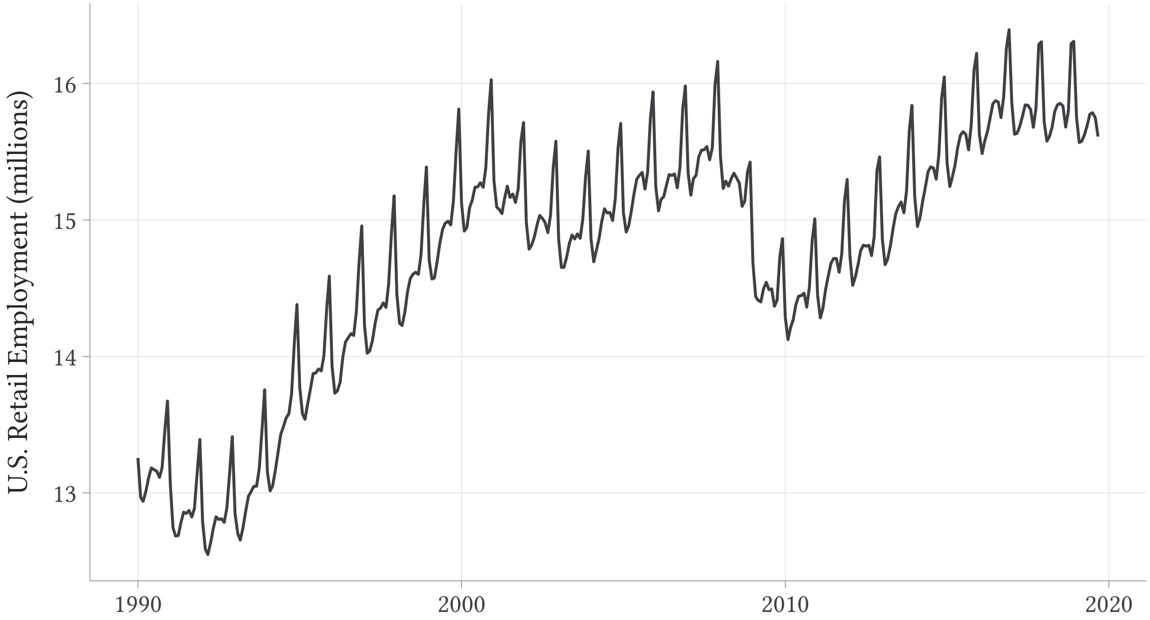
Moving average to remove seasonality, S_t

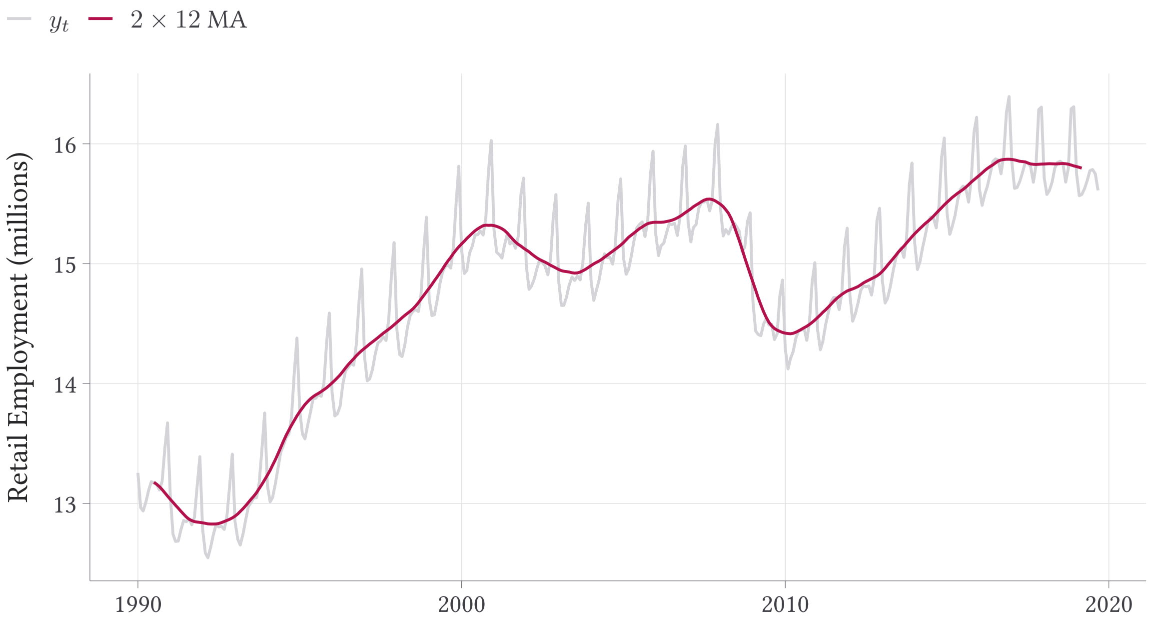
Or, can do the following:

$\rightarrow$ 12-month rolling average

$\rightarrow$ 2-month rolling average of the 12-month rolling average

Sometimes called the 2×12 MA


U.S. Retail Employment (millions)
16
15
14
13
1990
2000
2010
2020



y_t
2×12 MA
Retail Employment (millions)
16
15
14
13
1990
2000
2010
2020

2×12 MA

Our 2×12 moving-average serves as the classical estimate of $\hat{T}_t$, i.e. the time-trend

2×12 MA

Our 2×12 moving-average serves as the classical estimate of $\hat{T}_t$, i.e. the time-trend

For quarterly data, you would do

$$\hat{y}_t = \frac{1}{8}y_{t-2} + \frac{1}{4}y_{t-1} + \frac{1}{4}y_t + \frac{1}{4}y_{t+1} + \frac{1}{8}y_{t+2}$$

De-trending our data

Now, we can "de-trend" our data by forming $y_t - \hat{T}_t$

$\rightarrow$ $\hat{T}_t$ is our 2×12 moving average estimate

What remains is

$$y_t - \hat{T}_t \approx S_t + \varepsilon_t$$

Estimating seasonality

We want to know how does $y_t - \hat{T}_t$ cycle throughout the year

$\rightarrow$ E.g. is retail employment systematically higher in November and December?

The classical way to estimate this is take the average of $y_t - \hat{T}_t$ separately for each month

$\rightarrow$ Each month's average serves as the estimated month's "seasonal trend", $\hat{S}_t$

- Takes the same value year over year

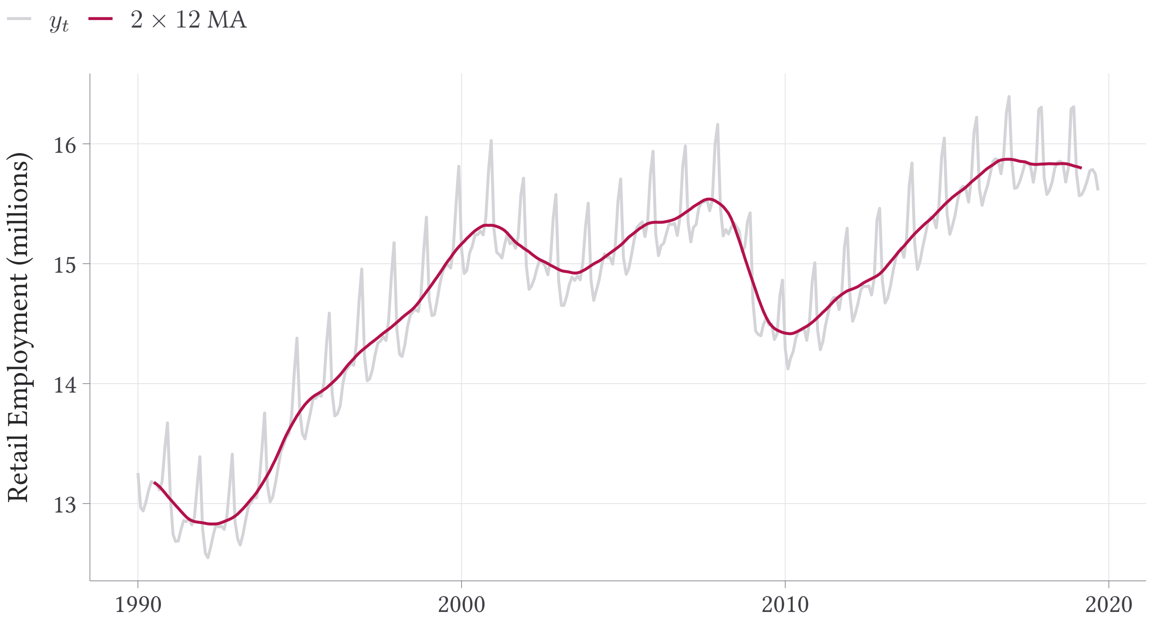
Seasonality estimation in R

The classical way to estimate this is take the average of $y_t - \hat{T}_t$ separately for each month

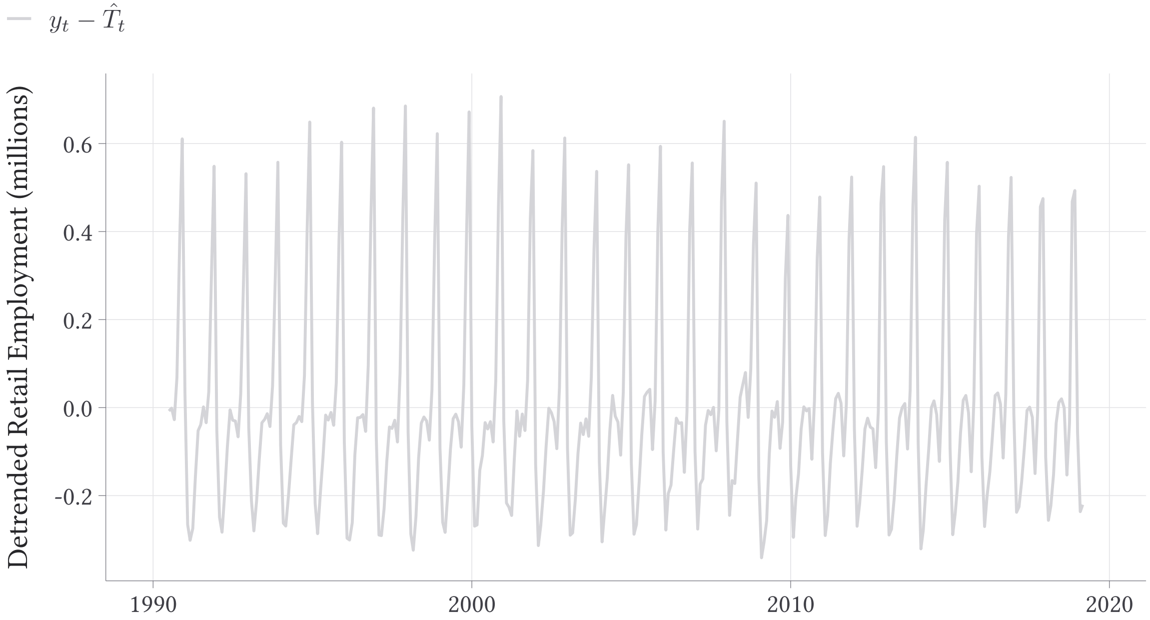
$\rightarrow$ This can be done by regression $y_t - \hat{T}_t$ on a set of month indicators (and no intercept)

In R, this can be done with

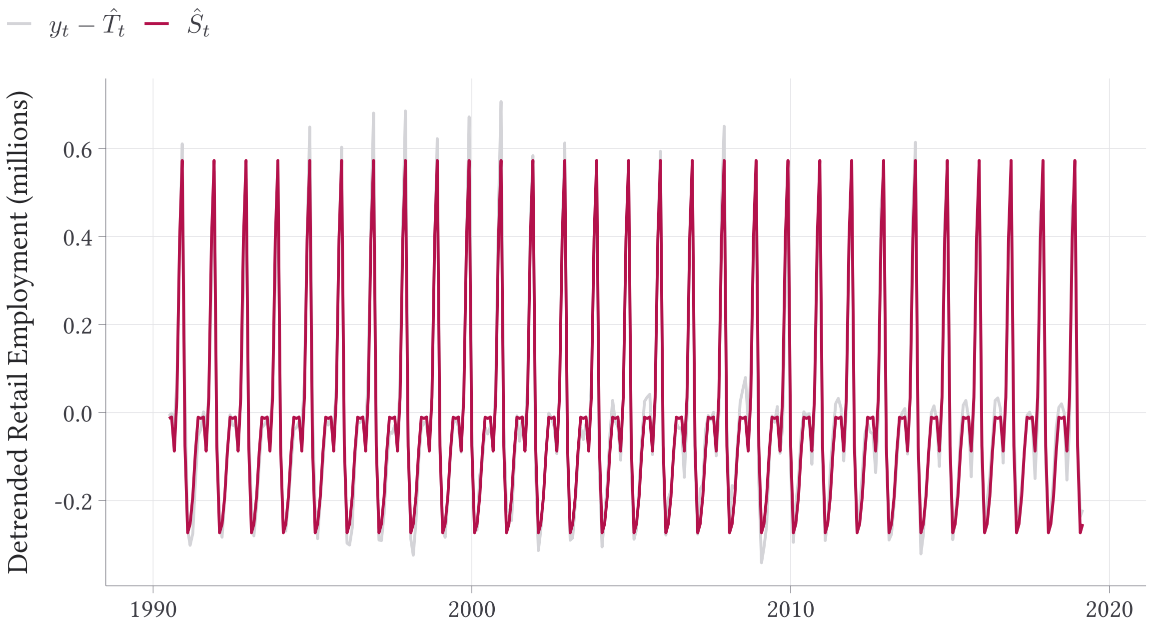
```
feols(y_minus_trend ~ 0 + i(month(date)), data = df)
```



y_t
2×12 MA
Retail Employment (millions)
16
15
14
13
1990
2000
2010
2020



$y_t - \hat{T}_t$
Detrended Retail Employment (millions)
0.6
0.4
0.2
0.0
-0.2
1990
2000
2010
2020



$y_t - \hat{T}_t$
$\hat{S}_t$
Detrended Retail Employment (millions)
0.6
0.4
0.2
0.0
-0.2
1990
2000
2010
2020

Residual

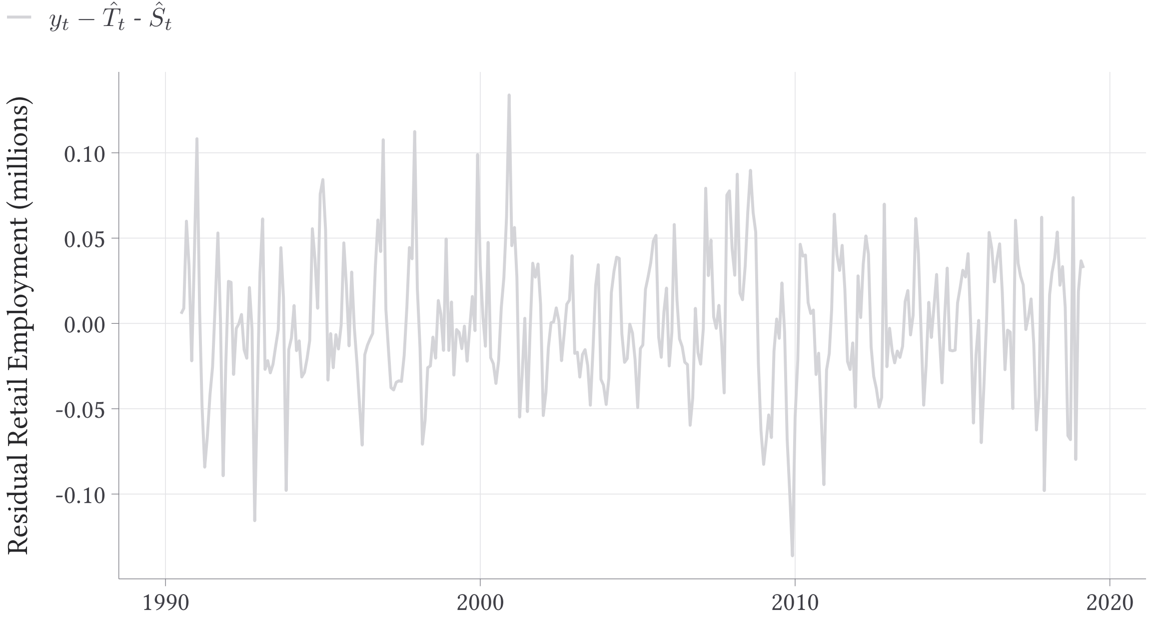
$$y_t - \hat{T}_t - \hat{S}_t \approx \varepsilon_t$$

$\rightarrow$ $\hat{T}_t$ is the 2×12 moving average estimate of trends

$\rightarrow$ $\hat{S}_t$ is the monthly average of $y_t - \hat{T}_t$

What remains after this is a de-trended and de-seasoned data, i.e. random fluctuations

$\rightarrow$ Should visually inspect this to see how good we did at removing trends and seasonality


$y_t - \hat{T}_t$ - $\hat{S}_t$
Residual Retail Employment (millions)
0.10
0.05
0.00
-0.05
-0.10
1990
2000
2010
2020

"Rules" for forecasting

Typically, we will want to only use data from period t or prior in our model

$\rightarrow$ E.g. I can't use y_{t+2} to predict tomorrow's y_{t+1}

When predicting the future, I can't view use data from the future

$\rightarrow$ So the model I learn can only use past data

Simplest forecasting method

The simplest forecasting method is to use y_{t-1}, the previous period's value, as the forecast for y_t:

$$\hat{y}_t = y_{t-1}$$

This method is going to use only information from the most recent observations

$\rightarrow$ Maybe the most recent observation is the most-relevant for predicting today

- I.e. autocorrleation is high

$\rightarrow$ If μ_t is really wild (i.e. no trends/seasonality), then we should only use recent information

Cons of using y_{t-1} as a forecast

Using y_{t-1} could fail when:

$\rightarrow$ The data has trends or seasonality that y_{t-1} doesn't capture

- Using August's jacket sales to predict September's jacket sales will not do well

Cons of using y_{t-1} as a forecast

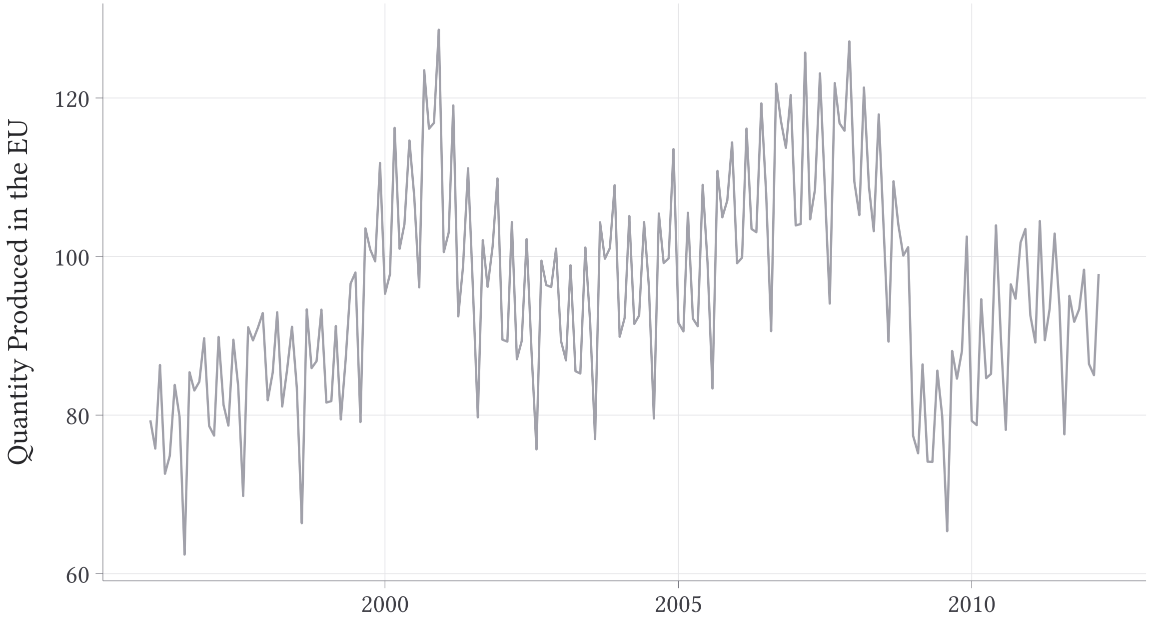
Using y_{t-1} could fail when:

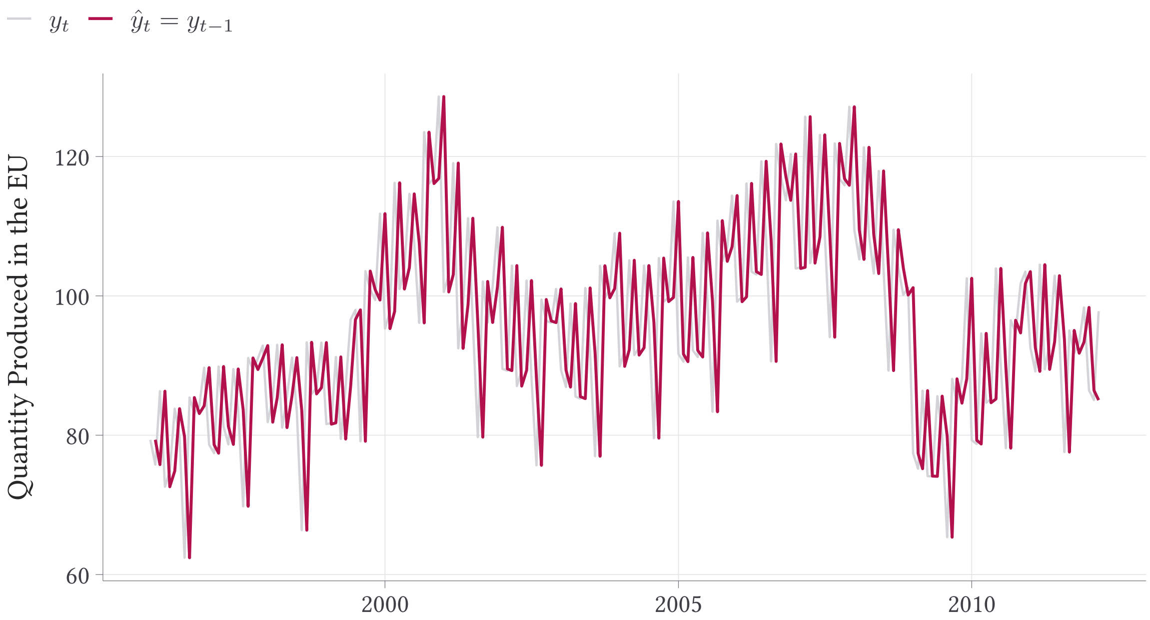
$\rightarrow$ The data has trends or seasonality that y_{t-1} doesn't capture

- Using August's jacket sales to predict September's jacket sales will not do well

$\rightarrow$ y_{t-1} can be quite *noisy*

- Maybe yesterday's value of y was weird because of a bad news story that turned out to not be a big deal


Quantity Produced in the EU
120
100
80
60
2000
2005
2010



y_t
$\hat{y}_t = y_{t-1}$
Quantity Produced in the EU
120
100
80
60
2000
2005
2010

Predicition Error

On the last slide, it's hard to see, but the $\hat{y}_t = y_{t-1}$ does a bad job at predicting y_t

$\rightarrow$ The data jumps around too much, so yesterday's value is only weakly predictive of today's value

We can try to improve on the simple method by smoothing over the last K periods:

$$\hat{y}_t = \sum_{k=1}^{K} w_k y_{t-k},$$

where

- $\rightarrow$ K is the number of lags to smooth over
- $\rightarrow$ w_k is the weights put on the k-th lagged value of y

Smoothing Methods

$$\hat{y}_t = \sum_{k=1}^{K} w_k y_{t-k},$$

For example:

$\rightarrow$ $K = 1$ and $w_1 = 1$ is the simple method $\hat{y}_t = y_{t-1}$

Smoothing Methods

$$\hat{y}_t = \sum_{k=1}^{K} w_k y_{t-k},$$

For example:

→ $K = 1$ and $w_1 = 1$ is the simple method $\hat{y}_t = y_{t-1}$

→ $K = 3$ and $w_3 = \frac{1}{3}$ is the average of three-previous periods

Say we use an average of the K most recent observations:

$$\hat{y}_t = \sum_{k=1}^{K} \frac{1}{K} y_{t-k},$$

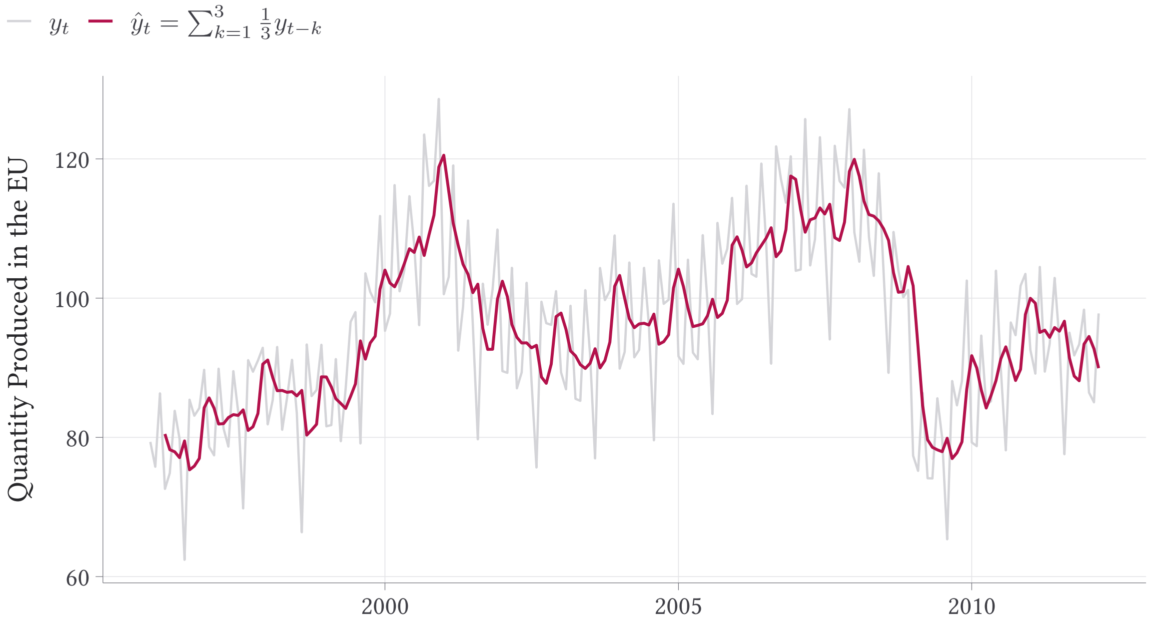
Average of previous ys

Say we use an average of the K most recent observations:

$$\hat{y}_t = \sum_{k=1}^{K} \frac{1}{K} y_{t-k},$$

As you move ahead one time period, you lose 1 observation $(t - K)$ and gain one observation t

$\rightarrow$ The most recent observation y_t updates what we think the moving average is



y_t
$\hat{y}_t = \sum_{k=1}^{3} \frac{1}{3} y_{t-k}$
Quantity Produced in the EU
60
80
100
120
2000
2005
2010

Prediction

Note for the next period y_{t+1}, we can form our out-of-sample forecast as:

$$\ell_{t+1} = \sum_{k=1}^{K} \frac{1}{K} y_{t-k},$$

$\rightarrow$ This is not true of our moving average

Introduction to Time-Series

Learning from Time-Series

Time-series Statistics

Smoothing Methods

- Moving Averages
- Trends and Seasonality

Exponential Smoothing

- Simple Exponential Smoothing
- Trends and Seasonality with SES

Forecasting future values

Say we want to forecast y_{T+1} given information up until time T.

Option 1: Use the last value, $\ell_{T+1} = y_T$

$\rightarrow$ Based on idea that $\mu_{T+1} \approx \mu_T$

$\rightarrow$ But, can be highly sensitive to ε_T

Forecasting future values

Say we want to forecast y_{T+1} given information up until time T.

Option 1: Use the last value, $\ell_{T+1} = y_T$

$\rightarrow$ Based on idea that $\mu_{T+1} \approx \mu_T$

$\rightarrow$ But, can be highly sensitive to ε_T

Option 2: Average of last K periods, $\ell_{T+1} = \frac{1}{K} \sum_{k=0}^{K-1} y_{T-k}$

$\rightarrow$ Less sensitive since averaging over ε_t

$\rightarrow$ Uses potentially less useful information μ_{T-K+1} might be different than μ_{T+1}

Exponential Smoothing

Exponential Smoothing is a blend of these two methods

$\rightarrow$ Weighs over multiple periods to avoid sensitivity on ε_T

$\rightarrow$ Put *higher* weight on *closer* time periods

Simple Exponential Smoothing

The simple exponential smoothing method keeps track of a prediction using ℓ_t

Start out with some initial level ℓ_0 (perhaps set to y_1). Then, we will update ℓ_t in a *recursive manner*:

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

We learn this period's y_t and update our level ℓ_t for the next period

$\rightarrow$ When $y_t > \ell_{t-1}$, we revise up ℓ_t

$\rightarrow$ When $y_t < \ell_{t-1}$, we revise down ℓ_t

How much to update, α

$$\begin{aligned}\ell_t &= \alpha y_t + (1-\alpha)\ell_{t-1} \\ &= \ell_{t-1} + \alpha\left(y_t - \ell_{t-1}\right)\end{aligned}$$

α tells us how much to update

$\rightarrow$ $\alpha = 1$ means throw out old prediction and use y_t

$\rightarrow$ $\alpha = 0$ means do not update at all

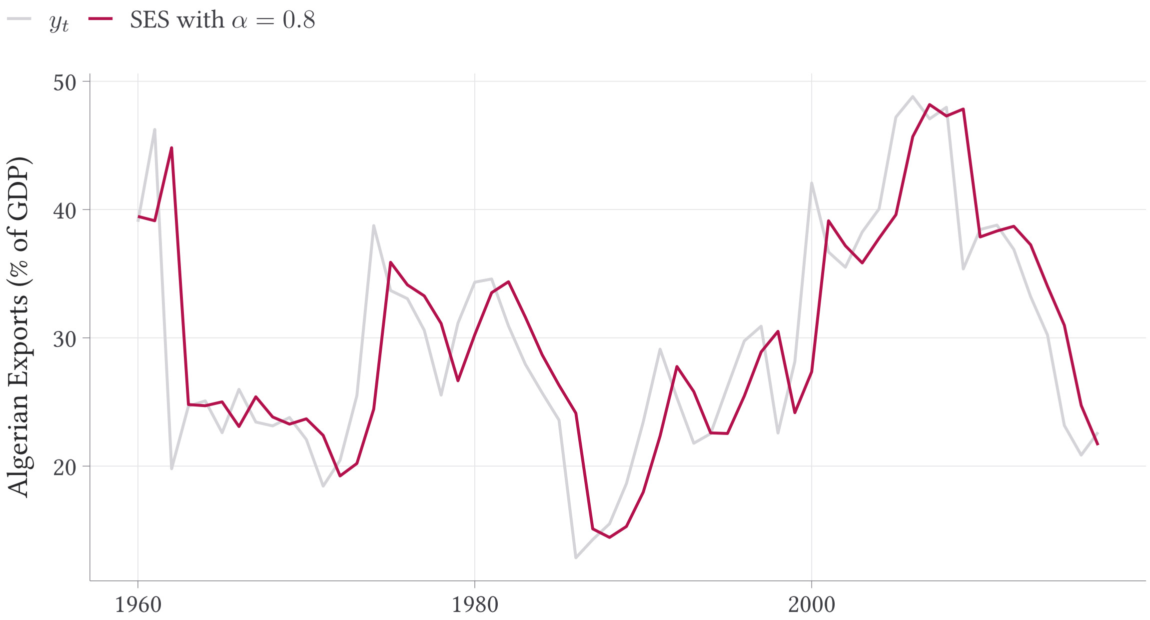
$\rightarrow$ $1 > \alpha > 0$ means updating more (close to 1) or less strongly (close to 0)

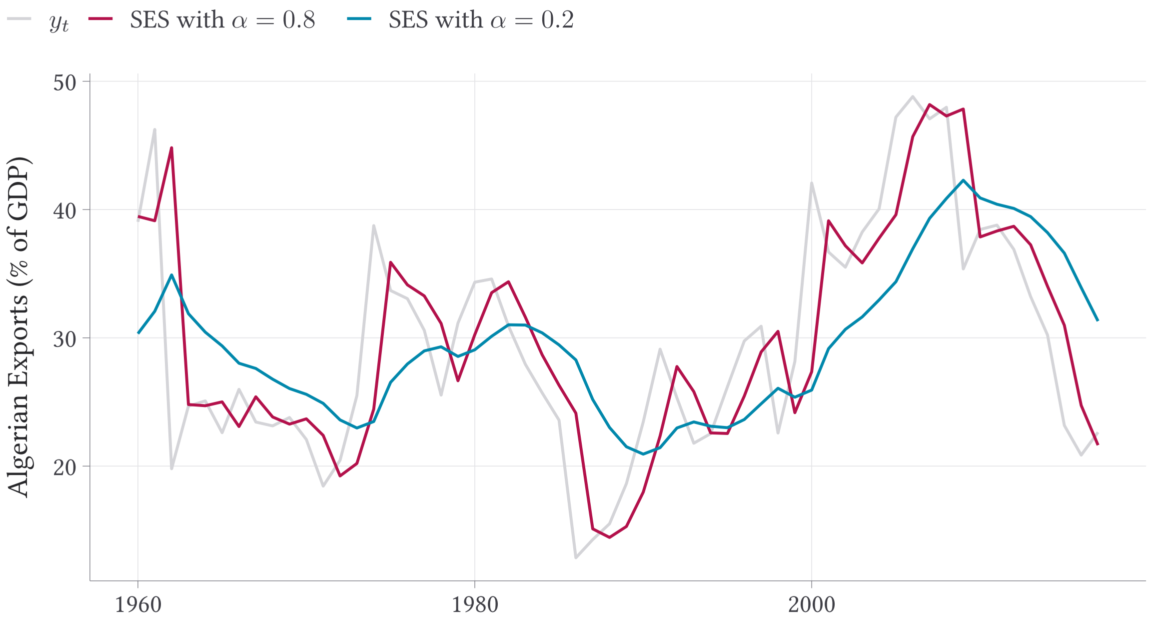
Can think of α as the "memory" of the smoother

Updating parameter α

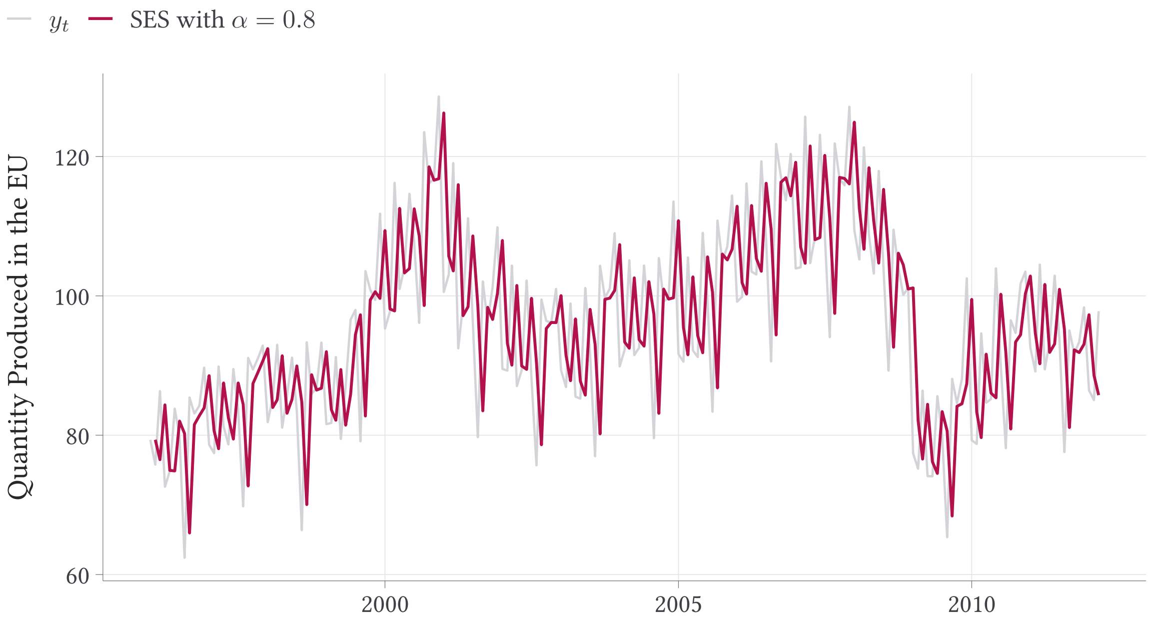
$$\ell_t = \ell_{t-1} + \alpha\left(y_t - \ell_{t-1}\right)$$

Let's look at a couple examples of α to build intuition on how this works

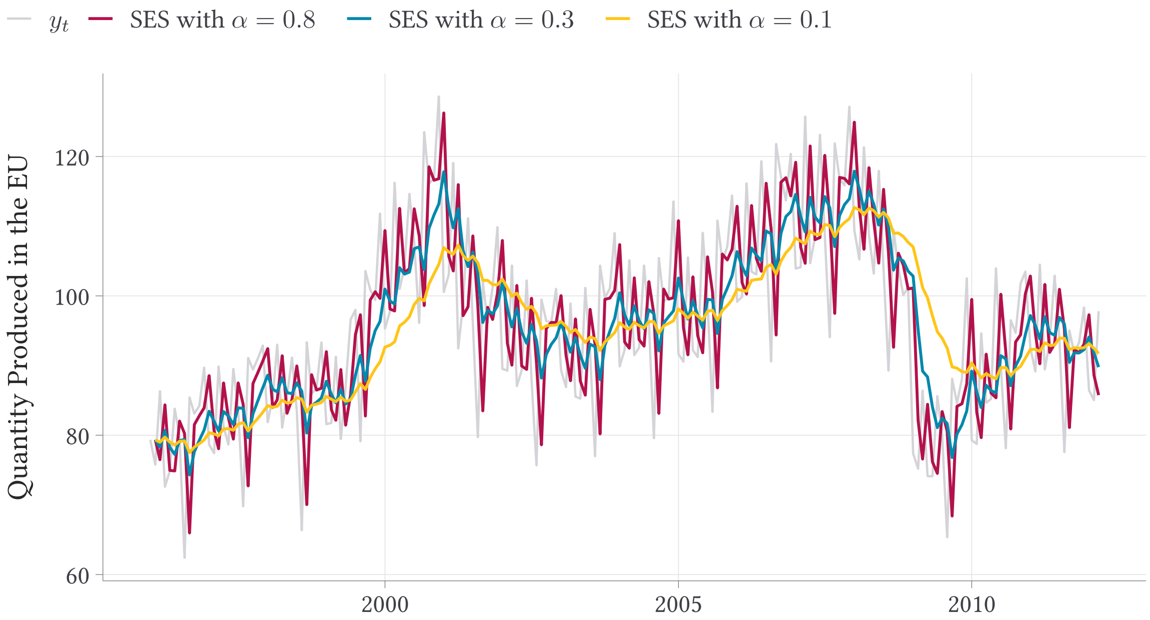

y_t
SES with $\alpha = 0.8$
Algerian Exports (% of GDP)
50
40
30
20
1960
1980
2000



y_t
SES with $\alpha = 0.8$
SES with $\alpha = 0.2$
Algerian Exports (% of GDP)
50
40
30
20
1960
1980
2000



y_t
SES with $\alpha = 0.8$
Quantity Produced in the EU
120
100
80
60
2000
2005
2010



y_t
SES with $\alpha = 0.8$
SES with $\alpha = 0.3$
SES with $\alpha = 0.1$
Quantity Produced in the EU
120
100
80
60
2000
2005
2010

We can trace out how ℓ_t works as follows:

$$\begin{aligned}\ell_t &= \alpha y_t + (1-\alpha)\ell_{t-1} \\ &= \alpha y_t + (1-\alpha)\left(\alpha y_{t-1} + (1-\alpha)\ell_{t-2}\right)\end{aligned}$$

Simple Exponential Smoothing and Recursion

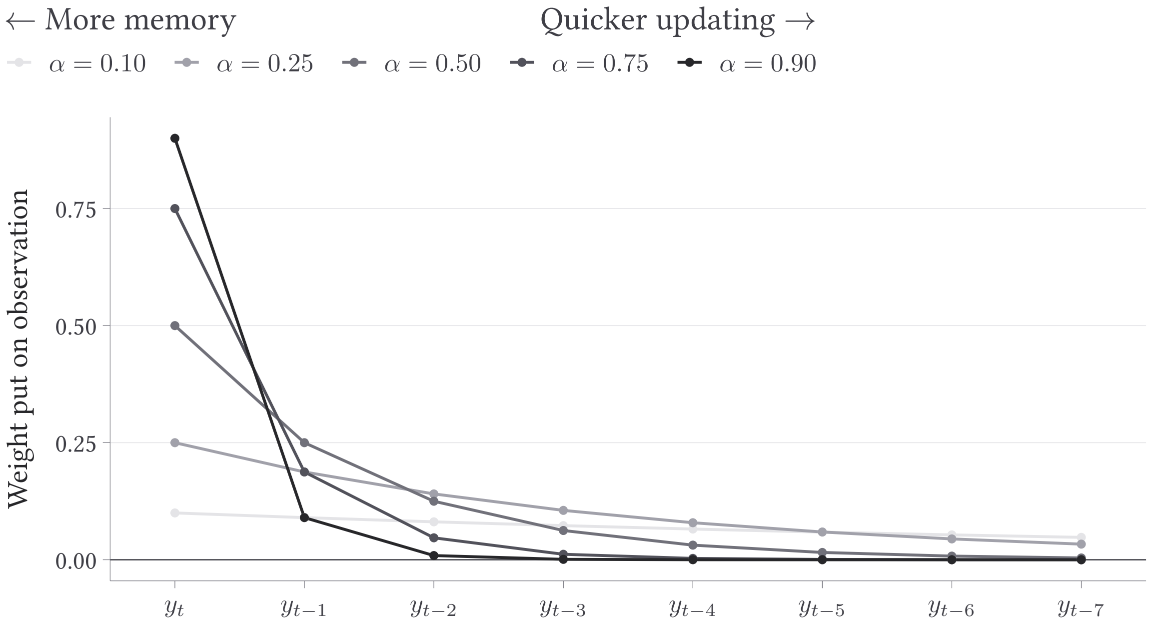
We can trace out how ℓ_t works as follows:

$$
\begin{aligned}
\ell_t &= \alpha y_t + (1-\alpha)\ell_{t-1} \\
&= \alpha y_t + (1-\alpha)\left(\alpha y_{t-1} + (1-\alpha)\ell_{t-2}\right) \\
&= \alpha y_t + \alpha(1-\alpha) y_{t-1} + (1-\alpha)^2 \ell_{t-3}
\end{aligned}
$$

You can repeat this process many times

$$
\begin{aligned}
\ell_{t+1} &= \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2\ell_{t-1} \\
&= \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-1} + (1-\alpha)^3\ell_{t-2} \\
&= \alpha y_t + \alpha(1-\alpha)y_{t-1} + \alpha(1-\alpha)^2 y_{t-1} + \alpha(1-\alpha)^3 y_{t-2} + (1-\alpha)^4\ell_{t-3}
\end{aligned}
$$

Simple Exponential Smoothing is actually taking a weighted average of past y values all the way back to the first-period


← More memory
Quicker updating →
$\alpha = 0.10$
$\alpha = 0.25$
$\alpha = 0.50$
$\alpha = 0.75$
$\alpha = 0.90$
Weight put on observation
0.75
0.50
0.25
0.00
y_t
y_{t-1}
y_{t-2}
y_{t-3}
y_{t-4}
y_{t-5}
y_{t-6}
y_{t-7}

Weights and ‘memory’

From the previous figure, it is clear that different values of α but different weights on long-run values

- $\rightarrow$ A large α is more ‘adaptable’ to recent values of y_t
  - It can respond much more quickly to changes in y
- $\rightarrow$ A small α puts weight more evenly on lags
  - Has more ‘memory’ and is therefore slower to adjust

The different emphasis that these weights put have implications for how the SES method deals with trends and seasonality

Implementing Simple Exponential Smoothing

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

Say we want to estimate ℓ_t using a time-series. How do we get ℓ_{t-1}? Well, we get it using ℓ_{t-2}

→ And to get ℓ_{t-2} we need ℓ_{t-3} …

→ And to get ℓ_{t-3} we need ℓ_{t-4} …

→ and turtles all the way down …

Simple Exponential Smoothing

Starting from period $t = 1$,

$$\ell_T = \alpha y_T + (1-\alpha)\ell_{T-1}$$

$$\ell_{T-1} = \alpha y_{T-1} + (1-\alpha)\ell_{T-2}$$

$$\vdots$$

$$\ell_2 = \alpha y_2 + (1-\alpha)\ell_1$$

$$\ell_1 = \alpha y_1 + (1-\alpha)\ell_0$$

So, we only need the starting point ℓ_0 and we can work 'our way up' from there

Choosing a starting point, ℓ_0

The starting point has very little impact on the forecast in period $T + 1$

$\rightarrow$ For ℓ_T, the weight on ℓ_0 is really small: $\alpha(1 - \alpha)^T$

The simplest solution is to just use $\ell_0 = y_1$

$\rightarrow$ Very little effect on forecasting into the future

SES by hand in R

```
simple_exp_smooth <- function(y, alpha = 0.8) {
  T <- length(y)
  l <- rep(NA, T)
  l_last <- y[1] # l_0

  for (t in 2:length(y)) { # updating l
    l[t] <- alpha * y[t] + (1 - alpha) * l_last
    l_last <- l[t]
  }
  return(l)
}
```

Choosing α (and ℓ_0) optimally

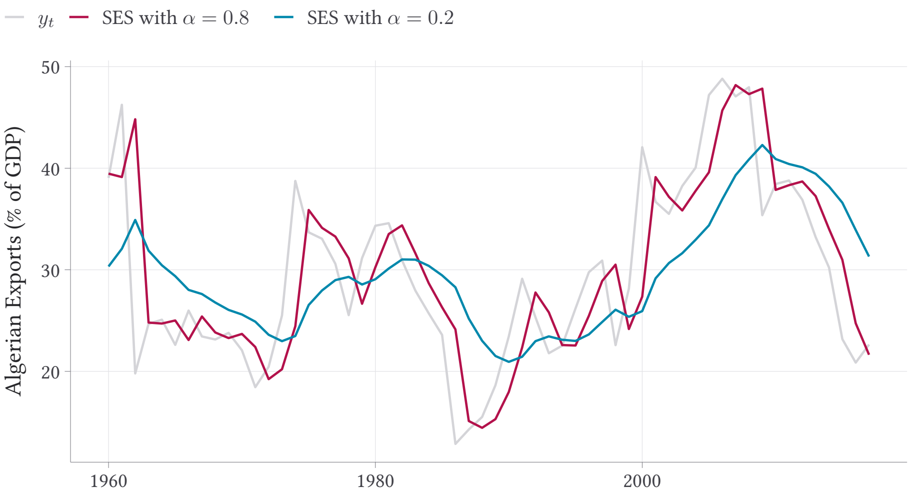
Can estimate the optimal α by minimizing the MSPE

$$\hat{\alpha} = \underset{\alpha \in [0,1]}{\operatorname{argmin}} \sum_{t=1}^{T} (y_t - \ell_t(\alpha))^2$$

where we note that ℓ_t is a function of α

$\rightarrow$ Can also search over values of ℓ_0 and jointly minimize

As an example, let's return to the Algerian exports example:



Algerian Exports Example

α	MSPE	α	MSPE
0.05	4237.193	0.55	2153.324
0.15	3543.732	0.65	2058.575
0.25	2967.187	0.75	2008.838
0.35	2565.885	0.85	1995.456
0.45	2311.215	0.95	2014.927

The MSPE appears to minimized around $\alpha = 0.85$. Indeed the optimal is $\hat{\alpha} = 0.840$

SES in R

The `forecast` package makes this much simpler:

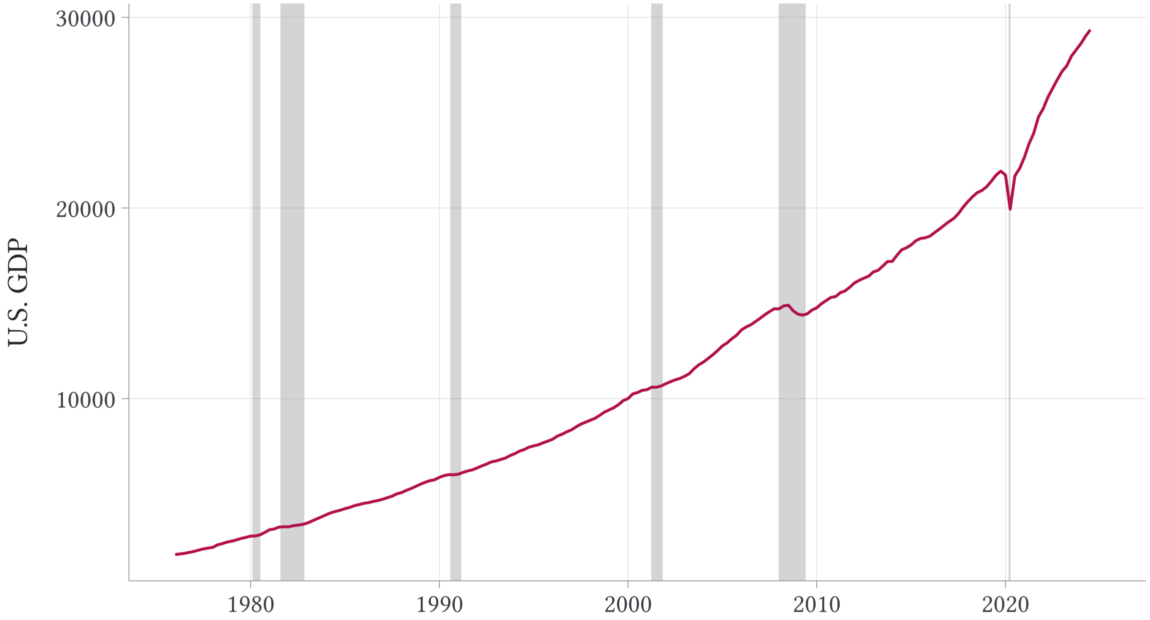
```
library(forecast)
ses(y, alpha = 0.7) # optimally selects l_0
ses(y)              # optimally selects alpha and l_0
```

Trends

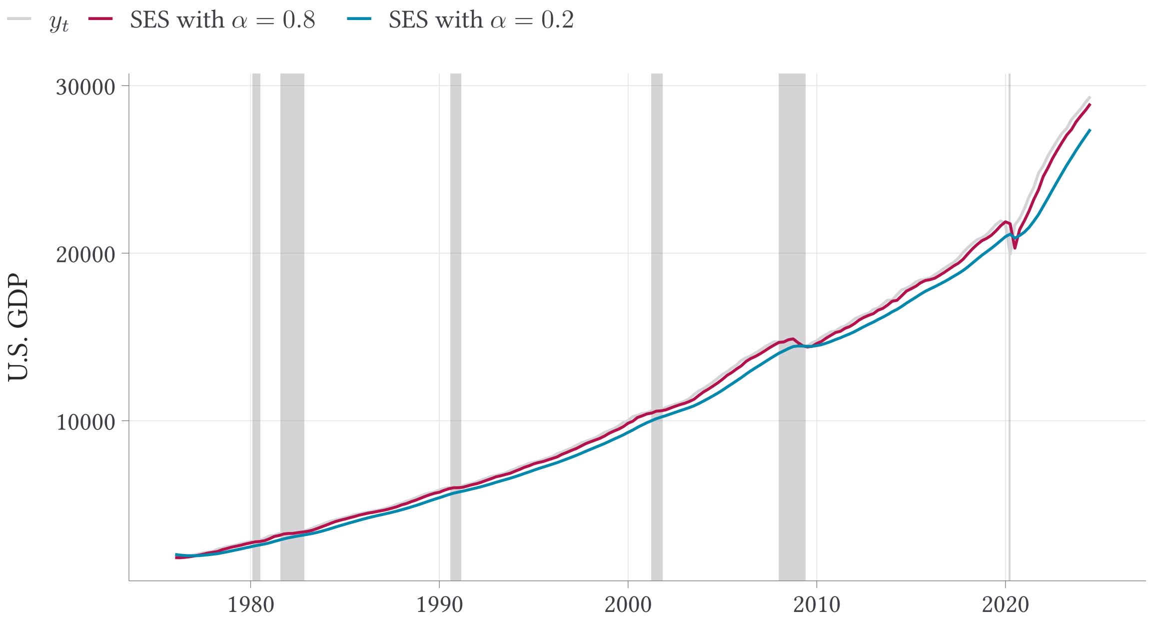
The *simple* exponential smoothing method can fail when the data has long-term trends

→ When α is smaller, we lean more on observations from the past (when the trend was lower)

For example, let's look at smoothing US GDP estimates



30000
20000
10000
U.S. GDP
1980
1990
2000
2010
2020


y_t
SES with $\alpha = 0.8$
SES with $\alpha = 0.2$
U.S. GDP
30000
20000
10000
1980
1990
2000
2010
2020

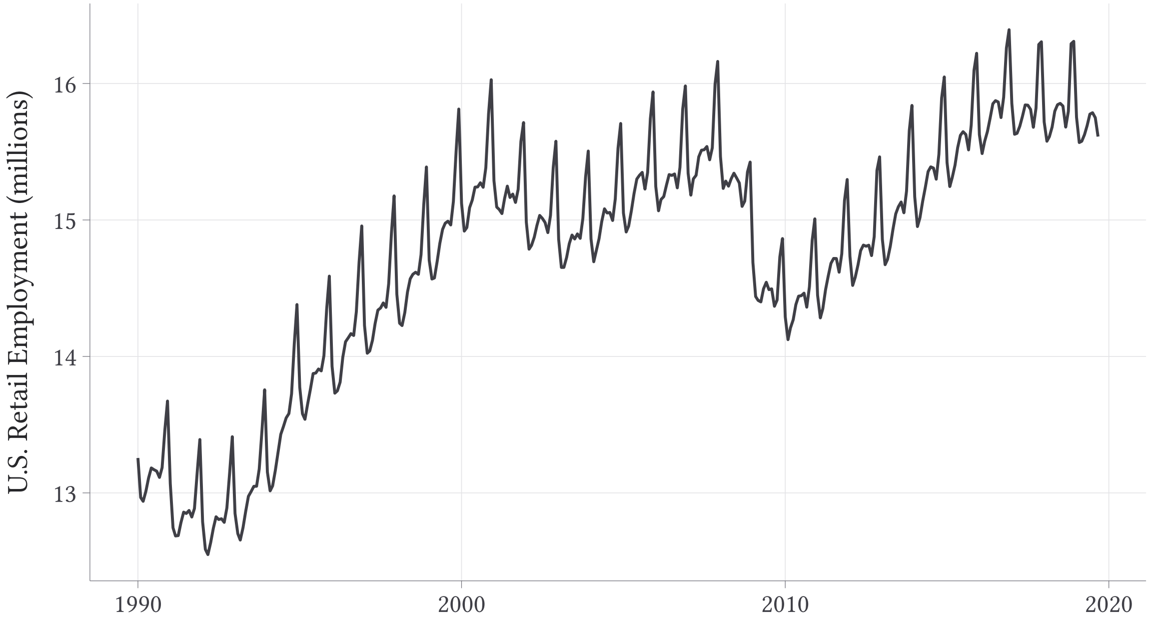
Simple Exponential Smoothing and Trends

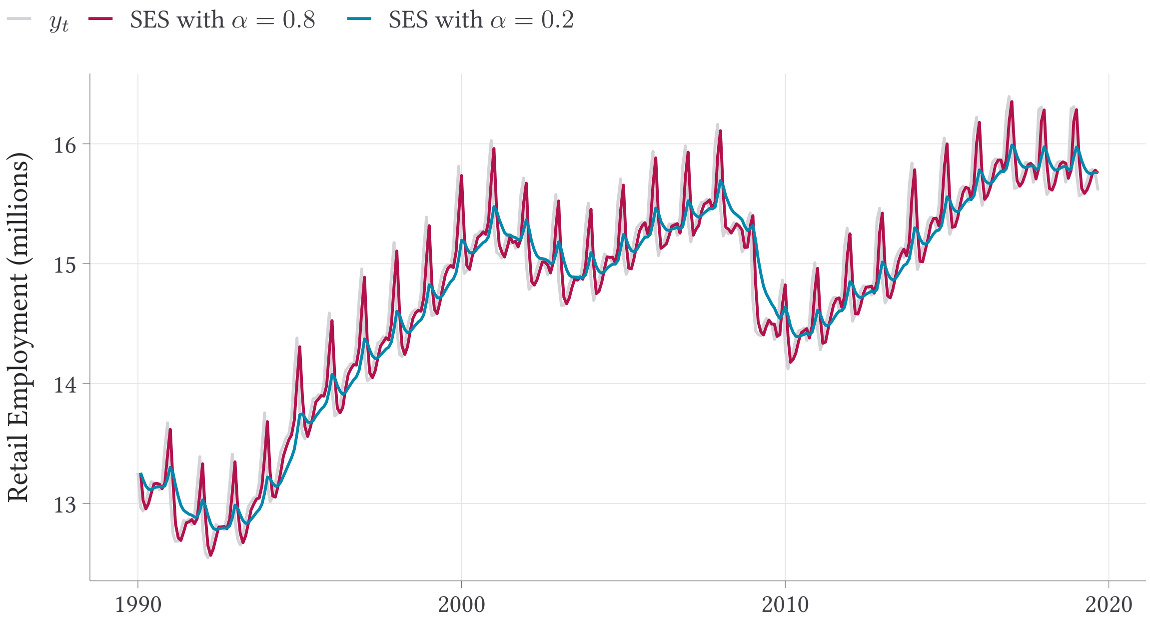
Since GDP is consistently trending upwards, old values of y_{t-k} are systematically lower

$\rightarrow$ Lower values of α will put more weight on older values, hence $\hat{y}_t$ being systematically too low

SES will also miss seasonal trends

$\rightarrow$ E.g. consider retail employment which is systematically higher around the end of the year


U.S. Retail Employment (millions)
16
15
14
13
1990
2000
2010
2020



y_t
SES with $\alpha = 0.8$
SES with $\alpha = 0.2$
Retail Employment (millions)
16
15
14
13
1990
2000
2010
2020

Simple Exponential Smoothing Method

Our forecast was based on

Forecast equation: $\hat{y}_{T+h} = \boxed{\ell_T}$

Level equation: $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1})$

The level equation updates with new observations with weight determined by α

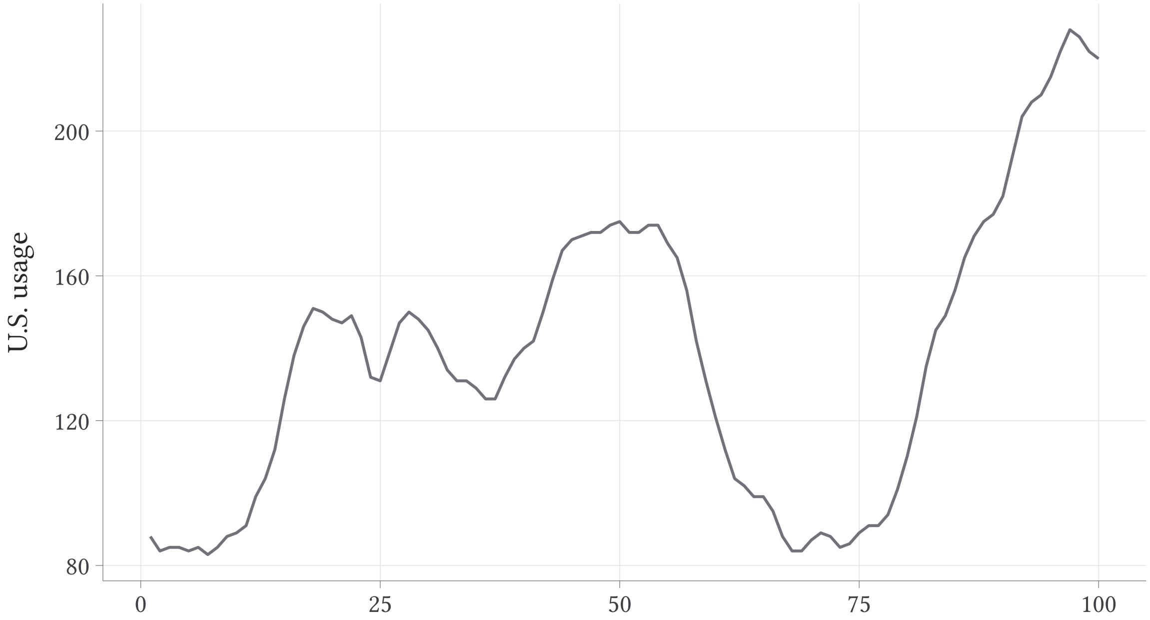
Holt Method for Trends

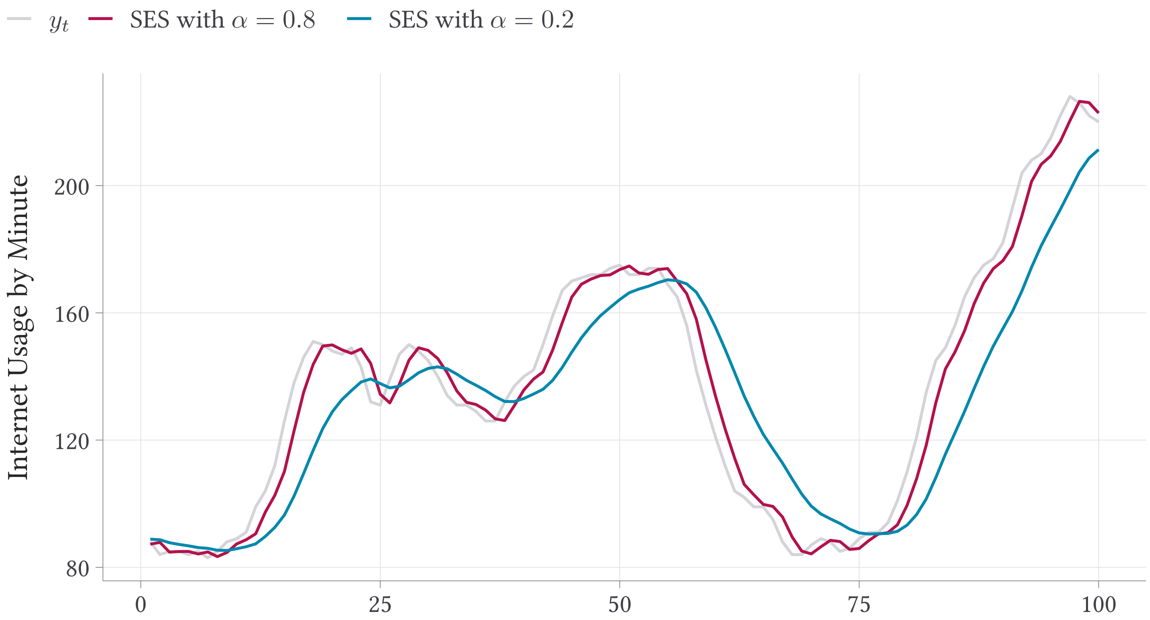
The Holt method adds in a trend term b_t that also will be updating with observations

$$\begin{aligned}
\text{Forecast equation:} \quad & \hat{y}_{T+h} = \ell_T + hb_T \\
\text{Level equation:} \quad & \ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1}) \\
\text{Trend equation:} \quad & b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1},
\end{aligned}$$

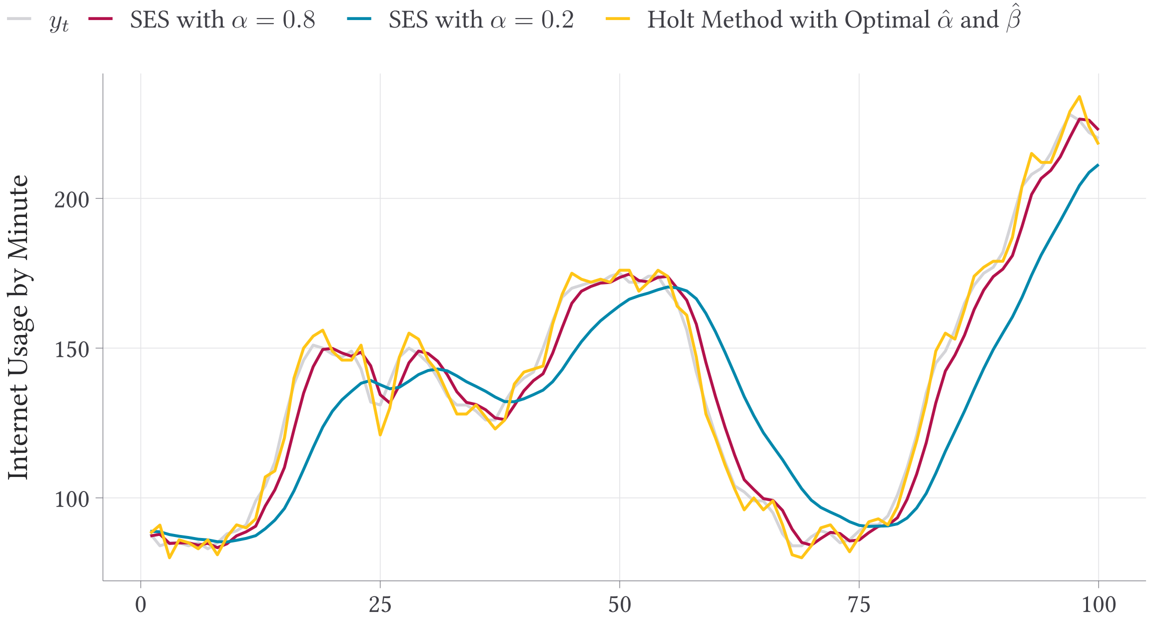
Level equation balances current y_t with what we would forecast it to be $\ell_{t-1} + b_{t-1}$

Trend equation updates between the change in $\ell_t - \ell_{t-1}$ and the old trend b_{t-1}


U.S. usage
200
160
120
80
0
25
50
75
100



y_t
SES with $\alpha = 0.8$
SES with $\alpha = 0.2$
Internet Usage by Minute
80
120
160
200
0
25
50
75
100



y_t
SES with $\alpha = 0.8$
SES with $\alpha = 0.2$
Holt Method with Optimal $\hat{\alpha}$ and $\hat{\beta}$
Internet Usage by Minute
200
150
100
0
25
50
75
100

Holt Method in R

```r
library(forecast)
holt(y, alpha = 0.7, beta = 0.15)
holt(y) # optimally selects alpha and beta
```

Choosing α and β optimally

Similar to before, we can estimate the optimal α and β by minimizing the MSPE over *both* α and β

$\rightarrow$ Will tend to select high α and small β for trending data.

$$\hat{y}_{T+h} = \ell_T + \boxed{hb_T}$$

The Holt method forecasts using a h time period extension of the slope b_T

$\rightarrow$ Often times we expect the trend to go away (can not go up forever)

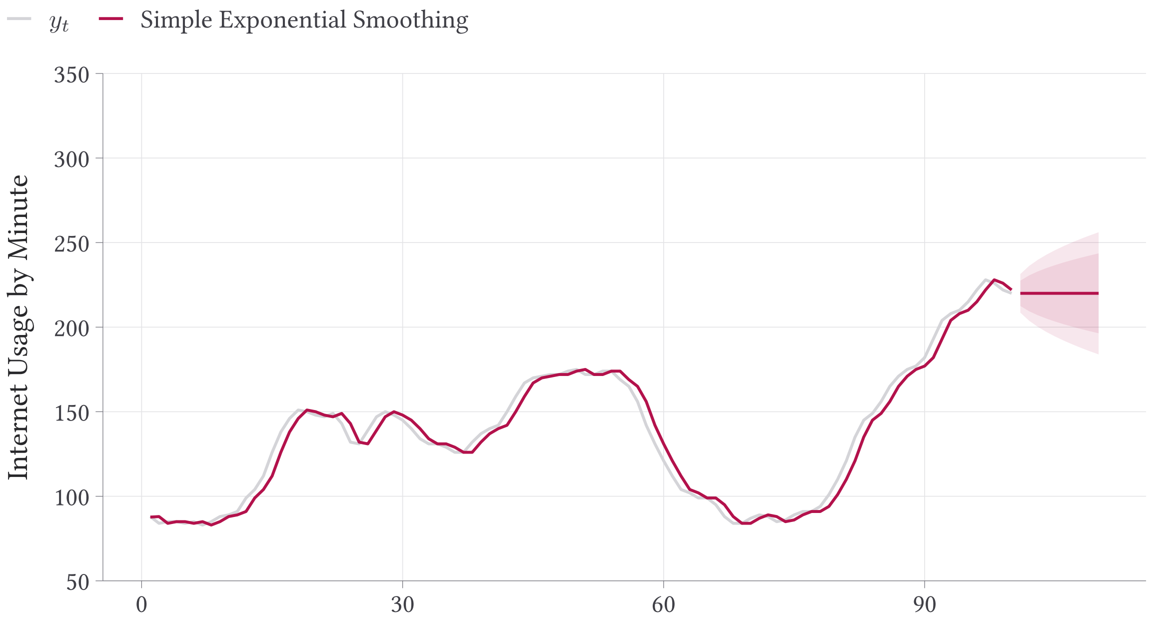
We can add a dampening parameter to our model to address this

Dampening in Holt method

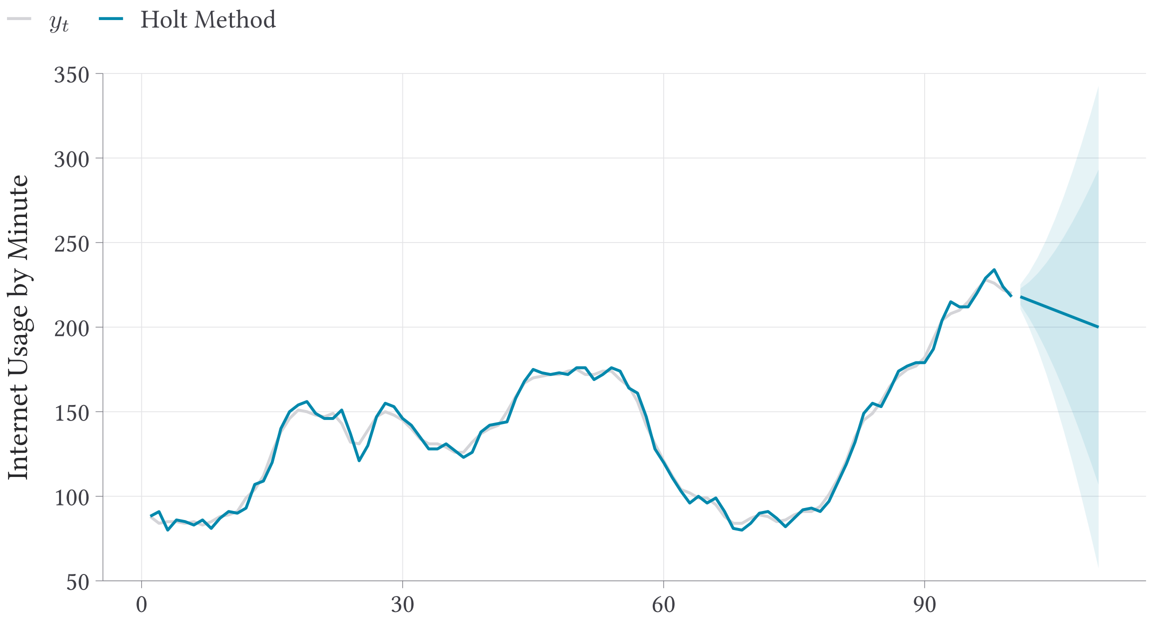
Forecast equation:	$\hat{y}_{T+h} = \ell_T + (\phi + \phi^2 + \cdots + \phi^h) b_T$
Level equation:	$\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + \phi b_{t-1})$
Trend equation:	$b_t = \beta(\ell_t - \ell_{t-1}) + (1 - \beta)\phi b_{t-1}$

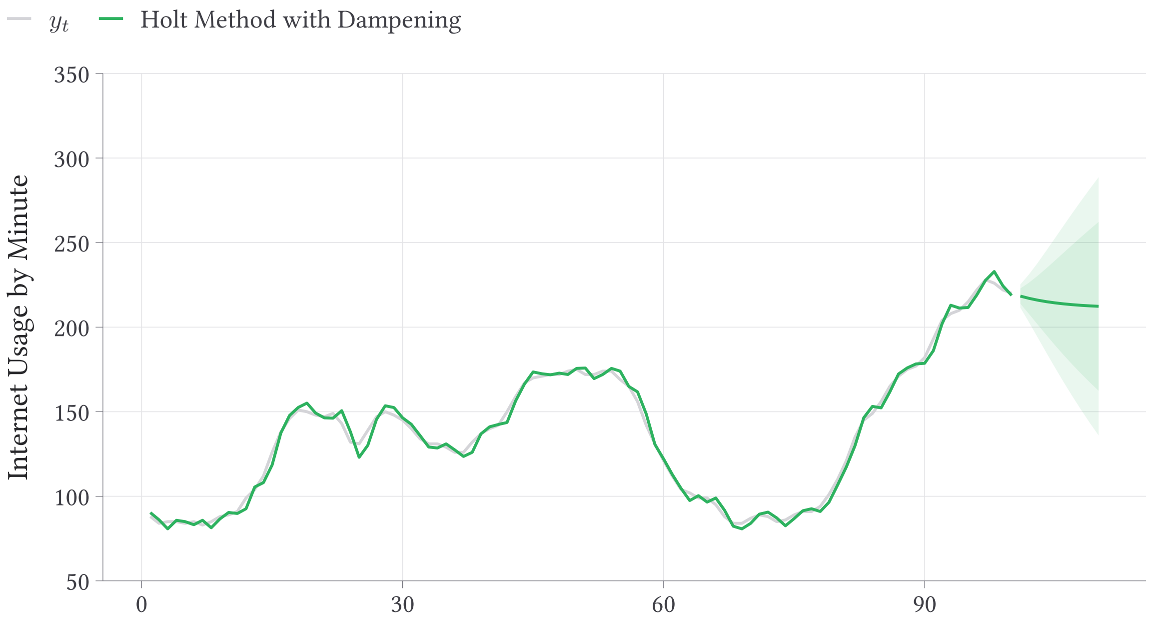
Same idea, but now ϕ ‘dampens’ the trend b_t

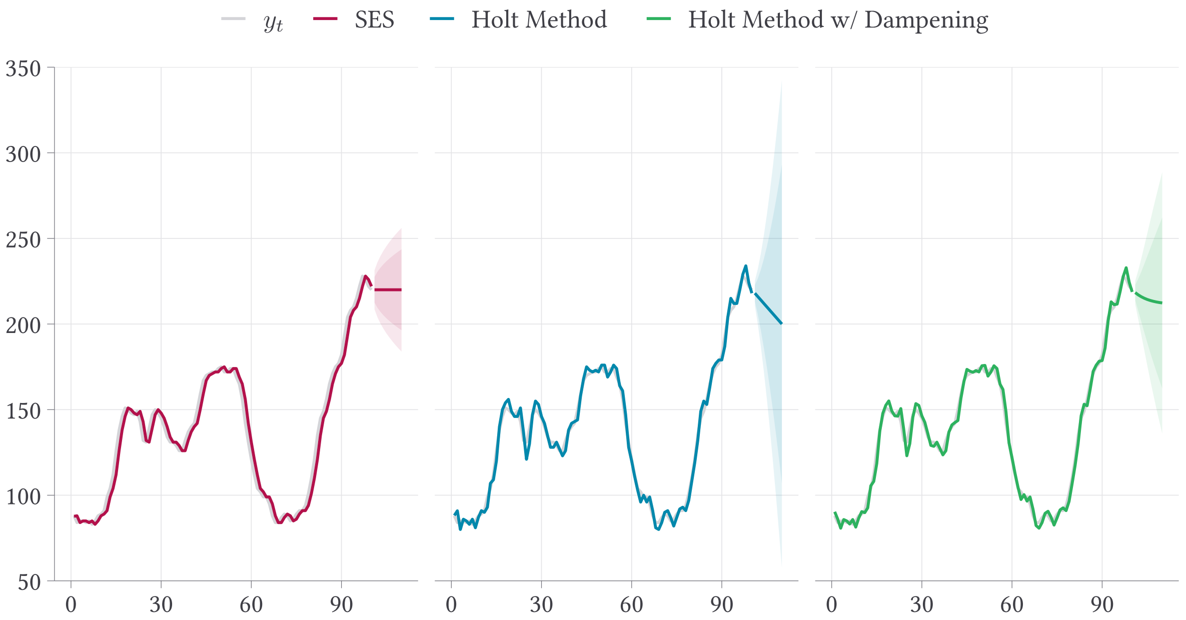
$\rightarrow$ Add `damped = TRUE` to `holt` call to add a dampening parameter to the model



y_t
Simple Exponential Smoothing
Internet Usage by Minute
350
300
250
200
150
100
50
0
30
60
90


y_t
Holt Method
Internet Usage by Minute
350
300
250
200
150
100
50
0
30
60
90


y_t
Holt Method with Dampening
Internet Usage by Minute
350
300
250
200
150
100
50
0
30
60
90


y_t
SES
Holt Method
Holt Method w/ Dampening
350
300
250
200
150
100
50
0
30
60
90

Holt-Winters Method

I will save you the details (FPP3 Section 8.3), but you can also adapt the Holt method to add seasonality to the pattern

Use function `hw` just like `ses` and `holt`

$\rightarrow$ Also includes an option for `damped`