Topic 5: Time Series, Regression, Moving Averages, and Smoothing Methods

ECON 4753 — University of Arkansas

Prof. Kyle Butts

November 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

- You observe a set of observations x_t for $t \in \{1, \ldots, T\}$
- In general, we call t the 'period'

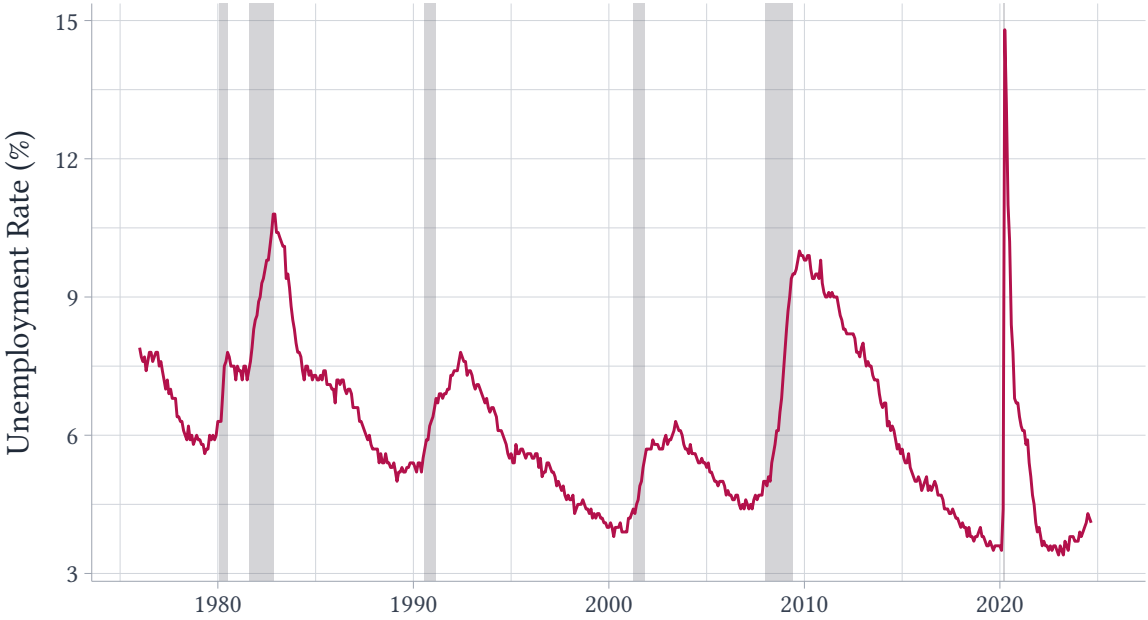
Time-series

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

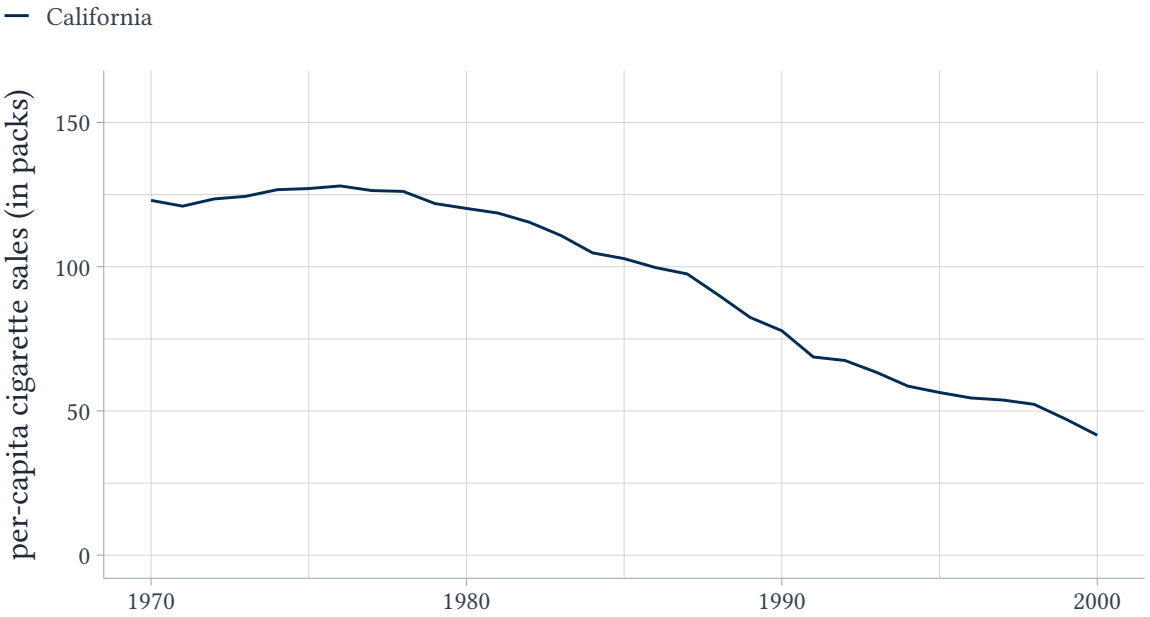
- You observe a set of observations x_t for $t \in \{1, \ldots, T\}$
- In general, we call t the 'period'

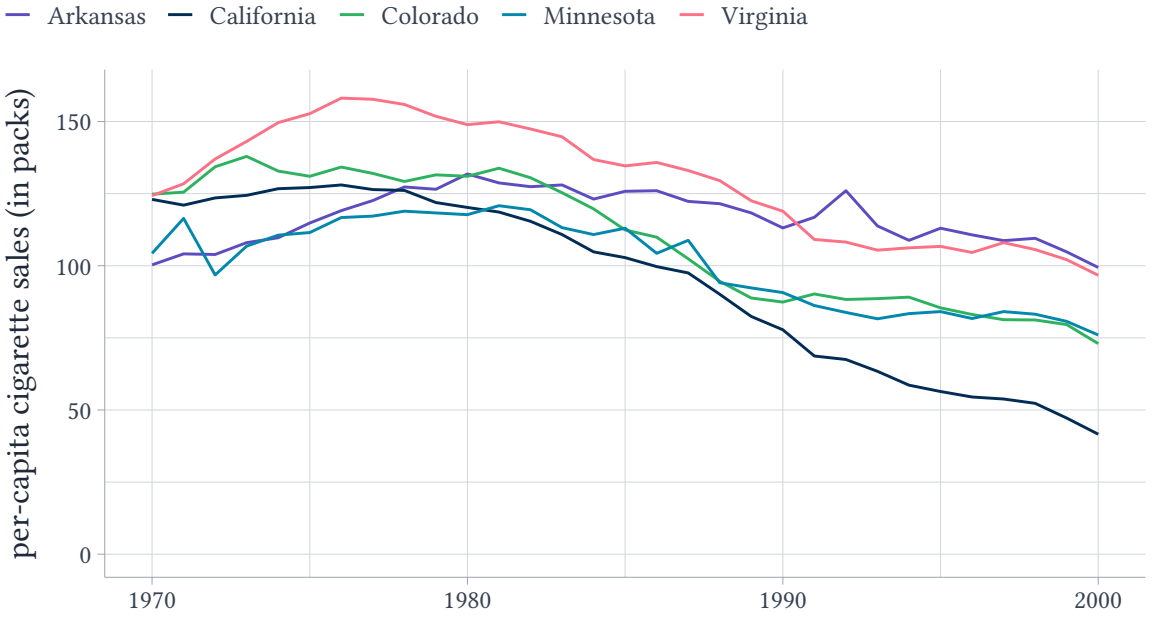
Examples include:

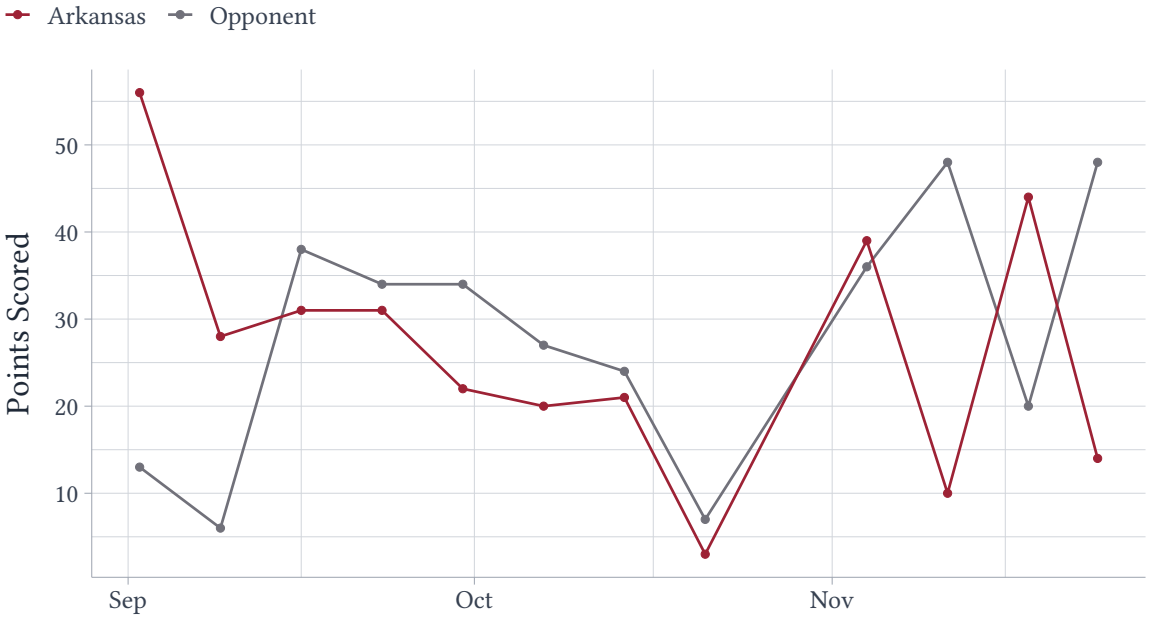
- Annual data on the DGP of a country
- Hourly stock price for a company
- Annual data on cigarette consumption per capita in a state
- A sport's teams number of points scored in games (unequally spaced)


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


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


California
per-capita cigarette sales (in packs)
150
100
50
0
1970
1980
1990
2000


Arkansas
California
Colorado
Minnesota
Virginia
per-capita cigarette sales (in packs)
150
100
50
0
1970
1980
1990
2000


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

What is special about time-series?

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

- That is, we view a set of individuals viewed at a point in time

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.

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

- This property is essential in forecasting; following a variable over time might let us predict futrue values

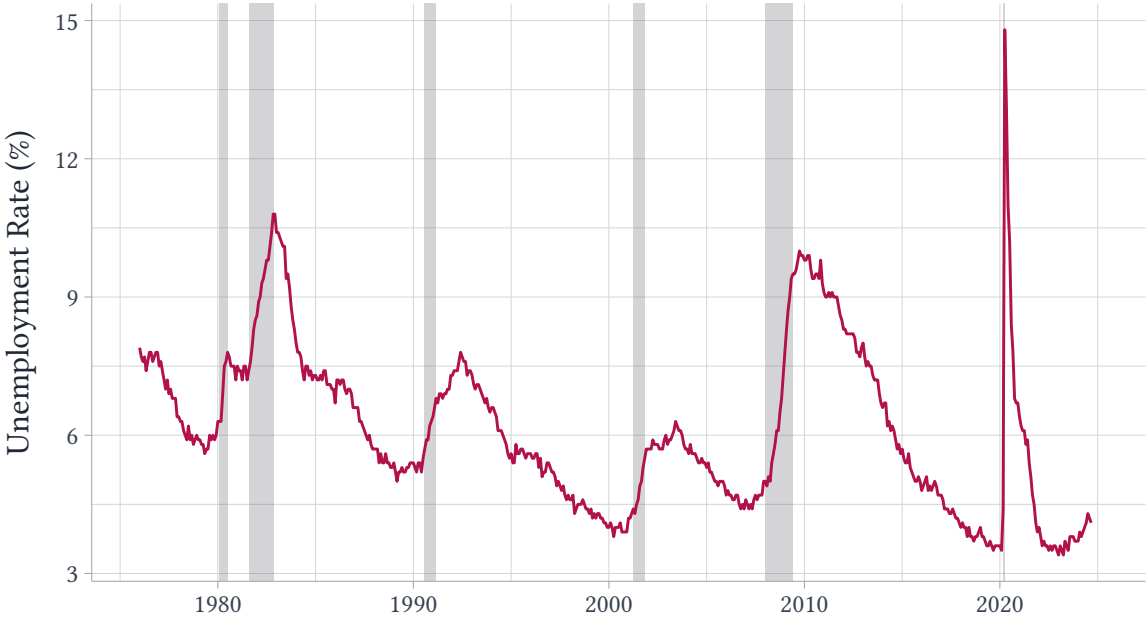
What is special about time-series?

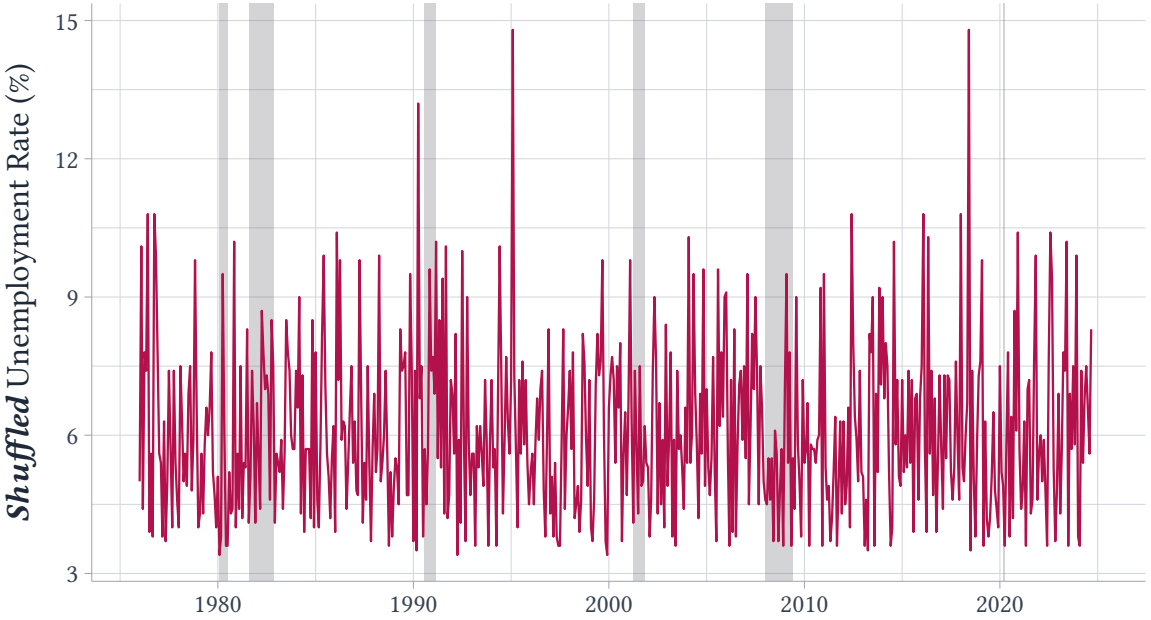
In time-series data, knowing last period's value of y_{t-1} is often very useful for this period's value of y_t

- This property is essential in forecasting; following a variable over time might let us predict futrue values

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

- 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

What we can gain from using time-series

Time-series forecasting can be useful to:

- Predict future values based on past data
- Inform decision-making by anticipating changes over time
- 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 evolves over time while ignoring random fluctuations (inference)
   - $\rightarrow$ E.g. learn about seasonality, trends, etc.
2. Predict future values of y_t (forecasting)
   - $\rightarrow$ 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

$$y_t = \mu_t + \varepsilon_t$$

- μ_t is the 'typical' or 'systematic' value of y at time t
- 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

- Without any more structure, this is is an impossible task

Learning from time-series

$$y_t = \mu_t + \varepsilon_t$$

All hope is not lost though. Looking at the previous time-series graphs, it is clear we can learn *something*

$$y_t = \mu_t + \varepsilon_t$$

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
   - → How does μ_t change over time?
   - → E.g. trend in GDP over time
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

Evaluating forecasting methods

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

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

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

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

- 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:

$$\gamma_k = \operatorname{cov}(y_t, y_{t-k}) = \mathbb{E}((y_t - \mu)(y_{t-k} - \mu))$$

where:

- μ is the mean of y_t,
- $\operatorname{cov}(y_t, y_{t-k})$ is the covariance between y_t and y_{t-k}.

Autocovariance

$$\gamma_k = \text{cov}\left(y_t, y_{t-k}\right) = \mathbb{E}((y_t - \mu)(y_{t-k} - \mu))$$

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

- 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.

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

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

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

Autocovariance

$$\gamma_k = \operatorname{cov}(y_t, y_{t-k}) = \mathbb{E}((y_t - \mu)(y_{t-k} - \mu))$$

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

$$\gamma_0 = \operatorname{cov}(y_t, y_t) = \operatorname{var}(y_t)$$

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:

$$\rho_k = \frac{\gamma_k}{\text{var}\left(y_t\right)} = \frac{\text{cov}\left(y_t, y_{t-k}\right)}{\text{var}\left(y_t\right)}$$

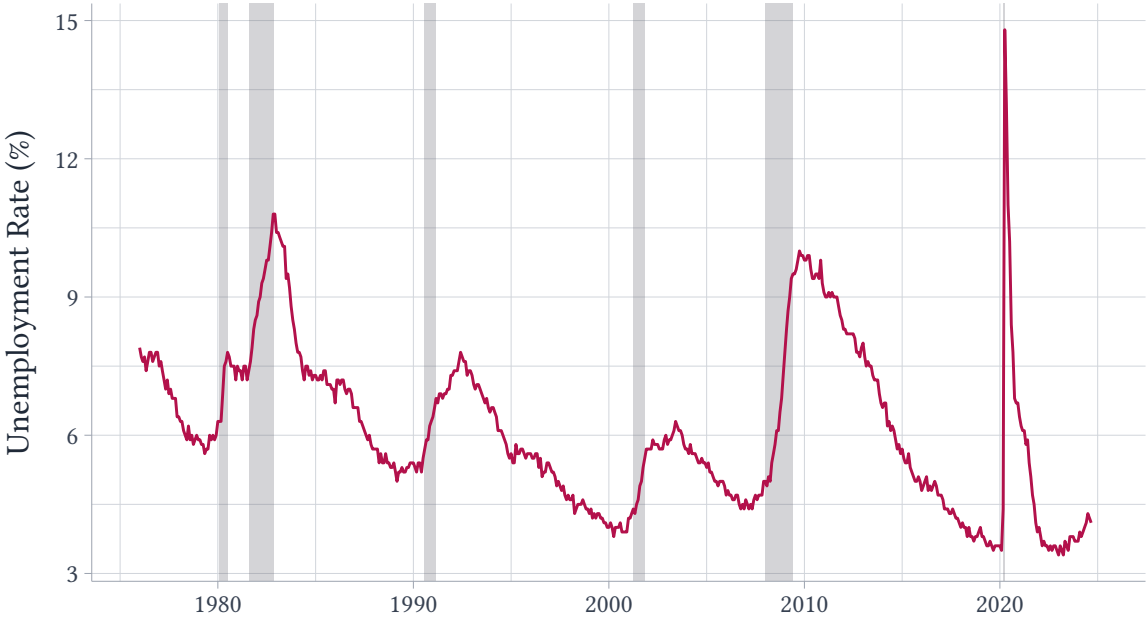
where:

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

Autocorrelation

$$\rho_k = \frac{\gamma_k}{\text{var}\left(y_t\right)} = \frac{\text{cov}\left(y_t, y_{t-k}\right)}{\text{var}\left(y_t\right)}$$

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


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

Unemployment Example

In the unemployment example, the time-series

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

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

Unemployment Example

In the unemployment example, the time-series

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

- 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 Example

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

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

- Shocks to last year's unemployment seem to 'persist' into the current period

Unemployment Example

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

Unemployment 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!

Unemployment 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!

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

Local Methods

The previous topic introduced smoothing methods for inference and prediction in time-series. The central idea to smoothing methods was to use 'local' information:

- To form a forecast $\hat{y}_t$, use observations "close" to t

We studied the advantages and disadvantages of these methods:

- Pro: these methods do good at picking up on sudden changes to μ_t (if the smoothing was not too extreme)
- But, they did a bad job at learning about seasonality and trends
  - $\rightarrow$ Holts-Winter method can help with this

Time-Series Regression Methods

In this topic, we will use our trusted friend *regression* to conduct inference on time-series data

Advantages of regression:

- Regression uses all the observations to fit the model, so can better learn trends and seasonality
- Regression can include other predictors (e.g. predicting GDP given unemployment)

The main disadvantage is that compared to smoothing methods (later), regression does a less-good job at predicting short-term changes

Time-series Regression Set-up

All that we have learned in cross-sectional regressions will apply in the case of time-series.

- The 'unit of observation' is a time-period t
- The outcome variable is the variable measured at time t, y_t
- we will have a set of explanatory variables for each time-period

Example, we will will discuss regressing y_t on indicators for day of the week (Sunday through Saturday)

- $\implies$ regression coefficients will estimate the average y for each day of the week

Regression refresher

For a set of explanatory variables, X_t, we predict y_t using the model

$$\hat{y}_t = X_t\beta$$

The coefficient β is estimated by minimizing the mean-squared prediction error

$$\frac{1}{T}\sum_{t=1}^{T}(y_t - X_t\beta)^2$$

- The only difference from cross-sectional regression is in the choice of X_t (e.g. day-of-week indicators)

Missing data and Regression

One distinct advantage about regression is that we do not require ‘complete’ time-series data

- E.g. if we are missing data for some months, we can still estimate our model.
  - $\rightarrow$ Smoothing methods have a difficult time with missing data

Inference in Time-series Regression

Inference is more complicated with time-series regressions

- Shocks to one time-period u_t can show up and impact other time-periods u_s
- In principle, all observations are relatted with one-another

Inference in Time-series Regression

Inference is more complicated with time-series regressions

- Shocks to one time-period u_t can show up and impact other time-periods u_s
- In principle, all observations are relatted with one-another

The idea of 'repeated sampling' is a bit odd in this context too:

- Our sample is one realization of the time-series
- Resampling is like 'rewinding' and getting a new history for $t = 1, \ldots, T$

Time-series Predictors

This section will introduce a set of useful predictors that can be used (in-combination) to perform inference on time-series data

Estimating seasonal trends

The first predictor we will discuss is estimation of seasonal, or repeated, patterns:

- Quarterly patterns
- Monthly patterns
- Day-of-week patterns
- Time-of-day patterns

These methods will rely on using a set of indicator variables

Estimating seasonal trends

The easiest way to estimate these patterns is to use the `lubridate` package to get the relevant variable:

- Quarterly: `quarter(date)`
- Monthly: `month(date, label = TRUE)`
- Day-of-week: `wday(date, label = TRUE)`
- Time-of-day: `hour(date)`

Then, you can create indicator variables using `i()` in your call to `feols` (from the `fixest` package)

Estimating seasonal trends

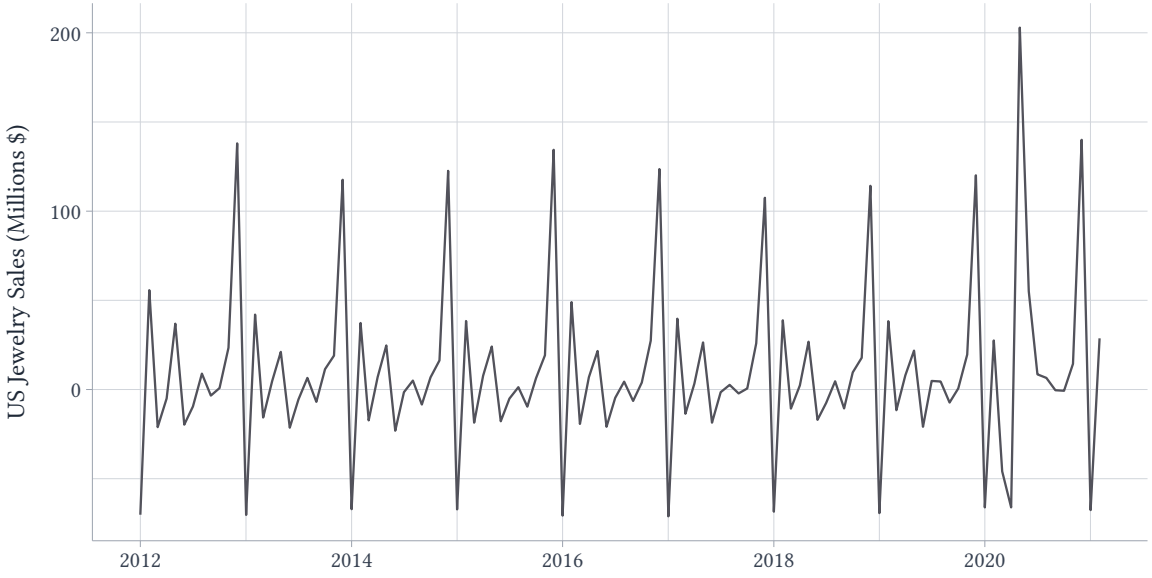
For example, consider the estimating a monthly pattern. The regression will be given by

$$y_t = \alpha + \sum_{m=1}^{12} \mathbb{1}[\texttt{Month}(t) = m]\beta_m + u_t$$

Remember that since we have an intercept, one of the indicator variables will drop out.

$\hat{\beta}_m$ represent the difference in average y_t for month m relative to the omitted month

- E.g. if we do not include January, then $\hat{\beta}_2$ is the difference between Februaries' mean y relative to January's

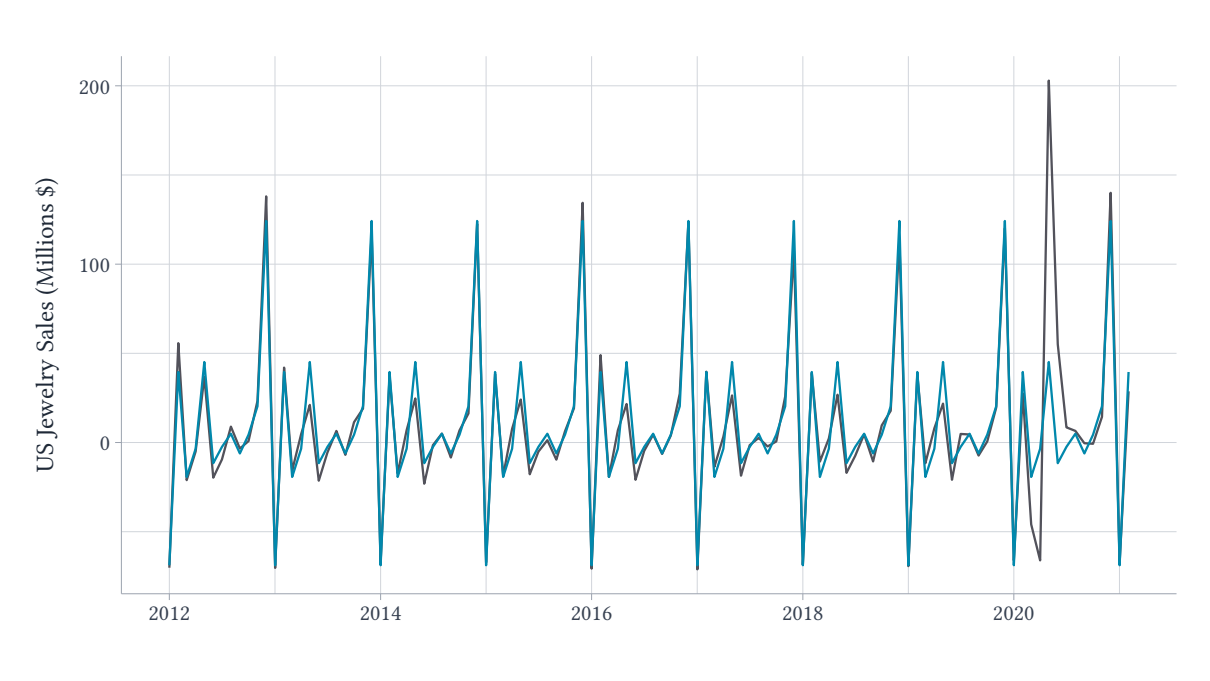

US Jewelry Sales (Millions $)
200
100
0
2012
2014
2016
2018
2020

Predicting jewerly sales monthly pattern

```
df$month = month(df$date, label = TRUE)
est_monthly <- feols(
  sales ~ i(month), data = df, vcov = "hc1"
)

# predict y using the model
sales_hat <- predict(est_monthly)
```

	Estimate	Std. Error	**t** value	Pr(>|**t**|)	
(Intercept)	-68.8100	0.556885	-123.56221	< 2.2e-16	***
month::Feb	108.3500	2.711906	39.95346	< 2.2e-16	***
month::Mar	49.4989	3.563492	13.89056	< 2.2e-16	***
month::Apr	65.3322	7.953131	8.21465	8.8182e-13	***
month::May	113.9544	19.769283	5.76422	9.5184e-08	***
month::Jun	57.2211	8.369328	6.83700	6.9588e-10	***
month::Jul	66.3544	2.017959	32.88196	< 2.2e-16	***
month::Aug	73.7322	0.928813	79.38332	< 2.2e-16	***
month::Sep	62.6767	1.275046	49.15639	< 2.2e-16	***
month::Oct	73.1767	1.561164	46.87316	< 2.2e-16	***
month::Nov	89.1433	1.553690	57.37526	< 2.2e-16	***
month::Dec	193.0322	3.738664	51.63134	< 2.2e-16	***


US Jewelry Sales (Millions $)
200
100
0
2012
2014
2016
2018
2020

Significance tests

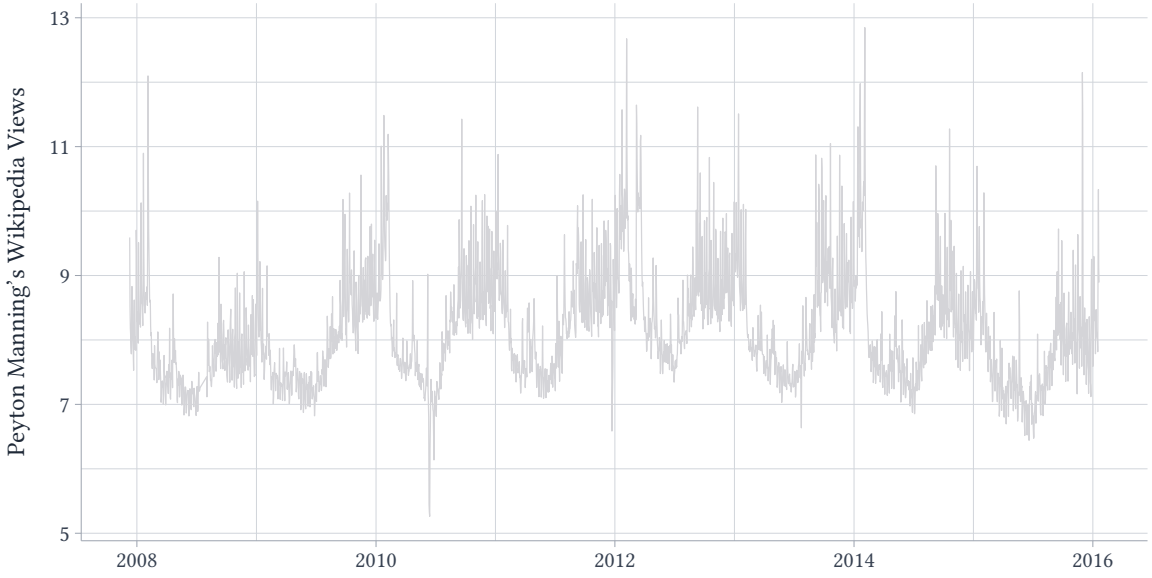
$\hat{\beta}_m$ represent the difference in average y_t for month m relative to the omitted month

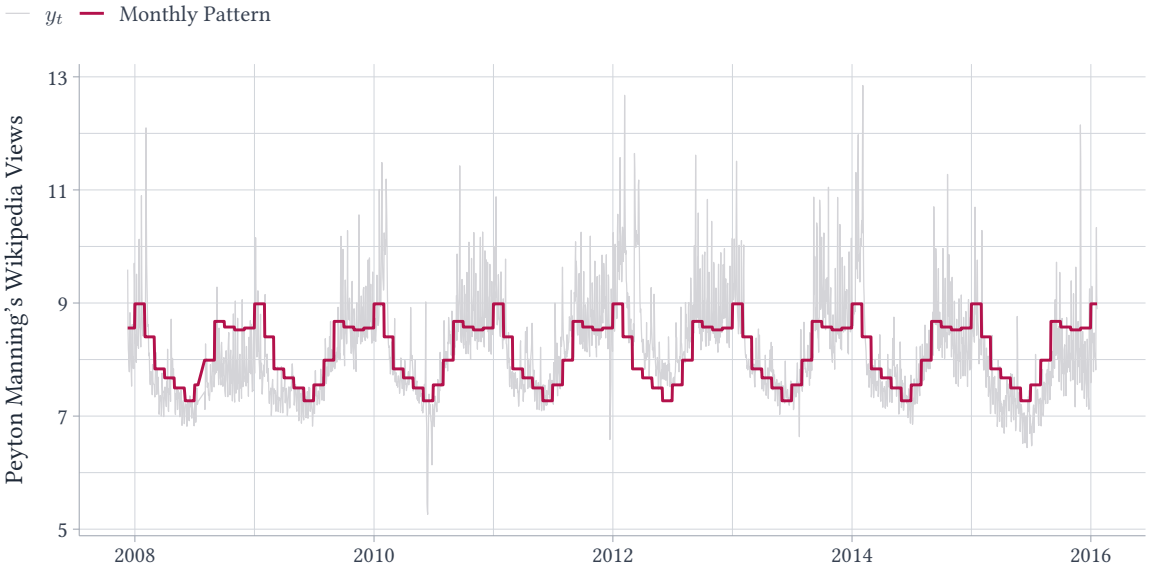
We can use a t-test to compare if the month's average y_t is significantly different than the omitted month:

- Test the null that $\hat{\beta}_m = 0$

Another example

The following slides will present another example using daily data on the number of views on Peyton Manning's wikipedia page


Peyton Manning's Wikipedia Views
13
11
9
7
5
2008
2010
2012
2014
2016


y_t
Monthly Pattern
Peyton Manning's Wikipedia Views
13
11
9
7
5
2008
2010
2012
2014
2016

Practice Questions

On the following slide, I will present the results from our regression with monthly indicators. Answer the following questions (take a picture):

- What is the omitted category?
- In which month are the wikipedia views the highest?
- Are views significantly lower in February than January?

```
OLS estimation, Dep. Var.: views
             Estimate Std. Error   t value   Pr(>|t|)
(Intercept)  8.985181   0.051221 175.42034  < 2.2e-16 ***
month::Feb  -0.583683   0.084752  -6.88696 6.9644e-12 ***
month::Mar  -1.150467   0.073183 -15.72035  < 2.2e-16 ***
month::Apr  -1.309271   0.057717 -22.68421  < 2.2e-16 ***
month::May  -1.487304   0.056482 -26.33250  < 2.2e-16 ***
month::Jun  -1.714787   0.057479 -29.83353  < 2.2e-16 ***
month::Jul  -1.433526   0.060080 -23.86035  < 2.2e-16 ***
month::Aug  -0.996372   0.058421 -17.05493  < 2.2e-16 ***
month::Sep  -0.310229   0.070998  -4.36952 1.2890e-05 ***
month::Oct  -0.409981   0.067193  -6.10155 1.1904e-09 ***
month::Nov  -0.461025   0.066963  -6.88482 7.0678e-12 ***
month::Dec  -0.427606   0.065228  -6.55556 6.5398e-11 ***
```

Estimating time-trends

While estimating seasonatity / recurring patterns is relatively simple, the trends of a time-series is much more difficult and flexible

- The general path of the time-series can evolve in many different ways

In general, we should look at the time-series data to inform us about how we want to model the trends.

- The point of this section is to show you some methods you can use and when each might work well

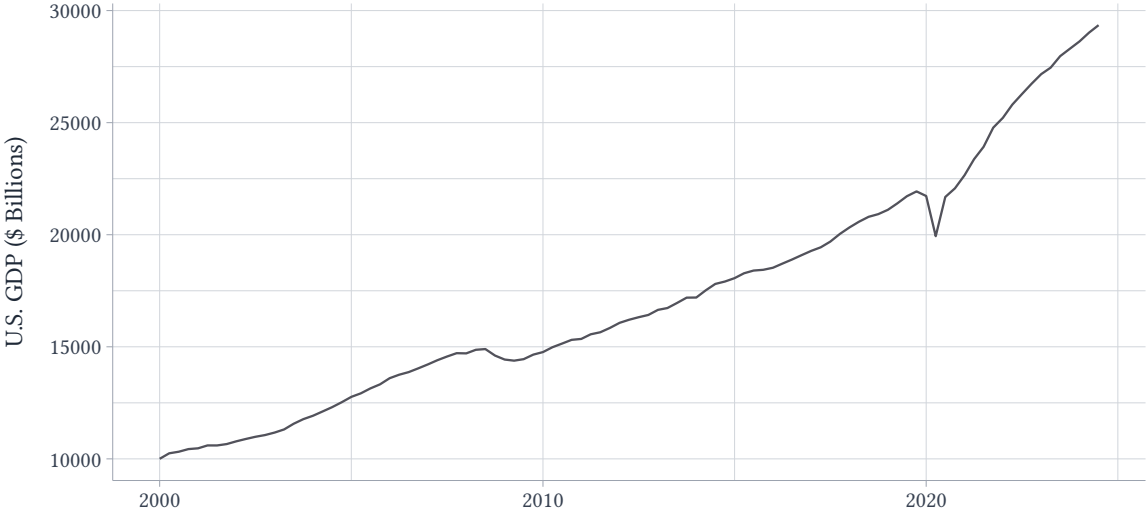
Linear time-trends

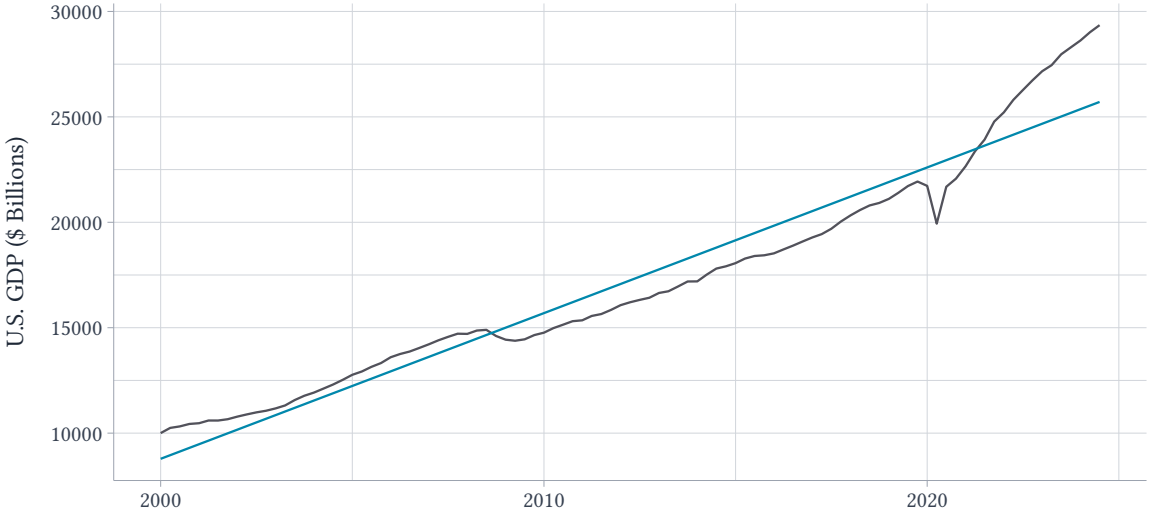
The simplest model for trends is a linear time-trend:

$$y_t = \alpha + \lambda t + u_t$$

Linear time-trends imply that the outcome variable y_t grows (on average) by λ for every-period

This is a quite restrictive model, but is often times sufficient


U.S. GDP ($ Billions)
30000
25000
20000
15000
10000
2000
2010
2020


U.S. GDP ($ Billions)
30000
25000
20000
15000
10000
2000
2010
2020

Estimating linear time trends in R

$$y_t = \alpha + \lambda t + u_t$$

To run this regression, we need to generate the t variable

- A column that contains $1, 2, 3, \ldots, T$

Two notes:

- Note if we have missing observations, that is okay just skip those numbers
- t does not need to start at 1; it will just change the $\hat{\alpha}$

Estimating linear time trends in R

Remember that a `Date` object in R actually is just a number (he number of days since "1970-01-01")

- This means internally consecutive days look like $t, t+1, t+2, \ldots$ like we need!
- Or, monthly data is spaced by 28/30/31 days

So for a linear time-trend, you can just use 'date' as a continuous variable

Estimating linear time trends in R

```
feols(gdp ~ date, data = df, vcov = "hc1")
```

```
OLS estimation, Dep. Var.: gdp
Observations: 99
Standard-errors: Heteroskedasticity-robust
                Estimate Std. Error  t value  Pr(>|t|)
(Intercept) -11945.78793 898.176248 -13.3000 < 2.2e-16 ***
date             1.89173   0.063438  29.8201 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Interpreting linear time trend

```
                Estimate Std. Error  t value  Pr(>|t|)
(Intercept) -11945.78793 898.176248 -13.3000 < 2.2e-16 ***
date             1.89173   0.063438  29.8201 < 2.2e-16 ***
```

Every 1 day, U.S. GDP is predicted to grow by 1.89 billion $

- Every quarter, $91 * 1.89 = 171.99$ billion $
- Every year, $365 * 1.89 = 689.85$ billion $

Forecasting with linear time trend

To forecast into the future, you just extend the time-trend $T+1, T+2, \ldots$

The forecasted value becomes

$$\hat{y}_{T+k} = \hat{\alpha} + \hat{\lambda}(T+k)$$

Forecasting with linear time trend

To forecast into the future, you just extend the time-trend $T+1, T+2, \ldots$

The forecasted value becomes

$$\hat{y}_{T+k} = \hat{\alpha} + \hat{\lambda}(T+k)$$

! Note of Caution: Always *be careful extending time-trends* too far out

- E.g. say you predicted a slightly higher $\hat{\lambda}$ than the true growth rate, extending that out 15 periods means you will estimate $\hat{y}_{T+15}$ to be way too high
- More, time-trends tend to plateau
  $\rightarrow$ e.g. virality tends to die out

Linear Time-trends Failure

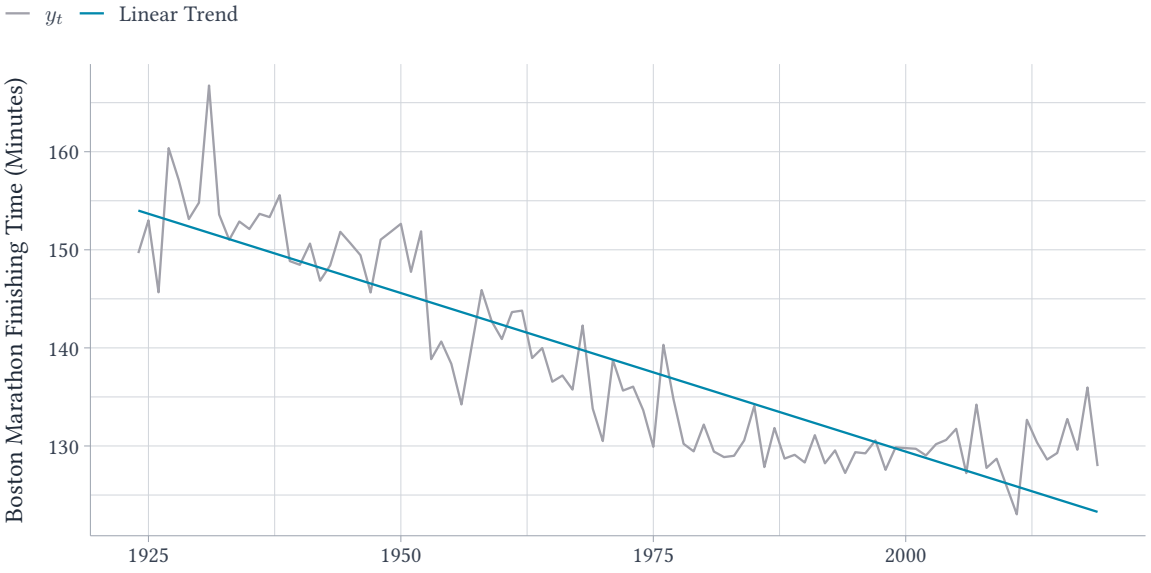
While time-trends do a great job at summarizing succinctly a trend in y_t, it often can be too crude

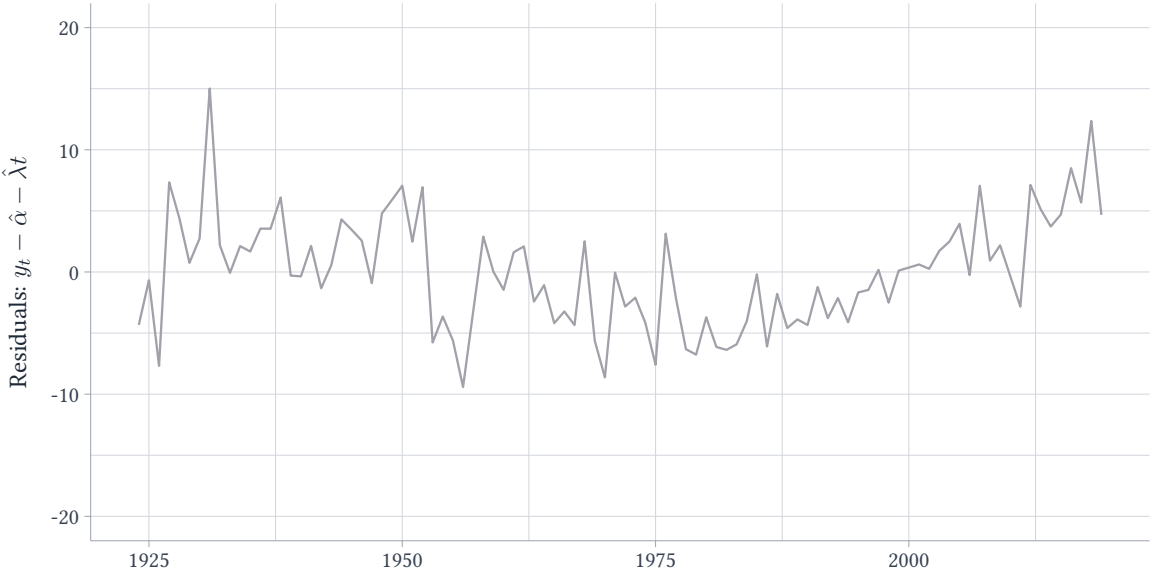
- Perhaps the trend changes over time (e.g. improvements slow down)
- Especially at risk when the time-series is long

One way to check for this is to look at the difference between $y_t - \hat{y}_t$

- Visually inspect if the residual has any remaining trends

Let's look at the example of the time for the winner of the Boston Marathon


y_t
Linear Trend
Boston Marathon Finishing Time (Minutes)
160
150
140
130
1925
1950
1975
2000


Residuals: $y_t - \hat{\alpha} - \hat{\lambda}t$
20
10
0
-10
-20
1925
1950
1975
2000

Quadratic trends

Given our discussion in cross-sectional regression, you might be tempted to model more flexible 'trends' via higher-order polynomial terms

$$y_t = \alpha + \lambda_1 t + + \lambda_2 t^2 u_t$$

× Do not do this! When you forecast into the future, the higher order polynomials can shoot off very quickly!

Changing trends

One way to improve our model without introducing too much complexity is to break up the series into different ‘epoch’ (i.e. eras / moments).

- Estimate a separate trend for each epoch

This is called a piecewise linear trends

In our marathon times example:

- An initial epoch of small changes in time,
- Followed by a steep decline in times,
- and then a final epoch where things leveled off

Changing trends

More formally, let B_ℓ be the breakpoints for each epoch. Then, we can write our model as

$$y_t = \alpha + \lambda_1 t + \sum_{\ell=2}^{L} \mathbb{1}[t \geq B_\ell] * (t - B_\ell) * \lambda_\ell + u_t$$

- At each point $B_2, \ldots, B_L$, the slope changes
- $\hat{\lambda}_1$ is the slope of the first epoch
- $\hat{\lambda}_2, \ldots, \hat{\lambda}_L$ are the difference in slopes from the preivous epoch

For example, to get the slope in the third-epoch we do $\hat{\lambda}_1 + \hat{\lambda}_2 + \hat{\lambda}_3$

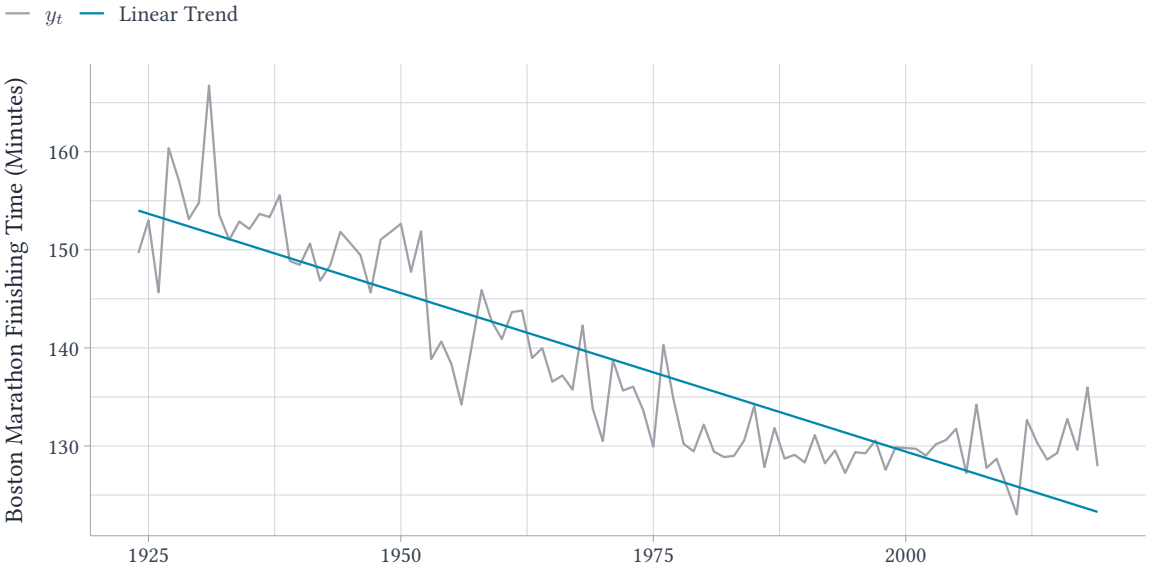
title

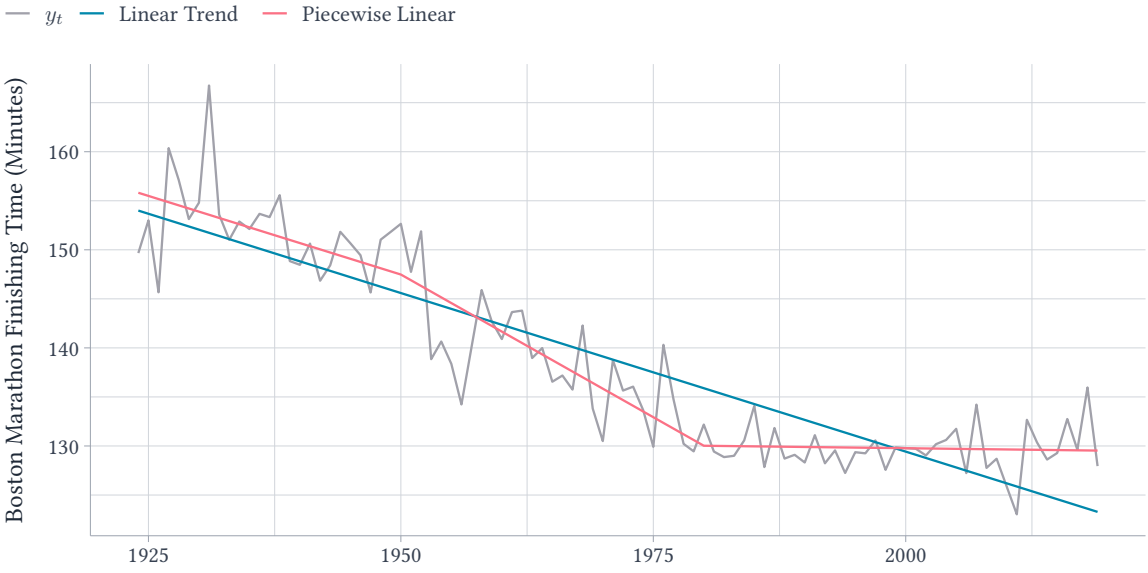
In R, you can define epoch like:

```
df$trend_1 <- df$year
df$trend_2 <- (df$year - 1950) * (df$year > 1950)
df$trend_3 <- (df$year - 1980) * (df$year > 1980)
```

Then you estimate the piece-wise time trend model with

```
feols(y ~ trend_1 + trend_2 + trend_3, data = df)
```


y_t
Linear Trend
Boston Marathon Finishing Time (Minutes)
160
150
140
130
1925
1950
1975
2000


y_t
Linear Trend
Piecewise Linear
Boston Marathon Finishing Time (Minutes)
160
150
140
130
1925
1950
1975
2000

Forecasting with changing trends

$$y_t = \alpha + \lambda_1 t + \sum_{\ell=2}^{L} \mathbb{1}[t \geq B_\ell] * (t - B_\ell) * \lambda_\ell + u_t$$

For forecasting, note that your prediction will be based on the slope of the final epoch

- This is because $T + k \geq B_L$, so all the λ_ℓ are 'active'

Choosing breakpoints

We selected the breakpoints, more or less, visually

- 1950 and 1980 looked to be when breaks happen

Choosing breakpoints

We selected the breakpoints, more or less, visually

- 1950 and 1980 looked to be when breaks happen

Alternatively, we could try and 'detect' break-points

- The goal is to select B_ℓ to do the best job at predicting y_t
- Perhaps even select the number of breaks L

Choosing breakpoints

We selected the breakpoints, more or less, visually

- 1950 and 1980 looked to be when breaks happen

Alternatively, we could try and 'detect' break-points

- The goal is to select B_ℓ to do the best job at predicting y_t
- Perhaps even select the number of breaks L

The rough idea would be to select $B_1, \ldots, B_L$ to minimize mean-squared prediction error

- But, we must be careful not to overfit the data! Can either 'regularize' or try and hold out a random portion of the time-series to evaluate out-of-sample MSPE

Flexibly modelling trends

The linear time-trend λt is useful for:

- Summarizing the time-series trend succinctly
- Useful for forecasting into the future
- But can be over-simplifying of a model

If inference is the goal, we can more flexibly model trends using more fine-grained indicator variables (e.g. month $\times$ year)

- This limits the ability to extrapolate into the future because we can not estimate the coefficient for years outside our sample

Monthly patterns vs. Year-by-month

What we have seen so far is how to estimate a recurring monthly pattern

- Each year has the same predicted value in a given month

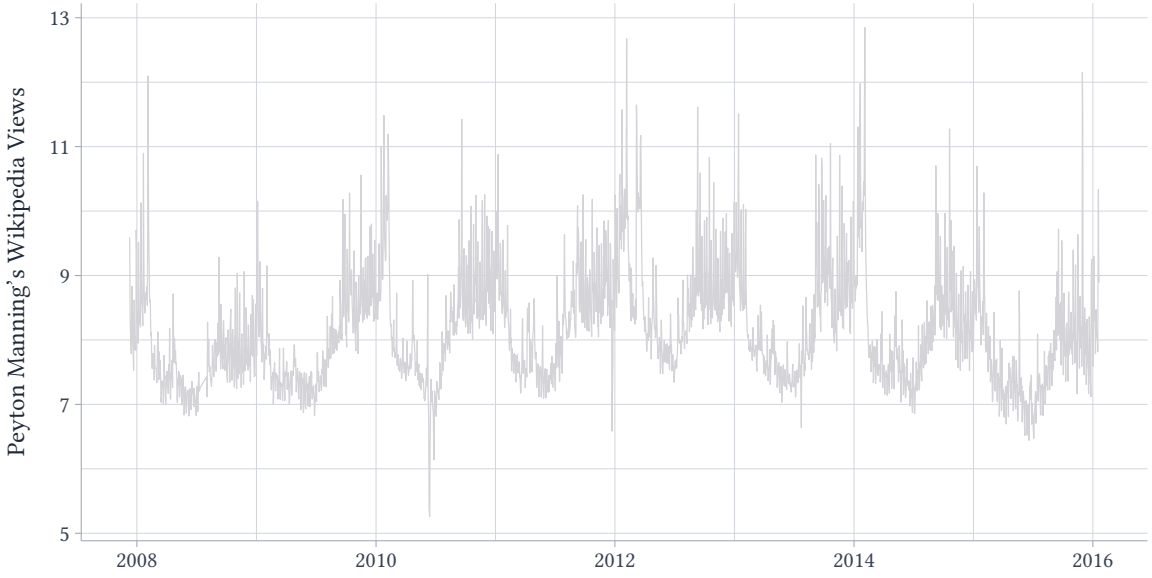
Monthly patterns vs. Year-by-month

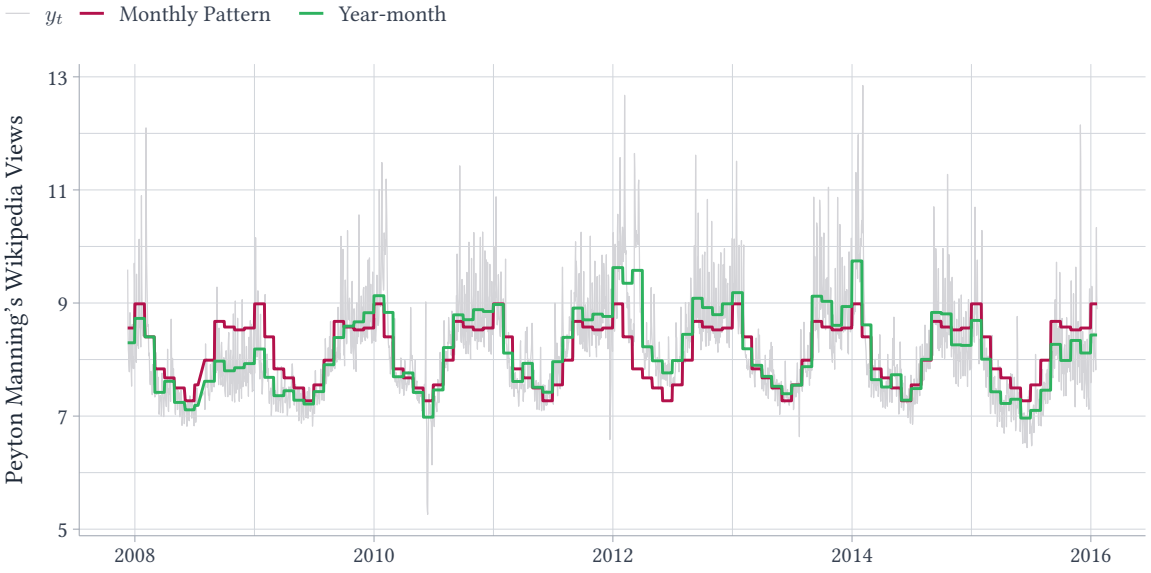
What we have seen so far is how to estimate a recurring monthly pattern

- Each year has the same predicted value in a given month

When you have something like weekly or daily data, you can estimate a year-by-month pattern

- The estimate for January 2015 is different from January 2016
- In `R`, you can use the `yearmonth()` function from the `tsibble` package


Peyton Manning's Wikipedia Views
13
11
9
7
5
2008
2010
2012
2014
2016


y_t
Monthly Pattern
Year-month
Peyton Manning's Wikipedia Views
13
11
9
7
5
2008
2010
2012
2014
2016

Monthly patterns vs. Year-by-month

Note that while year-by-month offers more detail, it is harder to interpret

- Monthly patterns are easier to convey and more informative for future forecasting
- Year-by-month patterns are more flexible at inference, but can not be used for forecasting in the future (January 2025 is different from Janaury 2024)

Monthly patterns vs. Year-by-month

Note that while year-by-month offers more detail, it is harder to interpret

- Monthly patterns are easier to convey and more informative for future forecasting
- Year-by-month patterns are more flexible at inference, but can not be used for forecasting in the future (January 2025 is different from Janaury 2024)

When your time-series has a longer interval (e.g. monthly), you can not use year-by-month indicator variables

- If you have monthly data, your year-by-month indicators will *perfectly* fit the data

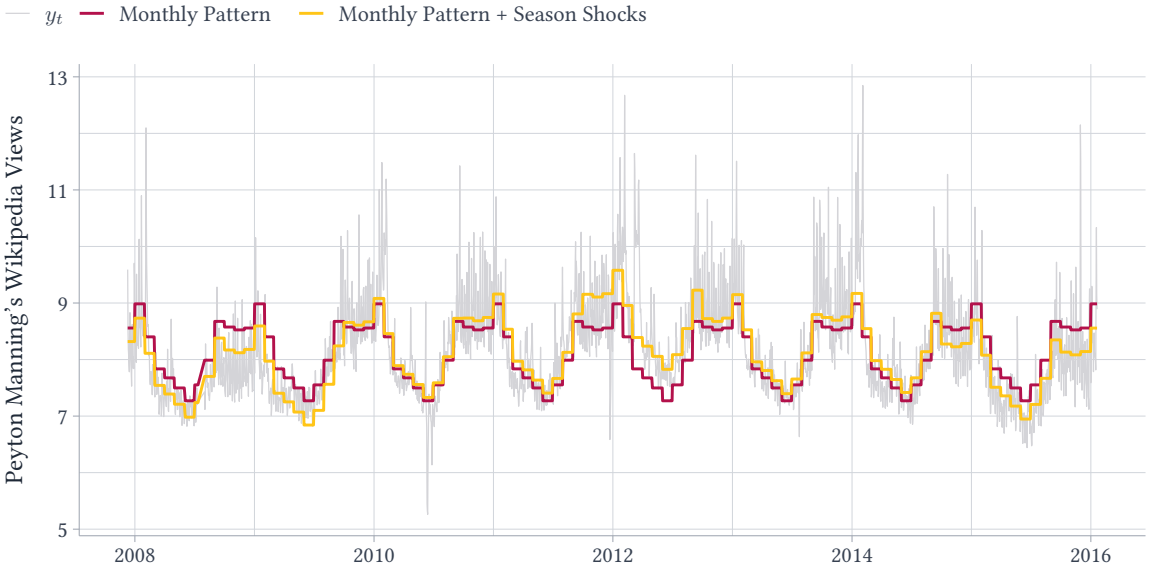
A mix of the two

One way to balance between these two approaches is to:

- Use month indicator variables for seasonality
- Use indicators for each year to let a level shift for each year

I did this for our Peyton Manning views dataset

- Because of domain knowledge, I use season effects rather than year effects (the ‘year’ starts in September)

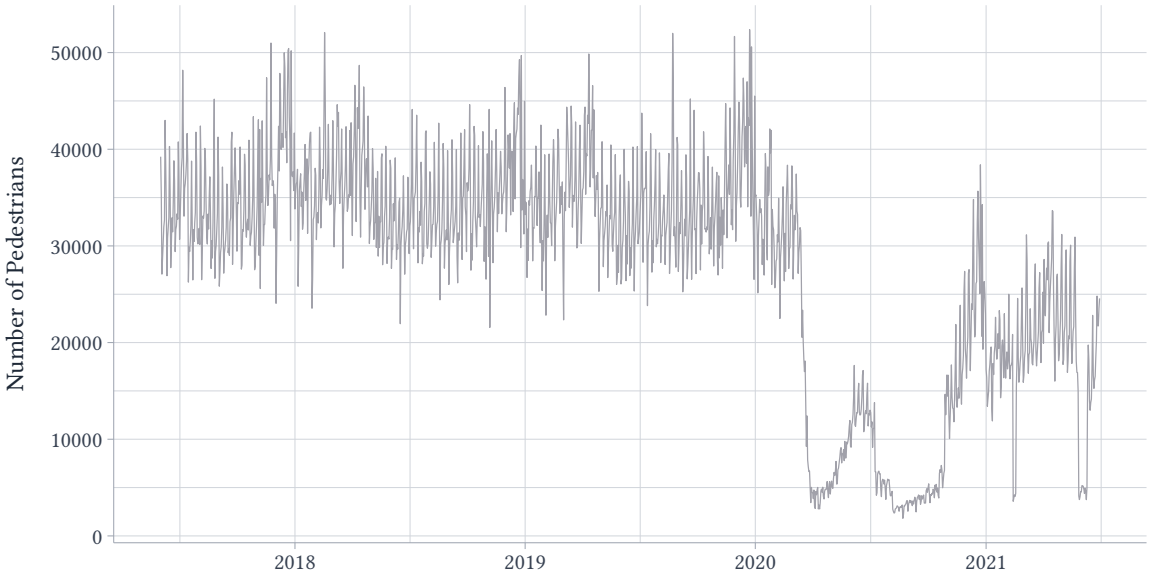

y_t
Monthly Pattern
Monthly Pattern + Season Shocks
Peyton Manning's Wikipedia Views
13
11
9
7
5
2008
2010
2012
2014
2016

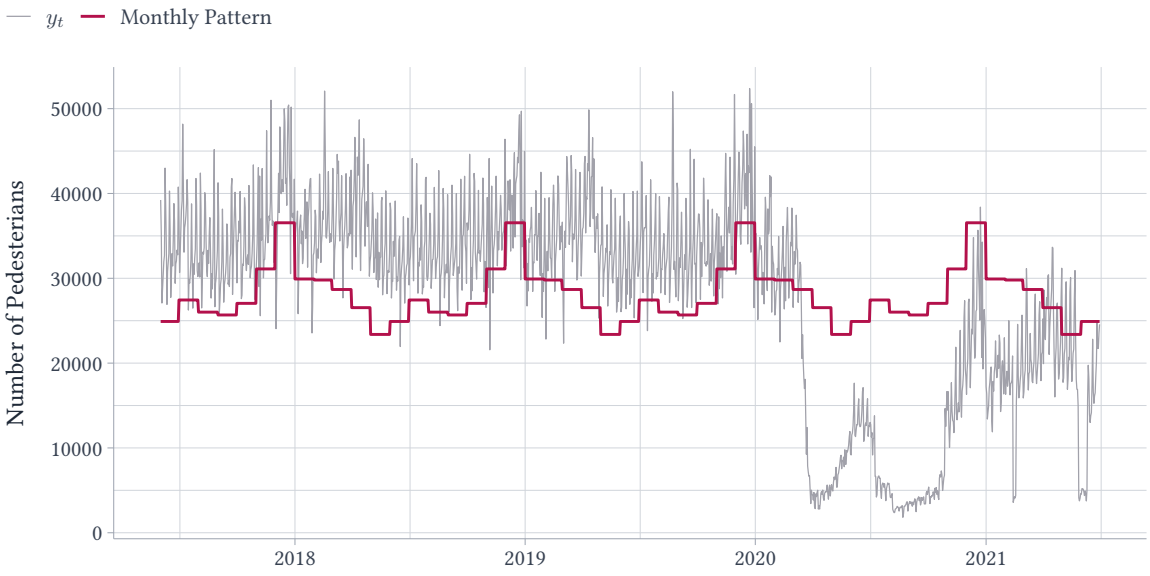
Outliers / Weird Shocks

Sometimes, your dataset will have really weird jumps

- E.g. Covid-19 pandemic shows up in a lot of time-series plots

These really odd periods of time, while few in number, can have a large effect on your forecasting models


Number of Pedestrians
50000
40000
30000
20000
10000
0
2018
2019
2020
2021


y_t
Monthly Pattern
Number of Pedesterians
50000
40000
30000
20000
10000
0
2018
2019
2020
2021

Dealing with Weird Shocks

For these periods, we can either

- (1) drop them from the regression (potentially losing valuable information)
- (2) or, add these shocks to our model

Dealing with Weird Shocks

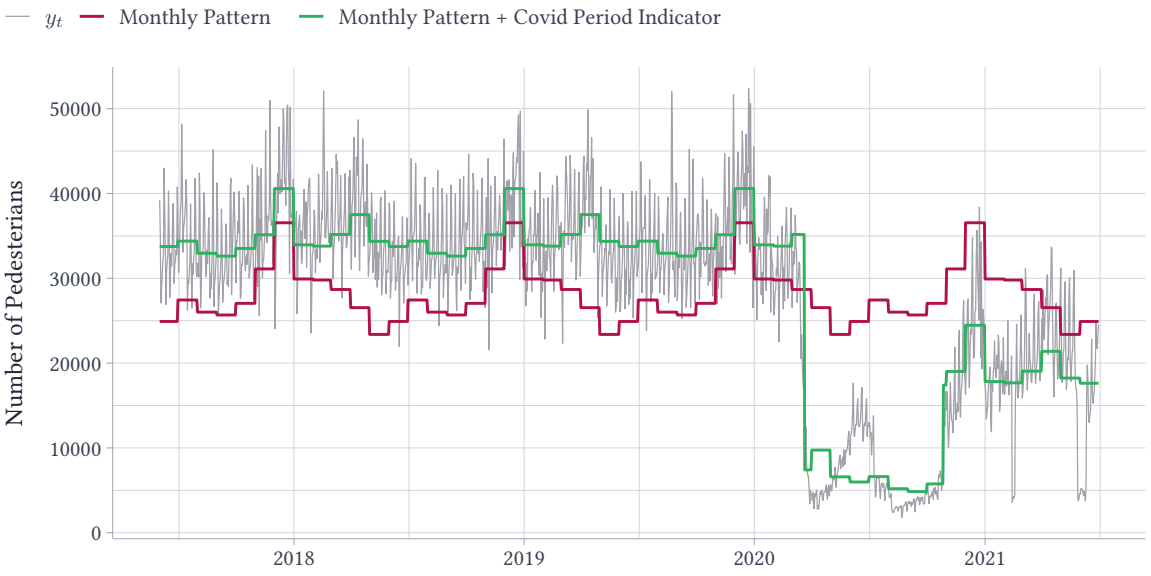
For these periods, we can either

- (1) drop them from the regression (potentially losing valuable information)
- (2) or, add these shocks to our model

To adapt our model, we will add indicator variables for ranges (similar to our Epoch time-trends)

$$y_t = \alpha + \sum_{m=1}^{12} \mathbb{1}[\texttt{Month}(t) = m]\beta_m + \text{Covid Period}_t\delta_1 + \text{Post-Covid Period}_t\delta_2 + u_t$$

- Let's see how this small change impacts our forecast performance


y_t
Monthly Pattern
Monthly Pattern + Covid Period Indicator
Number of Pedesterians
50000
40000
30000
20000
10000
0
2018
2019
2020
2021

Coding in R

```
df$covid_period <- (df$date >= ymd("2020-03-21")) &
  (df$date < ymd("2020-10-27"))

df$post_covid_period <- (df$date >= ymd("2020-10-27"))
feols(
  ppl ~ i(month) + i(covid_period) + i(post_covid_period),
  data = df, vcov = "hc1"
)
```

Dealing with Structural Changes

In some cases, we see fundamental changes to the economy

- Periods prior to some point have different seasonal patterns and trends

In this case, you could interact month indicators with pre- and post- indicators

- Month $\times$ Pre indicators estimate pattern *before* switch
- Month $\times$ Post indicators estimate pattern *after* switch
- Adding a Post indicator allows the average level to be different before/after

Additional covariates

So far all of our methods have just relied on using the `date` as the explanatory variable and have discussed different ways of using that:

- Seasonal patterns
- Linear, piecewise, or more flexible time-trends
- 'Discrete Breaks'

One huge advantage of regressions is that you can include other predictors in your model

- E.g. predicting sales on a given day using month indicators and *also* the price on the day

Including additional covariates

$$\text{sales}_t = \alpha + \sum_{m=1}^{12} \mathbb{1}[\texttt{Month}(t) = m]\beta_m + p_t\gamma + u_t$$

In this case, we include linearly the price charged at time t for the good

Including additional covariates

$$\text{sales}_t = \alpha + \sum_{m=1}^{12} \mathbb{1}[\texttt{Month}(t) = m]\beta_m + p_t\gamma + u_t$$

In this case, we include linearly the price charged at time t for the good

This regression model can be (approximately) interpreted as:

1. First, removing the portion of sales that are predicted by the variation in price over time
2. Second, running a time-series regression on the remaining variation

Usefullness of covariates

$$\text{sales}_t = \alpha + \sum_{m=1}^{12} \mathbb{1}[\texttt{Month}(t) = m]\beta_m + p_t\gamma + u_t$$

The big advantage is we can now make predictions into the future where we both:

- Set the price to what we intend for it to be
- And extrapolate the time-series pattern into the future

Inference on forecasts

So far we have discussed creating inference and forecasts $\hat{y}_t$ about time-series

We have remained silent on how to express uncertainty around our findings

- This is the second half of statistics!

Prediction from regression

For generality, we will consider the generic multiple regression model:

$$y_t = \beta_0 + X_{1,t}\beta_1 + \cdots + X_{K,t}\beta_K + u_t$$

- E.g. $K = 1$ and $X_{1,t} = t$ is the linear-time trend model

For a given value of $(x_1, \ldots, x_K)$, our prediction is given by

$$\hat{y} = \hat{\beta}_0 + x_1\hat{\beta}_1 + \cdots + x_K\hat{\beta}_K$$

Prediction from regression

Compared to the true expected value of y:

$$\mathbb{E}(y \mid x) = \beta_{0,0} + x_1\beta_{1,0} + \cdots + x_K\beta_{K,0}$$

The difference between the two is due to noise in the coefficient estimates:

$$\hat{y} - \mathbb{E}(y \mid x) = \left(\hat{\beta}_0 - \beta_{0,0}\right) + x_1\left(\hat{\beta}_1 - \beta_{1,0}\right) + \cdots + x_K\left(\hat{\beta}_K - \beta_{K,0}\right)$$

$\implies$ In repeated samples $\hat{\beta}$ are the only terms that vary

Inference on Regression Predictions

$$\hat{y} - \mathbb{E}(y \mid x) = \left(\hat{\beta}_0 - \beta_{0,0}\right) + x_1 \left(\hat{\beta}_1 - \beta_{1,0}\right) + \cdots + x_K \left(\hat{\beta}_K - \beta_{K,0}\right)$$

Remember, we know how to express uncertainty around each $\hat{\beta}$ using the SE($\hat{\beta}$) from our regression table

- However, that is not enough since $\hat{\beta}$ might be correlated with each-other

Inference on Regression Predictions

$$\hat{y} - \mathbb{E}(y \mid x) = \left(\hat{\beta}_0 - \beta_{0,0}\right) + x_1 \left(\hat{\beta}_1 - \beta_{1,0}\right) + \cdots + x_K \left(\hat{\beta}_K - \beta_{K,0}\right)$$

Remember, we know how to express uncertainty around each $\hat{\beta}$ using the SE($\hat{\beta}$) from our regression table

- However, that is not enough since $\hat{\beta}$ might be correlated with each-other

This means each term in the above are correlated, so inference is more difficult

- Fortunately, the `predict` function you've seen provides standard errors on our prediction

In-sample prediction

The first-thing we might want to do is predict $\hat{y}_t$ in our sample and add confidence intervals. This returns a data.frame with two-columns $\hat{y}$ and SE($\hat{y}$)

```
est <- feols(y ~ date, data = df, vcov = "hc1")
# In-sample predictions
predictions <- predict(est, se.fit = TRUE)
```

```
       fit   se.fit
1 20817.14 225.9461
2 20819.04 226.0038
3 20820.93 226.0615
4 20822.82 226.1192
```

Forecasting into the future

Then, we could try and predict out-of-sample. To do this, we need to create a new data.frame with the X variables we need

```
# The next 5 days from our sample
prediction_df <- data.frame(
  date = ymd("2021-06-30") + 1:5
)
predict(est_linear_trend, newdata = prediction_df, se.fit = TRUE)
```

Forecasting and confidence intervals

```
predict(est_linear_trend, newdata = prediction_df, se.fit = TRUE)
```

```
       fit   se.fit
1 23635.83 314.3643
```

For the first-prediction, form a 95% confidence interval for this prediction:

Forecasting and confidence intervals

```
predict(est_linear_trend, newdata = prediction_df, se.fit = TRUE)
```

```
       fit   se.fit
1 23635.83 314.3643
```

For the first-prediction, form a 95% confidence interval for this prediction:

$$23635.83 \pm 1.96 * 314.3643 = (23019.68, 24251.98)$$

Forecasting and confidence intervals

```
predictions <-
  predict(est_linear_trend, newdata = prediction_df, se.fit = TRUE)
predictions$ci_lower <- predictions$fit - 1.96 * predictions$se.fit
predictions$ci_upper <- predictions$fit + 1.96 * predictions$se.fit
```

```
       fit   se.fit ci_lower ci_upper
1 23635.83 314.3643 23019.67 24251.98
2 23637.72 314.4248 23021.45 24253.99
3 23639.61 314.4854 23023.22 24256.00
4 23641.50 314.5459 23024.99 24258.01
5 23643.39 314.6064 23026.77 24260.02
```

Smoothing Methods

Recall we said y_t was generated by

$$y_t = \mu_t + \varepsilon_t$$

- 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

- Want to "smooth" over random fluctuations

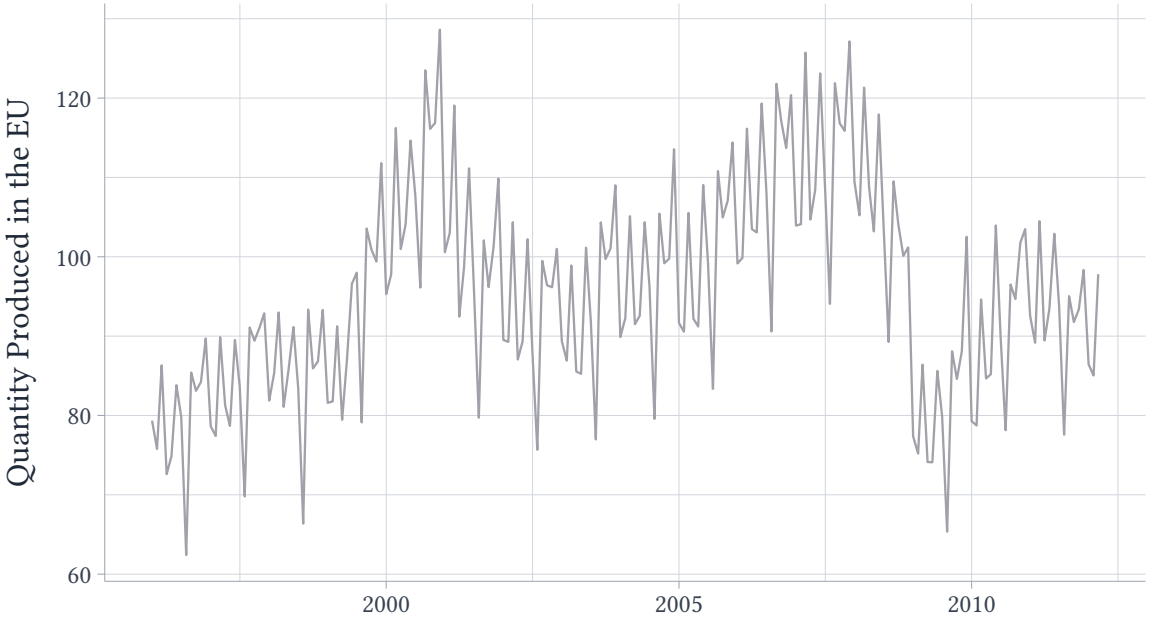
Example Electrical Manufacturing in the EU

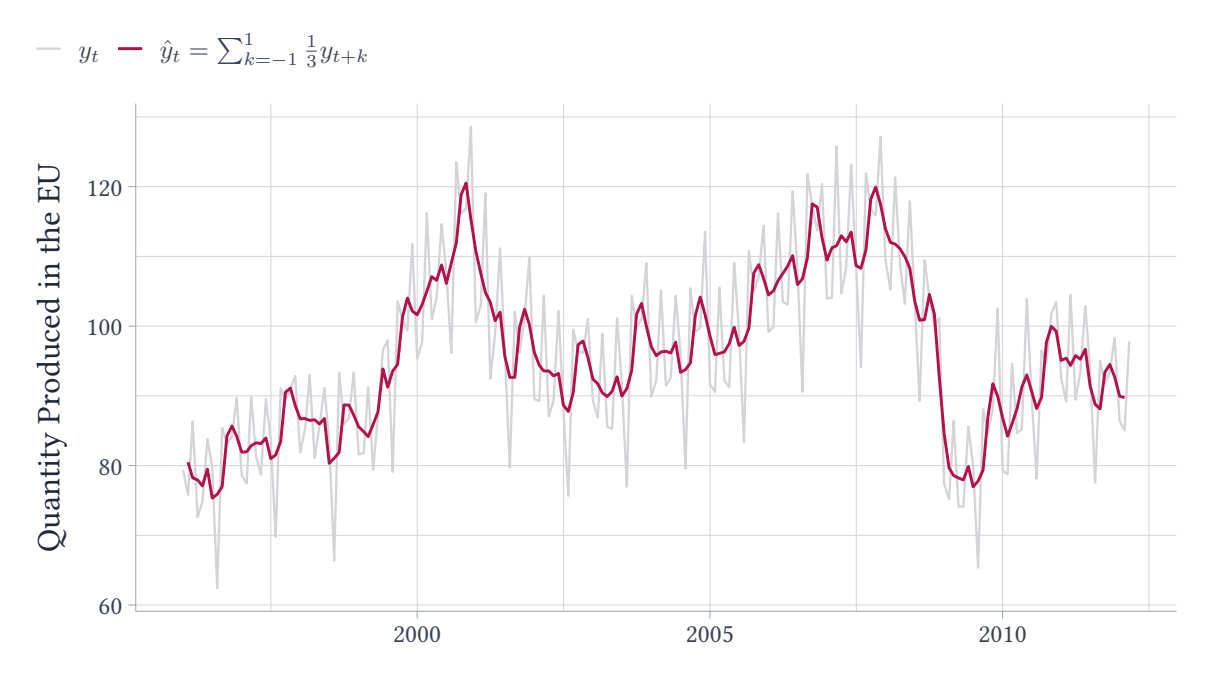
On the following slide I'm going to show you production figures across the European Union

- 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

Smoothing methods

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

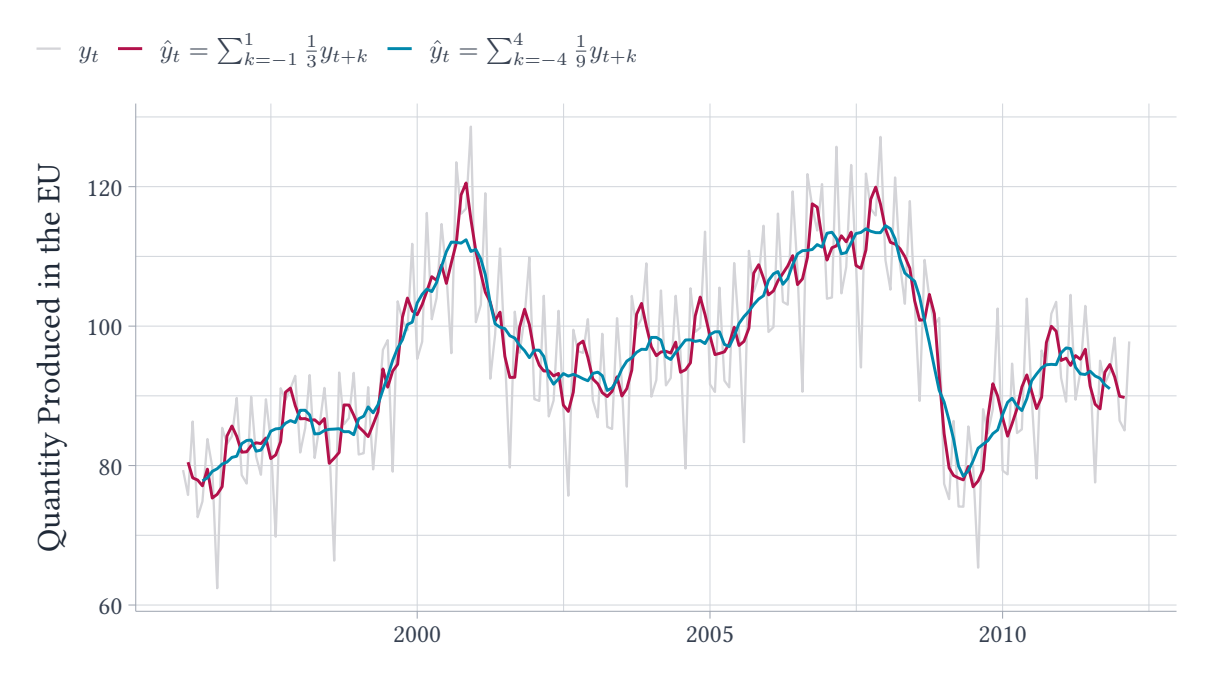
- This helped to smooth out some of the random fluctuations, perhaps better isolating systematic trends in y_t

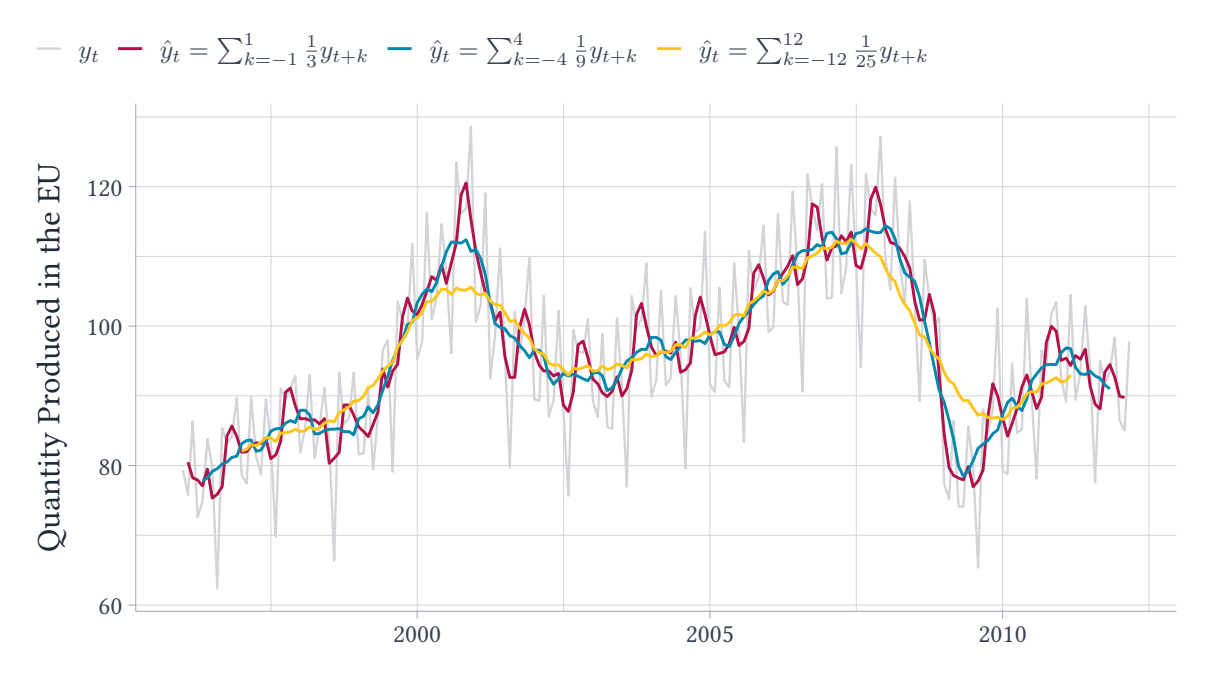
Smoothing methods

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

- 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

- K is the number of observations on each side of y_t we include
- $2K+1$ 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

- K is the number of observations on each side of y_t we include
- $2K+1$ 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

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

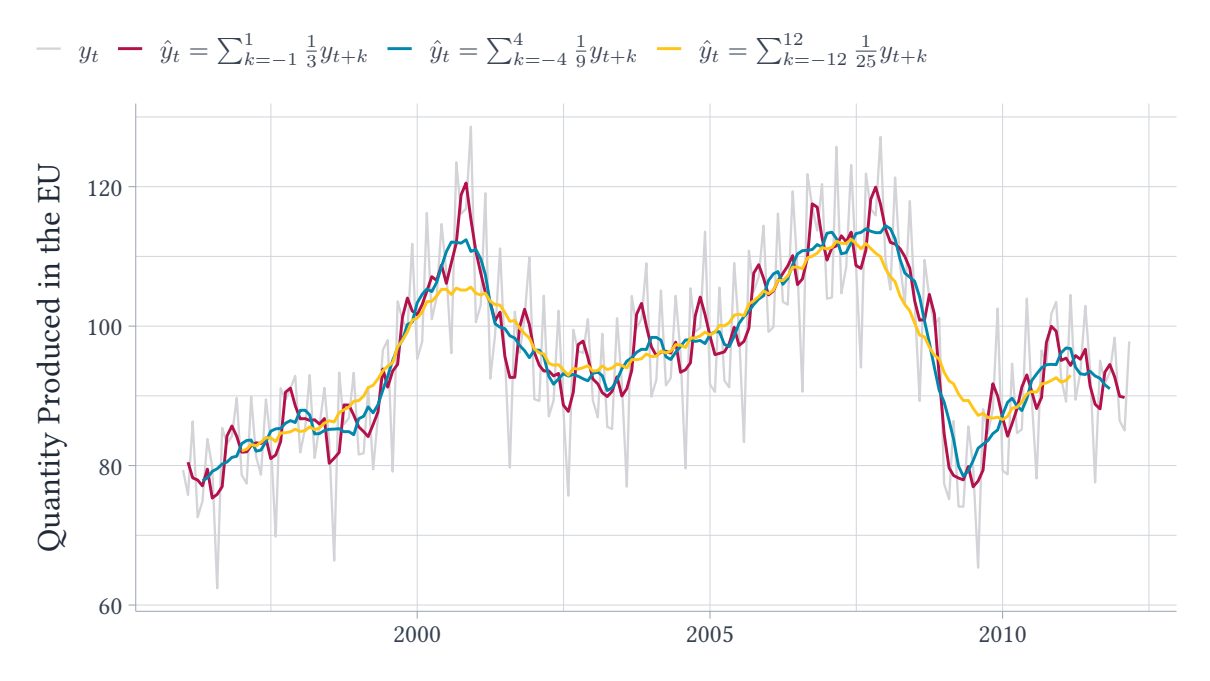
What happened to the end points?

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

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

- 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)

- 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)

- 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

- 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

- 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

- 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

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

Selecting K

There is a trade-off at play

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

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

- 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:

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

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

- 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

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

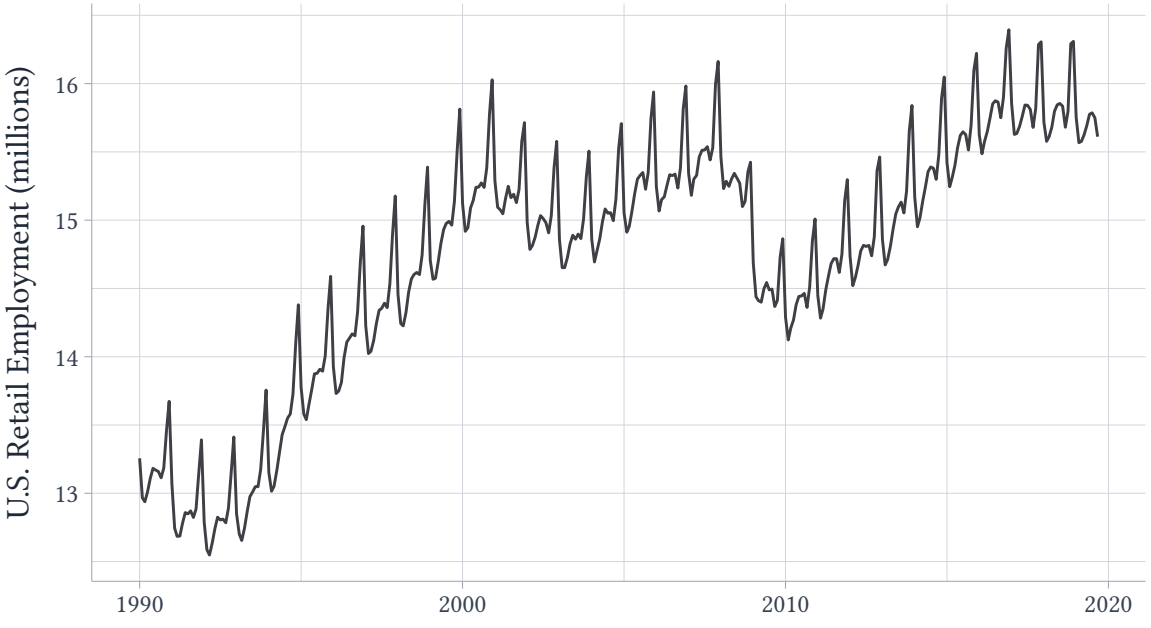
Basically a $\pm K$ smoothing average, but first and last get a half the weight

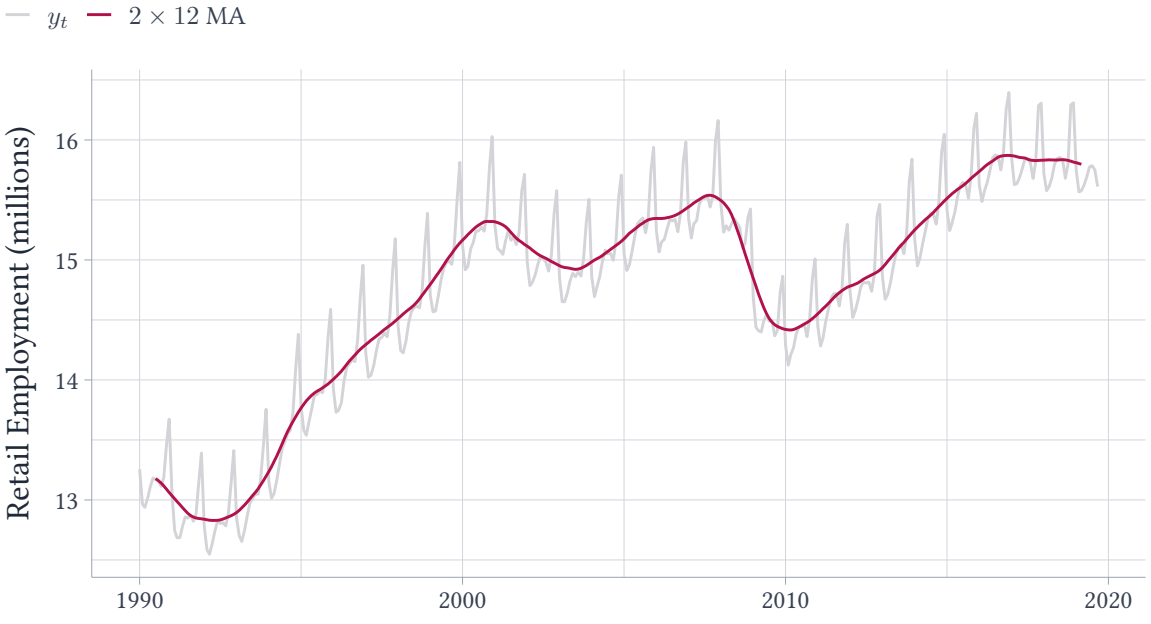
Moving average to remove seasonality, S_t

Or, can do the following:

- 12-month rolling average
- 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 $y_t - \hat{T}_t$

- $\hat{T}_t$ 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

- 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

- Each month's average serves as the estimated month's "seasonal trend", $\hat{S}_t$
  $\rightarrow$ Takes the same value year over year

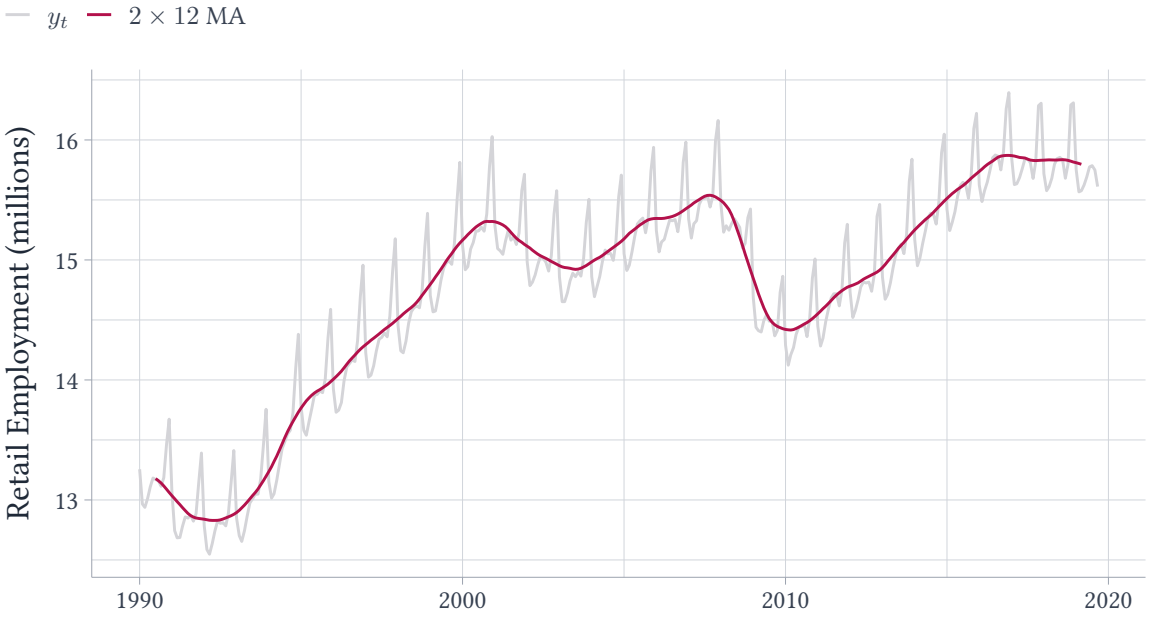
Seasonality estimation in R

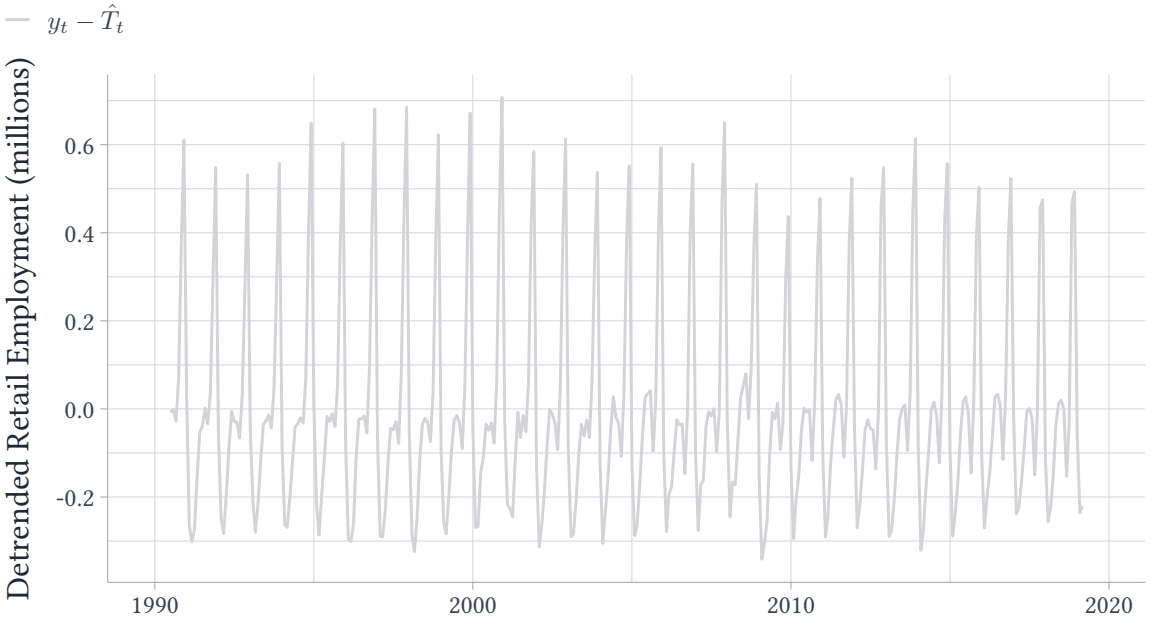
The classical way to estimate this is take the average of $y_t - \hat{T}_t$ separately for each month

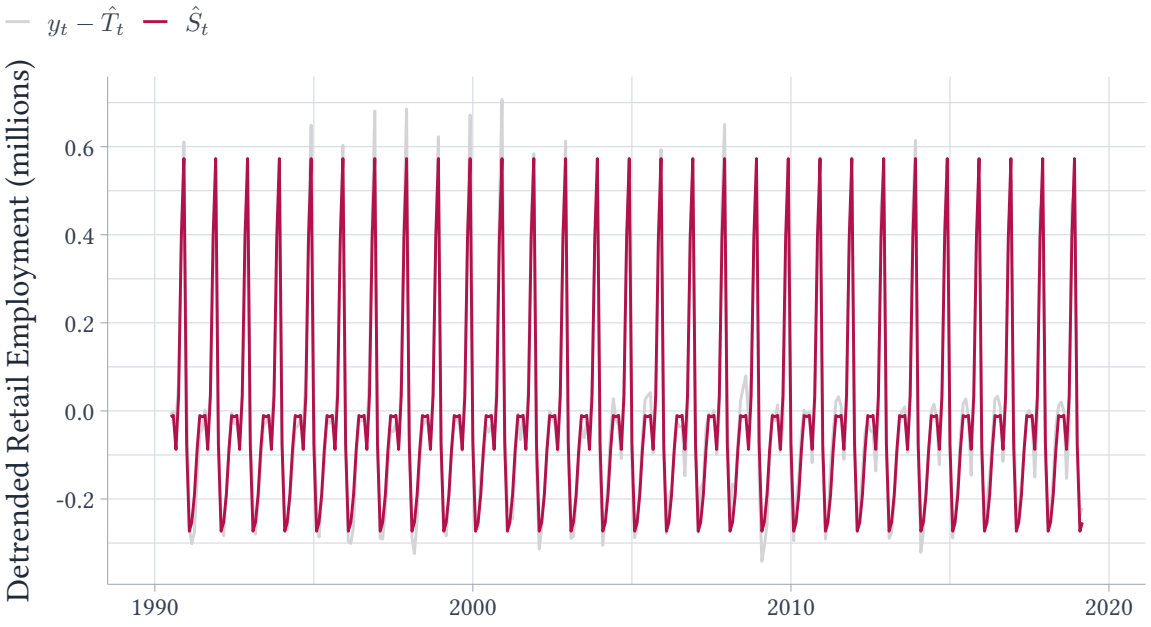
- This can be done by regression $y_t - \hat{T}_t$ on a set of month indicators (and no intercept)

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

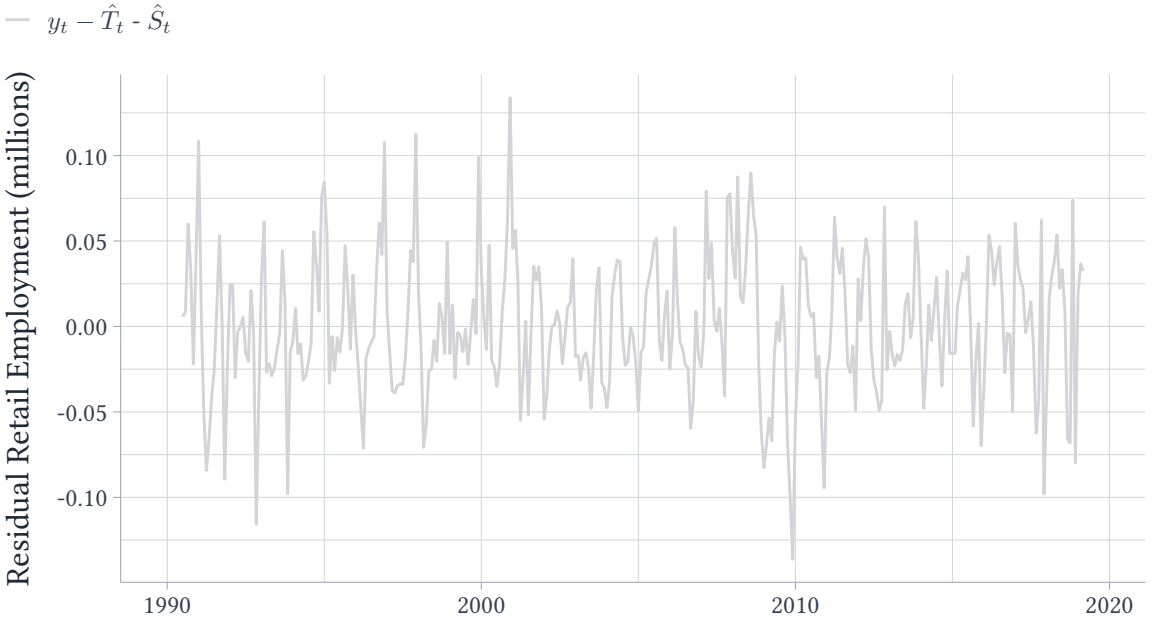
Residual

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

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

- 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

Smoothing Methods for Forecasting

Our previous goal was to learn the ‘systematic’ part of the time-series $y_t = \mu_t + \varepsilon_t$

“Rules” for forecasting

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

- 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

- 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

- Maybe the most recent observation is the most-relevant for predicting today
  $\rightarrow$ I.e. autocorrleation is high
- 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:

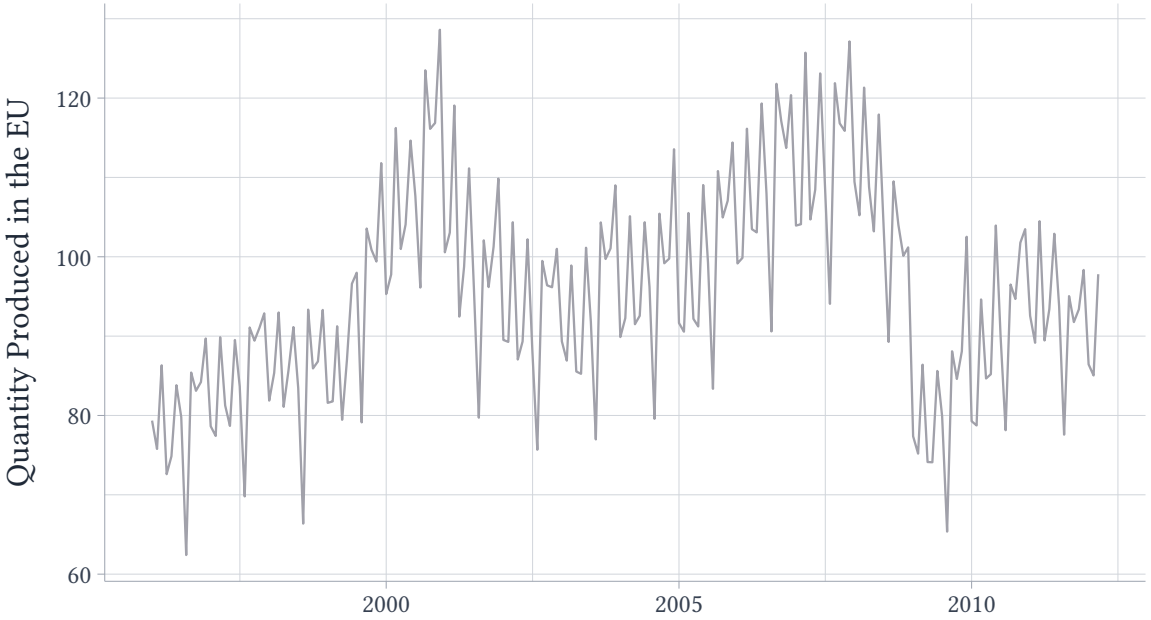
- The data has trends or seasonality that y_{t-1} doesn't capture

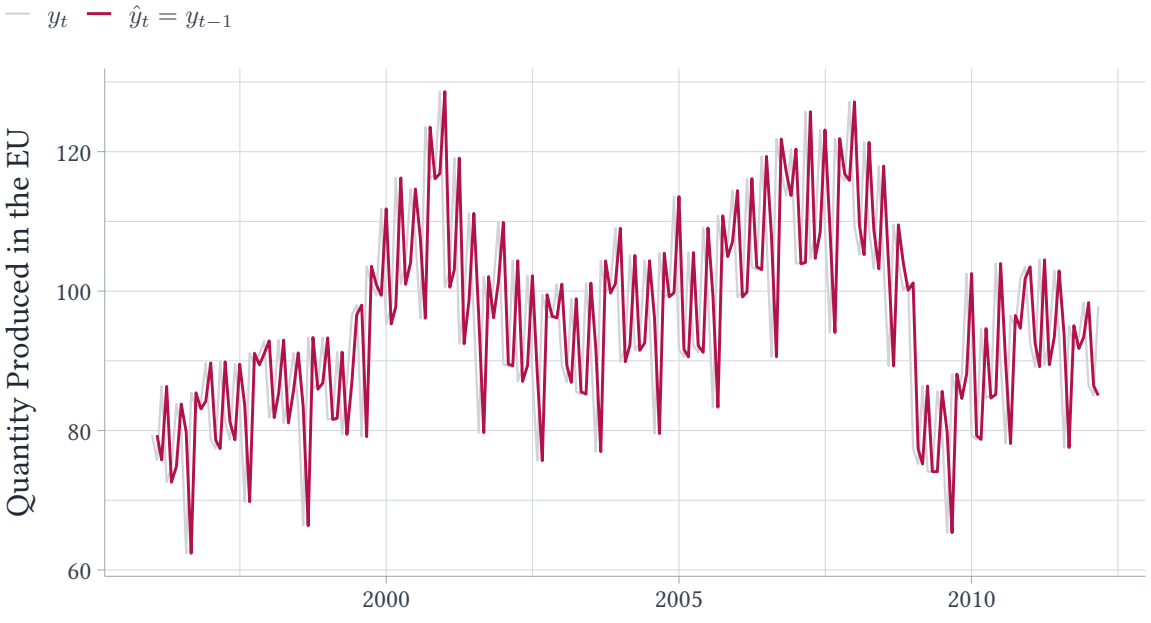
  $\rightarrow$ Using August's jacket sales to predict September's jacket sales will not do well

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

Using y_{t-1} could fail when:

- The data has trends or seasonality that y_{t-1} doesn't capture
  - $\rightarrow$ Using August's jacket sales to predict September's jacket sales will not do well
- y_{t-1} can be quite *noisy*
  - $\rightarrow$ 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

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

Smoothing Methods

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

- K is the number of lags to smooth over
- 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:

- $K = 1$ 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

Average of previous ys

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

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

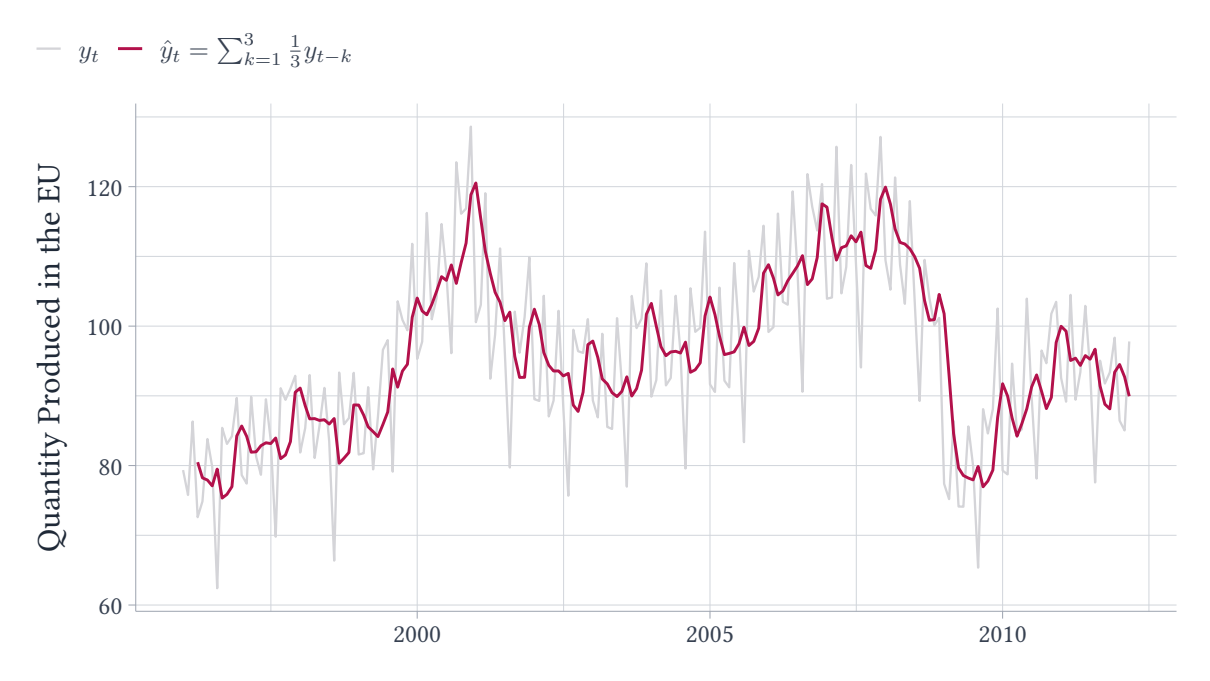
Average of previous ys

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

- 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
120
100
80
60
2000
2005
2010

Prediction

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

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

- This is not true of our mouving average

Exponential Smoothing

It turns out, there is a (typically) better method over the sample average of the previous K periods.

It is called Exponential Smoothing and is quite popular because it works well

- Also has some generalizations that allow it to be more flexible

Simple Exponential Smoothing

The simple exponential smoothing method forms predictions in a *recursive manner*:

$$\hat{y}_{t+1} = \alpha y_t + (1 - \alpha)\hat{y}_t$$

- To predict y in period $t + 1$, take a weighted sum of the observed y_t and the prediction $\hat{y}_{t-1}$

We learn the true value of y_t and update our prediction for the next period

- When $y_t > \hat{y}_t$, we revise up our forecast
- When $y_t < \hat{y}_t$, we revise down our forecast

How much to update, α

$$\hat{y}_{t+1} = \alpha y_t + (1 - \alpha)\hat{y}_t$$

α tells us how much to update

- $\alpha = 1$ means throw out old prediction and use y_t
- $\alpha = 0$ means do not update at all
- $1 > \alpha > 0$ means updating more (close to 1) or less strongly (close to 0)

Simple Exponential Smoothing

$$\hat{y}_{t+1} = \alpha y_t + (1 - \alpha)\hat{y}_t$$

There's a problem here though, because how do we get $\hat{y}_t$? Well we get it using $\hat{y}_{t-1}$

- And to get $\hat{y}_{t-1}$ we need $\hat{y}_{t-2}$...
- and turtles all the way down ...

Simple Exponential Smoothing

Starting from period $t = 1$,

$$\begin{aligned}
\hat{y}_2 &= \alpha y_1 + (1-\alpha)\ell_0 \\
\hat{y}_3 &= \alpha y_2 + (1-\alpha)\hat{y}_2 \\
&\vdots \\
\hat{y}_T &= \alpha y_{T-1} + (1-\alpha)\hat{y}_{T-1} \\
\hat{y}_{T+1} &= \alpha y_T + (1-\alpha)\hat{y}_T.
\end{aligned}$$

So, we only need the starting point ℓ_0

Simple Exponential Smoothing

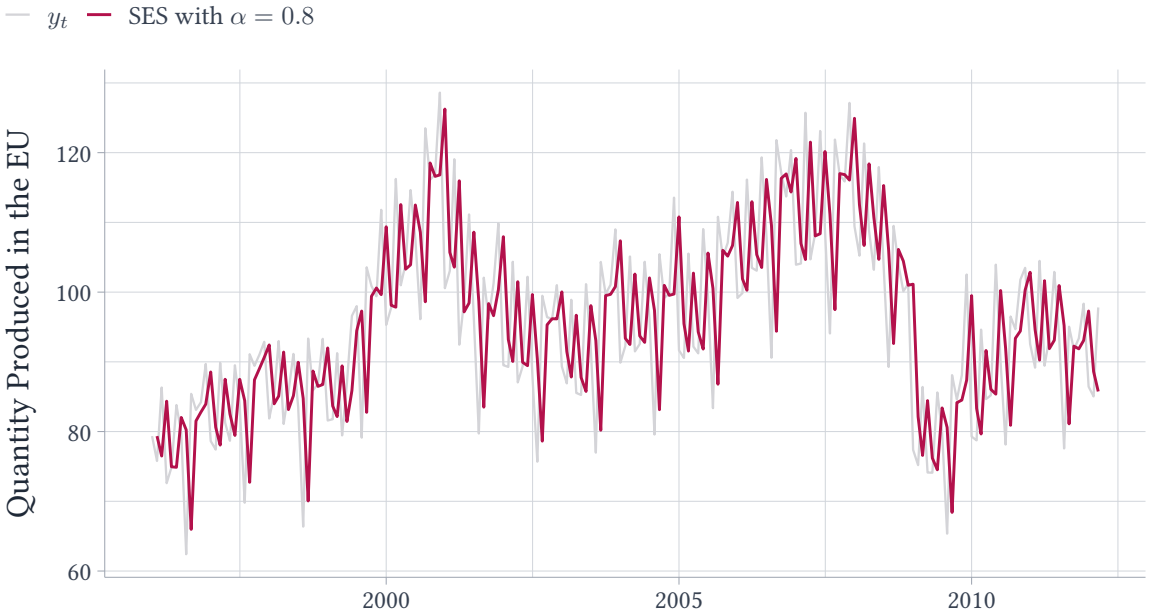
Since we usually do not care that much about predicting early period's y's, we can just pick ℓ_0 to be y_1

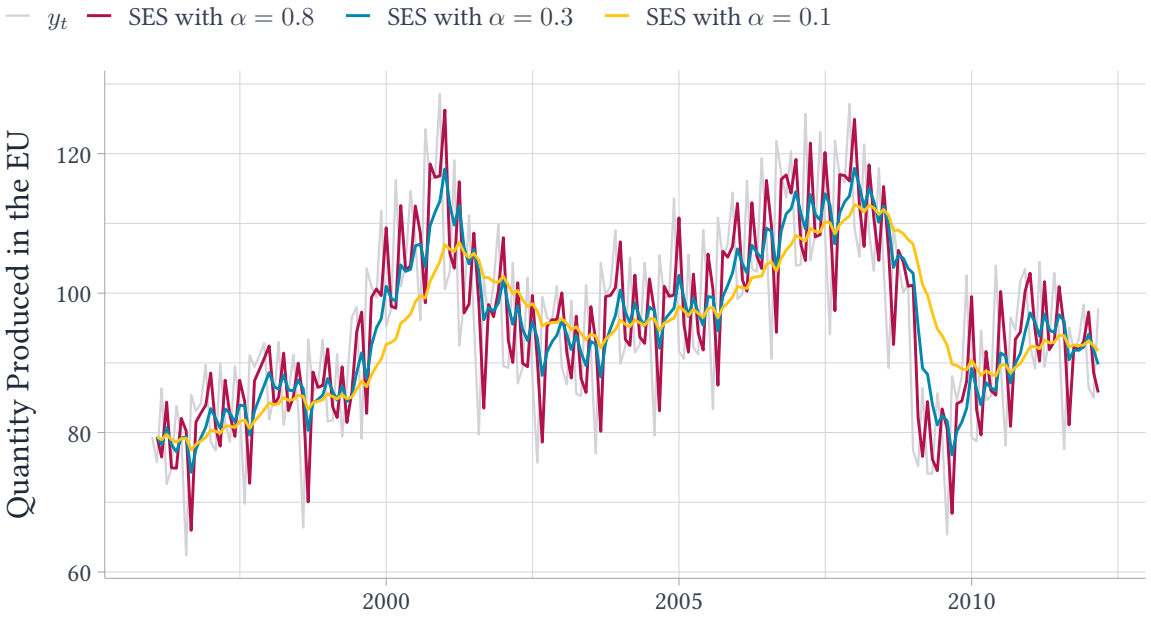
- It does not turn out to matter that much for forecasting if T is large

Updating parameter α

$$\hat{y}_{t+1} = \alpha y_t + (1 - \alpha)\hat{y}_t$$

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


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

Simple Exponential Smoothing and Recursion

We can trace out how $\hat{y}_t$ works as follows:

$$\begin{aligned}\hat{y}_{t+1} &= \alpha y_t + (1-\alpha)\hat{y}_t \\ &= \alpha y_t + (1-\alpha)\left(\alpha y_{t-1} + (1-\alpha)\hat{y}_{t-1}\right)\end{aligned}$$

Simple Exponential Smoothing and Recursion

We can trace out how $\hat{y}_t$ works as follows:

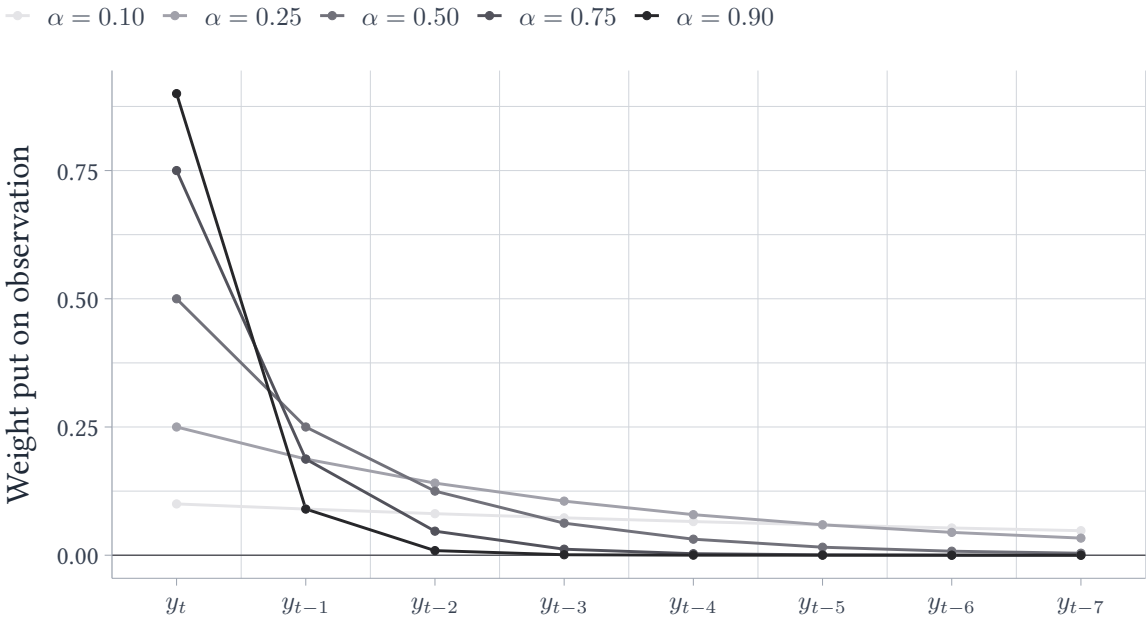
$$\begin{aligned}\hat{y}_{t+1} &= \alpha y_t + (1-\alpha)\hat{y}_t \\ &= \alpha y_t + (1-\alpha)\left(\alpha y_{t-1} + (1-\alpha)\hat{y}_{t-1}\right) \\ &= \alpha y_t + \alpha(1-\alpha) y_{t-1} + (1-\alpha)^2 \hat{y}_{t-1}\end{aligned}$$

Simple Exponential Smoothing and Recursion

You can repeat this process many times

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

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


$\alpha = 0.10$
$\alpha = 0.25$
$\alpha = 0.50$
$\alpha = 0.75$
$\alpha = 0.90$
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 ‘adaptability’

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

- A large α is more ‘adaptable’ in that it can respond much more quickly to changes in y
- A small α puts weight more evenly

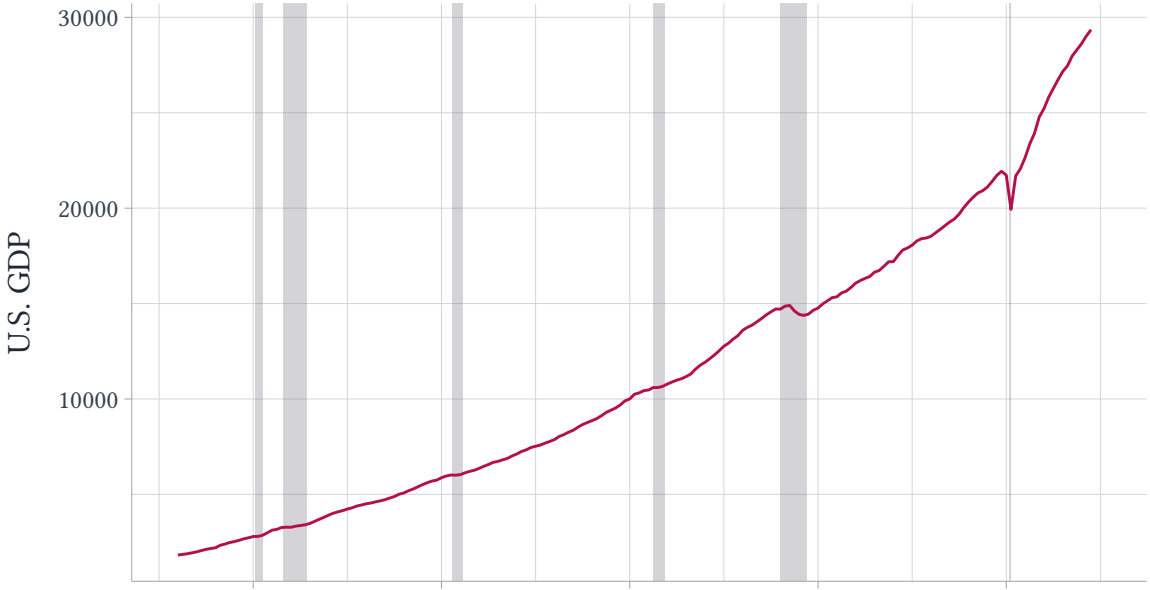
The different emphasis that these weights put have implications for how the SES method deals with trends and seasonality

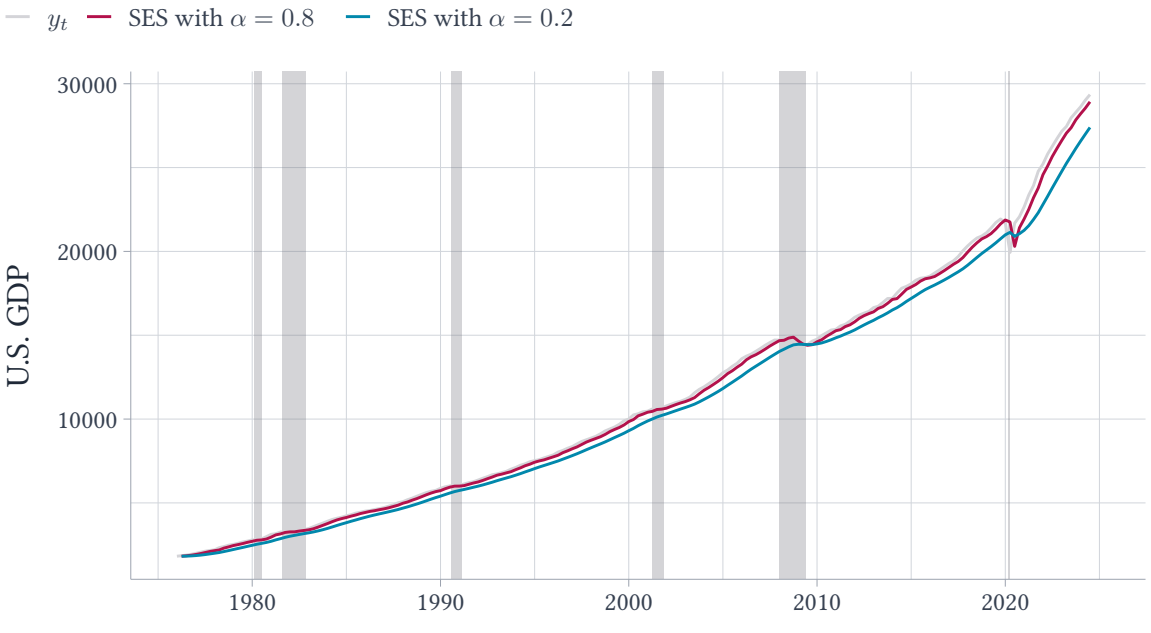
Trends

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


U.S. GDP
30000
20000
10000
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

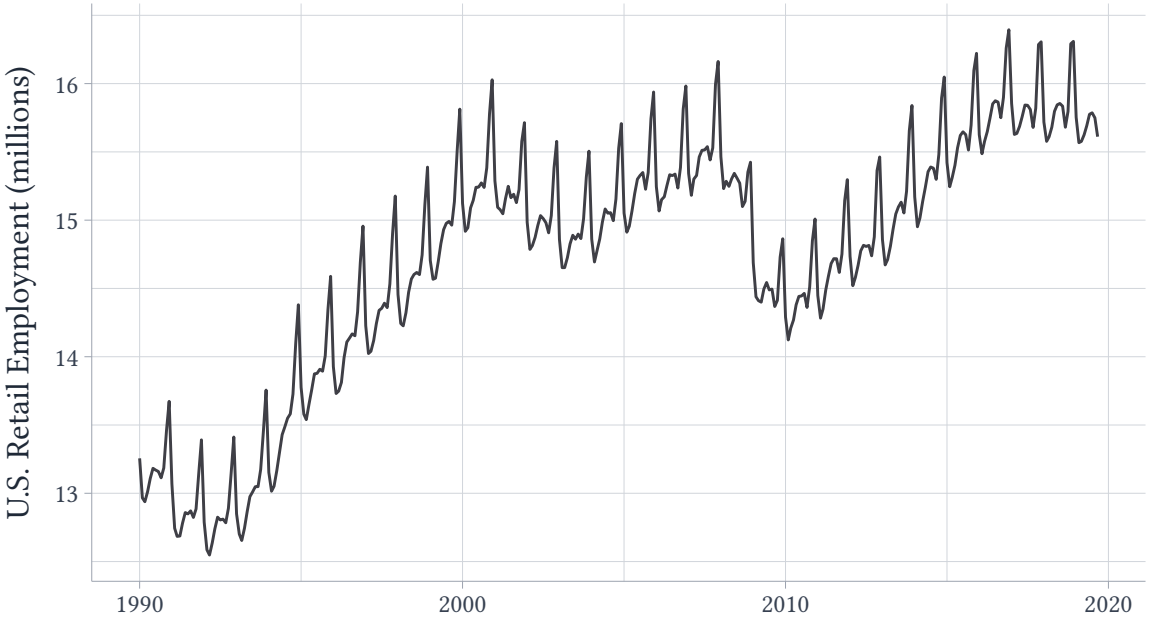
Simple Exponential Smoothing and Trends

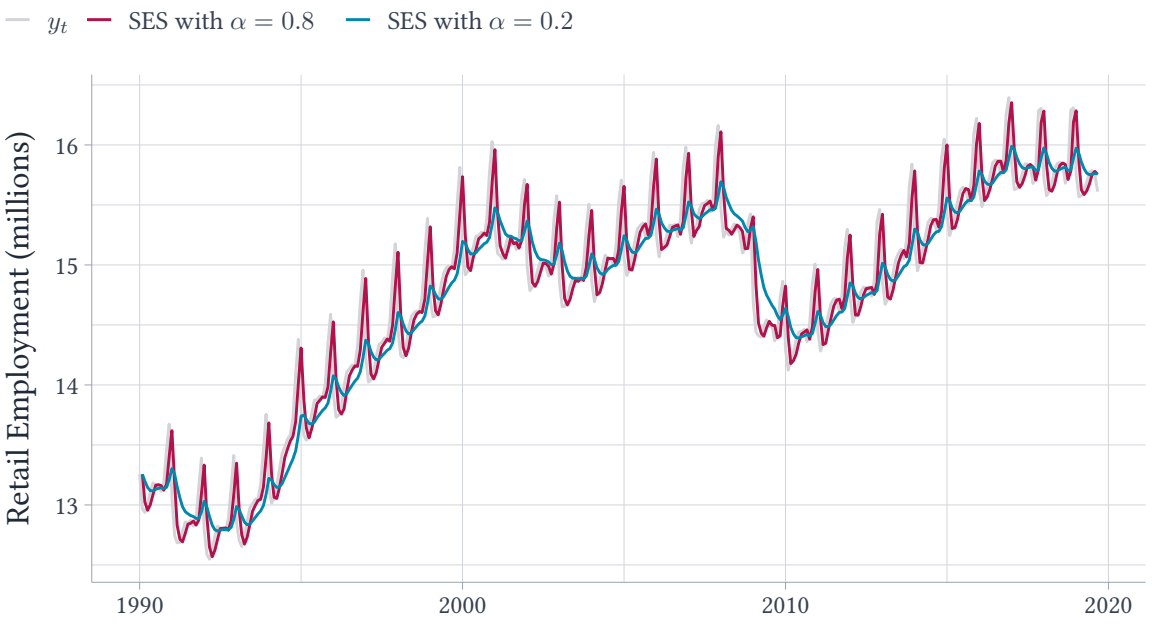
Since GDP is consistently trending upwards, old values of y_{t-k} are systematically lower

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

SES will also miss trends

- 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

Holt-Winters method

Once again, we face this problem with these methods that we focus mainly on using recent observations in terms of t

- This ignores *trends* and *seasonality*

The Holt-Winters method is a more advanced method that allows for (1) repeated seasonality (year-over-year) and (2) smoothly-evolving trends

- Allows us to not 'overfit' short term fluctuations as much by learning something about general longer-run trends