Topic 6: Regression with Time Series Data

ECON 4753 — University of Arkansas

Prof. Kyle Butts

April 2025

Time-series Regression

Time-series Predictors

- Seasonality
- Fourier Series
- Time-trends
- More flexibly modelling trends
- Step-functions
- Additional covariates

(Review of) Inference on regression forecasts

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

- Holts-Winter method can help with this

→ And, they required regular time-series

Time-Series Regression Methods

In this topic, we will focus on a different approach to conducting inference on time-series using our trusted friend *regression*:

$\rightarrow$ Regression uses all the observations to fit the model, so can better learn "long-term" trends and seasonality

$\rightarrow$ Can have irregular time-series

$\rightarrow$ Regression can include other predictors (e.g. predicting GDP given unemployment)

Compared to smoothing methods, regression does a less-good job at predicting short-term fluctuations

Time-series Regression Set-up

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

$\rightarrow$ The ‘unit of observation’ is a time-period t

$\rightarrow$ The outcome variable is the variable measured at time t, y_t

$\rightarrow$ we will have a set of explanatory variables for each time-period (including functions of t itself!)

Time-series Regression Set-up

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

$\rightarrow$ The 'unit of observation' is a time-period t

$\rightarrow$ The outcome variable is the variable measured at time t, y_t

$\rightarrow$ we will have a set of explanatory variables for each time-period (including functions of t itself!)

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

$\rightarrow$ 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

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

$\rightarrow$ 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

$\rightarrow$ E.g. if we are missing data for some months, we can still estimate our model.

- Smoothing methods have a difficult time with missing data

Inference in Time-series Regression

Inference is more complicated with time-series regressions

$\rightarrow$ Shocks to one time-period u_t can show up and impact other time-periods u_s

$\rightarrow$ In principle, all observations are related with one-another

Inference in Time-series Regression

Inference is more complicated with time-series regressions

$\rightarrow$ Shocks to one time-period u_t can show up and impact other time-periods u_s

$\rightarrow$ In principle, all observations are related with one-another

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

$\rightarrow$ Our sample is one realization of the time-series

$\rightarrow$ Resampling is like ‘rewinding’ and getting a new history for $t = 1, \dots, T$

Inference in Time-series Regression

Inference is more complicated with time-series regressions

$\rightarrow$ Shocks to one time-period u_t can show up and impact other time-periods u_s

$\rightarrow$ In principle, all observations are related with one-another

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

$\rightarrow$ Our sample is one realization of the time-series

$\rightarrow$ Resampling is like 'rewinding' and getting a new history for $t = 1, \dots, T$

This topic is well-studied, but a good starting point is to use Newey-West standard errors (`nw` in `fixest`)

Time-series Regression

Time-series Predictors

- Seasonality
- Fourier Series
- Time-trends
- More flexibly modelling trends
- Step-functions
- Additional covariates

(Review of) Inference on regression forecasts

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:

- $\rightarrow$ Quarterly patterns
- $\rightarrow$ Monthly patterns
- $\rightarrow$ Week-of-year patterns
- $\rightarrow$ Day-of-week patterns
- $\rightarrow$ 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. Then, create indicator variables using `i()` in your call to `feols` (from the `fixest` package)

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

Estimating seasonal trends

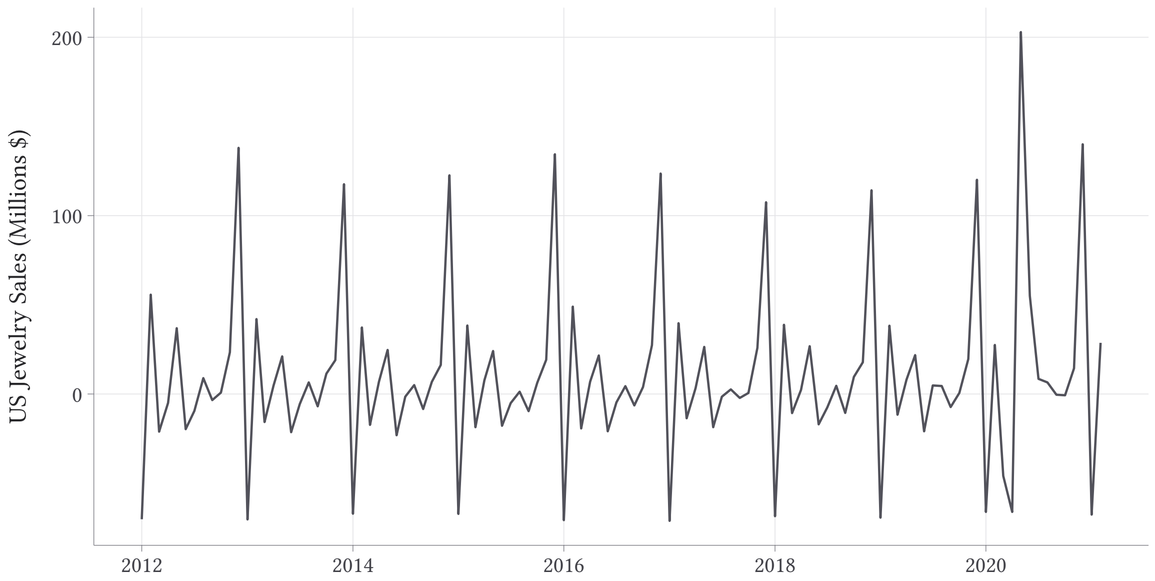
For example, consider the estimating a monthly pattern. The regression will be given by

$$y_t = \alpha + \sum_{m=2}^{12} \mathbb{1}[\texttt{Month}(t) = m]\beta_m + u_t$$

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

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

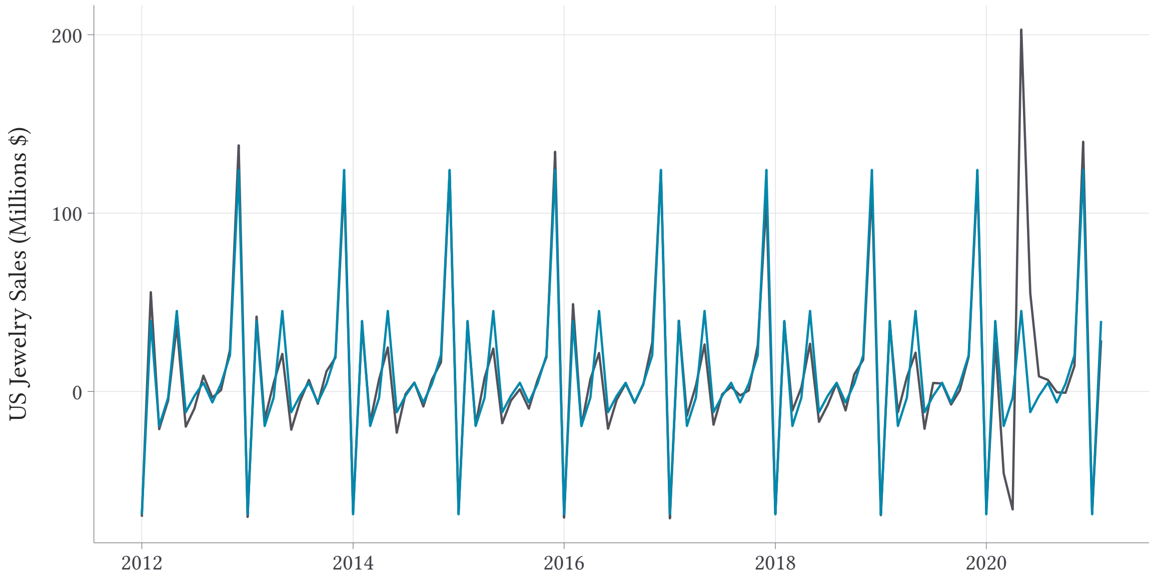

US Jewelry Sales (Millions $)
200
100
0
2012
2014
2016
2018
2020

Predicting jewerly sales monthly pattern

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

# predict sales ($) using the model
df$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

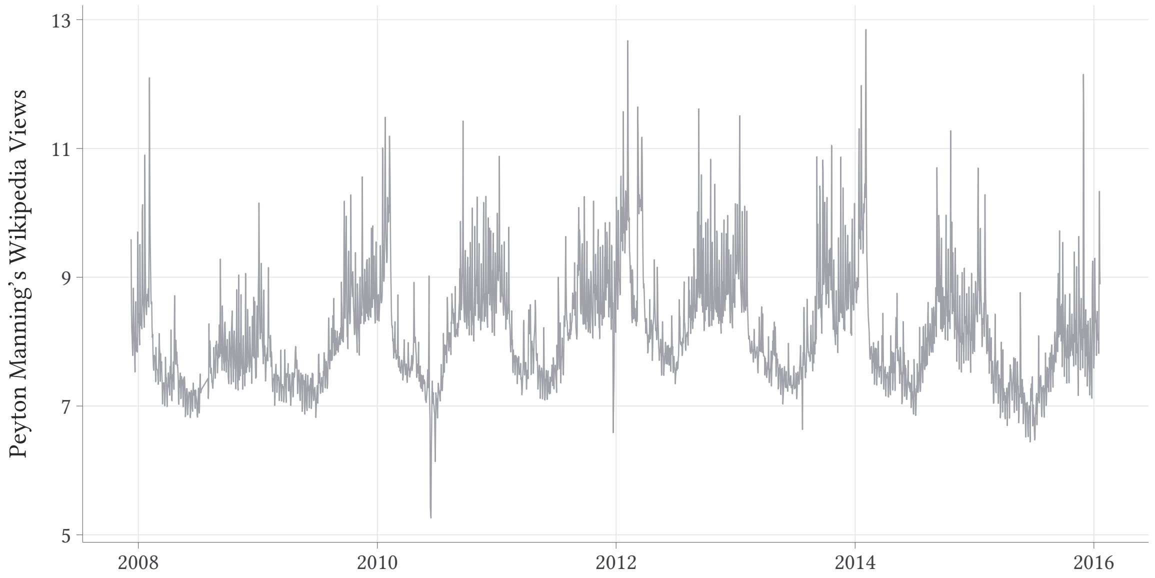
$\hat{\beta}_m$ represent the difference in average y_t for month m relative to the omitted month

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

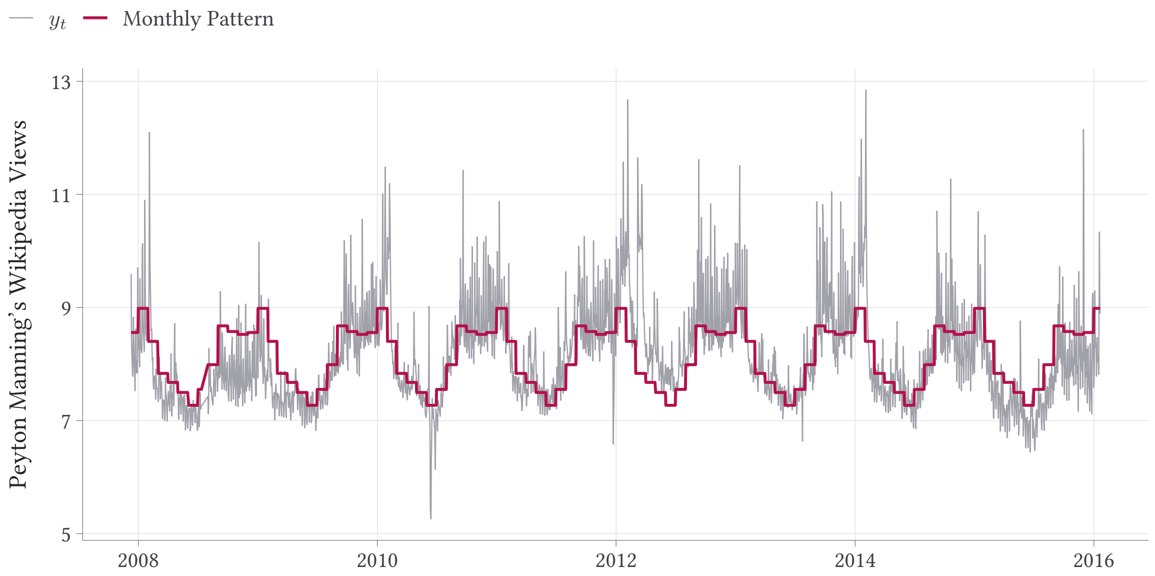
$\rightarrow$ Test the null that $\hat{\beta}_m = 0$

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):

- $\rightarrow$ What is the omitted category?
- $\rightarrow$ In which month are the wikipedia views the highest?
- $\rightarrow$ Are views significantly lower in February than January?
- $\rightarrow$ How would you change this regression model to test if December has signicantly more views than November?

```
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 ***
```

Fourier Series

An alternate to using indicators for seasonal patterns, another popular approach is to use a fourier series

Seasonality is a periodic pattern, repeats after a set amount of time, i.e. in a fixed period

$\rightarrow$ Similar to a taylor expansion, a fourier series is useful at approximating any periodic function

Fourier Series

An alternate to using indicators for seasonal patterns, another popular approach is to use a fourier series

Seasonality is a periodic pattern, repeats after a set amount of time, i.e. in a fixed period

$\rightarrow$ Similar to a taylor expansion, a fourier series is useful at approximating any periodic function

Let t denote the time period and m denote the seasonal period

$\rightarrow$ m is the number of periods in a cycle (e.g. daily data has 365 days in a year)

Fourier Series

$$x_{1,t} = \sin\left(2\pi\frac{t}{m}\right), x_{2,t} = \cos\left(2\pi\frac{t}{m}\right),$$

$$x_{3,t} = \sin\left(2\pi\frac{2t}{m}\right), x_{4,t} = \cos\left(2\pi\frac{2t}{m}\right),$$

$$x_{5,t} = \sin\left(2\pi\frac{3t}{m}\right), x_{6,t} = \cos\left(2\pi\frac{3t}{m}\right),$$

$\rightarrow$ t/m is how far along the period, m, you are

$\rightarrow$ 2π is there to determine how long before completing 1 cycle

Fourier Series

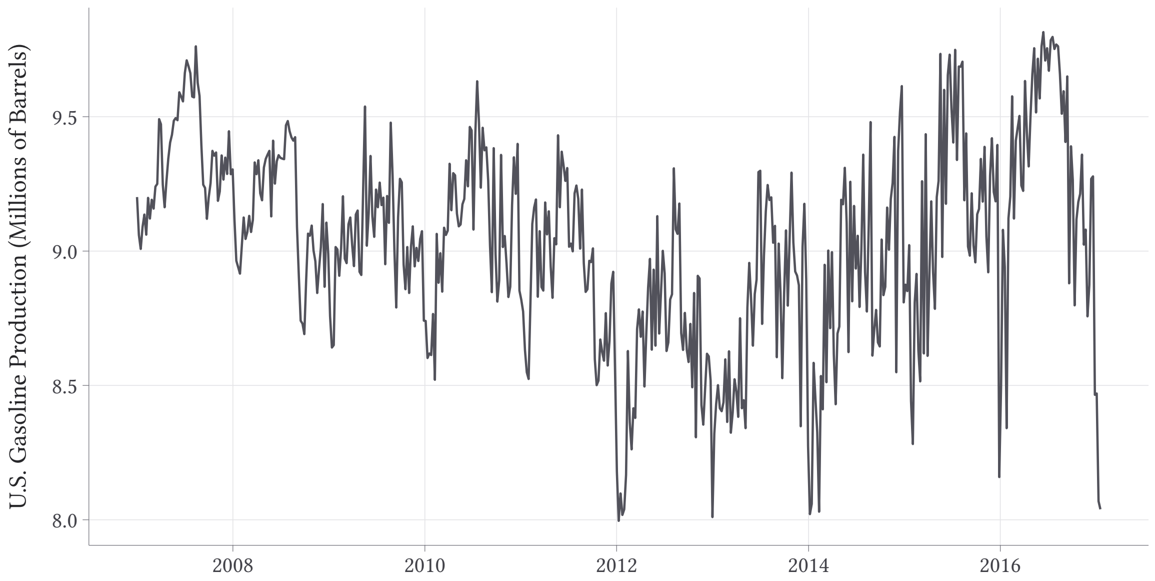
The order K says how many *pairs* of sine and cosine functions to include

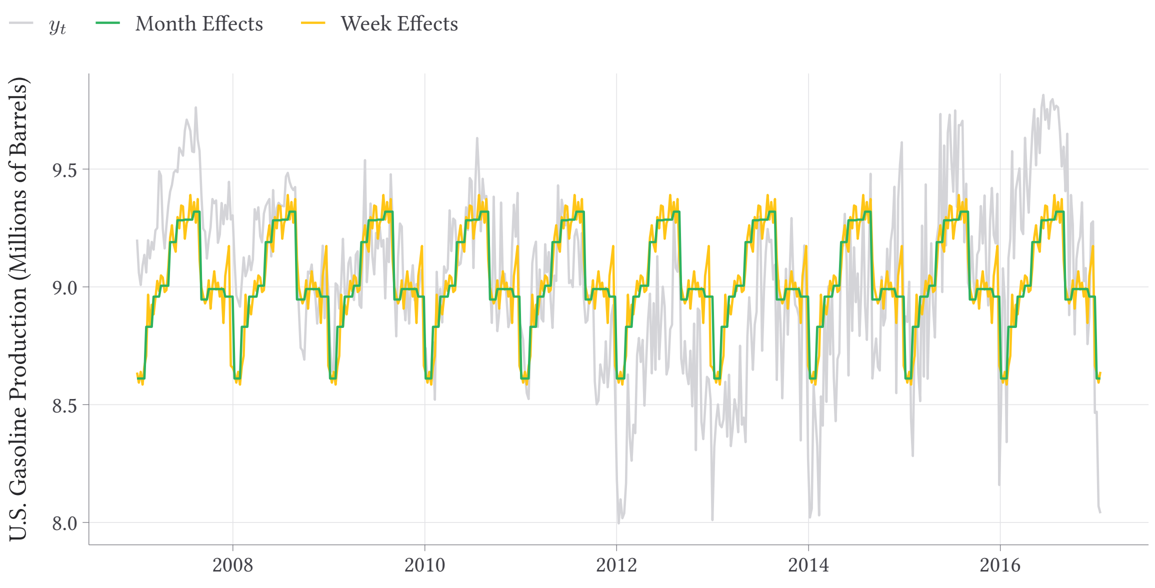
$\rightarrow$ Terms of the form $\sin\left(2\pi \frac{k*t}{m}\right)$ and $\cos\left(2\pi \frac{k*t}{m}\right)$ for $k = 1, \ldots, K$

As K gets larger, you can estimate more and more rapid “swings” in the seasonal pattern

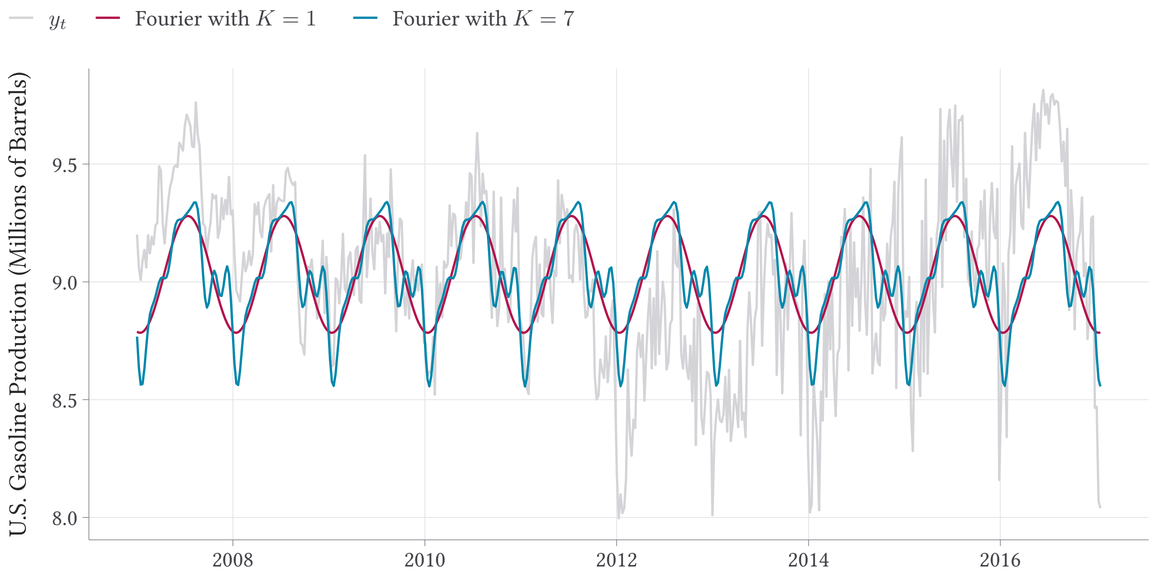
$\rightarrow$ As $K \rightarrow \infty$, you can approximate *any* periodic function

$\rightarrow$ This method is popular because even small K captures seasonality well!


U.S. Gasoline Production (Millions of Barrels)
9.5
9.0
8.5
8.0
2008
2010
2012
2014
2016



y_t
Month Effects
Week Effects
U.S. Gasoline Production (Millions of Barrels)
9.5
9.0
8.5
8.0
2008
2010
2012
2014
2016



y_t
Fourier with $K = 1$
Fourier with $K = 7$
U.S. Gasoline Production (Millions of Barrels)
9.5
9.0
8.5
8.0
2008
2010
2012
2014
2016

Fourier Advantages

A main advantage of a fourier series compared to the indicator approach is that there are typically far fewer parameters to estimate:

$\rightarrow$ $K = 1$ has 2 parameters to fit and does quite well!

$\rightarrow$ versus 12 parameters for monthly indicators

Estimating time-trends

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

$\rightarrow$ 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.

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

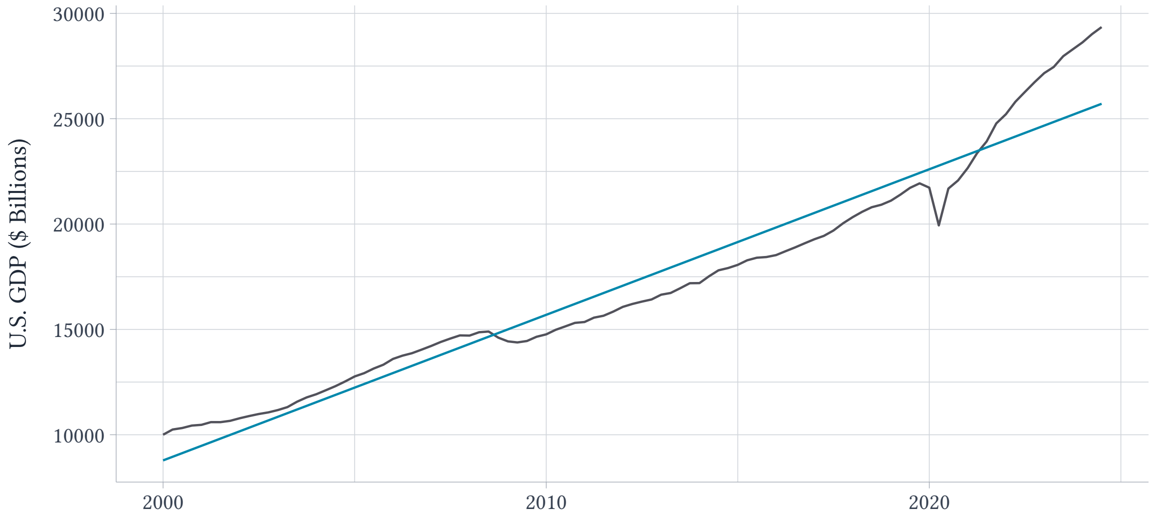
Linear time-trends

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

Estimating linear time trends in R

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

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

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

Two notes:

$\rightarrow$ Note if we have missing observations, that is okay just skip those numbers

$\rightarrow$ t can start at any value (not just 1)

- $\hat{\lambda}$ will be the same, $\hat{\alpha}$ will change

Estimating linear time trends in R

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

$\rightarrow$ This means internally consecutive days look like $t, t+1, t+2, \dots$ like we need!

$\rightarrow$ 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

$\rightarrow$ A 1 unit increase in `date` $==$ the following day

Estimating linear time trends in R

```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 $

$\rightarrow$ Every quarter, $91 * 1.89 = 171.99$ billion $

$\rightarrow$ 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 $T+1, T+2, \dots$

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

$\rightarrow$ 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

$\rightarrow$ More, time-trends tend to plateau (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

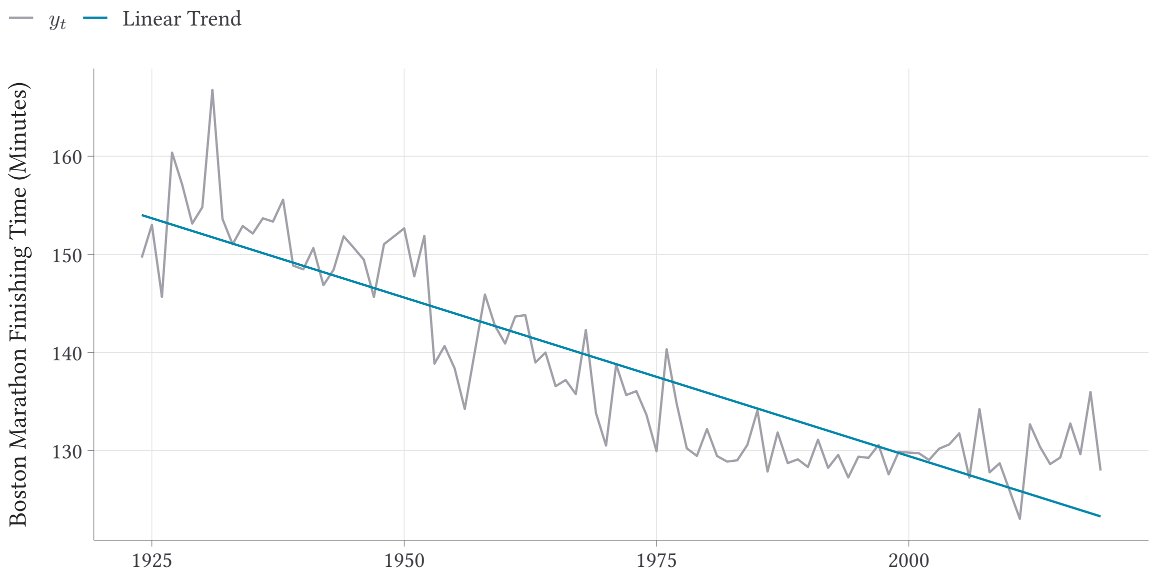
$\rightarrow$ Perhaps the trend changes over time (e.g. improvements slow down)

$\rightarrow$ Especially at risk when the time-series is long

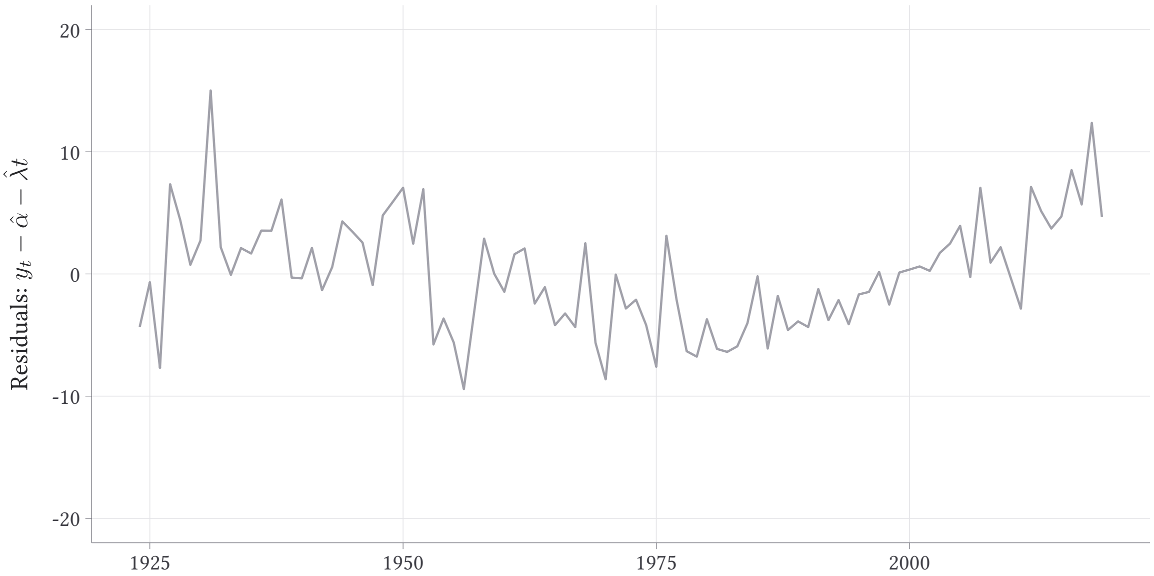
One way to check for this is to look at the difference between $y_t - \hat{y}_t$

$\rightarrow$ 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).

$\rightarrow$ Estimate a separate trend for each epoch

This is called a piecewise linear trends. In our marathon times example:

$\rightarrow$ An initial epoch of small changes in time,

$\rightarrow$ Followed by a steep decline in times,

$\rightarrow$ 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

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

$\rightarrow$ At each point $B_2, \dots, B_L$, the slope changes

$\rightarrow$ $\hat{\lambda}_1$ is the slope of the first epoch

$\rightarrow$ $\hat{\lambda}_2, \dots, \hat{\lambda}_L$ are the difference in slopes from the preivous epoch

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

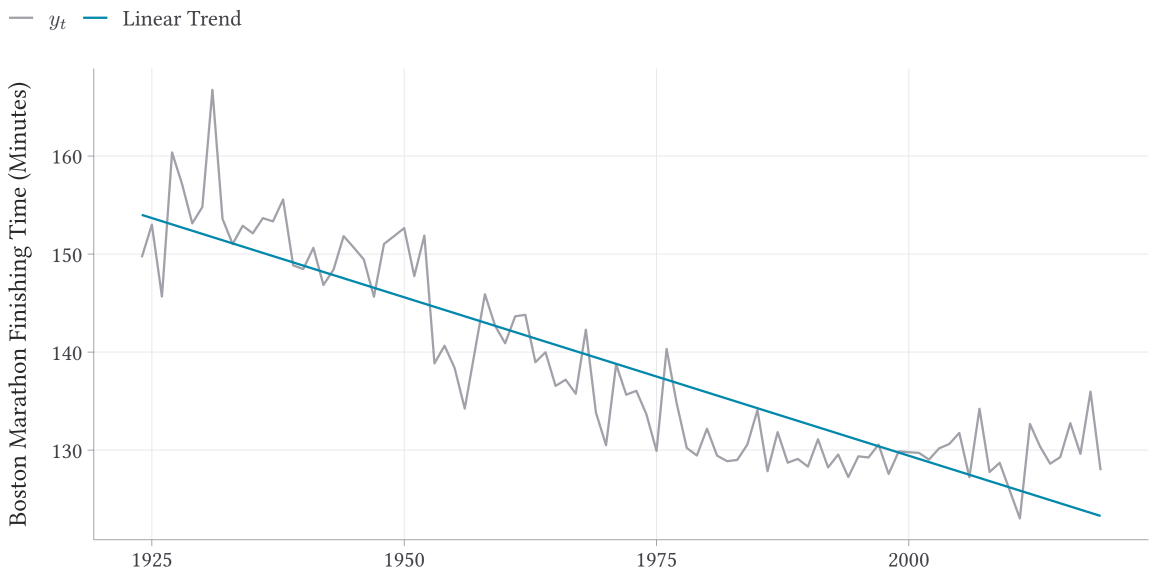
Epoch in R

In R, you can define epoch like:

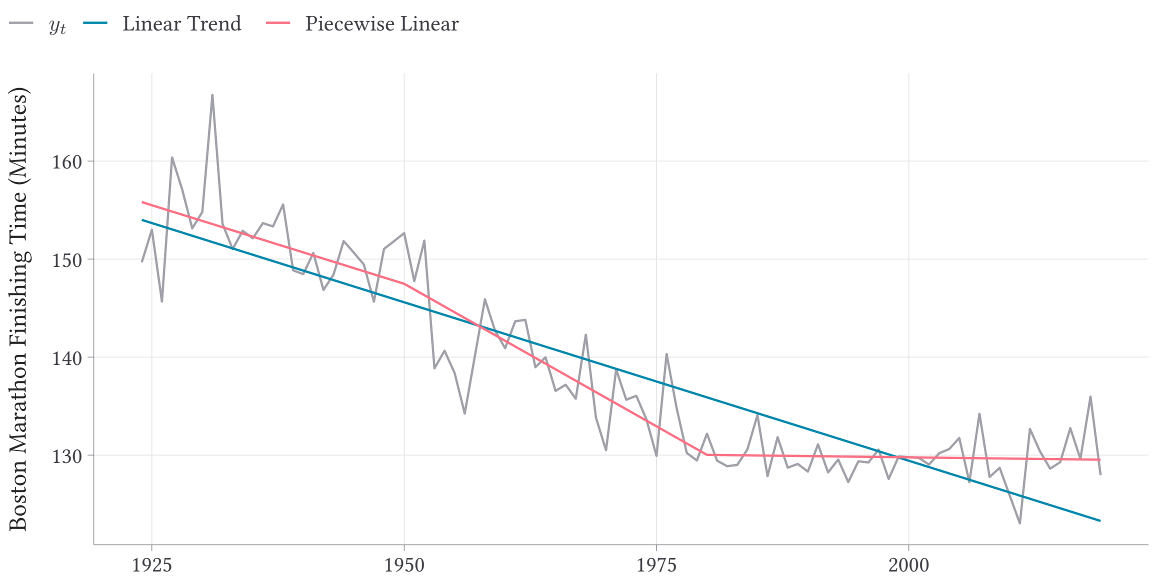
```r
df$trend_1 <- df$year
df$trend_2 <- (df$year - 1950) * (df$year > 1950)
df$trend_3 <- (df$year - 1980) * (df$year > 1980)
```

Then you estimate the piece-wise time trend model with

```r
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

Splines

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

Piece-wise linear models are actually just a specific version of splines!

$\rightarrow$ specifically $p = 1$ (linear), $s = 1$ (connected)

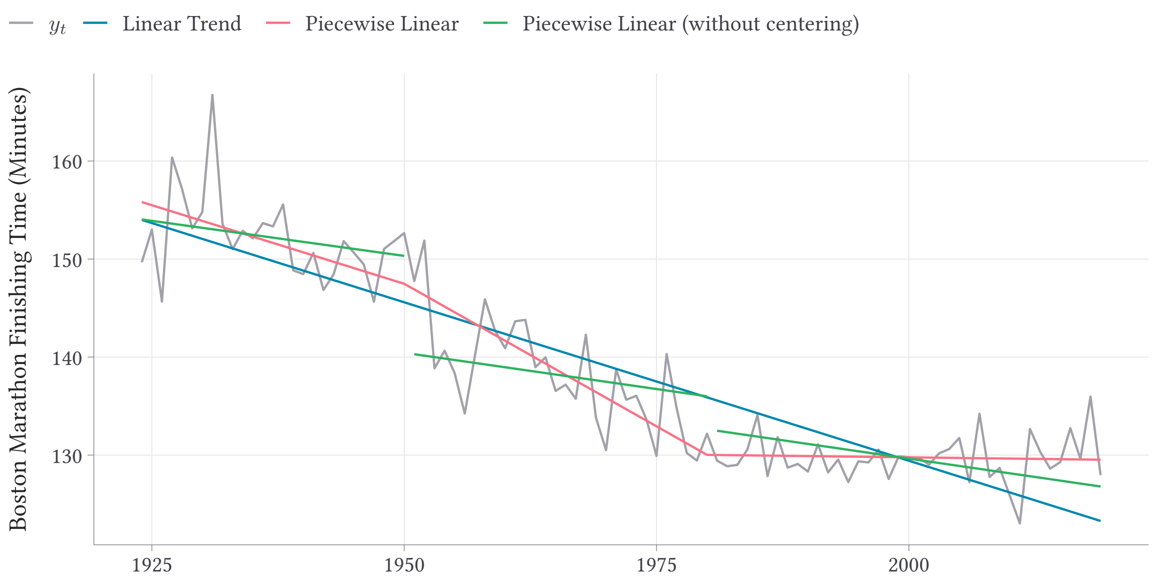
Why we need “Centering”

Say we ran the model without subtracting off B_ℓ:

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

This model would estimate three separate lines, all sharing a common intercept (α)

$\rightarrow$ By centering $t - B_\ell$, we are instead making the lines connect at the knots of B_ℓ



y_t
Linear Trend
Piecewise Linear
Piecewise Linear (without centering)
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

$\rightarrow$ 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 and nice round numbers

Choosing breakpoints

We selected the breakpoints, more or less, visually

$\rightarrow$ 1950 and 1980 looked to be when breaks happen and nice round numbers

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

$\rightarrow$ The goal is to select B_ℓ to do the best job at predicting y_t

$\rightarrow$ Perhaps even select the number of breaks L

Choosing breakpoints

We selected the breakpoints, more or less, visually

$\rightarrow$ 1950 and 1980 looked to be when breaks happen and nice round numbers

Alternatively, we could try and ‘detect’ break-points

$\rightarrow$ The goal is to select B_ℓ to do the best job at predicting y_t

$\rightarrow$ 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

$\rightarrow$ 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

Time-series Regression

Time-series Predictors

- Seasonality
- Fourier Series
- Time-trends
- More flexibly modelling trends
- Step-functions
- Additional covariates

(Review of) Inference on regression forecasts

Flexibly modelling trends

The linear time-trend λt is useful for:

$\rightarrow$ Summarizing the time-series trend succinctly

$\rightarrow$ Useful for forecasting into the future

$\rightarrow$ 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)

$\rightarrow$ 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

$\rightarrow$ Each year has the same predicted value in a given month

Monthly patterns vs. Year-by-month

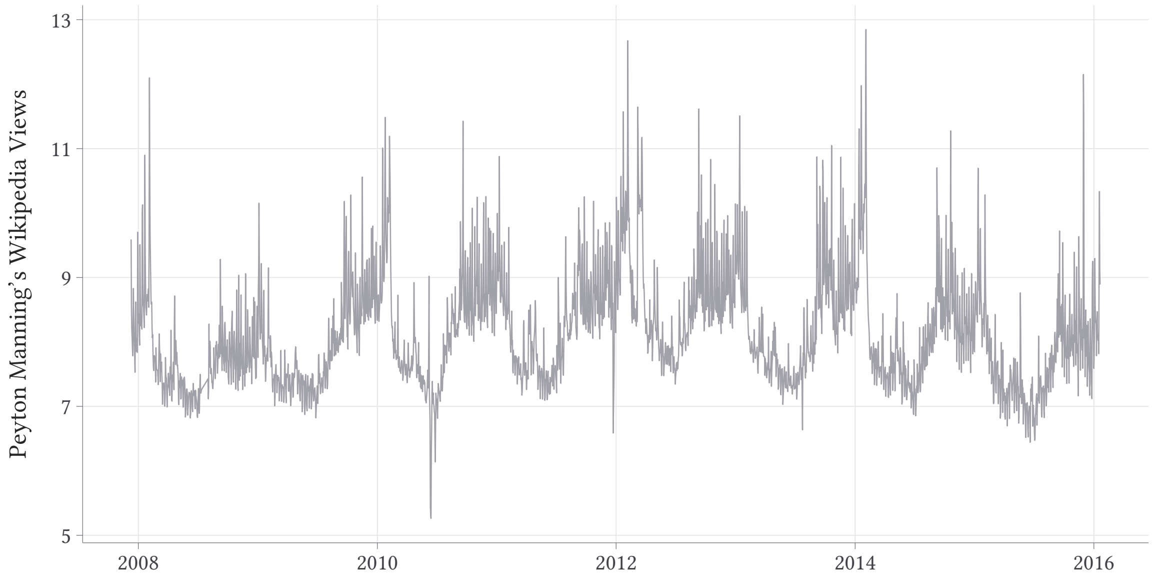
What we have seen so far is how to estimate a recurring monthly pattern

$\rightarrow$ Each year has the same predicted value in a given month

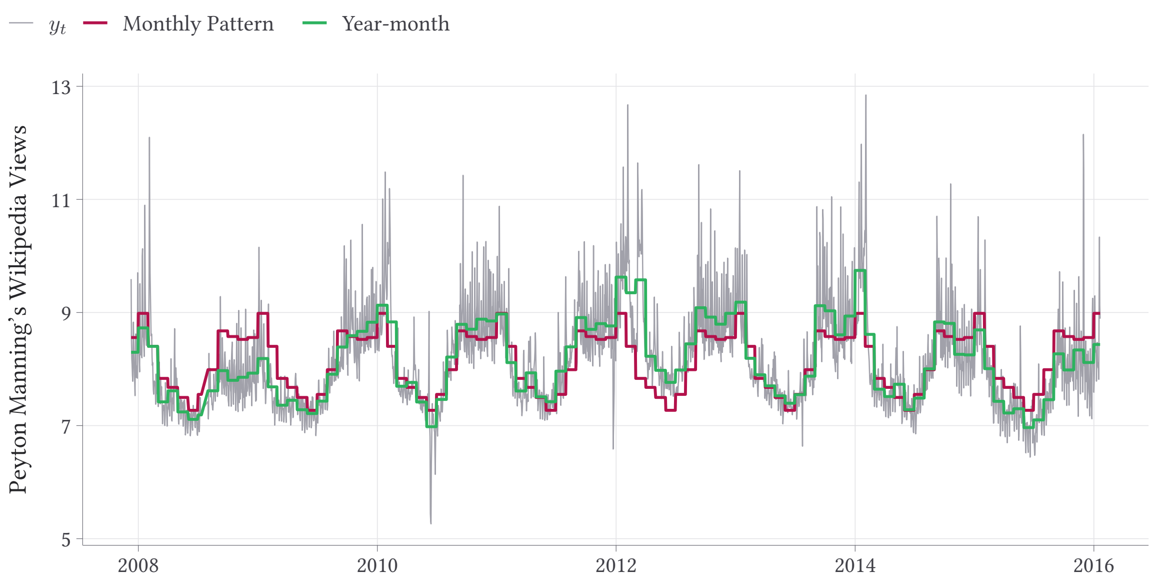
When you have something like weekly or daily data, you can estimate a year-by-month pattern

$\rightarrow$ The estimate for January 2015 is different from January 2016

$\rightarrow$ 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

$\rightarrow$ Monthly patterns are easier to convey and more informative for future forecasting

$\rightarrow$ 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

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

A mix of the two

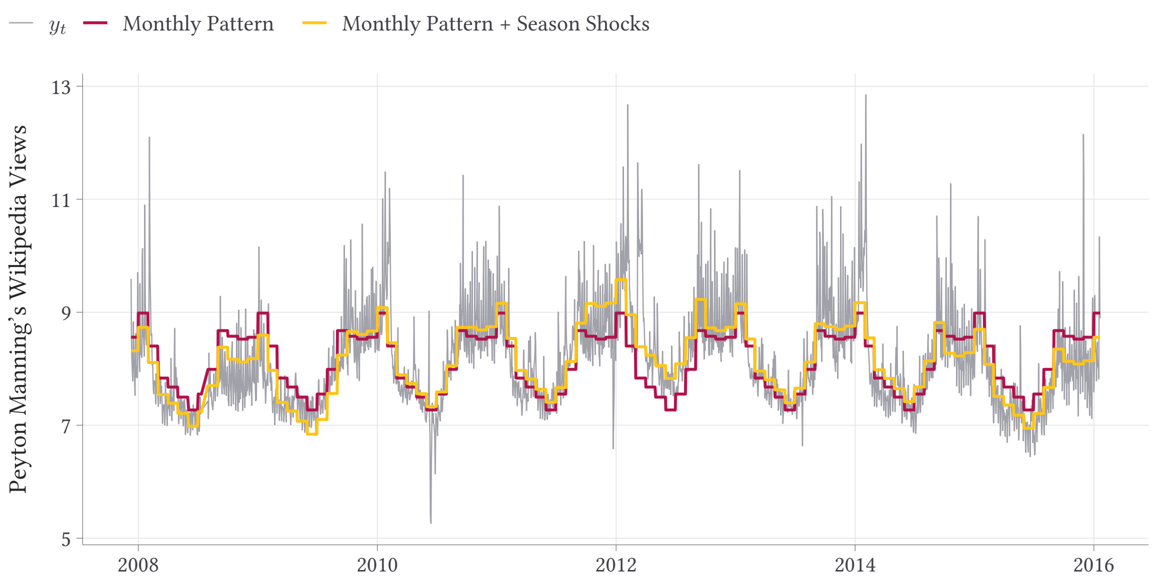
One way to balance between these two approaches is to:

→ 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

Time-series Regression

Time-series Predictors

- Seasonality
- Fourier Series
- Time-trends
- More flexibly modelling trends
- Step-functions
- Additional covariates

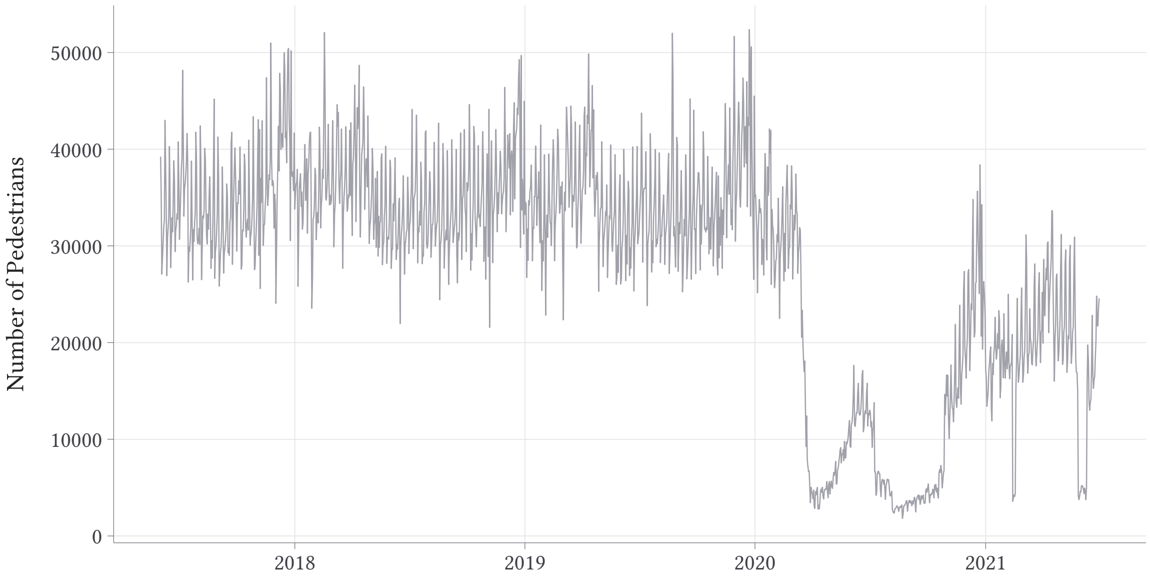
(Review of) Inference on regression forecasts

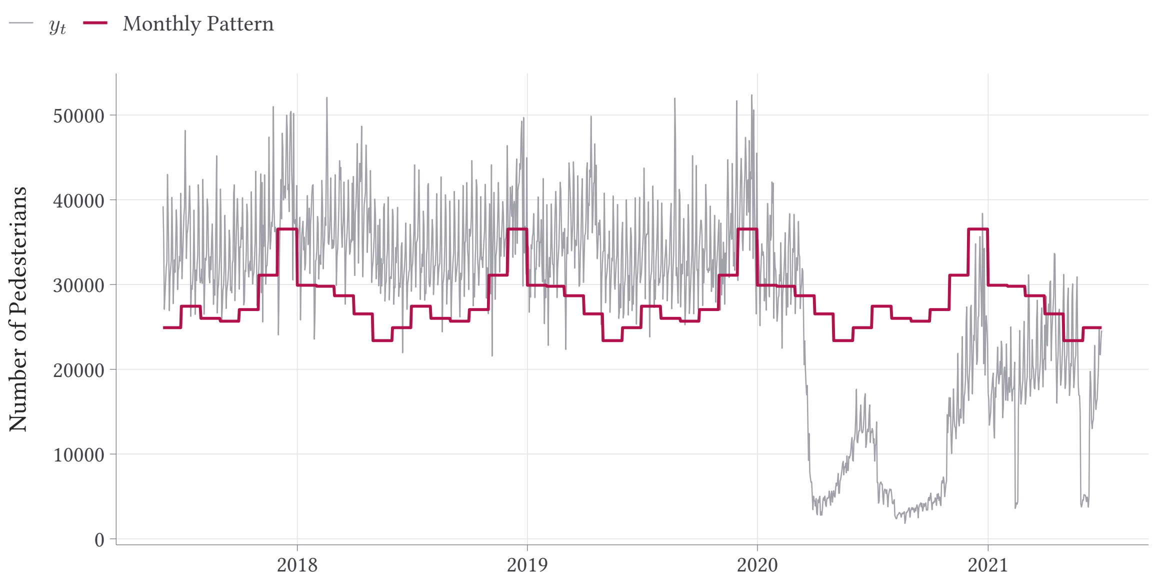
Outliers / Weird Shocks

Sometimes, your dataset will have really weird jumps

$\rightarrow$ 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
0
10000
20000
30000
40000
50000
2018
2019
2020
2021

Dealing with Weird Shocks

For these periods, we can either

$\rightarrow$ (1) drop them from the regression (potentially losing valuable information)

$\rightarrow$ (2) or, add these shocks to our model

Dealing with Weird Shocks

For these periods, we can either

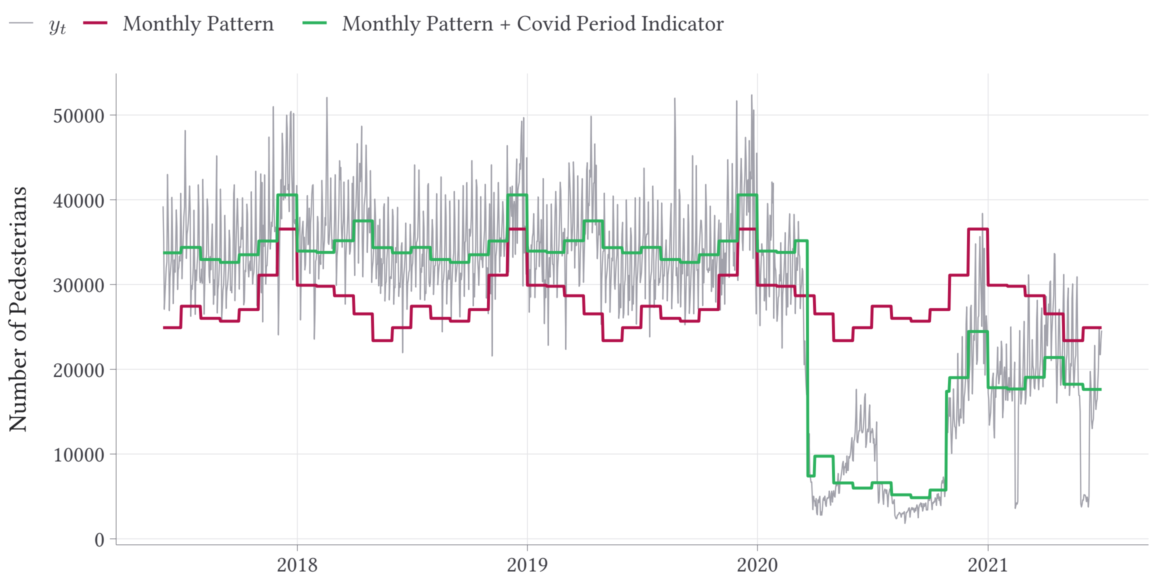
$\rightarrow$ (1) drop them from the regression (potentially losing valuable information)

$\rightarrow$ (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)

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

$\rightarrow$ 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

```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"))

est <- feols(
  ppl ~ i(month) + i(covid_period) + i(post_covid_period),
  data = df, vcov = "hc1"
)
df$ppl_hat <- predict(est)
```

Dealing with Structural Changes

In some cases, we see fundamental changes to the economy

$\rightarrow$ Periods prior to some point have different seasonal patterns and trends

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

$\rightarrow$ Month $\times$ Pre indicators estimate pattern *before* switch

$\rightarrow$ Month $\times$ Post indicators estimate pattern *after* switch

$\rightarrow$ Adding a Post indicator allows the average level to be different before/after

Time-series Regression

Time-series Predictors

Seasonality

Fourier Series

Time-trends

More flexibly modelling trends

Step-functions

Additional covariates

(Review of) Inference on regression forecasts

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:

- $\rightarrow$ Smoothing averages based on time
- $\rightarrow$ Seasonal patterns
- $\rightarrow$ Linear, piecewise, or more flexible time-trends

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

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

Including additional covariates

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

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

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

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

$\rightarrow$ Set the price to what we intend for it to be

$\rightarrow$ And extrapolate the time-series pattern into the future

Time-series Regression

Time-series Predictors

- Seasonality
- Fourier Series
- Time-trends
- More flexibly modelling trends
- Step-functions
- Additional covariates

(Review of) Inference on regression forecasts

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

$\rightarrow$ This is the second half of statistics!

Prediction from regression

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

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

$\rightarrow$ 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

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

Inference on Regression Predictions

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

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

$\rightarrow$ 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

$\rightarrow$ 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 $\text{SE}(\hat{y})$

```r
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

```r
# 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

```r
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
```