# Exponential smoothing: Introduction

## EC 361–001

Prof. Santetti

Spring 2024

# Materials

**Required readings**:

- `Hyndman & Athanasopoulos, ch. 8`
  - sections 8.1—8.2.

# Motivation

# Motivation

After studying benchmark forecasting models and some accuracy measures, it is time to move on to **more interesting** forecasting techniques.

One of them is **exponential smoothing**.

The key idea behind this method is that it generates forecasts that are **weighted averages** of past observations, with the weights **decaying exponentially** as the observations get *older*.

- Also, **more recent** observation get *higher* associated weights.

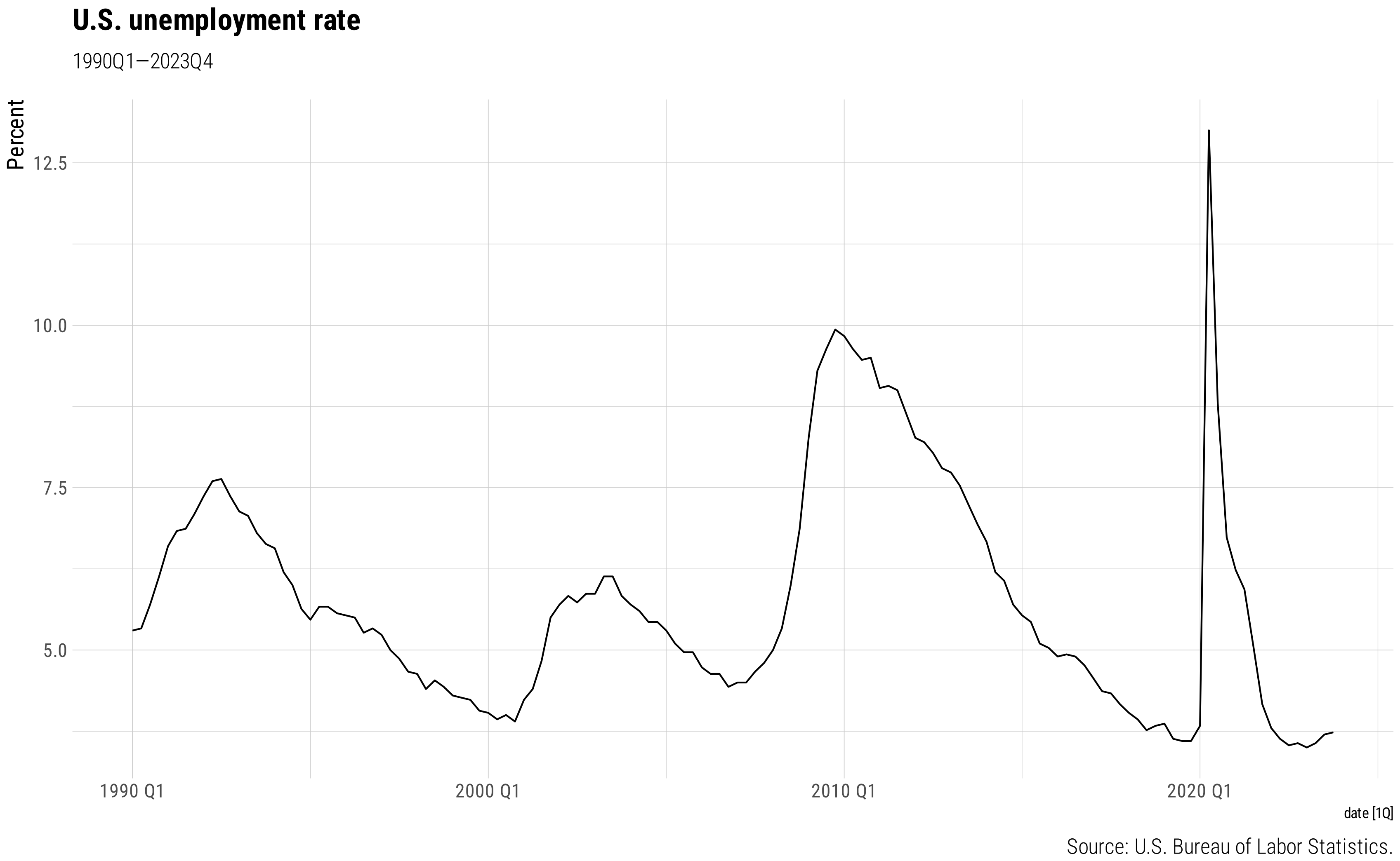
# Simple exponential smoothing

# Simple exponential smoothing

The starting point when studying exponential smoothing methods is the so-called **Simple Exponential Smoothing** (SES).

This method is well-suited for time series with **no** apparent **trend** or **seasonal** pattern.

# Simple exponential smoothing



# Simple exponential smoothing

The simple exponential smoothing method can be thought of as a **midpoint** between two extremes:

- The **mean**
- And the **naïve** methods.

Recall that the **mean** method fits the data and predicts future values according to the the **average** of the series.

- Thus, all observations have the same **weight**.

On the other hand, the **naïve** method assumes that the **most recent** observation is the *only* important one, and all previous observations provide *no information for the future*.

- Thus, all the **weight** is given to the *most recent* observation.

# Simple exponential smoothing

While these two benchmark methods have their logic, what simple exponential smoothing proposes is a method where **larger weights** are given to **most recent** observations, while more distant observations are to the present will be given **less importance**.

This way, the model is fitted to the data and forecasts are produced using **weighted averages**, where the weights decrease *exponentially* as observations come from further in the past (i.e., the *smallest weights* are associated with the *oldest* observations).

More formally,

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha) y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots$$

where $\alpha$ is called the **smoothing parameter**, lying between 0 and 1.

# Simple exponential smoothing

| | $\alpha = 0.2$ | $\alpha = 0.4$ | $\alpha = 0.6$ | $\alpha = 0.8$ |
|---|---|---|---|---|
| $y_T$ | 0.2000 | 0.4000 | 0.6000 | 0.8000 |
| $y_{T-1}$ | 0.1600 | 0.2400 | 0.2400 | 0.1600 |
| $y_{T-2}$ | 0.1280 | 0.1440 | 0.0960 | 0.0320 |
| $y_{T-3}$ | 0.1024 | 0.0864 | 0.0384 | 0.0064 |
| $y_{T-4}$ | 0.0819 | 0.0518 | 0.0154 | 0.0013 |
| $y_{T-5}$ | 0.0655 | 0.0311 | 0.0061 | 0.0003 |

- The smaller $\alpha$ is, more weight is given to observations from the *more distant past*.
- The larger $\alpha$ is, more weight is given to the *more recent observations*.
- What happens when $\alpha = 1$?

# Simple exponential smoothing

Each exponential smoothing method we will study can be represented by what we call its **component form**.

A method's component form comprises a set of **equations** illustrating the relevant components of the model.

In the case of **simple exponential smoothing**, the component form has **two pieces**:

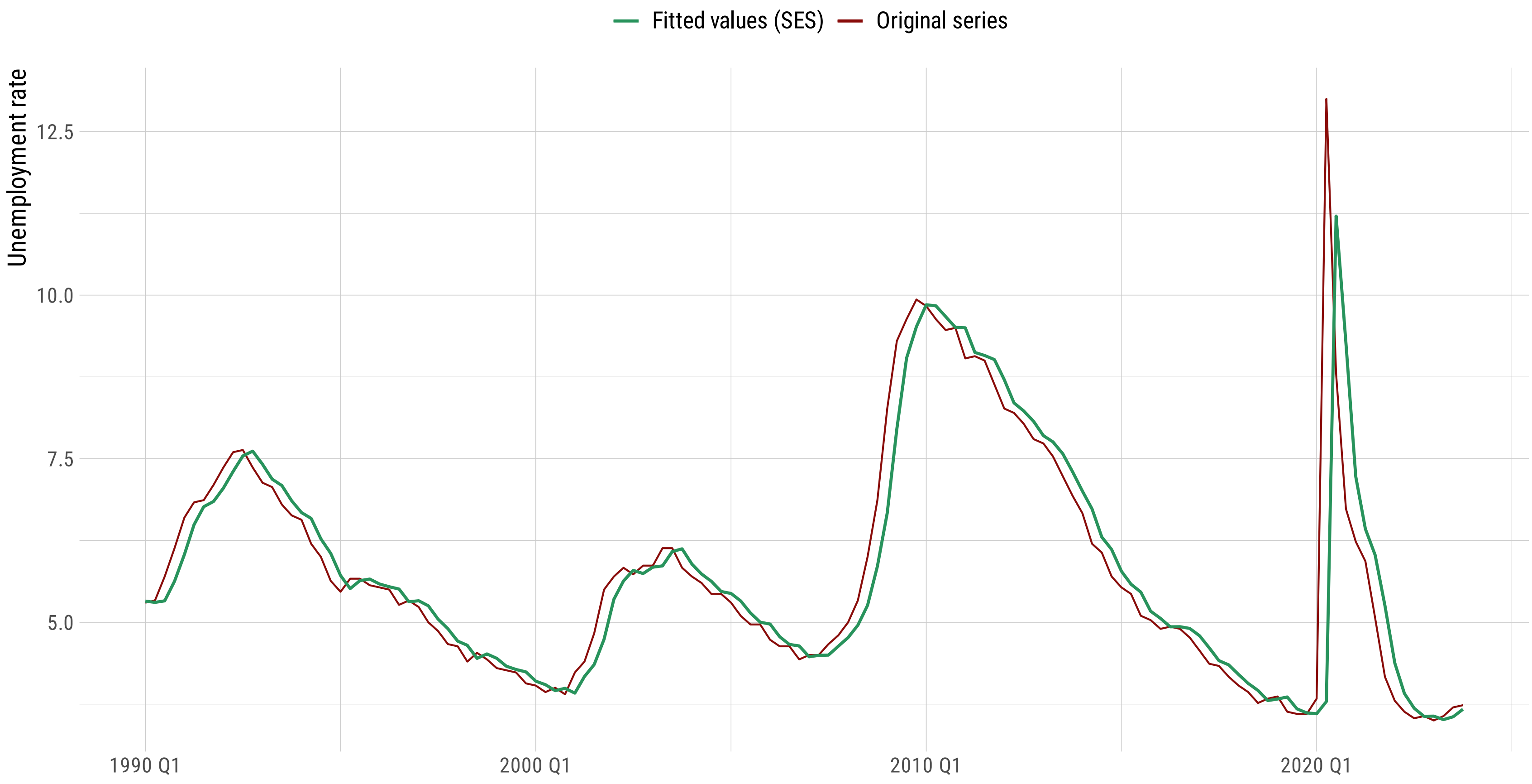
$$\text{Forecast equation: } \hat{y}_{t+h|t} = \ell_t$$

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

where $\ell_t$ is the **level** (or the smoothed value) of the series at time $t$.

# Simple exponential smoothing

Illustrating the **smoothing equation**:



# Simple exponential smoothing

```
unemp_ets_fit ← unemp_ts ▷
  model(SES = ETS(unrate ~ error("A") + trend("N") + season("N"))) # "A" = additive; "N" = none.

unemp_ets_fit ▷
  augment()
```

```
#> # A tsibble: 136 x 6 [1Q]
#> # Key:       .model [1]
#>    .model    date unrate .fitted  .resid  .innov
#>    <chr>    <qtr>  <dbl>   <dbl>   <dbl>   <dbl>
#>  1 SES    1990 Q1   5.3     5.32 -0.0239 -0.0239
#>  2 SES    1990 Q2   5.33    5.30  0.0287  0.0287
#>  3 SES    1990 Q3   5.7     5.33  0.372   0.372
#>  4 SES    1990 Q4   6.13    5.63  0.506   0.506
#>  5 SES    1991 Q1   6.6     6.03  0.565   0.565
#>  6 SES    1991 Q2   6.83    6.49  0.343   0.343
#>  7 SES    1991 Q3   6.87    6.77  0.100   0.100
#>  8 SES    1991 Q4   7.1     6.85  0.253   0.253
#>  9 SES    1992 Q1   7.37    7.05  0.316   0.316
#> 10 SES    1992 Q2   7.6     7.31  0.295   0.295
#> # ℹ 126 more rows
```

# Simple exponential smoothing

```r
unemp_ets_fit <- unemp_ts |>
  model(SES = ETS(unrate ~ error("A") + trend("N") + season("N"))) # "A" = additive; "N" = none.

unemp_ets_fit |>
  augment()
```

```
#> # A tsibble: 136 x 6 [1Q]
#> # Key:       .model [1]
#>    .model    date unrate .fitted  .resid  .innov
#>    <chr>    <qtr>  <dbl>   <dbl>   <dbl>   <dbl>
#>  1 SES    1990 Q1   5.3     5.32 -0.0239 -0.0239
#>  2 SES    1990 Q2   5.33    5.30  0.0287  0.0287
#>  3 SES    1990 Q3   5.7     5.33  0.372   0.372
#>  4 SES    1990 Q4   6.13    5.63  0.506   0.506
#>  5 SES    1991 Q1   6.6     6.03  0.565   0.565
#>  6 SES    1991 Q2   6.83    6.49  0.343   0.343
#>  7 SES    1991 Q3   6.87    6.77  0.100   0.100
#>  8 SES    1991 Q4   7.1     6.85  0.253   0.253
#>  9 SES    1992 Q1   7.37    7.05  0.316   0.316
#> 10 SES    1992 Q2   7.6     7.31  0.295   0.295
#> # ℹ 126 more rows
```

# Simple exponential smoothing

In order to **fit** an exponential smoothing model, we need to define **two values**:

- the smoothing parameter $\alpha$;
- The **initial value** for the level term $\ell$, $\ell_0$.

While these can be arbitrarily chosen, a better method is to obtain these by **minimizing** the Sum of Squared Residuals (SSR):

$$\text{SSR} = \sum_{t=1}^{T} (y_t - \hat{y}_{t|t-1})^2 = \sum_{t=1}^{T} e_t^2$$

Different from a **linear regression** context, this method involves a non-linear optimization, which requires **computational methods** for its solution.

# Simple exponential smoothing

```r
unemp_ets_fit |>
  report()
```

```
#> Series: unrate
#> Model: ETS(A,N,N)
#>   Smoothing parameters:
#>     alpha = 0.8055464
#>
#>   Initial states:
#>      l[0]
#>  5.323883
#>
#>   sigma^2:  0.8497
#>
#>      AIC     AICc      BIC
#> 649.9517 650.1336 658.6897
```

```r
unemp_ets_fit |>
  augment() |>
  head(7)
```

```
#> # A tsibble: 7 x 6 [1Q]
#> # Key:       .model [1]
#>   .model    date unrate .fitted  .resid  .innov
#>   <chr>    <qtr>  <dbl>   <dbl>   <dbl>   <dbl>
#> 1 SES    1990 Q1   5.3     5.32 -0.0239 -0.0239
#> 2 SES    1990 Q2   5.33    5.30  0.0287  0.0287
#> 3 SES    1990 Q3   5.7     5.33  0.372   0.372
#> 4 SES    1990 Q4   6.13    5.63  0.506   0.506
#> 5 SES    1991 Q1   6.6     6.03  0.565   0.565
#> 6 SES    1991 Q2   6.83    6.49  0.343   0.343
#> 7 SES    1991 Q3   6.87    6.77  0.100   0.100
```

# Simple exponential smoothing

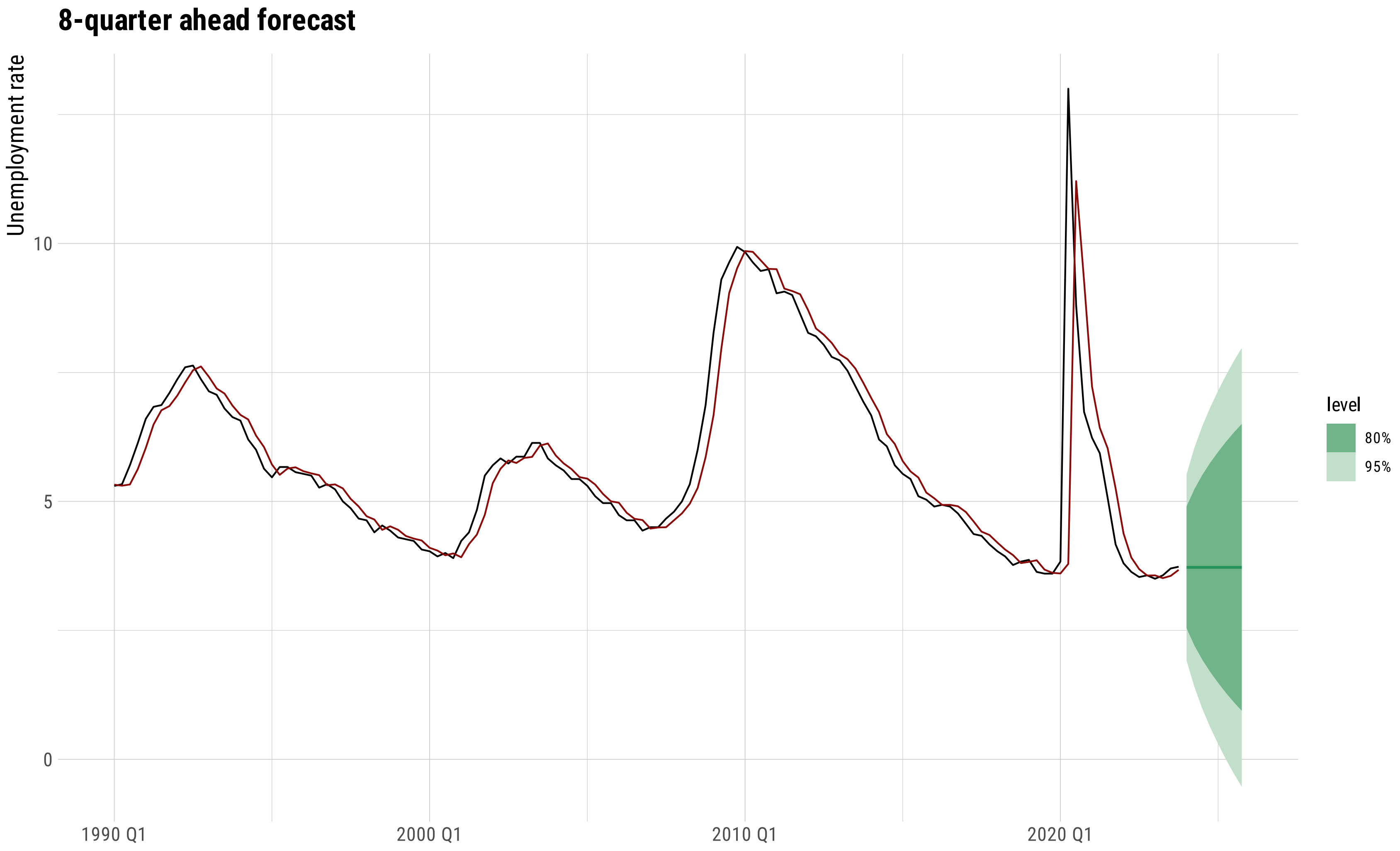
After the model has been fit, forecasts can be produced.

Simple exponential smoothing delivers **flat** forecasts.

In other words, all forecasts take the **same value**, equal to the last level component.

This should *make sense*, since it assumes **no trend** and **no seasonality**.

# Simple exponential smoothing



# Exponential smoothing with trend

# Exponential smoothing with trend

The simple exponential smoothing method is a good **starting point** when learning this technique.

But usually we deal with series that show a **trend** over time.

Thus, we need to further **build up** from this baseline method in order to incorporate other **features** the time series may have.

When considering the existence of a **trend**, the **component form** of exponential smoothing becomes:

$$\text{Forecast equation:}\ \ \hat{y}_{t+h|t} = \ell_t + hb_t$$

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

$$\text{Trend equation:}\ \ b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$$

# Exponential smoothing with trend

$$\text{Forecast equation: } \hat{y}_{t+h|t} = \ell_t + hb_t$$

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

$$\text{Trend equation: } b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$$

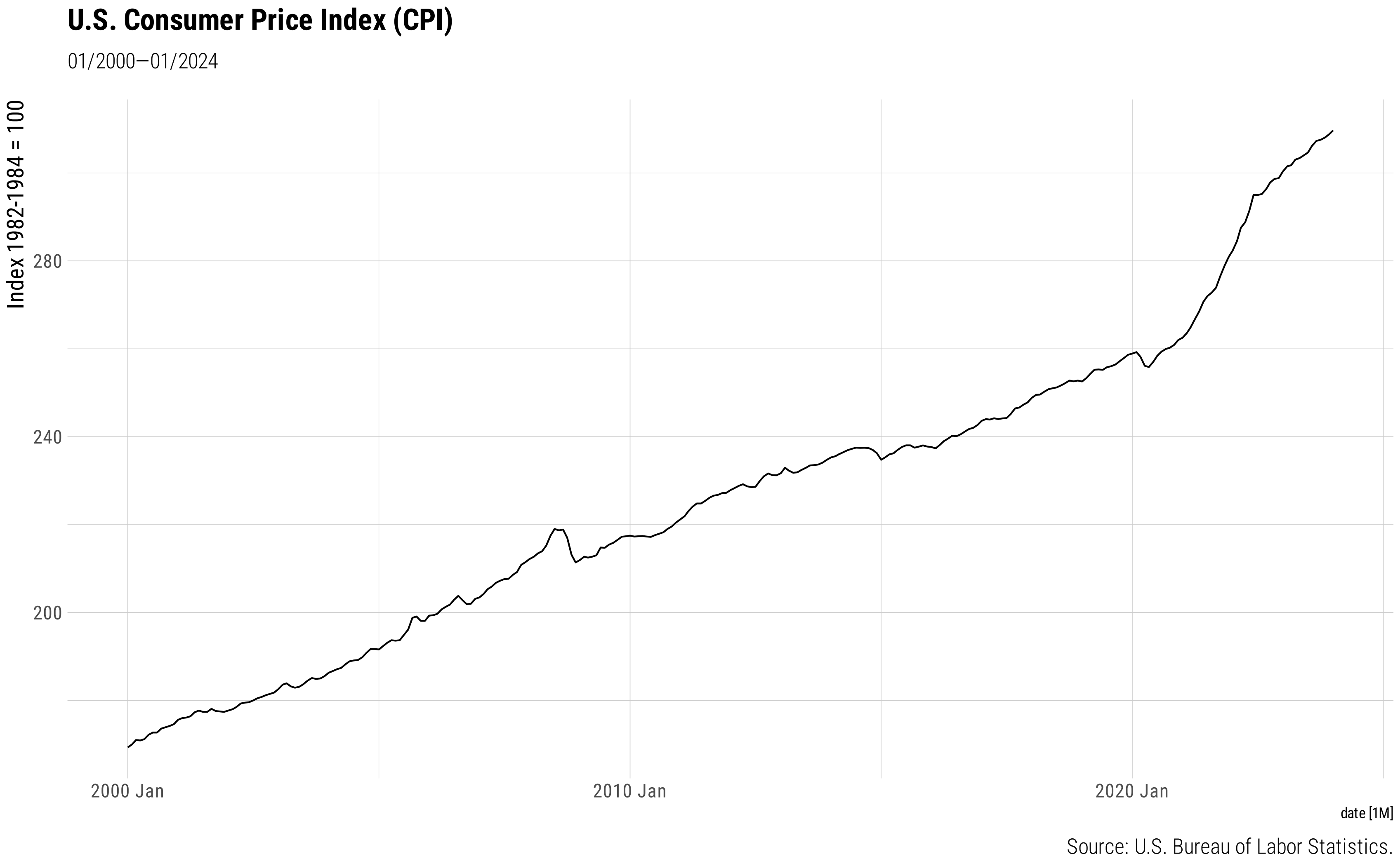
A few new terms:

- $b_t$ denotes an estimate of the trend (slope) of the series at time $t$;
- $\beta^*$ is the smoothing parameter for the trend (just like $\alpha$ is for the level).

Now, forecasts are **no longer flat**.

The $h$-step-ahead forecast is equal to the *last* estimated **level** plus $h$ times the *last* estimated **trend** value.

# Exponential smoothing with trend



# Exponential smoothing with trend

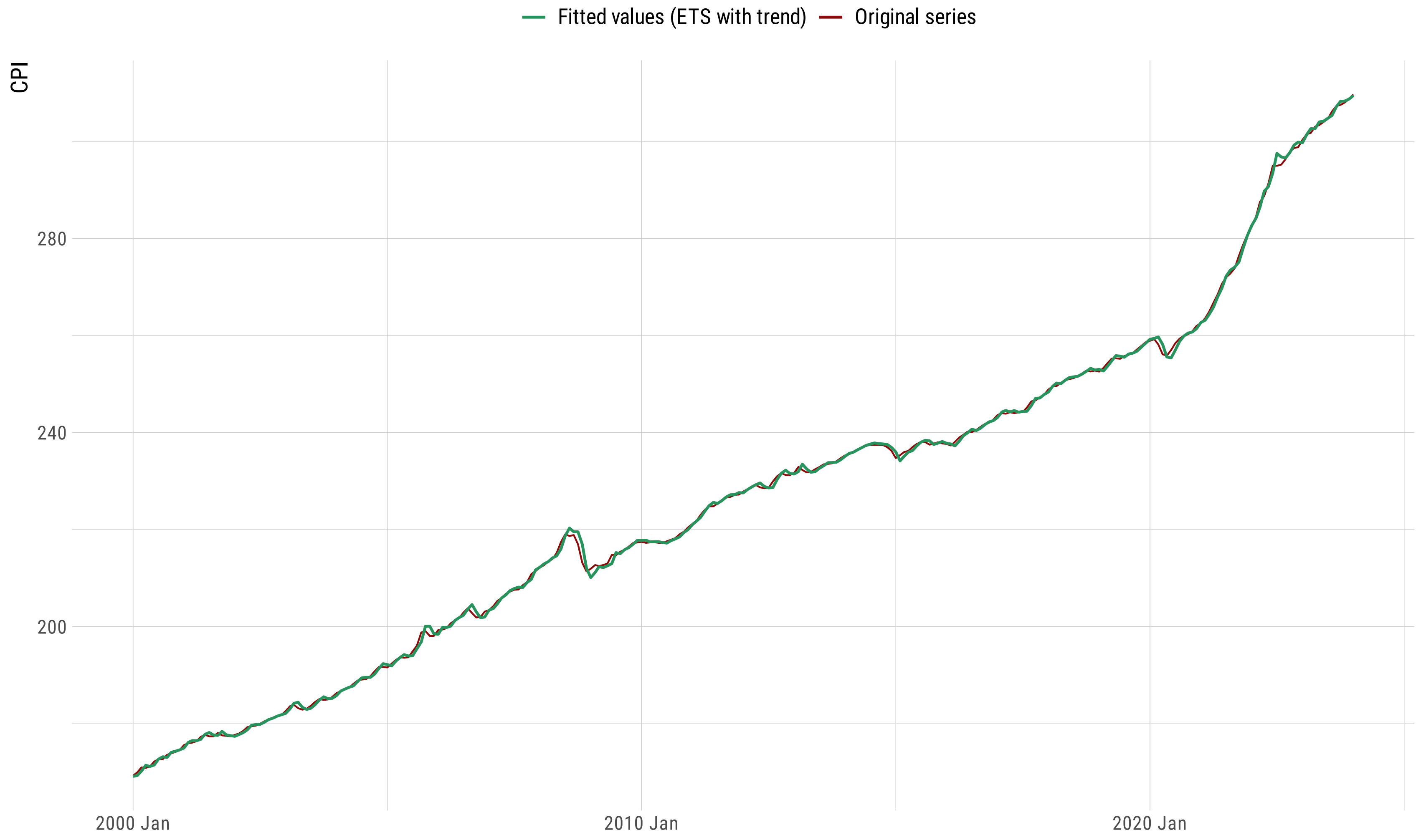
```r
cpi_ets_trend_fit <- cpi_ts |>
  model(ETS_Trend = ETS(cpi ~ error("A") +
                            trend("A") +
                            season("N")))

cpi_ets_trend_fit |>
  augment() |>
  head(5)
```

```
#> # A tsibble: 5 x 6 [1M]
#> # Key:       .model [1]
#>   .model        date   cpi .fitted   .resid   .innov
#>   <chr>        <mth> <dbl>   <dbl>    <dbl>    <dbl>
#> 1 ETS_Trend 2000 Jan  169.    169.  0.157    0.157
#> 2 ETS_Trend 2000 Feb  170     169.  0.651    0.651
#> 3 ETS_Trend 2000 Mar  171     170.  0.771    0.771
#> 4 ETS_Trend 2000 Apr  171.    171. -0.543   -0.543
#> 5 ETS_Trend 2000 May  171.    171.  0.00752  0.00752
```

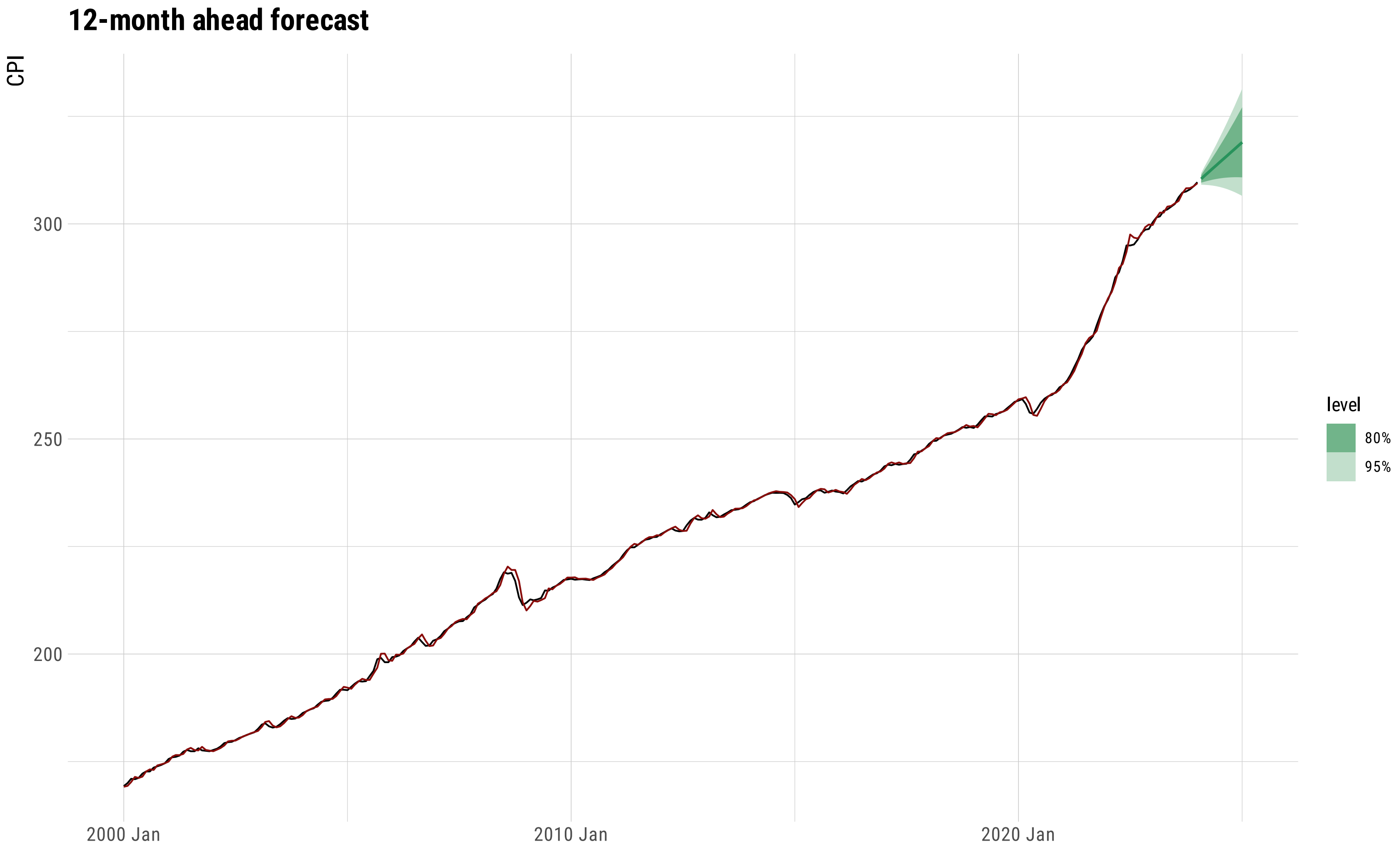
```r
cpi_ets_trend_fit |>
  report()
```

```
#> Series: cpi
#> Model: ETS(A,A,N)
#>   Smoothing parameters:
#>     alpha = 0.9999
#>     beta  = 0.2770026
#>
#>   Initial states:
#>      l[0]        b[0]
#>  169.1376 0.005344676
#>
#>   sigma^2:  0.4628
#>
#>      AIC     AICc      BIC
#> 1420.906 1421.118 1439.238
```

# Exponential smoothing with trend



# Exponential smoothing with trend



# Damped trend methods

# Damped trend methods

The previous method produces forecasts with a **constant trend** that either increases or decreases **indefinitely** into the future.

Many times, especially for longer forecast horizons, that may be **too extreme**.

Thus, there are methods that can "*dampen*" the trend to a flat line over longer forecast horizons.

Now our **component form** includes a **damping parameter**, $\phi$:

$$\text{Forecast equation: } \hat{y}_{t+h|t} = \ell_t + (\phi + \phi^2 + \cdots + \phi^h) b_t$$

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

$$\text{Trend equation: } b_t = \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)\phi b_{t-1}$$

If $\phi = 1$, we are back to the previous trend method.

# Damped trend methods

```r
cpi_ets_trend_damped_fit <- cpi_ts |>
  model(ETS_Trend_Damped = ETS(cpi ~ error("A") + trend("Ad") + season("N")))
```

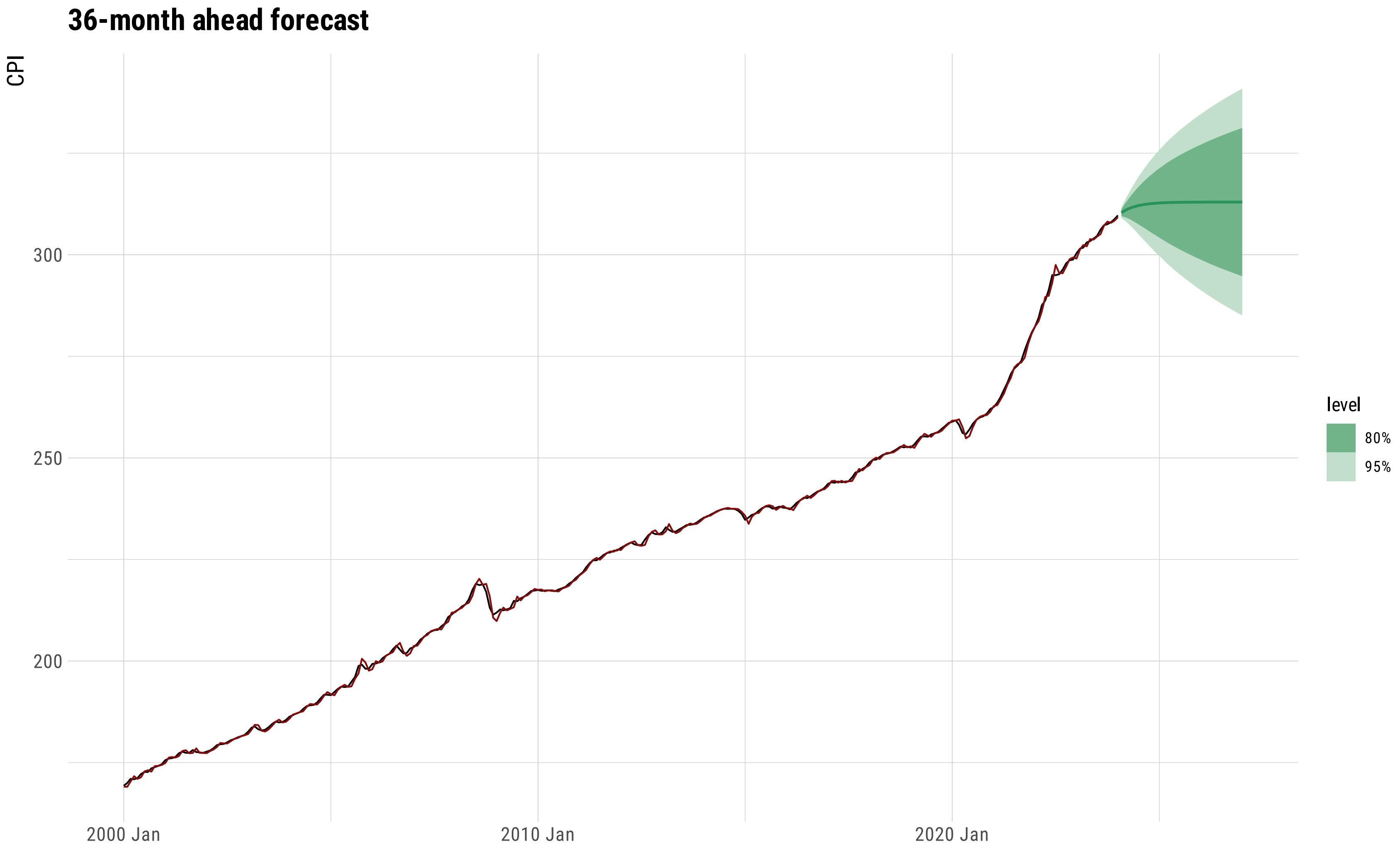
```r
cpi_ets_trend_damped_fit |>
  augment() |>
  head(5)
```

```
#> # A tsibble: 5 x 6 [1M]
#> # Key:       .model [1]
#>   .model                 date   cpi .fitted .resid .innov
#>   <chr>                 <mth> <dbl>   <dbl>  <dbl>  <dbl>
#> 1 ETS_Trend_Damped 2000 Jan  169.    169.  0.240  0.240
#> 2 ETS_Trend_Damped 2000 Feb  170     169.  0.946  0.946
#> 3 ETS_Trend_Damped 2000 Mar  171     170.  0.646  0.646
#> 4 ETS_Trend_Damped 2000 Apr  171.    172. -0.760 -0.760
#> 5 ETS_Trend_Damped 2000 May  171.    171.  0.215  0.215
```

```r
cpi_ets_trend_damped_fit |>
  report()
```

```
#> Series: cpi
#> Model: ETS(A,Ad,N)
#>   Smoothing parameters:
#>     alpha = 0.9999
#>     beta  = 0.728393
#>     phi   = 0.8000002
#>
#>   Initial states:
#>      l[0]       b[0]
#>  169.5424 -0.6028984
#>
#>   sigma^2:  0.4398
#>
#>      AIC     AICc      BIC
#> 1407.154 1407.452 1429.153
```

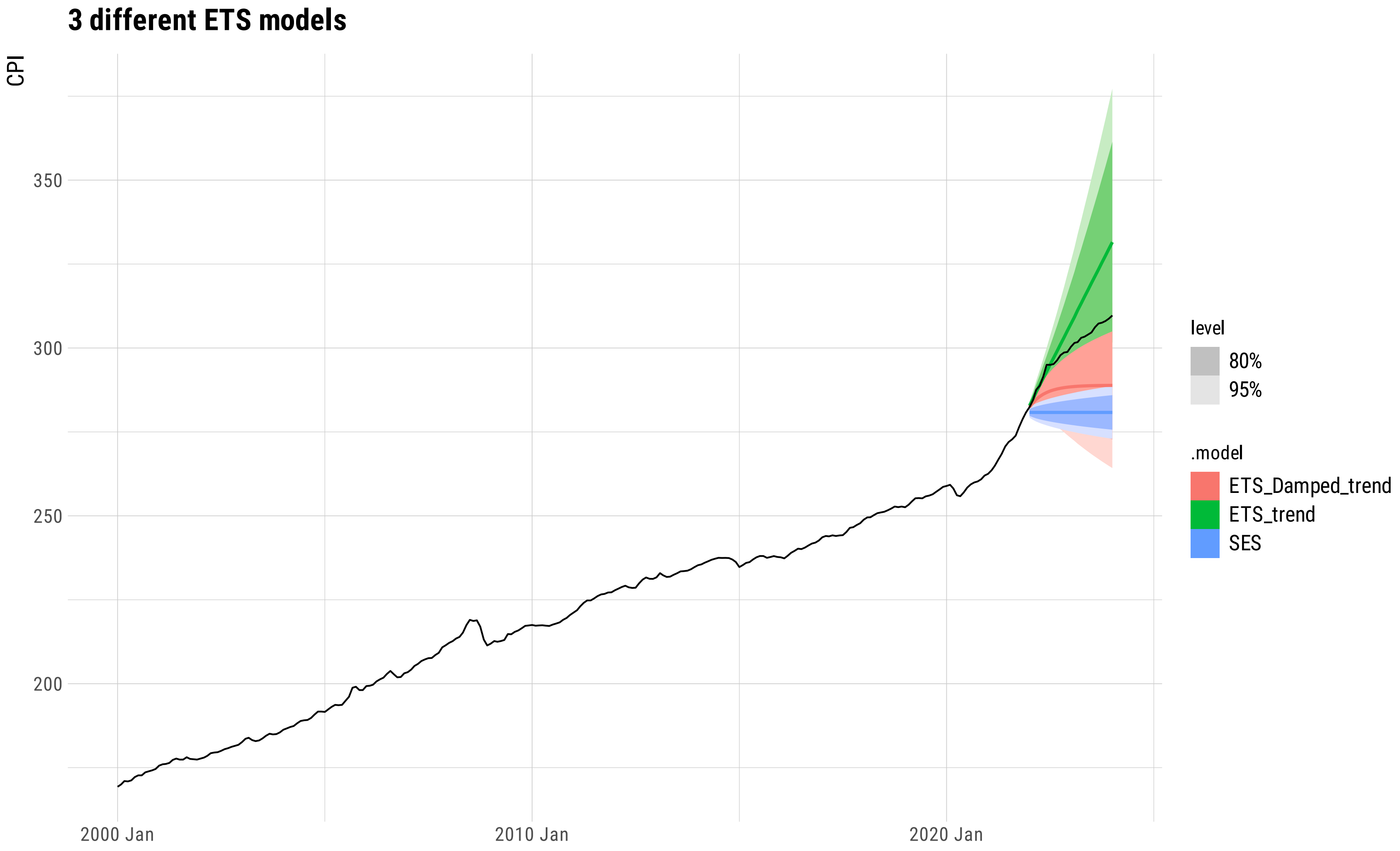
# Damped trend methods



# All together...

```
cpi_train ← cpi_ts ▷
  filter_index(. ~ "2021-12-01")

cpi_ets_models ← cpi_train ▷
  model(SES = ETS(cpi ~ error("A") + trend("N") + season("N")),
        ETS_trend = ETS(cpi ~ error("A") + trend("A") + season("N")),
        ETS_Damped_trend = ETS(cpi ~ error("A") + trend("Ad") + season("N")))

cpi_models_fc ← cpi_ets_models ▷
  forecast(h = 25)
```

# All together...



# All together...

```
cpi_models_fc ▷
  accuracy(cpi_ts) ▷
  select(.model, MAE, RMSE, MAPE, MASE)
```

```
#> # A tibble: 3 × 5
#>   .model             MAE  RMSE  MAPE  MASE
#>   <chr>            <dbl> <dbl> <dbl> <dbl>
#> 1 ETS_Damped_trend 11.5   13.0  3.80  2.37
#> 2 ETS_trend         8.33  11.0  2.73  1.71
#> 3 SES              18.3   19.8  6.06  3.77
```

# Next time: More exponential smoothing models