Forecasting residual analysis

EC 361–001

Prof. Santetti
Spring 2024

Materials

Required readings:

- `Hyndman & Athanasopoulos, ch. 6`
    - sections 6.3—6.4.

Motivation

Motivation

After introducing the four **benchmark** forecasting models, we will focus on what is **left over** after fitting a forecasting model.

Analogous to a **regression model**, even the best models **cannot** explain the entire variation exhibited by a variable of interest.

In our case, the **fitted/estimated** values of a variable $(\hat{y}_t)$ reflect our **best effort** to predict the future values of a variable of interest.

More formally, a forecasting model's **residuals** (e_t) are equal to the *difference between the observations and the corresponding fitted values*:

$$e_t = y_t - \hat{y}_t$$

Residual diagnostics

Residual diagnostics

Residuals are useful in checking whether a model has *adequately captured* the information in the data.

One thing that we **do not** want to observe in a forecasting model's residuals are **visible patterns**.

Thus, we need to learn some **tools** that can help us in exploring some possible patterns.

Residual diagnostics

Forecasting residuals should exhibit **two** main *properties*:

1. **No autocorrelation**:
    - The presence of autocorrelation in a model's residuals implies that **useful information** that the model has ignored.
2. **Zero mean**:
    - On average, what is left out of our explicit model should not have a value different than zero. Otherwise, our forecasting model may be **biased**.

In the **absence** of either (or both) of these properties, our task is to **improve** our forecasting model.

Residual diagnostics

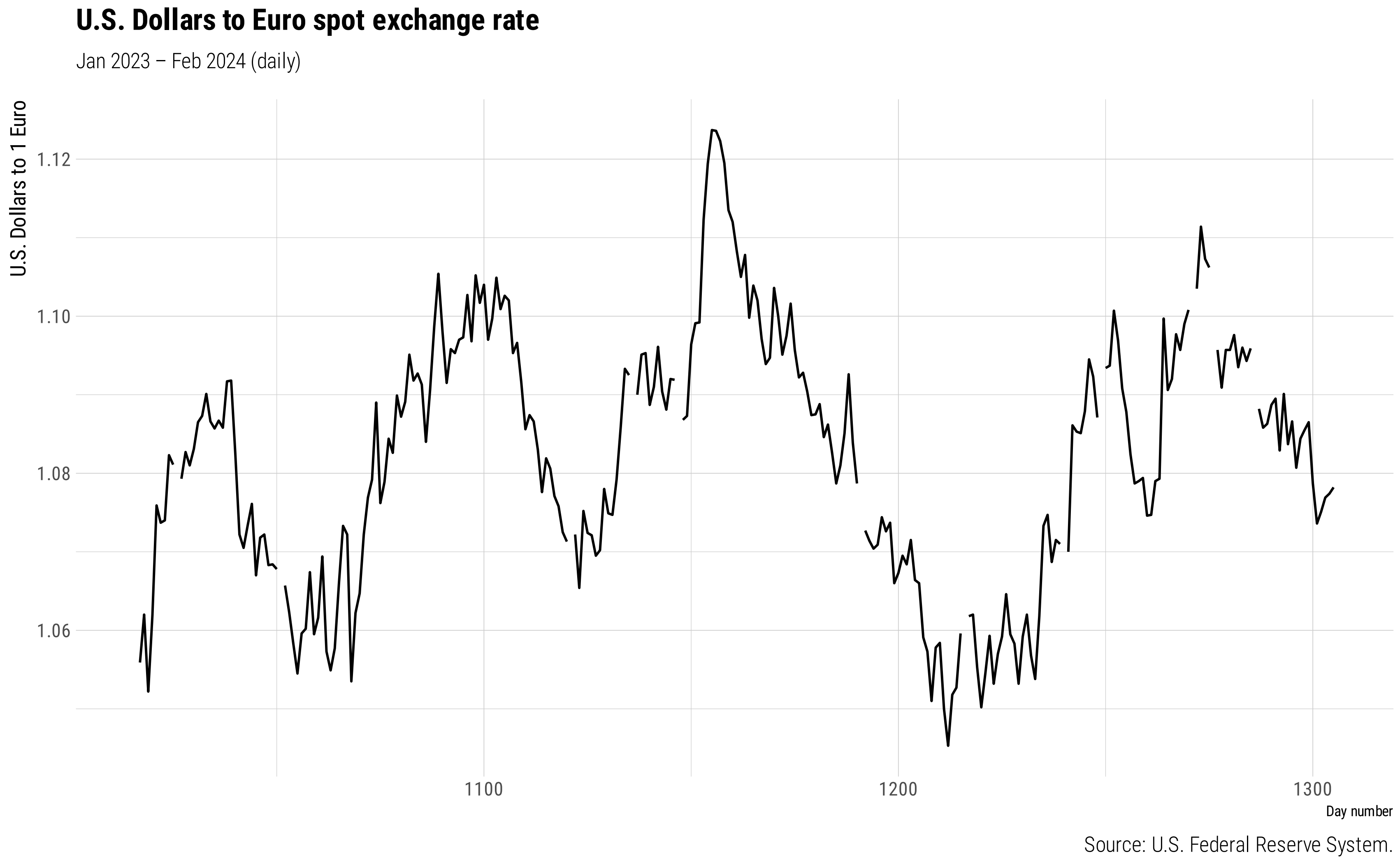
In addition to the two aforementioned properties, *two* **additional** residual features are **desirable**:

1. **Constant variance (homoskedasticity)**;
2. **Normally distributed residuals**.

Although not absolutely necessary, satisfying these two additional properties allows for better forecasting **prediction intervals**.

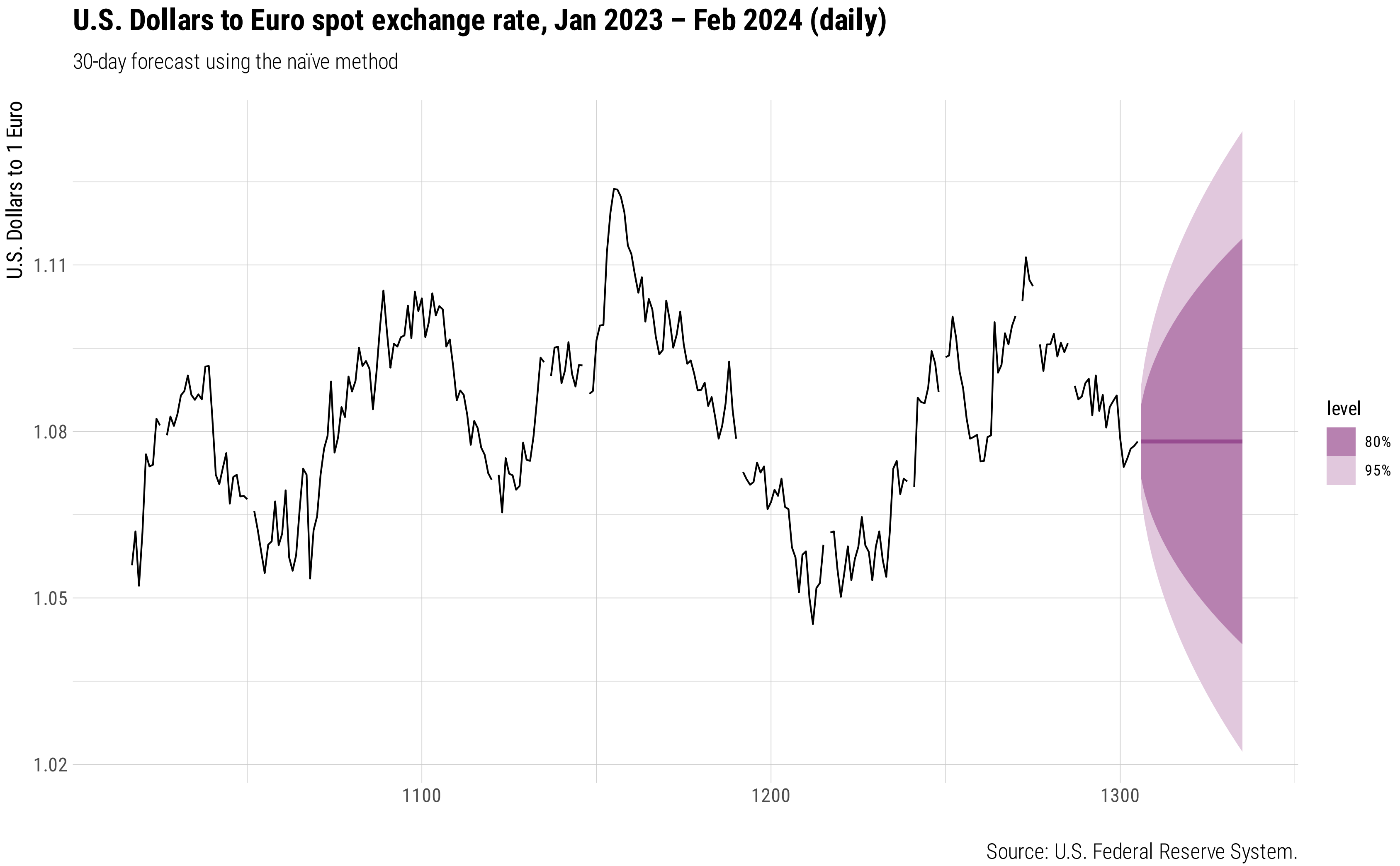
An example

An example



U.S. Dollars to Euro spot exchange rate
Jan 2023 – Feb 2024 (daily)
U.S. Dollars to 1 Euro
1.12
1.10
1.08
1.06
1100
1200
1300
Day number
Source: U.S. Federal Reserve System.

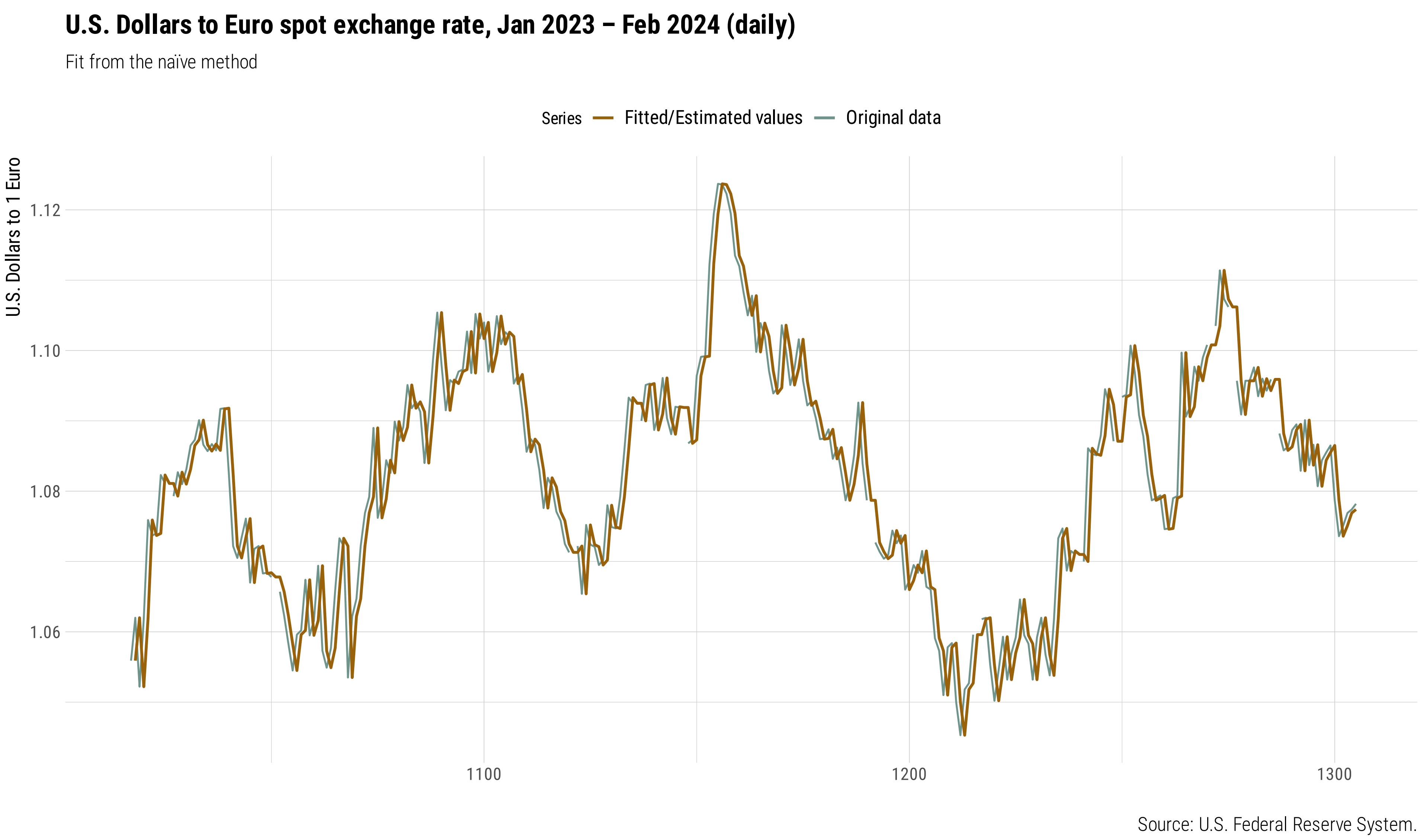
An example



U.S. Dollars to Euro spot exchange rate, Jan 2023 – Feb 2024 (daily)
30-day forecast using the naïve method
U.S. Dollars to 1 Euro
1.11
1.08
1.05
1.02
1100
1200
1300
level
80%
95%
Source: U.S. Federal Reserve System.

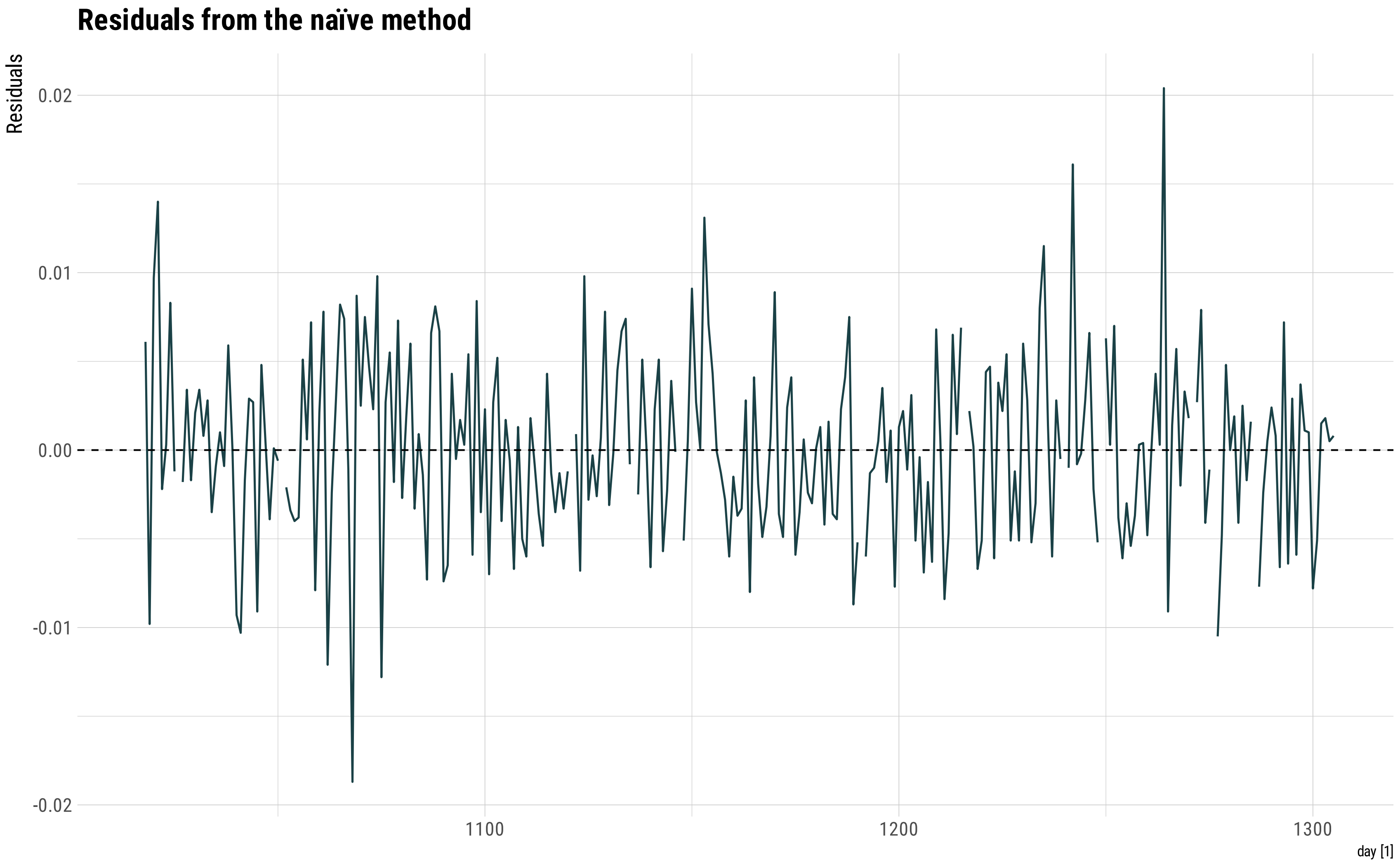
An example

```
#> # A tsibble: 290 x 6 [1]
#> # Key:       .model [1]
#>    .model        day  exch .fitted    .resid    .innov
#>    <chr>       <int> <dbl>   <dbl>     <dbl>     <dbl>
#>  1 naive_model  1016 NA      NA    NA        NA
#>  2 naive_model  1017  1.06   NA    NA        NA
#>  3 naive_model  1018  1.06    1.06  0.00610   0.00610
#>  4 naive_model  1019  1.05    1.06 -0.00980  -0.00980
#>  5 naive_model  1020  1.06    1.05  0.00970   0.00970
#>  6 naive_model  1021  1.08    1.06  0.0140    0.0140
#>  7 naive_model  1022  1.07    1.08 -0.00220  -0.00220
#>  8 naive_model  1023  1.07    1.07  0.000300  0.000300
#>  9 naive_model  1024  1.08    1.07  0.00830   0.00830
#> 10 naive_model  1025  1.08    1.08 -0.00120  -0.00120
#> # ℹ 280 more rows
```



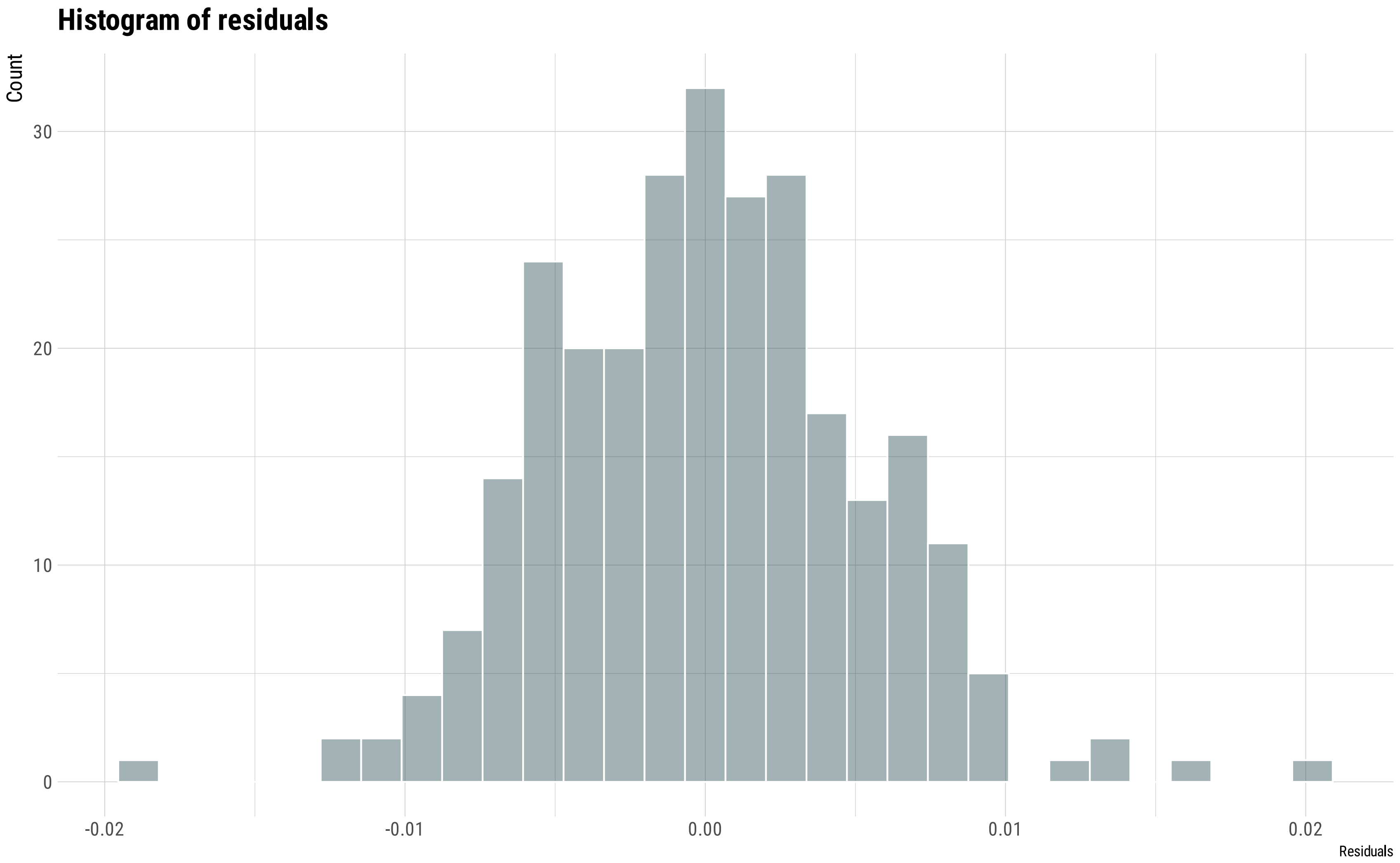
U.S. Dollars to Euro spot exchange rate, Jan 2023 – Feb 2024 (daily)
Fit from the naïve method
Series
Fitted/Estimated values
Original data
U.S. Dollars to 1 Euro
1.12
1.10
1.08
1.06
1100
1200
1300
Source: U.S. Federal Reserve System.

An example

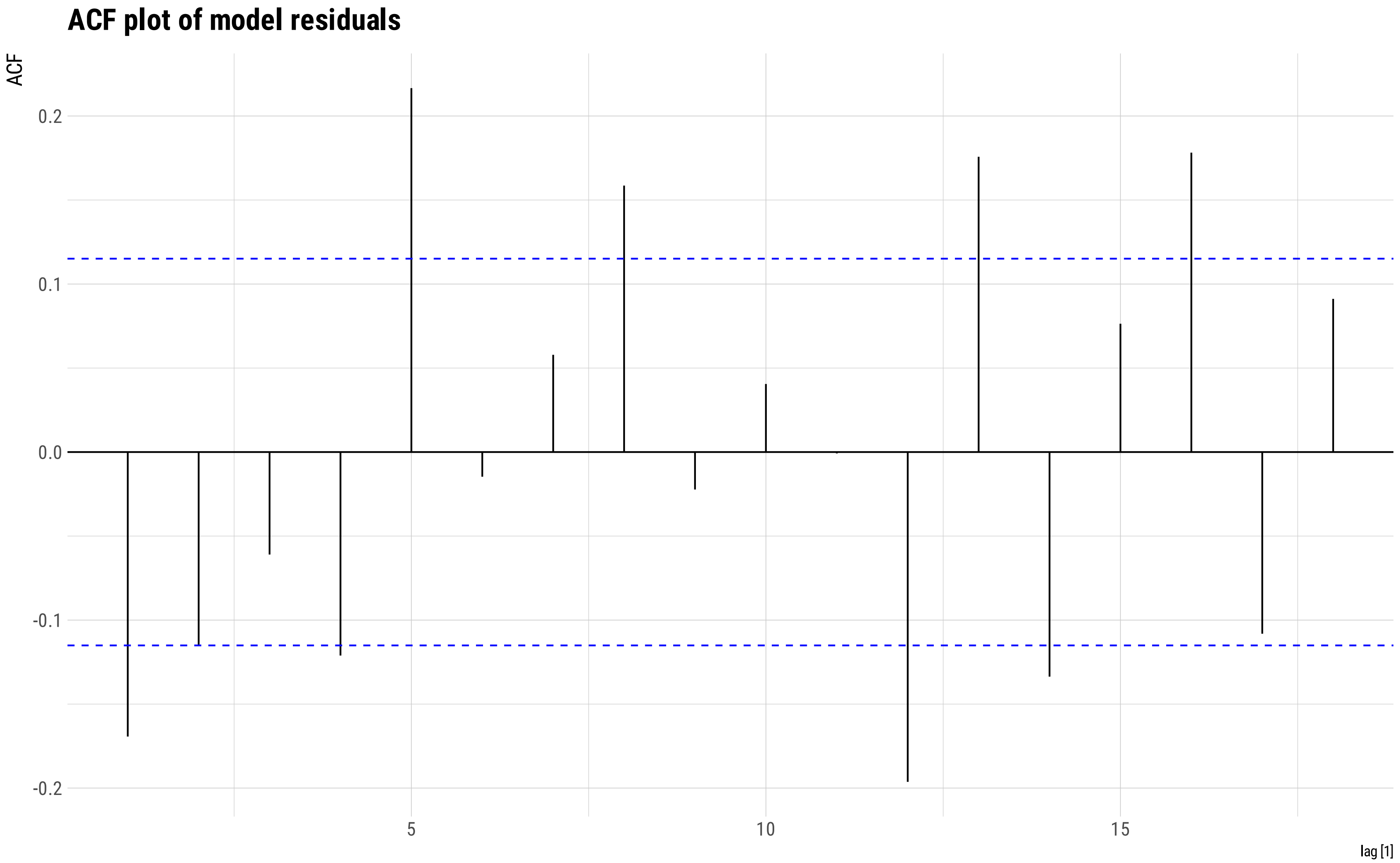


Residuals from the naïve method
Residuals
0.02
0.01
0.00
-0.01
-0.02
1100
1200
1300
day [1]

An example


Histogram of residuals
Count
30
20
10
0
-0.02
-0.01
0.00
0.01
0.02
Residuals

An example



Portmanteau tests

Portmanteau tests

Regarding residual autocorrelation, we can complement individual **ACF** analysis by using so-called **Portmanteau** tests.

The idea behind such tests is to consider a **whole set** of autocorrelation coefficients as a group, rather than testing for autocorrelation at individual lags, as we do in an ACF plot.

Thus, we can test whether the first l autocorrelations are significantly different from what would be expected from a **white noise** process.

When we test for autocorrelation in this grouped fashion, we call it a **Portmanteau** test.

We will further study **two** of these tests:

1. The *Box-Pierce* test;
2. The *Ljung-Box* test.

Portmanteau tests

- **The Box-Pierce test**:

The Box-Pierce test has the following **test statistic**:

$$Q = T \sum_{k=1}^{l} r_k^2$$

where T is the length of the time series; l is the number of lags you consider in the test; and r is the autocorrelation coefficient for each lag.

Regarding the **choice of lag** for this test, the textbook authors suggest $l = 10$ for *non-seasonal* data and $l = 2m$ for *seasonal* data (where m is the seasonal period).

Portmanteau tests

- **The Ljung-Box test**:

The Ljung-Box test has the following **test statistic**:

$$Q^* = T(T+2)\sum_{k=1}^{l}(T-k)^{-1}r_k^2$$

Compared to the Box-Pierce test, it tends to give **more accurate** results.

For both tests, we assume a **Chi-squared** (χ^2) distribution with l degrees-of-freedom.

Portmanteau tests

As a **null hypothesis**, we assume that the *residuals are not distinguishable from a white noise series*.

For our previous example,

```
#> # A tibble: 1 × 3
#>   .model      bp_stat bp_pvalue
#>   <chr>         <dbl>     <dbl>
#> 1 naive_model    11.2     0.340

#> # A tibble: 1 × 3
#>   .model      lb_stat lb_pvalue
#>   <chr>         <dbl>     <dbl>
#> 1 naive_model    11.5     0.321
```

What do we conclude?

Prediction intervals

Prediction intervals

The inherent **uncertainty** from any forecast model is expressed by a **probability distribution**.

The usual way to display forecast results is by showing its (average) **point forecast**, along with a **prediction interval**.

> A **prediction interval** gives an interval within which we expect y_t to lie with a specified probability.

For instance, a 95% prediction interval for an h-step forecast is given by

$$\hat{y}_{T+h|T} \pm 1.96\hat{\sigma}_h$$

where $\hat{\sigma}_h$ is the estimated **standard deviation** of the h-step forecast distribution.

Prediction intervals

In general, the prediction interval is obtained by:

$$\hat{y}_{T+h|T} \pm c\hat{\sigma}_h$$

where the value of c depends on the **confidence level** we assume fort the prediction interval.

The usual values are *1.64* for **90%**; *1.96* for **95%**; and *2.58* for **99%**.

Prediction intervals

The estimated standard deviation parameter ($\hat{\sigma}_h$) will have *different* definitions depending on the forecasting **method** we adopt.

For our 4 **benchmark** methods:

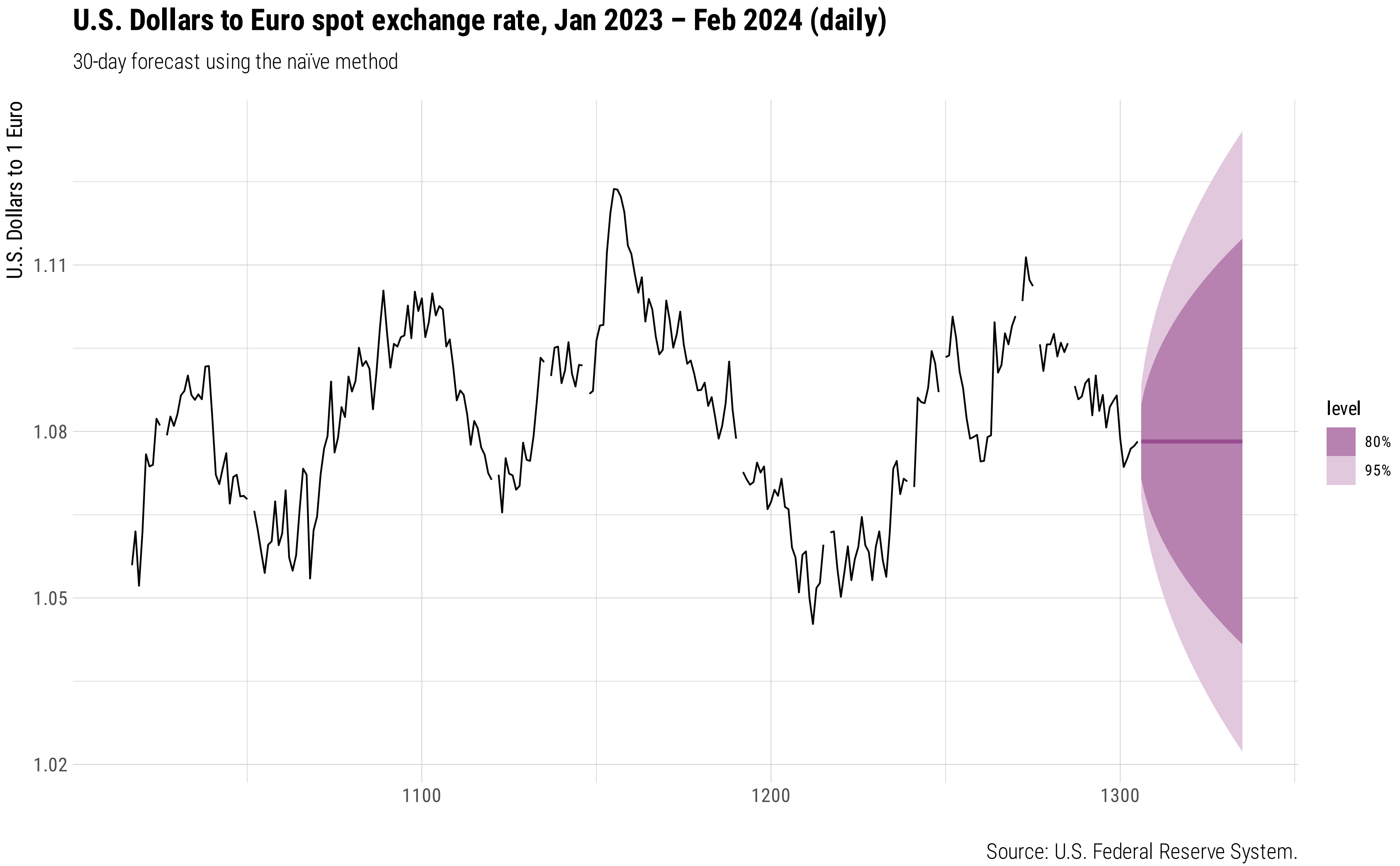
Mean:	$y_{T+h\|T} \sim N(\bar{y}, (1+1/T)\hat{\sigma}^2)$
Naïve:	$y_{T+h\|T} \sim N(y_T, h\hat{\sigma}^2)$
Seasonal naïve:	$y_{T+h\|T} \sim N(y_{T+h-m(k+1)}, (k+1)\hat{\sigma}^2)$
Drift:	$y_{T+h\|T} \sim N(y_T + \frac{h}{T-1}(y_T - y_1), h\frac{T+h}{T}\hat{\sigma}^2)$

Prediction intervals

For our previous example,

```
#> # A tsibble: 30 x 6 [1]
#> # Key:       .model [1]
#>    .model        day            exch .mean                  `80%`                  `95%`
#>    <chr>       <dbl>          <dist> <dbl>                 <hilo>                 <hilo>
#>  1 naive_model  1306 N(1.1, 2.7e-05)  1.08 [1.071523, 1.084877]80 [1.067989, 1.088411]95
#>  2 naive_model  1307 N(1.1, 5.4e-05)  1.08 [1.068758, 1.087642]80 [1.063760, 1.092640]95
#>  3 naive_model  1308 N(1.1, 8.1e-05)  1.08 [1.066636, 1.089764]80 [1.060514, 1.095886]95
#>  4 naive_model  1309 N(1.1, 0.00011)  1.08 [1.064847, 1.091553]80 [1.057778, 1.098622]95
#>  5 naive_model  1310 N(1.1, 0.00014)  1.08 [1.063271, 1.093129]80 [1.055368, 1.101032]95
#>  6 naive_model  1311 N(1.1, 0.00016)  1.08 [1.061846, 1.094554]80 [1.053188, 1.103212]95
#>  7 naive_model  1312 N(1.1, 0.00019)  1.08 [1.060535, 1.095865]80 [1.051184, 1.105216]95
#>  8 naive_model  1313 N(1.1, 0.00022)  1.08 [1.059316, 1.097084]80 [1.049319, 1.107081]95
#>  9 naive_model  1314 N(1.1, 0.00024)  1.08 [1.058170, 1.098230]80 [1.047567, 1.108833]95
#> 10 naive_model  1315 N(1.1, 0.00027)  1.08 [1.057087, 1.099313]80 [1.045910, 1.110490]95
#> # ℹ 20 more rows
```

Prediction intervals



Next time: Using transformations and decompositions