4. White noise and time series decomposition

OTexts.com/fpp/2/6
OTexts.com/fpp/6/
Student award

- To students who make the most contribution to the class, as voted by their peers.
- Contributions can be on Piazza or during the webex online sessions.
- Contribute questions, answers, comments, code suggestions, etc.
- Looking for engaged learners, not experts.
- At end of course, all students to vote for best contributor.
- $100 to the top voted student and $50 to the second student (plus t-shirts).
All sessions are at UTC 22:00 for the entire course.

Be aware when your time zone changes.

- Most of Europe changed last Sunday. So classes are now one hour earlier for most European residents.
- Most of USA and Canada change next Sunday. So classes will be one hour earlier for most North American residents from next week.
1 White noise
2 Time series decomposition
3 Seasonal adjustment
4 Forecasting and decomposition
Example: White noise

White noise

Time

0 10 20 30 40 50

−3 −2 −1 0 1 2

Forecasting using R

White noise
White noise data is uncorrelated across time with zero mean and constant variance. (Technically, we require independence as well.)
White noise data is uncorrelated across time with zero mean and constant variance. (Technically, we require independence as well.)

Think of white noise as completely uninteresting with no predictable patterns.
Example: White noise

\[ \begin{align*}
    r_1 &= 0.013 \\
    r_2 &= -0.163 \\
    r_3 &= 0.163 \\
    r_4 &= -0.259 \\
    r_5 &= -0.198 \\
    r_6 &= 0.064 \\
    r_7 &= -0.139 \\
    r_8 &= -0.032 \\
    r_9 &= 0.199 \\
    r_{10} &= -0.240
\end{align*} \]

Sample autocorrelations for white noise series. For uncorrelated data, we would expect each autocorrelation to be close to zero.
Sampling distribution of $r_k$ for white noise data is asymptotically $N(0, 1/T)$.

- 95% of all $r_k$ for white noise must lie within $\pm 1.96/\sqrt{T}$.
- If this is not the case, the series is probably not WN.
- Common to plot lines at $\pm 1.96/\sqrt{T}$ when plotting ACF. These are the critical values.
Sampling distribution of $r_k$ for white noise data is asymptotically $N(0,1/T)$.

- 95% of all $r_k$ for white noise must lie within $\pm 1.96/\sqrt{T}$.
- If this is not the case, the series is probably not WN.
- Common to plot lines at $\pm 1.96/\sqrt{T}$ when plotting ACF. These are the critical values.
Sampling distribution of $r_k$ for white noise data is asymptotically $N(0,1/T)$.

- 95% of all $r_k$ for white noise must lie within $\pm 1.96/\sqrt{T}$.
- If this is not the case, the series is probably not WN.
- Common to plot lines at $\pm 1.96/\sqrt{T}$ when plotting ACF. These are the critical values.
Sampling distribution of $r_k$ for white noise data is asymptotically $N(0, 1/T)$.

- 95% of all $r_k$ for white noise must lie within $\pm 1.96/\sqrt{T}$.
- If this is not the case, the series is probably not WN.
- Common to plot lines at $\pm 1.96/\sqrt{T}$ when plotting ACF. These are the **critical values**.
Example:

\( T = 50 \) and so critical values at 
\( \pm \frac{1.96}{\sqrt{50}} = \pm 0.28 \).

All autocorrelation coefficients lie within these limits, confirming that the data are white noise.

(More precisely, the data cannot be distinguished from white noise.)
Example: Pigs slaughtered

Number of pigs slaughtered in Victoria

Forecasting using R

White noise

9
Example: Pigs slaughtered

Forecasting using R

White noise

Lag

ACF

0 10 20 30 40

0.0

-0.2

0.2

Lag
Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 1990 through August 1995. (Source: Australian Bureau of Statistics.)

- Difficult to detect pattern in time plot.
- ACF shows some significant autocorrelation at lags 1, 2, and 3.
- \( r_{12} \) relatively large although not significant. This may indicate some slight seasonality.

These show the series is not a white noise series.
Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 1990 through August 1995. (Source: Australian Bureau of Statistics.)

- Difficult to detect pattern in time plot.
- ACF shows some significant autocorrelation at lags 1, 2, and 3.
- $r_{12}$ relatively large although not significant. This may indicate some slight seasonality.

These show the series is not a white noise series.
Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 1990 through August 1995. (Source: Australian Bureau of Statistics.)

- Difficult to detect pattern in time plot.
- ACF shows some significant autocorrelation at lags 1, 2, and 3.
- $r_{12}$ relatively large although not significant. This may indicate some slight seasonality.

These show the series is not a white noise series.
Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 1990 through August 1995. (Source: Australian Bureau of Statistics.)

- Difficult to detect pattern in time plot.
- ACF shows some significant autocorrelation at lags 1, 2, and 3.
- $r_{12}$ relatively large although not significant. This may indicate some slight seasonality.

These show the series is not a white noise series.
Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 1990 through August 1995. (Source: Australian Bureau of Statistics.)

- Difficult to detect pattern in time plot.
- ACF shows some significant autocorrelation at lags 1, 2, and 3.
- $r_{12}$ relatively large although not significant. This may indicate some slight seasonality.

These show the series is not a white noise series.
Example: Pigs slaughtered

Monthly total number of pigs slaughtered in the state of Victoria, Australia, from January 1990 through August 1995. (Source: Australian Bureau of Statistics.)

- Difficult to detect pattern in time plot.
- ACF shows some significant autocorrelation at lags 1, 2, and 3.
- $r_{12}$ relatively large although not significant. This may indicate some slight seasonality.

These show the series is not a white noise series.
We assume that the residuals are white noise (uncorrelated, mean zero, constant variance). If they aren’t, then there is information left in the residuals that should be used in computing forecasts.

So a standard residual diagnostic is to check the ACF of the residuals of a forecasting method.

We expect these to look like white noise.

Dow-Jones naive forecasts revisited

\[
\hat{y}_{t|t-1} = y_{t-1} \\
e_t = y_t - y_{t-1}
\]
We assume that the residuals are white noise (uncorrelated, mean zero, constant variance). If they aren’t, then there is information left in the residuals that should be used in computing forecasts.

So a standard residual diagnostic is to check the ACF of the residuals of a forecasting method.

We expect these to look like white noise.

Dow-Jones naive forecasts revisited

\[
\hat{y}_{t|t-1} = y_{t-1} \\
e_t = y_t - y_{t-1}
\]
ACF of residuals

- We assume that the residuals are white noise (uncorrelated, mean zero, constant variance). If they aren’t, then there is information left in the residuals that should be used in computing forecasts.
- So a standard residual diagnostic is to check the ACF of the residuals of a forecasting method.
- We expect these to look like white noise.

Dow-Jones naive forecasts revisited

\[ \hat{y}_{t|t-1} = y_{t-1} \]
\[ e_t = y_t - y_{t-1} \]
We assume that the residuals are white noise (uncorrelated, mean zero, constant variance). If they aren’t, then there is information left in the residuals that should be used in computing forecasts.

So a standard residual diagnostic is to check the ACF of the residuals of a forecasting method.

We expect these to look like white noise.

Dow-Jones naive forecasts revisited

\[ \hat{y}_{t|t-1} = y_{t-1} \]
\[ e_t = y_t - y_{t-1} \]
We assume that the residuals are white noise (uncorrelated, mean zero, constant variance). If they aren’t, then there is information left in the residuals that should be used in computing forecasts.

So a standard residual diagnostic is to check the ACF of the residuals of a forecasting method.

We expect these to look like white noise.

**Dow-Jones naive forecasts revisited**

\[
\hat{y}_{t \mid t-1} = y_{t-1} \\
e_t = y_t - y_{t-1}
\]
Forecasting Dow-Jones index

Change in Dow-Jones index

Day
Forecasting Dow-Jones index

Forecasting using R

White noise
These look like white noise.

But the ACF is a multiple testing problem.
These look like white noise.

But the ACF is a multiple testing problem.
Portmanteau tests

Consider a *whole set* of $r_k$ values, and develop a test to see whether the set is significantly different from a zero set.
Portmanteau tests

Consider a whole set of $r_k$ values, and develop a test to see whether the set is significantly different from a zero set.

**Box-Pierce test**

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

where $h$ is max lag being considered and $T$ is number of observations.

- My preferences: $h = 10$ for non-seasonal data, $h = 2m$ for seasonal data.
- If each $r_k$ close to zero, $Q$ will be small.
- If some $r_k$ values large (positive or negative), $Q$ will be large.
Portmanteau tests

Consider a whole set of $r_k$ values, and develop a test to see whether the set is significantly different from a zero set.

**Box-Pierce test**

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

where $h$ is max lag being considered and $T$ is number of observations.

- My preferences: $h = 10$ for non-seasonal data, $h = 2m$ for seasonal data.
- If each $r_k$ close to zero, $Q$ will be small.
- If some $r_k$ values large (positive or negative), $Q$ will be large.
Portmanteau tests

Consider a whole set of $r_k$ values, and develop a test to see whether the set is significantly different from a zero set.

**Box-Pierce test**

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

where $h$ is max lag being considered and $T$ is number of observations.

- My preferences: $h = 10$ for non-seasonal data, $h = 2m$ for seasonal data.
- If each $r_k$ close to zero, $Q$ will be small.
- If some $r_k$ values large (positive or negative), $Q$ will be large.
Portmanteau tests

Consider a whole set of $r_k$ values, and develop a test to see whether the set is significantly different from a zero set.

**Box-Pierce test**

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

where $h$ is max lag being considered and $T$ is number of observations.

- My preferences: $h = 10$ for non-seasonal data, $h = 2m$ for seasonal data.
- If each $r_k$ close to zero, $Q$ will be small.
- If some $r_k$ values large (positive or negative), $Q$ will be large.
Portmanteau tests

Consider a *whole set* of $r_k$ values, and develop a test to see whether the set is significantly different from a zero set.

**Ljung-Box test**

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

where $h$ is max lag being considered and $T$ is number of observations.

- My preferences: $h = 10$ for non-seasonal data, $h = 2m$ for seasonal data.
- Better performance, especially in small samples.
If data are WN, $Q^*$ has $\chi^2$ distribution with $(h - K)$ degrees of freedom where $K =$ no. parameters in model.

When applied to raw data, set $K = 0$.

For the Dow-Jones example,

```r
res <- residuals(naive(dj))

# lag=h and fitdf=K
> Box.test(res, lag=10, fitdf=0)
Box-Pierce test
X-squared = 14.0451, df = 10, p-value = 0.1709

> Box.test(res, lag=10, fitdf=0, type="Lj")
Box-Ljung test
X-squared = 14.4615, df = 10, p-value = 0.153
```
Portmanteau tests

- If data are WN, $Q^*$ has $\chi^2$ distribution with $(h - K)$ degrees of freedom where $K =$ no. parameters in model.

- When applied to raw data, set $K = 0$.

- For the Dow-Jones example,

```r
res <- residuals(naive(dj))

# lag=h and fitdf=K
> Box.test(res, lag=10, fitdf=0)
Box-Pierce test
X-squared = 14.0451, df = 10, p-value = 0.1709
> Box.test(res, lag=10, fitdf=0, type="Lj")
Box-Ljung test
X-squared = 14.4615, df = 10, p-value = 0.153
```
Portmanteau tests

- If data are WN, $Q^*$ has $\chi^2$ distribution with $(h - K)$ degrees of freedom where $K = \text{no. parameters in model}$.

- When applied to raw data, set $K = 0$.

- For the Dow-Jones example,

```r
res <- residuals(naive(dj))

# lag=h and fitdf=K
> Box.test(res, lag=10, fitdf=0)
Box-Pierce test
X-squared = 14.0451, df = 10, p-value = 0.1709
> Box.test(res, lag=10, fitdf=0, type="Lj")
Box-Ljung test
X-squared = 14.4615, df = 10, p-value = 0.153
```
Exercise

1. Calculate the residuals from a seasonal naive forecast applied to the quarterly Australian beer production data from 1992.

2. Test if the residuals are white noise.
Exercise

1. Calculate the residuals from a seasonal naive forecast applied to the quarterly Australian beer production data from 1992.

2. Test if the residuals are white noise.

```r
beer <- window(ausbeer,start=1992)
fc <- snaive(beer)
res <- residuals(fc)
Acf(res)
Box.test(res, lag=8, fitdf=0, type="Lj")
```
Outline

1 White noise

2 Time series decomposition

3 Seasonal adjustment

4 Forecasting and decomposition
Time series decomposition

\[ Y_t = f(S_t, T_t, E_t) \]

where

- \( Y_t \) = data at period \( t \)
- \( S_t \) = seasonal component at period \( t \)
- \( T_t \) = trend-cycle component at period \( t \)
- \( E_t \) = remainder (or irregular or error) component at period \( t \)

Additive decomposition: \( Y_t = S_t + T_t + E_t \).

Multiplicative decomposition: \( Y_t = S_t \times T_t \times E_t \).
Time series decomposition

\[ Y_t = f(S_t, T_t, E_t) \]

where

\[ Y_t = \text{data at period } t \]
\[ S_t = \text{seasonal component at period } t \]
\[ T_t = \text{trend-cycle component at period } t \]
\[ E_t = \text{remainder (or irregular or error) component at period } t \]

Additive decomposition: \( Y_t = S_t + T_t + E_t \).

Multiplicative decomposition: \( Y_t = S_t \times T_t \times E_t \).
Time series decomposition

\[ Y_t = f(S_t, T_t, E_t) \]

where

- \( Y_t \) = data at period \( t \)
- \( S_t \) = seasonal component at period \( t \)
- \( T_t \) = trend-cycle component at period \( t \)
- \( E_t \) = remainder (or irregular or error) component at period \( t \)

**Additive decomposition:** \( Y_t = S_t + T_t + E_t \).

**Multiplicative decomposition:** \( Y_t = S_t \times T_t \times E_t \).
Time series decomposition

- Additive model appropriate if magnitude of seasonal fluctuations does not vary with level.
- If seasonal are proportional to level of series, then multiplicative model appropriate.
- Multiplicative decomposition more prevalent with economic series.
- Logs turn multiplicative relationship into an additive relationship:

\[ Y_t = S_t \times T_t \times E_t \implies \log Y_t = \log S_t + \log T_t + \log E_t. \]
Additive model appropriate if magnitude of seasonal fluctuations does not vary with level.

If seasonal are proportional to level of series, then multiplicative model appropriate.

Multiplicative decomposition more prevalent with economic series.

Logs turn multiplicative relationship into an additive relationship:

\[ Y_t = S_t \times T_t \times E_t \implies \log Y_t = \log S_t + \log T_t + \log E_t. \]
Additive model appropriate if magnitude of seasonal fluctuations does not vary with level.

If seasonal are proportional to level of series, then multiplicative model appropriate.

Multiplicative decomposition more prevalent with economic series

Logs turn multiplicative relationship into an additive relationship:

\[ Y_t = S_t \times T_t \times E_t \Rightarrow \log Y_t = \log S_t + \log T_t + \log E_t. \]
Additive model appropriate if magnitude of seasonal fluctuations does not vary with level.

If seasonal are proportional to level of series, then multiplicative model appropriate.

Multiplicative decomposition more prevalent with economic series.

Logs turn multiplicative relationship into an additive relationship:

\[ Y_t = S_t \times T_t \times E_t \quad \Rightarrow \quad \log Y_t = \log S_t + \log T_t + \log E_t. \]
History of time series decomposition

- Classical method originated in 1920s.
  - In R: `decompose()`.
- Census II method introduced in 1957. Basis for modern X-12-ARIMA method.
- STL method introduced in 1983. It only allows additive decomposition.
  - In R: `stl()`.
- TRAMO/SEATS introduced in 1990s.
Classical method originated in 1920s. In R: `decompose()`.

Census II method introduced in 1957. Basis for modern X-12-ARIMA method.


TRAMO/SEATS introduced in 1990s.
History of time series decomposition

- Classical method originated in 1920s. In R: `decompose()`.
- Census II method introduced in 1957. Basis for modern X-12-ARIMA method.
- TRAMO/SEATS introduced in 1990s.
History of time series decomposition

- Classical method originated in 1920s. 
  In R: `decompose()`.

- Census II method introduced in 1957. 
  Basis for modern X-12-ARIMA method.

- STL method introduced in 1983. It only allows additive decomposition. 
  In R: `stl()`.

- TRAMO/SEATS introduced in 1990s.
History of time series decomposition

- Classical method originated in 1920s. In R: `decompose()`.
- Census II method introduced in 1957. Basis for modern X-12-ARIMA method.
- TRAMO/SEATS introduced in 1990s.
History of time series decomposition

- Classical method originated in 1920s. In R: `decompose()`.
- Census II method introduced in 1957. Basis for modern X-12-ARIMA method.
- TRAMO/SEATS introduced in 1990s.
Classical decomposition

Forecasting using R

Time series decomposition

![Graph showing data, seasonal, trend, and remainder components over time.](image)
STL decomposition

Forecasting using R

Time series decomposition
Examples: US house sales

Sales of new one-family houses, USA

Total sales


30 40 50 60 70 80 90

Forecasting using R

Time series decomposition

26
Examples: US house sales

Sales of new one-family houses, USA

<table>
<thead>
<tr>
<th>Year</th>
<th>Total Sales</th>
</tr>
</thead>
<tbody>
<tr>
<td>1975</td>
<td>30</td>
</tr>
<tr>
<td>1980</td>
<td>40</td>
</tr>
<tr>
<td>1985</td>
<td>50</td>
</tr>
<tr>
<td>1990</td>
<td>60</td>
</tr>
<tr>
<td>1995</td>
<td>70</td>
</tr>
</tbody>
</table>
Examples: US house sales

Time series decomposition

Forecasting using R

Time Data
30 50 70 90
Seasonal
−10 0 5
Examples: US house sales

Time series decomposition

Data

<table>
<thead>
<tr>
<th>Year</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>1975</td>
<td>30</td>
</tr>
<tr>
<td>1980</td>
<td>50</td>
</tr>
<tr>
<td>1985</td>
<td>70</td>
</tr>
<tr>
<td>1990</td>
<td>90</td>
</tr>
</tbody>
</table>

Error

<table>
<thead>
<tr>
<th>Year</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1975</td>
<td>-15</td>
</tr>
<tr>
<td>1980</td>
<td>-5</td>
</tr>
<tr>
<td>1985</td>
<td>5</td>
</tr>
<tr>
<td>1990</td>
<td>15</td>
</tr>
</tbody>
</table>

Time Series Analysis in R

- Time series decomposition
- Forecasting

Graphs showing time series data and error over time.
### Examples: US house sales

<table>
<thead>
<tr>
<th>Time</th>
<th>Data</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>1975</td>
<td>30</td>
<td>-15</td>
</tr>
<tr>
<td>1980</td>
<td>50</td>
<td>-5</td>
</tr>
<tr>
<td>1985</td>
<td>70</td>
<td>5</td>
</tr>
<tr>
<td>1990</td>
<td>90</td>
<td>15</td>
</tr>
<tr>
<td>1995</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

![Graph showing time series data and error]

**Forecasting using R**

**Time series decomposition**
Examples: US house sales

Forecasting using R

Time series decomposition

![Time series decomposition graph](image)
plot(decompose(hsales))
plot(stl(hsales,s.window="periodic"))
plot(stl(hsales,s.window=15))
Forecasting using R

Time series decomposition
Euro electrical equipment

Time series decomposition

Forecasting using R
Euro electrical equipment

```r
fit <- stl(elecequip, s.window=5)
plot(fit)
```
Seasonal sub-series plot of the seasonal component
Euro electrical equipment

Seasonal sub-series plot of the seasonal component

```r
monthplot(fit$time.series[, "seasonal"], ylab="Seasonal")
```
Euro electrical equipment

Forecasting using R

Time series decomposition

Data
Seasonal
Trend
Remainder
fit <- stl(elecequip, t.window=15, s.window="periodic", robust=TRUE)
plot(fit)
fit <- stl(elecequip, t.window=15, s.window="periodic", robust=TRUE)
plot(fit)

- `t.window` controls wiggliness of trend component.
- `s.window` controls variation in seasonal component.
Euro electrical equipment

fit <- stl(elecequip, t.window=15, s.window="periodic", robust=TRUE)
plot(fit)

- **t.window** controls wiggliness of trend component.

- **s.window** controls variation in seasonal component.
1 White noise

2 Time series decomposition

3 Seasonal adjustment

4 Forecasting and decomposition
Useful by-product of decomposition: an easy way to calculate seasonally adjusted data.

Additive decomposition: seasonally adjusted data given by

\[ Y_t - S_t = T_t + E_t \]

Multiplicative decomposition: seasonally adjusted data given by

\[ Y_t / S_t = T_t \times E_t \]
Useful by-product of decomposition: an easy way to calculate seasonally adjusted data.

Additive decomposition: seasonally adjusted data given by

\[ Y_t - S_t = T_t + E_t \]

Multiplicative decomposition: seasonally adjusted data given by

\[ \frac{Y_t}{S_t} = T_t \times E_t \]
Useful by-product of decomposition: an easy way to calculate seasonally adjusted data.

Additive decomposition: seasonally adjusted data given by

\[ Y_t - S_t = T_t + E_t \]

Multiplicative decomposition: seasonally adjusted data given by

\[ \frac{Y_t}{S_t} = T_t \times E_t \]
Euro electrical equipment

Electrical equipment manufacturing

Seasonally adjusted series

New orders index

2000 2005 2010
60 70 80 90 100 110 120 130
seasadj(obj)
where obj is the output from stl() or decompose().

Example

```r
plot(hsales,col="gray")
fit <- stl(hsales,s.window=15)
hsales.sa <- seasadj(fit)
lines(hsales.sa, col="red")
```
seasadj(obj)

where \texttt{obj} is the output from \texttt{stl()} or \texttt{decompose()}.

**Example**

```r
plot(hsales, col="gray")
fit <- stl(hsales, s.window=15)
hsales.sa <- seasadj(fit)
lines(hsales.sa, col="red")
```
Outline

1. White noise
2. Time series decomposition
3. Seasonal adjustment
4. Forecasting and decomposition
Forecast seasonal component by repeating the last year

Forecast seasonally adjusted data using non-seasonal time series method. E.g.,
  - Holt’s method — next topic
  - Random walk with drift model

Combine forecasts of seasonal component with forecasts of seasonally adjusted data to get forecasts of original data.

Sometimes a decomposition is useful just for understanding the data before building a separate forecasting model.
Forecasting and decomposition

- Forecast seasonal component by repeating the last year
- Forecast seasonally adjusted data using non-seasonal time series method. E.g.,
  - Holt’s method — next topic
  - Random walk with drift model

- Combine forecasts of seasonal component with forecasts of seasonally adjusted data to get forecasts of original data.
- Sometimes a decomposition is useful just for understanding the data before building a separate forecasting model.
Forecasting and decomposition

- Forecast seasonal component by repeating the last year
- Forecast seasonally adjusted data using non-seasonal time series method. E.g.,
  - Holt’s method — next topic
  - Random walk with drift model
- Combine forecasts of seasonal component with forecasts of seasonally adjusted data to get forecasts of original data.
- Sometimes a decomposition is useful just for understanding the data before building a separate forecasting model.
Forecast seasonal component by repeating the last year

Forecast seasonally adjusted data using non-seasonal time series method. E.g.,

- Holt’s method — next topic
- Random walk with drift model

Combine forecasts of seasonal component with forecasts of seasonally adjusted data to get forecasts of original data.

Sometimes a decomposition is useful just for understanding the data before building a separate forecasting model.
Forecast seasonal component by repeating the last year.

Forecast seasonally adjusted data using non-seasonal time series method. E.g.,

- Holt’s method — next topic
- Random walk with drift model

Combine forecasts of seasonal component with forecasts of seasonally adjusted data to get forecasts of original data.

Sometimes a decomposition is useful just for understanding the data before building a separate forecasting model.
Forecasting and decomposition

- Forecast seasonal component by repeating the last year
- Forecast seasonally adjusted data using non-seasonal time series method. E.g.,
  - Holt’s method — next topic
  - Random walk with drift model
- Combine forecasts of seasonal component with forecasts of seasonally adjusted data to get forecasts of original data.
- Sometimes a decomposition is useful just for understanding the data before building a separate forecasting model.
Naive forecasts of seasonally adjusted data

New orders index

2000 2005 2010
70 80 90 100 110 120
Forecasts from STL + Random walk

New orders index

2000 2005 2010
40 60 80 100 120
How to do this in R

```r
fit <- stl(elecequip, t.window=15,
          s.window="periodic", robust=TRUE)

eeadj <- seasadj(fit)
plot(naive(eeadj), xlab="New orders index")

fcast <- forecast(fit, method="naive")
plot(fcast, ylab="New orders index")
```
 Decomposition and prediction intervals

It is common to take the prediction intervals from the seasonally adjusted forecasts and modify them with the seasonal component.

This ignores the uncertainty in the seasonal component estimate.

It also ignores the uncertainty in the future seasonal pattern.
Decomposition and prediction intervals

- It is common to take the prediction intervals from the seasonally adjusted forecasts and modify them with the seasonal component.
- This ignores the uncertainty in the seasonal component estimate.
- It also ignores the uncertainty in the future seasonal pattern.
It is common to take the prediction intervals from the seasonally adjusted forecasts and modify them with the seasonal component. This ignores the uncertainty in the seasonal component estimate. It also ignores the uncertainty in the future seasonal pattern.