7. Transformations and adjustments

OTexts.com/fpp/2/4/
1 Exponential smoothing

2 Transformations

3 Adjustments
Exponential smoothing

**ets() function**
- Automatically chooses a model by default using the AIC, AICc or BIC.
- Can handle any combination of trend, seasonality and damping
- Produces prediction intervals for every model
- Ensures the parameters are admissible (equivalent to invertible)
- Produces an object of class ets.
Exponential smoothing

**ets() function**

- Automatically chooses a model by default using the AIC, AICc or BIC.
- Can handle any combination of trend, seasonality and damping.
- Produces prediction intervals for every model.
- Ensures the parameters are admissible (equivalent to invertible).
- Produces an object of class `ets`. 
Exponential smoothing

**ets() function**

- Automatically chooses a model by default using the AIC, AICc or BIC.
- Can handle any combination of trend, seasonality and damping
- Produces prediction intervals for every model
- Ensures the parameters are admissible (equivalent to invertible)
- Produces an object of class ets.
ets() function

- Automatically chooses a model by default using the AIC, AICc or BIC.
- Can handle any combination of trend, seasonality and damping
- Produces prediction intervals for every model
- Ensures the parameters are admissible (equivalent to invertible)
- Produces an object of class ets.
Exponential smoothing

ets() function

- Automatically chooses a model by default using the AIC, AICc or BIC.
- Can handle any combination of trend, seasonality and damping
- Produces prediction intervals for every model
- Ensures the parameters are admissible (equivalent to invertible)
- Produces an object of class ets.
Exponential smoothing

ets objects

- **Methods**: `coef()`, `plot()`, `summary()`, `residuals()`, `fitted()`, `simulate()` and `forecast()`

- `plot()` function shows time plots of the original time series along with the extracted components (level, growth and seasonal).
ets objects

- **Methods**: `coef()`, `plot()`, `summary()`, `residuals()`, `fitted()`, `simulate()` and `forecast()`

- `plot()` function shows time plots of the original time series along with the extracted components (level, growth and seasonal).
Exponential smoothing

Forecasting using R

Decomposition by ETS(M,Md,M) method

plot(fit)
Goodness-of-fit

> accuracy(fit)

<table>
<thead>
<tr>
<th>ME</th>
<th>RMSE</th>
<th>MAE</th>
<th>MPE</th>
<th>MAPE</th>
<th>MASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.17847</td>
<td>15.48781</td>
<td>11.77800</td>
<td>0.07204</td>
<td>2.81921</td>
<td>0.20705</td>
</tr>
</tbody>
</table>

> accuracy(fit2)

<table>
<thead>
<tr>
<th>ME</th>
<th>RMSE</th>
<th>MAE</th>
<th>MPE</th>
<th>MAPE</th>
<th>MASE</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.11711</td>
<td>15.90526</td>
<td>12.18930</td>
<td>-0.03765</td>
<td>2.91255</td>
<td>0.21428</td>
</tr>
</tbody>
</table>
Forecast intervals

Forecasts from ETS(M,Md,M)

> plot(forecast(fit, level=c(50, 80, 95)))
Forecast intervals

Forecasts from ETS(M,Md,M)

> plot(forecast(fit, fan=TRUE))
**Exponential smoothing**

The `ets()` function also allows refitting model to new data set.

```r
> usfit <- ets(usnetelec[1:45])
> test <- ets(usnetelec[46:55], model = usfit)

> accuracy(test)

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>RMSE</td>
<td>MAE</td>
<td>MPE</td>
<td>MAPE</td>
<td>MASE</td>
</tr>
<tr>
<td>-3.35419</td>
<td>58.02763</td>
<td>43.85545</td>
<td>-0.07624</td>
<td>1.18483</td>
<td>0.52452</td>
</tr>
</tbody>
</table>

> accuracy(forecast(usfit,10), usnetelec[46:55])

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ME</td>
<td>RMSE</td>
<td>MAE</td>
<td>MPE</td>
<td>MAPE</td>
<td>MASE</td>
</tr>
<tr>
<td>40.7034</td>
<td>61.2075</td>
<td>46.3246</td>
<td>1.0980</td>
<td>1.2620</td>
<td>0.6776</td>
</tr>
</tbody>
</table>
```
Unstable models

- ETS(M,M,A)
- ETS(M,M_d,A)
- ETS(A,N,M)
- ETS(A,A,M)
- ETS(A,A_d,M)
- ETS(A,M,N)
- ETS(A,M,A)
- ETS(A,M_d,M)
- ETS(A,M,A)
- ETS(A,M,M)
- ETS(A,M_d,N)
- ETS(A,M_d,A)
- ETS(A,M_d,M)
Unstable models

- ETS(M,M,A)
- ETS(M,M_d,A)
- ETS(A,N,M)
- ETS(A,A,M)
- ETS(A,A_d,M)
- ETS(A,M,N)
- ETS(A,M,A)
- ETS(A,M,M)
- ETS(A,M_d,N)
- ETS(A,M_d,A)
- ETS(A,M_d,M)

In practice, the models work fine for short- to medium-term forecasts provided the data are strictly positive.
ets(y, model="ZZZ", damped=NULL, alpha=NULL, beta=NULL, gamma=NULL, phi=NULL, additive.only=FALSE, lower=c(rep(0.0001,3),0.80), upper=c(rep(0.9999,3),0.98), opt.crit=c("lik","amse","mse","sigma"), nmse=3, bounds=c("both","usual","admissible"), ic=c("aic","aicc","bic"), restrict=TRUE)
The magic `forecast()` function

- `forecast` returns forecasts when applied to an `ets` object (or the output from many other time series models).
- If you use `forecast` directly on data, it will select an ETS model automatically and then return forecasts.
The magic forecast() function

- `forecast` returns forecasts when applied to an `ets` object (or the output from many other time series models).
- If you use `forecast` directly on data, it will select an ETS model automatically and then return forecasts.
1 Exponential smoothing

2 Transformations

3 Adjustments
Transformations to stabilize the variance

If the data show different variation at different levels of the series, then a transformation can be useful.

Denote original observations as $y_1, \ldots, y_n$ and transformed observations as $w_1, \ldots, w_n$.

### Mathematical transformations for stabilizing variation

<table>
<thead>
<tr>
<th>Transformation</th>
<th>Formula</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Square root</td>
<td>$w_t = \sqrt{y_t}$</td>
<td></td>
</tr>
<tr>
<td>Cube root</td>
<td>$w_t = \sqrt[3]{y_t}$</td>
<td>Increasing</td>
</tr>
<tr>
<td>Logarithm</td>
<td>$w_t = \log(y_t)$</td>
<td>strength</td>
</tr>
</tbody>
</table>

Logarithms, in particular, are useful because they are more interpretable: changes in a log value are relative (percent) changes on the original scale.
If the data show different variation at different levels of the series, then a transformation can be useful. Denote original observations as $y_1, \ldots, y_n$ and transformed observations as $w_1, \ldots, w_n$.

### Mathematical transformations for stabilizing variation

<table>
<thead>
<tr>
<th>Transformation</th>
<th>Formula</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Square root</td>
<td>$w_t = \sqrt{y_t}$</td>
<td></td>
</tr>
<tr>
<td>Cube root</td>
<td>$w_t = \sqrt[3]{y_t}$</td>
<td>Increasing</td>
</tr>
<tr>
<td>Logarithm</td>
<td>$w_t = \log(y_t)$</td>
<td>strength</td>
</tr>
</tbody>
</table>

Logarithms, in particular, are useful because they are more interpretable: changes in a log value are relative (percent) changes on the original scale.
Transformations to stabilize the variance

If the data show different variation at different levels of the series, then a transformation can be useful. Denote original observations as $y_1, \ldots, y_n$ and transformed observations as $w_1, \ldots, w_n$.

### Mathematical transformations for stabilizing variation

<table>
<thead>
<tr>
<th>Transformation</th>
<th>Formula</th>
<th>Effect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Square root</td>
<td>$w_t = \sqrt{y_t}$</td>
<td>↓</td>
</tr>
<tr>
<td>Cube root</td>
<td>$w_t = \sqrt[3]{y_t}$</td>
<td>Increasing</td>
</tr>
<tr>
<td>Logarithm</td>
<td>$w_t = \log(y_t)$</td>
<td>strength</td>
</tr>
</tbody>
</table>

Logarithms, in particular, are useful because they are more interpretable: changes in a log value are relative (percent) changes on the original scale.
Transformations to stabilize the variance

If the data show different variation at different levels of the series, then a transformation can be useful. Denote original observations as $y_1, \ldots, y_n$ and transformed observations as $w_1, \ldots, w_n$.

<table>
<thead>
<tr>
<th>Mathematical transformations for stabilizing variation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Square root: $w_t = \sqrt{y_t}$</td>
</tr>
<tr>
<td>Cube root: $w_t = \sqrt[3]{y_t}$</td>
</tr>
<tr>
<td>Logarithm: $w_t = \log(y_t)$</td>
</tr>
</tbody>
</table>

Logarithms, in particular, are useful because they are more interpretable: changes in a log value are **relative (percent)** changes on the original scale.
Transformations

Square root electricity production

Year
40 60 80 100 120
Transformations

Cube root electricity production

Year


12 14 16 18 20 22 24
Transformations

Forecasting using R

Log electricity production

![Graph of log electricity production](image-url)

<table>
<thead>
<tr>
<th>Year</th>
<th>Log electricity production</th>
</tr>
</thead>
<tbody>
<tr>
<td>1960</td>
<td>7.5</td>
</tr>
<tr>
<td>1970</td>
<td>8.0</td>
</tr>
<tr>
<td>1980</td>
<td>8.5</td>
</tr>
<tr>
<td>1990</td>
<td>9.0</td>
</tr>
</tbody>
</table>
Inverse electricity production

Year
−8e−04 −6e−04 −4e−04 −2e−04
Each of these transformations is close to a member of the family of **Box-Cox transformations**:

\[ w_t = \begin{cases} 
\log(y_t), & \lambda = 0; \\
(y_t^\lambda - 1)/\lambda, & \lambda \neq 0.
\end{cases} \]

- \( \lambda = 1 \): (No substantive transformation)
- \( \lambda = \frac{1}{2} \): (Square root plus linear transformation)
- \( \lambda = 0 \): (Natural logarithm)
- \( \lambda = -1 \): (Inverse plus 1)
Box-Cox transformations

Each of these transformations is close to a member of the family of **Box-Cox transformations**:

\[ w_t = \begin{cases} 
\log(y_t), & \lambda = 0; \\
(y_t^\lambda - 1)/\lambda, & \lambda \neq 0.
\end{cases} \]

- **\( \lambda = 1 \)**: (No substantive transformation)
- **\( \lambda = \frac{1}{2} \)**: (Square root plus linear transformation)
- **\( \lambda = 0 \)**: (Natural logarithm)
- **\( \lambda = -1 \)**: (Inverse plus 1)
Box-Cox transformations

Each of these transformations is close to a member of the family of **Box-Cox transformations**:

\[
W_t = \begin{cases} 
\log(y_t), & \lambda = 0; \\
(y_t^\lambda - 1)/\lambda, & \lambda \neq 0.
\end{cases}
\]

- \(\lambda = 1\): (No substantive transformation)
- \(\lambda = \frac{1}{2}\): (Square root plus linear transformation)
- \(\lambda = 0\): (Natural logarithm)
- \(\lambda = -1\): (Inverse plus 1)
Each of these transformations is close to a member of the family of **Box-Cox transformations**:

\[ w_t = \begin{cases} 
\log(y_t), & \lambda = 0; \\
(y_t^\lambda - 1)/\lambda, & \lambda \neq 0.
\end{cases} \]

- \( \lambda = 1 \): (No substantive transformation)
- \( \lambda = \frac{1}{2} \): (Square root plus linear transformation)
- \( \lambda = 0 \): (Natural logarithm)
- \( \lambda = -1 \): (Inverse plus 1)
Box-Cox transformations

Each of these transformations is close to a member of the family of **Box-Cox transformations**:

\[
W_t = \begin{cases} 
\log(y_t), & \lambda = 0; \\
\frac{(y_t^\lambda - 1)}{\lambda}, & \lambda \neq 0.
\end{cases}
\]

- $\lambda = 1$: (No substantive transformation)
- $\lambda = \frac{1}{2}$: (Square root plus linear transformation)
- $\lambda = 0$: (Natural logarithm)
- $\lambda = -1$: (Inverse plus 1)
Box-Cox transformations

\[ \lambda = 1.00 \]

Monthly electricity production

Year


Graph showing monthly electricity production with \( \lambda = 1.00 \).
Box-Cox transformations

- $y_t^\lambda$ for $\lambda$ close to zero behaves like logs.
- If some $y_t = 0$, then must have $\lambda > 0$.
- If some $y_t < 0$, no power transformation is possible unless all $y_t$ adjusted by adding a constant to all values.
- Choose a simple value of $\lambda$. It makes explanation easier.
- Results are relatively insensitive to value of $\lambda$.
- Often no transformation ($\lambda = 1$) needed.
- Transformation often makes little difference to forecasts but has large effect on PI.
- Choosing $\lambda = 0$ is a simple way to force forecasts to be positive.
Box-Cox transformations

- $y_t^\lambda$ for $\lambda$ close to zero behaves like logs.
- If some $y_t = 0$, then must have $\lambda > 0$
- If some $y_t < 0$, no power transformation is possible unless all $y_t$ adjusted by adding a constant to all values.
- Choose a simple value of $\lambda$. It makes explanation easier.
- Results are relatively insensitive to value of $\lambda$
- Often no transformation ($\lambda = 1$) needed.
- Transformation often makes little difference to forecasts but has large effect on PI.
- Choosing $\lambda = 0$ is a simple way to force forecasts to be positive.
Box-Cox transformations

- $y_t^\lambda$ for $\lambda$ close to zero behaves like logs.
- If some $y_t = 0$, then must have $\lambda > 0$.
- If some $y_t < 0$, no power transformation is possible unless all $y_t$ adjusted by adding a constant to all values.

Choose a simple value of $\lambda$. It makes explanation easier.

- Results are relatively insensitive to value of $\lambda$.
- Often no transformation ($\lambda = 1$) needed.
- Transformation often makes little difference to forecasts but has large effect on PI.
- Choosing $\lambda = 0$ is a simple way to force forecasts to be positive.
Box-Cox transformations

- $y_t^\lambda$ for $\lambda$ close to zero behaves like logs.
- If some $y_t = 0$, then must have $\lambda > 0$.
- If some $y_t < 0$, no power transformation is possible unless all $y_t$ adjusted by adding a constant to all values.
- Choose a simple value of $\lambda$. It makes explanation easier.
- Results are relatively insensitive to value of $\lambda$.
- Often no transformation ($\lambda = 1$) needed.
- Transformation often makes little difference to forecasts but has large effect on PI.
- Choosing $\lambda = 0$ is a simple way to force forecasts to be positive.
Box-Cox transformations

- $y_t^\lambda$ for $\lambda$ close to zero behaves like logs.
- If some $y_t = 0$, then must have $\lambda > 0$
- If some $y_t < 0$, no power transformation is possible unless all $y_t$ adjusted by **adding a constant to all values**.
- Choose a simple value of $\lambda$. It makes explanation easier.
- Results are relatively insensitive to value of $\lambda$
  - Often no transformation ($\lambda = 1$) needed.
  - Transformation often makes little difference to forecasts but has large effect on PI.
  - Choosing $\lambda = 0$ is a simple way to force forecasts to be positive.
Box-Cox transformations

- $y_t^\lambda$ for $\lambda$ close to zero behaves like logs.
- If some $y_t = 0$, then must have $\lambda > 0$
- If some $y_t < 0$, no power transformation is possible unless all $y_t$ adjusted by adding a constant to all values.
- Choose a simple value of $\lambda$. It makes explanation easier.
- Results are relatively insensitive to value of $\lambda$
- Often no transformation ($\lambda = 1$) needed.
- Transformation often makes little difference to forecasts but has large effect on PI.
- Choosing $\lambda = 0$ is a simple way to force forecasts to be positive.
Box-Cox transformations

- $y_t^\lambda$ for $\lambda$ close to zero behaves like logs.
- If some $y_t = 0$, then must have $\lambda > 0$
- If some $y_t < 0$, no power transformation is possible unless all $y_t$ adjusted by adding a constant to all values.
- Choose a simple value of $\lambda$. It makes explanation easier.
- Results are relatively insensitive to value of $\lambda$
- Often no transformation ($\lambda = 1$) needed.
- Transformation often makes little difference to forecasts but has large effect on PI.
- Choosing $\lambda = 0$ is a simple way to force forecasts to be positive.
Box-Cox transformations

- $y_t^\lambda$ for $\lambda$ close to zero behaves like logs.
- If some $y_t = 0$, then must have $\lambda > 0$
- If some $y_t < 0$, no power transformation is possible unless all $y_t$ adjusted by adding a constant to all values.
- Choose a simple value of $\lambda$. It makes explanation easier.
- Results are relatively insensitive to value of $\lambda$
- Often no transformation ($\lambda = 1$) needed.
- Transformation often makes little difference to forecasts but has large effect on PI.
- Choosing $\lambda = 0$ is a simple way to force forecasts to be positive.
Back-transformation

We must reverse the transformation (or back-transform) to obtain forecasts on the original scale. The reverse Box-Cox transformations are given by

\[ y_t = \begin{cases} 
\exp(w_t), & \lambda = 0; \\
(\lambda W_t + 1)^{1/\lambda}, & \lambda \neq 0.
\end{cases} \]

plot(BoxCox(elec, lambda=1/3))
fit <- snaive(elec, lambda=1/3)
plot(fit)
plot(fit, include=120)
Back-transformation

We must reverse the transformation (or back-transform) to obtain forecasts on the original scale. The reverse Box-Cox transformations are given by

\[
y_t = \begin{cases} 
\exp(w_t), & \lambda = 0; \\
(\lambda W_t + 1)^{1/\lambda}, & \lambda \neq 0.
\end{cases}
\]

```r
plot(BoxCox(elec, lambda=1/3))
fit <- snaive(elec, lambda=1/3)
plot(fit)
plot(fit, include=120)
```
Automated Box-Cox transformations

BoxCox.lambda(elec)

- This attempts to balance the seasonal fluctuations and random variation across the series.
- Always check the results.
- A low value of $\lambda$ can give extremely large prediction intervals.
BoxCox.lambda(elec)

- This attempts to balance the seasonal fluctuations and random variation across the series.
- Always check the results.
- A low value of $\lambda$ can give extremely large prediction intervals.
Automated Box-Cox transformations

BoxCox.lambda(elec)

- This attempts to balance the seasonal fluctuations and random variation across the series.

- **Always check the results.**

- A low value of $\lambda$ can give extremely large prediction intervals.
Automated Box-Cox transformations

BoxCox.lambda(elec)

- This attempts to balance the seasonal fluctuations and random variation across the series.
- Always check the results.
- A low value of $\lambda$ can give extremely large prediction intervals.
ETS and transformations

- A Box-Cox transformation followed by an additive ETS model is often better than an ETS model without transformation.

- A Box-Cox transformation followed by STL + ETS is often better than an ETS model without transformation.

- It makes no sense to use a Box-Cox transformation and a non-additive ETS model.
A Box-Cox transformation followed by an additive ETS model is often better than an ETS model without transformation.

A Box-Cox transformation followed by STL + ETS is often better than an ETS model without transformation.

It makes no sense to use a Box-Cox transformation and a non-additive ETS model.
A Box-Cox transformation followed by an additive ETS model is often better than an ETS model without transformation.

A Box-Cox transformation followed by STL + ETS is often better than an ETS model without transformation.

It makes no sense to use a Box-Cox transformation and a *non-additive* ETS model.
1 Exponential smoothing

2 Transformations

3 Adjustments
Some of the variation in a time series may be due to the variation in the number of trading days each month. It is a good idea to adjust for this known source of variation to allow study of other interesting features.

- Month length
- Trading day
Calendar adjustments

Some of the variation in a time series may be due to the variation in the number of trading days each month. It is a good idea to adjust for this known source of variation to allow study of other interesting features.

- Month length
- Trading day
Month length adjustment

If this is not removed, it shows up as a seasonal effect, which may not cause problems though it does make any seasonal pattern hard to interpret. It is easily adjusted for:

\[
y_t^* = y_t \times \frac{\text{no. of days in an average month}}{\text{no. of days in month } t} = y_t \times \frac{365.25/12}{\text{no. of days in month } t}
\]

where \( y_t \) has already been transformed if necessary.

\textit{monthdays} gives the number of days in each month or quarter.
Month length adjustment

If this is not removed, it shows up as a seasonal effect, which may not cause problems though it does make any seasonal pattern hard to interpret. It is easily adjusted for:

\[ y_t^* = y_t \times \frac{\text{no. of days in an average month}}{\text{no. of days in month } t} \]

\[ = y_t \times \frac{365.25/12}{\text{no. of days in month } t} \]

where \( y_t \) has already been transformed if necessary.

\text{monthdays} gives the number of days in each month or quarter.
Trading day adjustment

- occurs in monthly data when there is also a weekly cycle, since proportions of various days in given month vary from year to year.

- number of trading days is predictable, but effects of various days are unknown.

- **Simplest case:** All trading days assumed to have same effect.

\[ y_t^* = y_t \times \frac{\text{no. of trading days in average month}}{\text{no. of trading days in month } t}. \]

where \( y_t \) has already been adjusted for month length and transformed if necessary.

- If weekly cycle more complex, these effects must be estimated from data.
Trading day adjustment

- occurs in monthly data when there is also a weekly cycle, since proportions of various days in given month vary from year to year.
- number of trading days is predictable, but effects of various days are unknown.

Simplest case: All trading days assumed to have same effect.

\[ y_t^* = y_t \times \frac{\text{no. of trading days in average month}}{\text{no. of trading days in month } t}. \]

where \( y_t \) has already been adjusted for month length and transformed if necessary.

- If weekly cycle more complex, these effects must be estimated from data.
Trading day adjustment

- occurs in monthly data when there is also a weekly cycle, since proportions of various days in given month vary from year to year.
- number of trading days is predictable, but effects of various days are unknown.
- **Simplest case:** All trading days assumed to have same effect.

\[ y_t^* = y_t \times \frac{\text{no. of trading days in average month}}{\text{no. of trading days in month } t}. \]

where \( y_t \) has already been adjusted for month length and transformed if necessary.

- If weekly cycle more complex, these effects must be estimated from data.
Trading day adjustment

- occurs in monthly data when there is also a weekly cycle, since proportions of various days in given month vary from year to year.
- number of trading days is predictable, but effects of various days are unknown.
- **Simplest case:** All trading days assumed to have same effect.

\[ y_t^* = y_t \times \frac{\text{no. of trading days in average month}}{\text{no. of trading days in month } t} . \]

where \( y_t \) has already been adjusted for month length and transformed if necessary.

- If weekly cycle more complex, these effects must be estimated from data.
Explainable variation

Examples:

- Calendar variation
- Increasing population
- Inflation
- Strikes
- Changes in government
- Changes in law

Try and understand all possible sources of variation before modelling the time series.
Explainable variation

Examples:

- Calendar variation
- Increasing population
- Inflation
- Strikes
- Changes in government
- Changes in law

Try and understand all possible sources of variation before modelling the time series.
Explainable variation

Examples:

- Calendar variation
- Increasing population
- Inflation
- Strikes
- Changes in government
- Changes in law

Try and understand all possible sources of variation before modelling the time series.
Explainable variation

Examples:

- Calendar variation
- Increasing population
- Inflation
- Strikes
- Changes in government
- Changes in law

Try and understand all possible sources of variation before modelling the time series.
Explainable variation

Examples:

- Calendar variation
- Increasing population
- Inflation
- Strikes
- Changes in government
- Changes in law

Try and understand all possible sources of variation before modelling the time series.
Explainable variation

Examples:
- Calendar variation
- Increasing population
- Inflation
- Strikes
- Changes in government
- Changes in law

Try and understand all possible sources of variation before modelling the time series.
Explainable variation

Examples:

- Calendar variation
- Increasing population
- Inflation
- Strikes
- Changes in government
- Changes in law

Try and understand all possible sources of variation before modelling the time series.
Explainable variation

Examples:

- Calendar variation
- Increasing population
- Inflation
- Strikes
- Changes in government
- Changes in law

Try and understand all possible sources of variation before modelling the time series.