# `pybats-detection`: A python package for outlier and structural changes detection in time series analysis

André Menezes and Eduardo Gabriel

28 janeiro, 2022

## Contents



## Smoothing

A brief introduction of the Smoothing class in a simulated example. A time series $\mathbf{Y} = (y_1, \ldots, y_T)$ was generated using the *RandomDLM* class which has the arguments (n, V, W): the number of observations, observational variance and state vector variance. This class has three methods that simulate data using different mechanisms:

- *.level*: dynamic level model;
- *.growth*: dynamic growth model;
- *.level_with_covariates*: dynamic level model where $Y$ is simulated given $X$, a matrix of fixed covariates.

For now, we stick with *.level*, simulating $n = 100$ observations with both observational and state vector variance equals to one 1, the starting level is set to 100. The simulated data is plotted below.

```
>>> # Generating level data model
>>> np.random.seed(66)
>>> rdlm = RandomDLM(n=100, V=1, W=1)
>>> df_simulated = rdlm.level(
>>>     start_level=100,
>>>     dict_shift={})
>>> y = df_simulated["y"]
```

The Smoothing class allows you to perform a retrospective analysis for $\mathbf{Y}$, obtaining the distribution of $(\boldsymbol{\theta}_{T-k}|D_T)$, for $k \geq 1$, the k-step smoothed distribution for the state vector at time $T$, which is analogous to the k-step ahead forecast distribution $(\boldsymbol{\theta}_{t+k}|D_t)$.

To use Smoothing, first it is necessary to define the model components with prior values, which is done with the *dlm* class available in the *pybats* package. In this case, it was considered a DLM with level and growth.

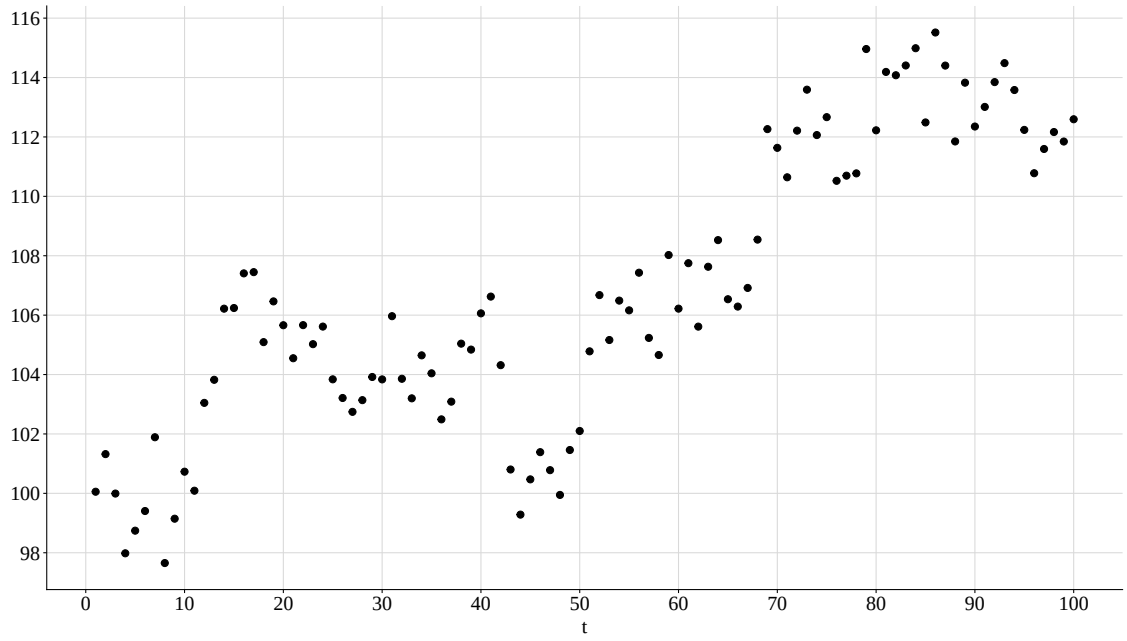



Figure 1: Simulated data

The prior vector and covariances are defined by **a** and **R**. Lastly, the discount factor denoted by *deltrend* is a constant in the interval (0, 1), which is used to coordinate the adaptive capacity of predictions with increasing variance of model components.

Given this, the method $.fit$ will initialize the model and the loop forecast, observe and update begin. The prior and posterior moments $(\mathbf{a}_t, \mathbf{m}_t, \mathbf{C}_t, \mathbf{R}_t)$ will be computed for all $t$ and saved. Subsequently, these moments will be used to obtain the moments for $(\boldsymbol{\theta}_{T-k}|D_T)$, recursively with $k \geq 1$, and denoted by $(\mathbf{a}_T(-k), \mathbf{m}_T(-k), \mathbf{C}_T(-k), \mathbf{R}_T(-k))$.

```
>>> # Define model components
>>> a = np.array([100, 0])
>>> R = np.eye(2)
>>> np.fill_diagonal(R, val=1)
>>> mod = dlm(a, R, ntrend=2, deltrend=.95)
>>>
>>> # Fit with monitoring
>>> smooth = Smoothing(mod=mod)
>>> smooth_fit = smooth.fit(y=y)
```

This will return a dictionary with moments for: smoothed and filtered predictive distributions and for the posterior distributions of the model components. Below the results for the predictive distributions

## smoothed predictive

The results for the smoothed predictive distribution consists of: $f_T(-k), q_T(-k)$ and the bounds for the credibility interval $(ci\_lower, ci\_upper)$. Given by

$$f_T(-k) = \mathbf{F}^{'}\mathbf{a}_T(-k), \qquad q_T(-k) = \mathbf{F}^{'}\mathbf{R}_T(-k)\mathbf{F}$$

The credibility interval is is obtained from the corresponding smoothed distributions for the mean response of the series. Since $V$ is considered unknown, then

$$(\mu_T(-k)|D_T) \sim T_{n_T}[f_T(-k), q_T(-k)]$$

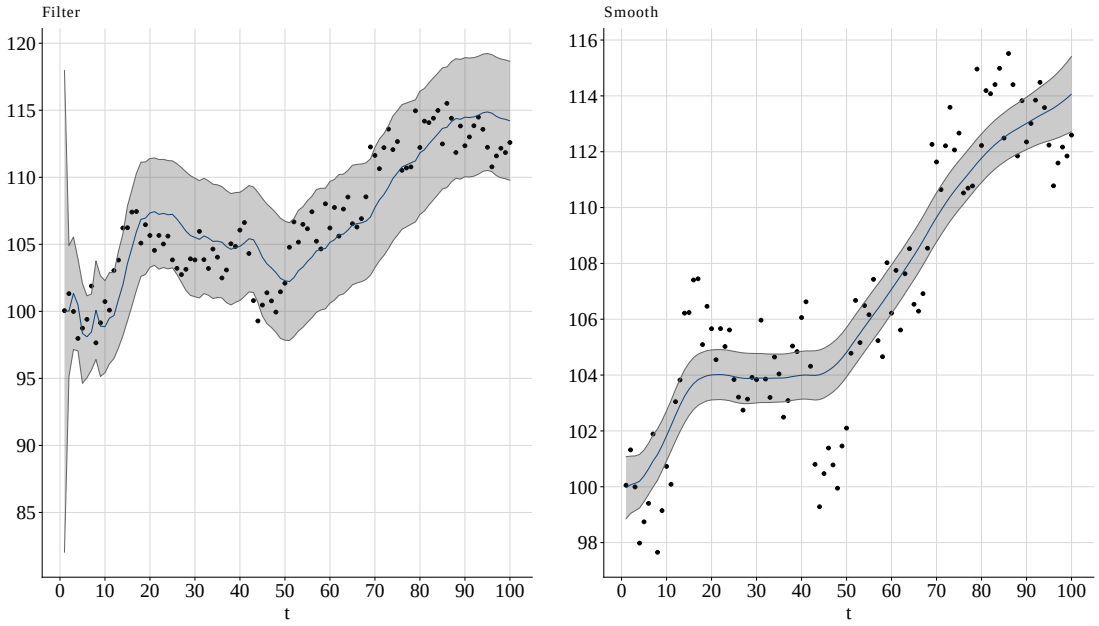
For this simulated example, the results for the smoothed predictive distribution for the mean response are

```
>>> smooth_fit.get('smooth').get('predictive').round(2).head()
```

Table 1: Smothed predictive distribution results

| fk | qk | t | df | ci_lower | ci_upper |
|---|---|---|---|---|---|
| 99.96 | 0.31 | 1 | 1 | 98.85 | 101.1 |
| 100.07 | 0.27 | 2 | 2 | 99.04 | 101.1 |
| 100.12 | 0.24 | 3 | 3 | 99.14 | 101.1 |
| 100.20 | 0.23 | 4 | 4 | 99.24 | 101.2 |
| 100.39 | 0.22 | 5 | 5 | 99.47 | 101.3 |

Plotting the filtered vs smoothed predictive distributions results is possible to see difference, primarily in the length of the credibility interval.




## smoothed posterior

The results for the posterior distributions are analogous, where

- parameter: Indicator for the respective state space parameter in $\boldsymbol{\theta}$;
- mean: The smoothed posterior distribution mean for time $t = T - k$ ($\mathbf{m}(-k)$);
- variance: The smoothed posterior distribution variance for time $t$ ($\mathbf{C}(-k)$).
- credibility interval ($ci\_lower, ci\_upper$): The credibility interval obtained from the corresponding smoothed posterior distributions. Since $V$ is considered unknown, then

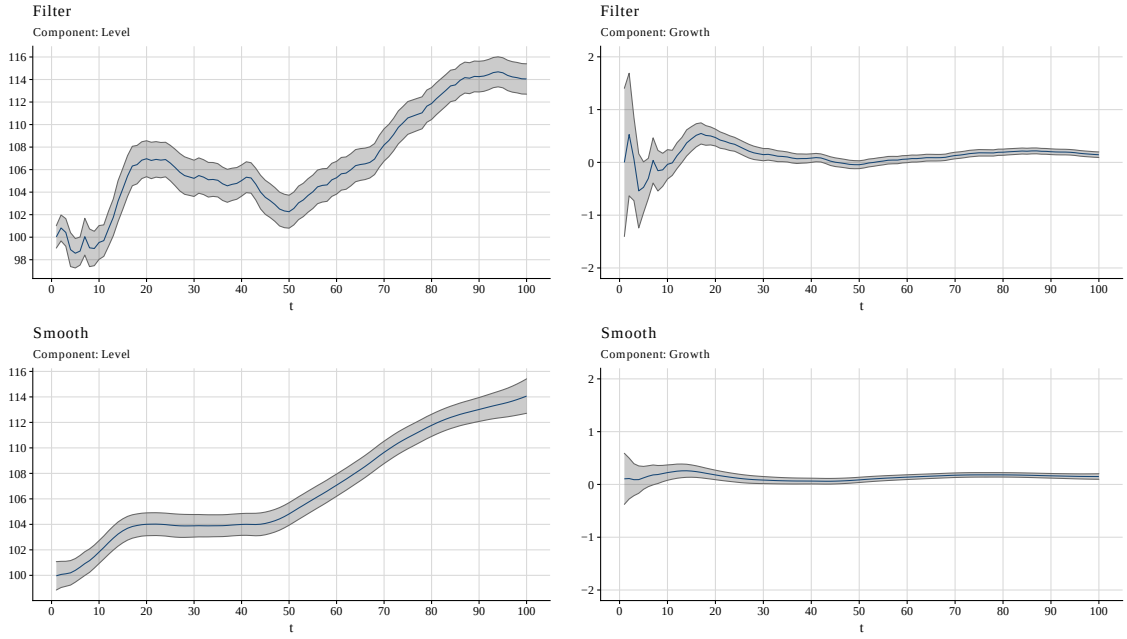
$$(\boldsymbol{\theta}_{T-k}|D_T) \sim T_{n_T}[\mathbf{a}_T(-k), \mathbf{R}_T(-k)].$$

```
>>> smooth_fit.get('smooth').get('posterior').round(2).head()
```

Table 2: Smothed posterior distribution results

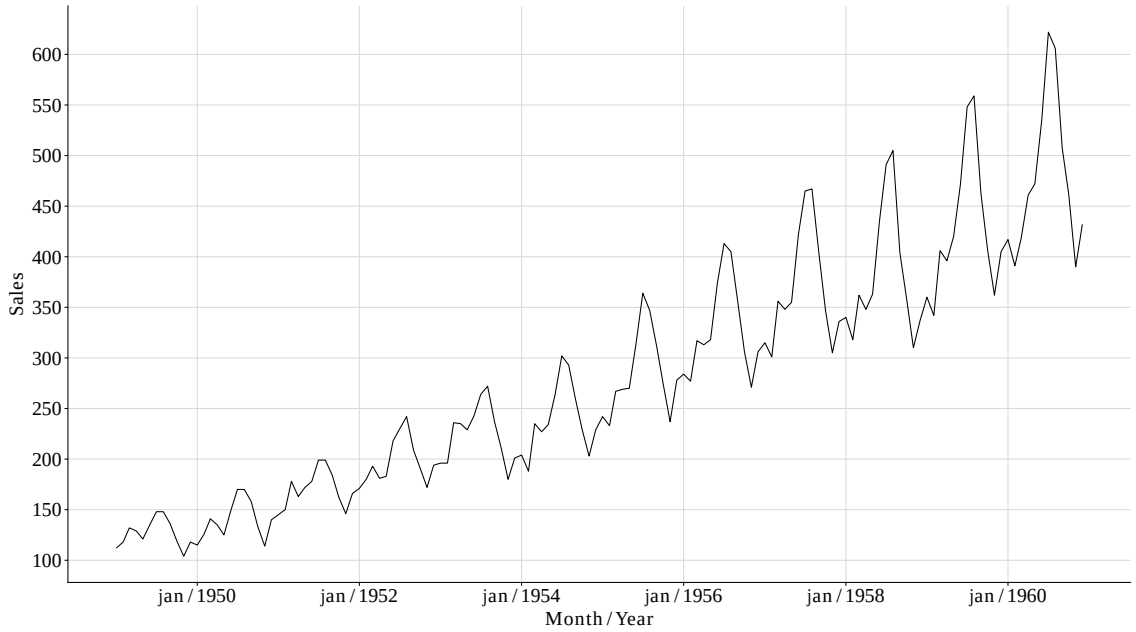
| parameter | mean | variance | t | ci_lower | ci_upper |
|---|---|---|---|---|---|
| theta_1 | 99.96 | 0.31 | 1 | 98.85 | 101.1 |
| theta_1 | 100.07 | 0.27 | 2 | 99.04 | 101.1 |
| theta_1 | 100.12 | 0.24 | 3 | 99.14 | 101.1 |
| theta_1 | 100.20 | 0.23 | 4 | 99.24 | 101.2 |
| theta_1 | 100.39 | 0.22 | 5 | 99.47 | 101.3 |

As before we plot the results for filtered and smoothed distributions, in this case for each state space parameter. As expected, the smoothed posterior distributions show a less erratic behavior with shorter credibility intervals.




## Aplication: AirPassangers dataset

Below is a practical example with the classic Box & Jenkins airline data, Monthly totals of international airline passengers (1949 to 1960). This data has a clear multiplicative seasonality, using a linear model (with additive seasonality) may be a naive approximation for this data. But, just for the sake of comparison between filtered and smoothing we stick with the linear model.

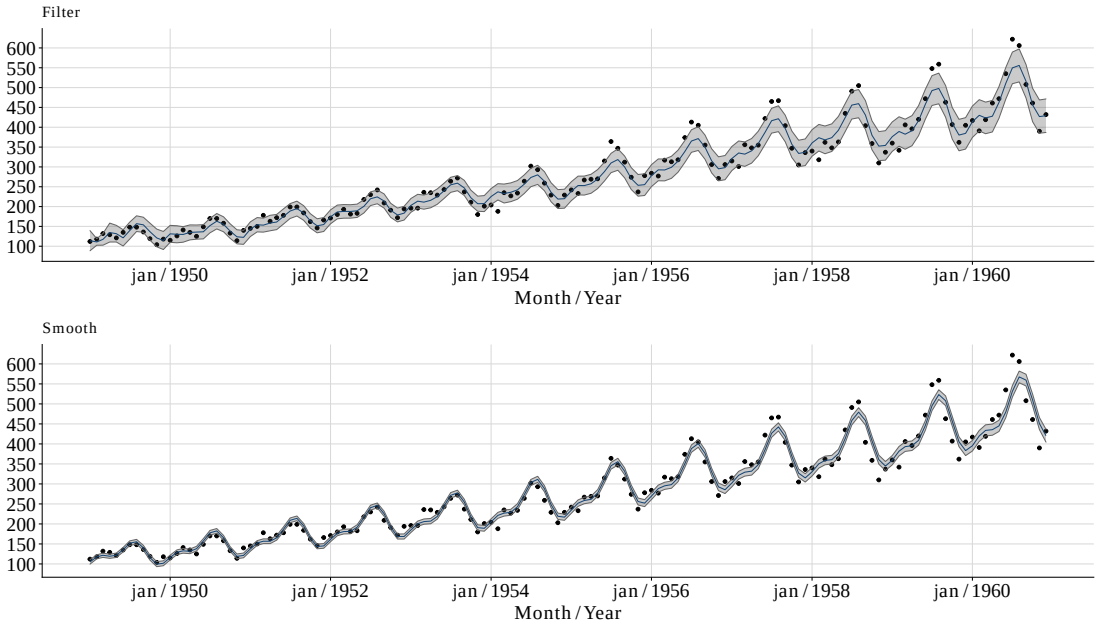



Using a normal DLM with three main components: Trend, Growth and Seasonality. The seasonality is modeled using the Fourier form representation, which depends on the parity of a period $p$ and the number of harmonics components. Formally, the $\mathbf{r}^{th}$ harmonic component is given by

$$S_r(.) = a_r \cos(\alpha r) + b_r \sin(\alpha r), \quad r = 1, \ldots, h, \quad a_r = 2\pi/p, \quad h <= p/2$$

Here it was specified a yearly seasonal effect with period $p = 12$ and the first two harmonics. The discount factor for the level and growth components is set to 0.95, and 0.98 for the seasonal components. The results are plotted below.

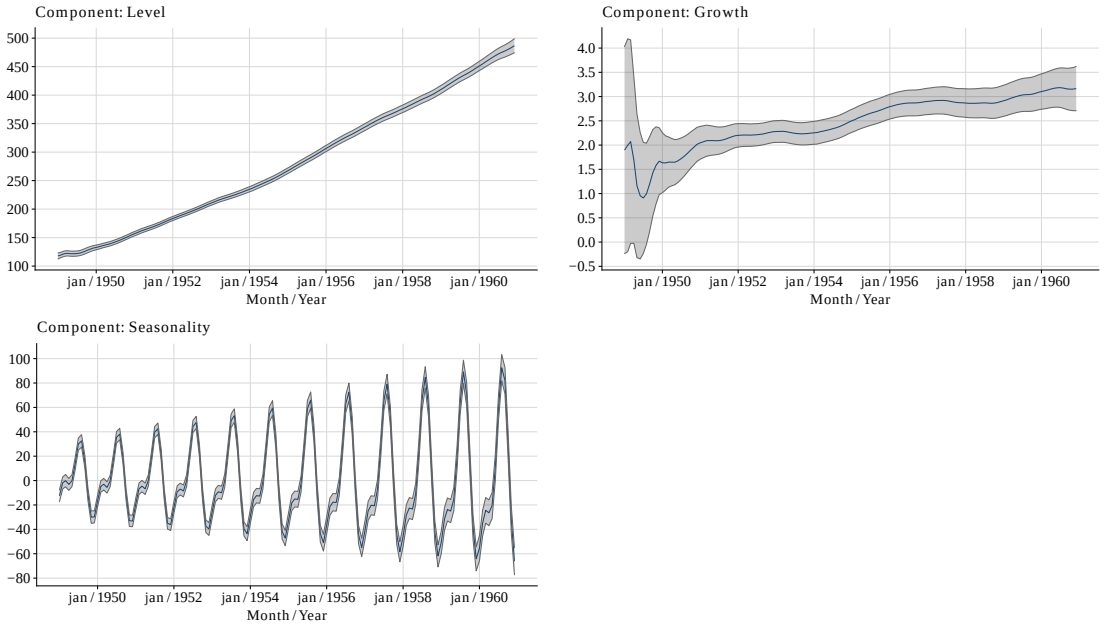
```
>>> a = np.array([112, 0, 1, -1, 1, -1])
>>> R = np.eye(6)
>>> np.fill_diagonal(R, val=1)
>>> mod = dlm(a, R, ntrend=2, deltrend=.95, delseas=.98,
>>>           seasPeriods=[12], seasHarmComponents=[[1, 2]])
```




Since the seasonality was modeled using harmonic components, the model has a total of six parameters: level, growth and four for seasonality $(a_1, b_1, a_2, b_2)$. For simplicity, the results for de posterior distributions considered the sum of the harmonic components, whose moments are given by

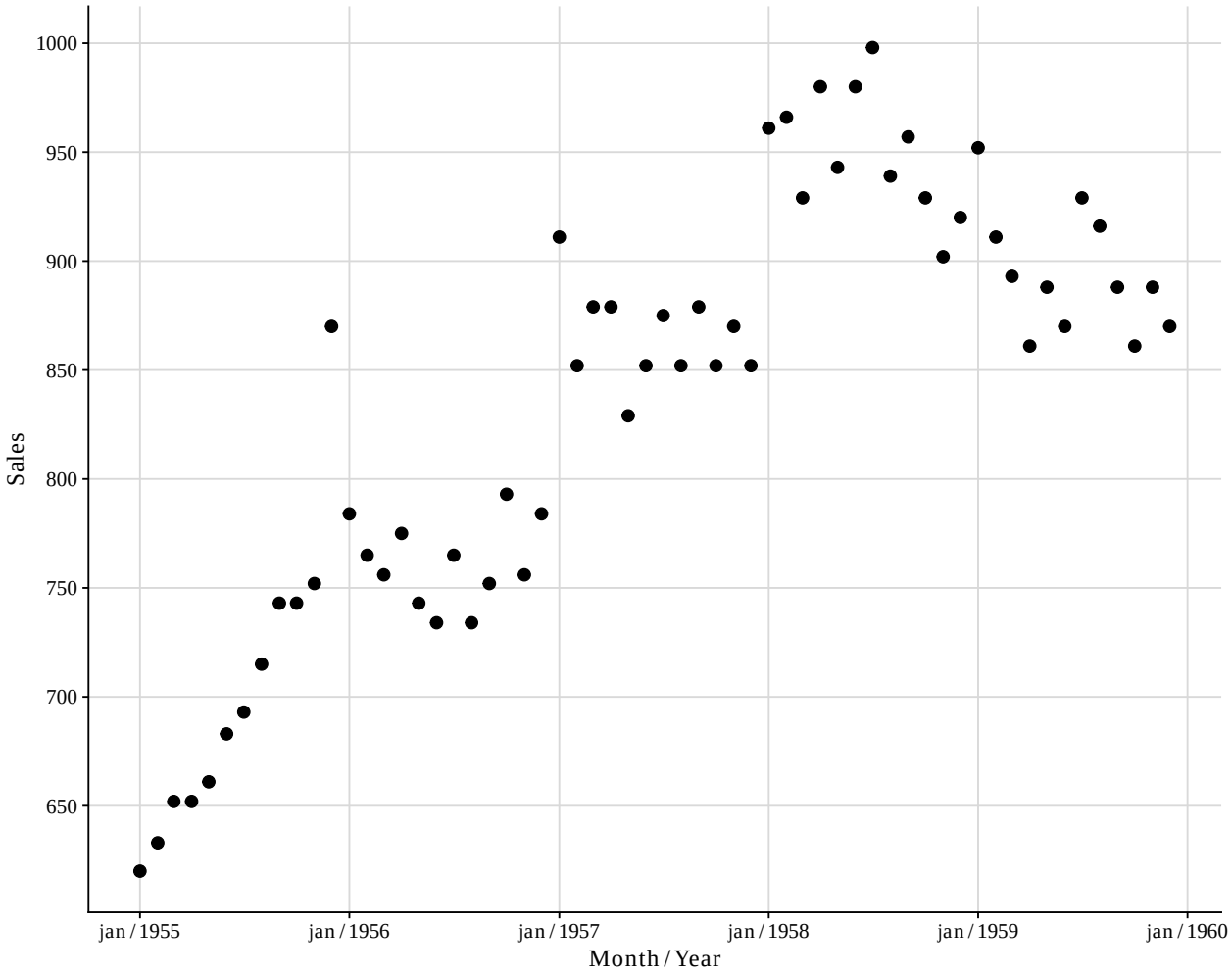
$$\mu_{seas} = F_{seas}^{\top}\mathbf{a}_T(-k), \qquad \sigma^2_{seas} = F_{seas}^{\top}\mathbf{R}_T(-k)F_{seas}$$

where $F_{seas}^{\top} = [0, 0, 1, 0, 1]$. The results are ploted below.



## Manual Intervention

### CP6




- Performing the fit (filter and smoothing) without interventions.

```
>>> # Define the growth model
>>> a = np.array([600, 1])
>>> R = np.array([[100, 0], [0, 25]])
>>> mod = dlm(a, R, ntrend=2, deltrend=[0.90, 0.98])
>>>
>>> # Filter and Smooth without intervention
>>> smooth = Smoothing(mod=mod)
>>> out_no_int = smooth.fit(y=cp6["sales"])
>>> dict_filter_no_int = out_no_int.get("filter")
>>> dict_smooth_no_int = out_no_int.get("smooth")
```

- Performing the fit (filter and smoothing) with interventions

```
>>> list_interventions = [
>>>     {"time_index": 12, "which": ["variance", "noise"],
>>>      "parameters": [{"v_shift": "ignore"},
>>>                     {"h_shift": np.array([0, 0]),
```

```
>>>                       "H_shift": np.array([[1000, 25], [25, 25]])}]
>>>      },
>>>     {"time_index": 25, "which": ["noise", "variance"],
>>>      "parameters": [{"h_shift": np.array([80, 0]),
>>>                      "H_shift": np.array([[100, 0], [0, 0]])},
>>>                     {"v_shift": "ignore"}]},
>>>     {"time_index": 37, "which": ["subjective"],
>>>      "parameters": [{"a_star": np.array([970, 0]),
>>>                      "R_star": np.array([[50, 0], [0, 5]])}]}
>>> ]
>>> manual_interventions = ManualIntervention(mod=mod)
>>> out_int = manual_interventions.fit(
>>>     y=cp6["sales"], interventions=list_interventions)
>>> dict_filter_int = out_int.get("filter")
>>> dict_smooth_int = out_int.get("smooth")
```

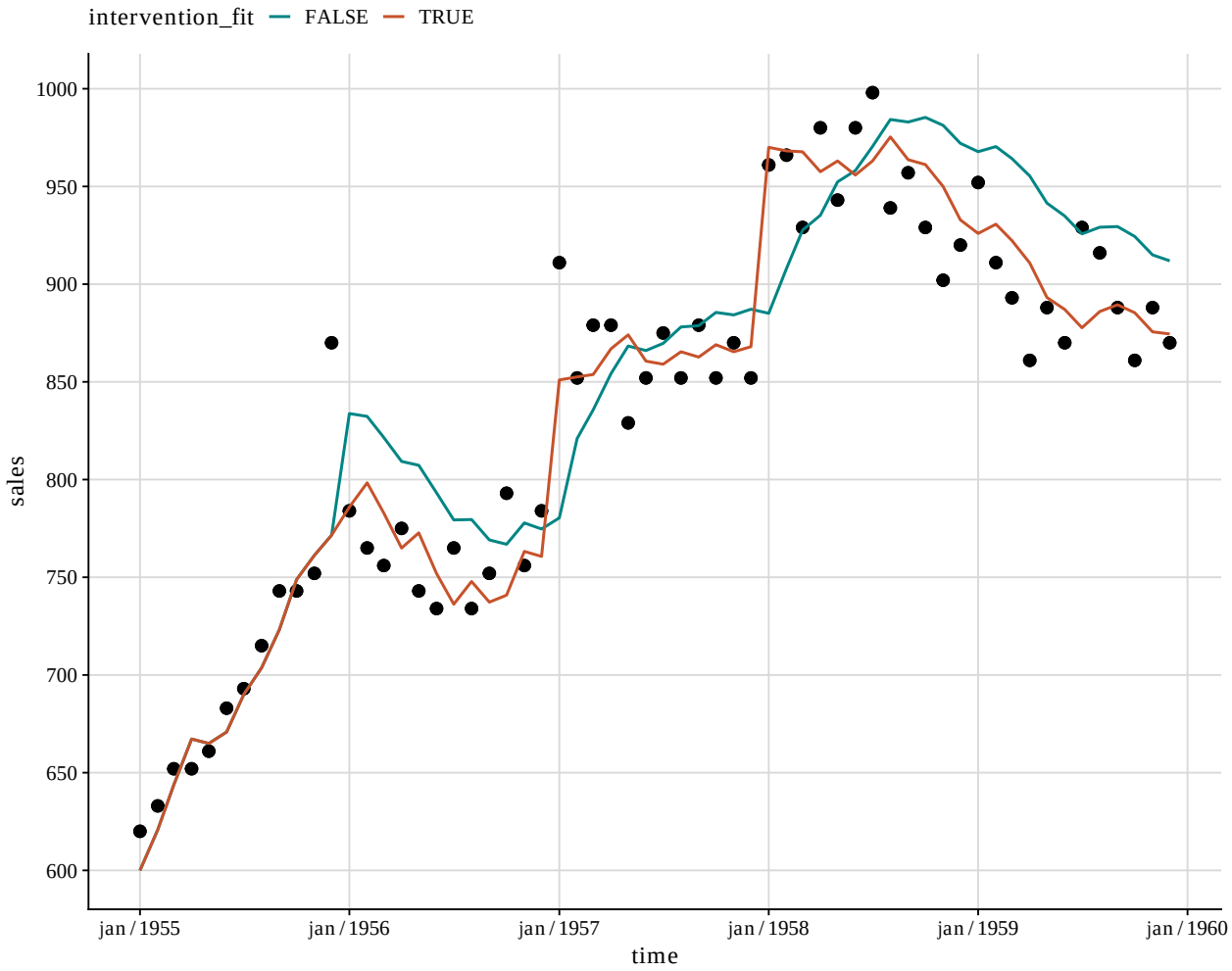
- Organizing the data into two data.frame: `data_predictive` and `data_posterior`.

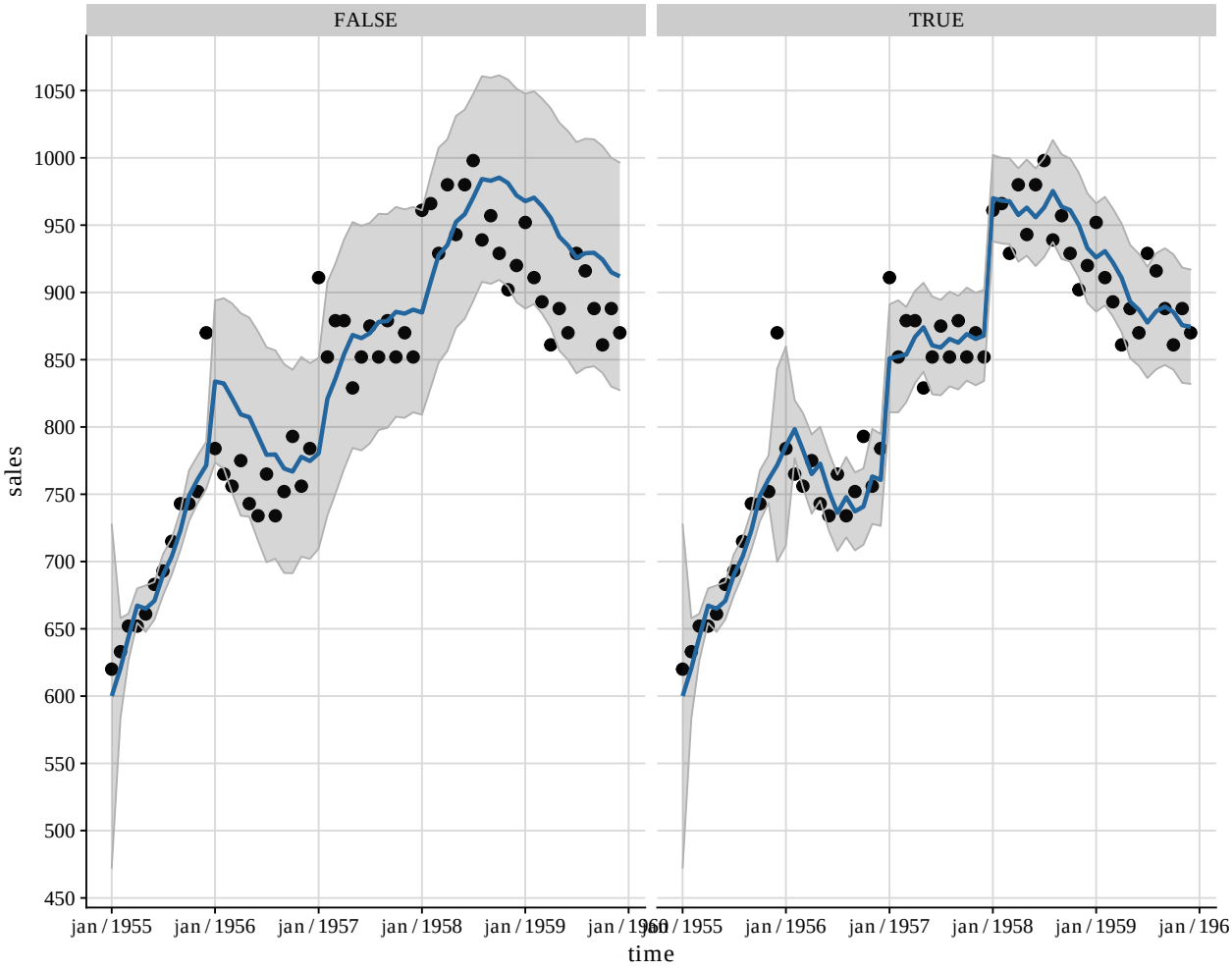
```
>>> # Filter
>>> data_predictive_filter_int = dict_filter_int.get("predictive").copy()
>>> data_predictive_filter_no_int = dict_filter_no_int.get("predictive").copy()
>>> data_predictive_filter_no_int["intervention_type"] = "nothing"
>>> data_predictive_filter_int["intervention_fit"] = True
>>> data_predictive_filter_no_int["intervention_fit"] = False
>>> cols = data_predictive_filter_int.columns
>>> data_predictive_filter = pd.concat(
>>>   [data_predictive_filter_int, data_predictive_filter_no_int[cols]]
>>> )
>>> # Smooth
>>> data_predictive_smooth_int = dict_smooth_int.get("predictive").copy()
>>> data_predictive_smooth_no_int = dict_smooth_no_int.get("predictive").copy()
>>> data_predictive_smooth_no_int["intervention_type"] = "nothing"
>>> data_predictive_smooth_int["intervention_fit"] = True
>>> data_predictive_smooth_no_int["intervention_fit"] = False
>>> cols = data_predictive_smooth_int.columns
>>> data_predictive_smooth = pd.concat(
>>>   [data_predictive_smooth_int, data_predictive_smooth_no_int[cols]]
>>> )
>>> # Append data
>>> data_predictive_smooth["type"] = "smooth"
>>> data_predictive_filter["type"] = "filter"
>>> data_predictive_smooth.rename(columns={"fk": "f", "qk": "q"}, inplace=True)
>>> cols_ord = ["t", "intervention_fit", "type", "f", "q", 'ci_lower', 'ci_upper']
>>> data_predictive = pd.concat(
>>>     [data_predictive_filter[cols_ord], data_predictive_smooth[cols_ord]])
>>> data_predictive = data_predictive.join(cp6)
```

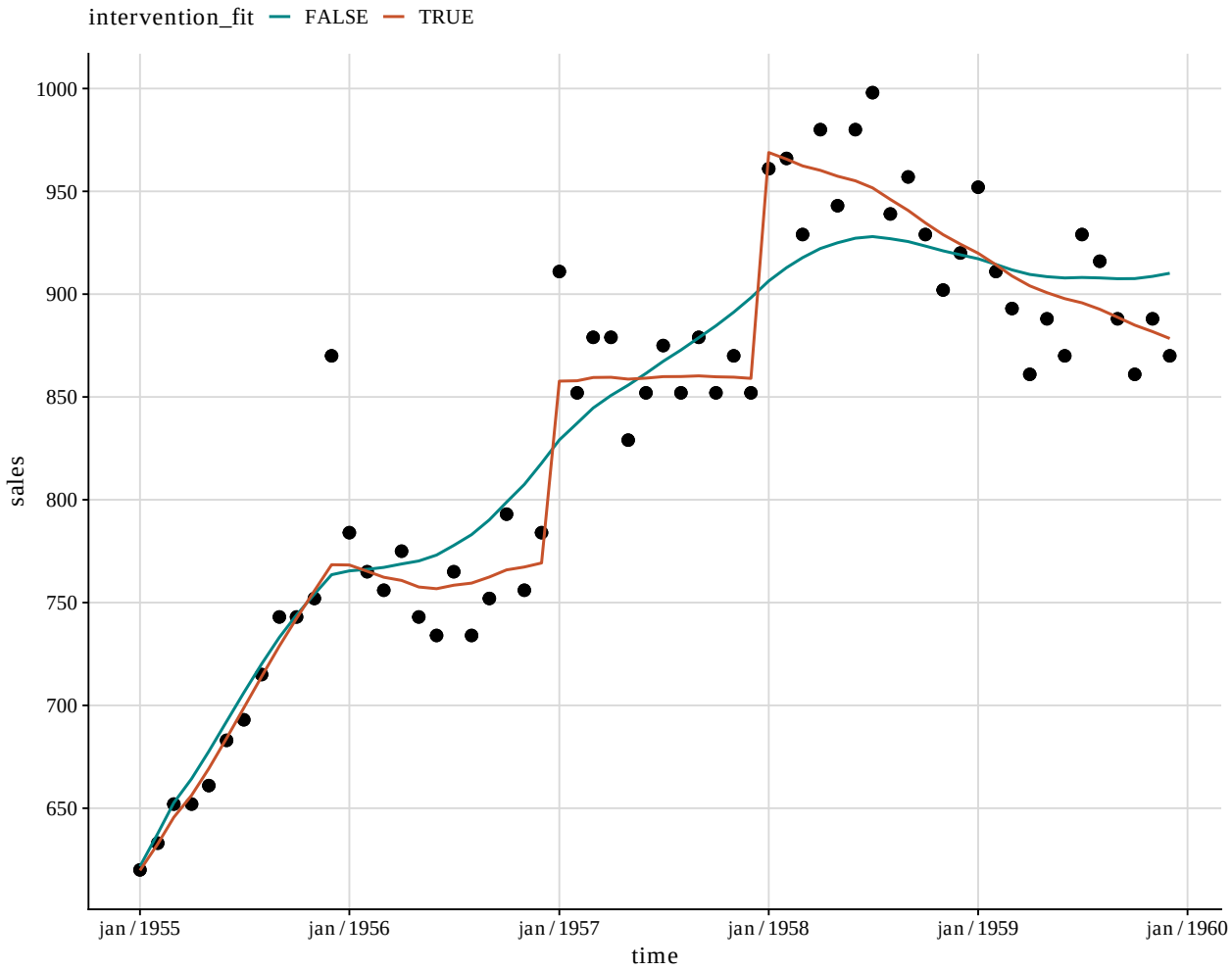
```
>>> # Filter
>>> data_posterior_filter_int = dict_filter_int.get("posterior").copy()
>>> data_posterior_filter_no_int = dict_filter_no_int.get("posterior").copy()
>>> data_posterior_filter_no_int["intervention_type"] = "nothing"
>>> data_posterior_filter_int["intervention_fit"] = True
>>> data_posterior_filter_no_int["intervention_fit"] = False
>>> cols = data_posterior_filter_int.columns
```

```
>>> data_posterior_filter = pd.concat(
>>>   [data_posterior_filter_int, data_posterior_filter_no_int[cols]]
>>> )
>>> # Smooth
>>> data_posterior_smooth_int = dict_smooth_int.get("posterior").copy()
>>> data_posterior_smooth_no_int = dict_smooth_no_int.get("posterior").copy()
>>> data_posterior_smooth_no_int["intervention_type"] = "nothing"
>>> data_posterior_smooth_int["intervention_fit"] = True
>>> data_posterior_smooth_no_int["intervention_fit"] = False
>>> cols = data_posterior_smooth_int.columns
>>> data_posterior_smooth = pd.concat(
>>>   [data_posterior_smooth_int, data_posterior_smooth_no_int[cols]]
>>> )
>>> # Append data
>>> data_posterior_smooth["type"] = "smooth"
>>> data_posterior_filter["type"] = "filter"
>>> data_posterior_filter.set_index(data_posterior_filter["t"].values-1, inplace=True)
>>> data_posterior_smooth.set_index(data_posterior_smooth["t"].values-1, inplace=True)
>>>
>>> cols_ord = ["t", "intervention_fit", "type", "parameter", "mean", "variance",
>>>             'ci_lower', 'ci_upper']
>>> data_posterior = pd.concat(
>>>     [data_posterior_filter[cols_ord], data_posterior_smooth[cols_ord]])
>>> data_posterior = data_posterior.join(cp6)
```

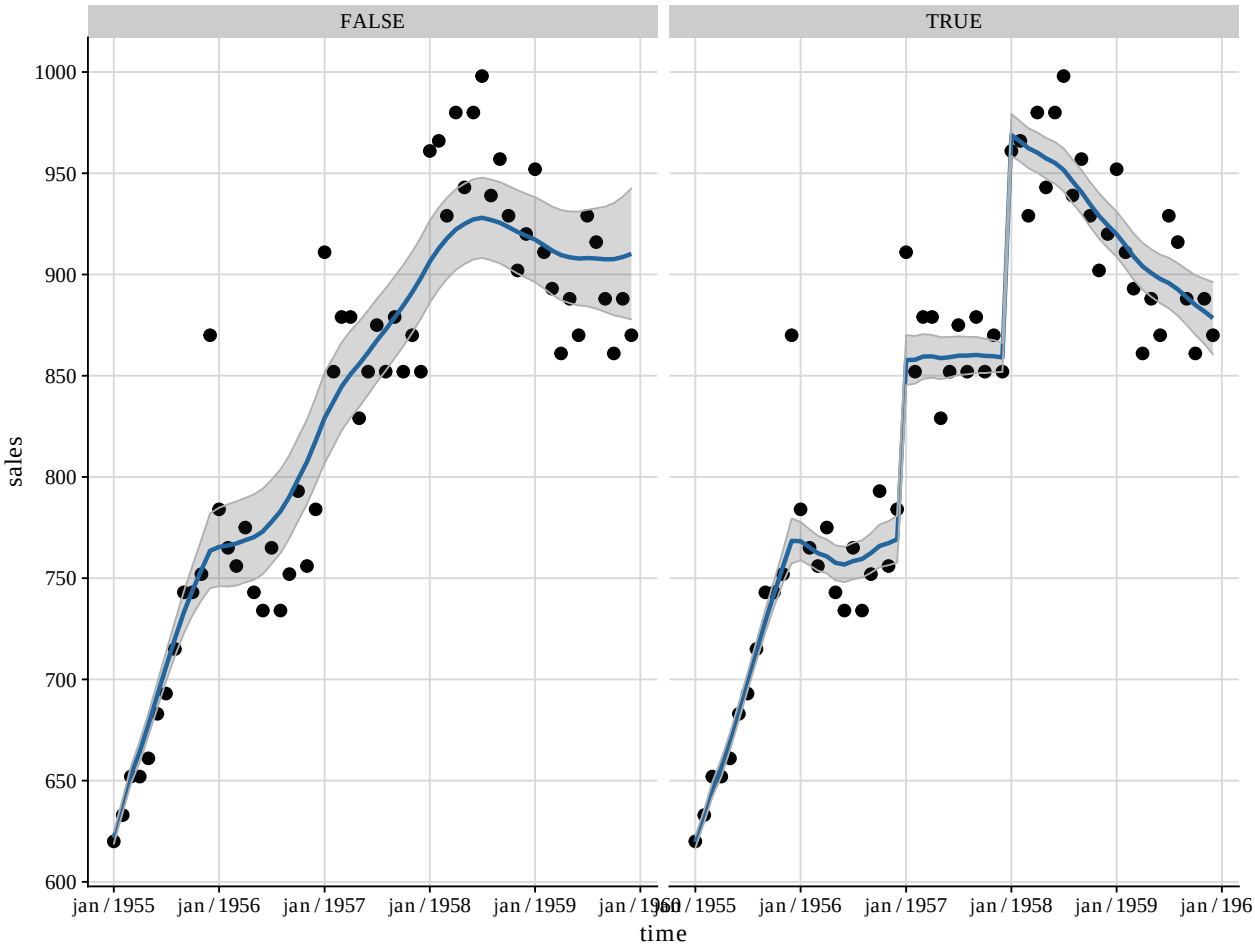
- Plotting some results


intervention_fit
FALSE
TRUE
1000
950
900
850
800
750
700
650
600
sales
jan / 1955
jan / 1956
jan / 1957
jan / 1958
jan / 1959
jan / 1960
time

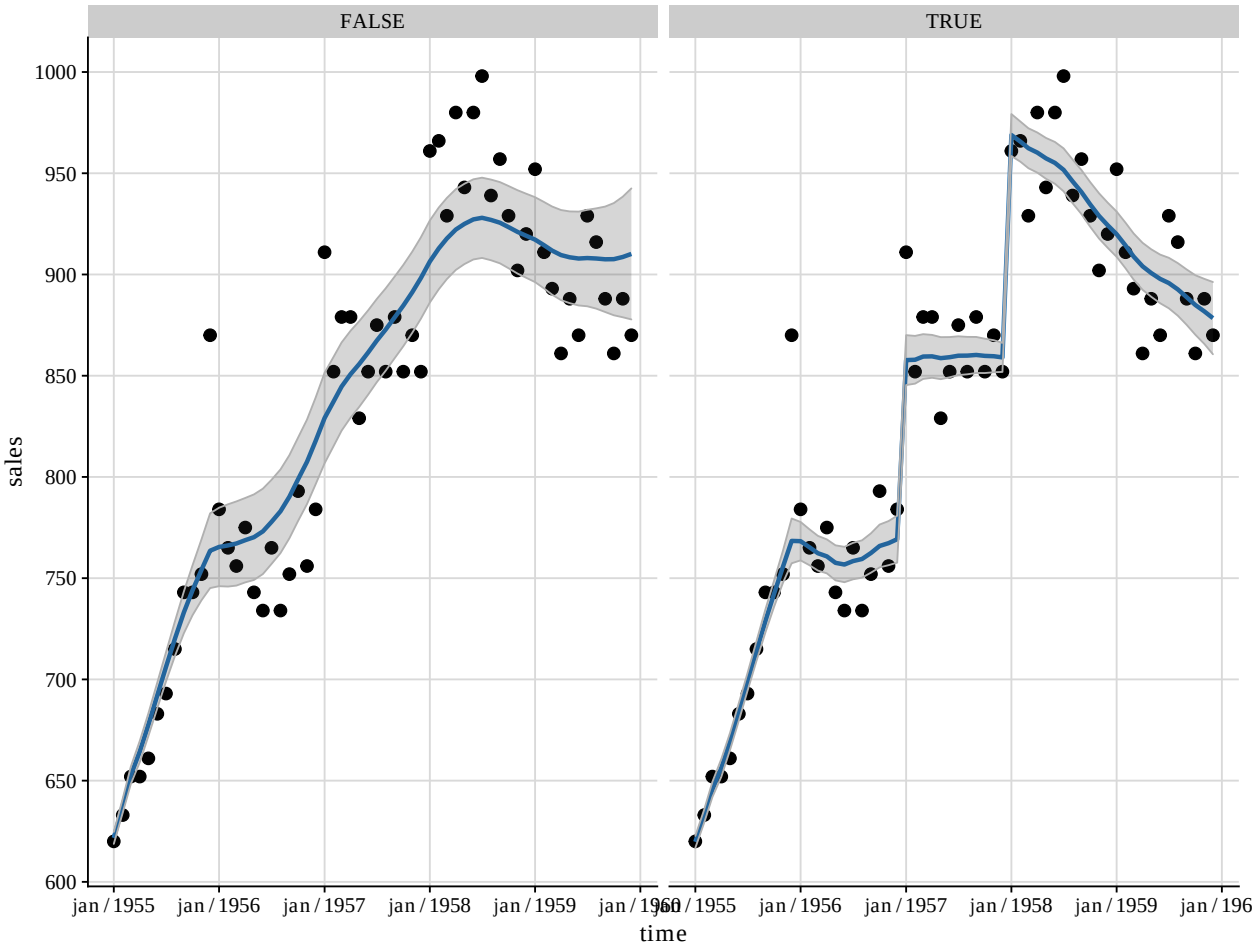

FALSE
TRUE
sales
1050
1000
950
900
850
800
750
700
650
600
550
500
450
jan / 1955
jan / 1956
jan / 1957
jan / 1958
jan / 1959
jan / 1960
jan / 196
time


intervention_fit
FALSE
TRUE
1000
950
900
850
800
750
700
650
sales
jan / 1955
jan / 1956
jan / 1957
jan / 1958
jan / 1959
jan / 1960
time

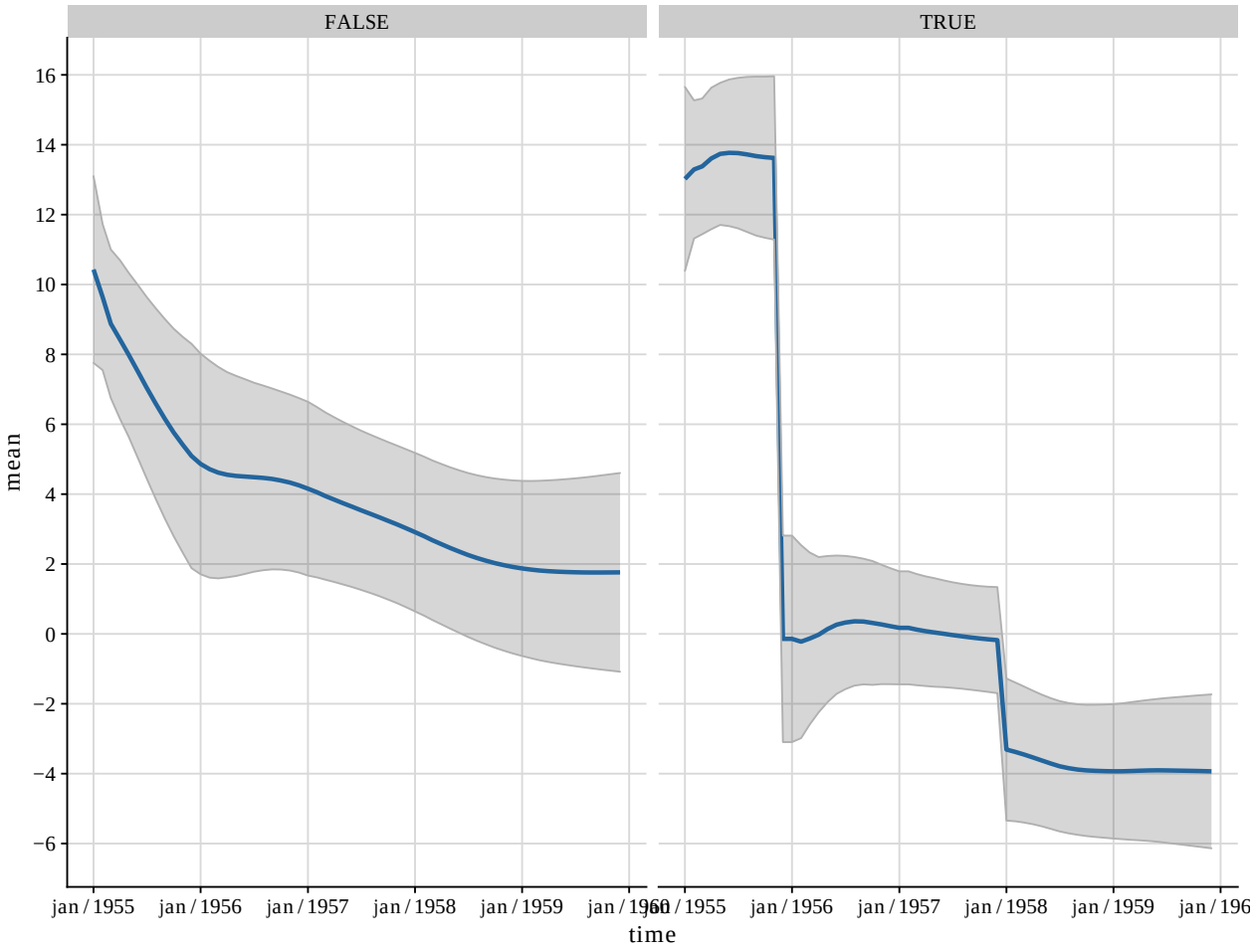
Smooth predictive with and without intervention

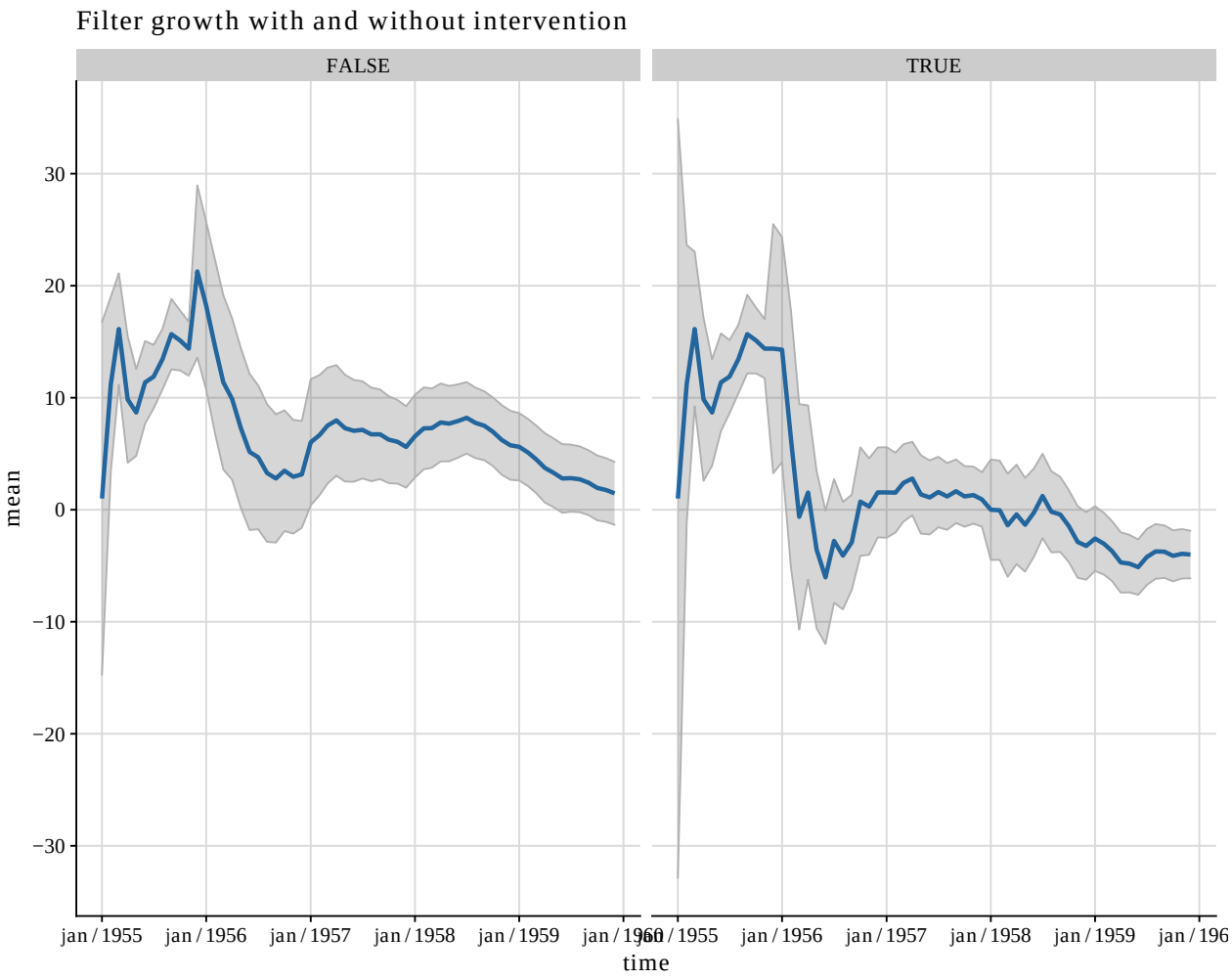


Smooth level with and without intervention



Smooth growth with and without intervention



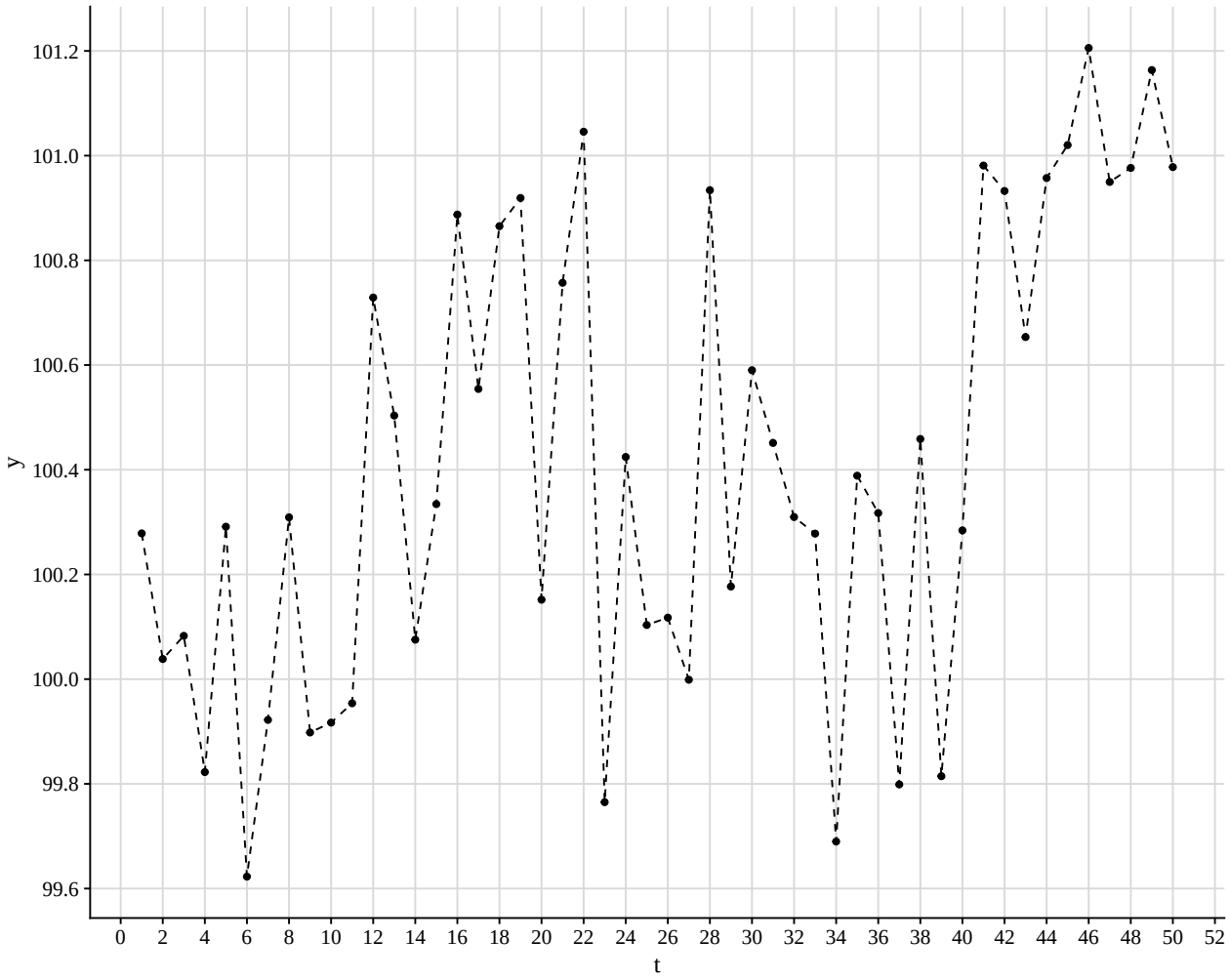



# Automatic Monitoring

## Simulated examples

### Level Change

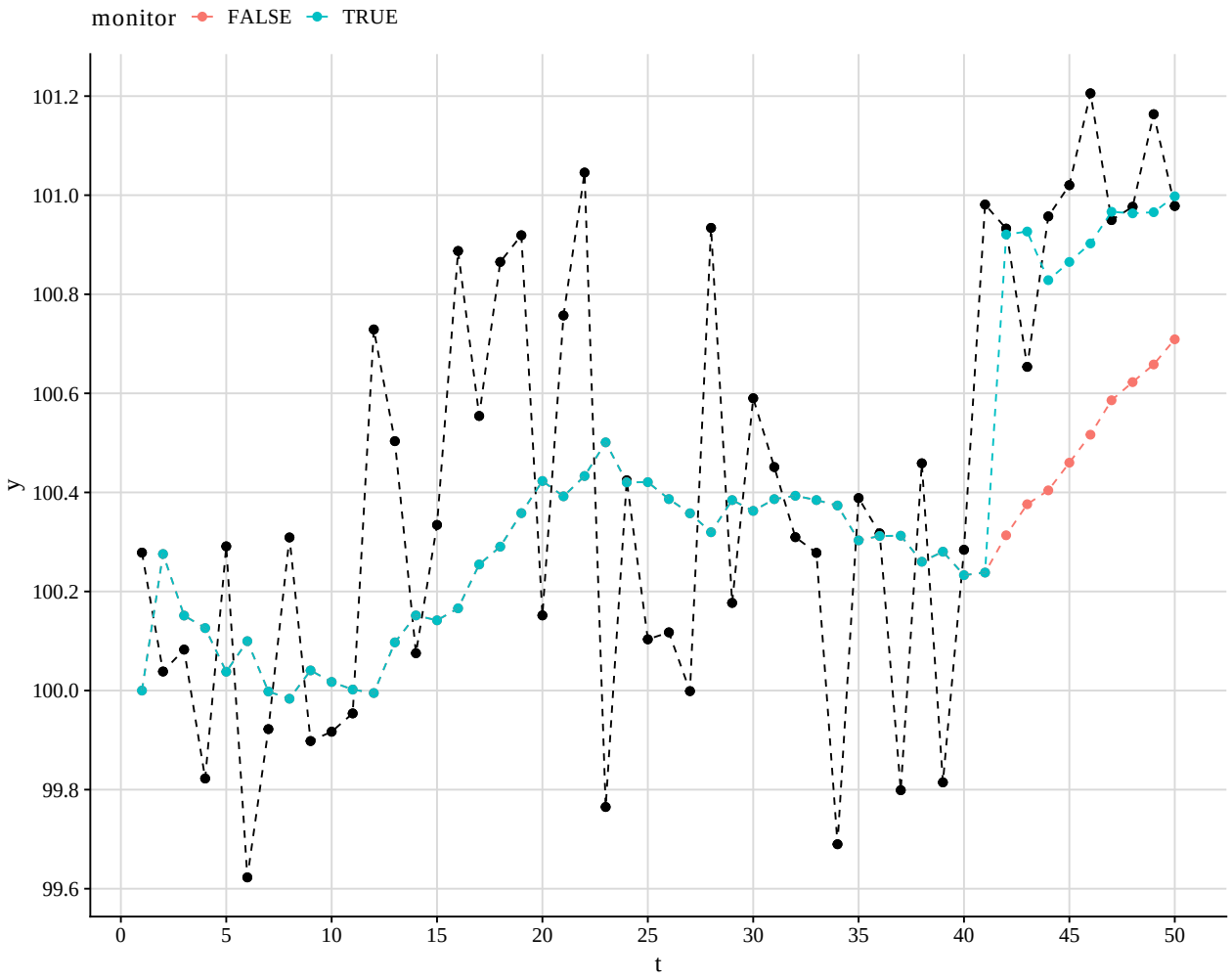
```
>>> np.random.seed(66)
>>> rdlm = RandomDLM(n=50, V=0.1, W=0.005)
>>> df_simulated = rdlm.level(
>>>     start_level=100,
>>>     dict_shift={"t": [40],
>>>                 "level_mean_shift": [1],
>>>                 "level_var_shift": [1]})
>>> df_simulated.loc[40:50, "y"] = 101 + np.random.normal(0, 0.2, 10)
```




```
>>> a = np.array([100])
>>> R = np.eye(1)
>>> R[[0]] = 100
>>> mod = dlm(a, R, ntrend=1, deltrend=0.9)
>>>
>>> # Fit without monitoring
>>> fit_without_monitor = Smoothing(mod=mod).fit(y=df_simulated["y"])
>>> df_res = fit_without_monitor.get("filter").get("predictive")
>>> df_res["monitor"] = False
>>>
>>> # Fit with monitoring
>>> monitor = AutomaticMonitoring(mod=mod, bilateral=False)
>>> fit_monitor = monitor.fit(y=df_simulated["y"], h=3, tau=0.135, change_var=[100])
```

```
## Parametric change detected at time 43 with H=12.0902, L=3.7693 and l=3
```

```
>>> df_tmp = fit_monitor.get("filter").get("predictive")
>>> df_tmp["monitor"] = True
>>> cols_ord = ["t", "y", "f", "q", "ci_lower", "ci_upper", "monitor", "e",
>>>             "H", "L", "l"]
>>> df_res = pd.concat([df_res, df_tmp[cols_ord]]).reset_index(drop=True)
```

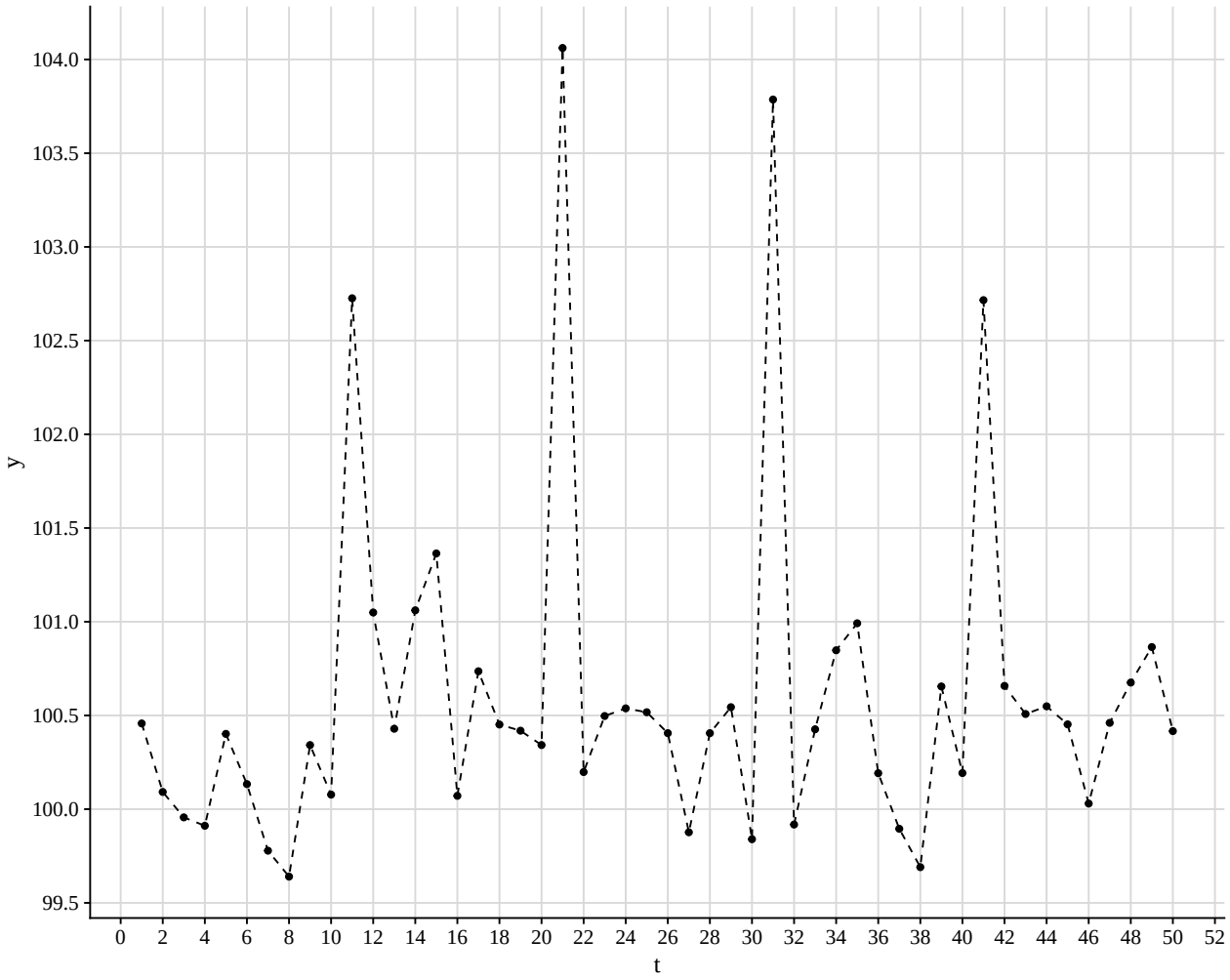



```
##     t      y      f        q         e         H         L l
## 1   1 100.28 100.00 101.0000  0.027708 1.000e+00 1.000e+00 1
## 2   2 100.04 100.28   1.0509 -0.231445 1.000e+00 1.000e+00 1
## 3   3 100.08 100.15   0.5419 -0.093343 1.000e+00 1.000e+00 1
## 4   4  99.82 100.13   0.3630 -0.504038 1.000e+00 1.000e+00 1
## 5   5 100.29 100.04   0.2899  0.470105 1.000e+00 1.000e+00 1
## 6   6  99.62 100.10   0.2424 -0.968590 1.000e+00 1.000e+00 1
## 7   7  99.92 100.00   0.2338 -0.156969 1.000e+00 1.000e+00 1
## 8   8 100.31  99.98   0.2013  0.725492 1.000e+00 1.000e+00 1
## 9   9  99.90 100.04   0.1879 -0.328868 1.000e+00 1.000e+00 1
## 10 10  99.92 100.02   0.1692 -0.244168 1.000e+00 1.000e+00 1
## 11 11  99.95 100.00   0.1533 -0.122789 1.301e+02 1.301e+02 1
## 12 12 100.73  99.99   0.1397  1.963608 2.489e-01 2.489e-01 1
## 13 13 100.50 100.10   0.1694  0.987129 4.658e+00 1.159e+00 2
## 14 14 100.08 100.15   0.1682 -0.185936 1.572e+02 1.572e+02 1
## 15 15 100.33 100.14   0.1567  0.486534 2.091e+01 2.091e+01 1
## 16 16 100.89 100.17   0.1487  1.870796 3.288e-01 3.288e-01 1
## 17 17 100.55 100.25   0.1700  0.726979 1.017e+01 3.342e+00 2
## 18 18 100.87 100.29   0.1651  1.414333 1.293e+00 1.293e+00 1
## 19 19 100.92 100.36   0.1734  1.346991 1.583e+00 1.583e+00 1
## 20 20 100.15 100.42   0.1801 -0.639038 6.122e+02 6.122e+02 1
## 21 21 100.76 100.39   0.1747  0.873260 6.555e+00 6.555e+00 1
```

```
## 22 22 101.05 100.43   0.1726  1.474577 1.079e+00 1.079e+00 1
## 23 23  99.77 100.50   0.1812 -1.729299 1.612e+04 1.612e+04 1
## 24 24 100.42 100.42   0.1959  0.009252 8.755e+01 8.755e+01 1
## 25 25 100.10 100.42   0.1879 -0.731956 8.091e+02 8.091e+02 1
## 26 26 100.12 100.39   0.1844 -0.626588 5.898e+02 5.898e+02 1
## 27 27 100.00 100.36   0.1801 -0.845808 1.138e+03 1.138e+03 1
## 28 28 100.93 100.32   0.1781  1.455528 1.143e+00 1.143e+00 1
## 29 29 100.18 100.38   0.1849 -0.482666 3.830e+02 3.830e+02 1
## 30 30 100.59 100.36   0.1801  0.535852 1.804e+01 1.804e+01 1
## 31 31 100.45 100.39   0.1758  0.154334 5.666e+01 5.666e+01 1
## 32 32 100.31 100.39   0.1704 -0.202337 1.652e+02 1.652e+02 1
## 33 33 100.28 100.38   0.1654 -0.262065 1.976e+02 1.976e+02 1
## 34 34  99.69 100.37   0.1608 -1.705145 1.499e+04 1.499e+04 1
## 35 35 100.39 100.30   0.1695  0.207462 4.831e+01 4.831e+01 1
## 36 36 100.32 100.31   0.1649  0.013050 8.656e+01 8.656e+01 1
## 37 37  99.80 100.31   0.1604 -1.282407 4.218e+03 4.218e+03 1
## 38 38 100.46 100.26   0.1631  0.491867 2.058e+01 2.058e+01 1
## 39 39  99.81 100.28   0.1599 -1.164556 2.962e+03 2.962e+03 1
## 40 40 100.28 100.23   0.1613  0.127470 6.141e+01 6.141e+01 1
## 41 41 100.98 100.24   1.7367  1.872524 3.271e-01 3.271e-01 1
## 42 42 100.93 100.92   0.2812  1.515974 9.532e-01 3.118e-01 2
## 43 43 100.65 100.93   0.2122  0.669203 1.209e+01 1.000e+00 0
## 44 44 100.96 100.83   0.1874  0.297470 3.688e+01 3.688e+01 1
## 45 45 101.02 100.87   0.1728  0.372924 2.941e+01 2.941e+01 1
## 46 46 101.21 100.90   0.1632  0.750340 9.478e+00 9.478e+00 1
## 47 47 100.95 100.97   0.1575 -0.042189 1.022e+02 1.022e+02 1
## 48 48 100.98 100.96   0.1513  0.034045 8.128e+01 8.128e+01 1
## 49 49 101.16 100.97   0.1460  0.517697 1.905e+01 1.905e+01 1
## 50 50 100.98 101.00   0.1423 -0.051842 1.052e+02 1.052e+02 1
```

### Outliers

```
>>> np.random.seed(66)
>>> rdlm = RandomDLM(n=50, V=0.1, W=0.01)
>>> df_simulated = rdlm.level(
>>>     start_level=100,
>>>     dict_shift={"t": [10, 11, 20, 21, 30, 31, 40, 41],
>>>                 "level_mean_shift": [2, -2, 3, -3, 3.4, -3.4, 3, -3],
>>>                 "level_var_shift": [1, 1, 1, 1, 1, 1, 1, 1]})
```

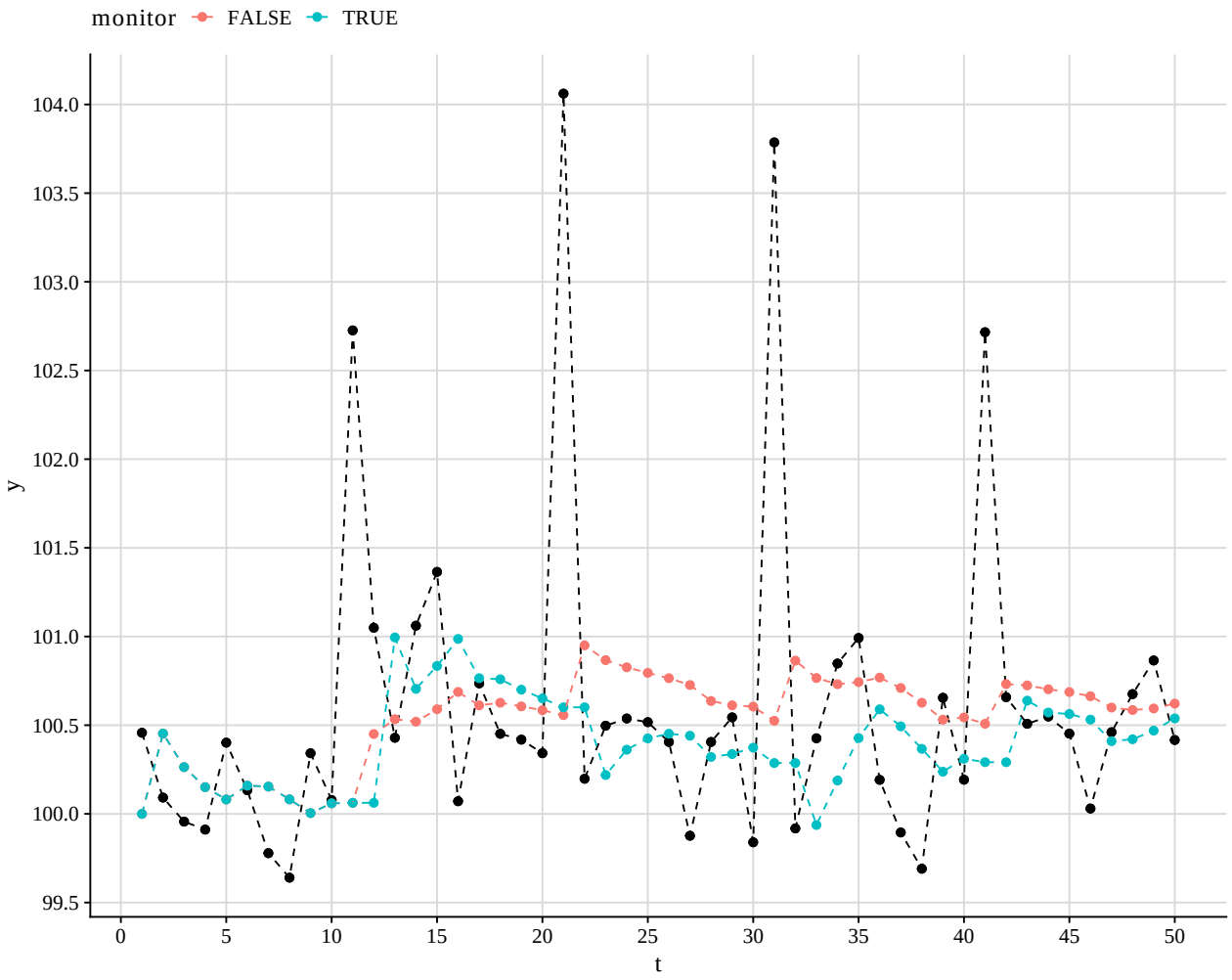



```
>>> a = np.array([100])
>>> R = np.eye(1)
>>> R[[0]] = 100
>>> mod = dlm(a, R, ntrend=1, deltrend=0.9)
>>>
>>> # Fit without monitoring
>>> fit_without_monitor = Smoothing(mod=mod).fit(y=df_simulated["y"])
>>> df_res = fit_without_monitor.get("filter").get("predictive")
>>> df_res["monitor"] = False
>>>
>>> # Fit with monitoring
>>> monitor = AutomaticMonitoring(mod=mod, bilateral=False)
>>> fit_monitor = monitor.fit(y=df_simulated["y"], h=4, tau=0.135, change_var=[100])
```

```
## Potential outlier detected at time 11 with H=0.0, L=0.0 and l=1
## Potential outlier detected at time 21 with H=0.0, L=0.0 and l=1
## Potential outlier detected at time 31 with H=0.0, L=0.0 and l=1
## Potential outlier detected at time 41 with H=0.0, L=0.0 and l=1
```

```
>>> df_tmp = fit_monitor.get("filter").get("predictive")
>>> df_tmp["monitor"] = True
>>>
>>> # Append
```

```
>>> cols_ord = ["t", "monitor", "y", "f", "q", "ci_lower", "ci_upper"]
>>> df_res = pd.concat([df_res[cols_ord], df_tmp[cols_ord]]).reset_index(drop=True)
```

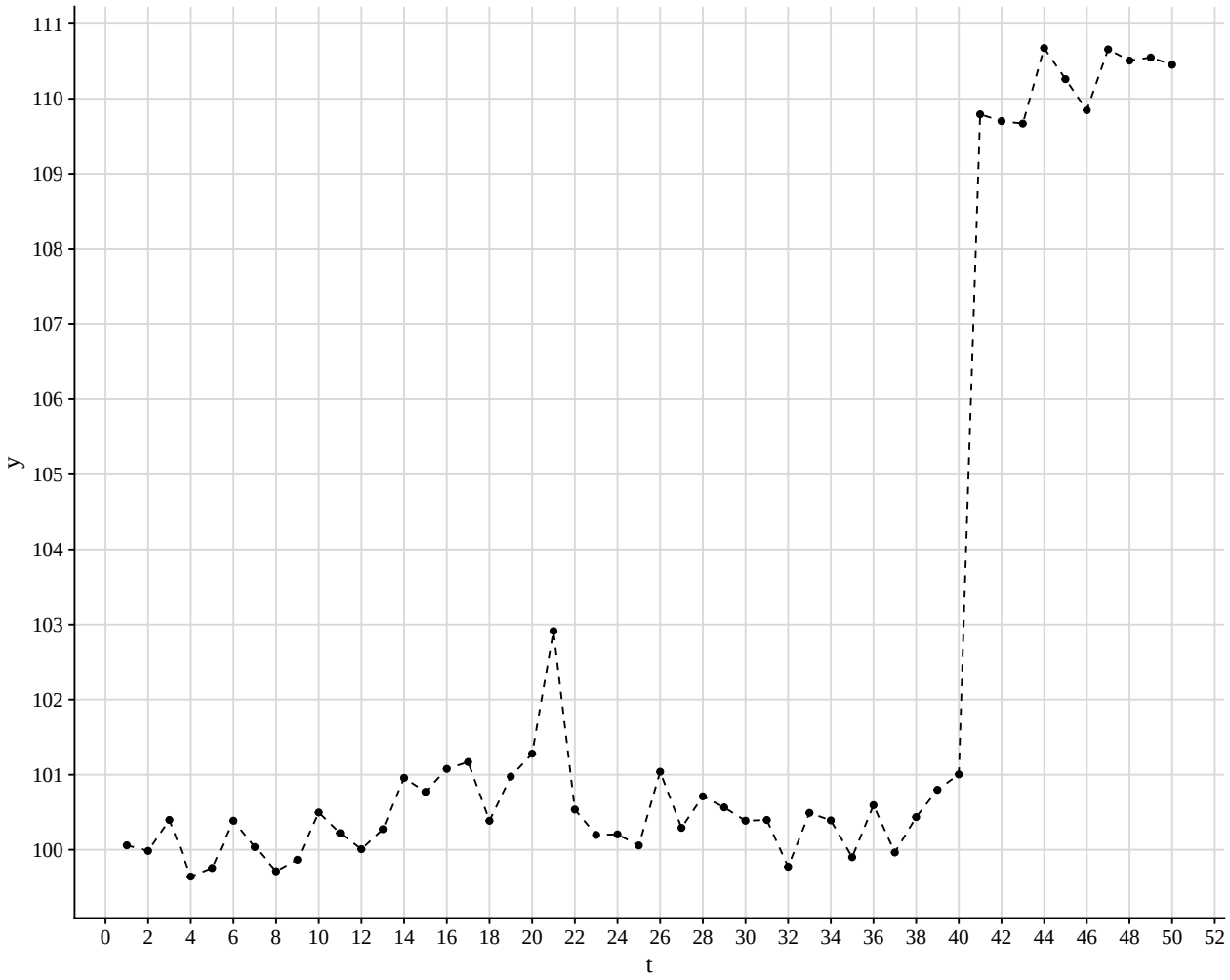



```
##     t      y      f        q        e         H         L l
## 1   1 100.46 100.00 101.0000  0.04550 1.000e+00 1.000e+00 1
## 2   2 100.09 100.45   1.0522 -0.35192 1.000e+00 1.000e+00 1
## 3   3  99.96 100.26   0.5612 -0.41035 1.000e+00 1.000e+00 1
## 4   4  99.91 100.15   0.3958 -0.38023 1.000e+00 1.000e+00 1
## 5   5 100.40 100.08   0.3080  0.57742 1.000e+00 1.000e+00 1
## 6   6 100.13 100.16   0.2631 -0.05013 1.000e+00 1.000e+00 1
## 7   7  99.78 100.15   0.2196 -0.80119 1.000e+00 1.000e+00 1
## 8   8  99.64 100.08   0.2057 -0.97397 1.000e+00 1.000e+00 1
## 9   9 100.34 100.00   0.2015  0.75197 1.000e+00 1.000e+00 1
## 10 10 100.08 100.06   0.1905  0.04164 1.000e+00 1.000e+00 1
## 11 11 102.73 100.06   0.1716  6.42939 2.020e-08 1.000e+00 0
## 12 12 101.05 100.06   2.6468  0.60672 2.633e+02 2.633e+02 1
## 13 13 100.43 100.99   0.2847 -1.06005 2.069e+05 2.069e+05 1
## 14 14 101.06 100.70   0.2200  0.75899 1.432e+02 1.432e+02 1
## 15 15 101.36 100.83   0.1908  1.21403 2.319e+01 2.319e+01 1
## 16 16 100.07 100.99   0.1851 -2.12732 1.479e+07 1.479e+07 1
## 17 17 100.74 100.77   0.2172 -0.06276 3.832e+03 3.832e+03 1
```

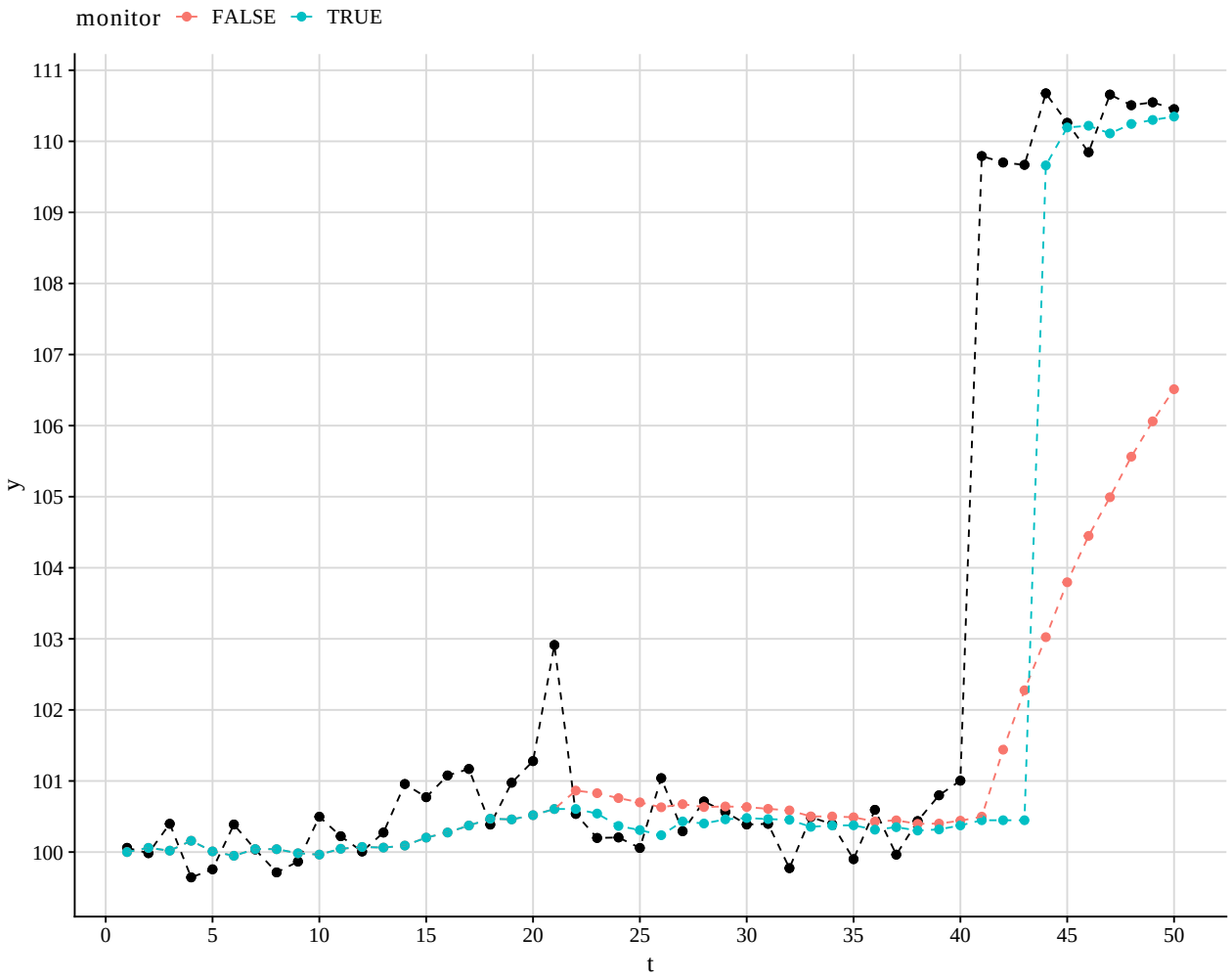
```
## 18 18 100.45 100.76   0.1991 -0.68855 4.683e+04 4.683e+04 1
## 19 19 100.42 100.70   0.1896 -0.64679 3.962e+04 3.962e+04 1
## 20 20 100.34 100.65   0.1811 -0.72604 5.440e+04 5.440e+04 1
## 21 21 104.06 100.60   0.1748  8.27597 1.252e-11 1.000e+00 0
## 22 22 100.20 100.60   2.8231 -0.23975 7.778e+03 7.778e+03 1
## 23 23 100.50 100.22   0.2904  0.51567 3.789e+02 3.789e+02 1
## 24 24 100.54 100.36   0.2146  0.38011 6.517e+02 6.517e+02 1
## 25 25 100.52 100.43   0.1847  0.21290 1.272e+03 1.272e+03 1
## 26 26 100.41 100.45   0.1668 -0.11343 4.692e+03 4.692e+03 1
## 27 27  99.88 100.44   0.1541 -1.43700 9.347e+05 9.347e+05 1
## 28 28 100.41 100.32   0.1561  0.21363 1.268e+03 1.268e+03 1
## 29 29 100.54 100.34   0.1477  0.53839 3.460e+02 3.460e+02 1
## 30 30  99.84 100.37   0.1419 -1.41664 8.616e+05 8.616e+05 1
## 31 31 103.79 100.29   0.1452  9.18538 3.294e-13 1.000e+00 0
## 32 32  99.92 100.29   2.3444 -0.24081 7.811e+03 7.811e+03 1
## 33 33 100.43  99.94   0.2445  0.98890 5.708e+01 5.708e+01 1
## 34 34 100.85 100.19   0.1868  1.52652 6.645e+00 6.645e+00 1
## 35 35 100.99 100.43   0.1739  1.35290 1.331e+01 1.331e+01 1
## 36 36 100.19 100.59   0.1676 -0.97163 1.453e+05 1.453e+05 1
## 37 37  99.90 100.49   0.1610 -1.49086 1.159e+06 1.159e+06 1
## 38 38  99.69 100.37   0.1622 -1.67843 2.455e+06 2.455e+06 1
## 39 39 100.65 100.24   0.1671  1.01996 5.041e+01 5.041e+01 1
## 40 40 100.19 100.31   0.1649 -0.29059 9.532e+03 9.532e+03 1
## 41 41 102.72 100.29   0.1591  6.07876 8.212e-08 1.000e+00 0
## 42 42 100.66 100.29   2.5693  0.22844 1.195e+03 1.195e+03 1
## 43 43 100.51 100.64   0.2699 -0.25147 8.151e+03 8.151e+03 1
## 44 44 100.55 100.57   0.2016 -0.05057 3.649e+03 3.649e+03 1
## 45 45 100.45 100.56   0.1758 -0.26400 8.570e+03 8.570e+03 1
## 46 46 100.03 100.53   0.1616 -1.24712 4.373e+05 4.373e+05 1
## 47 47 100.46 100.41   0.1574  0.12725 1.792e+03 1.792e+03 1
## 48 48 100.68 100.42   0.1499  0.65818 2.143e+02 2.143e+02 1
## 49 49 100.86 100.47   0.1451  1.03864 4.678e+01 4.678e+01 1
## 50 50 100.42 100.54   0.1433 -0.32205 1.081e+04 1.081e+04 1
```

### Outlier and Level Change

```
>>> np.random.seed(66)
>>> rdlm = RandomDLM(n=50, V=0.1, W=0.01)
>>> df_simulated = rdlm.level(
>>>     start_level=100,
>>>     dict_shift={"t": [20, 21, 40],
>>>                 "level_mean_shift": [3, -3, 10],
>>>                 "level_var_shift": [1, 1, 1]})
```




```
## Potential outlier detected at time 21 with H=0.0, L=0.0 and l=1
## Potential outlier detected at time 41 with H=0.0, L=0.0 and l=1
## Potential outlier detected at time 42 with H=0.0, L=0.0 and l=1
```

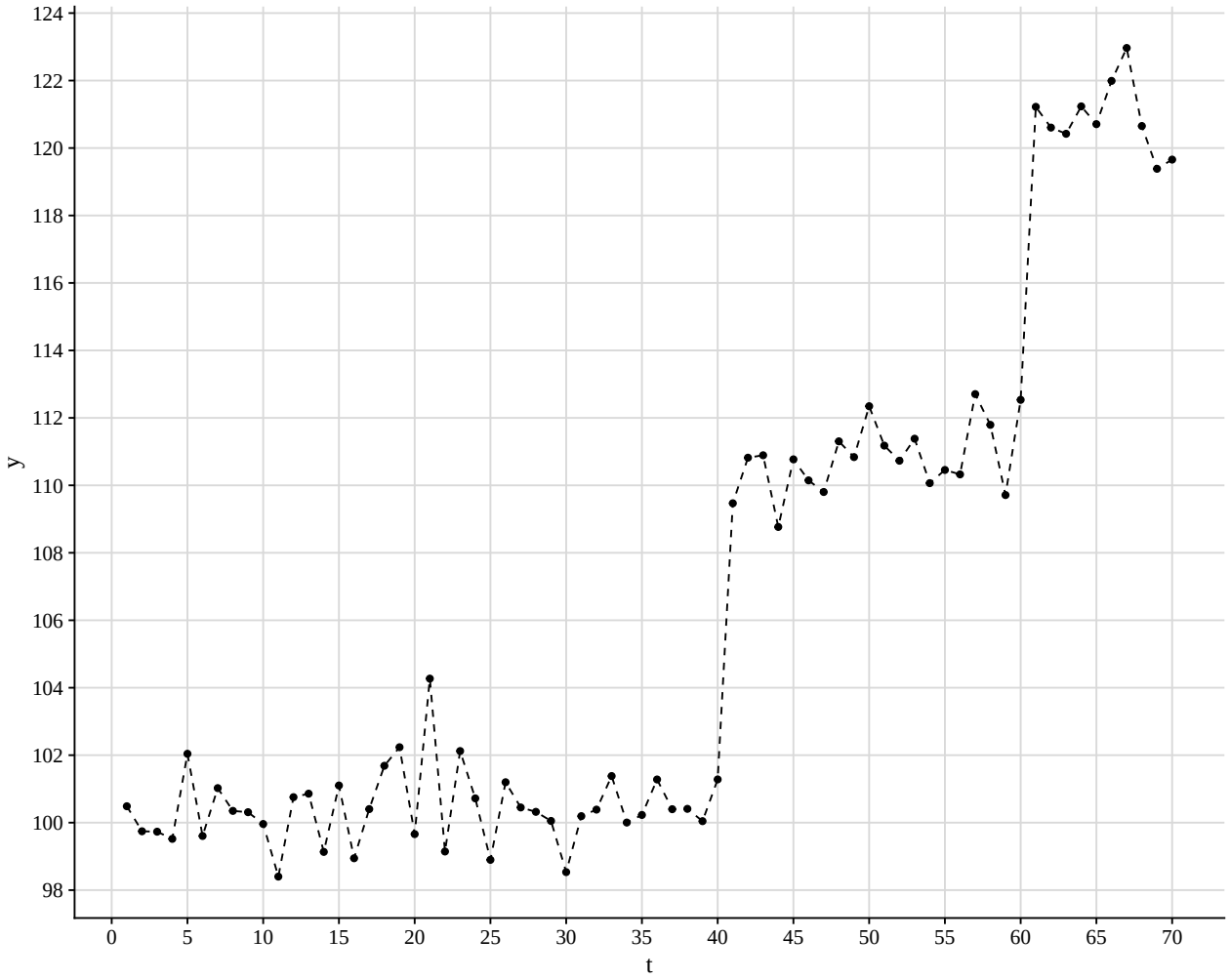



```
##      t      y      f        q         e         H         L l
## 1    1 100.06 100.00 101.0000  0.005906 1.000e+00 1.000e+00 1
## 2    2  99.98 100.06   1.0501 -0.072702 1.000e+00 1.000e+00 1
## 3    3 100.40 100.02   0.5288  0.520070 1.000e+00 1.000e+00 1
## 4    4  99.64 100.16   0.3850 -0.830693 1.000e+00 1.000e+00 1
## 5    5  99.76 100.01   0.3390 -0.433305 1.000e+00 1.000e+00 1
## 6    6 100.39  99.95   0.2817  0.828823 1.000e+00 1.000e+00 1
## 7    7 100.04 100.04   0.2619 -0.010396 1.000e+00 1.000e+00 1
## 8    8  99.71 100.04   0.2247 -0.689692 1.000e+00 1.000e+00 1
## 9    9  99.87  99.98   0.2085 -0.256097 1.000e+00 1.000e+00 1
## 10 10 100.50  99.96   0.1868  1.232935 1.000e+00 1.000e+00 1
## 11 11 100.22 100.05   0.1939  0.402009 5.970e+02 5.970e+02 1
## 12 12 100.01 100.07   0.1790 -0.151071 5.455e+03 5.455e+03 1
## 13 13 100.27 100.06   0.1645  0.521995 3.695e+02 3.695e+02 1
## 14 14 100.96 100.09   0.1552  2.198895 4.513e-01 4.513e-01 1
## 15 15 100.77 100.20   0.1940  1.290263 1.710e+01 7.717e+00 2
## 16 16 101.08 100.27   0.2014  1.788822 2.327e+00 2.327e+00 1
## 17 17 101.17 100.37   0.2267  1.670946 3.729e+00 3.729e+00 1
## 18 18 100.39 100.47   0.2486 -0.164802 5.763e+03 5.763e+03 1
## 19 19 100.98 100.46   0.2354  1.064636 4.216e+01 4.216e+01 1
## 20 20 101.28 100.52   0.2365  1.564744 5.703e+00 5.703e+00 1
## 21 21 102.91 100.61   0.2523  4.593624 3.122e-05 1.000e+00 0
```

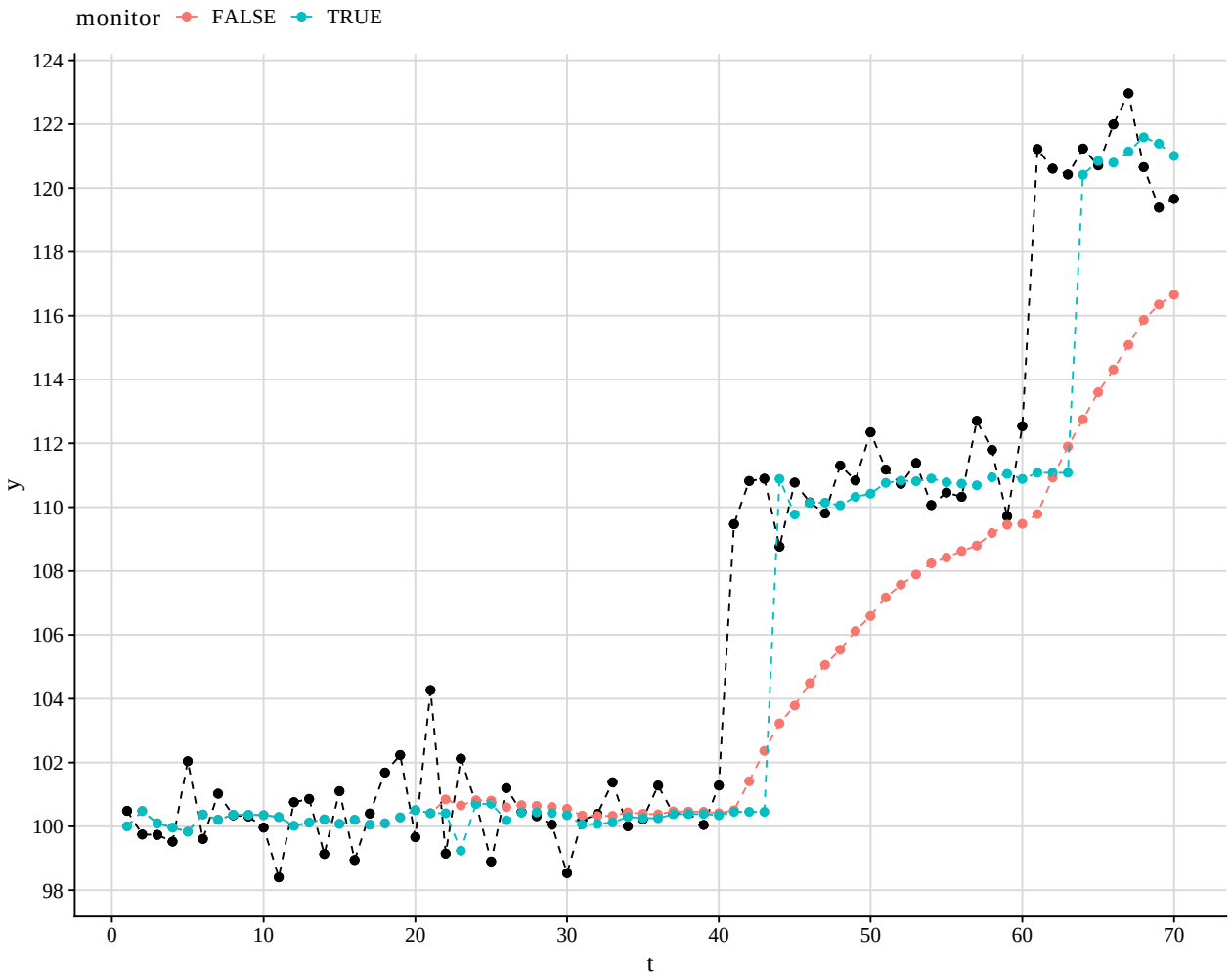
```
## 22 22 100.54 100.61   3.0569 -0.039535 3.492e+03 3.492e+03 1
## 23 23 100.20 100.54   0.4340 -0.521028 2.396e+04 2.396e+04 1
## 24 24 100.21 100.37   0.3238 -0.284362 9.297e+03 9.297e+03 1
## 25 25 100.06 100.31   0.2789 -0.476265 2.003e+04 2.003e+04 1
## 26 26 101.04 100.24   0.2543  1.591679 5.121e+00 5.121e+00 1
## 27 27 100.29 100.43   0.2591 -0.270771 8.805e+03 8.805e+03 1
## 28 28 100.71 100.40   0.2437  0.628259 2.415e+02 2.415e+02 1
## 29 29 100.56 100.46   0.2339  0.215907 1.257e+03 1.257e+03 1
## 30 30 100.39 100.48   0.2229 -0.192873 6.448e+03 6.448e+03 1
## 31 31 100.40 100.46   0.2133 -0.144727 5.318e+03 5.318e+03 1
## 32 32  99.77 100.45   0.2047 -1.503593 1.220e+06 1.220e+06 1
## 33 33 100.49 100.36   0.2113  0.298013 9.050e+02 9.050e+02 1
## 34 34 100.39 100.37   0.2042  0.039400 2.546e+03 2.546e+03 1
## 35 35  99.90 100.38   0.1972 -1.072491 2.175e+05 2.175e+05 1
## 36 36 100.59 100.31   0.1972  0.629152 2.407e+02 2.407e+02 1
## 37 37  99.96 100.35   0.1932 -0.879131 1.004e+05 1.004e+05 1
## 38 38 100.44 100.30   0.1914  0.304634 8.814e+02 8.814e+02 1
## 39 39 100.80 100.32   0.1864  1.111950 3.489e+01 3.489e+01 1
## 40 40 101.00 100.37   0.1871  1.455834 8.817e+00 8.817e+00 1
## 41 41 109.79 100.45   0.1919 21.329571 2.637e-34 1.000e+00 0
## 42 42 109.70 100.45   2.3525  6.033368 9.847e-08 1.000e+00 0
## 43 43 109.67 100.45 218.4051  0.623938 2.457e+02 2.457e+02 1
## 44 44 110.68 109.66   0.3536  1.705067 3.253e+00 3.253e+00 1
## 45 45 110.26 110.19   0.2776  0.124457 1.812e+03 1.812e+03 1
## 46 46 109.85 110.22   0.2413 -0.758511 6.195e+04 6.195e+04 1
## 47 47 110.66 110.11   0.2243  1.154759 2.940e+01 2.940e+01 1
## 48 48 110.51 110.24   0.2171  0.565532 3.104e+02 3.104e+02 1
## 49 49 110.55 110.30   0.2082  0.544045 3.383e+02 3.383e+02 1
## 50 50 110.45 110.35   0.2010  0.232877 1.174e+03 1.174e+03 1
```

### Outlier and Two Level Change

```
>>> np.random.seed(66)
>>> rdlm = RandomDLM(n=70, V=1, W=0.01)
>>> df_simulated = rdlm.level(
>>>     start_level=100,
>>>     dict_shift={"t": [20, 21, 40, 60],
>>>                 "level_mean_shift": [5, -5, 10, 10],
>>>                 "level_var_shift": [1, 1, 1, 1]})
```




```
## Potential outlier detected at time 21 with H=0.001, L=0.001 and l=1
## Potential outlier detected at time 41 with H=0.0, L=0.0 and l=1
## Potential outlier detected at time 42 with H=0.0154, L=0.0154 and l=1
## Potential outlier detected at time 61 with H=0.0, L=0.0 and l=1
## Potential outlier detected at time 62 with H=0.0551, L=0.0551 and l=1
```



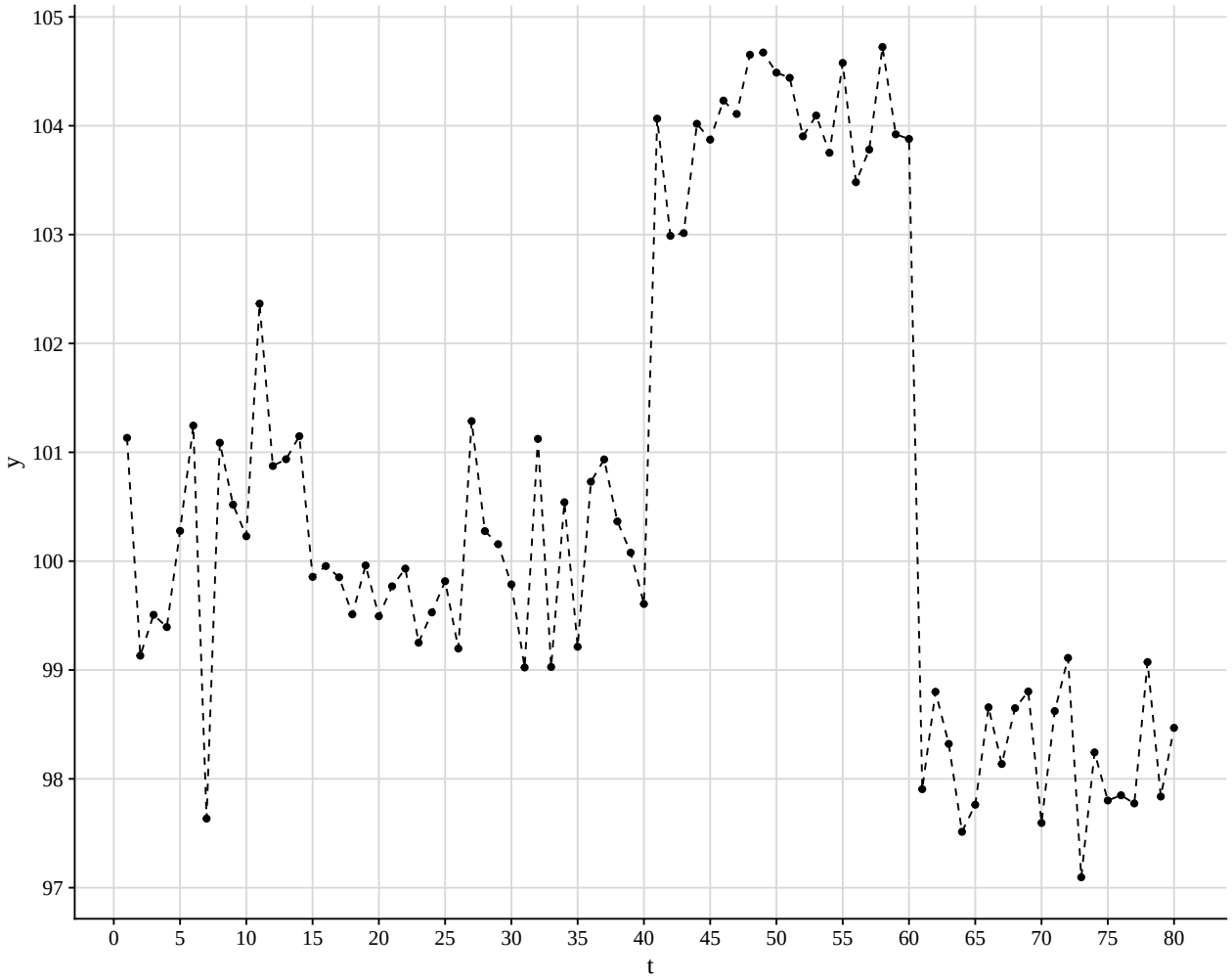

```
##      t      y      f         q         e        H        L l
## 1    1 100.49 100.00  101.0000  0.048367 1.000e+00 1.000e+00 1
## 2    2  99.74 100.48    1.0525 -0.718276 1.000e+00 1.000e+00 1
## 3    3  99.73 100.10    0.6649 -0.445731 1.000e+00 1.000e+00 1
## 4    4  99.52  99.96    0.4735 -0.642503 1.000e+00 1.000e+00 1
## 5    5 102.04  99.83    0.3923  3.525704 1.000e+00 1.000e+00 1
## 6    6  99.61 100.37    1.0952 -0.732493 1.000e+00 1.000e+00 1
## 7    7 101.02 100.21    0.9953  0.817334 1.000e+00 1.000e+00 1
## 8    8 100.35 100.36    0.9354 -0.016446 1.000e+00 1.000e+00 1
## 9    9 100.31 100.36    0.8193 -0.057104 1.000e+00 1.000e+00 1
## 10 10  99.96 100.35    0.7292 -0.459832 1.000e+00 1.000e+00 1
## 11 11  98.40 100.29    0.6707 -2.310043 3.071e+07 3.071e+07 1
## 12 12 100.76 100.02    0.9064  0.775188 1.342e+02 1.342e+02 1
## 13 13 100.86 100.12    0.8732  0.790556 1.262e+02 1.262e+02 1
## 14 14  99.13 100.22    0.8455 -1.181524 3.364e+05 3.364e+05 1
## 15 15 101.10 100.08    0.8641  1.100947 3.646e+01 3.646e+01 1
## 16 16  98.94 100.21    0.8724 -1.351576 6.642e+05 6.642e+05 1
## 17 17 100.40 100.05    0.9120  0.366429 6.883e+02 6.883e+02 1
## 18 18 101.69 100.09    0.8658  1.710841 3.179e+00 3.179e+00 1
## 19 19 102.24 100.28    0.9514  2.003486 9.862e-01 9.862e-01 1
## 20 20  99.66 100.51    1.0926 -0.810314 7.621e+04 7.516e+04 2
## 21 21 104.27 100.41    1.0729  3.725633 1.005e-03 1.000e+00 0
```

```
## 22 22  99.14 100.41   12.9975 -0.351295 1.215e+04 1.215e+04 1
## 23 23 102.12  99.24    1.8561  2.118274 6.231e-01 6.231e-01 1
## 24 24 100.72 100.70    1.6467  0.014348 2.815e+03 1.754e+03 2
## 25 25  98.90 100.71    1.4135 -1.522634 1.317e+06 1.317e+06 1
## 26 26 101.20 100.19    1.4001  0.851742 9.879e+01 9.879e+01 1
## 27 27 100.45 100.43    1.3329  0.014637 2.811e+03 2.811e+03 1
## 28 28 100.32 100.44    1.2500 -0.103953 4.518e+03 4.518e+03 1
## 29 29 100.05 100.41    1.1829 -0.334215 1.135e+04 1.135e+04 1
## 30 30  98.53 100.35    1.1301 -1.709695 2.782e+06 2.782e+06 1
## 31 31 100.19 100.06    1.1890  0.125275 1.806e+03 1.806e+03 1
## 32 32 100.39 100.08    1.1409  0.289497 9.364e+02 9.364e+02 1
## 33 33 101.38 100.12    1.1002  1.200294 2.450e+01 2.450e+01 1
## 34 34 100.00 100.30    1.1082 -0.278882 9.095e+03 9.095e+03 1
## 35 35 100.23 100.26    1.0728 -0.028141 3.336e+03 3.336e+03 1
## 36 36 101.28 100.25    1.0377  1.007153 5.306e+01 5.306e+01 1
## 37 37 100.40 100.38    1.0344  0.015935 2.797e+03 2.797e+03 1
## 38 38 100.41 100.38    1.0034  0.024566 2.702e+03 2.702e+03 1
## 39 39 100.04 100.39    0.9744 -0.347833 1.198e+04 1.198e+04 1
## 40 40 101.28 100.35    0.9503  0.957780 6.464e+01 6.464e+01 1
## 41 41 109.47 100.46    0.9465  9.264824 2.397e-13 1.000e+00 0
## 42 42 110.82 100.46   11.6002  3.043024 1.542e-02 1.000e+00 0
## 43 43 110.89 100.46 1076.9772  0.318020 8.354e+02 8.354e+02 1
## 44 44 108.77 110.88    1.7314 -1.608805 1.858e+06 1.858e+06 1
## 45 45 110.77 109.77    1.3492  0.861384 9.506e+01 9.506e+01 1
## 46 46 110.15 110.14    1.1933  0.007497 2.893e+03 2.893e+03 1
## 47 47 109.80 110.14    1.0943 -0.323459 1.087e+04 1.087e+04 1
## 48 48 111.30 110.06    1.0306  1.227059 2.202e+01 2.202e+01 1
## 49 49 110.84 110.32    1.0139  0.509124 3.890e+02 3.890e+02 1
## 50 50 112.35 110.42    0.9785  1.944294 1.250e+00 1.250e+00 1
## 51 51 111.18 110.76    1.0199  0.413232 5.708e+02 5.708e+02 1
## 52 52 110.73 110.83    0.9911 -0.099277 4.434e+03 4.434e+03 1
## 53 53 111.38 110.81    0.9626  0.579885 2.931e+02 2.931e+02 1
## 54 54 110.07 110.90    0.9431 -0.856058 9.151e+04 9.151e+04 1
## 55 55 110.46 110.78    0.9325 -0.335897 1.143e+04 1.143e+04 1
## 56 56 110.32 110.74    0.9122 -0.431681 1.676e+04 1.676e+04 1
## 57 57 112.71 110.68    0.8946  2.137120 5.778e-01 5.778e-01 1
## 58 58 111.79 110.94    0.9492  0.877814 8.901e+01 5.143e+01 2
## 59 59 109.71 111.04    0.9424 -1.371258 7.186e+05 7.186e+05 1
## 60 60 112.53 110.88    0.9544  1.690431 3.450e+00 3.450e+00 1
## 61 61 121.22 111.08    0.9827 10.232314 5.000e-15 1.000e+00 0
## 62 62 120.61 111.08   12.2306  2.724598 5.511e-02 1.000e+00 0
## 63 63 120.42 111.08 1137.0190  0.277123 9.839e+02 9.839e+02 1
## 64 64 121.23 120.42    1.8053  0.609810 2.600e+02 2.600e+02 1
## 65 65 120.71 120.85    1.3414 -0.119504 4.808e+03 4.808e+03 1
## 66 66 121.99 120.80    1.1742  1.103500 3.609e+01 3.609e+01 1
## 67 67 122.97 121.14    1.1057  1.733631 2.902e+00 2.902e+00 1
## 68 68 120.65 121.59    1.0963 -0.893458 1.063e+05 1.063e+05 1
## 69 69 119.38 121.39    1.0635 -1.942896 7.072e+06 7.072e+06 1
## 70 70 119.66 121.00    1.0872 -1.291934 5.232e+05 5.232e+05 1
```

### Bilateral Level Change

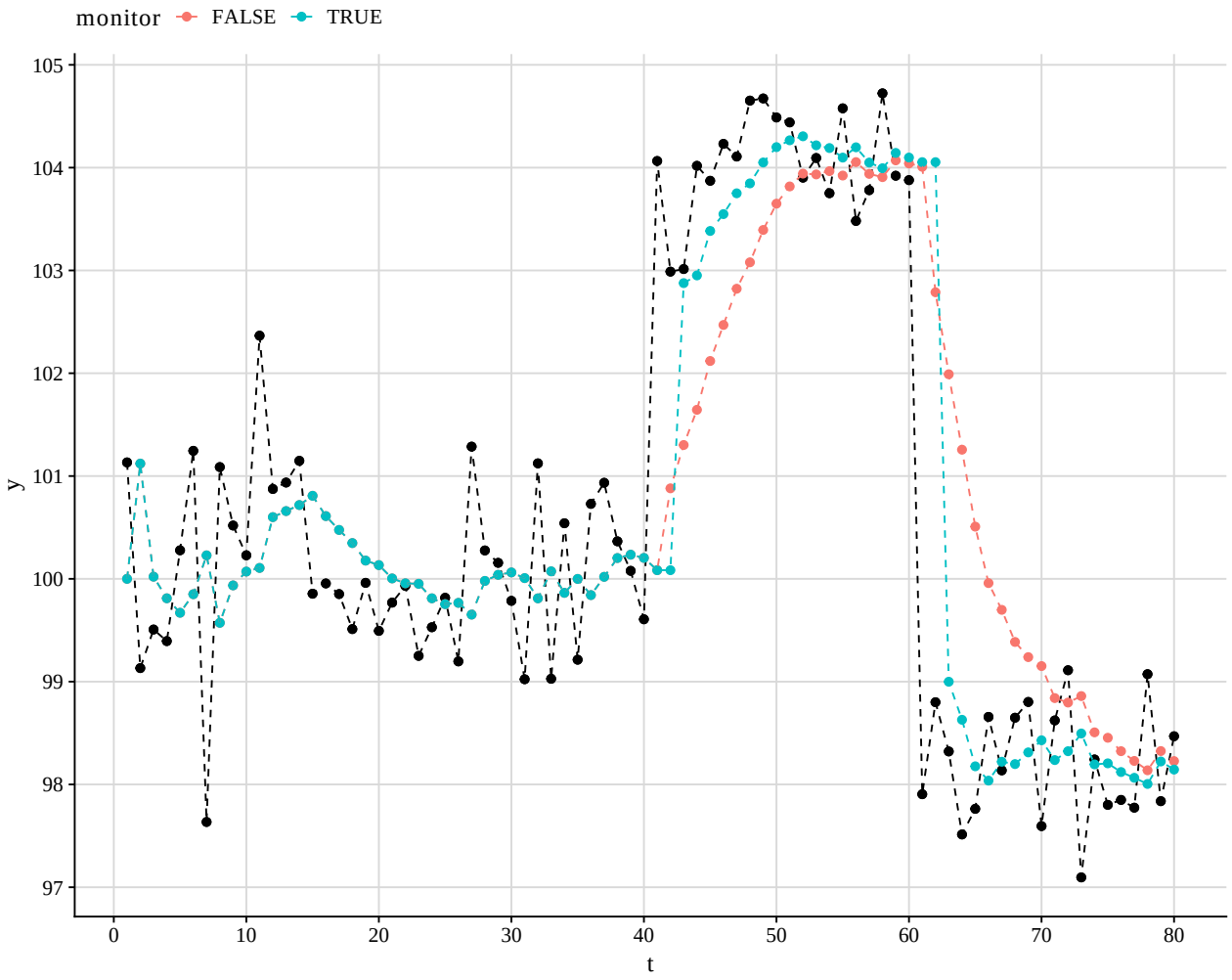
```
>>> np.random.seed(66)
>>> y1 = np.random.normal(loc=100, scale=0.8, size=40)
```

```
>>> y2 = np.random.normal(loc=104, scale=0.5, size=20)
>>> y3 = np.random.normal(loc=98, scale=0.5, size=20)
>>> y = np.concatenate([y1, y2, y3])
>>> t = np.arange(0, len(y)) + 1
>>> df_simulated = pd.DataFrame({"t": t, "y": y})
```




```
## Upper potential outlier detected at time 41 with H=0.0001, L=0.0001 and l=1
## Lower potential outlier detected at time 61 with H=0.0, L=0.0 and l=1
```

```
>>> ggplot(py$df_res, aes(x = t, y = y)) + geom_point(size = 2) + geom_line(linetype = "dashed") +
>>>     geom_point(aes(y = f, col = monitor), size = 2) + geom_line(aes(y = f,
>>>     col = monitor), linetype = "dashed") + scale_x_continuous(breaks = scales::pretty_breaks(10)) +
>>>     scale_y_continuous(breaks = scales::pretty_breaks(10))
```




```
>>> py$df_tmp %>%
>>>     select(t, y, f, q, e, H_lower, L_lower, l_lower, H_upper, L_upper,
>>>         l_upper)
```

```
##     t      y      f        q        e   H_lower   L_lower l_lower
## 1   1 101.13 100.00 101.0000  0.11269 1.000e+00 1.000e+00       1
## 2   2  99.13 101.12   1.1330 -1.86908 1.000e+00 1.000e+00       1
## 3   3  99.51 100.02   1.5682 -0.41039 1.000e+00 1.000e+00       1
## 4   4  99.39  99.81   1.1097 -0.39523 1.000e+00 1.000e+00       1
## 5   5 100.28  99.67   0.8686  0.65198 1.000e+00 1.000e+00       1
## 6   6 101.25  99.85   0.7570  1.60292 1.000e+00 1.000e+00       1
## 7   7  97.63 100.23   0.9045 -2.72678 1.000e+00 1.000e+00       1
## 8   8 101.09  99.57   1.6048  1.19583 1.000e+00 1.000e+00       1
## 9   9 100.52  99.94   1.6612  0.45189 1.000e+00 1.000e+00       1
## 10 10 100.23 100.07   1.5154  0.12842 1.000e+00 1.000e+00       1
## 11 11 102.37 100.11   1.3706  1.92990 6.713e+06 6.713e+06       1
## 12 12 100.87 100.60   1.6732  0.21167 6.951e+03 6.951e+03       1
## 13 13 100.94 100.66   1.5441  0.22333 7.283e+03 7.283e+03       1
## 14 14 101.15 100.72   1.4349  0.35921 1.254e+04 1.254e+04       1
## 15 15  99.86 100.81   1.3483 -0.82044 1.120e+02 1.120e+02       1
## 16 16  99.96 100.61   1.3183 -0.57070 3.041e+02 3.041e+02       1
## 17 17  99.85 100.48   1.2641 -0.55467 3.242e+02 3.242e+02       1
```

```
## 18 18  99.51 100.35   1.2141 -0.75940 1.429e+02 1.429e+02       1
## 19 19  99.96 100.18   1.1859 -0.19892 1.345e+03 1.345e+03       1
## 20 20  99.49 100.13   1.1281 -0.60229 2.680e+02 2.680e+02       1
## 21 21  99.77 100.00   1.0933 -0.22518 1.211e+03 1.211e+03       1
## 22 22  99.93  99.96   1.0456 -0.02454 2.702e+03 2.702e+03       1
## 23 23  99.25  99.95   0.9998 -0.70097 1.806e+02 1.806e+02       1
## 24 24  99.53  99.81   0.9783 -0.28471 9.545e+02 9.545e+02       1
## 25 25  99.82  99.75   0.9421  0.06360 3.845e+03 3.845e+03       1
## 26 26  99.20  99.77   0.9059 -0.59823 2.723e+02 2.723e+02       1
## 27 27 101.29  99.65   0.8842  1.73763 3.111e+06 3.111e+06       1
## 28 28 100.28  99.98   0.9478  0.30431 1.007e+04 1.007e+04       1
## 29 29 100.16 100.04   0.9181  0.12169 4.850e+03 4.850e+03       1
## 30 30  99.79 100.06   0.8879 -0.29325 9.224e+02 9.224e+02       1
## 31 31  99.02 100.01   0.8616 -1.05991 4.296e+01 4.296e+01       1
## 32 32 101.12  99.81   0.8649  1.41205 8.459e+05 8.459e+05       1
## 33 33  99.03 100.07   0.8909 -1.10743 3.553e+01 3.553e+01       1
## 34 34 100.54  99.86   0.8968  0.71473 5.200e+04 5.200e+04       1
## 35 35  99.21 100.00   0.8843 -0.83581 1.053e+02 1.053e+02       1
## 36 36 100.73  99.84   0.8769  0.94767 1.320e+05 1.320e+05       1
## 37 37 100.93 100.02   0.8744  0.97843 1.493e+05 1.493e+05       1
## 38 38 100.37 100.20   0.8734  0.17387 5.976e+03 5.976e+03       1
## 39 39 100.08 100.24   0.8517 -0.17034 1.508e+03 1.508e+03       1
## 40 40  99.61 100.20   0.8310 -0.65477 2.172e+02 2.172e+02       1
## 41 41 104.06 100.08   0.8194  4.39709 1.297e+11 1.297e+11       1
## 42 42 102.99 100.08  17.0461  0.70342 4.970e+04 4.970e+04       1
## 43 43 103.01 102.88   1.4261  0.11467 4.716e+03 4.716e+03       1
## 44 44 104.02 102.95   1.0645  1.03340 1.860e+05 1.860e+05       1
## 45 45 103.87 103.38   0.9551  0.49947 2.198e+04 2.198e+04       1
## 46 46 104.23 103.55   0.8853  0.72574 5.434e+04 5.434e+04       1
## 47 47 104.11 103.75   0.8450  0.38908 1.413e+04 1.413e+04       1
## 48 48 104.65 103.85   0.8101  0.89455 1.067e+05 1.067e+05       1
## 49 49 104.67 104.05   0.7934  0.69911 4.885e+04 4.885e+04       1
## 50 50 104.49 104.20   0.7758  0.32738 1.104e+04 1.104e+04       1
## 51 51 104.44 104.27   0.7552  0.20053 6.648e+03 6.648e+03       1
## 52 52 103.90 104.31   0.7360 -0.46916 4.564e+02 4.564e+02       1
## 53 53 104.09 104.22   0.7213 -0.14506 1.669e+03 1.669e+03       1
## 54 54 103.75 104.19   0.7052 -0.52314 3.678e+02 3.678e+02       1
## 55 55 104.58 104.10   0.6936  0.57543 2.978e+04 2.978e+04       1
## 56 56 103.48 104.20   0.6835 -0.86654 9.312e+01 9.312e+01       1
## 57 57 103.78 104.05   0.6792 -0.32603 8.091e+02 8.091e+02       1
## 58 58 104.72 103.99   0.6676  0.89203 1.057e+05 1.057e+05       1
## 59 59 103.92 104.14   0.6645 -0.27237 1.003e+03 1.003e+03       1
## 60 60 103.88 104.10   0.6534 -0.27149 1.006e+03 1.006e+03       1
## 61 61  97.91 104.05   0.6429 -7.66839 1.422e-10 1.000e+00       0
## 62 62  98.80 104.05  13.5188 -1.42887 9.821e+00 9.821e+00       1
## 63 63  98.32  99.00   1.1488 -0.63253 2.374e+02 2.374e+02       1
## 64 64  97.51  98.63   0.8691 -1.19660 2.487e+01 2.487e+01       1
## 65 65  97.76  98.18   0.7838 -0.46765 4.592e+02 4.592e+02       1
## 66 66  98.66  98.04   0.7298  0.72567 5.432e+04 5.432e+04       1
## 67 67  98.14  98.22   0.6987 -0.10050 1.994e+03 1.994e+03       1
## 68 68  98.65  98.20   0.6719  0.55019 2.692e+04 2.692e+04       1
## 69 69  98.80  98.31   0.6539  0.60663 3.374e+04 3.374e+04       1
## 70 70  97.59  98.43   0.6401 -1.04345 4.589e+01 4.589e+01       1
## 71 71  98.62  98.24   0.6353  0.48382 2.065e+04 2.065e+04       1
```

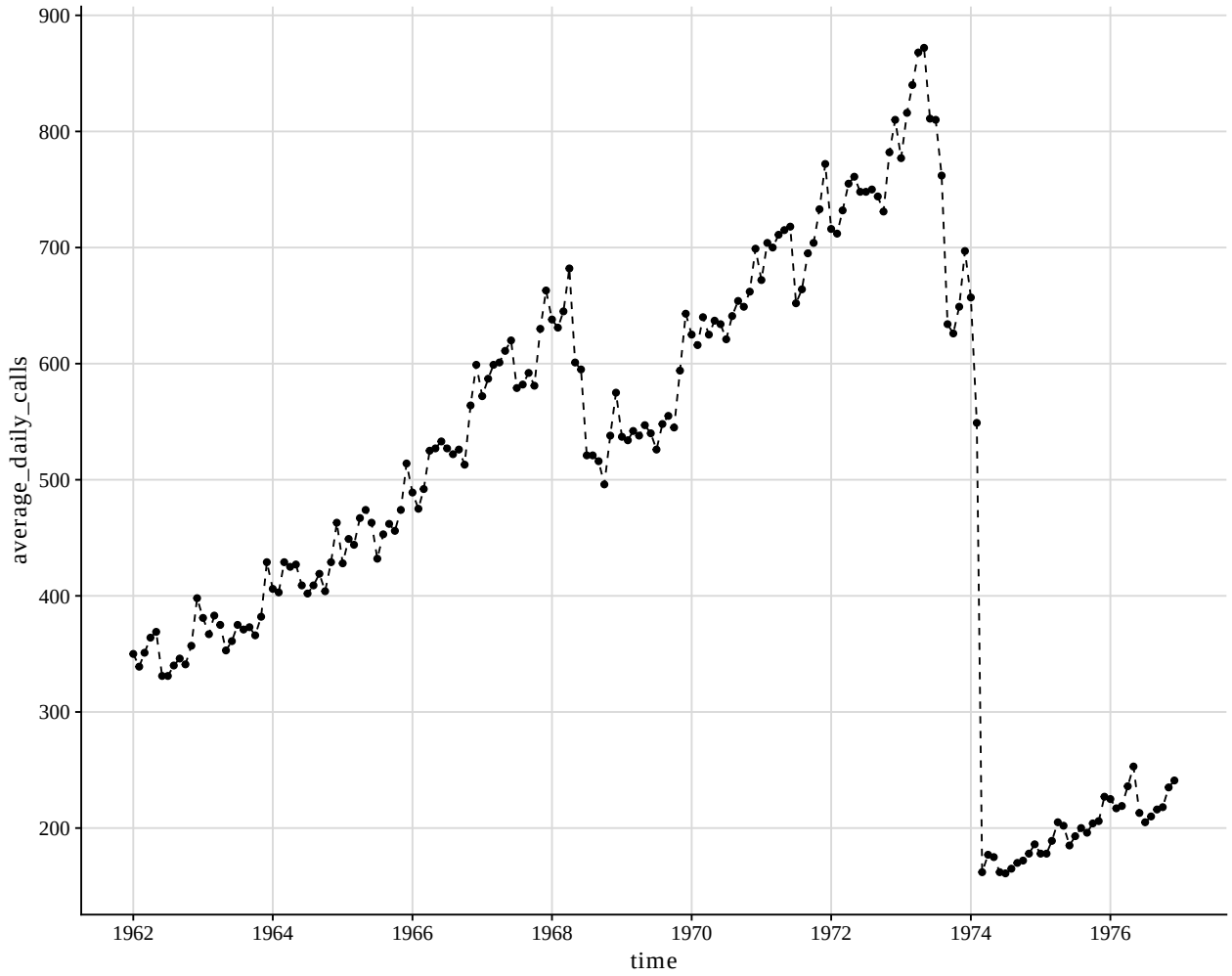
```
## 72 72  99.11  98.32   0.6242  0.99725 1.610e+05 1.610e+05       1
## 73 73  97.10  98.50   0.6209 -1.77631 2.447e+00 2.447e+00       1
## 74 74  98.24  98.20   0.6370  0.06006 3.790e+03 3.790e+03       1
## 75 75  97.80  98.21   0.6264 -0.51024 3.872e+02 3.872e+02       1
## 76 76  97.85  98.12   0.6187 -0.34507 7.497e+02 7.497e+02       1
## 77 77  97.77  98.06   0.6103 -0.37192 6.734e+02 6.734e+02       1
## 78 78  99.07  98.00   0.6025  1.37608 7.326e+05 7.326e+05       1
## 79 79  97.84  98.22   0.6087 -0.49467 4.121e+02 4.121e+02       1
## 80 80  98.47  98.14   0.6023  0.41762 1.584e+04 1.584e+04       1
##      H_upper   L_upper l_upper
## 1  1.000e+00 1.000e+00       1
## 2  1.000e+00 1.000e+00       1
## 3  1.000e+00 1.000e+00       1
## 4  1.000e+00 1.000e+00       1
## 5  1.000e+00 1.000e+00       1
## 6  1.000e+00 1.000e+00       1
## 7  1.000e+00 1.000e+00       1
## 8  1.000e+00 1.000e+00       1
## 9  1.000e+00 1.000e+00       1
## 10 1.000e+00 1.000e+00       1
## 11 1.324e+00 1.324e+00       1
## 12 1.278e+03 1.278e+03       1
## 13 1.220e+03 1.220e+03       1
## 14 7.085e+02 7.085e+02       1
## 15 7.936e+04 7.936e+04       1
## 16 2.923e+04 2.923e+04       1
## 17 2.741e+04 2.741e+04       1
## 18 6.217e+04 6.217e+04       1
## 19 6.606e+03 6.606e+03       1
## 20 3.316e+04 3.316e+04       1
## 21 7.337e+03 7.337e+03       1
## 22 3.288e+03 3.288e+03       1
## 23 4.921e+04 4.921e+04       1
## 24 9.310e+03 9.310e+03       1
## 25 2.311e+03 2.311e+03       1
## 26 3.263e+04 3.263e+04       1
## 27 2.856e+00 2.856e+00       1
## 28 8.825e+02 8.825e+02       1
## 29 1.832e+03 1.832e+03       1
## 30 9.633e+03 9.633e+03       1
## 31 2.068e+05 2.068e+05       1
## 32 1.050e+01 1.050e+01       1
## 33 2.501e+05 2.501e+05       1
## 34 1.709e+02 1.709e+02       1
## 35 8.439e+04 8.439e+04       1
## 36 6.731e+01 6.731e+01       1
## 37 5.952e+01 5.952e+01       1
## 38 1.487e+03 1.487e+03       1
## 39 5.892e+03 5.892e+03       1
## 40 4.091e+04 4.091e+04       1
## 41 6.852e-05 1.000e+00       0
## 42 1.788e+02 1.788e+02       1
## 43 1.884e+03 1.884e+03       1
## 44 4.777e+01 4.777e+01       1
```

```
## 45 4.043e+02 4.043e+02       1
## 46 1.635e+02 1.635e+02       1
## 47 6.287e+02 6.287e+02       1
## 48 8.325e+01 8.325e+01       1
## 49 1.819e+02 1.819e+02       1
## 50 8.047e+02 8.047e+02       1
## 51 1.337e+03 1.337e+03       1
## 52 1.947e+04 1.947e+04       1
## 53 5.325e+03 5.325e+03       1
## 54 2.416e+04 2.416e+04       1
## 55 2.984e+02 2.984e+02       1
## 56 9.543e+04 9.543e+04       1
## 57 1.098e+04 1.098e+04       1
## 58 8.409e+01 8.409e+01       1
## 59 8.861e+03 8.861e+03       1
## 60 8.830e+03 8.830e+03       1
## 61 6.248e+16 6.248e+16       1
## 62 9.048e+05 9.048e+05       1
## 63 3.743e+04 3.743e+04       1
## 64 3.573e+05 3.573e+05       1
## 65 1.935e+04 1.935e+04       1
## 66 1.636e+02 1.636e+02       1
## 67 4.456e+03 4.456e+03       1
## 68 3.300e+02 3.300e+02       1
## 69 2.633e+02 2.633e+02       1
## 70 1.936e+05 1.936e+05       1
## 71 4.304e+02 4.304e+02       1
## 72 5.520e+01 5.520e+01       1
## 73 3.632e+06 3.632e+06       1
## 74 2.344e+03 2.344e+03       1
## 75 2.295e+04 2.295e+04       1
## 76 1.185e+04 1.185e+04       1
## 77 1.320e+04 1.320e+04       1
## 78 1.213e+01 1.213e+01       1
## 79 2.156e+04 2.156e+04       1
## 80 5.609e+02 5.609e+02       1
```

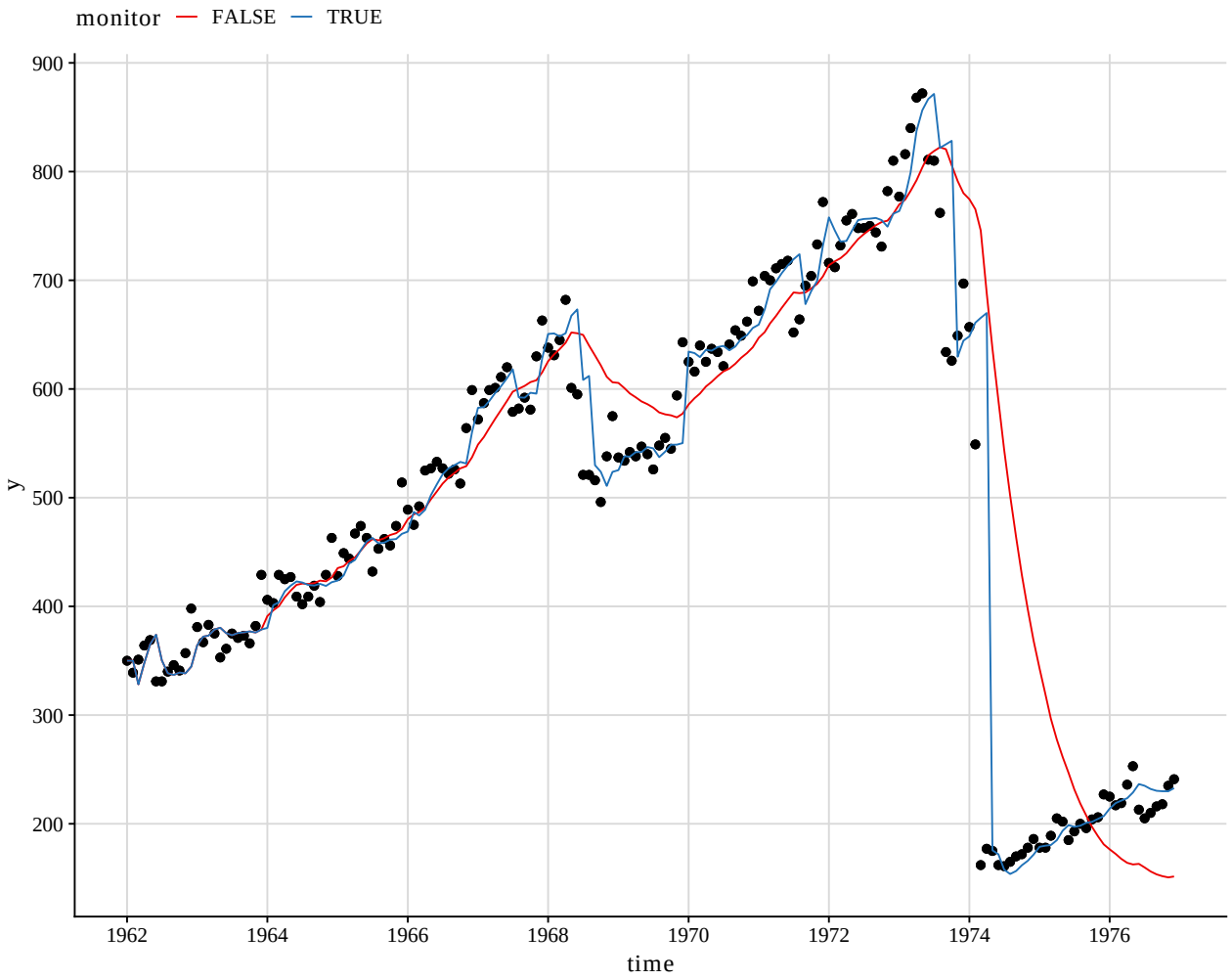
## Real data applications

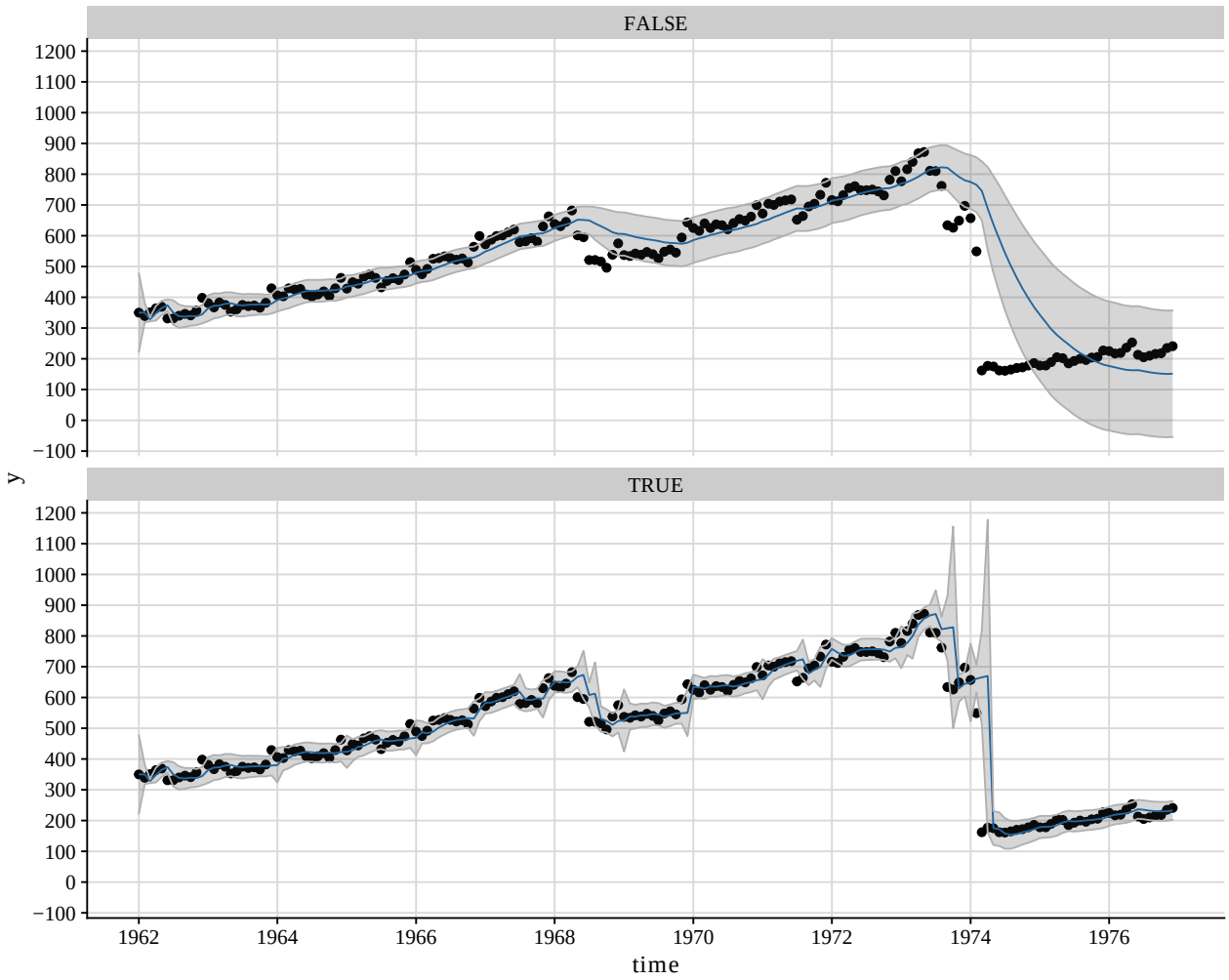
### Telephone Calls

```
## Rows: 180
## Columns: 2
## $ time                <date> 1962-01-01, 1962-02-01, 1962-03-01, 196~
## $ average_daily_calls <dbl> 350, 339, 351, 364, 369, 331, 331, 340, ~
```




```
## Upper potential outlier detected at time 24 with H=0.0062, L=0.0062 and l=1
## Upper potential outlier detected at time 36 with H=0.047, L=0.047 and l=1
## Upper potential outlier detected at time 48 with H=0.0107, L=0.0107 and l=1
## Upper parametric change detected at time 61 with H=371.5075, L=0.8666 and l=3
## Lower parametric change detected at time 69 with H=73.4896, L=21.1134 and l=3
## Upper parametric change detected at time 73 with H=1017.8894, L=5.1031 and l=3
## Lower potential outlier detected at time 77 with H=0.0005, L=0.0005 and l=1
## Lower potential outlier detected at time 79 with H=0.0001, L=0.0001 and l=1
## Upper potential outlier detected at time 84 with H=0.0588, L=0.0588 and l=1
## Upper potential outlier detected at time 95 with H=0.0689, L=0.0689 and l=1
## Upper potential outlier detected at time 108 with H=0.0727, L=0.0727 and l=1
## Lower potential outlier detected at time 115 with H=0.0001, L=0.0001 and l=1
## Lower parametric change detected at time 121 with H=9.8175, L=9.8175 and l=3
## Upper potential outlier detected at time 132 with H=0.0287, L=0.0287 and l=1
## Upper parametric change detected at time 137 with H=1.3781, L=0.0149 and l=3
## Lower potential outlier detected at time 138 with H=0.0095, L=0.0095 and l=1
## Lower potential outlier detected at time 140 with H=0.0283, L=0.0283 and l=1
## Lower potential outlier detected at time 141 with H=0.0016, L=0.0016 and l=1
## Upper potential outlier detected at time 144 with H=0.1144, L=0.1144 and l=1
## Lower potential outlier detected at time 146 with H=0.0, L=0.0 and l=1
## Lower potential outlier detected at time 147 with H=0.0, L=0.0 and l=1
```


monitor
FALSE
TRUE
900
800
700
600
500
400
300
200
y
1962
1964
1966
1968
1970
1972
1974
1976
time




```
##                   time   y     f        q        e   H_lower
## 1  1961-12-31 21:00:00 350 350.0  101.000  0.00000 1.000e+00
## 2  1962-01-31 21:00:00 339 350.0   53.653 -1.50175 1.000e+00
## 3  1962-02-28 21:00:00 351 328.3    4.409 10.80510 1.000e+00
## 4  1962-03-31 21:00:00 364 348.0   74.226  1.85430 1.000e+00
## 5  1962-04-30 21:00:00 369 365.0   83.023  0.43429 1.000e+00
## 6  1962-05-31 21:00:00 331 373.9   60.367 -5.52633 1.000e+00
## 7  1962-06-30 21:00:00 331 350.3  280.258 -1.15428 1.000e+00
## 8  1962-07-31 21:00:00 340 338.5  268.177  0.09294 1.000e+00
## 9  1962-08-31 21:00:00 346 336.9  223.792  0.60535 1.000e+00
## 10 1962-09-30 21:00:00 341 339.1  199.316  0.13333 1.000e+00
## 11 1962-10-31 21:00:00 357 338.4  174.300  1.40781 1.000e+00
## 12 1962-11-30 21:00:00 398 344.5  182.370  3.95840 1.000e+00
## 13 1962-12-31 21:00:00 381 364.2  377.451  0.86221 1.000e+00
## 14 1963-01-31 21:00:00 367 372.1  361.838 -0.27018 1.000e+00
## 15 1963-02-28 21:00:00 383 373.1  332.615  0.54294 1.000e+00
## 16 1963-03-31 21:00:00 375 378.7  312.363 -0.20765 1.000e+00
## 17 1963-04-30 21:00:00 353 380.4  290.231 -1.60653 1.000e+00
## 18 1963-05-31 21:00:00 361 375.3  311.412 -0.81209 1.000e+00
## 19 1963-06-30 21:00:00 375 373.5  302.112  0.08850 1.000e+00
## 20 1963-07-31 21:00:00 371 375.5  283.996 -0.26852 1.000e+00
## 21 1963-08-31 21:00:00 373 376.1  268.783 -0.18907 1.399e+03
```

```
## 22 1963-09-30 21:00:00 366 377.0  254.718 -0.68730 1.907e+02
## 23 1963-10-31 22:00:00 382 376.0  246.863  0.38242 1.376e+04
## 24 1963-11-30 22:00:00 429 378.7  236.325  3.27150 1.437e+09
## 25 1963-12-31 22:00:00 406 380.2  737.882  0.95073 1.336e+05
## 26 1964-01-31 22:00:00 403 401.1  339.721  0.10145 4.473e+03
## 27 1964-02-29 22:00:00 429 403.7  269.347  1.54092 1.416e+06
## 28 1964-03-31 21:00:00 425 414.0  259.528  0.68410 4.600e+04
## 29 1964-04-30 21:00:00 427 419.0  243.010  0.51466 2.336e+04
## 30 1964-05-31 21:00:00 409 423.0  230.631 -0.91990 7.522e+01
## 31 1964-06-30 21:00:00 402 422.0  226.004 -1.32784 1.471e+01
## 32 1964-07-31 21:00:00 409 419.7  229.720 -0.70405 1.784e+02
## 33 1964-08-31 21:00:00 419 419.2  225.306 -0.01128 2.849e+03
## 34 1964-09-30 21:00:00 404 420.7  218.311 -1.13070 3.237e+01
## 35 1964-10-31 21:00:00 429 418.8  220.263  0.68522 4.621e+04
## 36 1964-11-30 21:00:00 463 422.3  217.224  2.76467 1.893e+08
## 37 1964-12-31 21:00:00 428 423.7  670.553  0.16498 5.767e+03
## 38 1965-01-31 22:00:00 449 428.4  308.846  1.17159 3.233e+05
## 39 1965-02-28 22:00:00 444 439.4  258.623  0.28767 9.421e+03
## 40 1965-03-31 21:00:00 467 442.7  232.660  1.59164 1.735e+06
## 41 1965-04-30 21:00:00 474 451.6  232.174  1.47039 1.068e+06
## 42 1965-05-31 21:00:00 463 459.6  234.091  0.21942 7.170e+03
## 43 1965-06-30 21:00:00 432 463.0  226.561 -2.05920 7.892e-01
## 44 1965-07-31 21:00:00 453 458.0  243.320 -0.31957 8.302e+02
## 45 1965-08-31 21:00:00 462 458.7  238.287  0.21283 6.984e+03
## 46 1965-09-30 21:00:00 456 461.4  233.433 -0.35057 7.334e+02
## 47 1965-10-31 21:00:00 474 461.9  229.171  0.79614 7.201e+04
## 48 1965-11-30 21:00:00 514 466.7  227.455  3.13515 8.330e+08
## 49 1965-12-31 22:00:00 489 468.8  799.744  0.71489 5.203e+04
## 50 1966-01-31 22:00:00 475 486.9  324.948 -0.66095 2.119e+02
## 51 1966-02-28 22:00:00 492 483.6  263.148  0.51729 2.360e+04
## 52 1966-03-31 21:00:00 525 488.8  238.810  2.33980 3.459e+07
## 53 1966-04-30 21:00:00 527 502.3  250.430  1.56111 1.536e+06
## 54 1966-05-31 21:00:00 533 512.3  253.125  1.30124 5.431e+05
## 55 1966-06-30 21:00:00 527 521.5  254.930  0.34462 1.183e+04
## 56 1966-07-31 21:00:00 522 527.0  250.481 -0.31884 8.327e+02
## 57 1966-08-31 21:00:00 526 529.9  246.572 -0.24769 1.107e+03
## 58 1966-09-30 21:00:00 513 532.9  242.591 -1.27771 1.798e+01
## 59 1966-10-31 21:00:00 564 531.4  784.462  2.07913 1.220e+07
## 60 1966-11-30 22:00:00 599 560.3  345.623  3.43606 2.776e+09
## 61 1966-12-31 22:00:00 572 582.7  302.384  0.52061 2.392e+04
## 62 1967-01-31 22:00:00 587 583.4  277.399  0.21762 7.119e+03
## 63 1967-02-28 22:00:00 599 588.9  264.224  0.61950 3.552e+04
## 64 1967-03-31 21:00:00 601 596.4  258.300  0.28376 9.275e+03
## 65 1967-04-30 21:00:00 611 602.5  253.691  0.53094 2.493e+04
## 66 1967-05-31 21:00:00 620 609.9  250.921  0.63560 3.789e+04
## 67 1967-06-30 21:00:00 579 618.1  836.410 -2.47785 1.479e-01
## 68 1967-07-31 21:00:00 582 592.3  349.730 -1.83396 1.943e+00
## 69 1967-08-31 21:00:00 592 592.0  286.227 -0.92571 7.349e+01
## 70 1967-09-30 21:00:00 581 596.5  262.822 -0.95757 6.470e+01
## 71 1967-10-31 21:00:00 630 595.7  928.883  2.14392 1.580e+07
## 72 1967-11-30 22:00:00 663 628.0  349.863  3.17999 9.967e+08
## 73 1967-12-31 22:00:00 638 650.6  301.960  0.26863 8.730e+03
## 74 1968-01-31 22:00:00 631 651.1  284.202 -1.19203 2.533e+01
## 75 1968-02-29 22:00:00 645 648.4  282.576 -0.20144 1.332e+03
```

```
## 76 1968-03-31 21:00:00 682 651.2  279.036  1.84191 4.722e+06
## 77 1968-04-30 21:00:00 601 667.4  288.028 -3.91180 4.774e-04
## 78 1968-05-31 21:00:00 595 673.2 1542.864 -1.99143 1.035e+00
## 79 1968-06-30 21:00:00 521 608.3  403.266 -4.34945 8.291e-05
## 80 1968-07-31 21:00:00 521 612.0 2631.603 -1.77356 2.474e+00
## 81 1968-08-31 21:00:00 516 529.9  430.229 -0.66807 2.060e+02
## 82 1968-09-30 21:00:00 496 523.8  356.039 -1.47538 8.154e+00
## 83 1968-10-31 21:00:00 538 510.8  352.342  1.44789 9.763e+05
## 84 1968-11-30 21:00:00 575 523.7  358.331  2.70845 1.512e+08
## 85 1968-12-31 21:00:00 537 525.4 2569.300  0.22956 7.467e+03
## 86 1969-01-31 21:00:00 534 538.1  449.195 -0.19559 1.363e+03
## 87 1969-02-28 21:00:00 542 537.7  377.060  0.22238 7.255e+03
## 88 1969-03-31 21:00:00 538 541.9  365.266 -0.20414 1.317e+03
## 89 1969-04-30 21:00:00 547 541.9  359.179  0.26744 8.689e+03
## 90 1969-05-31 21:00:00 540 546.5  348.637 -0.35022 7.344e+02
##      L_lower l_lower   H_upper   L_upper l_upper
## 1  1.000e+00       1 1.000e+00 1.000e+00       1
## 2  1.000e+00       1 1.000e+00 1.000e+00       1
## 3  1.000e+00       1 1.000e+00 1.000e+00       1
## 4  1.000e+00       1 1.000e+00 1.000e+00       1
## 5  1.000e+00       1 1.000e+00 1.000e+00       1
## 6  1.000e+00       1 1.000e+00 1.000e+00       1
## 7  1.000e+00       1 1.000e+00 1.000e+00       1
## 8  1.000e+00       1 1.000e+00 1.000e+00       1
## 9  1.000e+00       1 1.000e+00 1.000e+00       1
## 10 1.000e+00       1 1.000e+00 1.000e+00       1
## 11 1.000e+00       1 1.000e+00 1.000e+00       1
## 12 1.000e+00       1 1.000e+00 1.000e+00       1
## 13 1.000e+00       1 1.000e+00 1.000e+00       1
## 14 1.000e+00       1 1.000e+00 1.000e+00       1
## 15 1.000e+00       1 1.000e+00 1.000e+00       1
## 16 1.000e+00       1 1.000e+00 1.000e+00       1
## 17 1.000e+00       1 1.000e+00 1.000e+00       1
## 18 1.000e+00       1 1.000e+00 1.000e+00       1
## 19 1.000e+00       1 1.000e+00 1.000e+00       1
## 20 1.000e+00       1 1.000e+00 1.000e+00       1
## 21 1.399e+03       1 6.351e+03 6.351e+03       1
## 22 1.907e+02       1 4.659e+04 4.659e+04       1
## 23 1.376e+04       1 6.457e+02 6.457e+02       1
## 24 1.437e+09       1 6.183e-03 1.000e+00       0
## 25 1.336e+05       1 6.649e+01 6.649e+01       1
## 26 4.473e+03       1 1.987e+03 1.987e+03       1
## 27 1.416e+06       1 6.273e+00 6.273e+00       1
## 28 4.600e+04       1 1.932e+02 1.932e+02       1
## 29 2.336e+04       1 3.805e+02 3.805e+02       1
## 30 7.522e+01       1 1.181e+05 1.181e+05       1
## 31 1.471e+01       1 6.040e+05 6.040e+05       1
## 32 1.784e+02       1 4.982e+04 4.982e+04       1
## 33 2.849e+03       1 3.119e+03 3.119e+03       1
## 34 3.237e+01       1 2.745e+05 2.745e+05       1
## 35 4.621e+04       1 1.923e+02 1.923e+02       1
## 36 1.893e+08       1 4.695e-02 1.000e+00       0
## 37 5.767e+03       1 1.541e+03 1.541e+03       1
## 38 3.233e+05       1 2.748e+01 2.748e+01       1
```

```
## 39 9.421e+03       1 9.432e+02 9.432e+02       1
## 40 1.735e+06       1 5.122e+00 5.122e+00       1
## 41 1.068e+06       1 8.318e+00 8.318e+00       1
## 42 7.170e+03       1 1.239e+03 1.239e+03       1
## 43 7.892e-01       1 1.126e+07 1.126e+07       1
## 44 6.552e+02       2 1.070e+04 1.070e+04       1
## 45 6.984e+03       1 1.272e+03 1.272e+03       1
## 46 7.334e+02       1 1.212e+04 1.212e+04       1
## 47 7.201e+04       1 1.234e+02 1.234e+02       1
## 48 8.330e+08       1 1.067e-02 1.000e+00       0
## 49 5.203e+04       1 1.708e+02 1.708e+02       1
## 50 2.119e+02       1 4.193e+04 4.193e+04       1
## 51 2.360e+04       1 3.765e+02 3.765e+02       1
## 52 3.459e+07       1 2.569e-01 2.569e-01       1
## 53 1.536e+06       1 5.787e+00 1.486e+00       2
## 54 5.431e+05       1 1.636e+01 1.636e+01       1
## 55 1.183e+04       1 7.511e+02 7.511e+02       1
## 56 8.327e+02       1 1.067e+04 1.067e+04       1
## 57 1.107e+03       1 8.029e+03 8.029e+03       1
## 58 1.798e+01       1 4.943e+05 4.943e+05       1
## 59 1.220e+07       1 7.287e-01 7.287e-01       1
## 60 2.776e+09       1 3.201e-03 2.333e-03       2
## 61 2.392e+04       1 3.715e+02 1.000e+00       0
## 62 7.119e+03       1 1.248e+03 1.248e+03       1
## 63 3.552e+04       1 2.501e+02 2.501e+02       1
## 64 9.275e+03       1 9.581e+02 9.581e+02       1
## 65 2.493e+04       1 3.565e+02 3.565e+02       1
## 66 3.789e+04       1 2.345e+02 2.345e+02       1
## 67 1.479e-01       1 6.009e+07 6.009e+07       1
## 68 2.873e-01       2 4.574e+06 4.574e+06       1
## 69 1.000e+00       0 1.209e+05 1.209e+05       1
## 70 6.470e+01       1 1.373e+05 1.373e+05       1
## 71 1.580e+07       1 5.623e-01 5.623e-01       1
## 72 9.967e+08       1 8.915e-03 5.013e-03       2
## 73 8.730e+03       1 1.018e+03 1.000e+00       0
## 74 2.533e+01       1 3.509e+05 3.509e+05       1
## 75 1.332e+03       1 6.673e+03 6.673e+03       1
## 76 4.722e+06       1 1.882e+00 1.882e+00       1
## 77 1.000e+00       0 1.861e+10 1.861e+10       1
## 78 1.035e+00       1 8.587e+06 8.587e+06       1
## 79 1.000e+00       0 1.072e+11 1.072e+11       1
## 80 2.474e+00       1 3.592e+06 3.592e+06       1
## 81 2.060e+02       1 4.314e+04 4.314e+04       1
## 82 8.154e+00       1 1.090e+06 1.090e+06       1
## 83 9.763e+05       1 9.101e+00 9.101e+00       1
## 84 1.512e+08       1 5.879e-02 1.000e+00       0
## 85 7.467e+03       1 1.190e+03 1.190e+03       1
## 86 1.363e+03       1 6.518e+03 6.518e+03       1
## 87 7.255e+03       1 1.225e+03 1.225e+03       1
## 88 1.317e+03       1 6.745e+03 6.745e+03       1
## 89 8.689e+03       1 1.023e+03 1.023e+03       1
## 90 7.344e+02       1 1.210e+04 1.210e+04       1
##  [ reached 'max' / getOption("max.print") -- omitted 90 rows ]
```