

– 20 Shades of λ –

Modeling Trends in Time-Series using Fused Lasso*

Martin Haas[†] & Benedikt Schmehr[‡]

February 6, 2019

Abstract

This paper aims to discuss a trend estimation method for Time-Series data using regularization methods, in particular the Fused Lasso developed by R. Tibshirani et al. (2005) and extended by Qian et al. (2016). We include q th order polynomials at each point in time, solving for a penalized least squares solution while urging successive parameters to be equal. From this we obtain a sparse solution for our trend estimation parameters.

*Final examination for completion of the seminar Quantitative Methods II at Zeppelin Universität, Friedrichshafen - *Supervisor:* MSc Philipp Heiler, University of Konstanz

[†]m.haas@zeppelin-university.net

[‡]b.schmehr@zeppelin-university.net

Contents

1	Introduction	5
2	Literature review	6
3	Data	7
3.1	Simulation	7
3.2	Real trend data	9
4	Model	10
4.1	Modeling approaches	10
4.2	Formal Model Definitions	11
4.2.1	P-GFL	11
4.2.2	P-SGFL	14
4.3	Estimation	14
4.4	Selection of the Tuning Parameter	15
5	Results	16
5.1	P-GFL	16
5.2	P-SGFL	17
6	Discussion and Conclusion	18
References		19
A	Appendix A P-GFL Results	21
A.1	Linear Model (P-GFL)	21
A.1.1	Simulated Trend Data (P-GFL)	21
A.1.2	Real Trend Data (P-GFL)	25
A.2	Quadratic Model (P-GFL)	28
A.2.1	Simulated Trend Data (P-GFL)	28
A.2.2	Real Trend Data (P-GFL)	32
A.3	Cubic Model (P-GFL)	35
A.3.1	Simulated Trend Data (P-GFL)	35

A.3.2	Real Trend Data (P-GFL)	39
Appendix B	P-SGFL Results	42
B.1	Cubic Model (P-SGFL)	42
B.1.1	Simulated Trend Data (P-SGFL)	42
B.1.2	Real Trend Data (P-SGFL)	43
Appendix C	Comparison of P-GFL and P-SGFL	49

List of Figures

1	Simulated time-series plots	8
2	Real trend data	9
3	The effect of the tuning parameter	16
4	Selected IC for all trends in the linear model	21
5	Polynomial trend estimation using the linear model	22
6	Jump trend estimation using the linear model	23
7	Seasonal trend estimation using the linear model	24
8	Selected IC for all trends in the linear model	25
9	JNJ trend estimation using the linear model	26
10	PG trend estimation using the linear model	27
11	Selected IC for all trends in the quadratic model	28
12	Polynomial trend estimation using the quadratic model	29
13	Jump trend estimation using the quadratic model	30
14	Seasonal trend estimation using the quadratic model	31
15	Selected IC for all trends in the linear model	32
16	JNJ trend estimation using the quadratic model	33
17	PG trend estimation using the quadratic model	34
18	Selected IC for all trends in the cubic model	35
19	Polynomial trend estimation using the cubic model	36
20	Jump trend estimation using the cubic model	37
21	Seasonal trend estimation using the cubic model	38
22	Selected IC for all trends in the linear model	39

23	JNJ trend estimation using the cubic model	40
24	PG trend estimation using the cubic model	41
25	Selected IC for all trends in the sparse cubic model	42
29	Selected IC for all trends in the sparse cubic model	43
26	Polynomial trend estimation using the sparse cubic model	44
27	Jump trend estimation using the sparse cubic model	45
28	Seasonal trend estimation using the cubic model	46
30	JNJ trend estimation using the sparse cubic model	47
31	PG trend estimation using the sparse cubic model	48
32	Comparison of P-GFL and P-SGFL	50

1 Introduction

Time-series data often inherits trends which can either interfere in the statistical analysis or be the actual objects of interest themselves. For instance, trends might overlay given information within the time-series that we actually want to measure and thereby influence the magnitude of observed effects and factors. Also, many time-series models assume that the considered data is stationary, a prerequisite which frequently has to be rejected for data which follows some (deterministic) trend, making these models inappropriate to use. In order to still utilize such models one would consider transforming the data to be stationary e.g. by calculating the first differences or logarithms of it. However, these approaches can't be applied to every time-series, or the transformed data might still contain deterministic trends. In these cases, other methods for de-trending are required.

This issue gave rise to a research area on sophisticated trend-estimation methods in statistical analysis. Some well-known examples for trend filtering methods are Hodrick-Prescott filtering, moving average filtering, smoothing splines and exponential smoothing. Some of these methods tend to be infeasible when working with large time-series data sets or produce presumably dense (i.e. overfitted) solutions. The motivation of this paper is to develop a trend-estimation method that can be applied to deterministic trends in time-series and is feasible for large data sets. By exploiting the features of regularization methods, we aim to prevent our model from overfitting and urge it to produce sparse solutions. Because time-series data provides a natural ordering, the Grouped and Fused Lasso are reasonable approaches to accomplish this task. Regarding the popularity of regularization in recent years, there still exists a rather scarce literature on filtering trends in the time-series context using regularization methods. One of the few would be the ℓ_1 Trend Filtering method proposed by Kim et al. (2009) and developed further by R. J. Tibshirani (2014).

In this paper we propose a trend estimation method that models trend data as smooth polynomials of the time t . The term smoothness here means that the model should *not overfit* the trend data, which might be noisy. Additionally, the model should be able to conform to structural breaks or jumps in the trend. To achieve this, we build on the Lasso regularization methods, which was originally

proposed by R. Tibshirani (1996). More specifically, we utilize the *Grouped Fused Lasso* (GFL) of Qian et al. (2016), which is a generalization of the Lasso. Doing this, we develop the following two approaches.

The *first approach* employs the GFL-method on polynomials of t as regressors. The method works through shrinking and selecting regression coefficients, that are grouped, in order to get a sparse polynomial estimate of the trend. It is closely related to *piecewise regression* and aims at identifying the breakpoints, or pieces, for which then standard OLS-regression is employed. Our *second approach* builds on top of this. In the GFL, sparsity is only employed on the group level. This means that for a given order of polynomial, say three, only cubic polynomials get fitted to the trend. By introducing sparsity on the individual regressor level as well, we aim to make the resulting polynomial more “flexible”, i.e. that it also allows for polynomials of different order.

The remainder of this paper will be structured as follows. Within section 2 we give a further insight into the relevant literature. In section 3 we explain how we simulated the data used to train the model and describe the real trend data we filtered with our model. We introduce our model in section 4 and section 5 presents the results of our trend estimations. We discuss our results and conclude in section 6. Further, in the Appendix we provide plots, tables and other information not included in the running text.

2 Literature review

In recent years several trend estimation models have been developed in diverse settings employing a wide range of different statistical methods. Among them, also some models utilizing regularization methods were proposed (Kim et al. 2009; R. J. Tibshirani 2014; Qian et al. 2016). We build our theoretical and practical work on the following two foundations of the established literature:

Kim et al. (2009, p. 340) suggest their ℓ_1 trend filtering as a variation of the Hodrick-Prescott filter with the main difference, that “the ℓ_1 trend filter produces trend estimates that are smooth in the sense of being piecewise linear”. For this reason they recommend using their method to filter time-series that follow some

“underlying piecewise linear trend” (Kim et al. 2009, p. 340). Thus, their method is to some extend restricted to estimating changes in the underlying *linear* trend of some observed time-series that could also exhibit a *non-linear* trend, e.g. (piecewise) *quadratic* or *cubic*. We can even think of trends that show structural breaks, i.e. points where the underlying functional form is not continuously differentiable. So far, the method by Kim et al. (2009) in its reported form is not suited for these types of trends, which leaves it open to extensions.

Qian et al. (2016) propose a method that is able to “correctly determine the unknown number of breaks” in a time-series. In particular, they use the Group Fused Lasso to detect structural changes in a linear regression framework by estimation of the true trend parameters and checking for changes in their behavior and magnitude. For example if a set of parameters $(\beta_{t0}, \beta_{t1}, \dots, \beta_{tq})$ changes from $t = i$ to $t = j$ with $i \neq j$ it hints towards a structural change in the time-series. We widely obtain their method in our model setup.

We contribute to the two main references presented above in two ways: We aim to (*i*) escape the linearity restriction of the ℓ_1 filtering by including q th order polynomials at each point in time and shrinking towards a sparse solution and (*ii*) enable the modeling of trends with structural breaks by estimating their number and position and incorporating them in the model.

3 Data

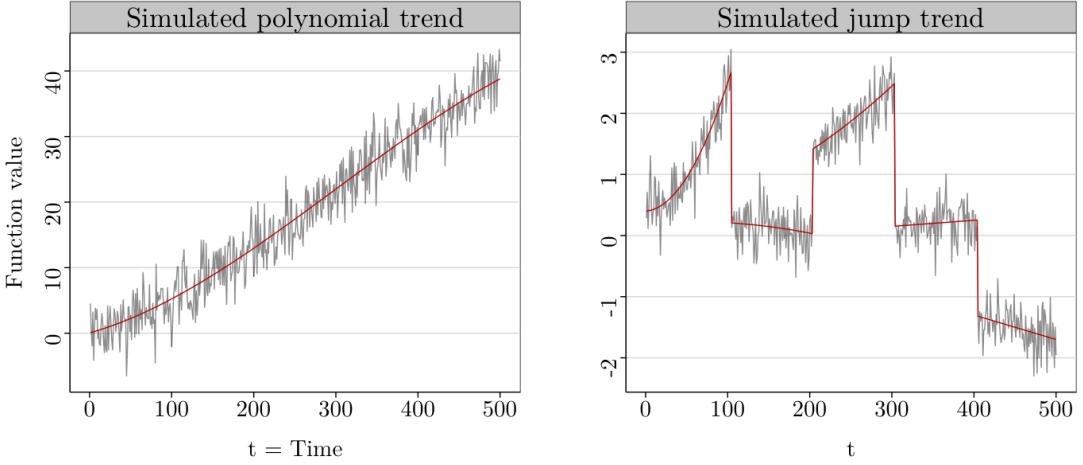
3.1 Simulation

We simulate training data according to the following equation, which will further be called the Data Generating Process (*DGP*):

$$y_t = f(t) + \varepsilon_t \quad (3.1.1)$$

where $f(t)$ is the deterministic part of the trend (i.e. a function of time) and ε_t is a (potentially Gaussian) white noise term. For the deterministic part of the trend we simulate three different types of functions: continuous polynomials, piecewise polynomial functions and seasonal trend functions as defined in the following and

two of them shown in [Figure 1](#).



[Figure 1](#): *Two possible versions of the simulated trends with the functional form of a (piecewise) polynomial. The red line corresponds to the true underlying trend and the gray line represents the trend after adding the white noise term.*

The simulated continuous polynomial can be described mathematically using [Equation 3.1.1](#), where we set the deterministic trend component as a function of the form given by [Equation 3.1.2](#).

$$f(t) = \sum_{i=0}^q b_i t^i. \quad (3.1.2)$$

We use [Equation 3.1.1](#) and [3.1.2](#) again to define the *piecewise* polynomial functions, which can be described mathematically as continuous polynomial functions within certain *non-overlapping* intervals being subsets of the full set of time points.

$$f(t) = \begin{cases} \sum_{i=0}^q b_i t^i & \text{for } t \in I_j \subseteq \{x_t\}_{t=1}^T \\ \text{undef.} & \text{else} \end{cases} \quad (3.1.3)$$

with j the number of intervals. The simulated seasonal trend is modeled the same way as the piecewise polynomial, where the functions within the intervals are simply constants, while manually defining the seasons (usually four) as intervals of equal length.

As a convention, we set the variance of the white noise term in such a way that we obtain a signal-to-noise ratio of 4 (80% signal to 20% noise), following Hastie et al. (2009, p. 401). The jump points of the piecewise polynomial trend as well as the actual value of the white noise term are chosen randomly but are seeded to make our results reproducible. The coefficients of the underlying polynomial function are set manually. For each of the possible simulations we draw 500 data points from the function of the trend and add the noise according to our DGP.

3.2 Real trend data

As for the real trend data example we use intraday quotes of two selected stocks (Johnson & Johnson and Procter & Gamble) on a frequency of one second. In total we are looking at 982,800 seconds of data. We calculate the continuously compounded returns and account for a frequently used intraday volatility measure, the absolute 5-minute returns. We achieve this by summing up every 300 successive 1-second returns and taking the absolute value of the outcome. This leaves us with a total of 3,234 data points, corresponding to 42 trading days. Finally, we obtain the average trading day, by calculating the averages over all 5-minute-intervals.

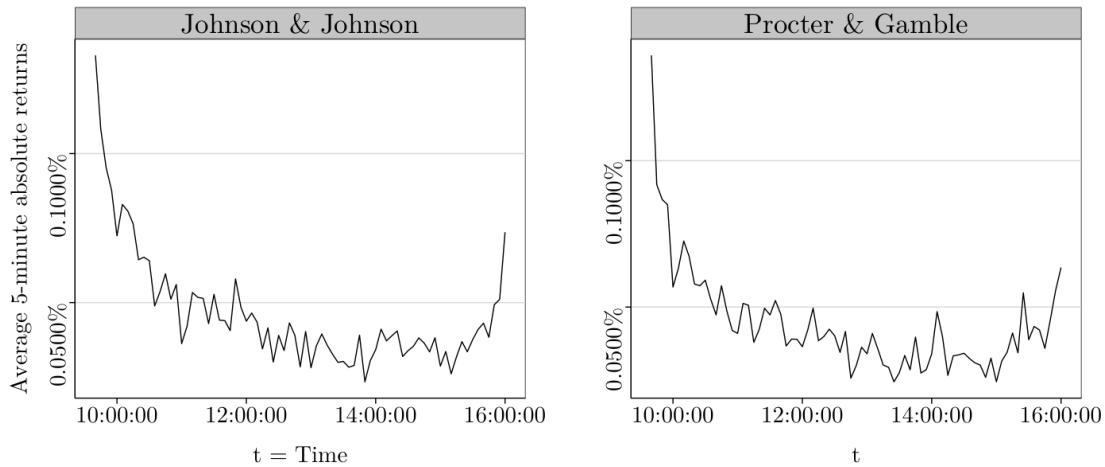


Figure 2: The plots show the average absolute 5-minute returns of 42 trading days.

Especially 5-minute absolute returns exhibit some characteristic pattern of intraday data, shown graphically in Figure 2. The intraday volatility, on average,

seems to peak at the beginning and at the end of a trading day. This phenomenon however can be explained by the common habit of traders to position themselves at the start and at the end of a day, whereas the low at midday might be due to lunch breaks. In any case, a clear pattern like this can be viewed as an “intraday trend” that might overlay information that we are interested in. Thus, we try to estimate this particular intraday trend using our model. We report the results within section 5.

4 Model

In this section we aim to first give an intuition for our trend estimation approaches and continue to formalize them. We then detail out the estimation procedure, including the final model selection aspects.

4.1 Modeling approaches

Our first approach to trend-estimation aims at the “construction” of a polynomial as estimate of the true trend in noisy data. To achieve this, we construct a theoretical “maximum model”, that contains all terms of a polynomial of order up to q at each point in time. This means that each point in time has a *group* of regressors and parameters. Now, for a given time interval, it might be that a specific functional form of the trend is the same for all points. The correct identification of these intervals will be achieved through the method of (Qian et al. 2016), which employs a combination of the Fused Lasso and the Grouped Lasso. The final trend estimate is then derived by estimating individual OLS-regressions between the given break points. This approach will be called *P-GFL* and carried out for every considered level of q individually.

However, the Grouped Fused Lasso (GFL) procedure only selects variables on the group level, meaning that the individual terms of the polynomials are fixed. To select individual variables at the group level, one can use the Sparse Grouped Lasso (Simon et al. 2013; Friedman et al. 2010). By employing this method, we can use one maximum model with a high order (high q), and the polynomial order of individual intervals gets selected “automatically”. This *second approach* will be

called Polynomial Sparse Fused Grouped Lasso (*P-SGFL*).

4.2 Formal Model Definitions

4.2.1 P-GFL

For a given time-series, we model **each observation** as a polynomial of order q :

$$y_t = \beta_{t0}t^0 + \beta_{t1}t^1 + \cdots + \beta_{tq}t^q + \varepsilon_t$$

$$y_t = \left(\sum_{j=0}^q \beta_{tj}t^j \right) + \varepsilon_t$$

with

- $t = 1, 2, \dots, T$ the time
- $y_t = y_1, y_2, \dots, y_T$ the given time series
- $j = 0, 1, \dots, q$ the exponents of the polynomial terms
- β_{tj} the model parameters
- ε_t an error term

Using matrix notation, this can be expressed as:

$$y_t = \boldsymbol{\tau}'_t \boldsymbol{\beta}_t + \varepsilon_t$$

where we define:

$$\boldsymbol{\beta}_t \equiv \begin{bmatrix} \beta_{t0} & \beta_{t1} & \dots & \beta_{tq} \end{bmatrix}' \quad \text{and} \quad \boldsymbol{\tau}_t \equiv \begin{bmatrix} t^0 & t^1 & \dots & t^q \end{bmatrix}'$$

We further use the column vector as default.

Modeling every single point in time, through a regression as defined above, is a special case of *piecewise linear regression*, where the sample is split into the maximum number of *pieces*. The splitting of the sample is achieved by using indicator functions, which in practice are dummy vectors. We define:

$$\underset{T \times 1}{\boldsymbol{t}} \equiv \begin{bmatrix} 1 & 2 & \dots & T \end{bmatrix}' \quad \text{and} \quad \underset{T \times 1}{\boldsymbol{I}_i} \equiv I_i(\boldsymbol{t}) = \begin{cases} 1 & \text{for } t = i \\ 0 & \text{else} \end{cases}$$

which e.g. is for $i = 1$ and $t = 4$:

$$\underset{4 \times 1}{\boldsymbol{I}_1} = \begin{bmatrix} 1 & 0 & 0 & 0 \end{bmatrix}'$$

The piecewise regression can now be expressed as:

$$\begin{aligned} \boldsymbol{y} &= \boldsymbol{I}_1 \boldsymbol{\tau}'_1 \boldsymbol{\beta}_1 + \boldsymbol{I}_2 \boldsymbol{\tau}'_2 \boldsymbol{\beta}_2 + \dots + \boldsymbol{I}_T \boldsymbol{\tau}'_T \boldsymbol{\beta}_T + \boldsymbol{\varepsilon} \\ &= \sum_{i=1}^T \boldsymbol{I}_i \boldsymbol{\tau}'_i \boldsymbol{\beta}_i + \boldsymbol{\varepsilon} \end{aligned}$$

with

$$\underset{T \times 1}{\boldsymbol{y}} \equiv \begin{bmatrix} y_1 & y_2 & \dots & y_T \end{bmatrix}' \quad \text{and} \quad \underset{T \times 1}{\boldsymbol{\varepsilon}} \equiv \begin{bmatrix} \varepsilon_1 & \varepsilon_2 & \dots & \varepsilon_T \end{bmatrix}'$$

Note that the subscript i of the parameter vector now indicates the splitting points of the piecewise regression, which coincide exactly with t . Finally defining $\boldsymbol{X}_i \equiv \boldsymbol{I}_i \boldsymbol{\tau}'_i$ gives us the model in a more usual form.

$$\boldsymbol{y} = \sum_{i=1}^T \boldsymbol{X}_i \boldsymbol{\beta}_i + \boldsymbol{\varepsilon} \tag{4.2.1}$$

where \boldsymbol{X}_i now has the dimensions $T \times (q+1)$.

From [Equation 4.2.1](#) we can derive the number of parameters p of our model: each parameter vector $\boldsymbol{\beta}_i$ consists of $q+1$ elements; furthermore, we have T parameter vectors (one for each point in time). Thus, we have $p = (q+1)T$ parameters.

This means that the number of parameters exceeds the number of observations for polynomials of order q greater than 0, leading to standard OLS-estimation to be infeasible for higher polynomials. Therefore, as we need to strengthen the requirement of *variable selection* further, we assume the following.

Assumption 1 *Sparsity of the parameters:*

$$\pi = \sum_{i=1}^T \|\beta_i\|_0 \leq T$$

The true number of parameters, π , is smaller than the number of observations, T .

In order to estimate the polynomial trend, we follow the approach of Qian et al. (2016), who employ a variant of the *Fused Lasso* (R. Tibshirani et al. 2005), which optimizes the following objective function:

$$\min_{\beta_i} \frac{1}{T} \left\| \mathbf{y} - \sum_{i=1}^T \mathbf{X}_i \beta_i \right\|_2^2 + \lambda \sum_{i=2}^T |\beta_i - \beta_{i-1}| \quad (4.2.2)$$

where $\|\cdot\|_2$ denotes the ℓ_2 norm and λ is a tuning or penalty parameter. Following the approach further, we define: $\boldsymbol{\theta}_1 = \boldsymbol{\beta}_1$, $\boldsymbol{\theta}_i = \boldsymbol{\beta}_i - \boldsymbol{\beta}_{i-1}$ for $i = 2, 3, \dots, T$ and

$${}_{(q+1)T \times 1} \boldsymbol{\theta} \equiv [\boldsymbol{\theta}_1' \ \boldsymbol{\theta}_2' \ \dots \ \boldsymbol{\theta}_T']' \quad \text{and} \quad {}_{(q+1) \times T} \boldsymbol{\beta} \equiv [\boldsymbol{\beta}_1' \ \boldsymbol{\beta}_2' \ \dots \ \boldsymbol{\beta}_T']'$$

which gives us the full parameter vectors. Finally, by defining:

$${}_{T \times T(q+1)} \mathbf{X} = \begin{bmatrix} \boldsymbol{\tau}_1' & & & \\ & \boldsymbol{\tau}_2' & & \\ & & \ddots & \\ & & & \boldsymbol{\tau}_T' \end{bmatrix}, \quad {}_{T(q+1) \times T(q+1)} \mathbf{A}^* = \begin{bmatrix} \mathbb{I}_p & & & \\ \mathbb{I}_p & \mathbb{I}_p & & \\ \vdots & \vdots & \ddots & \\ \mathbb{I}_p & \mathbb{I}_p & \dots & \mathbb{I}_p \end{bmatrix}$$

and ${}_{T \times (q+1)T} \mathbf{X}^* = \mathbf{X} \mathbf{A}^*$, we can now rewrite Equation 4.2.1 in an alternative format as $\mathbf{y} = \mathbf{X} \boldsymbol{\beta} + \boldsymbol{\varepsilon} = \mathbf{X}^* \boldsymbol{\theta} + \boldsymbol{\varepsilon}$ and minimizing Equation 4.2.2 is equal to the optimization of the following *Grouped Lasso* (Yuan et al. 2006) objective function:

$$\min_{\boldsymbol{\theta}_i} \frac{1}{T} \|\mathbf{y} - \mathbf{X}^* \boldsymbol{\theta}\|_2^2 + \lambda \sum_{i=2}^T |\boldsymbol{\theta}_i| \quad (4.2.3)$$

Note that at this state we have defined our P-GFL approach, which equals the GFL by Qian et al. (2016) with polynomials in t as predictors. As the goal of the P-GFL

is to determine the dates at which breaks in the time series occur, we define:

$$\mathbf{T} = \begin{bmatrix} T_1 & T_2 & \dots & T_m \end{bmatrix} \quad \text{as well as } T_0 \equiv 1 \text{ and } T_{m+1} \equiv T + 1$$

The variable m now indicates the total number and T the date of the break points in the time series. We estimate m as the number of nonzero entries in the estimated Grouped Lasso parameters, $\hat{m} = \|\hat{\boldsymbol{\theta}}_\lambda\|_0$ and the break dates \mathbf{T} as those values of the time vector \mathbf{t} , for which the respective parameter vector $\boldsymbol{\theta}_{t\lambda} \neq 0$.

4.2.2 P-SGFL

The extension of the P-GFL with sparsity simply requires an additional penalty on the individual parameters (Simon et al. 2013). The objective function to be optimized for the P-SGFL is given by extending [Equation 4.2.3](#) in the following way:

$$\min_{\boldsymbol{\theta}_i} \frac{1}{T} \|\mathbf{y} - \mathbf{X}^* \boldsymbol{\theta}\|_2^2 + (1 - \alpha)\lambda \sum_{i=2}^T |\boldsymbol{\theta}_i| + \alpha\lambda \|\boldsymbol{\beta}\|_1 \quad (4.2.4)$$

The α parameter weights the amount of group Lasso against the normal Lasso penalty. We set it to 0.5 by default, which means that groups of variables and individual variables get an equal penalty.

4.3 Estimation

Standard OLS-estimation of the penalized Least Squares problems defined above is not feasible, since they are not easily differentiable. Even if they were, our modeling approach always inherits a larger number of parameters than observations, since $p = (q + 1)T > T$, which means that there is no unique solution to the OLS first order condition $\mathbf{X}^{*\prime} \mathbf{X}^* \boldsymbol{\theta} = \mathbf{X}^{*\prime} \mathbf{y}$.

So, in order to estimate our models, we employ the “orthogonalizing EM” algorithm developed by Xiong et al. (2016), which is available conveniently in the R-package `oem`. We choose it, since it allows for individual customization of the penalization parameters and calculates the solutions in a reasonable amount of time for our given problems. We employ the same estimation method for both the P-GFL and the P-SGFL.

We standardize the regressor matrix \mathbf{X}^* , so that the sum of squares of each column is equal to one. This ensures that all estimated parameters have the same scale and the penalty applies equally to every variable (Xiong et al. 2016, p. 10). Since we do not standardize or center our dependent variable (the trend to be estimated), we include an intercept in the estimation step. The resulting estimated trend is then calculated as:

$$\hat{\mathbf{y}} = \mathbf{X}_{pred}^* \hat{\boldsymbol{\theta}}$$

where we now include intercept terms in the regressor matrix and parameter vector:

$$\mathbf{X}_{pred}^*_{T \times T(q+1)+1} = \begin{bmatrix} 1 & \boldsymbol{\tau}_1' \\ 1 & \boldsymbol{\tau}_2' & \boldsymbol{\tau}_2' \\ \vdots & \vdots & \vdots & \ddots \\ 1 & \boldsymbol{\tau}_T' & \boldsymbol{\tau}_T' & \dots & \boldsymbol{\tau}_T' \end{bmatrix}_{T(q+1)+1 \times 1} \quad \hat{\boldsymbol{\theta}} = \begin{bmatrix} \hat{\theta}_0 \\ \hat{\theta}_1 \\ \vdots \\ \hat{\theta}_T \end{bmatrix}$$

Since we are only interested in the estimated trend $\hat{\mathbf{y}}$, we do not transform the GFL-parameters $\hat{\boldsymbol{\theta}}$ back into the Fused Lasso parameters $\hat{\boldsymbol{\beta}}$. The estimated number of breakpoints, \hat{m} , is calculated as described in [subsubsection 4.2.1](#).

4.4 Selection of the Tuning Parameter

The choice of the tuning parameter λ is a crucial step in the Lasso estimation procedure. In our method, it is applied to the *fused parameters* $\boldsymbol{\theta}$. Increasing the tuning parameter increases the penalty, resulting in smaller parameter estimates. Since our fusing parameters $\boldsymbol{\theta}$ measure the differences between consecutive “normal” parameters $\boldsymbol{\beta}$, an increased tuning parameter would make parameters near to each other more alike, or even equal. This is illustrated in [Figure 3](#): as lambda gets larger, the estimated trend becomes more “even”. For the maximum value of λ , all parameters get reduced to zero and only the intercept remains. For λ approaching zero, our model would fit the date perfectly, since we included polynomials for each observed data point.

In accordance with Qian et al. (2016, p. 1387), tuning parameter choice is done

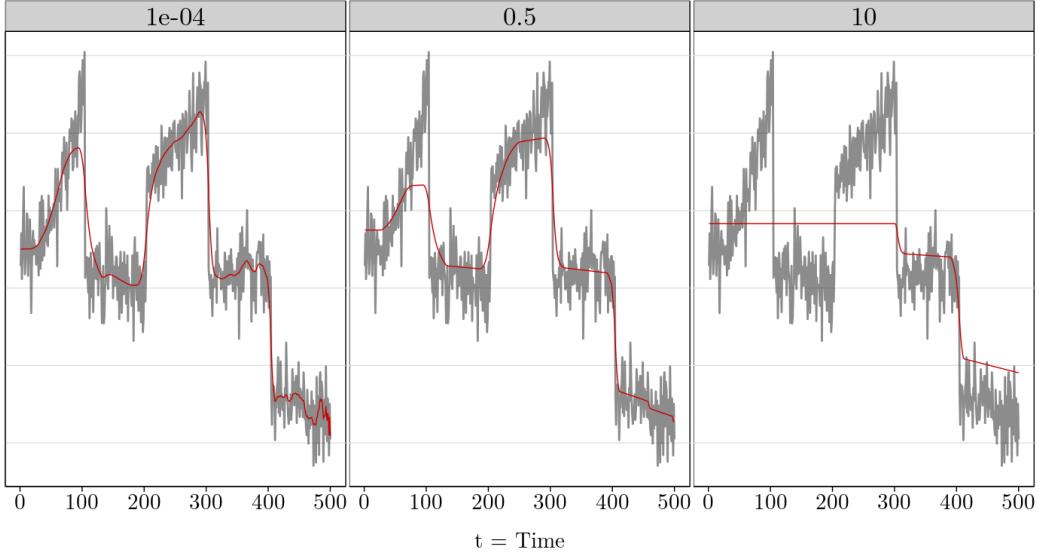


Figure 3: Example of estimated trends (red line) for different tuning parameters

by selecting the λ for which the following information criterion is minimized:

$$IC_\lambda = \log(\hat{\sigma}_\lambda^2) + \rho_T p(\hat{m}_\lambda + 1),$$

where we estimate $\hat{\sigma}_\lambda^2$ as

$$\hat{\sigma}_\lambda^2 = \frac{1}{T} \sum_{i=1}^T (y_i - \hat{y}_{\lambda i})^2$$

and set $\rho = 1/\sqrt{T}$ (Qian et al. 2016, p. 1392). Note that we use lambda in the subscript to denote variables for a particular value of lambda.

5 Results

5.1 P-GFL

For the P-GLF, we consider three orders of polynomials for our regressors: linear ($q = 1$), quadratic ($q = 2$) and cubic ($q = 3$) and consequently apply these estimates to each of the three trends (polynomial, jump and seasonal), giving us 9 results.

In accordance with Qian et al. (2016), we consider 20 values of λ . The full

results, i.e. the fitted trends as well as the distribution of the estimated m over the information criteria can be looked up in the Appendix [P-GFL Results](#).

As can be seen there, all three models are able to conform somewhat to the trends, even the complicated jump and seasonal trend. However the fit of the model seems to be uneven: the estimated trend tends to fit “faster” as t becomes larger. While the parameters with higher t accommodate to the data, sections with lower t seem to be asymmetrically penalized by the tuning parameter. For low values of λ , one can observe overfitting of the right end of the estimated trend, while the left end is underfitted.

Another issue is that the information criterion (IC) always selects the model with the lowest number of break points m . This might be due to the low number (20) of considered parameters, which makes their distribution too “coarse”. But our observation holds even for estimations with 300 values for the tuning parameter. The `oem`-package also allows to set the size of the minimum value of λ as a fraction of the maximum λ , which is the value at which all coefficients are pushed down to zero. For fractions of 0.01 (the default), 0.001, 0.0001 and 0.00001, the result still remains the same.

For the real data results, also reported in the Appendix, we observe similar problems. Still, our method seems to match the general behavior of the average intraday trend quite well, separating some stronger bumps while keeping sparse in its nature. Nevertheless we note some overfitting on the far right of the plot. Though, as lamda increases it urges the overfitting to decrease it in turn causes the left side to underfit.

5.2 P-SGFL

The results of our second approach, which aimed at fitting a smooth polynomial on the data, is reported in appendix [subsection B.1](#). The P-SGFL still exhibits the asymmetric fitting of the trend and the IC also always selects the model with the lowest number of break points.

For the current setting of $\alpha = 0.5$, the two estimations are virtually indistinguishable, as can be seen from graphical comparison alone in [Figure 32](#) in appendix [Comparison of P-GFL and P-SGFL](#).

6 Discussion and Conclusion

In this paper, we modeled deterministic trends via piecewise regressions with polynomials of the time t as regressors. Through the fusing of groups of parameters over different time-steps, we aimed to model a smooth polynomial as well as structural breaks. For the estimation of the parameters we relied on the Grouped Fused Lasso and the Sparse Grouped Fused Lasso, both methods from the domain of statistical learning.

However, as it turned out, we were not able to establish a working trend-estimation method. More precisely, our results have the following *limitations*:

- Asymmetric fit of the trend estimates
- Inability of the information criterion to select the correct number of break-points
- Inconclusive results of the P-SGFL approach

The problem of asymmetric fit is that for a given value of the tuning parameter, parts of the estimated trend with higher t fit “closer” to the data as parts with lower t . This might be due to the fact that the predictor matrix \mathbf{X}^* used in estimation is not correctly standardized Consider this: since the upper triangle of this matrix consist of zeros, and movement from the left columns to the right indicates moving from t_1 to T , the predictors are of different size. Potential solutions to this problem include the development of a correct standardization method or a weighting scheme to counterbalance this effect on the level of the tuning parameter.

The inability of the information criterion to select the correct model might be caused due to the asymmetric fitting problem since it is calculated from the results of the estimation procedure, m and $\hat{\sigma}_\lambda$. It makes further continuation of this project with Monte-Carlo simulations and performance evaluation of different methods infeasible, since the problem of the choice of the tuning parameter is unsolved.

The P-SGFL extension relies on the correct working of the P-GFL model and therefore limited in the same way.

Author's Declaration

We hereby declare that the work in this dissertation was carried out in accordance with the requirements of the university's regulations and that it has not been submitted for any other academic award. Except where indicated by specific references in the text, the work is the candidates own work. Work done in collaboration with, or with the assistance of others, is indicated as such. Any views expressed in the dissertation are those of the authors.

References

- Friedman, Jerome et al. (Feb. 11, 2010). "A note on the group lasso and a sparse group lasso (corrected version)". In: Original: arXiv: 1001.0736. URL: <http://statweb.stanford.edu/~tibs/ftp/sparse-glasso.pdf>.
- Hastie, Trevor et al. (2009). *The elements of statistical learning: data mining, inference, and prediction*. 2nd ed. New York, NY: Springer.
- Kim, Seung-Jean et al. (2009). "Trend Filtering". In: *SIAM review* 51.2, pp. 339–360. URL: <https://www.jstor.org/stable/pdf/25662288.pdf>.
- Qian, Junhui et al. (Dec. 2016). "Shrinkage estimation of regression models with multiple structural changes". In: *Econometric Theory* 32.6, pp. 1376–1433. ISSN: 0266-4666, 1469-4360. DOI: [10.1017/S0266466615000237](https://doi.org/10.1017/S0266466615000237). URL: http://www.journals.cambridge.org/abstract_S0266466615000237 (visited on 10/13/2018).
- Simon, Noah et al. (Apr. 2013). "A Sparse-Group Lasso". In: *Journal of Computational and Graphical Statistics* 22.2, pp. 231–245. ISSN: 1061-8600, 1537-2715. DOI: [10.1080/10618600.2012.681250](https://doi.org/10.1080/10618600.2012.681250). URL: <http://www.tandfonline.com/doi/abs/10.1080/10618600.2012.681250> (visited on 01/08/2019).
- Tibshirani, Robert (1996). "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)*, pp. 267–288. URL: https://www.jstor.org/stable/2346178?seq=1#metadata_info_tab_contents.
- Tibshirani, Robert et al. (Feb. 2005). "Sparsity and smoothness via the fused lasso". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*

67.1, pp. 91–108. ISSN: 1369-7412, 1467-9868. DOI: [10.1111/j.1467-9868.2005.00490.x](https://doi.org/10.1111/j.1467-9868.2005.00490.x). URL: <http://doi.wiley.com/10.1111/j.1467-9868.2005.00490.x> (visited on 10/16/2018).

Tibshirani, Ryan J. (Feb. 2014). “Adaptive piecewise polynomial estimation via trend filtering”. In: *The Annals of Statistics* 42.1, pp. 285–323. ISSN: 0090-5364. DOI: [10.1214/13-AOS1189](https://doi.org/10.1214/13-AOS1189). URL: <http://projecteuclid.org/euclid-aos/1395234979> (visited on 01/10/2019).

Xiong, Shifeng et al. (July 2, 2016). “Orthogonalizing EM: A Design-Based Least Squares Algorithm”. In: *Technometrics* 58.3, pp. 285–293. ISSN: 0040-1706, 1537-2723. DOI: [10.1080/00401706.2015.1054436](https://doi.org/10.1080/00401706.2015.1054436). URL: <http://www.tandfonline.com/doi/full/10.1080/00401706.2015.1054436> (visited on 01/15/2019).

Yuan, Ming et al. (Feb. 2006). “Model selection and estimation in regression with grouped variables”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 68.1, pp. 49–67. ISSN: 1369-7412, 1467-9868. DOI: [10.1111/j.1467-9868.2005.00532.x](https://doi.org/10.1111/j.1467-9868.2005.00532.x). URL: <http://doi.wiley.com/10.1111/j.1467-9868.2005.00532.x> (visited on 01/14/2019).

Appendix A P-GFL Results

A.1 Linear Model (P-GFL)

A.1.1 Simulated Trend Data (P-GFL)

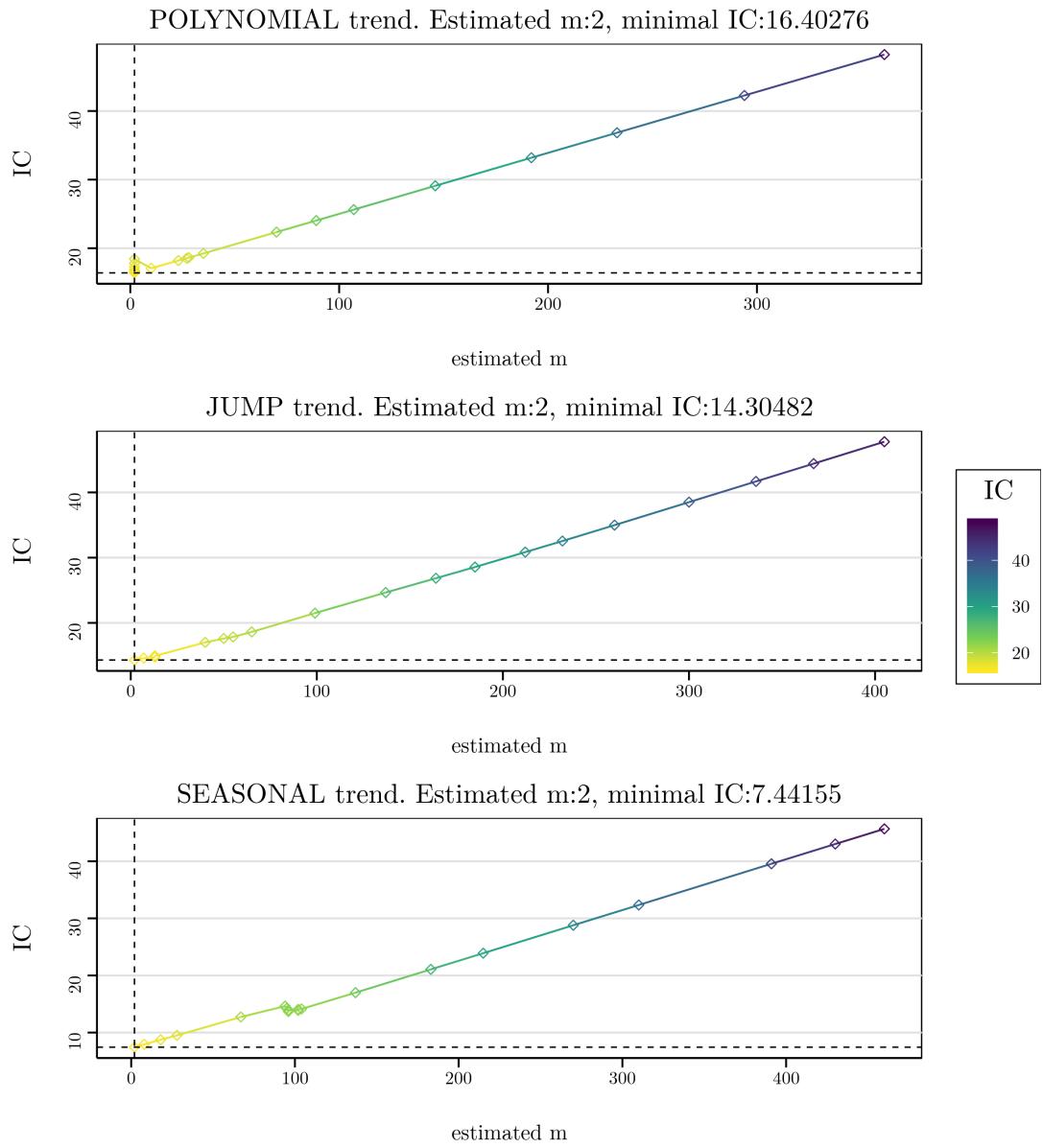


Figure 4: Selected IC for the simulated trends in the linear model. The dashed lines indicate the selected IC and m .

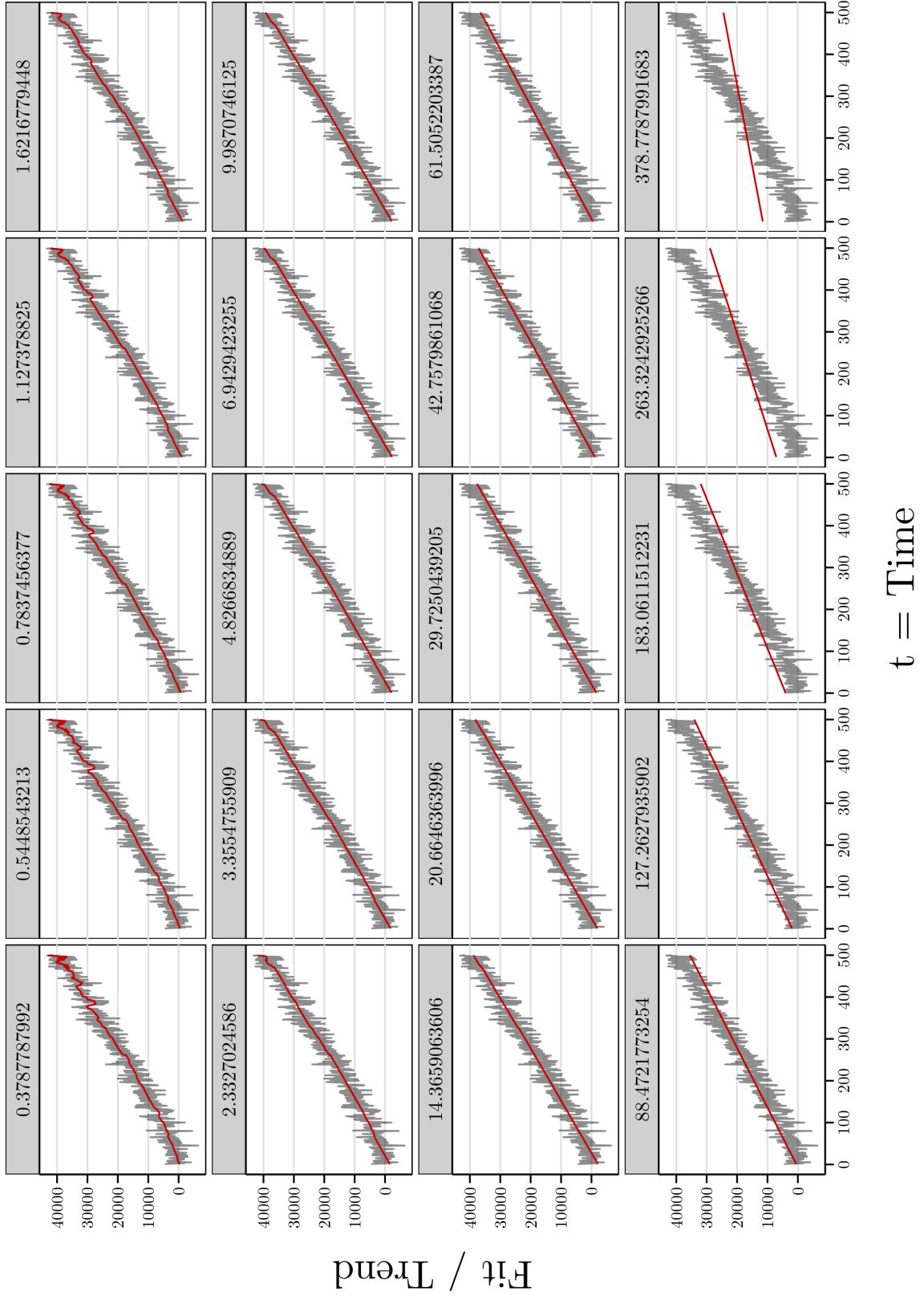


Figure 5: Polynomial time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter λ , using the linear model.

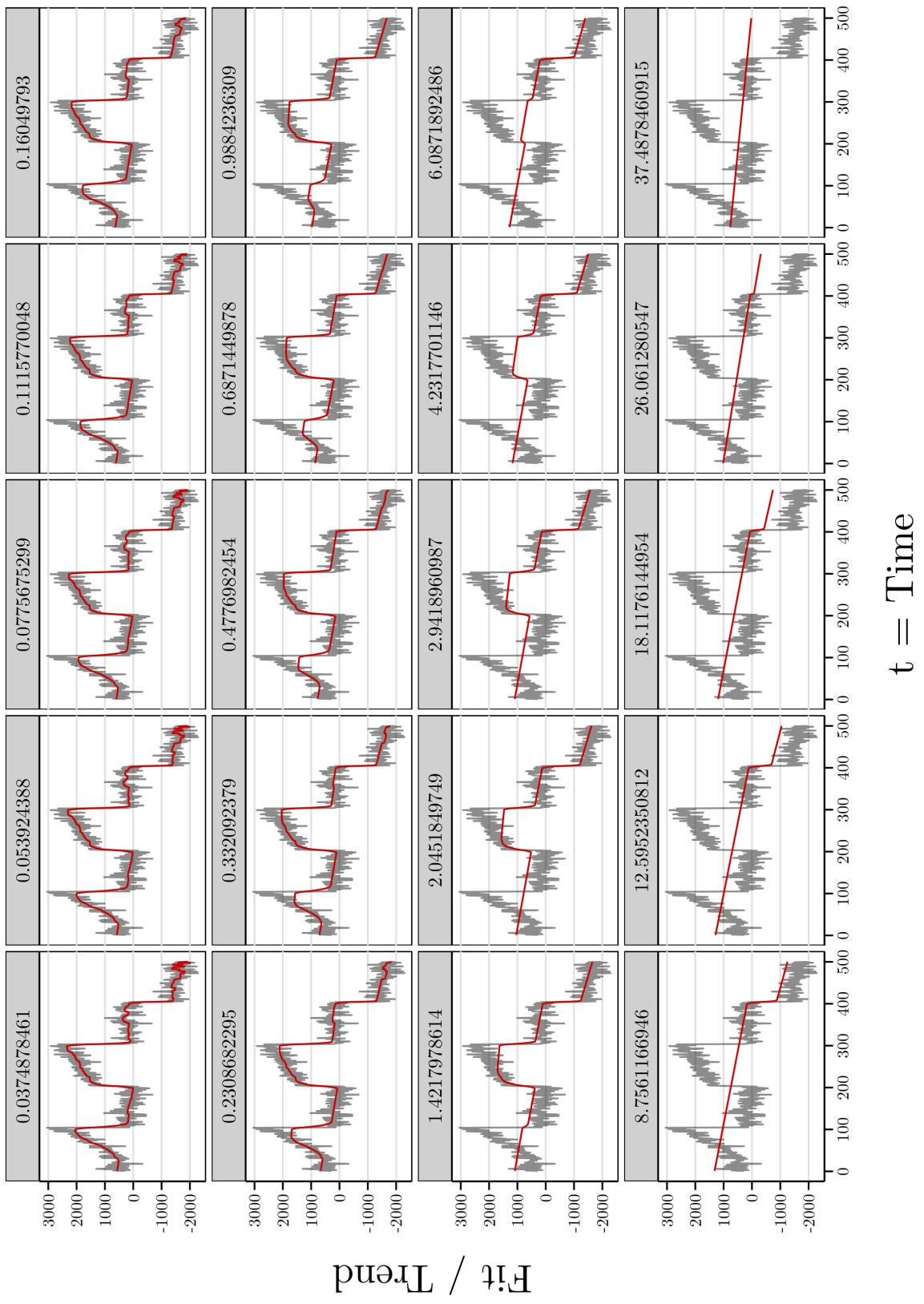


Figure 6: Jump time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter λ , using the linear model.

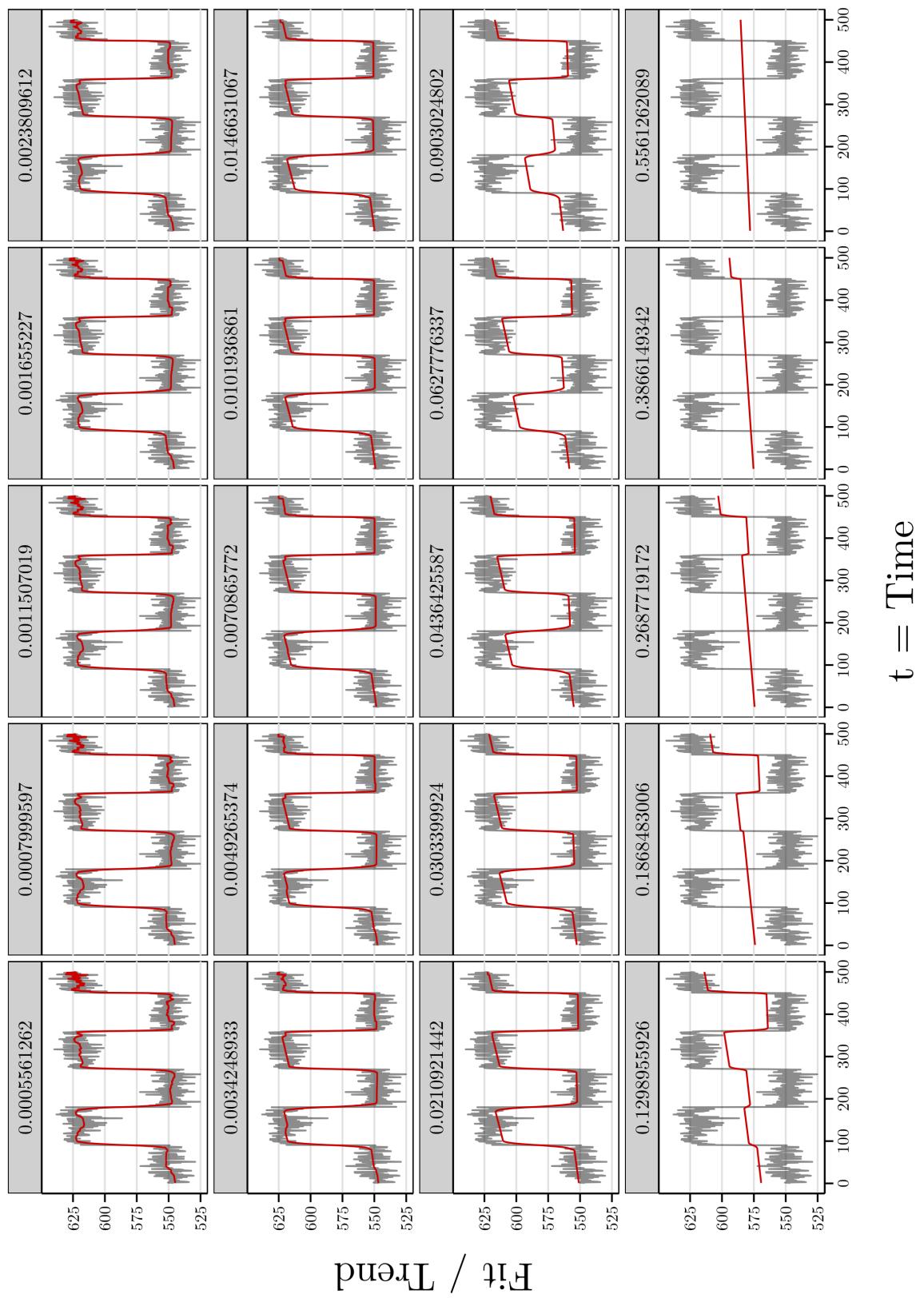


Figure 7: Seasonal time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter λ , using the linear model.

A.1.2 Real Trend Data (P-GFL)

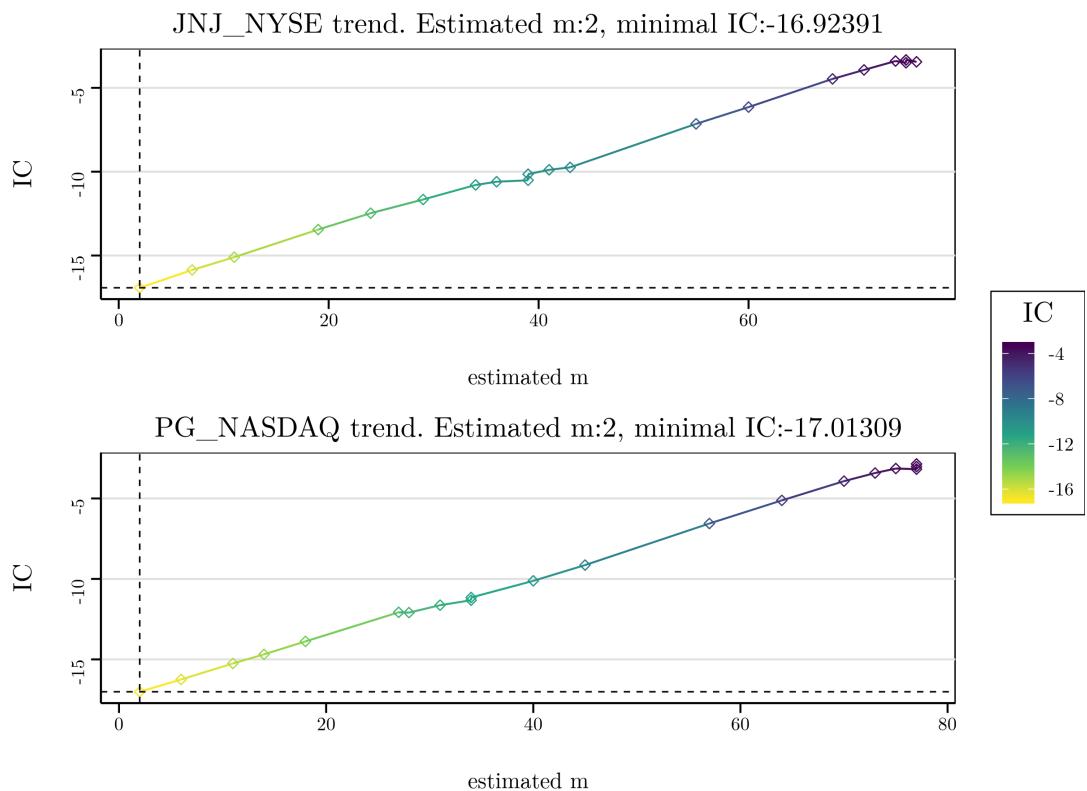
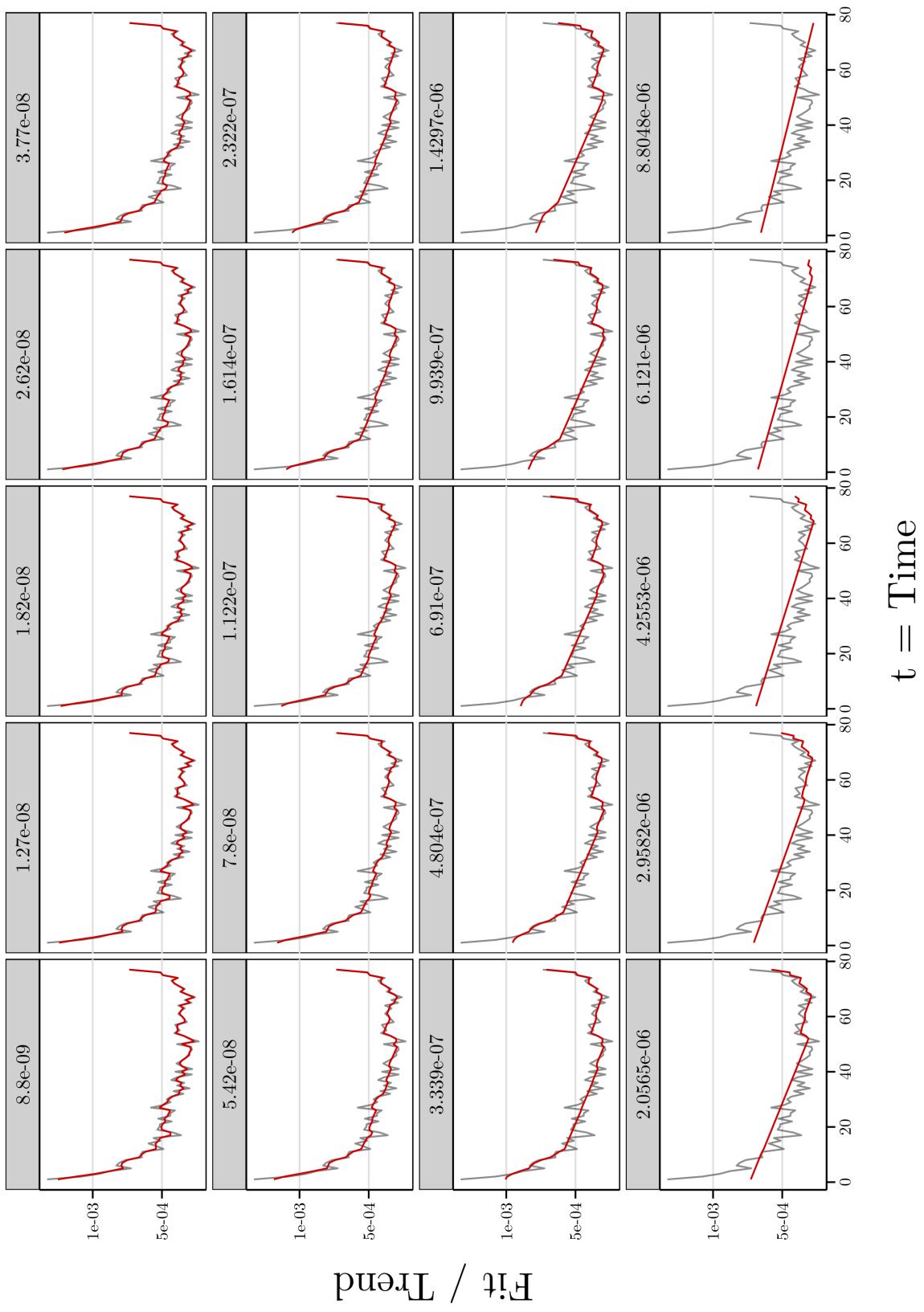


Figure 8: Selected IC for the real trends in the linear model. The dashed lines indicate the selected IC and m .

Figure 9: *JNJ intraday time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter λ , using the linear model.*



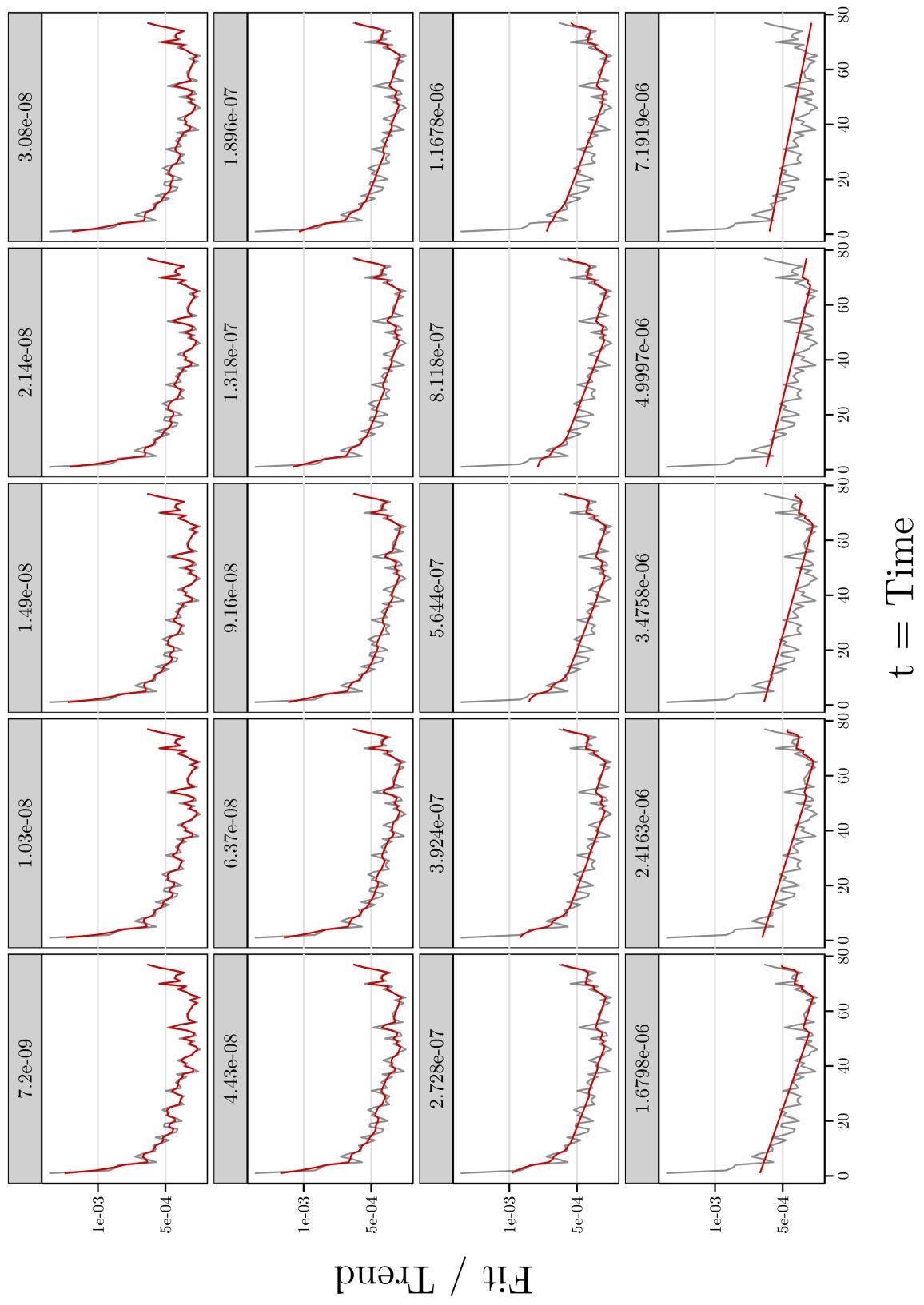


Figure 10: PG intraday time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter λ , using the linear model.

A.2 Quadratic Model (P-GFL)

A.2.1 Simulated Trend Data (P-GFL)

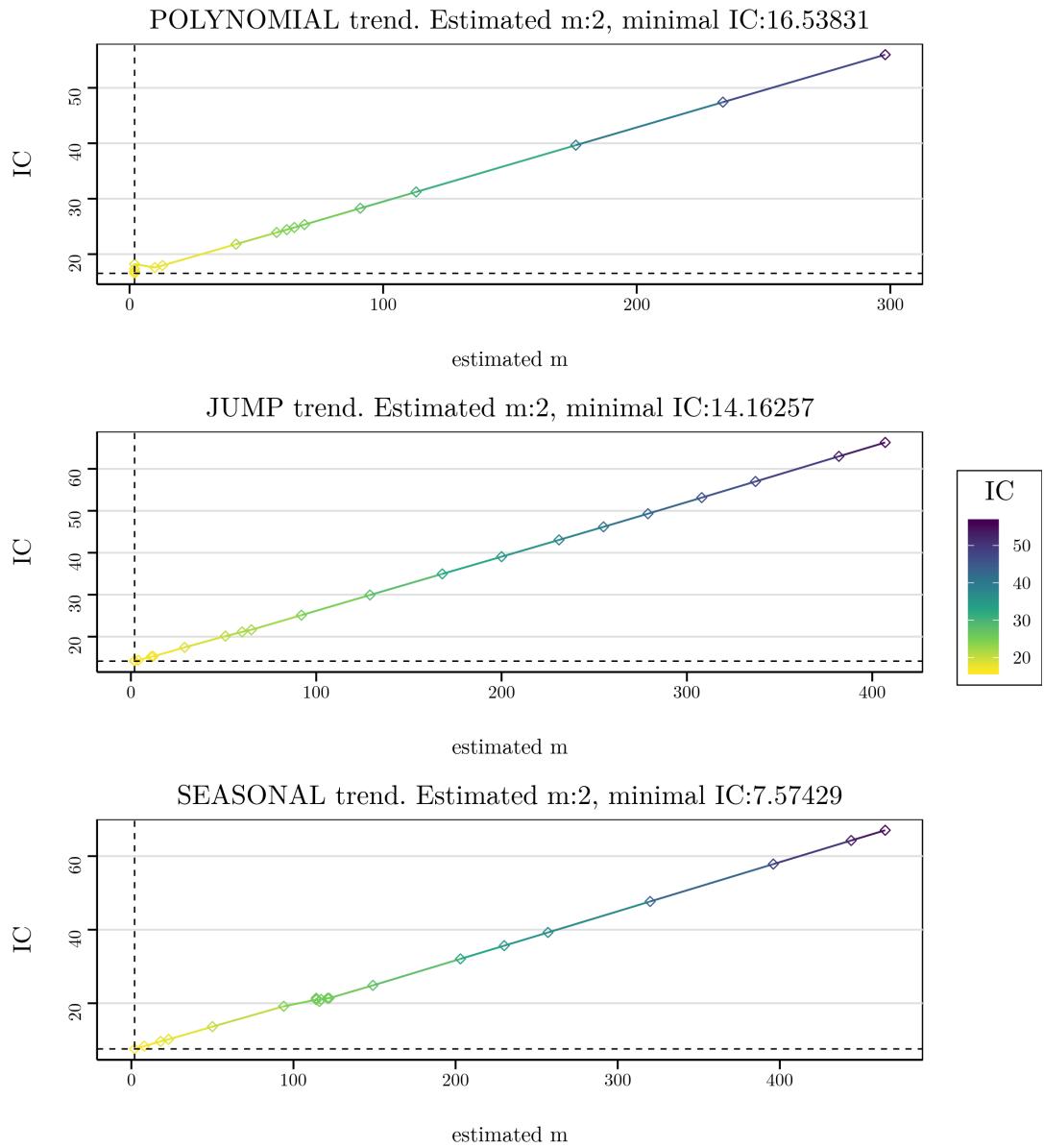


Figure 11: Selected IC for the simulated trends in the quadratic model. The dashed lines indicate the selected IC and m.

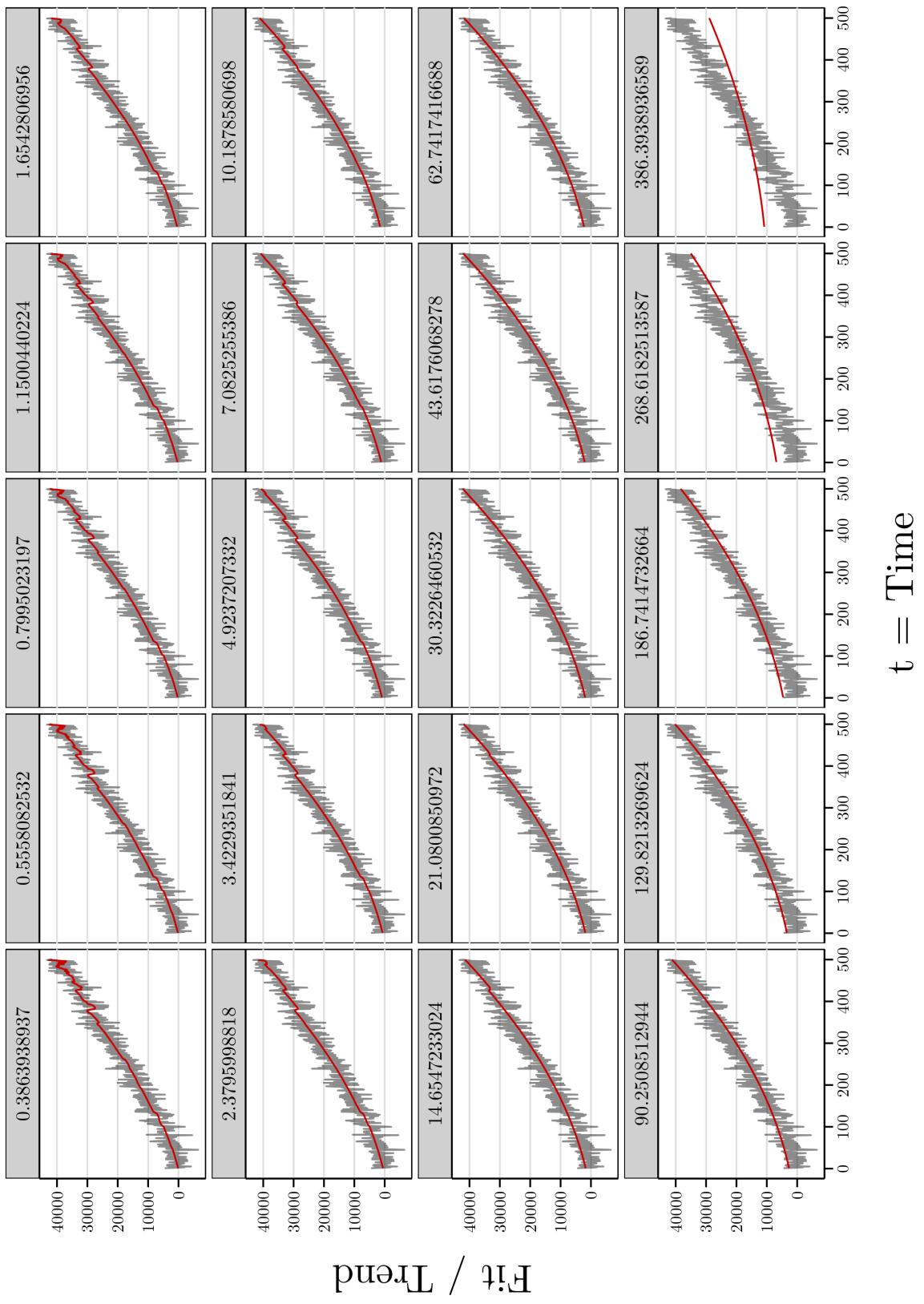


Figure 12: Polynomial time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter λ , using the quadratic model.

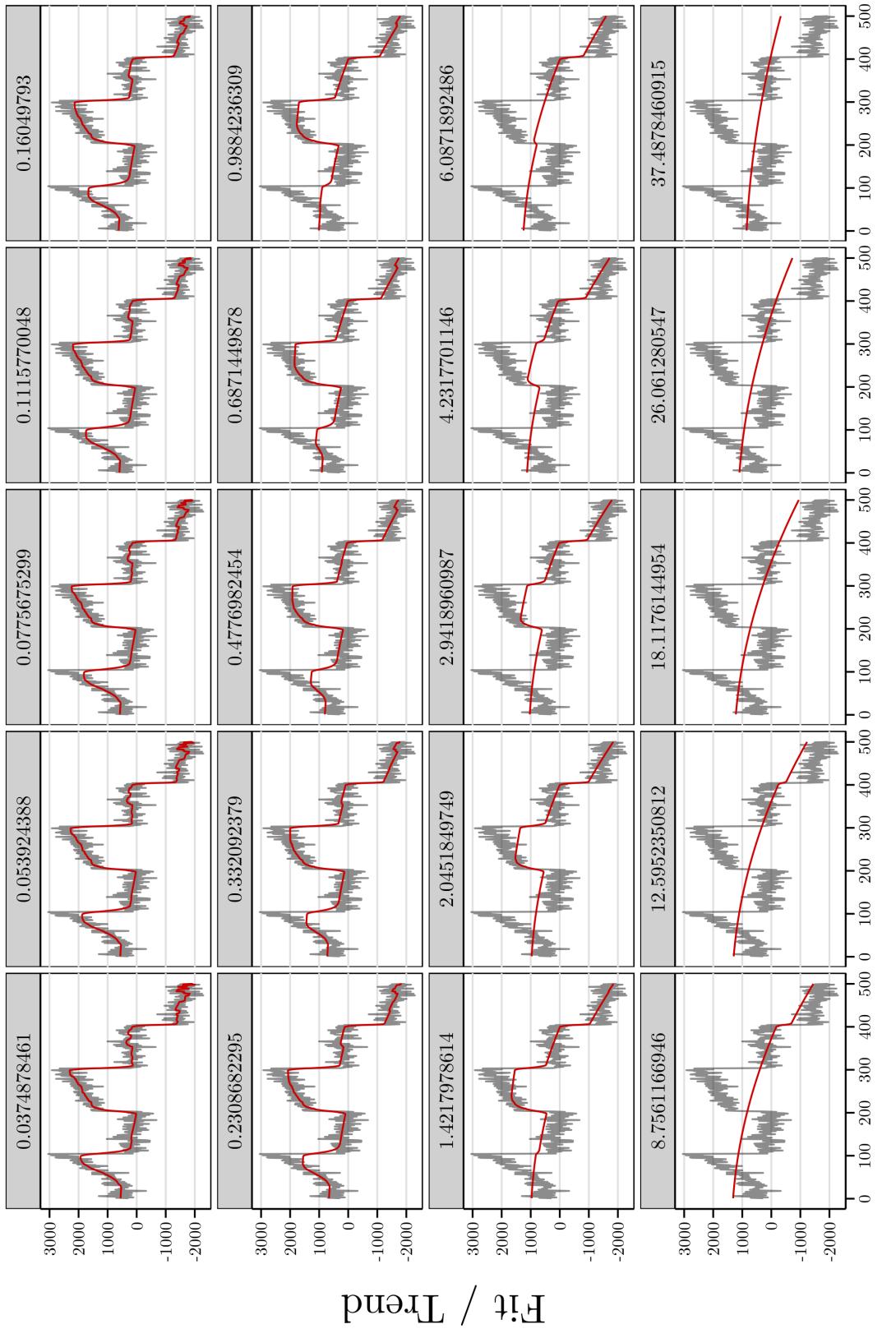


Figure 13: Jump time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter λ , using the quadratic model.

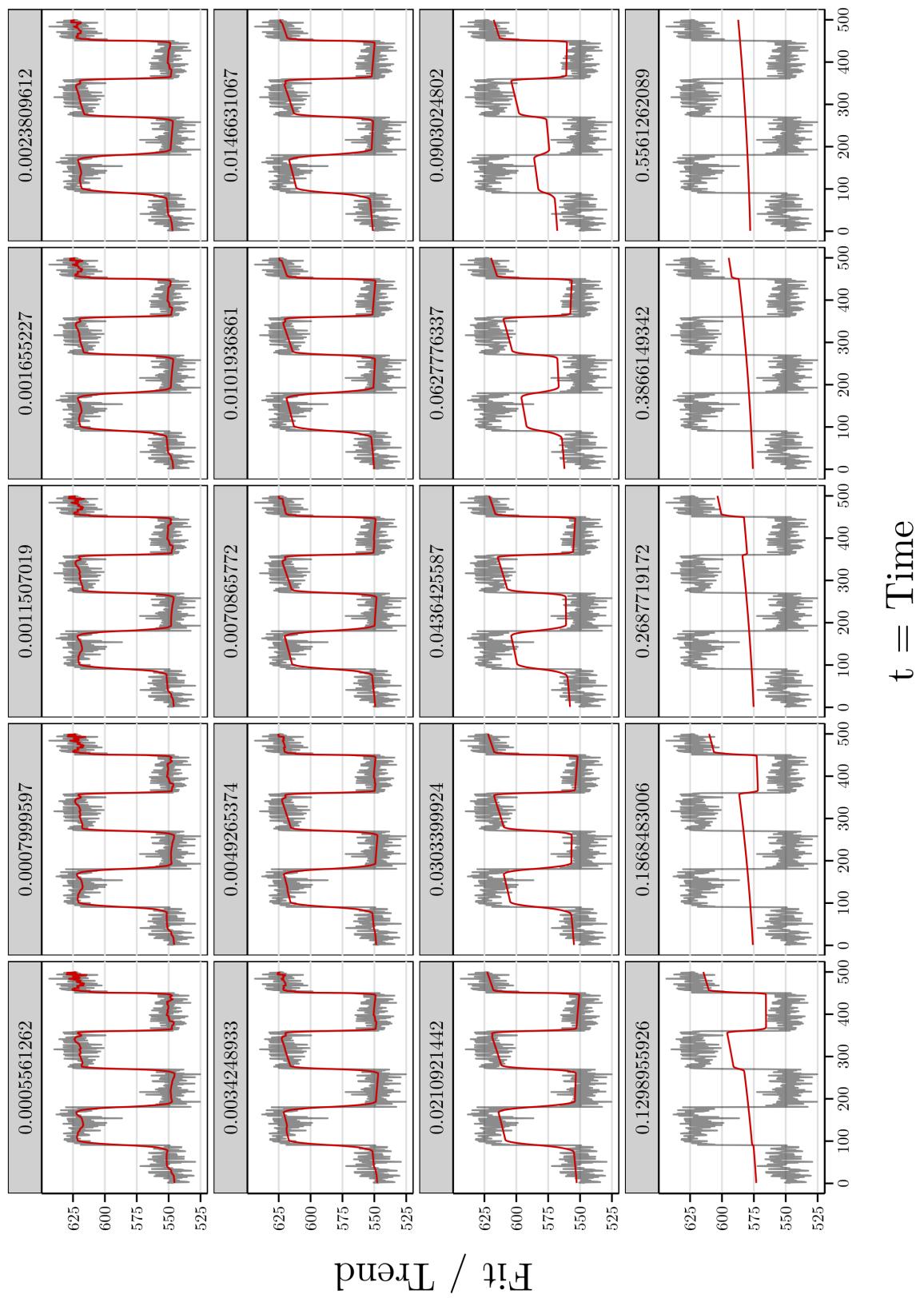


Figure 14: Seasonal time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter λ , using the quadratic model.

A.2.2 Real Trend Data (P-GFL)

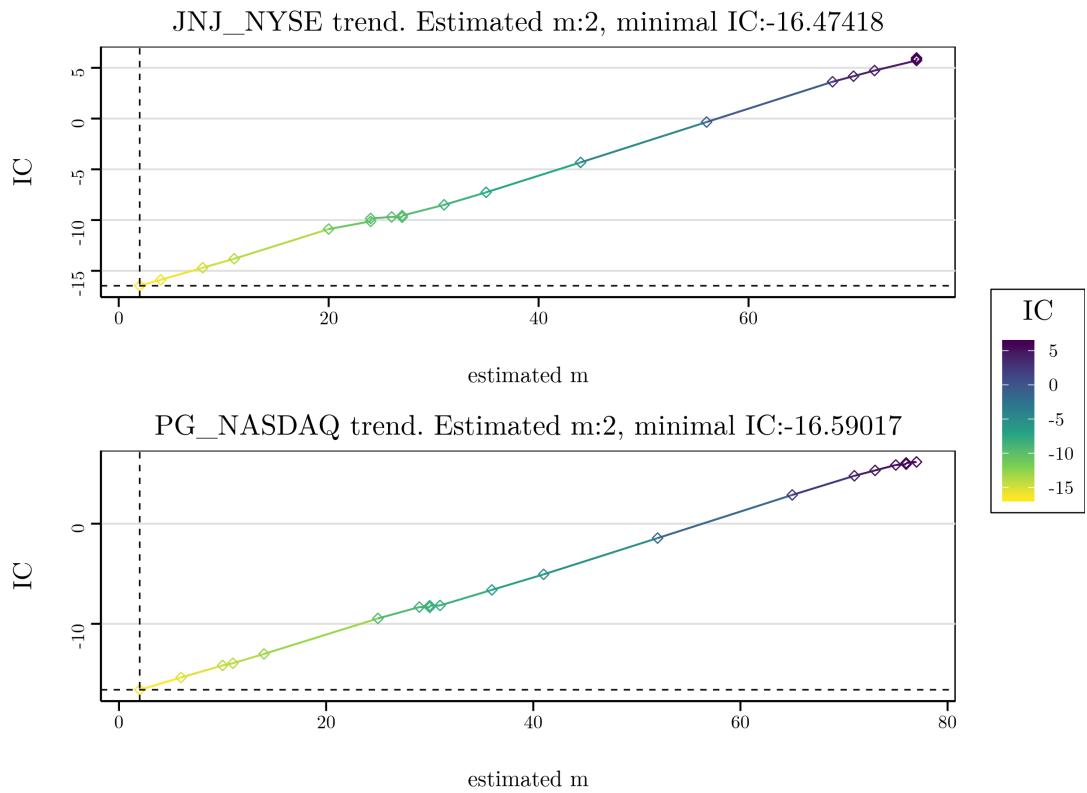


Figure 15: Selected IC for the real trends in the linear model. The dashed lines indicate the selected IC and m .

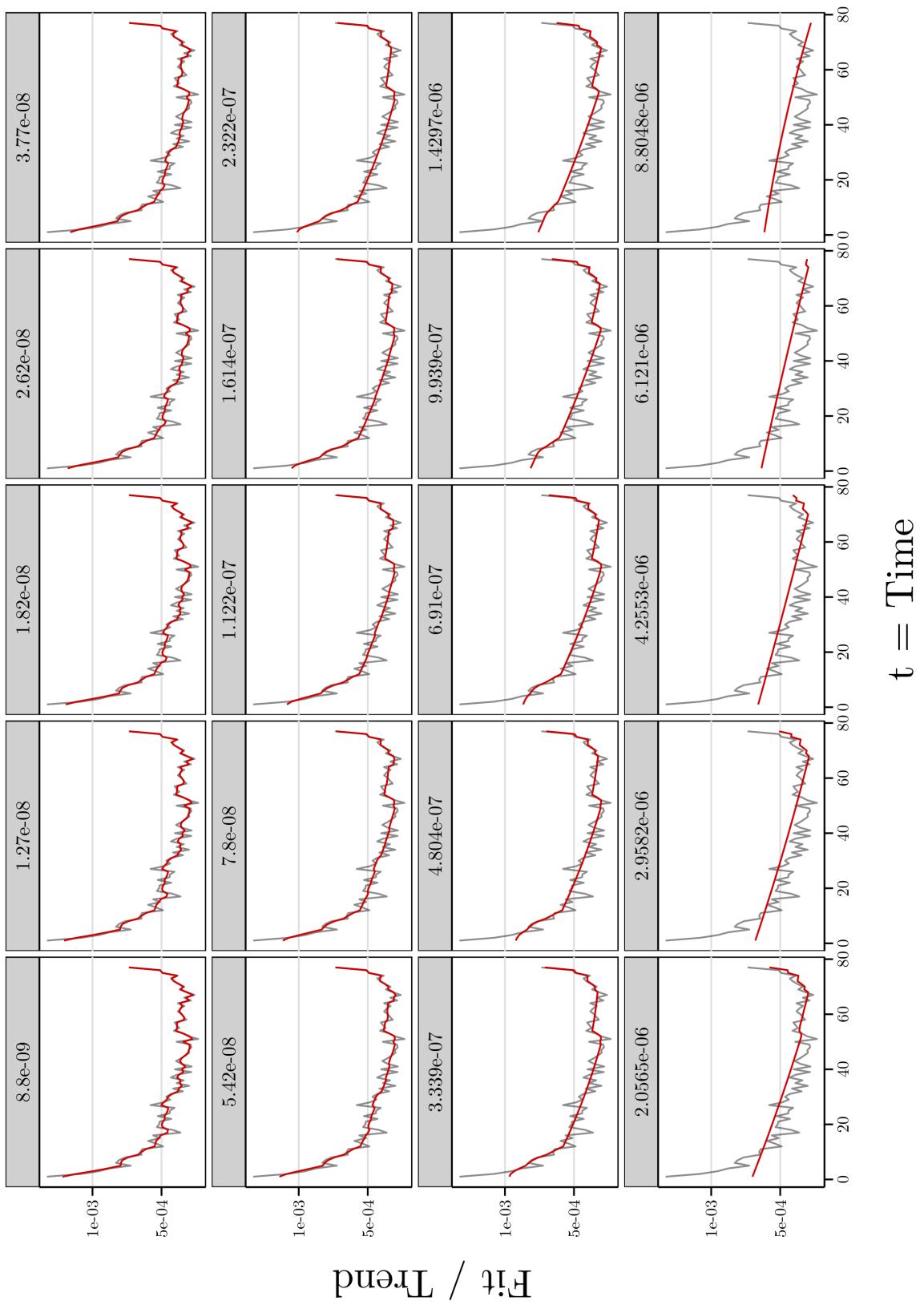


Figure 16: *JNJ intraday time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter lambda, using the quadratic model.*

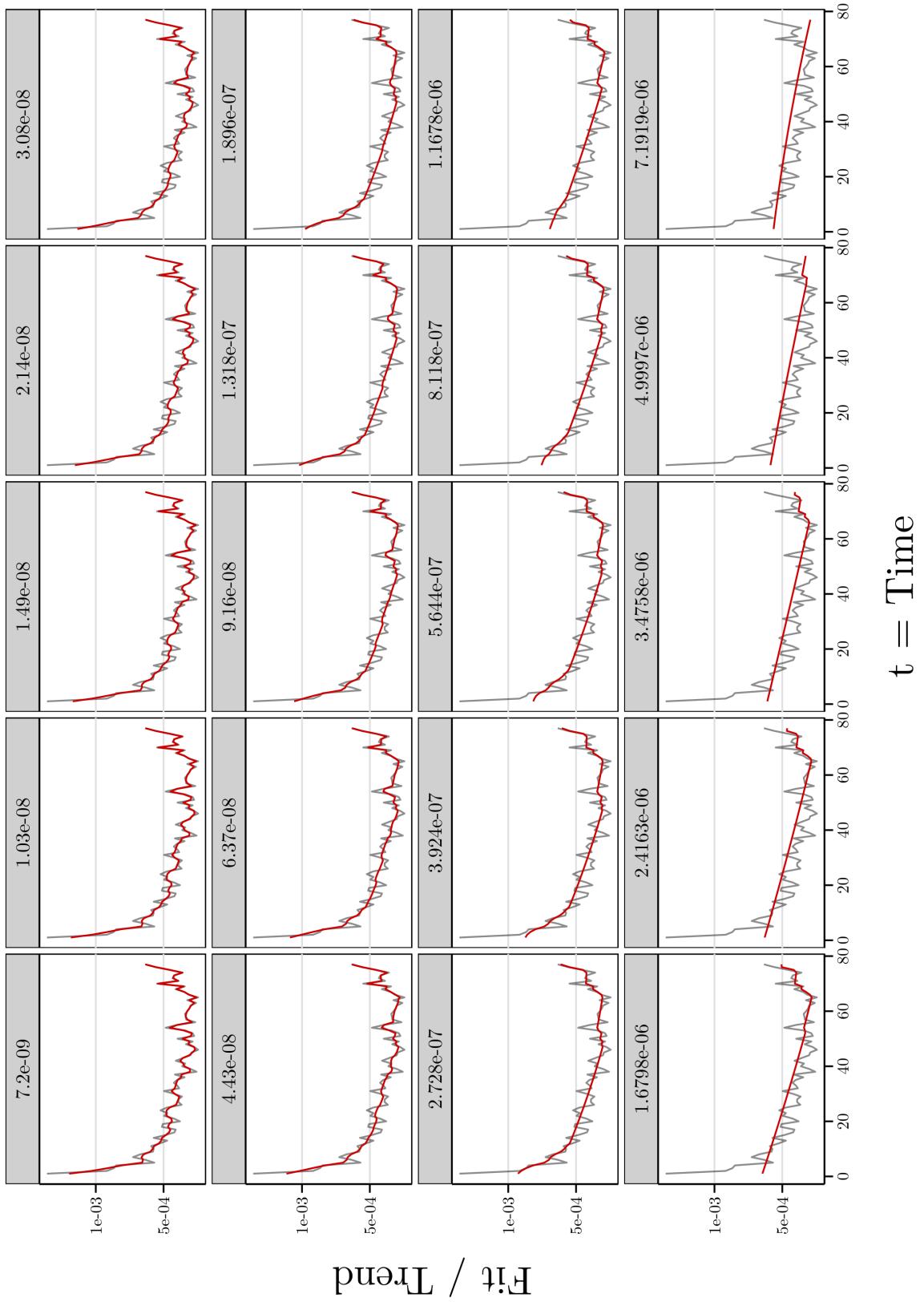


Figure 17: PG intraday time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter λ , using the quadratic model.

A.3 Cubic Model (P-GFL)

A.3.1 Simulated Trend Data (P-GFL)

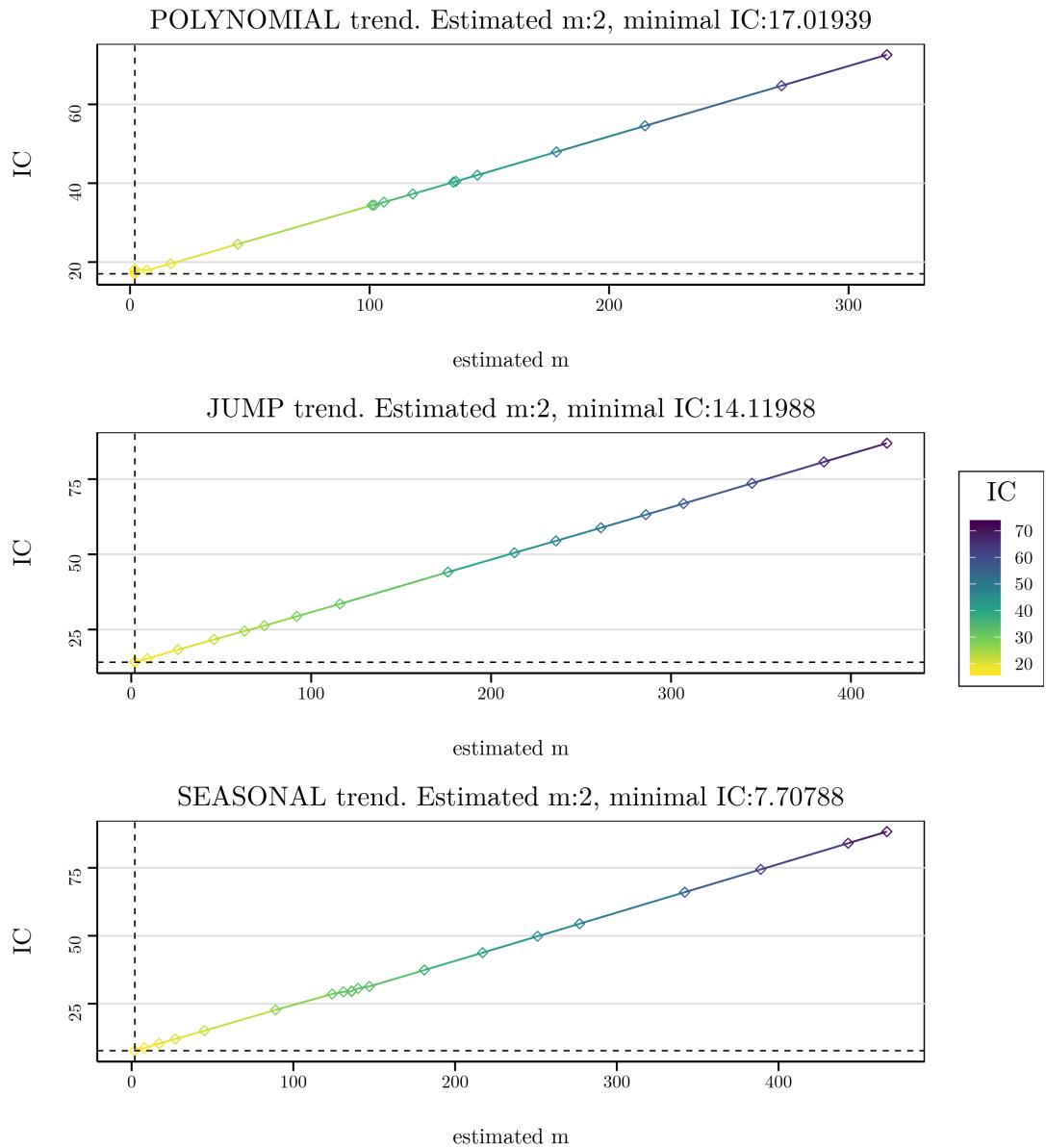


Figure 18: Selected IC for the simulated trends in the cubic model. The dashed lines indicate the selected IC and m .

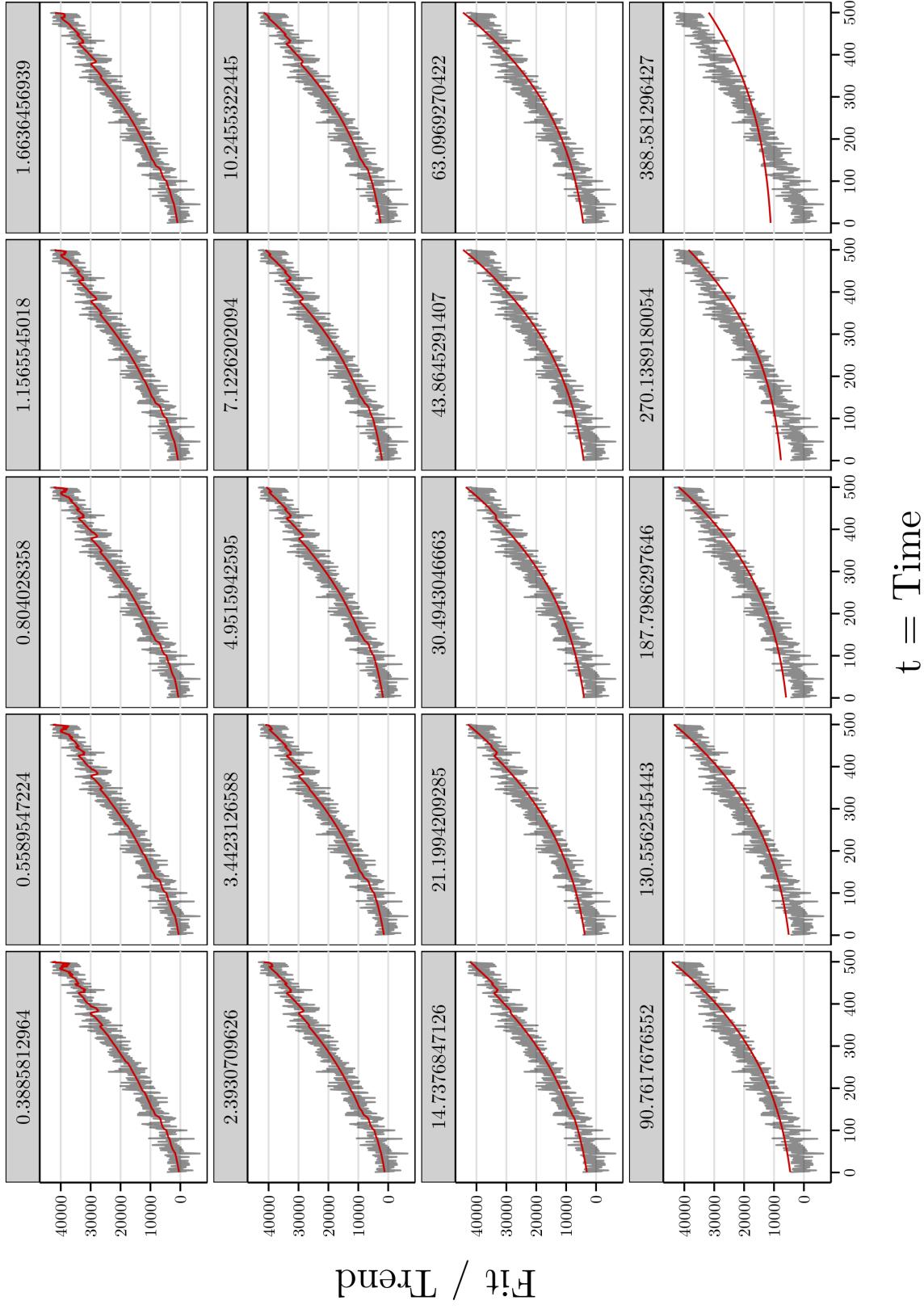


Figure 19: Polynomial time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter λ , using the cubic model.

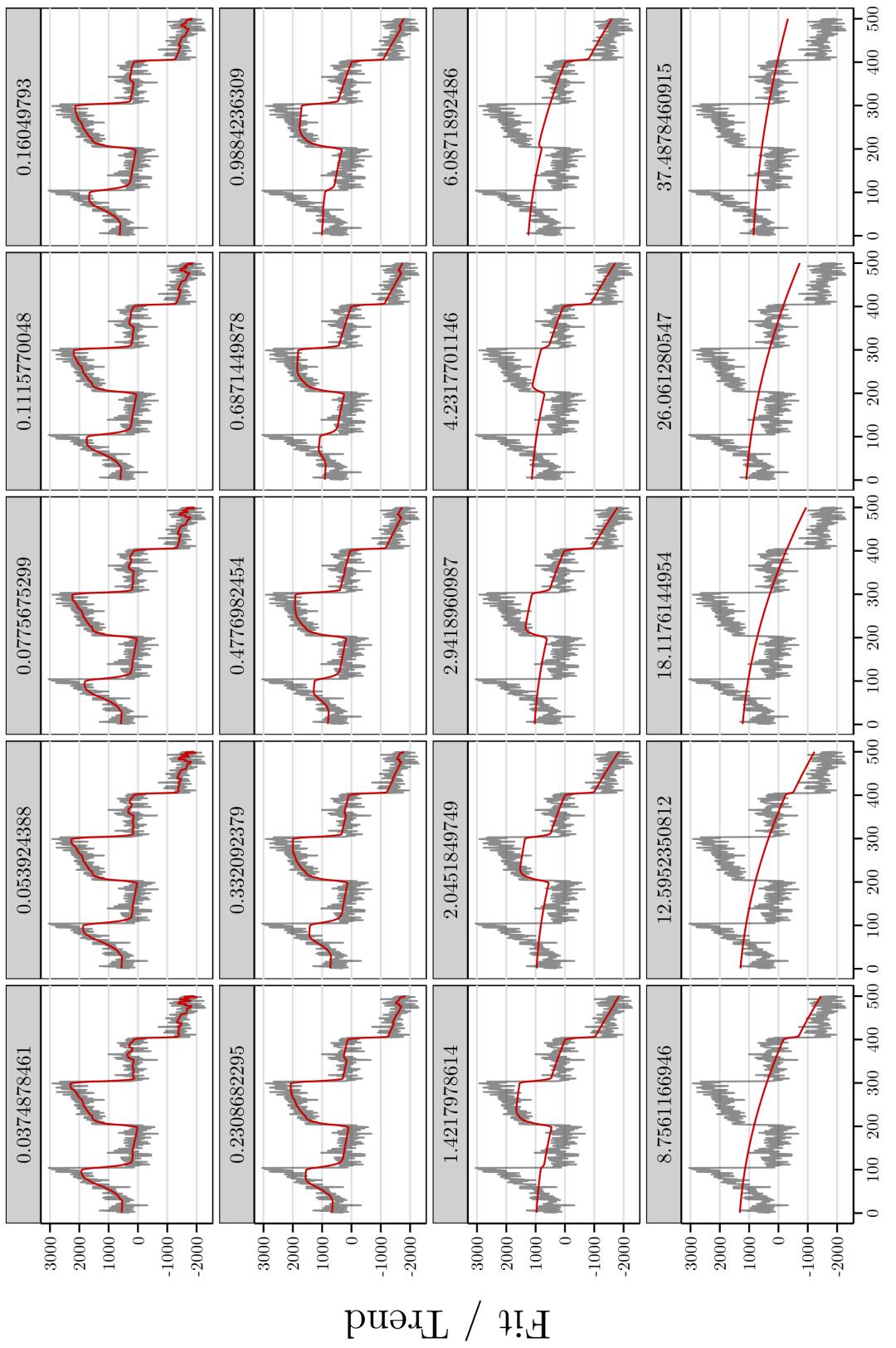


Figure 20: Jump time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter λ , using the cubic model.

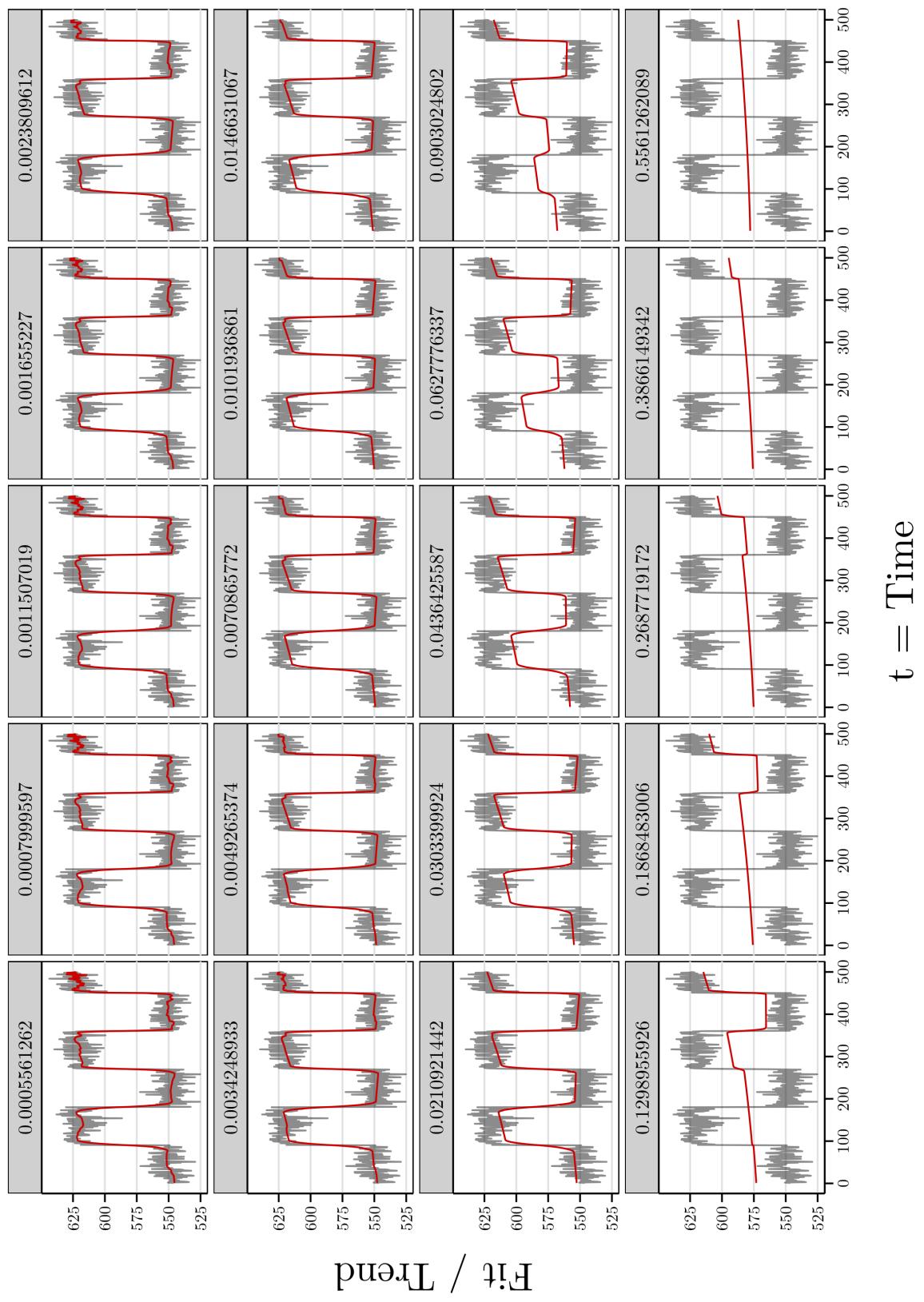


Figure 21: Seasonal time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter λ , using the cubic model.

A.3.2 Real Trend Data (P-GFL)

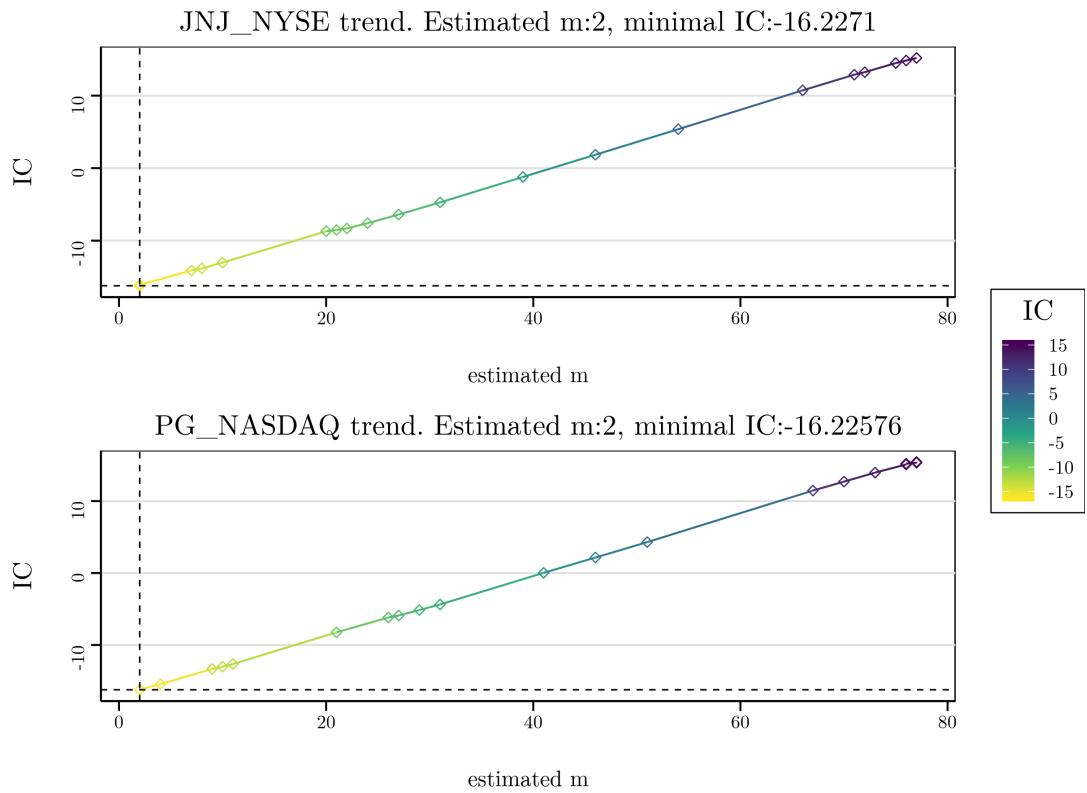


Figure 22: Selected IC for the real trends in the linear model. The dashed lines indicate the selected IC and m .

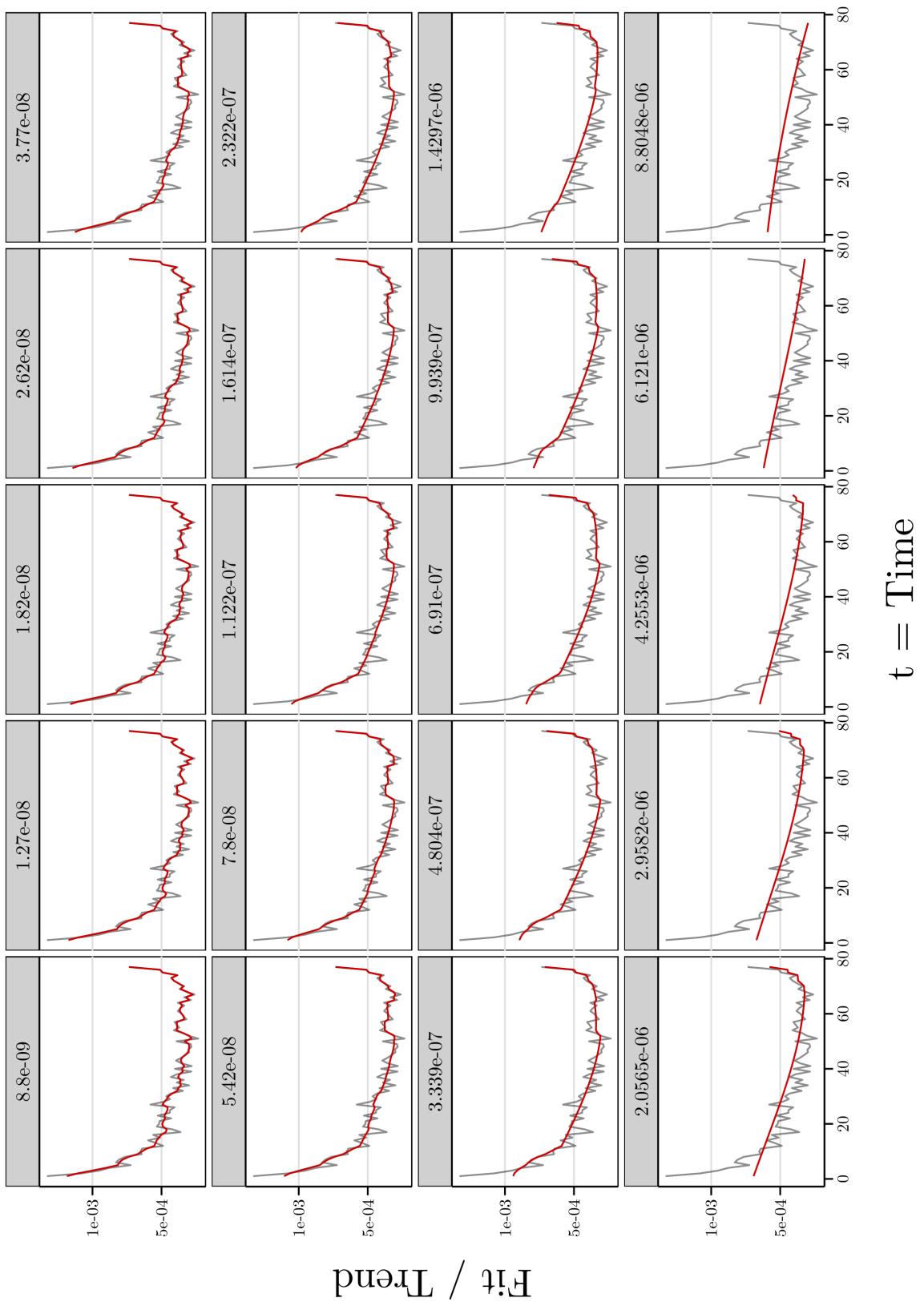


Figure 23: *JNJ intraday time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter lambda, using the cubic model.*

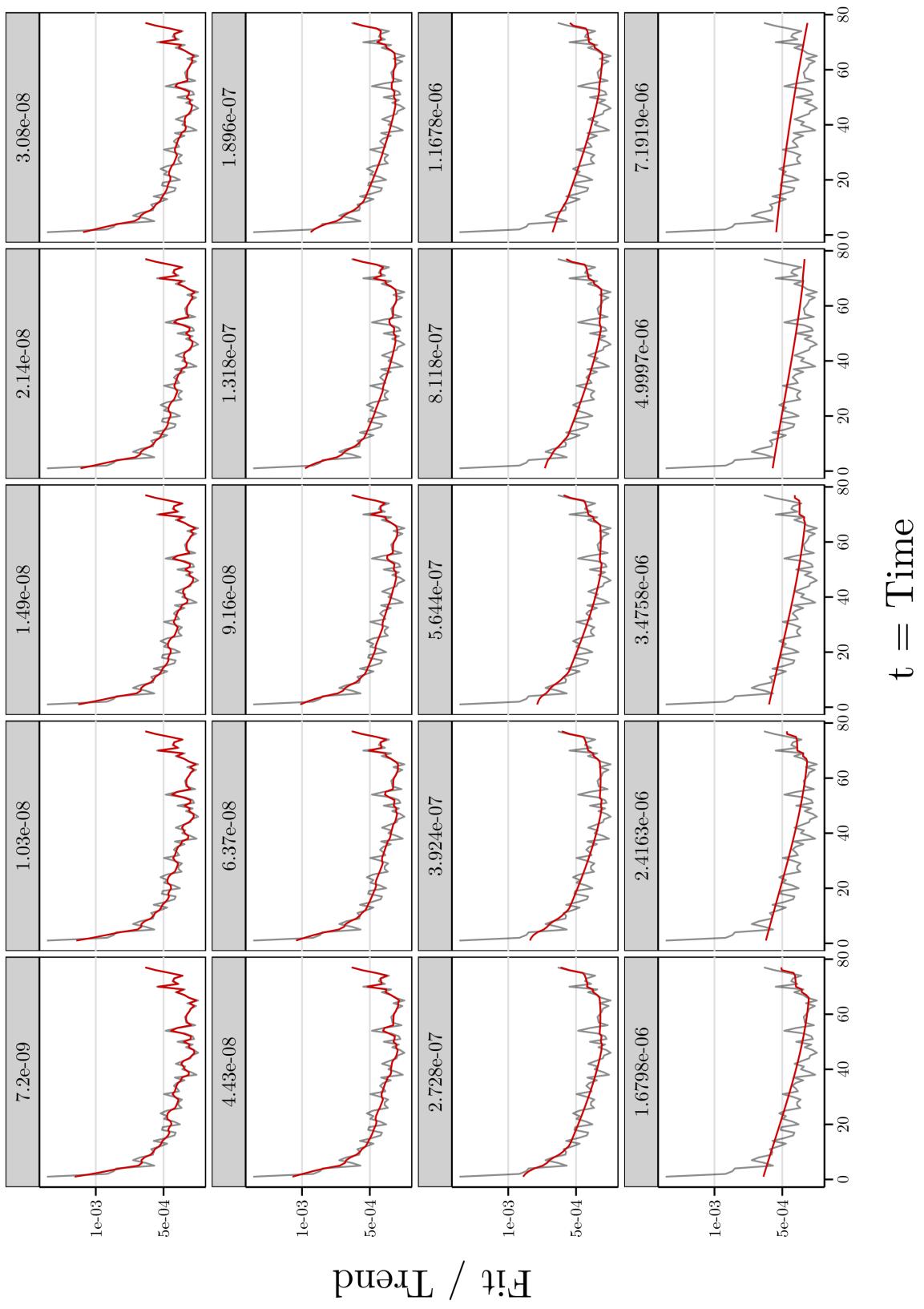


Figure 24: PG intraday time-series trend (grey) and P-GFL-Estimate for all 20 values of the tuning-parameter λ , using the cubic model.

Appendix B P-SGFL Results

B.1 Cubic Model (P-SGFL)

B.1.1 Simulated Trend Data (P-SGFL)

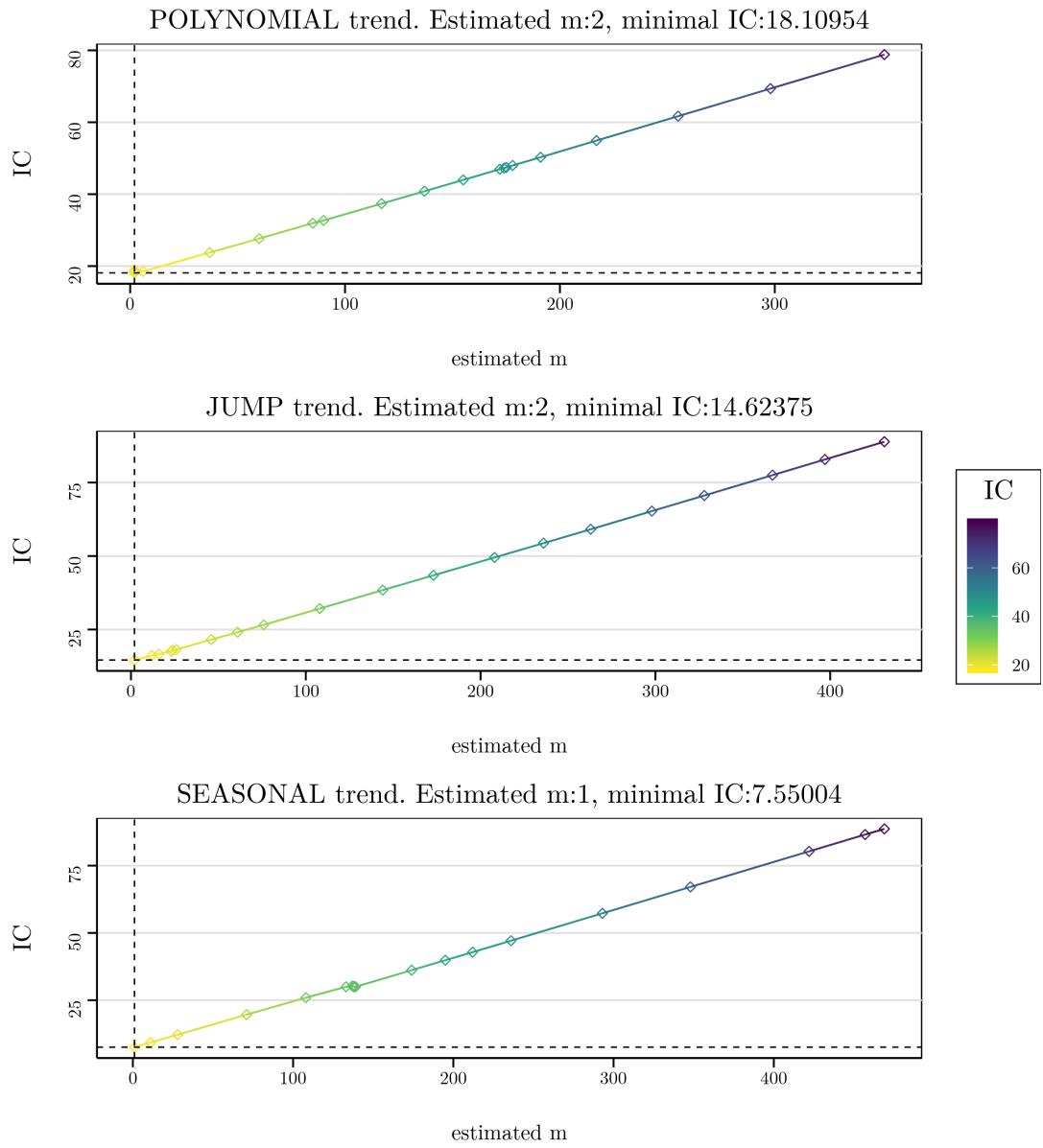


Figure 25: Selected IC for the simulated trends in the sparse cubic model (P-SGFL). The dashed lines indicate the selected IC and m .

B.1.2 Real Trend Data (P-SGFL)

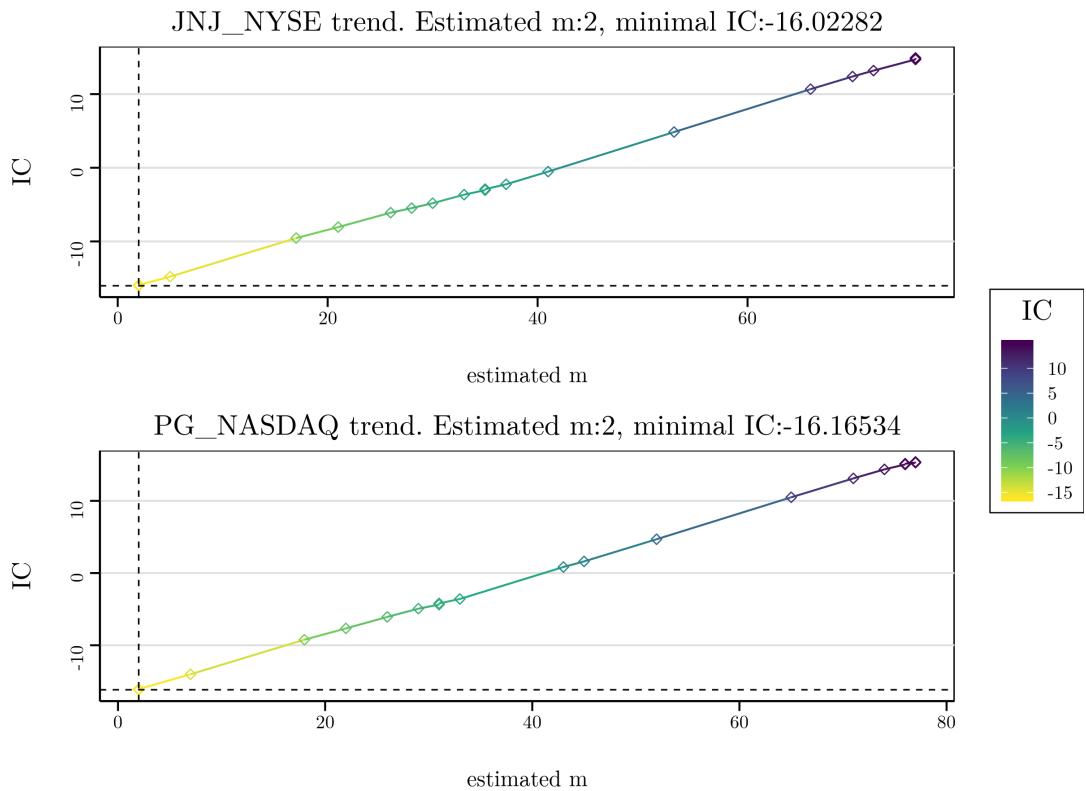


Figure 29: Selected IC for the real trends in the sparse cubic model. The dashed lines indicate the selected IC and m .

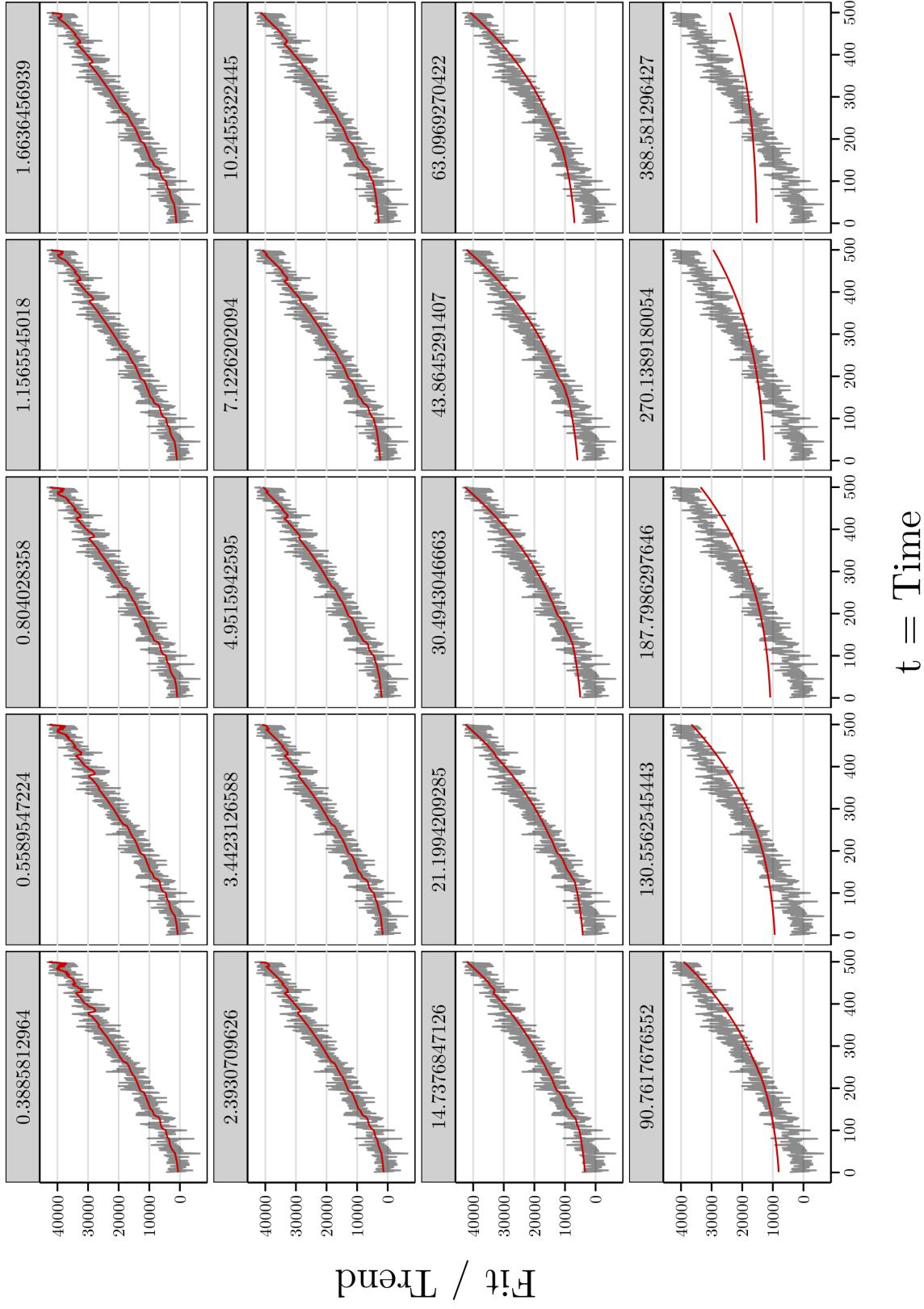


Figure 26: Polynomial time-series trend (grey) and P-SGFL-Estimate for all 20 values of the tuning-parameter λ , using the sparse cubic model.

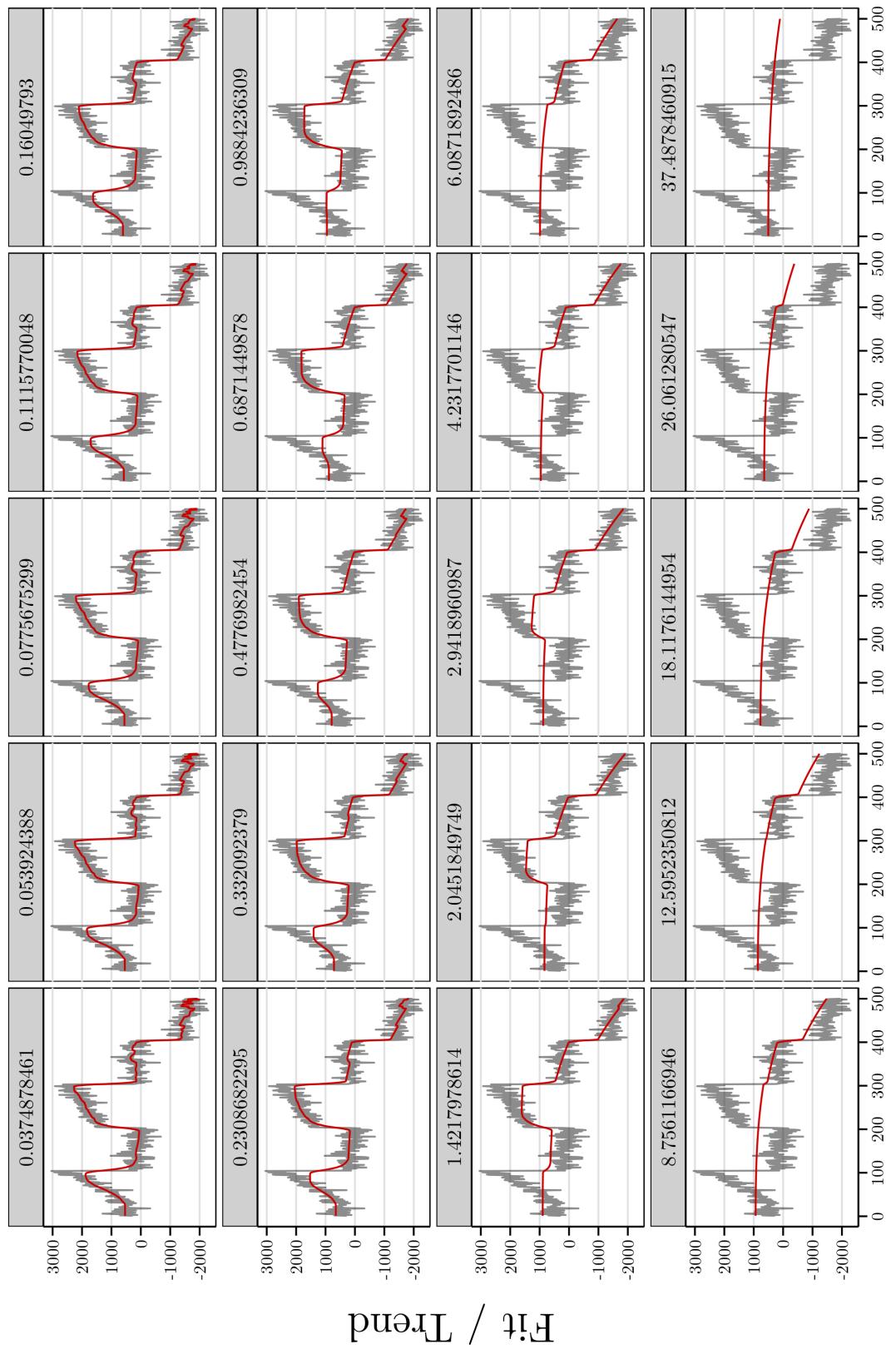


Figure 27: Jump time-series trend (grey) and P-SGFL-Estimate for all 20 values of the tuning-parameter λ , using the sparse cubic model.

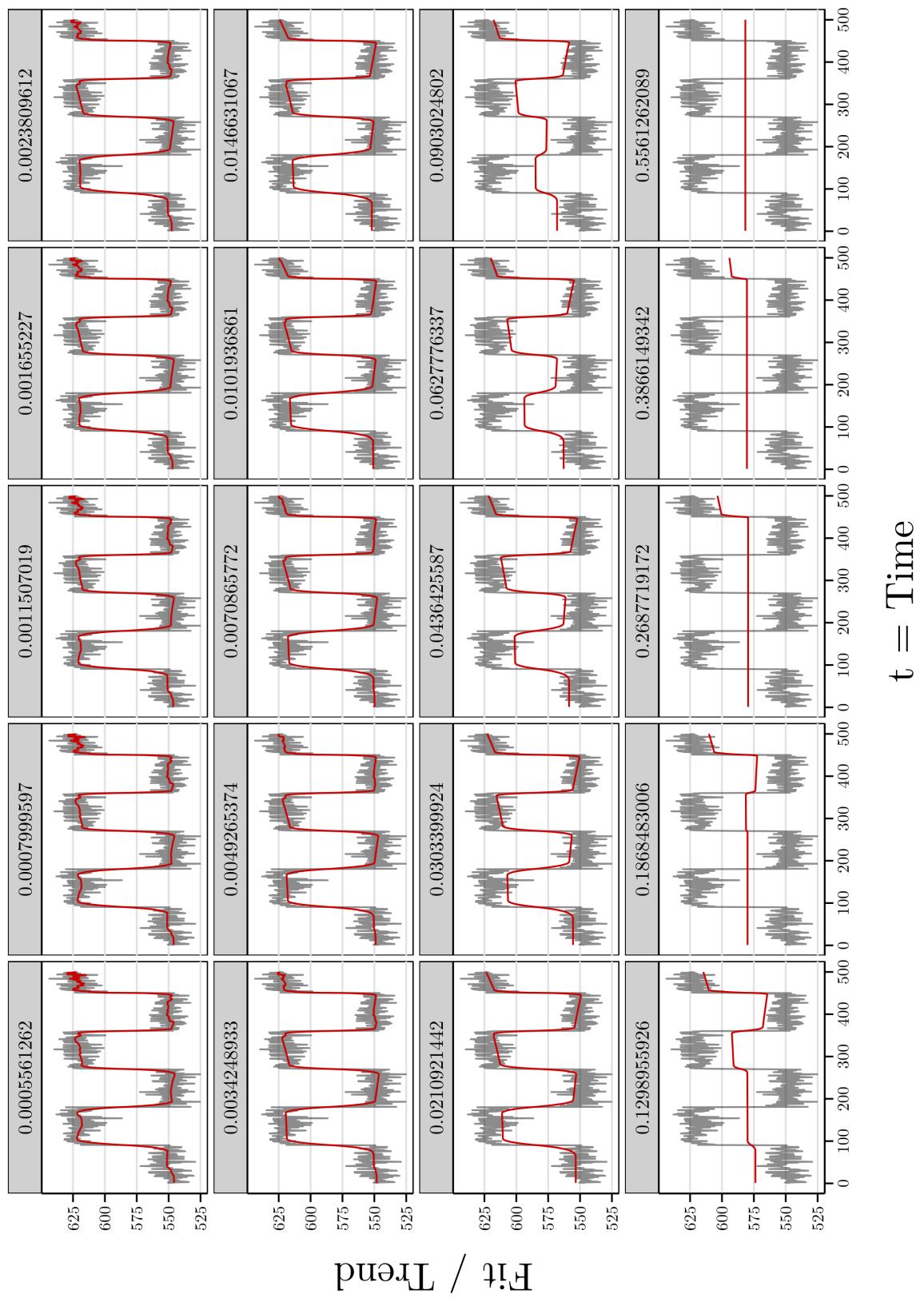


Figure 28: Seasonal time-series trend (grey) and P-SGFL-Estimate for all 20 values of the tuning-parameter λ , using the sparse cubic model.

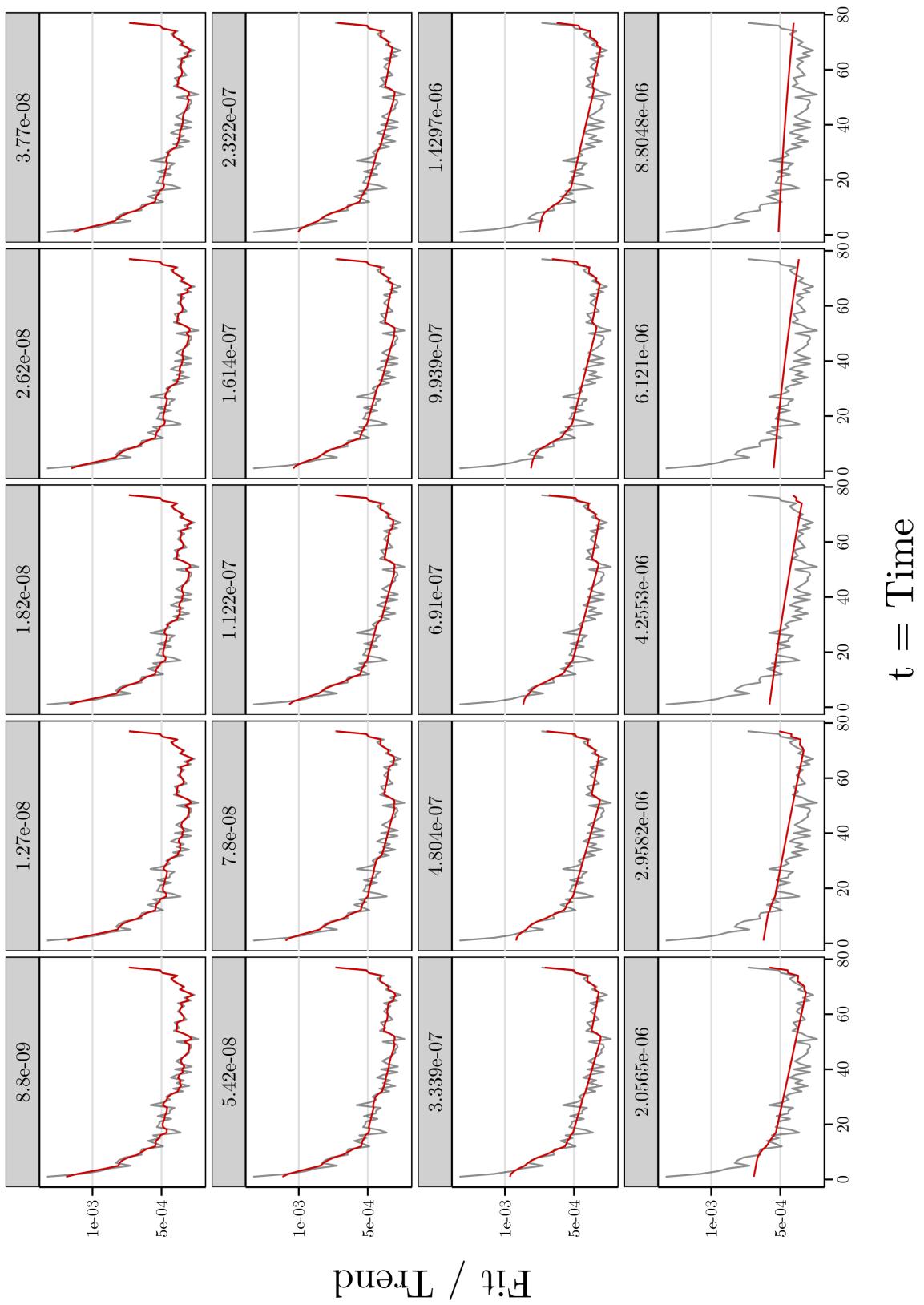


Figure 30: *JNJ intraday time-series trend (grey) and P-SGFL-Estimate for all 20 values of the tuning-parameter lambda, using the sparse cubic model.*

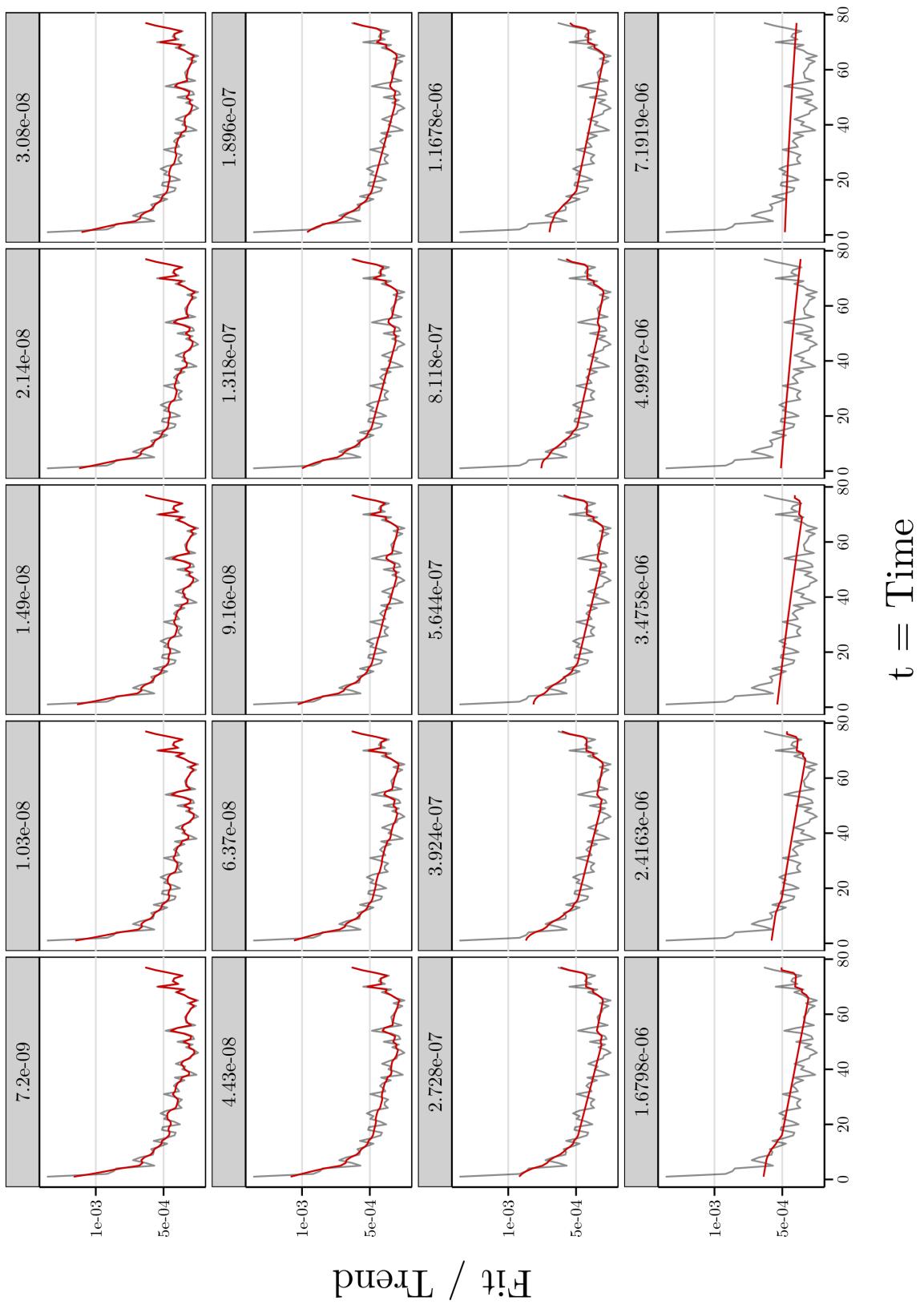


Figure 31: *JNJ intraday time-series trend (grey) and P-SGFL-Estimate for all 20 values of the tuning-parameter lambda, using the sparse cubic model.*

Appendix C Comparison of P-GFL and P-SGFL

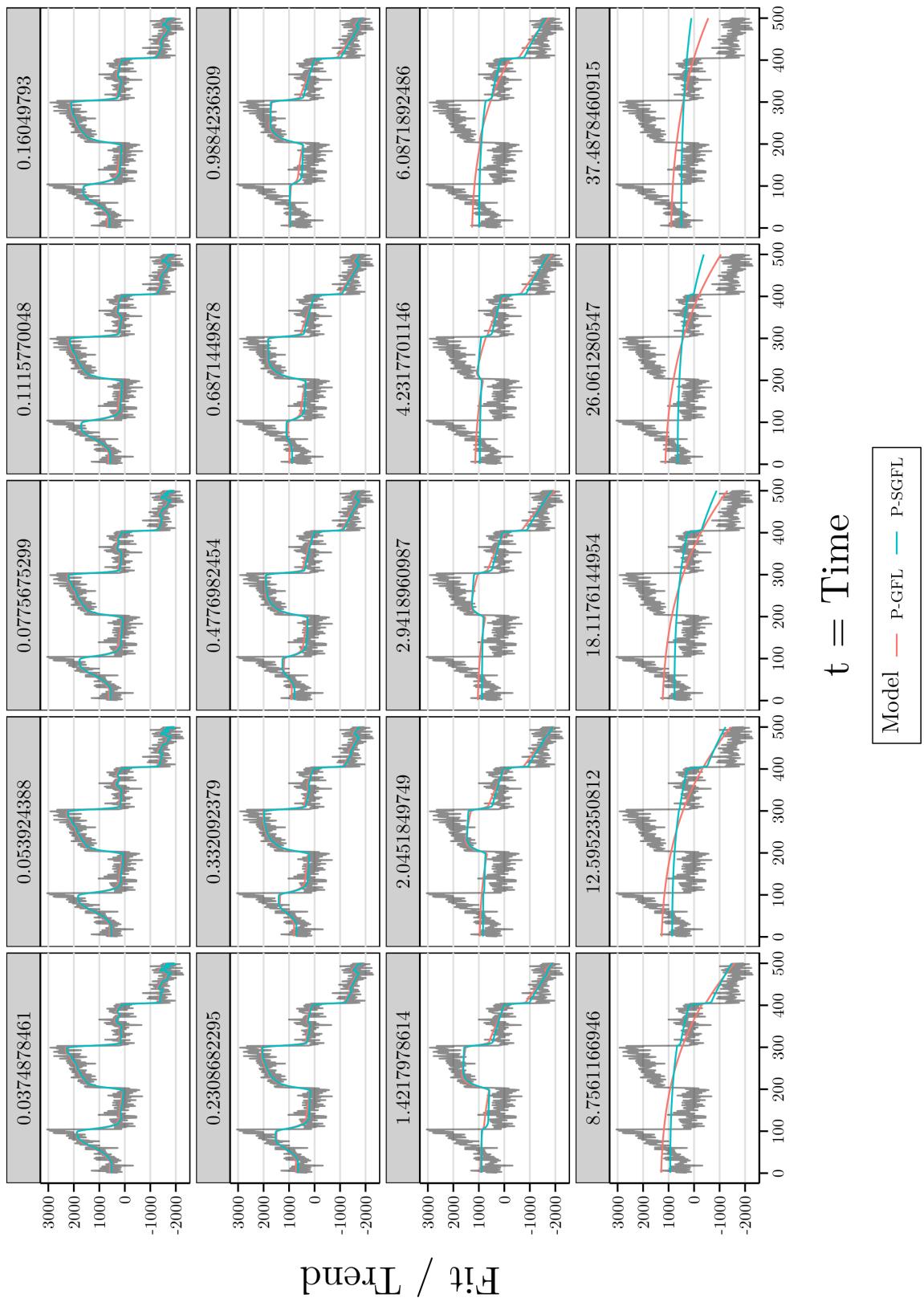


Figure 32: Estimates for the polynomial trend for both models, with a slight offset make both lines visible.