Temporal Hawkes process

Francesco Serafini

02/06/2021

Temporal Hawkes process

We are going to consider a one-dimensional temporal Hawkes process model with intensity given:

$$\lambda(t)|\mathcal{H}_t = \mu + K \sum_{i:t_i<t} h(t - t_i, \boldsymbol{\theta})$$

Where the summation is over all events $t_i \in \mathcal{H}_t$ in the history of the process happend before the evaluation point t. The parameters of the model are μ the background rate, K the productivity parameter which regulates the number of maximum offspring that an event may have, and $\boldsymbol{\theta}$ parameters of the triggering function $h(\cdot, \boldsymbol{\theta})$ which regulates the temporal distribution of the offsprings. We are going to consider $\boldsymbol{\theta} = (c, p)$ such that $c > 0$, and $p > 1$, and

$$h(t - t_i, c, p) = \frac{(p-1)c^{p-1}}{(t - t_i + c)^p}$$

This process at time t can be decomposed in the sum of $n = 1 + |\mathcal{H}_t|$ sub-processes $\lambda_j(t), j = 0, ..., n-1$, where $\lambda_0(t) = \mu$ is an homogeneous poisson process representing the background, and $n - 1$ inhomogeneous poisson processes with intensity

$$\lambda_i(t) = Kh(t - t_i, \boldsymbol{\theta})$$

representing the influence of each observation t_i on the present t.

The integral with respect to $t \in (0, T)$ of the intensity gives us the expected number of points in the interval $(0, T)$, in this case, considering $\mathcal{H}_t = (t_1, ..., t_N)$ such that $t_i < T$, it is given by:

$$\Lambda(T) = \int_0^T \mu + K \sum_{i:t_i<t} \frac{(p-1)c^{p-1}}{(t-t_i+c)^p} dt$$

$$= \mu T + K \sum_{i=1}^{N} (p-1)c^{p-1} \int_{t_i}^{T} (t-t_i+c)^{-p} dt$$

$$= \mu T + K \sum_{i=1}^{N} (p-1)c^{p-1} \left(\frac{(t-t_i+c)^{1-p}}{1-p} \Big|_{t_i}^{T} \right)$$

$$= \mu T + K \sum_{i=1}^{N} c^{p-1} \left(-(t-t_i+c)^{1-p} \Big|_{t_i}^{T} \right)$$

$$= \mu T + K \sum_{i=1}^{N} \left(1 - c^{p-1}(T-t_i+c)^{1-p}\right)$$

To continue the decomposition used before, we can rewrite the intergral as

$$\Lambda(T) = \int_0^T \lambda_0(t) dt + \sum_{i=1}^{N} \int_{t_i}^{T} \lambda_i(t) dt$$

Where $\lambda_0(t) = \mu$ is the homogeneous background process, and $\lambda_i(t) = Kh(t-t_i)$ are the processes triggered by each observation t_i in the history of the process. The number of offspring generated by each of these triggered sub-processes is given by:

$$\begin{aligned}
\Lambda_i &= \int_{t_i}^{\infty} Kh(t-t_i, \boldsymbol{\theta}) dt \\
&= Kc^{p-1} \left(-(t-t_i+c)^{1-p} \Big|_{t_i}^{\infty} \right) \\
&= K
\end{aligned}$$

Where the last equation is valid only if $p > 1$. So, as we said before, the parameter K regulates the expected number of offsprings generated by an event while the parameters $\boldsymbol{\theta} = (c, p)$ regulates the decay in time of in the distribution of these offsprings.

To efficiently sample in $(0, T)$ from this model we can use this decomposition, we first generate the background events and for each of the events in the background we generate the respective offsprings, we keep only the offsprings smaller than T, we repeat the process for the offsprings of the offsprings until we do not have any event generated in $(0, T)$.

To do that, we need, first of all to generate from $\lambda_i(t)$, we can do that efficiently using inverse sampling. In fact, for any $c > 0$ and $p > 1$

$$\int_{t_i}^{\infty} h(t-t_i) dt = 1$$

So $h(t-t_i)$ can be seen as a density, from which we can calculate the probability that a point is between t_i, T,

$$F_i(T) = \int_{t_i}^{T} h(t - t_i)dt = 1 - c^{p-1}(T - t_i + c)^{1-p}$$

A technique to extract n samples from $h(t - t_i)$ is called inverse sampling technique and it follows:

1. sample n elements $u_1, ..., u_n$ from $Unif(0, 1)$
2. transform the samples $t_i = F_i^{-1}(u)$

The trasnformed $t_1, ..., t_n$ represents a sample from $h(t - t_i)$. Here the inverse of the cumulative distribution function is given by:

$$F_i^{-1}(u) = c(1 - u)^{\frac{1}{1-p}} + t_i - c$$

Belowe we show a historgram of the events generated by an observations in $t_i = 1$ and considering $c = 0.01$ and $p = 1.5$. Red line represents the value of $h(t - t_i, c, p)$, calculated at t equal to the midpoints of the histograms's bins.

```
## Loading required package: viridisLite
```

```
## Loading required package: sp
```

```
## Loading required package: Matrix
```

```
## Loading required package: foreach
```

```
## Loading required package: parallel
```

```
## This is INLA_21.02.23 built 2021-03-15 10:11:24 UTC.
##  - See www.r-inla.org/contact-us for how to get help.
##  - To enable PARDISO sparse library; see inla.pardiso()
##  - Save 273.9Mb of storage running 'inla.prune()'
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:inlabru':
##
##     like
```

```
##
## Attaching package: 'metR'
```

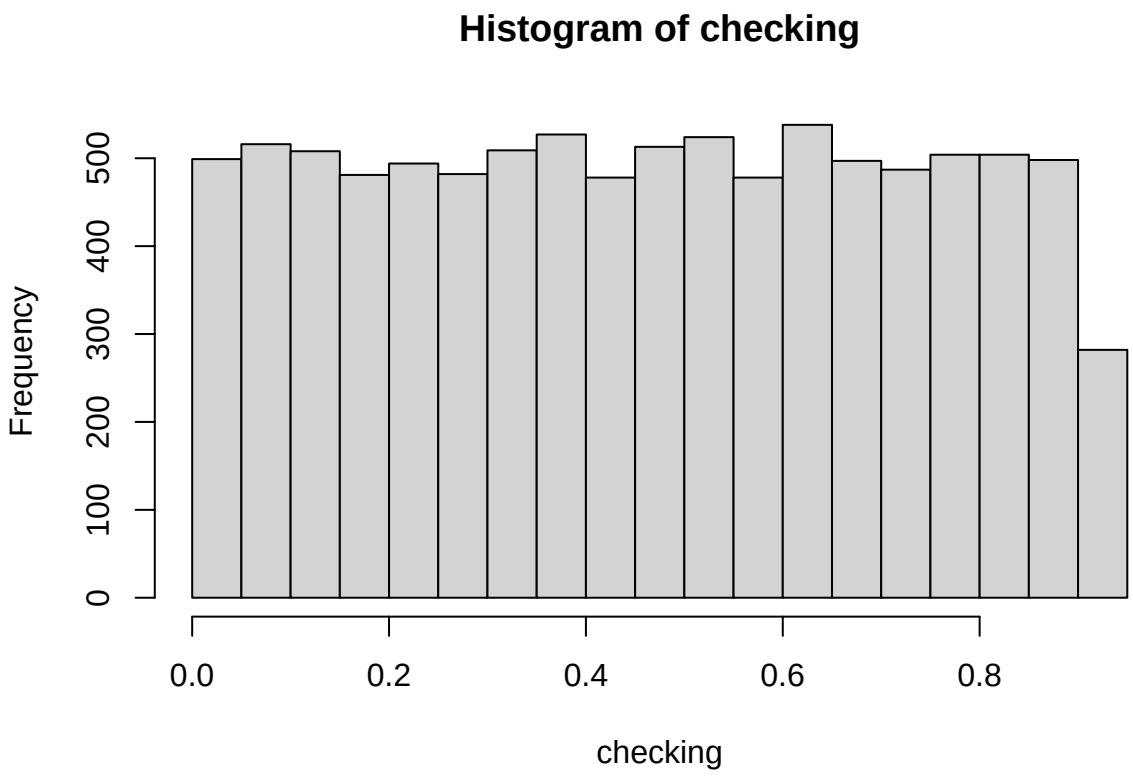
```
## The following object is masked from 'package:INLA':
##
##     f
```

```
##
## Attaching package: 'matrixStats'

## The following object is masked from 'package:dplyr':
##
##     count

## [1] 1.000005 2.995863

## [1] 1 3
```



Having the ability of sampling the observations generated by a generic evet in the process, we can build an algorithm to sample from the process itself. To obtain a sample in $(0, T)$, supposing that there are no events prior to time 0 and that we have no information about events in $(0, T)$, from an Hawkes process with conditional intesity given by:

$$\lambda(t)|\mathcal{H}_t = \mu + K \sum_{i:t_i<t} h(t - t_i, c, p)$$

it is sufficient to:

1. Sample N_μ from a Poisson variable with intesity μ and sample N_μ points uniformly in $(0, T)$. These points will be our background events
2. For each background event t_i,
   - sample N_K from a Poisson variable with intensity K and sample N_K points from $h(t - t_i, c, p)$. Discard all the events after T. The remaining events will be the first generation of offsprings, namely, the events generated by the background events.
3. If no event has been generated at step 2., we stop and the sample is constitute by background events solely. If we have generated events at step 2., we repeat step 2. but with the first generation of offsprings in place of the background events. The resulting events will be the second generation of offsprings.

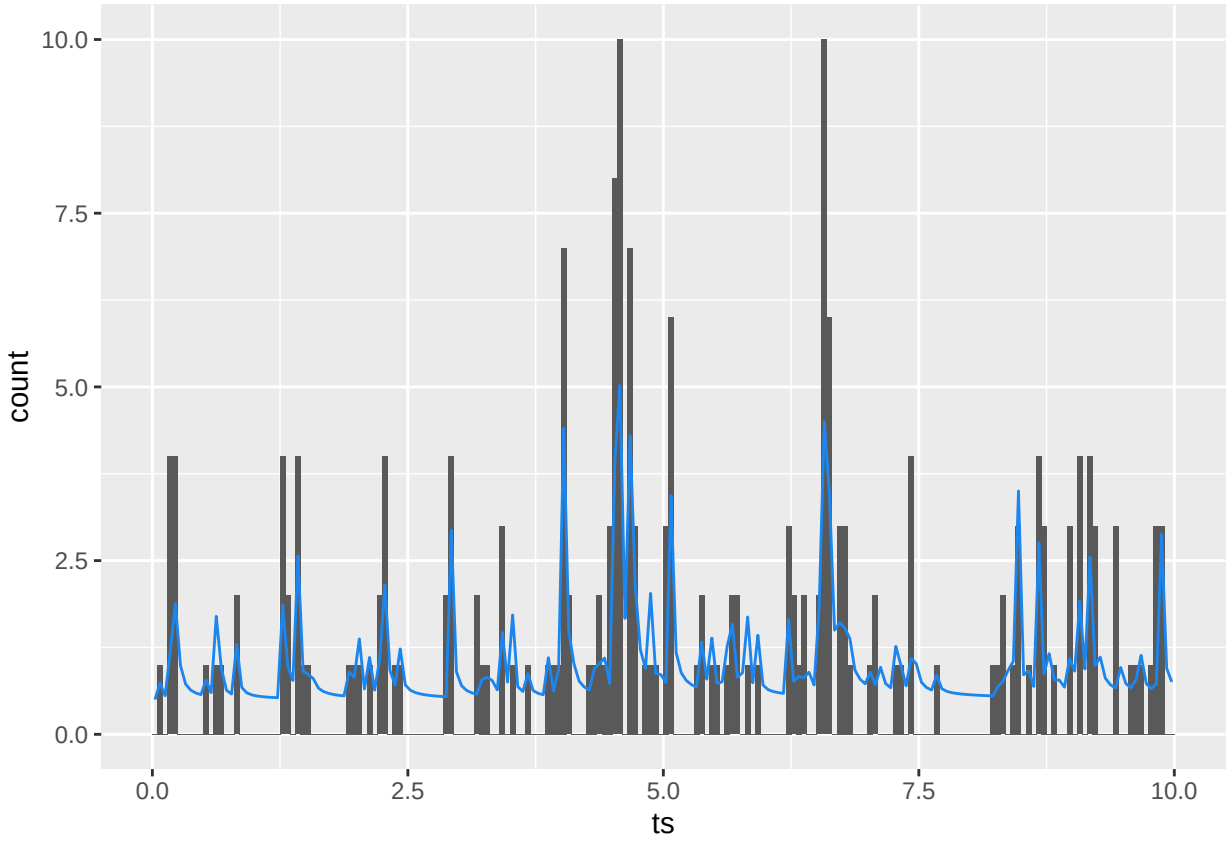
4. Repeat as long as you have a non-empty generation.

Here an example using $\mu = 10, K = 0.9, c = 0.01, p = 1.5$ in the interval $(0, 10)$. First plot shows the histogram of the observations and the intensity of the process, calculated at the midpoints of the histogram's bins, and multiplied by the bin's width. Second plot, for each time t, shows the cumulative sum up to time t of the quantities shown in the first plot.

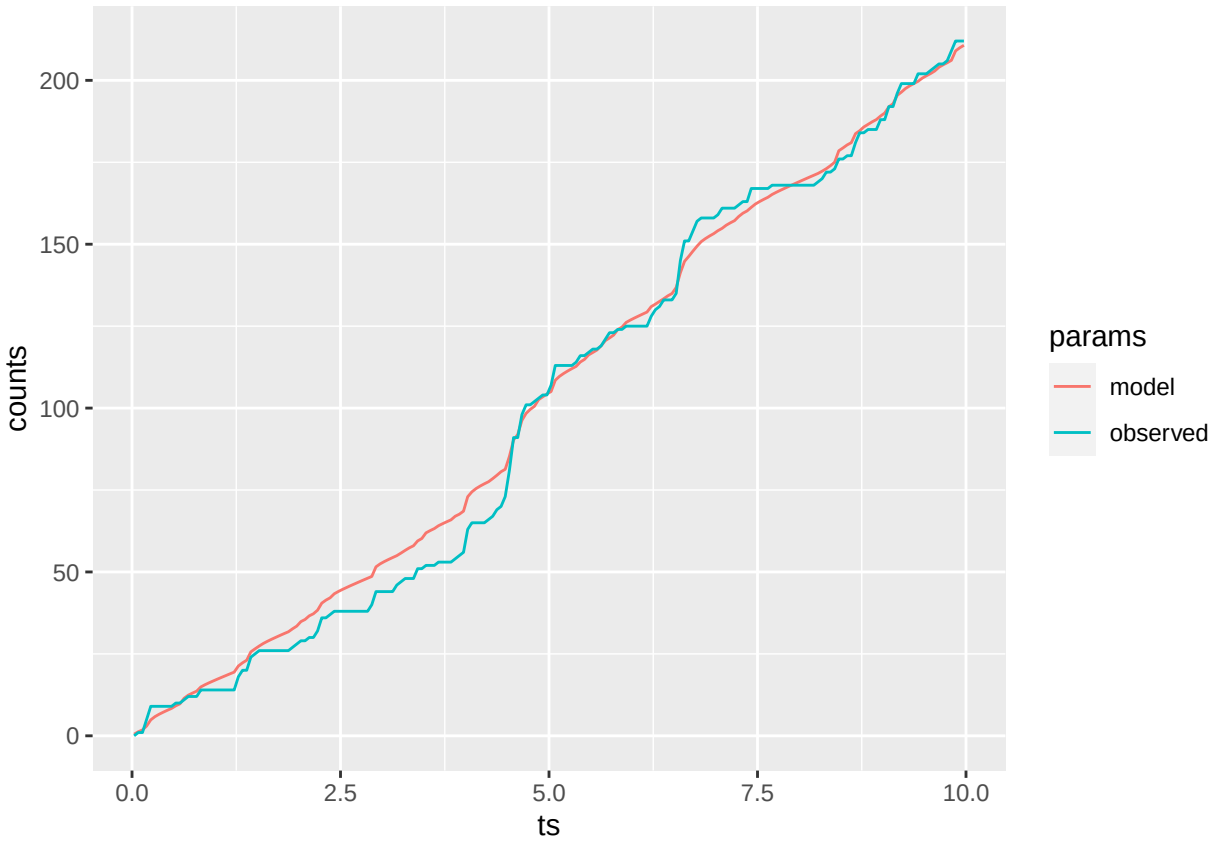
```
parms <- c(10, 0.5, 0.01, 0.5)
Tlim = 10
ss <- sample.hawkes(parms, Tlim)
```

```
## [1] 110   0
## [1] 51  1
## [1] 24  2
## [1] 9 3
## [1] 6 4
## [1] 5 5
## [1] 2 6
## [1] 1 7
## [1] 1 8
## [1] 2 9
## [1]  1 10
```

```
toplot.hist.lambda(parms, ss, Tlim)
```




```
toplot.hist.lambda(parms, ss, Tlim, cumulative = T)
```



Likelihood

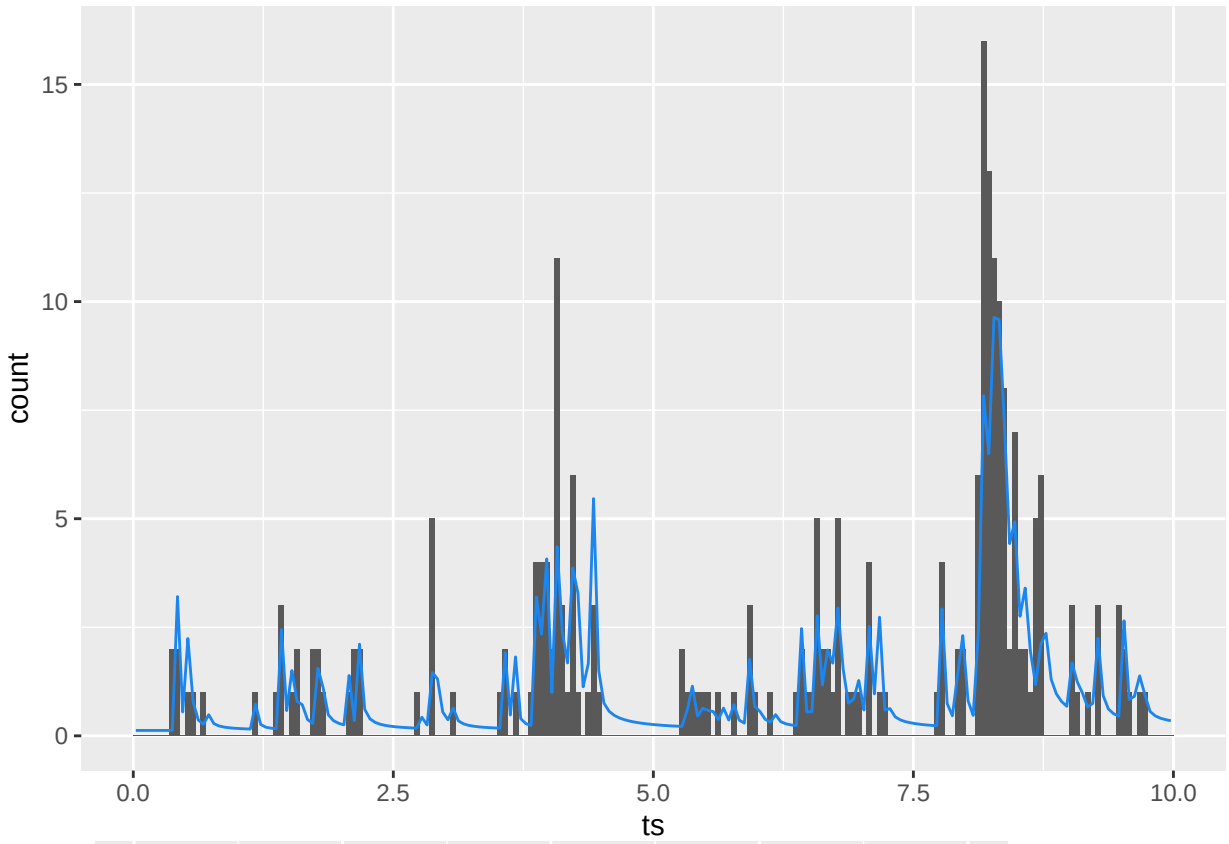
For point process models, given a set of observations $\mathcal{H}_t = (t_1, ..., t_N)$ in $(0, T)$, the expression of the log-likelihood is given by:

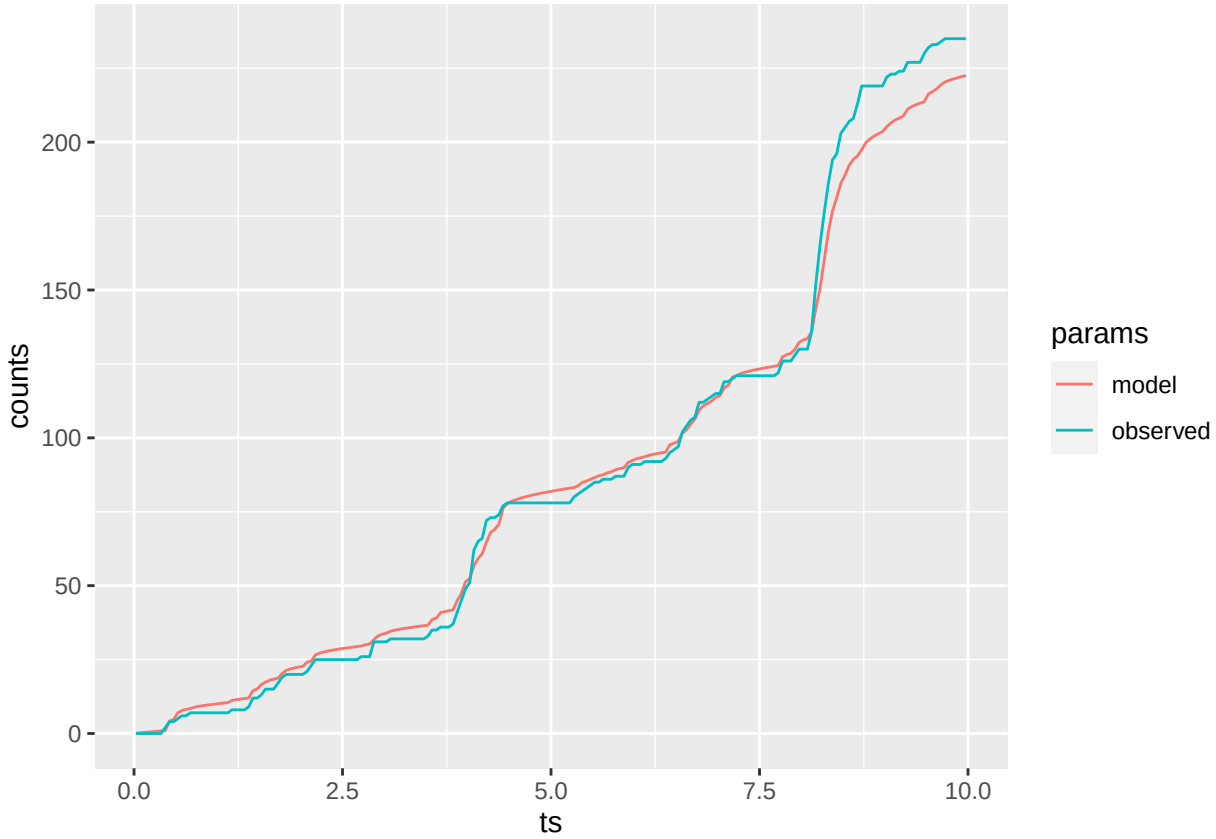
$$\mathcal{L}(\boldsymbol{\theta}, \mathcal{H}_t) = -\int_0^T \lambda(t)dt + \sum_{i=1}^{N} \log \lambda(t_i)$$

Which in out case is give by (we omit $\mathcal{H}_t$):

$$\mathcal{L}(\mu, K, c, p) = -\left(\mu T + K \sum_{i=1}^{N} (1 - c^{p-1}(T - t_i + c)^{1-p})\right) + \sum_{i=1}^{N} \log\left(\mu + K \sum_{j:t_j<t_i} \frac{(p-1)c^{p-1}}{(t_i - t_j + c)^p}\right)$$

Below, we are going to sample from a known parametrization and to retrieve the maximimum likelihood (ML) estimates of the parameters. To sample we use the following value of the parameters $\mu = 2.5, K = 0.9, c = 0.01, p = 1.5$ and $T = 10$. We show the histogram and the cumulative plot.



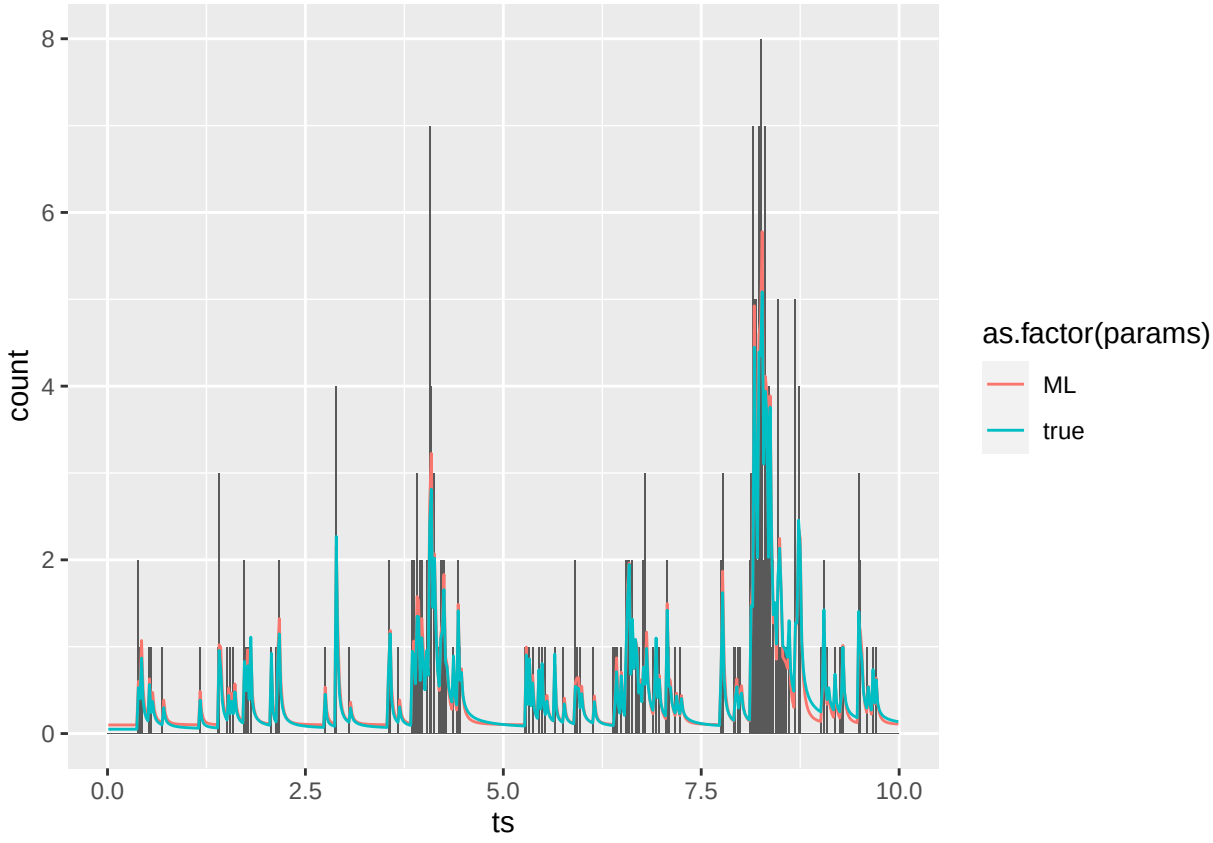


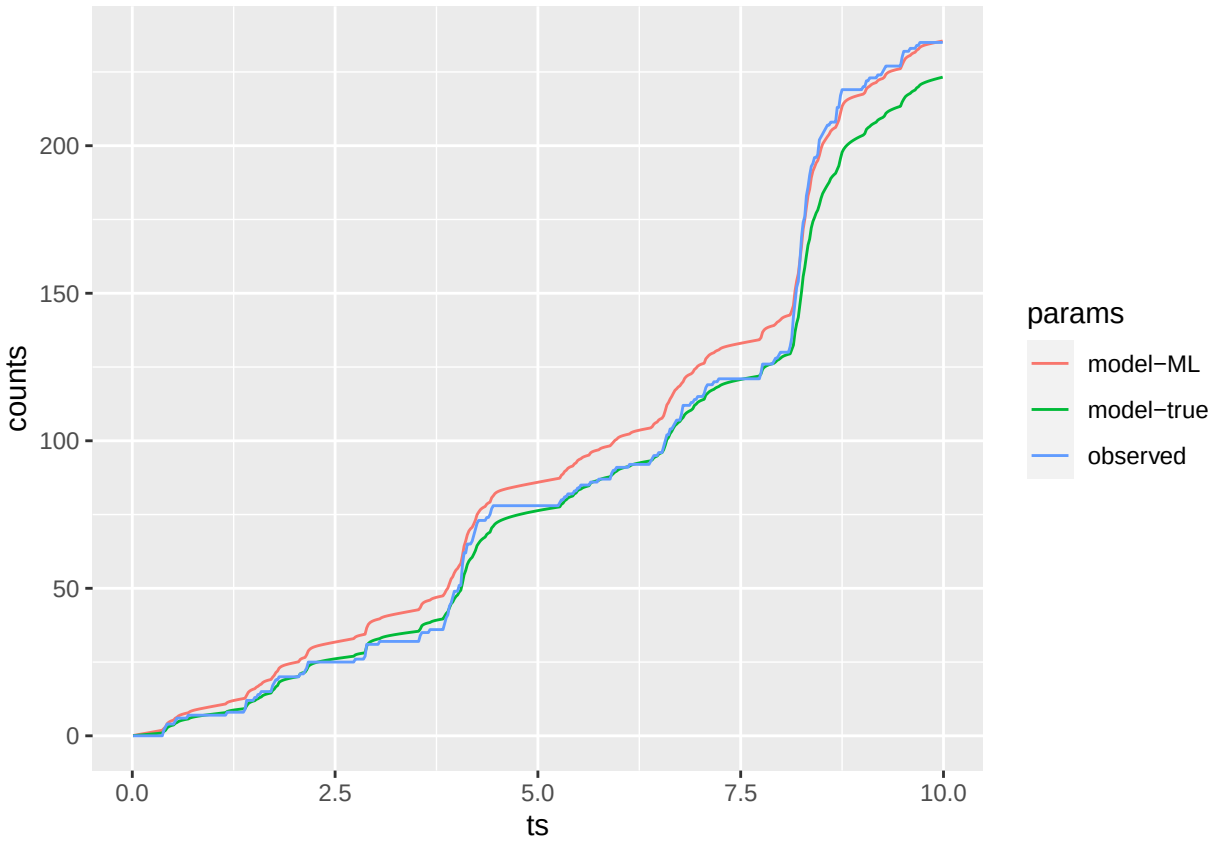
Here, the ML estimates of the paramters

```
## [1] 0
```

```
##           [,1]      [,2]      [,3]     [,4]
## true 2.500000 0.9000000 0.0100000 1.500000
## ML   4.950631 0.7893472 0.0399391 2.787613
```

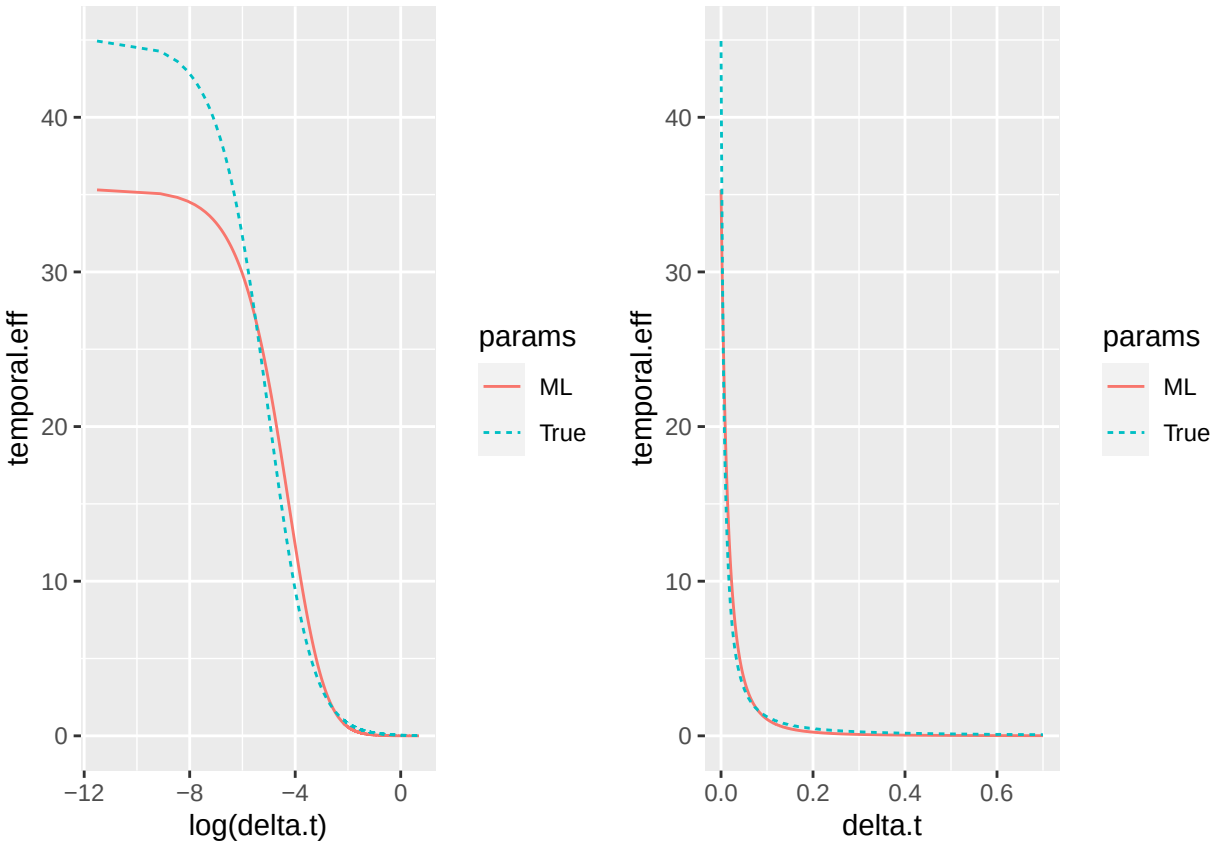
And the plot to compare the cumulative and bins counts.





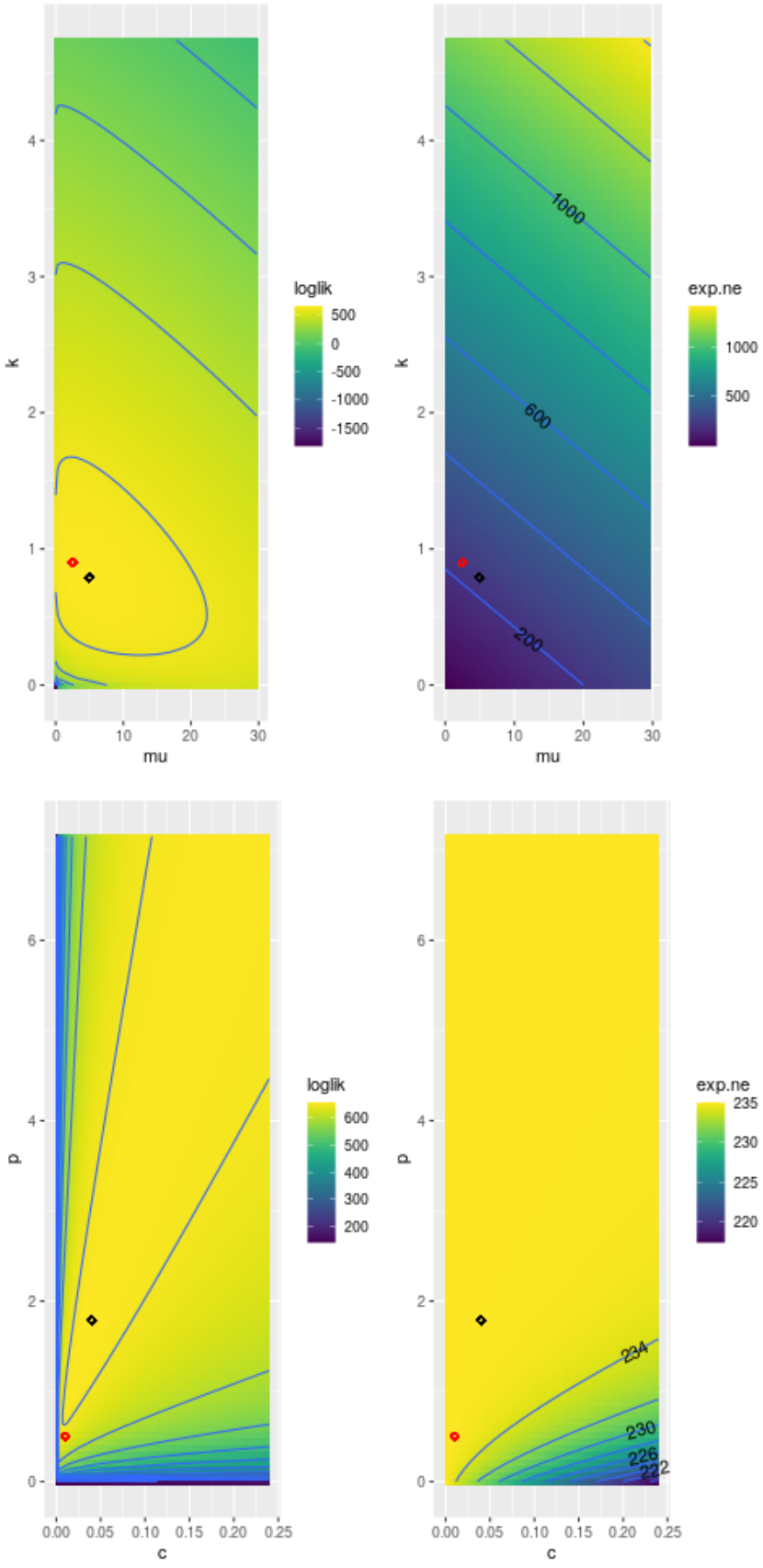
Here, we compare the difference in triggering effect between the true parameters and the ML estimates. We show the quantity $Kh(\Delta_t, c, p)$ as a function of $\Delta_t = t - t_i$.

```
## Warning: Removed 26000 row(s) containing missing values (geom_path).
```



Below we compare the expected number of points using the parameters that has generated the observations, using the ML estimates and the number of points in the sample used to obtain the ML estimates.

Below, we show log-likelihood of the model varying two parameters at the time. We split the parameters of the model in two groups: the productivity parameters μ, K and the Omori's law parameters c, p. Black diamond shows the ML estimate while the red one the value of the parameters generating the data.


k
mu
loglik
500
0
-500
-1000
-1500
exp.ne
1000
600
200
p
c
600
500
400
300
200
235
230
225
220
234
230
226
222

First approximation method - Univariate analysis

Here, we explore the differences in two ways of approximating an Hawkes process model. The first one, relies on a linear approximation of the log-intensity. Considering now, $\boldsymbol{\theta} = (\mu, K, c, p)$ the approximation around a point $\boldsymbol{\theta}_0$ is given by:

$$\overline{\log \lambda}(t, \boldsymbol{\theta}) | \boldsymbol{\theta}_0, \mathcal{H}_t = \frac{1}{\lambda_0} \sum_{i=1}^{4} (\theta_i - \theta_{0,i}) \frac{\partial}{\partial \theta_i} \lambda \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}$$

Where, $\lambda_0 = \lambda(t, \boldsymbol{\theta}_0)$.

The parameters of the model has to be positive, except for $p > 1$. To ensure that, we are goig to consider a different parametrization. The parameters of interest are now $\boldsymbol{\theta} = (\theta_1, ..., \theta_4)$ such that, $\mu = \exp \theta_1$, $K = \exp \theta_2$, $c = \exp \theta_3$, $p - 1 = \exp(\theta_4)$. The derivatives with respect the new parametrization are:

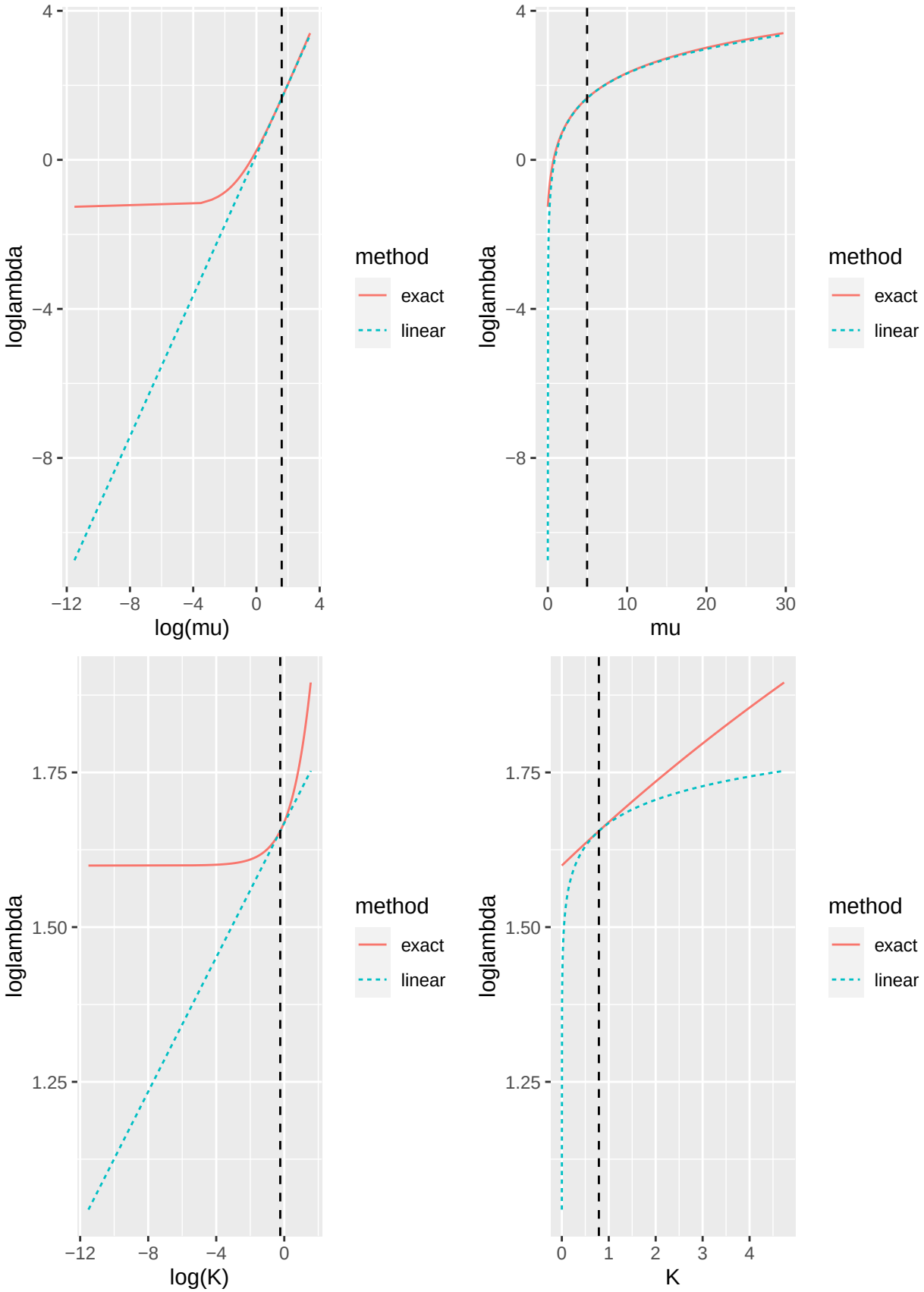
$$\frac{\partial}{\partial \theta_1} \lambda = \exp(\theta_1)$$

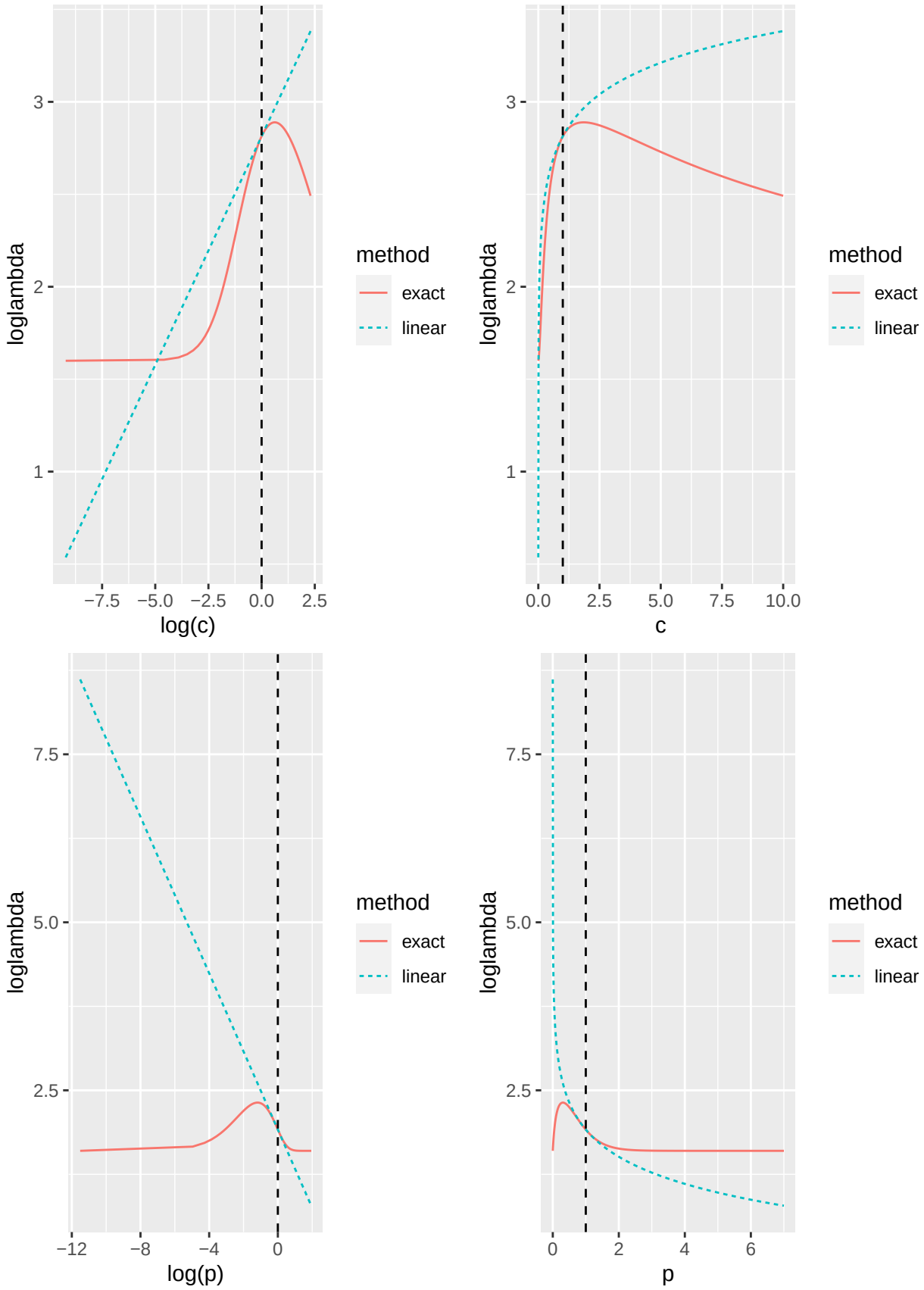
$$\frac{\partial}{\partial \theta_2} \lambda = \exp(\theta_2) \sum_{t_i < t} h(t - t_i, c, p)$$

$$\frac{\partial}{\partial \theta_3} \lambda = K(p-1) \exp((p-1)\theta_3) \sum_{t_i < t} (t - t_i + \exp(\theta_3))^{-p-1} \left[(p-1)(t - t_i) - \exp(\theta_3) \right]$$

$$\frac{\partial}{\partial \theta_4} \lambda = K \exp(\theta_4) c^{\exp(\theta_4)-1} \sum_{t_i < t} (t - t_i + c)^{-\exp \theta_4}$$

First thing, we check that the linearization works using the sample generated previously, here we vary a parameter and keep the others fixed at the value of the ML estimator. The exact and linearized log-intesity are considered at time $t = 5$, the latter is approximated around the ML estimator. Below, the plot shows the exact and linearized log-intensity as function of one parameter, the figure on the left shows the parameter in log scale, the plot on the right on its natural scale (except p which is $p - 1$ here). We remark that the log-intensity is linearized in the log scale and not in the natural scale of the parameters.


loglambda
log(mu)
mu
log(K)
K
method
exact
linear


loglambda
log(c)
c
log(p)
p
method
exact
linear
3
2
1
7.5
5.0
2.5
−7.5
−5.0
−2.5
0.0
2.5
0.0
2.5
5.0
7.5
10.0
−12
−8
−4
0
0
2
4
6

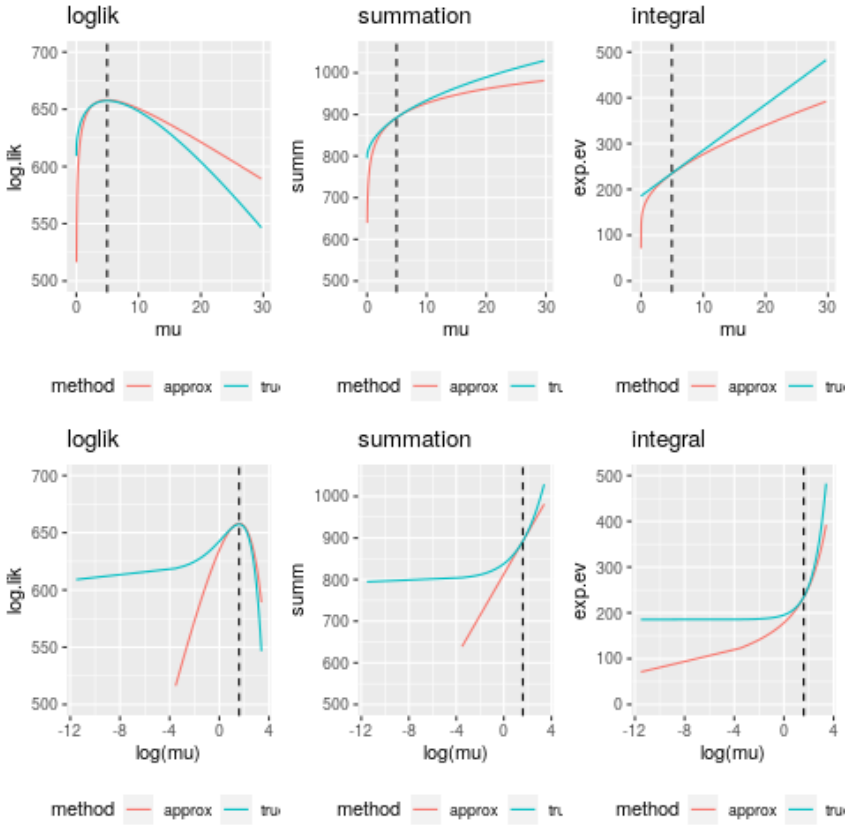
I am noticing that, the log-intensity with respect to $\theta_1 = \log(\mu)$ and $\theta_2 = \log(K)$ is just $\exp(\theta_1) + const$ in the first case and $const + \exp(\theta_2) * const_2$ in the second case. For this two, I think that linearizing is okay and provides an approximated log-intesity similar to the true one. In the case of $\theta_3 = log(c)$ and $\theta_4 = \log(p-1)$, instead, the log-intensity is "bell-shaped" and perhaps I think it will be more dangerous to approximate the log-intesity linearly.

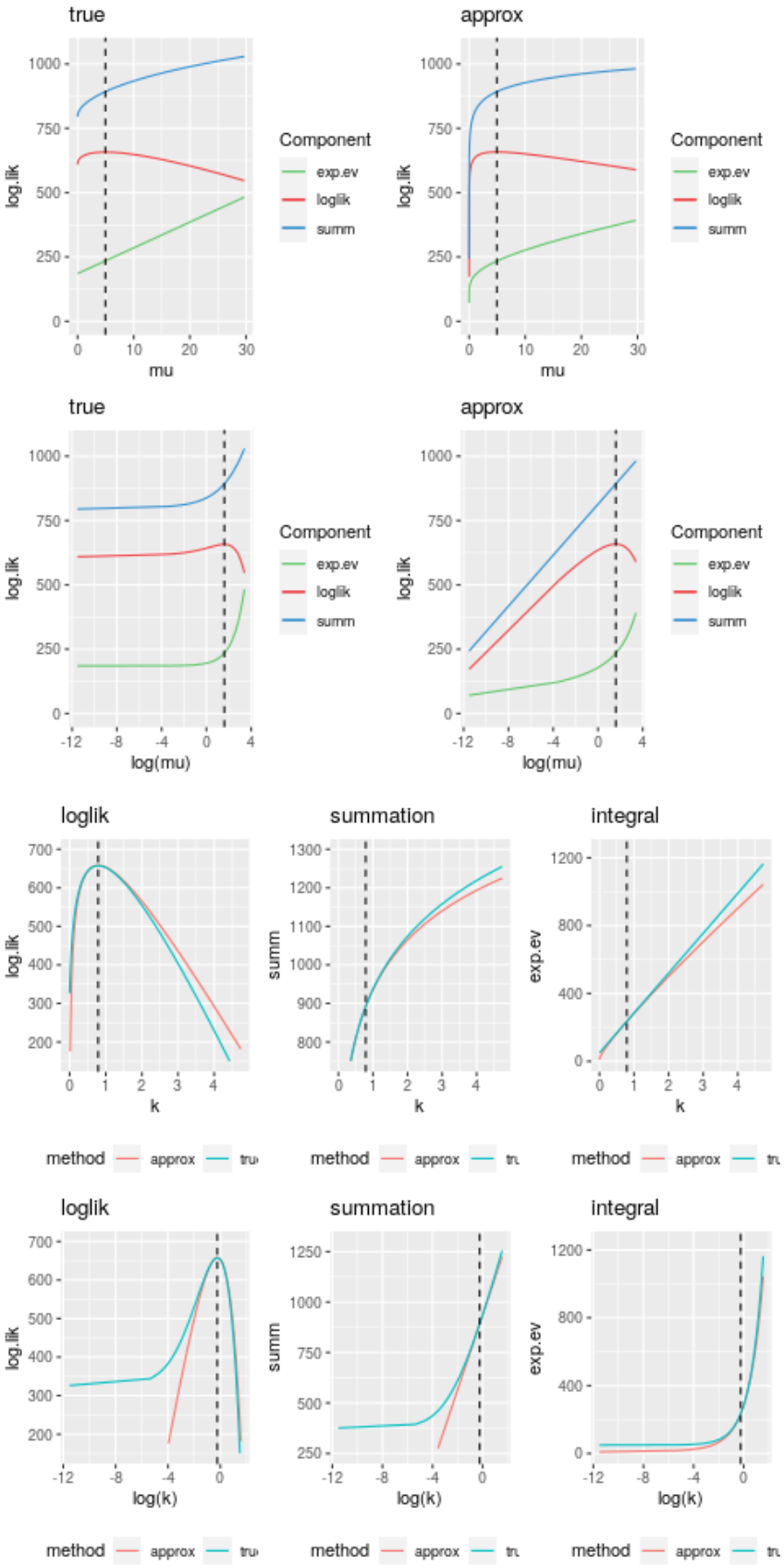
To check how the linearization works with respect one parameter at the time, we look at the values of the approximated log-likelihood of the parameters 1) keeping the other parameters equal to their ML estimate, the linearization is wrt to the ML estimate; 2) using data simulated from some known parametrization.

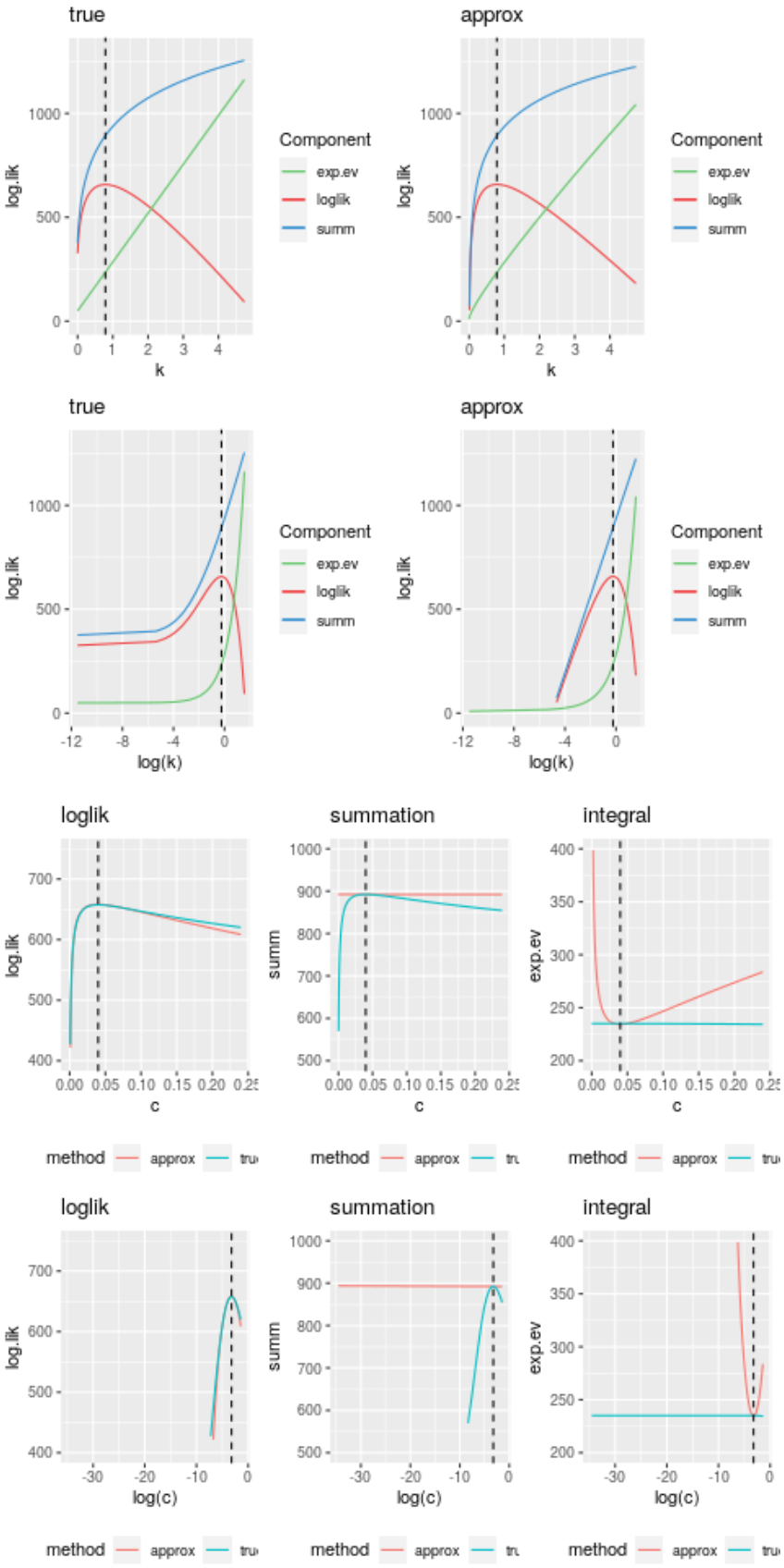
```
##           [,1]      [,2]      [,3]     [,4]
## true 2.500000 0.9000000 0.0100000 0.500000
## ML   4.950631 0.7893472 0.0399391 1.787613
```

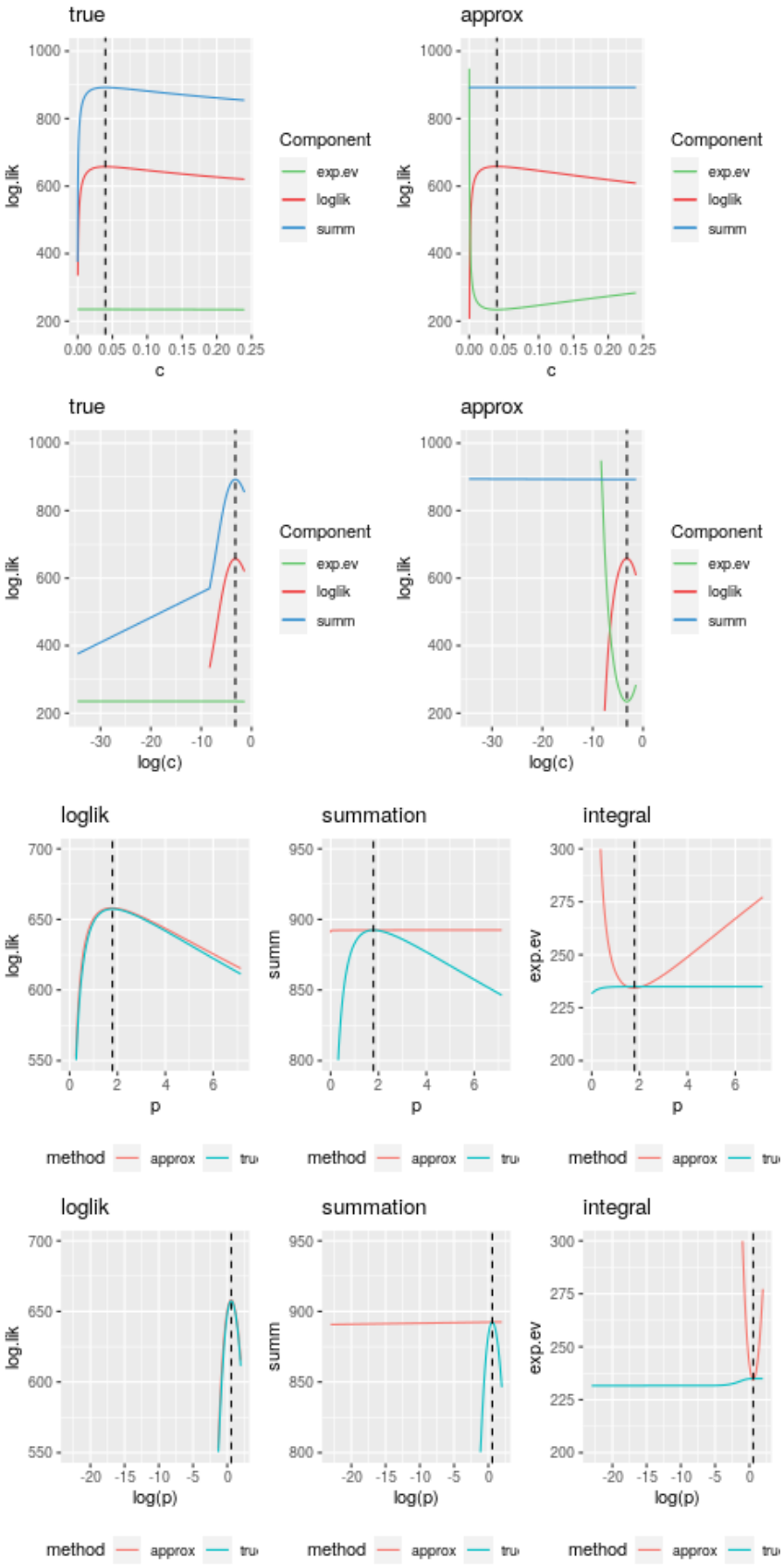
Given a observations $\mathcal{H}_t = (t_1, ..., t_N), t_i \in (0, T)$ and a set of points $t_{m1}, ..., t_{mP}$ such that $t_{mj} \in (0, T)$ and $t_{m(j+1)} - t_{mj} = w$ (the points are equidistant), the approximated log-likelihood is given by:

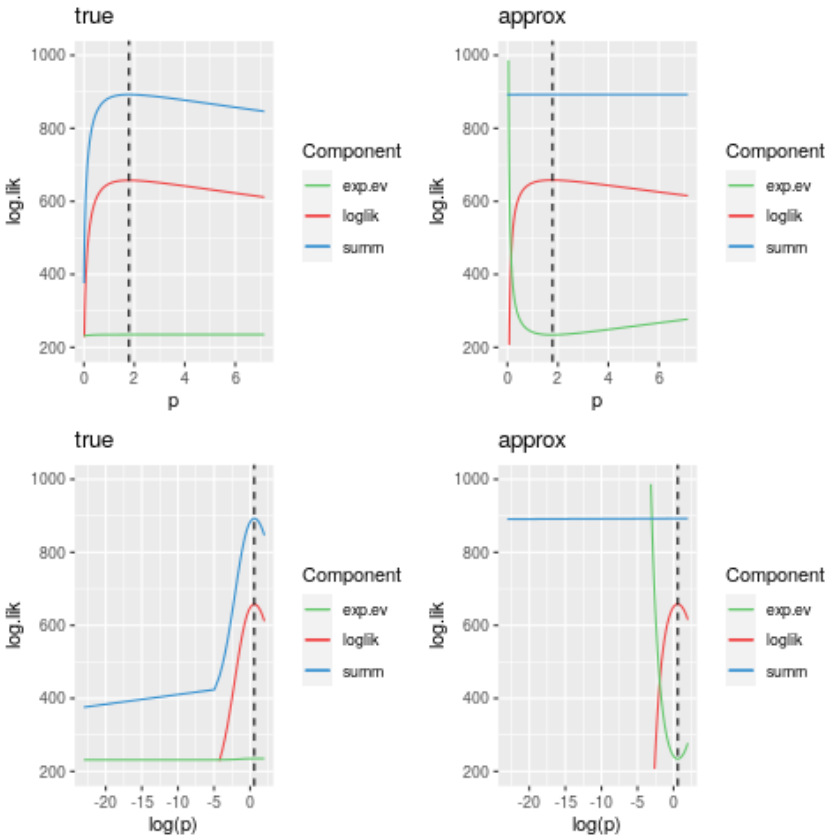
$$\overline{\mathcal{L}}(\boldsymbol{\theta}) = -\sum_{j=1}^{P} \exp\{\overline{\log \lambda}(t_{mj}, \boldsymbol{\theta})\} w + \sum_{i=1}^{N} \overline{\log \lambda}(t_i, \boldsymbol{\theta})$$




true
approx
log.lik
mu
log(mu)
Component
exp.ev
loglik
summ
loglik
summation
integral
k
log(k)
summ
exp.ev
method
approx


true
approx
log.lik
k
log(k)
Component
exp.ev
loglik
summ
loglik
summation
integral
summ
exp.ev
c
log(c)
method
approx


true
approx
log.lik
c
log(c)
Component
exp.ev
loglik
summ
loglik
summation
integral
p
log(p)
summ
exp.ev
method
approx



Second approximation method - Univariate Analysis

Another way to approximate an Hawkes process model relies on the following view of the likelihood:

$$\mathcal{L}(\mu, K, c, p) = -\mu T - \sum_{i=1}^{N} I_h(t_i, c, p) + \sum_{i=1}^{N} \log \lambda(t_i)$$

Where

$$I_h(t_i, \boldsymbol{\theta}) = K(1 - c^{p-1}(T - t_i + c)^{1-p})$$

This likelihood can be seen as the sum of three Poisson counts likelihood with appropriate centroids, exposures and counts. In a Poisson counts model, the domain, which in this case is $(0, T)$, is divided in B regions (bins) with centroids $t_{c1}, ..., t_{cB}$, having areas (width) E_{ci}. The observed data are the number of points observed in each region N_{ci}. Assuming that, in each region, the intensity is constant and equal to $\lambda(t_{ci})$, the expected number of points in each region is $E_{ci}\lambda(t_{ci})$. The log-likelihood of the model is

$$\mathcal{L}_{PC} = -\sum_{i=1}^{B} E_{ci}\lambda(t_{ci}) + \sum_{i=1}^{B} \log(\lambda(t_{ci}))N_{ci}$$

Let's see how we can use Poisson counts models to approximate the single component of the Point process likelihood.

Let's suppose to have observed $t_1, ..., t_N$ points, considering the observed points as centroids of the regions, setting for each region i, $E_{ci} = 0$ and $N_{ci} = 1$, the resulting log-likelihood of the Poisson counts model is

$$\mathcal{L}_{PC} = \sum_{i=1}^{N} \log \lambda(t_i)$$

Which is the summation component of the Point process likelihood.

If we consider the observed points as centroids but we set $E_{ci} = 1$ and $N_{ci} = 0$ we obtain that

$$\mathcal{L}_{PC} = -\sum_{i=1}^{N} \lambda(t_i)$$

Now, considering $\lambda(t_i) = I_h(t_i)$ we obtain the triggering part of the integral component of the Point process log-likelihood.

Finally, considering just one region with exposure $E_1 = 1$, counts $N_1 = 0$ and $\lambda(t) = \mu T$ we obtain the background part of the integral component of the point process log-likelihood.

So we can see decompose the log-likelihood of a Point process model with intesity $\lambda(t)$ and observations $t_1, ..., t_N$, as product of three Poisson counts likelihood with intensity $\lambda_1(t), \lambda_2(t), \lambda_3(t)$ with appropriate centroids, exposures and counts. The intensities of the Poisson counts models are given by:

$$\lambda_1(t) = \lambda(t)$$
$$\lambda_2(t) = I_h(t)$$
$$\lambda_3(t) = \lambda_3 = T\mu$$

The Point process log-likelihood can be rewritten as

$$\mathcal{L} = -\lambda_3 - \sum_{i=1}^{N} \lambda_2(t_i) + \sum_{i=1}^{N} \log \lambda(t_i)$$

Use of the decomposition

We can use this decomposition to approximate each component separately. INLA works with Poisson counts models for which the log-intensity is linear. If it is not-linear the model is approximated as in the first approximation method exposed in the previous section. Therefore, in place of the summation component we have the linearized log-intensity as before.

If $\log I_h$ is not linear in the parameters it will be linearized and $\overline{\log I_h}$ will be used.

Regarding the third component, considering $\mu = \exp(\theta_1)$ we see that

$$\log \lambda_3 = \log(T) + \theta_1$$

which is linear in θ_1 and thus, it does not need to be approximated.

The resulting approximated likelihood is given by

$$\overline{\mathcal{L}_2}(\boldsymbol{\theta}) = -\mu T - \sum_{i=1}^{N} \exp\left\{\overline{\log I_h}(t_i, \boldsymbol{\theta})\right\} + \sum_{i=1}^{N} \overline{\log \lambda}(t_i, \boldsymbol{\theta})$$

Where

$$\overline{\log I_h}(t_i, \boldsymbol{\theta}) = \log I_h(t_i, \boldsymbol{\theta}_0) + \sum_j (\theta_j - \theta_{0j}) \frac{\partial}{\partial \theta_j} \log I_h \Big|_{\boldsymbol{\theta}=\boldsymbol{\theta}_0}$$

Where

$$I_h(t_i, \boldsymbol{\theta}) = K(1 - c^{p-1}(T - t_i + c)^{1-p})$$

Considering that $\mu = \exp(\theta_1), K = \exp(\theta_2), c = \exp(\theta_3), p - 1 = \exp(\theta_4)$

$$I_h(t_i, \boldsymbol{\theta}) = \exp(\theta_2)(1 - \exp\{\theta_3 \exp(\theta_4)\}[T - t_i + \exp(\theta_3)]^{-\exp(\theta_4)})$$

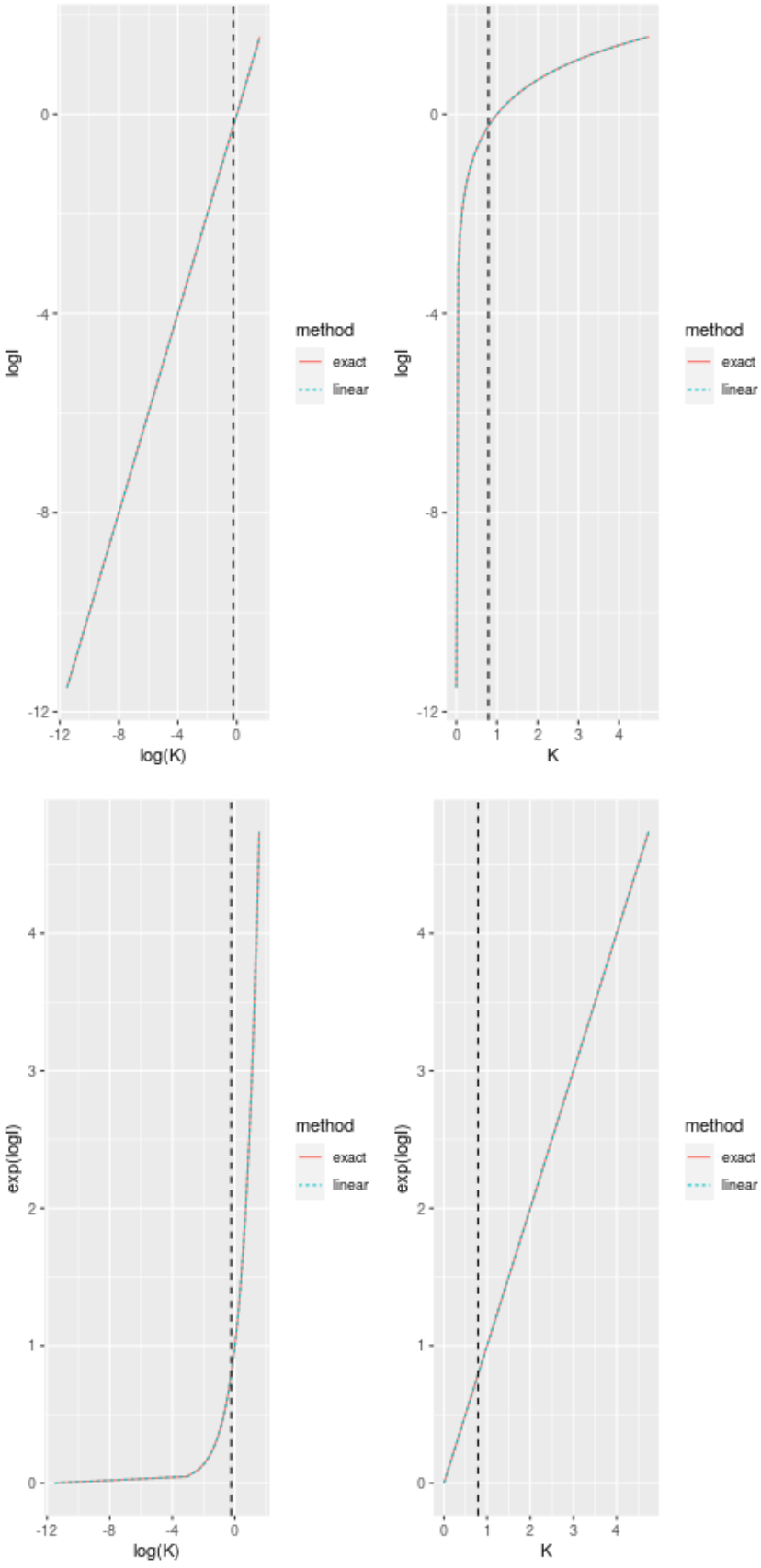
We have that

$$\frac{\partial}{\partial \theta_2} \log I_h = 1$$

$$\frac{\partial}{\partial \theta_3} \log I_h = \frac{1}{I_h}\Big[- \exp(\theta_2)\Big(\exp\big\{\theta_3 \exp(\theta_4) + \theta_4\big\}\big(T - t_i + \exp(\theta_3)\big)^{-\exp(\theta_4)-1}(T - t_i)\Big)\Big]$$

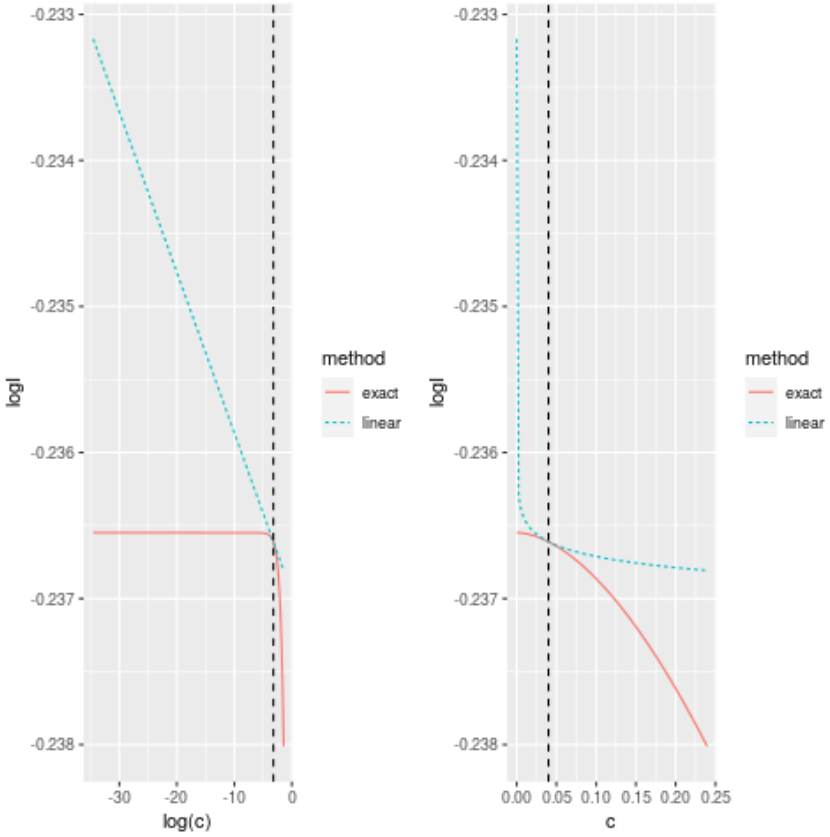
$$\frac{\partial}{\partial \theta_4} \log I_h = \frac{1}{I_h}\Big[- \exp(\theta_2) \exp\big(\theta_4 + \theta_3 \exp(\theta_4)\big)\big(T - t_i + \exp(\theta_3)\big)^{-\exp\theta_4}\big(\theta_3 - \log(T - t_i + \exp\theta_3)\big)\Big]$$

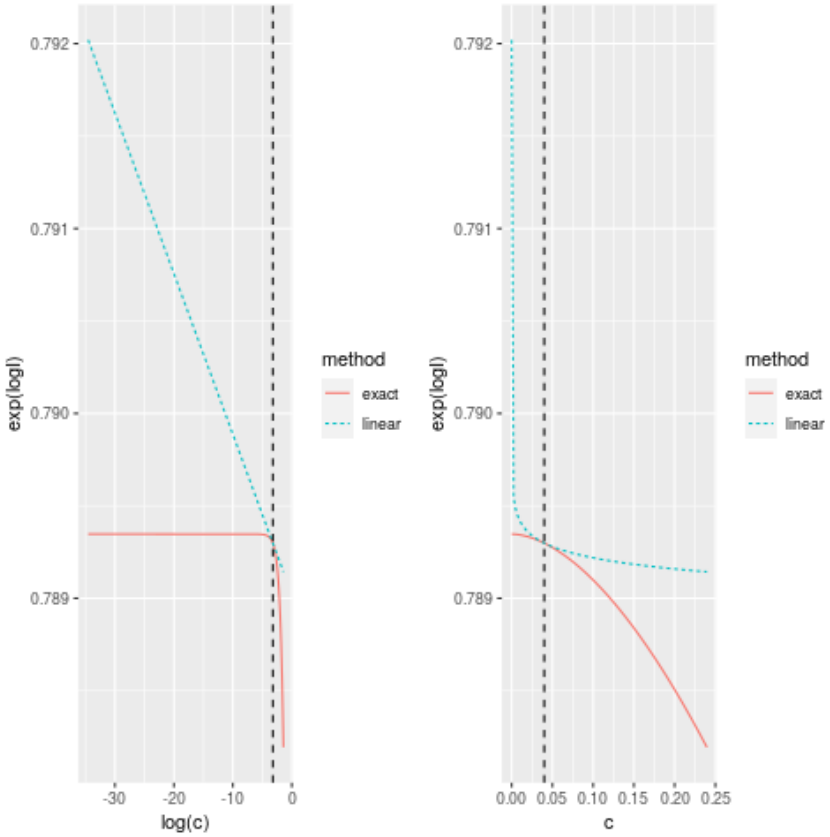
Below, we show how the linearization approximate the function $\log I_h$ as a fuction of one parameter (K, c, p) at the time. We remark that the linearization is performed with respect the log of the parameters, except for p for which the linearization is taken with respect to $\theta_4 = \log(p - 1)$. We notice that, the linearization approximate perfectly the functio with respect to K, this is because $\log I_h$ is a linear function of θ_2. We show the approximate and exact $\log I_h$ as a function of θ_j and $\exp\theta_j$. We do the same with respect to I_h


logl
log(K)
K
exp(logl)
method
exact
linear

```
## pdf
##   2
```

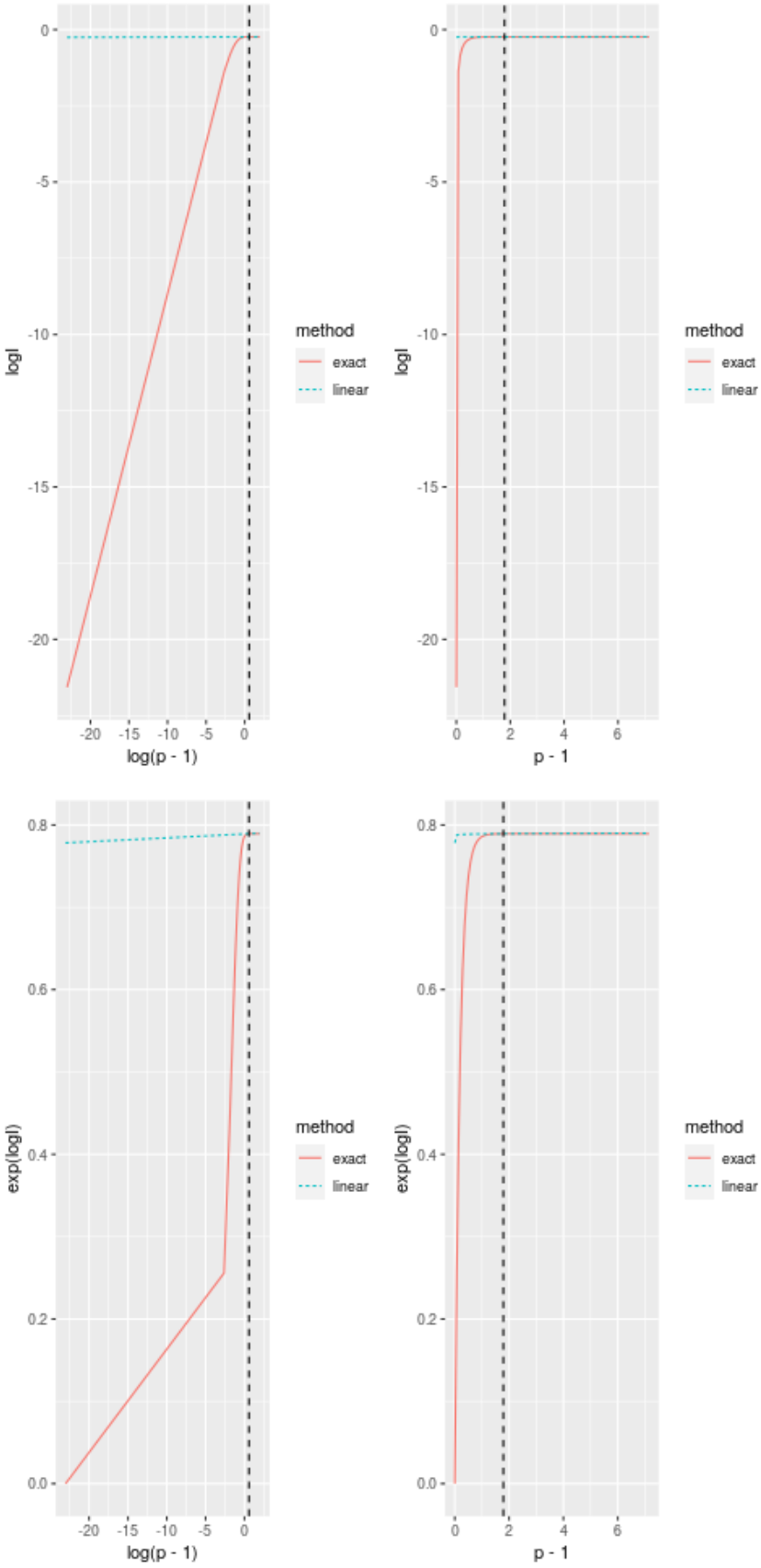
```
## pdf
##   2
```


-0.233
-0.234
-0.235
-0.236
-0.237
-0.238
logl
-30
-20
-10
0
log(c)
method
exact
linear
0.00
0.05
0.10
0.15
0.20
0.25
c


0.792
0.791
0.790
0.789
exp(logl)
-30
-20
-10
0
log(c)
method
exact
linear
0.00
0.05
0.10
0.15
0.20
0.25
c

```
## pdf
##   2
```

```
## pdf
##   2
```

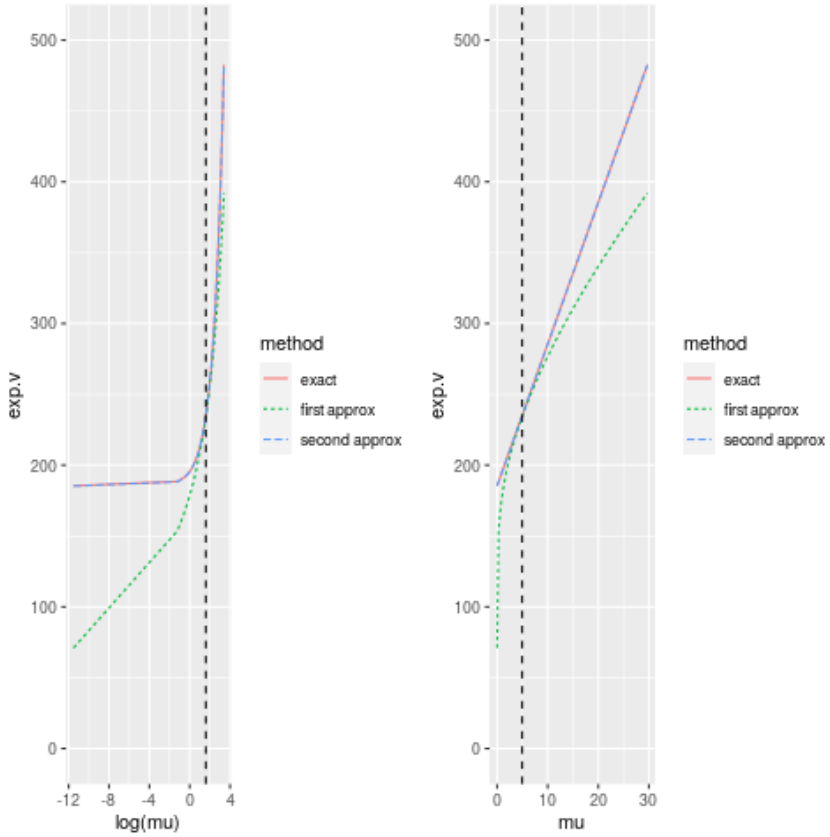

0
-5
-10
-15
-20
logl
-20 -15 -10 -5 0
log(p - 1)
method
exact
linear
0 2 4 6
p - 1
0.8
0.6
0.4
0.2
0.0
exp(logl)

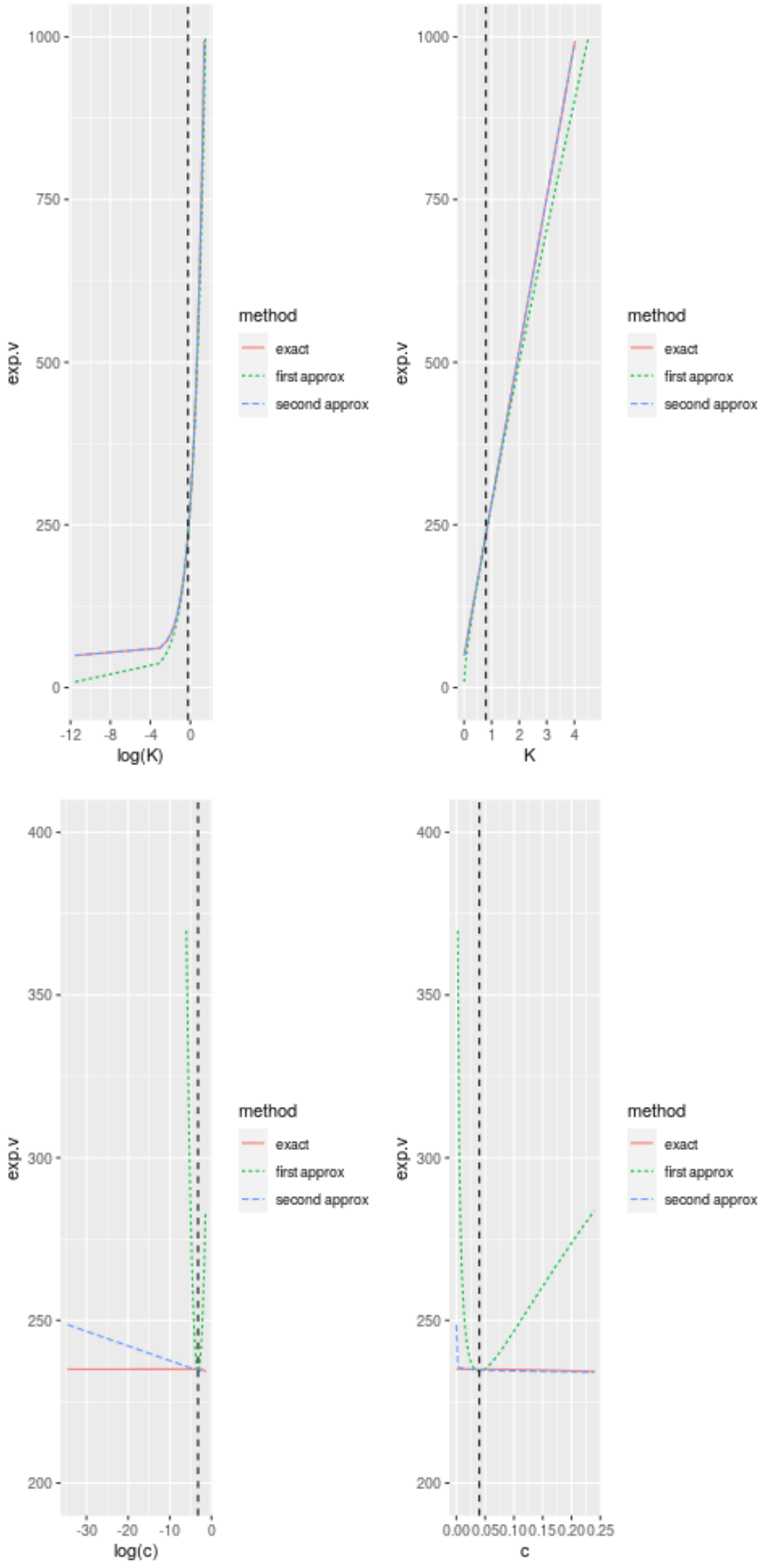
Expected value approximation comparison - Univariate Analysis

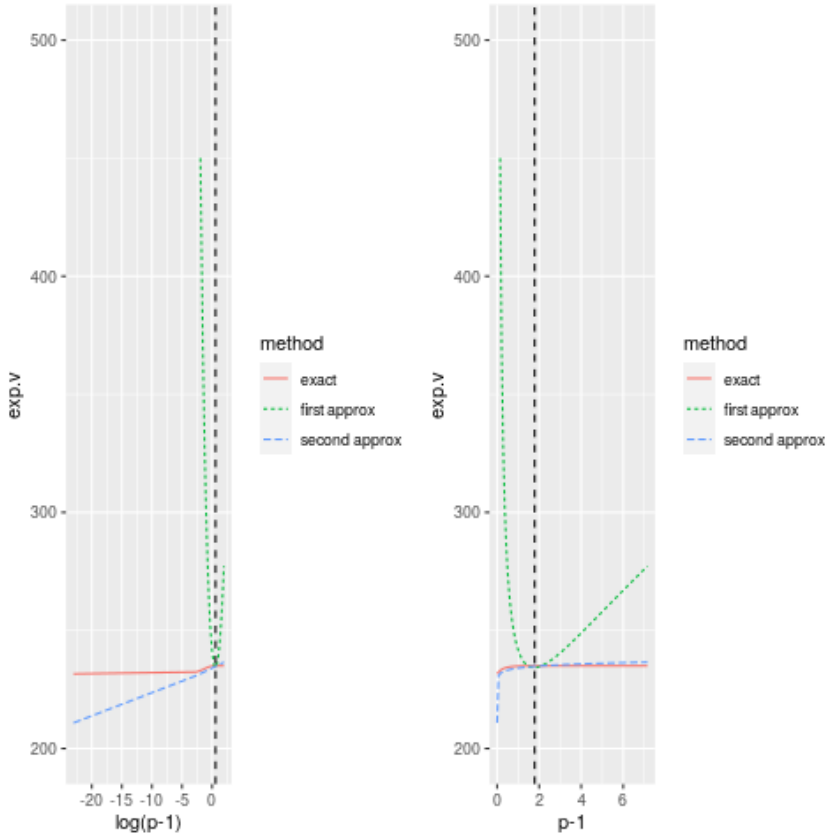
Here, we compare the two expected value approximations. The first one is based on numerical integration of the exponential of the linearized log-intensity while the second is based on a linearization of the log of the triggerig component of the integral. We expect the second method to be more accurate than the first one.

We compare the expected values varying one parameter at the time

Varying μ and K the second method is exact. Also with respect to c and p the second approximation method provides better approximation of the expected value. However, we are going to see in the next section that this may not be desirable.



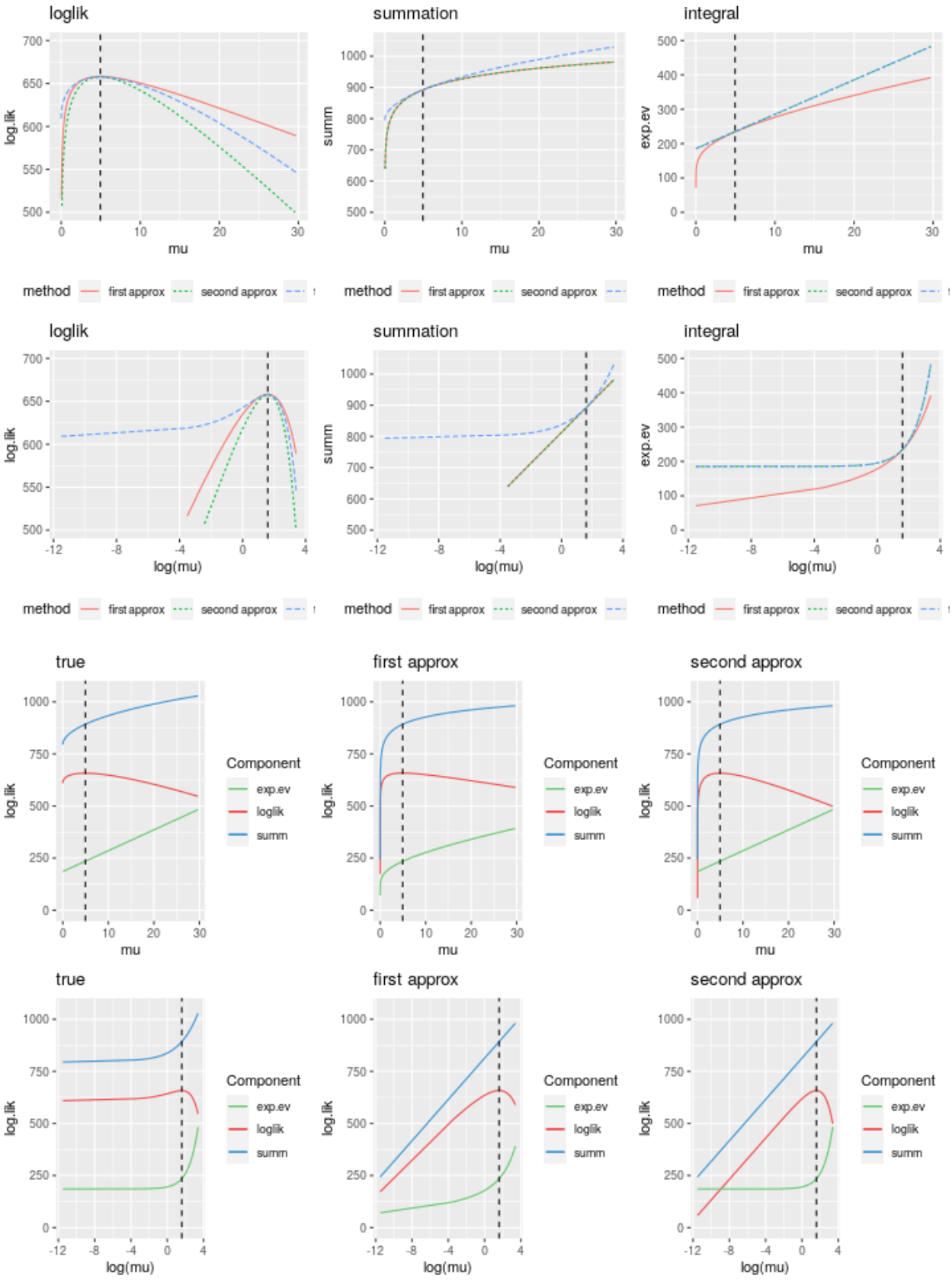

exp.v
1000
750
500
250
0
-12
-8
-4
0
log(K)
method
exact
first approx
second approx
exp.v
1000
750
500
250
0
0
1
2
3
4
K
method
exact
first approx
second approx
exp.v
400
350
300
250
200
-30
-20
-10
0
log(c)
method
exact
first approx
second approx
exp.v
400
350
300
250
200
0.00
0.05
0.10
0.15
0.20
0.25
c
method
exact
first approx
second approx

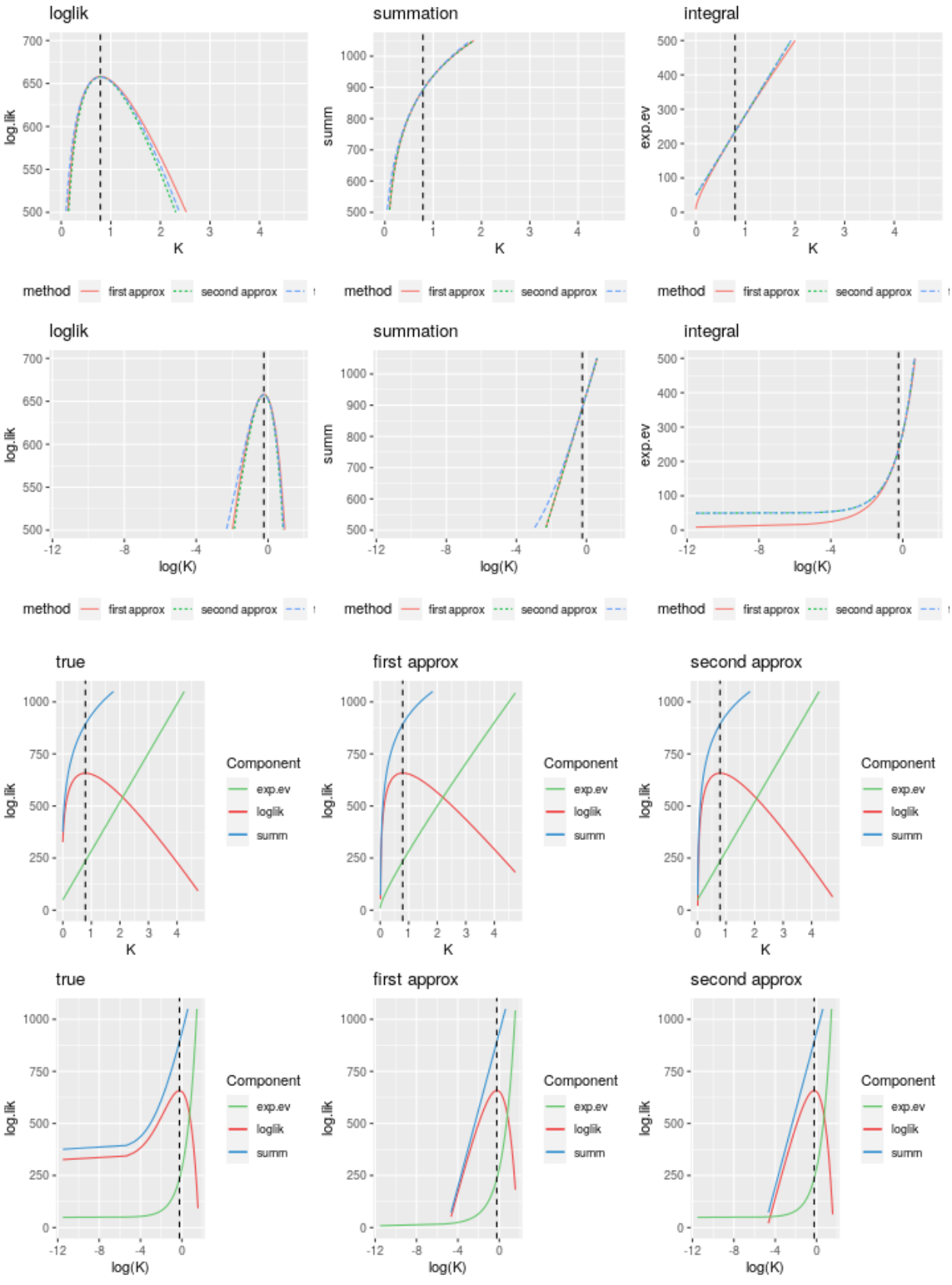


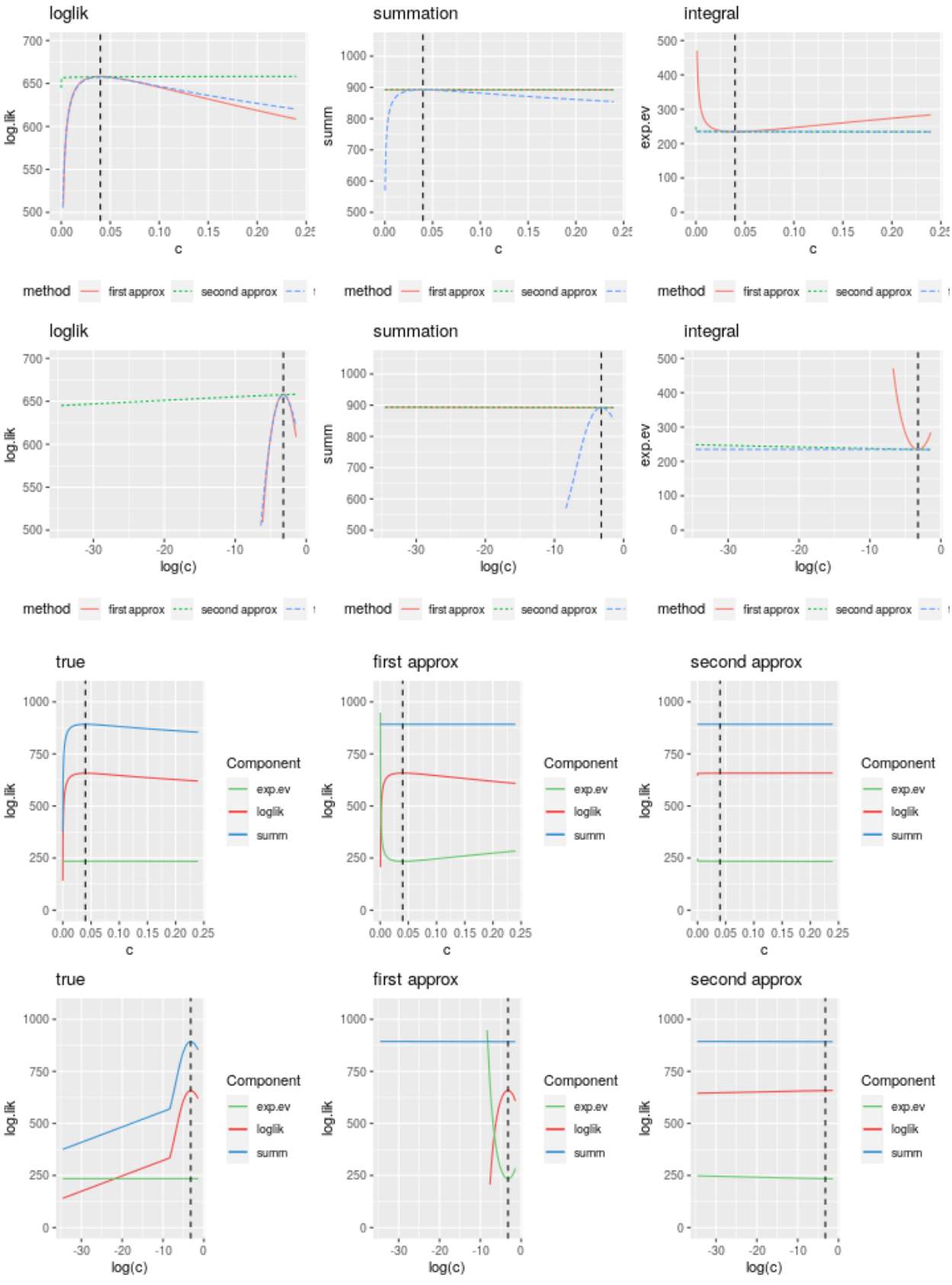
Log-likelihood approximation comparison - Univariate Analysis

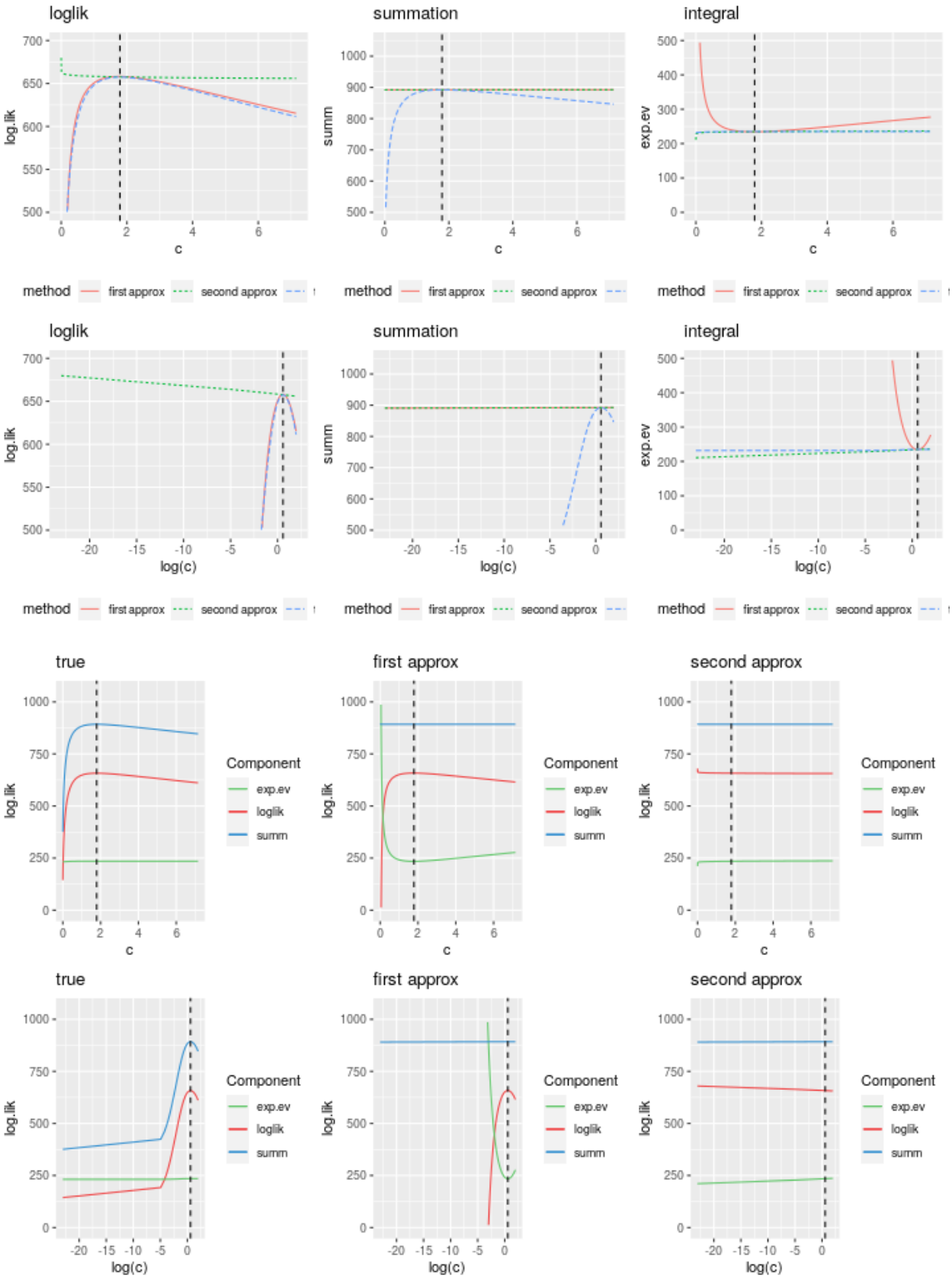
Here we compare the approximated log-likelihoods using the two methods exposed before. They differs solely on how they approximate the integral of the intensity over the domain of interest (integral component of the log-likelihood). We have seen before that the second approximation method provides better approximations of the expected value and we say that, however, it may not be desirable. Here, we show why, showing the log-likelihood as function of one parameter at the time, the others are fixed to their ML estimate. Also the linearizations are performed with respect to the ML estimates of the parameters.

It is clear looking at the log-likelihood as function of c and $p-1$ that, regarding the first approximation, the error in the integral component approximation somehow compensates for the error in the summation component approximation. I have no idea why, but the resulting approximated log-likelihood is similar to the true one. The second approximation method, instead, even if it approximate better the integral component, brings the same error on the summation component. This error is not compensated by the error in the integral componenet and leads to a biased log-likelihood approximation.


loglik
summation
integral
log.lik
summ
exp.ev
mu
log(mu)
method
first approx
second approx
true
first approx
second approx
Component
exp.ev
loglik
summ

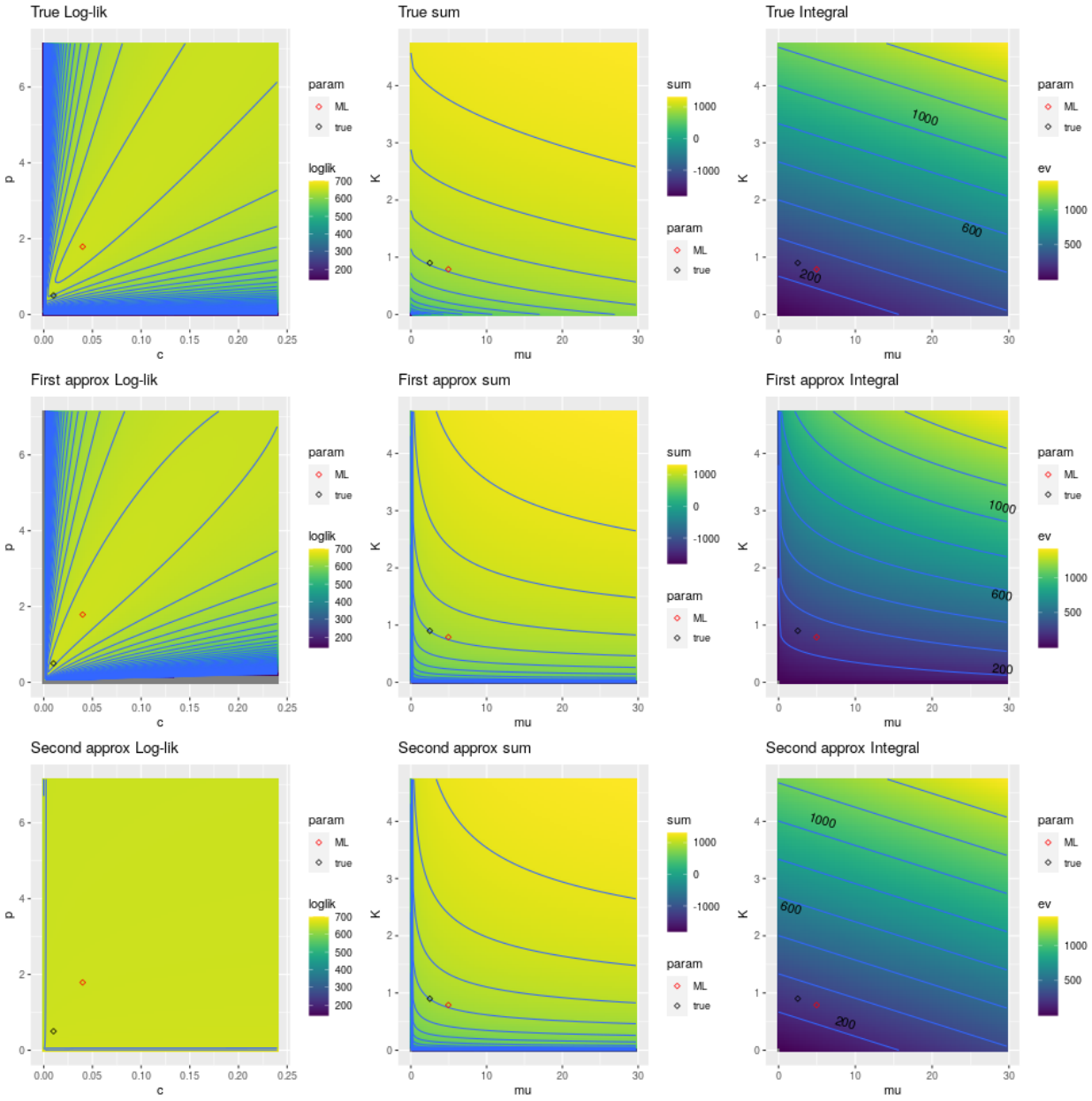

loglik
summation
integral
log.lik
summ
exp.ev
K
log(K)
method
first approx
second approx
true
first approx
second approx
Component
exp.ev
loglik
summ

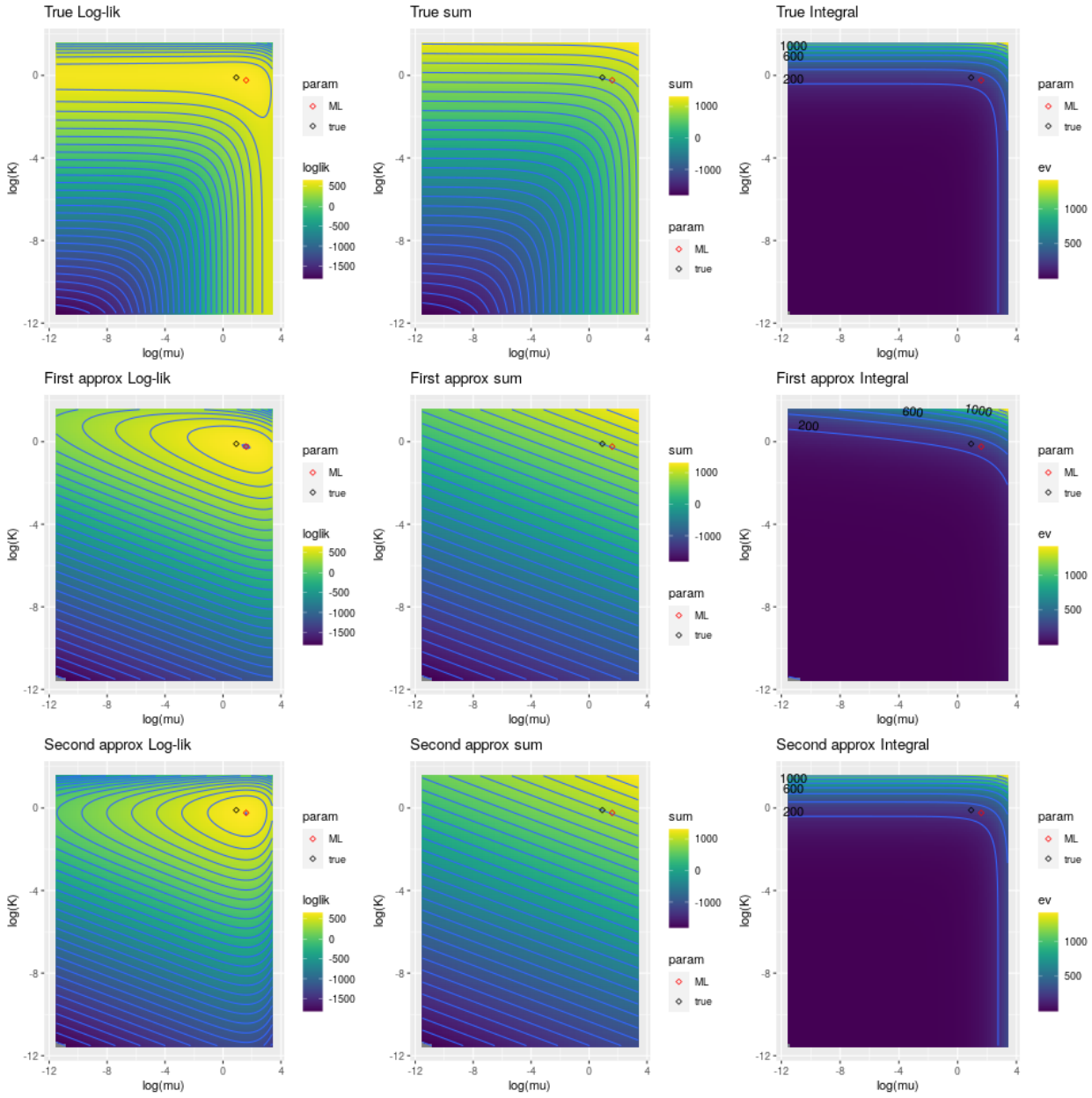

loglik
summation
integral
log.lik
summ
exp.ev
c
log(c)
method
first approx
second approx
true
first approx
second approx
Component
exp.ev
loglik
summ

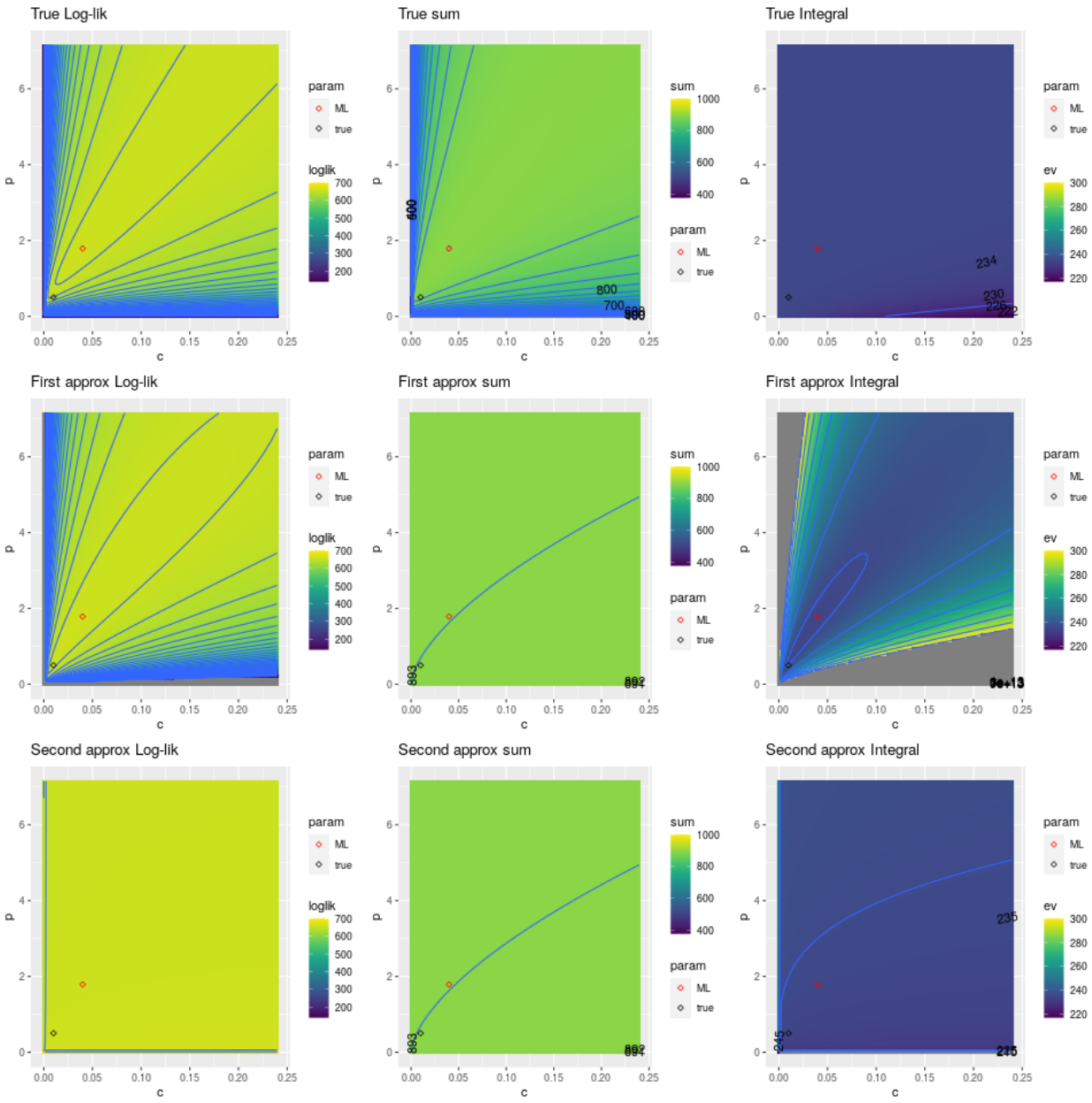

loglik
summation
integral
log.lik
summ
exp.ev
c
log(c)
method
first approx
second approx
true
first approx
second approx
Component
exp.ev
loglik
summ

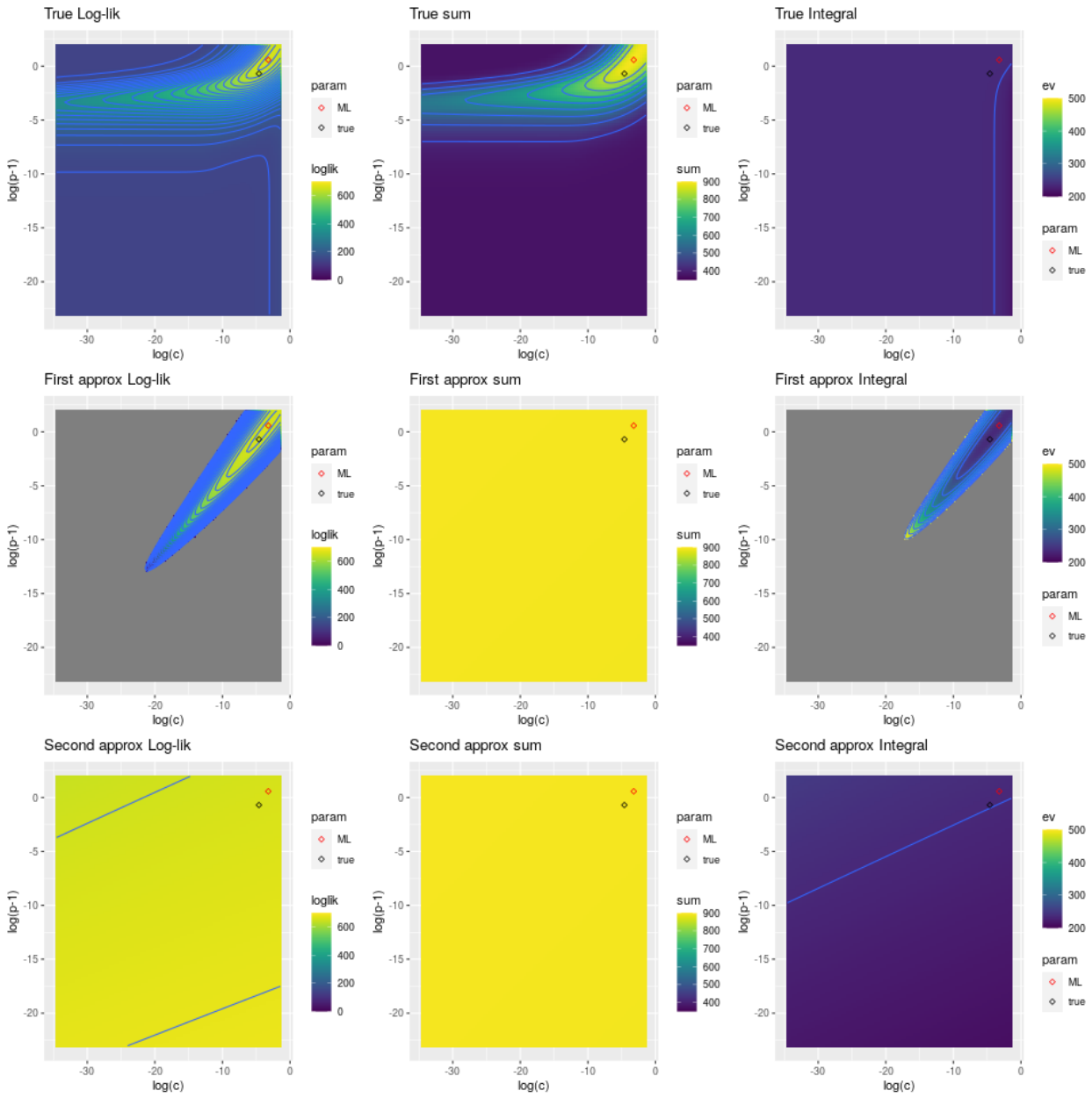
Log-likelihood approximation comparison - Bivariate Analysis

Here, we basically repeat the visual analysis done in the previous section but varying two parameters at the time. Given the role of each parameters we consider only two couples: the productiivity parameters μ, K and the tempora triggering parameters c, p.




True Log-lik
True sum
True Integral
First approx Log-lik
First approx sum
First approx Integral
Second approx Log-lik
Second approx sum
Second approx Integral
log(K)
log(mu)
param
ML
true
loglik
sum
ev


True Log-lik
True sum
True Integral
First approx Log-lik
First approx sum
First approx Integral
Second approx Log-lik
Second approx sum
Second approx Integral
param
ML
true
loglik
sum
ev
p
c


True Log-lik
True sum
True Integral
First approx Log-lik
First approx sum
First approx Integral
Second approx Log-lik
Second approx sum
Second approx Integral
log(p-1)
log(c)
param
ML
true
loglik
sum
ev