

# Using the R package chngpt

Youyi Fong

December 17, 2020

## 1 Strength and weakness of this package

There are many R packages for fitting models with change points/thresholds. Similar to the *segmented* package (Muggeo, 2008), this package is designed to work with threshold regression models, not structural change models. Unique strengths of this package include:

- Supports fourteen different continuous two-phase models (Son and Fong, 2020).
- Implements fast grid search, a super fast and exact algorithm, for linear models (Elder and Fong, 2019).
- Implements smooth approximation, a fast and accurate algorithm, for logistic models (Fong et al., 2017a).
- Provides bootstrap-based confidence intervals for both independent and time series data and supports parallel processing to improve speed (Son and Fong, 2020).
- Provides model-robust analytical confidence intervals for logistic regression models (Fong et al., 2017b).
- Supports hypothesis testing (Fong et al., 2015, 2017a).

The weakness of this package include:

- Support for multi-threshold models is limited to one type of two-threshold model.
- Support for random effects models is limited to random intercepts linear mixed models.

## 2 Types of threshold effects supported

This package supports 14 types of continuous two-phase models (one threshold), 1 type of continuous three-phase model (two thresholds), and 2 types of discontinuous (jump) two-phase models.

### 2.1 Continuous two-phase models

Continuous two-phase models are continuous at the threshold. They are also known as kink models or broken-stick models. The package support the following continuous two-phase models:

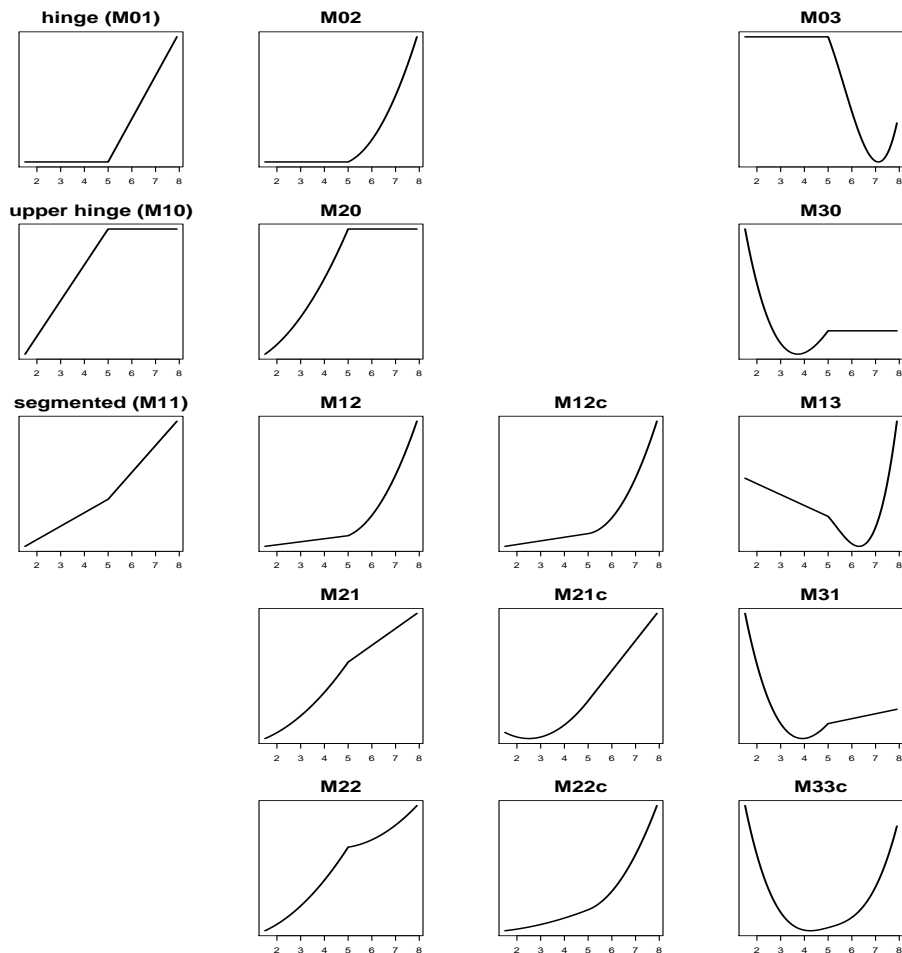


Figure 2.1: Types of continuous two-phase models supported in chngpt.

Piecewise linear two-phase models are studied in Fong et al. (2017b) and Elder and Fong (2019), two-phase polynomial models are studied in Son and Fong (2020). The two digits in the model names refer to the highest order of polynomials before and after the threshold, respectively. If the model name ends with 'c', the model is constrained and become smoother. The parameterization

are adopted in the package:

$$\eta = \alpha_1 + \alpha_2^T z + \beta_1 (x - e)_+ \quad (\text{hinge, M01})$$

$$\eta = \alpha_1 + \alpha_2^T z + \beta_1 (x - e)_+ + \beta_2 (x - e)_+^2 \quad (\text{M02})$$

$$\eta = \alpha_1 + \alpha_2^T z + \beta_1 (x - e)_+ + \beta_2 (x - e)_+^2 + \beta_3 (x - e)_+^3 \quad (\text{M03})$$

$$\eta = \alpha_1 + \alpha_2^T z + \beta_1 (x - e)_- \quad (\text{upper hinge, M10})$$

$$\eta = \alpha_1 + \alpha_2^T z + \beta_1 (x - e)_- + \beta_2 (x - e)_-^2 \quad (\text{M20})$$

$$\eta = \alpha_1 + \alpha_2^T z + \beta_1 (x - e)_- + \beta_2 (x - e)_-^2 + \beta_3 (x - e)_-^3 \quad (\text{M30})$$

$$\eta = \alpha_1 + \alpha_2^T z + \gamma x + \beta_1 (x - e)_+ \quad (\text{segmented, M11})$$

$$\eta = \alpha_1 + \alpha_2^T z + \gamma x + \beta_1 (x - e)_+ + \beta_2 (x - e)_+^2 \quad (\text{M12})$$

$$\eta = \alpha_1 + \alpha_2^T z + \gamma x + \beta_1 (x - e)_+ + \beta_2 (x - e)_+^2 + \beta_3 (x - e)_+^3 \quad (\text{M13})$$

$$\eta = \alpha_1 + \alpha_2^T z + \gamma x + \beta_1 (x - e)_- + \beta_2 (x - e)_-^2 \quad (\text{M21})$$

$$\eta = \alpha_1 + \alpha_2^T z + \gamma x + \beta_1 (x - e)_- + \beta_2 (x - e)_-^2 + \beta_3 (x - e)_-^3 \quad (\text{M31})$$

$$\eta = \alpha_1 + \alpha_2^T z + \beta_{1,-} (x - e)_- + \beta_{1,+} (x - e)_+ + \beta_{2,-} (x - e)_-^2 + \beta_{2,+} (x - e)_+^2 \quad (\text{M22})$$

$$\eta = \alpha_1 + \alpha_2^T z + \gamma x + \beta_{2,-} (x - e)_-^2 + \beta_{2,+} (x - e)_+^2 \quad (\text{M22c})$$

$$\eta = \alpha_1 + \alpha_2^T z + \gamma x + \beta_2 (x - e)^2 + \beta_{3,-} (x - e)_-^3 + \beta_{3,+} (x - e)_+^3 \quad (\text{M33c})$$

where  $e$  denote the threshold parameter,  $x$  is the predictor with threshold effect,  $z$  denote a vector of additional predictors, and  $(x - e)_+ = x - e$  if  $x > e$  and 0 otherwise, and  $(x - e)_- = x - e$  if  $x \leq e$  and 0 otherwise.

## 2.2 Discontinuous two-phase models

The following discontinuous two-phase models are supported in the *chngp* package:

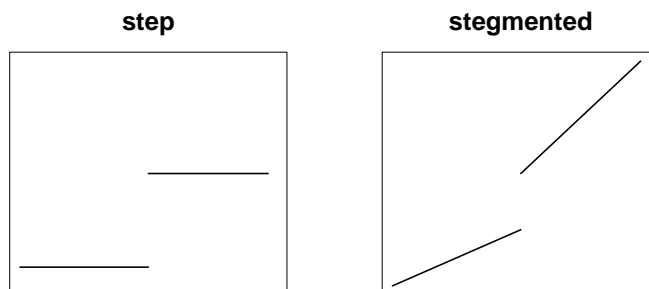


Figure 2.2: Types of discontinuous threshold effects supported in *chngp*.

The models can be written as

$$\eta = \alpha_1 + \alpha_2^T z + \beta_1 I(x > e) \quad (\text{step})$$

$$\eta = \alpha_1 + \alpha_2^T z + \beta_1 (x - e)_+ + \gamma x + \beta_2 I(x > e), \quad (\text{stegmented})$$

where  $e$  denote the threshold parameter,  $x$  is the predictor with threshold effect,  $z$  denote a vector of additional predictors, and

$$I(x > e) = \begin{cases} 1 & \text{if } x > e \\ 0 & \text{if otherwise} \end{cases} .$$

## 2.3 Continuous three-phase models

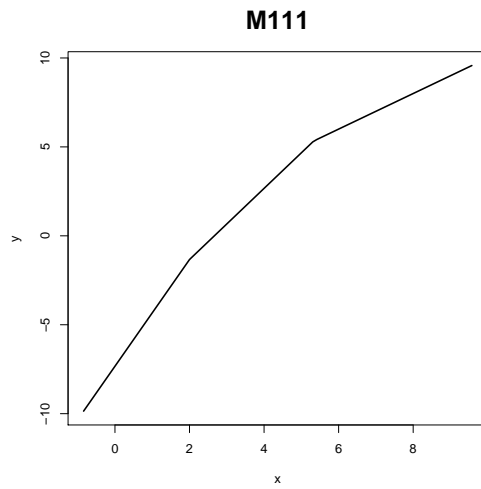


Figure 2.3: A three-phase segmented model.

$$\eta = \alpha_1 + \alpha_2^T z + \beta_1 (x - e)_- + \beta_2 (x - f)_- + \beta_3 x \quad (\text{M111})$$

### 3 Examples - estimation for fixed effects models

Some general notes:

- The fitted model has a component named `best.fit`, which is the model fit conditional on the estimated threshold parameter.
- The recommended `ci.bootstrap.size` is 1000 in real problems.
- P values are not provided for the threshold estimates because it does not make sense to make it a default null hypothesis that the threshold parameter is 0.
- Wild, sieve, and wild sieve bootstrap methods are implemented for time series data through the *bootstrap.type* argument.

### 3.1 Continuous two-phase linear regression

For continuous two-phase linear regression, we have developed a grid search method for estimation that is super fast (Fong, 2019; Elder and Fong, 2019). Together with the observation that bootstrap confidence intervals have better coverage than robust analytical confidence intervals (Fong et al., 2017b) for continuous two-phase linear regression, internally we set the default estimation method to be fast grid search and the default variance method to be bootstrap.

#### 3.1.1 Segmented model

To fit a segmented linear regression model, we call

```
fit=chngptm(formula.1=V3_BioV3B~1, formula.2=~NAb_score, dat.mtct.2, type="segmented", family="gaussian")
summary(fit)
```

---

Change point model type: segmented

Coefficients:

	est	p.value*	(lower	upper)
(Intercept)	-22.33152	1.593423e-08	-30.07675	-14.58628
NAb_score	67.23925	2.212981e-14	49.98398	84.49452
(NAb_score-chngp <sub>t</sub> ) <sub>+</sub>	-64.83129	3.692679e-14	-81.61413	-48.04845

Threshold:

est	(lower	upper)
0.4653923	0.4535000	0.4772845

In the output above, the row starting with (NAb\_score-chngp<sub>t</sub>)<sub>+</sub> corresponds to  $\beta_1$  in equation (segmented, M11). In other words, it is the change in slope as the covariate NAb\_score crosses the threshold. Note that there is an asterisk next to p.value. This is because bootstrap procedures to generate confidence intervals do not readily lead to p values. The presented p values are approximations, obtained assuming that the bootstrap sampling distributions are normal.

To get an estimate of the slope after threshold, we call

```
lincomb(fit, comb=c(0,1,1), alpha=0.05)
```

---

est	lb	ub
2.40795883	-0.06780353	4.88372120

To perform a likelihood ratio test, we call

```
library(lmtest)
fit.0=lm(V3_BioV3B~1, dat.mtct.2)
lrtest(fit, fit.0)
```

---

Likelihood ratio test

```
Model 1: V3_BioV3B ~NAb_score + x.mod.e
Model 2: V3_BioV3B ~1
#Df LogLik Df Chisq Pr(>Chisq)
1 5 -354.95
```

```
2 2 -431.50 -3 153.1 < 2.2e-16 ***
```

```
---
```

```
Signif. codes: 0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Calling `plot(fit)` makes the following figure.

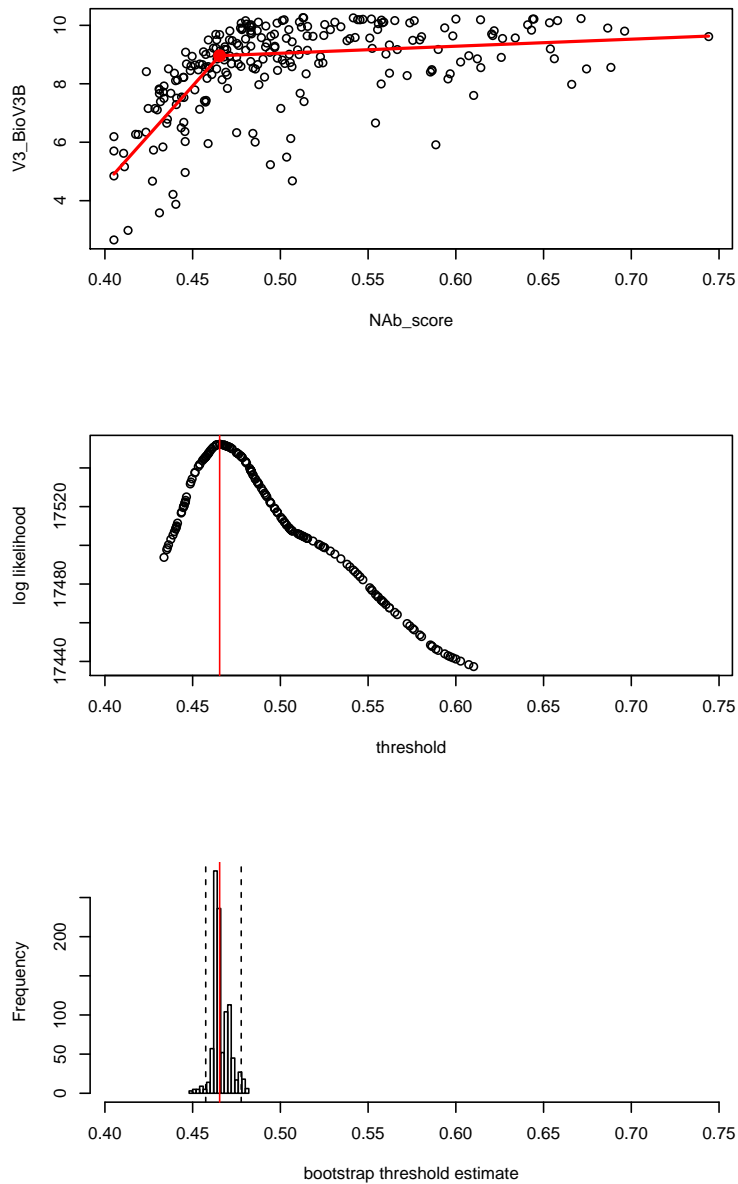


Figure 3.1: Scatterplot, profile likelihood plot, and bootstrap distribution of threshold estimates.



### 3.1.2 Other models

To specify the types of threshold effects, use the *type* argument. For example,

```
fit=chngptm(formula.1=pressure~1, formula.2=~temperature, data=pressure,
  type="M02", family="gaussian", var.type="bootstrap")
summary(fit)
```

---

Change point model threshold.type: M02

Coefficients:

	est	p.value*	(lower	upper)
(Intercept)	8.278463507	0.4733673	-14.35129837	30.9082254
(temperature-chngpt)+	0.007124705	0.9944183	-2.00325636	1.9890069
I((temperature-chngpt)+^2)	0.039305656	0.3644561	-0.04564143	0.1242527

Threshold:

	est	Std. Error	(lower	upper)
	220.00000	20.40816	160.00000	240.00000

Suppose the samples are autocorrelated and/or heteroscedastic, the *bootstrap.type* argument can be set. For example,

```
dat=sim.chngpt(mean.model="thresholded", threshold.type="M20", n=100, seed=1, mu.x=5, beta=c(10,1),
  x.distr="lin", e.=5, family="gaussian", alpha=0, sd=3, coef.z=log(1.4), heteroscedastic=FALSE, ar=.5)
fit=chngptm(y~z, ~x, type="M20", data=dat, family="gaussian", est.method="fastgrid",
  var.type="bootstrap", bootstrap.type="wildsieve")
summary(fit)
```

---

Change point model threshold.type: M20

Coefficients:

	est	Std. Error*	(lower	upper)	p.value*
(Intercept)	0.8548963	0.6553686	-0.3449894	2.2240556	1.920798e-01
z	0.1641522	0.2262669	-0.2530587	0.6339074	4.681572e-01
(x-chngpt)-	7.3787555	1.7589341	3.4668998	10.3619215	2.728531e-05
(x-chngpt)-^2	0.4412215	0.4086794	-0.2973006	1.3047227	2.803081e-01

Threshold:

	est	Std. Error	(lower	upper)
	5.6787879	0.3793032	5.1616162	6.6484848

## 3.2 Continuous two-phase logistic regression

For continuous two-phase logistic regression, a fast grid search method for estimation is not yet available. In addition, we have observed that bootstrap confidence intervals have similar coverage as robust analytical confidence intervals (Fong et al., 2017b). Thus, we recommend either `var.type="bootstrap"` or `var.type="robust"` in the call to `chngpptm`. Note that when it is set to *robust*, an auxiliary fit needs to be supplied, which is generally a smooth parametric model with enough but not too many degrees of freedom.

To estimate a hinge logistic regression model, we call

```
library(splines)
fit=chngpptm(formula.1=y~birth, formula.2=~NAb_SF162LS, dat.mtct,
  type="hinge", family="binomial",
  est.method="smoothapprox", var.type="robust",
  aux.fit=glm(y~birth + ns(NAb_SF162LS,3), dat.mtct, family="binomial"))
summary(fit)
```

---

Change point model type: hinge

Coefficients:

	OR	p.value	(lower	upper)
(Intercept)	0.7026523	0.341429662	0.3388366	1.4571044
birthVaginal	1.2397649	0.523159883	0.6393632	2.4039809
(NAb_SF162LS-chngppt)+	0.6712371	0.001332547	0.5270730	0.8548327

Threshold:

	26.3%	(lower	upper)
	7.373374	5.472271	8.186464

The `chngpptm` function supports the use of `cbind` in the formula, as the `glm` function does. For example,

```
dat.2=sim.chngppt("thresholded", "step", n=200, seed=1, beta=1, alpha=-1,
  x.distr="norm", e.=4, family="binomial")
dat.2$success=rbinom(nrow(dat.2), 10, 1/(1 + exp(-dat.2$eta)))
dat.2$failure=10-dat.2$success
fit.2a=chngpptm(formula.1=cbind(success,failure)~z, formula.2=~x,
  family="binomial", dat.2, type="step")
```

Getting bootstrap confidence intervals can take some time, but parallel processing is supported on Linux machines. For example,

```
system.time(chngpptm(formula.1=y~birth, formula.2=~NAb_SF162LS, dat.mtct, type="hinge", family="binomial",
  est.method="smoothapprox", var.type="bootstrap"))
system.time(chngpptm(formula.1=y~birth, formula.2=~NAb_SF162LS, dat.mtct, type="hinge", family="binomial",
  est.method="smoothapprox", var.type="bootstrap", ncpus=10))
```

---

```
user system elapsed
20.057 0.141 20.218
```

```
user system elapsed
19.500 1.256 2.673
```

### 3.3 Continuous two-phase Poisson regression

Only grid search method and bootstrap confidence intervals are supported, so getting the model fit with confidence intervals could take some time. If run on Linux machines, setting `ncpus` to the number of cores available can speed things up by `ncpus` fold.

```
counts <- c(18,17,15,20,10,20,25,13,12)
outcome <- as.integer(gl(3,1,9))
treatment <- gl(3,3)
print(d.AD <- data.frame(treatment, outcome, counts))
fit.4=chngptm(formula.1=counts ~treatment, formula.2=~outcome, data=d.AD,
  family="poisson", type="segmented", var.type="bootstrap")
summary(fit.4)
```

### 3.4 Discontinuous two-phase GLM

Confidence interval for discontinuous threshold regression models can be constructed by m-out-of-n bootstrap.

```
fit=chngptm(formula.1=mpg~hp, formula.2=~drat, mtcars, type="step",
  family="gaussian", var.type="bootstrap", ci.bootstrap.size=100, m.out.of.n=20)
summary(fit)
```

---

Change point model threshold.type: step

Coefficients:

	est	Std. Error*	(lower	upper)	p.value*
(Intercept)	27.29298302	2.89102342	21.62657712	32.95938892	3.706663e-21
hp	-0.05692654	0.01644498	-0.08915870	-0.02469439	5.369001e-04
drat>chngpt	5.24824935	2.72504835	-0.09284542	10.58934411	5.411325e-02

Threshold:

	est	Std. Error	(lower	upper)	p.value
	3.9200000	0.4693878	3.0000000	4.8400000	NA

### 3.5 Two-phase Cox regression

The *chnppt* package also provides some support for estimation of threshold Cox regression models. What is missing, though, is confidence intervals for parameter estimates and hypothesis testing methods. See the help page on *chnppt* for an example.

### 3.6 Two-phase models with interaction terms

In the following example we fit a model with an interaction term.

$$\eta = \beta_1 + \beta_2 z + \beta_3 x + \beta_4 (x - e)_+ + \beta_5 z x + \beta_6 z (x - e)_+$$

```
fit=chngptm(formula.1=mpg ~hp, formula.2=~hp*drat, mtcars, type="segmented",
  family="gaussian", var.type="bootstrap", ci.bootstrap.size=100)
summary(fit)
```

---

Change point model threshold.type: segmented

Coefficients:

	est	Std. Error*	(lower	upper)	p.value*
(Intercept)	71.0423961	107.7931740	-140.2322250	282.3170173	0.5098559
hp	-0.5714405	0.7521618	-2.0456777	0.9027967	0.4474155
drat	-14.3708279	35.7034558	-84.3496013	55.6079456	0.6873122
(drat-chngpt)+	21.6073593	73.6732299	-122.7921714	166.0068899	0.7693032
hp:drat	0.1658607	0.2482010	-0.3206132	0.6523346	0.5039730
hp:(drat-chngpt)+	-0.1970979	0.5108437	-1.1983515	0.8041557	0.6996239

Threshold:

est	Std. Error	(lower	upper)	p.value
3.2300000	0.4489796	2.3500000	4.1100000	NA

In the following example we fit a model with two interaction terms

$$\eta = \beta_1 + \beta_2 z_1 + \beta_3 z_2 + \beta_4 I(x > e) + \beta_5 z_1 I(x > e) + \beta_6 z_2 I(x > e)$$

```
fit=chngptm(formula.1=mpg~hp+wt, formula.2=~hp*drat+wt*drat, mtcars, type="step",
  family="gaussian", var.type="bootstrap", ci.bootstrap.size=100)
summary(fit)
```

---

Change point model threshold.type: step

Coefficients:

	est	Std. Error*	(lower	upper)	p.value*
(Intercept)	30.83332346	4.06186261	22.87207274	38.79457417	3.176122e-14
hp	-0.02389962	0.02760935	-0.07801395	0.03021471	3.866903e-01
wt	-2.58756410	1.17757075	-4.89560276	-0.27952543	2.799370e-02
drat>chngpt	11.69827186	28.02745000	-43.23553015	66.63207386	6.763959e-01
hp:I(drat>chngpt)	-0.00894615	0.20736123	-0.41537415	0.39748185	9.655877e-01
wt:I(drat>chngpt)	-3.22148003	21.48073350	-45.32371769	38.88075762	8.807878e-01

Threshold:

est	Std. Error	(lower	upper)	p.value
3.7000000	0.2806122	3.1500000	4.2500000	NA

### 3.7 Continuous three-phase linear regression

The following code fits a three-phase linear regression model. The default estimation method is *fastgrid* and the default variance type is *bootstrap*.

$$\eta = \beta_1 + \beta_2 z + \beta_3 x + \beta_4 (x - e)_+ + \beta_5 z x + \beta_6 z (x - e)_+$$

```
fit=chngptm (formula.1=pressure~1, formula.2=~temperature, pressure, type="M111",
  family="gaussian", ci.bootstrap.size=20)
summary(fit)
```

---

Change point model threshold.type: M111

Coefficients:

	est	Std. Error*	(lower	upper)	p.value*
(Intercept)	-3310.976868	1006.428099	-5283.575941	-1338.3777950	1.002481e-03
temperature	11.417859	3.015001	5.508457	17.3272614	1.524670e-04
(temperature-chngp <sub>t1</sub> )-	-3.862734	1.612508	-7.023249	-0.7022192	1.659850e-02
(temperature-chngp <sub>t2</sub> )-	-7.425005	1.700649	-10.758278	-4.0917324	1.265528e-05

Threshold:

	est	Std. Error	(lower	upper)	p.value
chngp <sub>t.1</sub>	240	30.61224	180	300	NA
chngp <sub>t.2</sub>	320	31.12245	259	381	NA

## 4 Examples - estimation for random effects models

### 4.1 Continuous two-phase linear regression with random intercepts

The following code fits the linear mixed model:

$$Y = a + \alpha^T z + \gamma x + \beta (x - e)_+ + \epsilon$$
$$a \sim N(0, \sigma_a)$$
$$\epsilon \sim N(0, \sigma_\epsilon)$$

Variance estimates are being developed.

```
dat=sim.twophase.ran.inte(threshold.type="segmented", n=50, seed=1)
fit = chngptm (formula.1=y~z+(1|id), formula.2=~x, family="gaussian", dat,
  type="segmented", est.method="grid", var.type="none")
summary(fit)
plot(fit, which=1, plot.individual.line=T, lcol="gray", lwd=.5)
```

---

No variance estimate available.

(Intercept)	z	x (x-chngpt)+	chngpt	
2.7154145	0.3514853	1.7894006	2.5695986	5.1571429

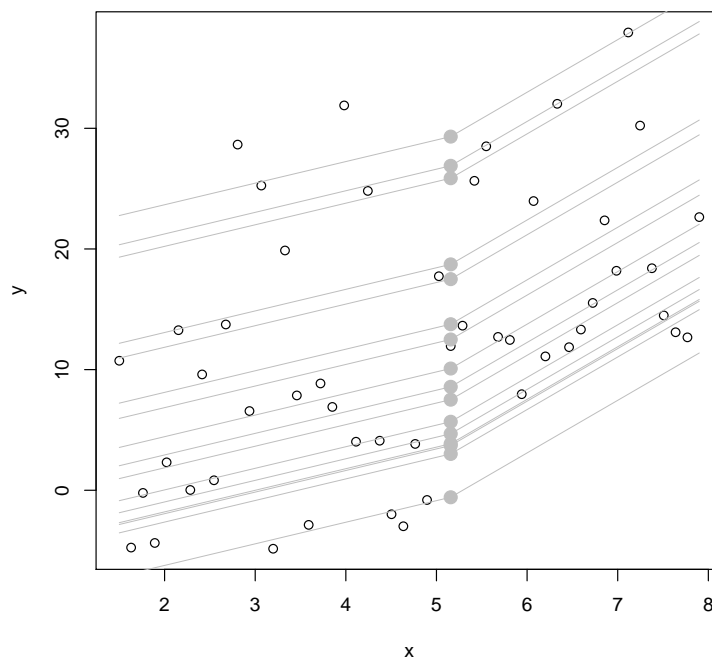


Figure 4.1: Each line corresponds to one id.

The code above also works if we replace segmented with other models, e.g. M20.



## 5 Examples - testing for independent data

An example in linear regression:

```
test=chngpt.test(formula.null=Volume~1, formula.chngpt=~Girth, trees,
  type="segmented", family="gaussian")
test
```

---

Maximum of Likelihood Ratio Statistics

```
data: trees
Maximal statistic = 17.694, change point = 15.388, p-value = 0.00014
alternative hypothesis: two-sided
```

The first line gives the type of test carried out, and it is maximal likelihood ratio test here, which is the default. In addition, a plot function can be called on the test object to show the score or likelihood ratio statistic as a function of candidate change points.

An example in logistic regression:

```
test=chngpt.test(formula.null=y~birth, formula.chngpt=~NAb_SF162LS, dat.mtct,
  type="hinge", family="binomial", main.method="score")
test
```

---

Maximum of Score Statistics

```
data: dat.mtct
Maximal statistic = 3.3209, change point = 7.0347, p-value = 0.00284
alternative hypothesis: two-sided
```

The first line gives the type of test carried out, and it may be maximal likelihood ratio test. In addition, a plot function can be called on the test object to show the score or likelihood ratio statistic as a function of candidate change points.

## 6 Further considerations

### 6.1 Model choice

- The choice of threshold effects is typically through a combination of domain knowledge and modeling. One modeling approach is to first examine the relationship using local polynomial regression. A nice tool for that is the R package `mgcv`, which provides automatic smoothness estimation.
- To choose among the segmented, hinge, and upper hinge models formally, we can use Wald tests. For example, if the question is framed as choosing between segmented and hinge models, we can fit a segmented model and then look at the slope before threshold in the summary function output. If the estimate is not significantly different from 0, then it is justifiable to fit a hinge model. We can also look at the slope after threshold, which is not displayed as part of the summary function output, but can be obtained by calling `lincomb` (see example in Section 3.1.1). If this estimate is not significantly different from 0, then it is justifiable to fit an upper hinge model.
  - If the hinge or upper hinge model is reasonable, it is preferred over the segmented model because the model can be estimated with substantially higher precision (Fong et al., 2017b; Elder and Fong, 2019).

### 6.2 Estimation methods

There are three types of search methods for finding the maximum likelihood estimator. Users generally do not need to worry about setting the argument, which is `est.method`, since the function chooses the most appropriate one by default. In the order of development, the three search methods are:

- grid search. The grid method is the most flexible, but also the most time-consuming.
- smooth approximation. The smooth approximation method (Fong et al., 2017a) involves approximating the likelihood function with a differentiable function to allow gradient-based search; it is recommended when grid search is too slow and fast grid search is not available.
- fast grid. This is a new type of methods (Fong, 2019; Elder and Fong, 2019; Son and Fong, 2020) that are super fast and gives exact solutions. The only downside is that it is only available for linear regression and independent data.

### 6.3 Confidence interval methods

- We recommend bootstrap confidence interval methods for all models. For linear models, this can be done very fast with fast grid search. For other models, this can take some time, but parallel processing with multiple cores help speed things up.
- Robust analytical confidence interval methods were developed in Fong et al. (2017b). The disadvantage of this method is that it needs an auxiliary model fit.

## 6.4 Hypothesis testing

Hypothesis testing methods that are implemented in this package are described in Fong et al. (2017a) and Fong et al. (2015).

## References

- Elder, A. and Fong, Y. (2019), “Estimation and Inference for Upper Hinge Regression Models,” *Environmental and Ecological Statistics*, 26, 287–302.
- Fong, Y. (2019), “Fast Bootstrap Confidence Intervals for Continuous Threshold Linear Regression,” *Journal of Computational and Graphical Statistics*, 28, 466–470.
- Fong, Y., Di, C. and Permar, S. (2015), “Change point testing in logistic regression models with interaction term,” *Statistics in medicine*, 34, 1483–1494.
- Fong, Y., Huang, Y., Gilbert, P. and Permar, S. (2017a), “chngp: threshold regression model estimation and inference,” *BMC Bioinformatics*, 18, 454–460.
- Fong, Y., Chong, D., Huang, Y. and Gilbert, P. (2017b), “Model-robust Inference for Continuous Threshold Regression Models,” *Biometrics*, 73, 452–462.
- Muggeo, V.M. (2008), “Segmented: an R package to fit regression models with broken-line relationships,” *R news*, 8, 20–25.
- Son, H. and Fong, Y. (2020), “Fast Grid Search and Bootstrap-based Inference for Continuous Two-phase Polynomial Regression Models,” *Environmetrics*, in press.