Picewise and segmented regression with R

Juan R Gonzalez

Contents

1 Introduction **1**

2 Picewise regression **1**

3 Estimating procedures **3**
3.1 Iterative searching 4

4 Segmented regression **6**

5 Exercise (to deliver) **8**

6 References **9**

7 Session information **9**

1 Introduction

Objectives

- Understand the concept of picewise and segmented regression
- Learn how to perform picewise and segmented regression with R
- Peform data analyses where the scientific question is to determine changes in the linear relationship of two continuous variables

2 Picewise regression

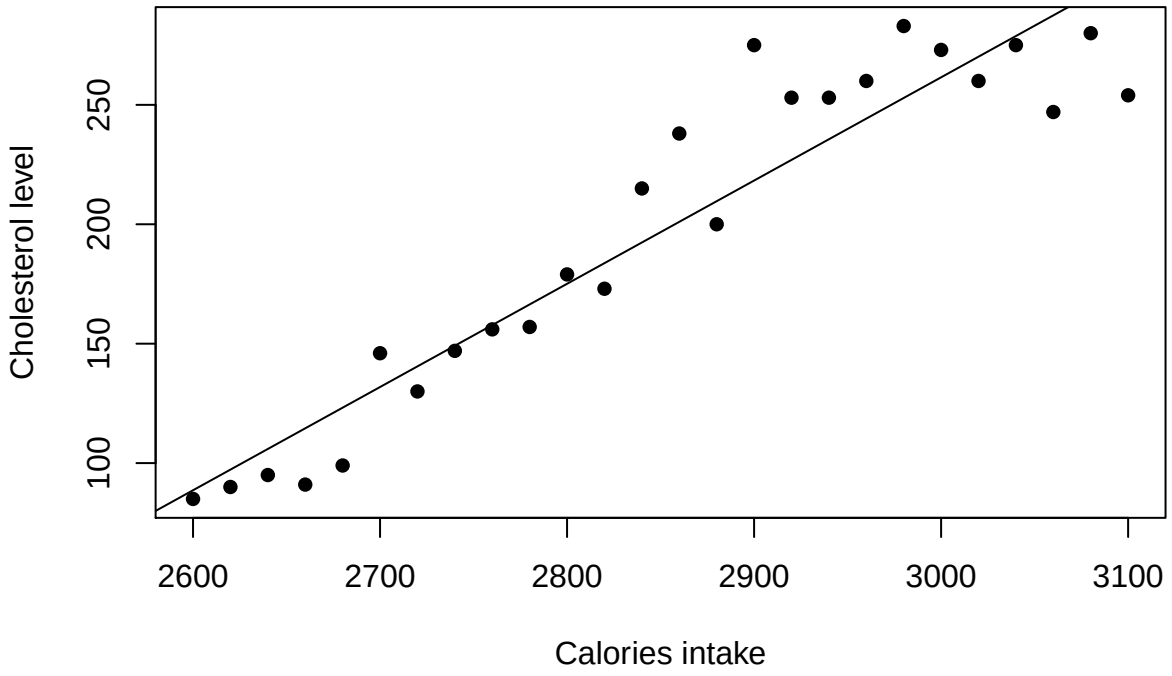
Piecewise regression comes about when you have 'breakpoints', where there are clearly two different linear relationships in the data with a sudden, sharp change in directionality. This crops up occasionally in biomedicine when dealing with, for example, cholesterol level and fat or calories intake. There is initially a rapid incress of cholesterol level as fat or calories consumption increases (cholesterol incressing becomes limiting at certain levels).

If you Google 'R piecewise regression', you may get a variety of methods and advice on how to run a piecewise regression. Essentially, you can do it manually or use different R packages to run the regression. Herein, we will review two methods: brute force iterative approaches (as in 'The R Book' by Crawley that can help to better understand the process of how to fit the models) and the `segmented` package. Using these approaches allows you to statistically estimate the breakpoint, which is better than just eyeballing it and fitting two models around what you think is the breakpoint. As we always comment on class, let statistics do the work for you objectively.

Let us start by illustrating the need for using a piecewise approach to our linear regression model. Consider the following plot of the calories intake and the cholesterol level:

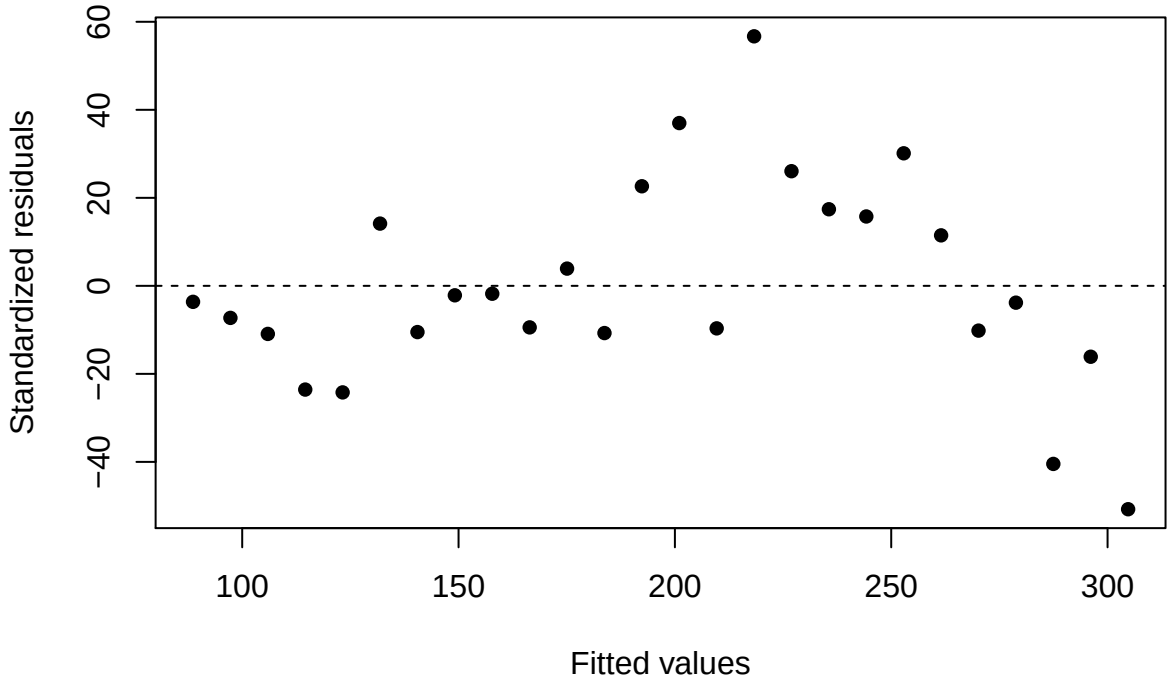
```
cholesterol <- read.delim("../data/cholesterol.txt")
plot(cholesterol$calories, cholesterol$cholesterol, xlab="Calories intake",
     ylab="Cholesterol level", pch=16)
```

```
lin.mod <- lm(cholesterol~calories, data=cholesterol)
abline(lin.mod)
```



The estimated regression line appears to fit the data fairly well in some overall sense, but it is clear that we could do better. The residuals versus fits plot also indicates that linear model is not fine:

```
plot(lin.mod$fitted.values, lin.mod$residuals, xlab="Fitted values",
     ylab="Standardized residuals", pch=16)
abline(h=0, lty=2)
```



We could instead split our original scatter plot into two pieces considereing calories intake above and below 2950 aproximately, but connected lines, one for each piece. As you can see, the estimated two-piece function, connected at those points (the dashed line) appears to do a much better job of describing the trend in the data. So, let's formulate a piecewise linear regression model for our data

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 (x_{i1} - 2950) x_{i2} + \epsilon_i$$

Alternatively, we could write our formulated piecewise model as:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2}^* + \epsilon_i$$

where:

- y_i is the cholesterol level of individual i
- x_{i1} is the calories intake of individual i
- x_{i2} is a dummy variable $(0, \text{ if } x_{i1} \leq 2950 \text{ and } 1, \text{ if } x_{i1} > 2950)$ of individual i
- x_{i2}^* denotes the $(x_{i1} - 2700)x_{i2}$ the interaction term

and the independent error terms ϵ_i follow a normal distribution with mean 0 and equal variance σ^2. The model can be estimated using different methods. Let us describre two of them.

3 Estimating procedures

3.1 Iterative searching

For illustrating purposes and for the sake of the simplicity, let us illustrate how to estimate the first breakpoint. The key point of the iterative search procedure described by Crawley is choosing the breakpoints. In this case, we can eyeball the data and say that the second breakpoint is somewhere between 2900 and 3000. Choose a wider range than you might think, just to be safe. Create a variable called breaks to hold these breakpoints:

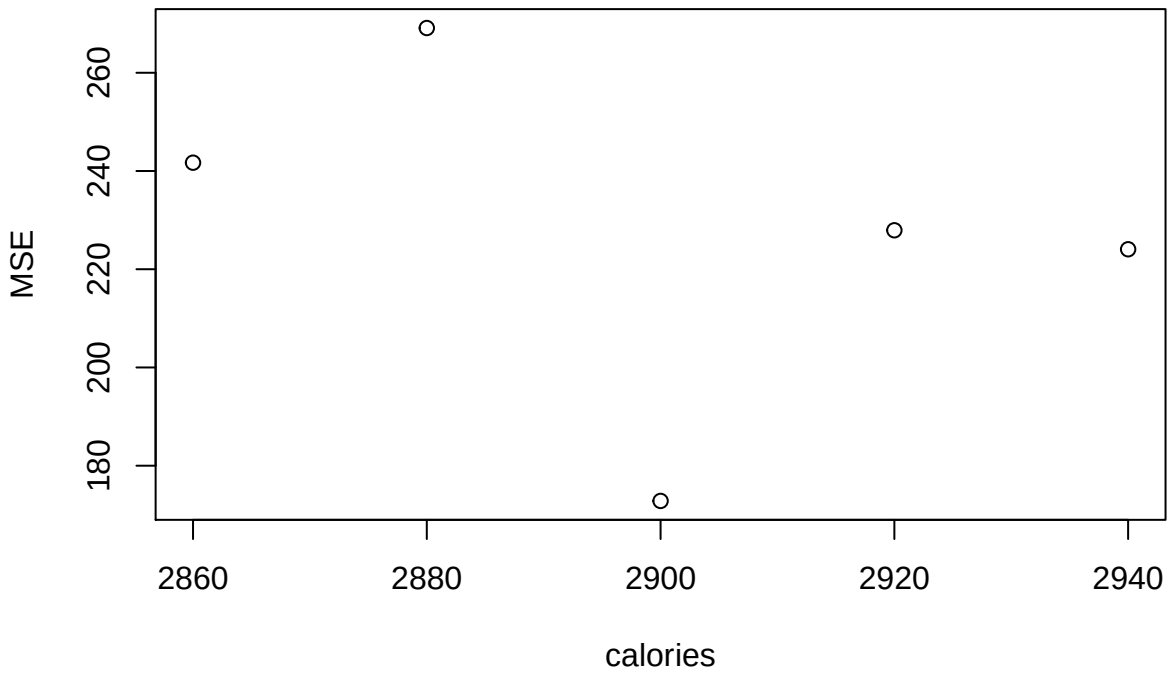
```
breaks <- with(cholesterol, calories[which(calories >= 2850 & calories <= 2950)])
```

Now we're going to iteratively search these breakpoints for the model that has the lowest residual MSE, using that as our criteria for the best model. Create an empty container for MSE values from each model, and use an iteration to run a linear regression for each possible breakpoint. Formulate the linear model exactly like the above formula.

```
mse <- numeric(length(breaks))
for(i in 1:length(breaks)){
 piece.mod <- lm(cholesterol ~ calories*(calories < breaks[i]) +
                   calories*(calories>=breaks[i]), data=cholesterol)
 mse[i] <- mean(piece.mod$residuals^2)
}
mse
## [1] 241.6946 269.1082 172.8291 227.9307 224.0628
```

If we plot MSE by breakpoints, we can visually estimate the breakpoint as the lowest point on the curve:

```
plot(breaks, mse, xlab="calories", ylab="MSE")
```



As expected the optimal break is 2900.

```
breaks[which(mse==min(mse))]
## [1] 2900
```

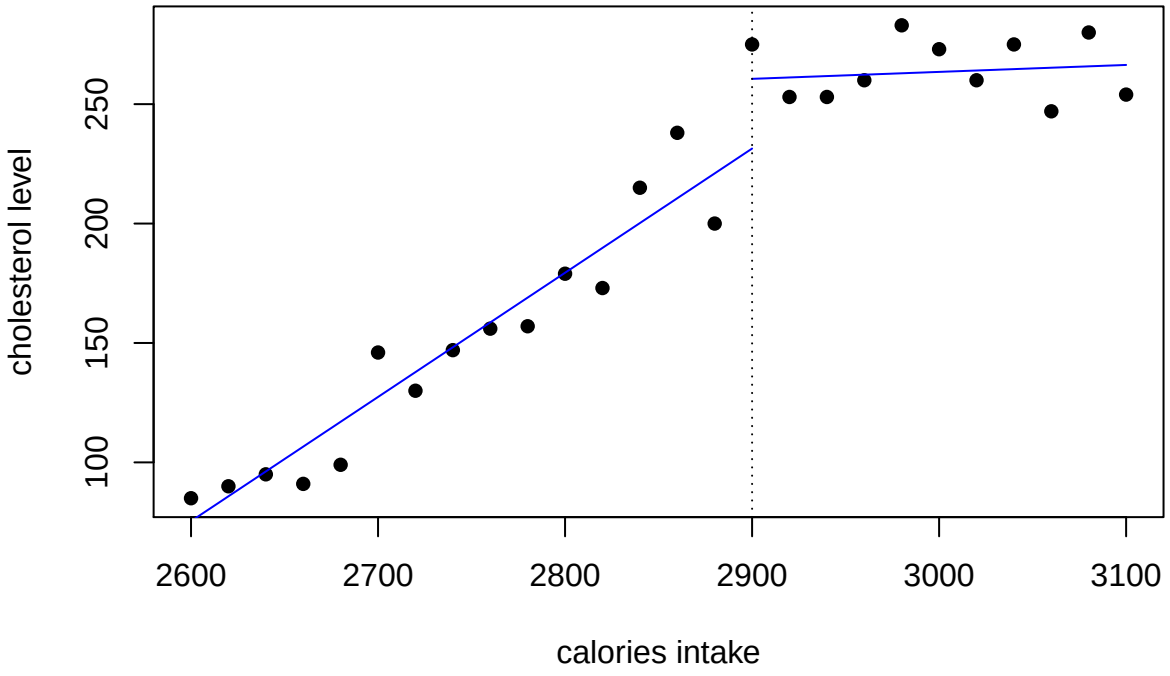
The picewise regression model is then estimated by executing:

```
piece.mod2 <- lm(cholesterol ~ calories*(calories < 2900)
                 + calories*(calories > 2900), data=cholesterol)
summary(piece.mod2)
##
## Call:
## lm(formula = cholesterol ~ calories * (calories < 2900) + calories *
##     (calories > 2900), data = cholesterol)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.4583 -10.0748  -0.5613  10.1182  28.9310
##
## Coefficients: (1 not defined because of singularities)
##                                Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   1.906e+02  2.303e+02   0.828    0.417
## calories                      2.909e-02  7.925e-02   0.367    0.717
## calories < 2900TRUE          -1.467e+03  2.587e+02  -5.671 1.25e-05 ***
## calories > 2900TRUE          -1.440e+01  1.743e+01  -0.826    0.418
## calories:calories < 2900TRUE  4.904e-01  9.017e-02   5.438 2.15e-05 ***
## calories:calories > 2900TRUE         NA         NA      NA       NA
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 14.4 on 21 degrees of freedom
## Multiple R-squared:  0.9647, Adjusted R-squared:  0.9579
## F-statistic: 143.4 on 4 and 21 DF,  p-value: 6.327e-15
```

The NA values arise from singularities and can be omitted from the interpretation. The intercept for the line when calories intake < 2900 is (Intercept) + calories $<$ 2900TRUE, or -1276.6. The slope of the line when calories intake < 2900 is calories + calories:calories $<$ 2900TRUE, or 0.52. So, when x is less than 2900, the formula is -1276.6 + 0.52*calories. For the second segmente (e.g intake > 2900), the intercept is (Intercept) + calories $>$ 2900TRUE, or 176.24 and the slope is just the variable calories, or 0.0291.

The predictive model can be visualy represented by executing:

```
with(cholesterol, plot(calories, cholesterol, pch=16,
                       xlab="calories intake", ylab="cholesterol level"))
curve( -1276.6 + 0.52*x, add=T, from=2600, to=2900, col="blue")
curve(176.2 + 0.0291*x, add=T, from=2900, to=3100, col="blue")
abline(v=2900, lty=3)
```



Notice that the segments were not constrained to *be touching* or continuous. This is inherent in the algorithm that we used. The next method will address this problem.

4 Segmented regression

The procedure of *segmented regression* uses maximum likelihood to fit a somewhat different parameterization of the model:

$$y \sim \beta_1 x + \beta_2 (x - c) + \gamma I(x > c)$$

$I(x > c)$ is a dummy variable as above, so when $x < c$, the model is essentially:

$$y \sim \beta_1 x + \beta_2 (x - c)$$

The γ term is simply a measure of the distance between the end of the first segment and the beginning of the next. The model converges when γ is minimized, thus this method constrains the segments to be (nearly) continuous. This is a major difference from the iterative approach in Method above.

This approach is implemented in the `segmented` Rpackage. To use this method, you first fit a generic linear model. You then use the `segmented( )` function to fit the piecewise regression. The `segmented( )` function takes for its arguments the generic linear model, `seg.Z` which is a one sided formula describing the predictor with a segment (we only have one predictor, `x`, which has the segment), and `psi`, which is a starting value of the breakpoint (as in other estimating methods you need to supply a best-guess estimate ot that parameter - in other words, the point you think the breakpoint can be located). More complicated models are a bit more complicated in terms of arguments, but this is a good starting example.

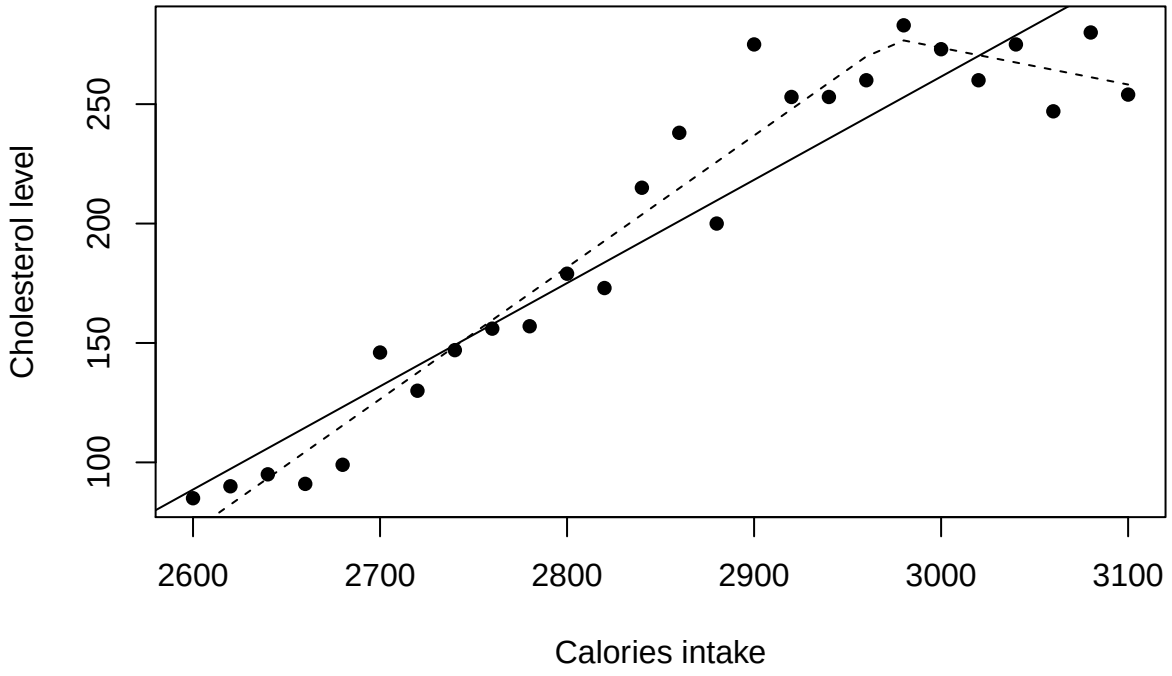
In our case, x is the predictor with a segment (it is the only predictor) and based on the first scatterplot (the first graph on the page). One might guess that the breakpoint is 2900. Therefore, the model is fitted by executing:

```
library(segmented)
lin.mod <- lm(cholesterol ~ calories, data=cholesterol)
segmented.mod <- segmented(lin.mod, seg.Z = ~ calories, psi=2900)
segmented.mod
## Call: segmented.lm(obj = lin.mod, seg.Z = ~calories, psi = 2900)
##
## Meaningful coefficients of the linear terms:
## (Intercept)     calories  U1.calories
##  -1363.6077       0.5519      -0.7058
##
## Estimated Break-Point(s):
## psi1.calories
##          2974
```

Notice that in that case the point were the line is changing is `segmented.mod$psi[2]` which is different from the one we obainted by using iterative seraching. This will be further investigate in the Exercises you have to deliver. In this case the figure illustrates (dashed lines) that this method better fits our data. IMPORTAN NOTE: `U1.x` is not the slope of the second segment. It is the difference in slopes between the second and first segment. So if your coefficients are calories = 0.5519 and U1.calories = -0.70578, then the slope of the second segment is 0.5519–0.7058=-0.1539 as you can verify it by executing

```
slope(segmented.mod)
## $calories
##           Est. St.Err. t value CI(95%).l CI(95%).u
## slope1  0.5519 0.03329  16.580    0.4828    0.6209
## slope2 -0.1539 0.15020  -1.025   -0.4654    0.1576
```

```
plot(cholesterol$calories, cholesterol$cholesterol, xlab="Calories intake",
     ylab="Cholesterol level", pch=16)
lin.mod <- lm(cholesterol~calories, data=cholesterol)
abline(lin.mod)
lines(cholesterol$calories, predict(segmented.mod), lty=2)
```



Further information about the `segmented regression` can be found in this paper Estimating regression models with unknown break-points (this is a hyperlink to the paper that is freely available at Statistics in Medicine).

5 Exercise (to deliver)

Data for exercises are in the repository https://github.com/isglobal-brge/TeachingMaterials/tree/master/Longitudinal_data_analysis/data

Exercise 1: Read the paper Estimating regression models with unknown break-points and answer these questions:

- Can this method to detect the existence or not of a change point?
- What are the main limitations of this method in practice? Enumerate a couple of them
- Figure 4 (bottom left) there are a longitudinal data where x-axis represents age and y-axis stands for the logit of having bronquitis. How many changes would you estimate to evaluate the relationship between the years of exposition and the probability of developing bronquitis? How many does the autor test? How may does he consider should be analyzed? Why does he conclude that (read pages 3067 and 3068).

Exercise 2: By using the `cholesterol` dataset I have used in this material (that is available at the folder of data for exercises) repeat the iterative searching procedure to better estimate the point of change. Explain what have you change in the code and why.

Exercise 3: The funcion `slope` in the `segmented` package contains an argument that is called `APC` that can be used for estimating the annual percentage change of the segments. Use data the data available at `mamaCat.txt` to perform a joinpoint regression analysis by using `segmented` regression and estimate the APC of each segment. Remember that the data contains breast cancer mortality of females in Catalonia of the period 1975-1997. Each column contains the next

variables: gender, year of mortality, number of deaths and at-risk population. NOTE: investigate whether `segmented` regression allows the use of different models that `lm`.

Exercise 4: [**Segmented regression adjusted by other covariates**]. Researchers have performed an experiment by collecting data of plants in three different organs accross time. They are interested in determining the time when these organs stop growing (or even become small). NOTE: this is important since it implies that they are looking for a single change point - observe that researches have their own scientific question and you must know how to translate it into a statistical one. Data are available in the obejct `plant` that is in the `segmented`libray (you can load them by using *data(plant)*). The organ is in the variable `group`. Perform the statistical analysis that is required to address researchers' question. Do not forget to create a plot tu visually evaluate the evolution accross time for each type of plant. Write up a little conclusion about the study (3-4 lines as much).

---

6 References

- Segmented regression paper
- The segmented package

7 Session information

```
## R version 3.3.2 (2016-10-31)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 14393)
##
## locale:
## [1] LC_COLLATE=Spanish_Spain.1252  LC_CTYPE=Spanish_Spain.1252
## [3] LC_MONETARY=Spanish_Spain.1252 LC_NUMERIC=C
## [5] LC_TIME=Spanish_Spain.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
## [1] segmented_0.5-1.4 knitr_1.15.1      BiocStyle_2.2.1
##
## loaded via a namespace (and not attached):
##  [1] backports_1.0.5 magrittr_1.5    rprojroot_1.2   tools_3.3.2
##  [5] htmltools_0.3.5 yaml_2.1.14     Rcpp_0.12.9     stringi_1.1.2
##  [9] rmarkdown_1.3   stringr_1.2.0   digest_0.6.11   evaluate_0.10
```