mixtureReg: A Quick Start

In this tutorial, we are going to show how to use **mixtureReg** to model data with two possible regimes.

Data

The data used for demostration purpose here is the CO2 data set from the **mixtools** package.

```
library(mixtools)
```

```
## mixtools package, version 1.0.4, Released 2016-01-11
## This package is based upon work supported by the National Science Foundation under Grant No. SES-0518
```

```
data("CO2data")
head(CO2data)
```

```
##     GNP  CO2 country
## 1 19.02 14.7     CAN
## 2  3.67  3.9     MEX
## 3 28.20 20.8     USA
## 4 40.94  9.0     JAP
## 5 10.61  8.3     KOR
## 6 20.09 16.0     AUS
```

A simple example

The motivation of mixture of regressions is that there can be two different regimes in the data so we want to fit two lines through the data.

We can easily achieve this by putting two regression formula into a list and feed it into the **mixtureReg** function.

In this case, the message shows that the model converges in 32 iterations.

```
library(mixtureReg)

mx1 <- mixtureReg(
  regData = CO2data,
  formulaList = list(formula(CO2 ~ GNP),
                     formula(CO2 ~ GNP))
)
```

```
## Loading required package: dplyr
```

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

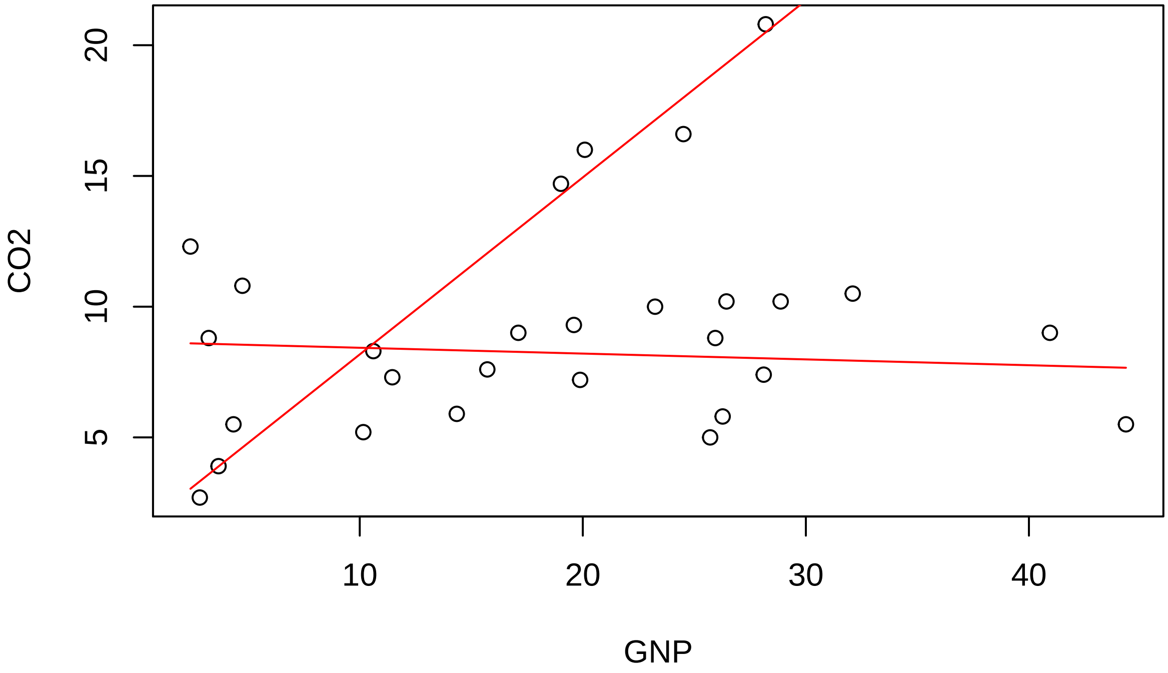
```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
## diff =  7.242733e-09
## iter =  32
## restart =  0
## log-likelihood =  -66.98373
```

The fit

We provide a plot method (S3 method) to visualize the predictions from the model. The circles below are the original data points and the red lines are predictions from our model.

```
plot(mx1, which = 1)
```

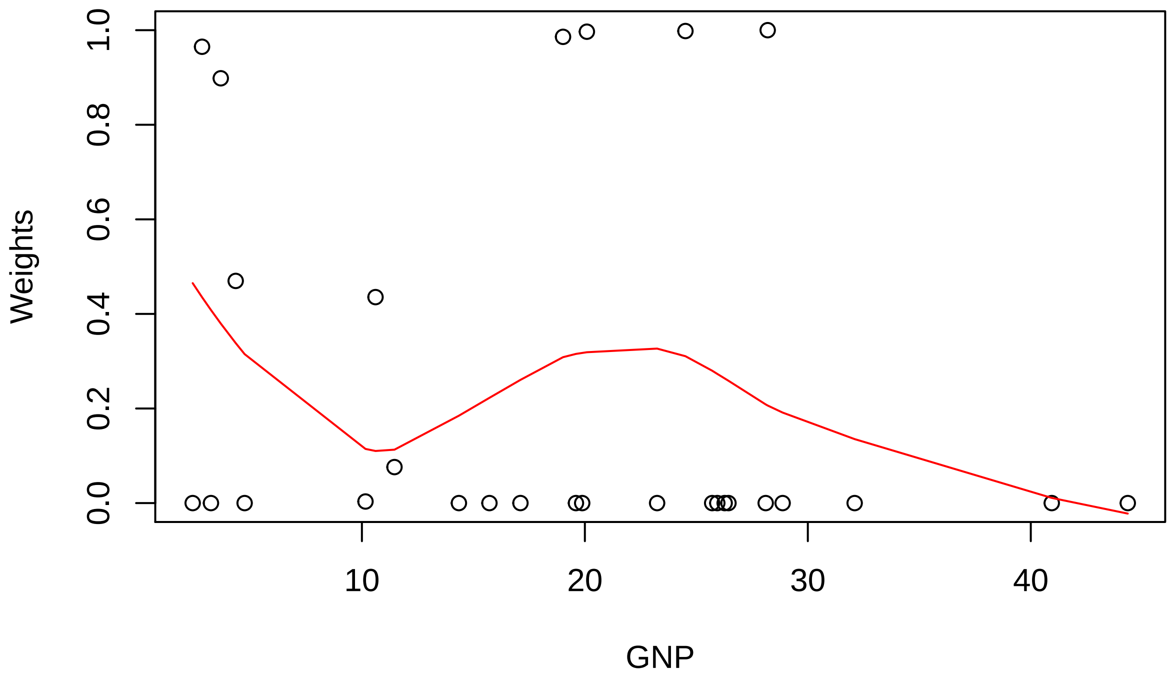


The weights

Other than predictions, The mixture of regressions also produces a weight estimate for each data point which indicates the posterior probability of membership to the first regression line.

We provide another plot method to visualize these weights. The circles below are the weights and the red line is a non-parametric fit through those weights.

```
plot(mx1, which = 2)
```



The iterations

(Mainly for debugging purposes) We also provide a *monitor* component for modelers to learn more about what are happening in iterations.

```
head(mx1$monitor)
```

```
##         diff iter restart    logLik     newLL   sigma1   sigma2     ratio
## 1 1.0000000    0       0 -77.93719        NA 2.778694 2.955875 0.9400580
## 2 0.1316328    1       0 -77.80556 -77.80556 2.695806 3.021592 0.8921805
## 3 0.3862011    2       0 -77.41936 -77.41936 2.532586 3.119528 0.8118491
## 4 0.7909071    3       0 -76.62845 -76.62845 2.278701 3.207566 0.7104143
## 5 0.9907000    4       0 -75.63775 -75.63775 2.018547 3.205237 0.6297653
## 6 0.8012655    5       0 -74.83648 -74.83648 1.841095 3.098706 0.5941497
##     lambda1   lambda2 error_message
## 1 0.5077512 0.4922488            NA
## 2 0.5092049 0.4907951            NA
## 3 0.5124791 0.4875209            NA
## 4 0.5211638 0.4788362            NA
## 5 0.5399321 0.4600679            NA
## 6 0.5672276 0.4327724            NA
```

Flexible modeling

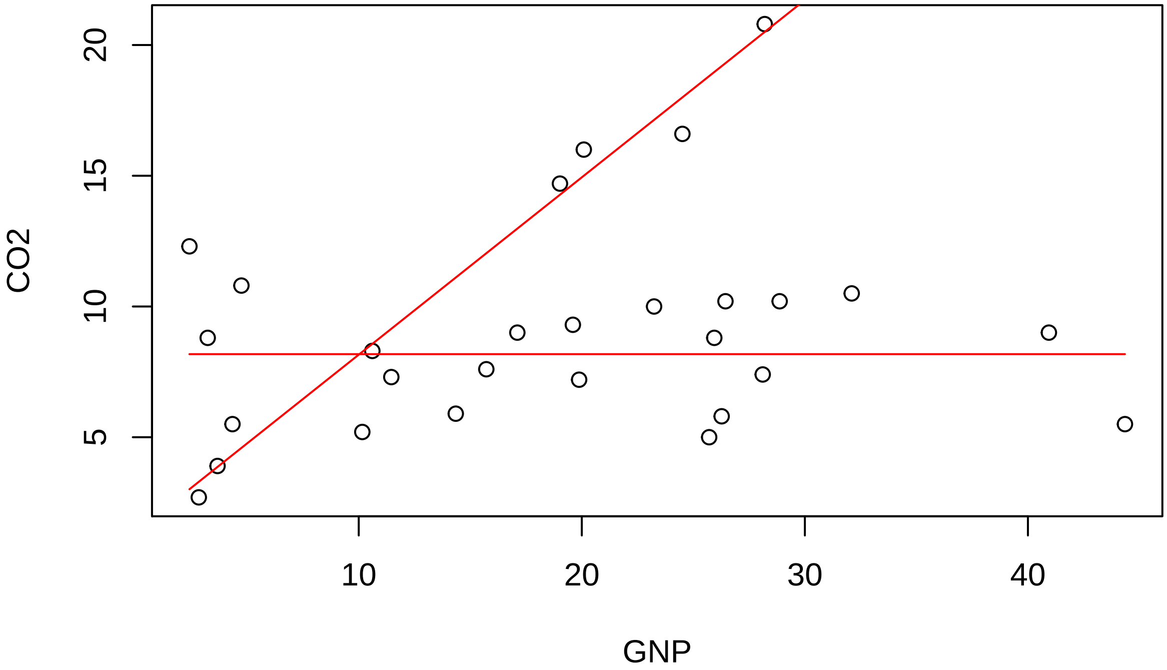
A nice feature of this package is that we can flexibly specify the formula as we would in **lm**.

For example, we can restrict one regression line to be horizontal with no slope coefficient:

```
mx2 <- mixtureReg(
  regData = CO2data,
  formulaList = list(formula(CO2 ~ 1),
                       formula(CO2 ~ GNP))
  )
```

```
## diff =  4.017906e-09
## iter =  29
## restart =  0
## log-likelihood =  -67.10555
```

```
plot(mx2, yName = "CO2", xName = "GNP", which = 1)
```

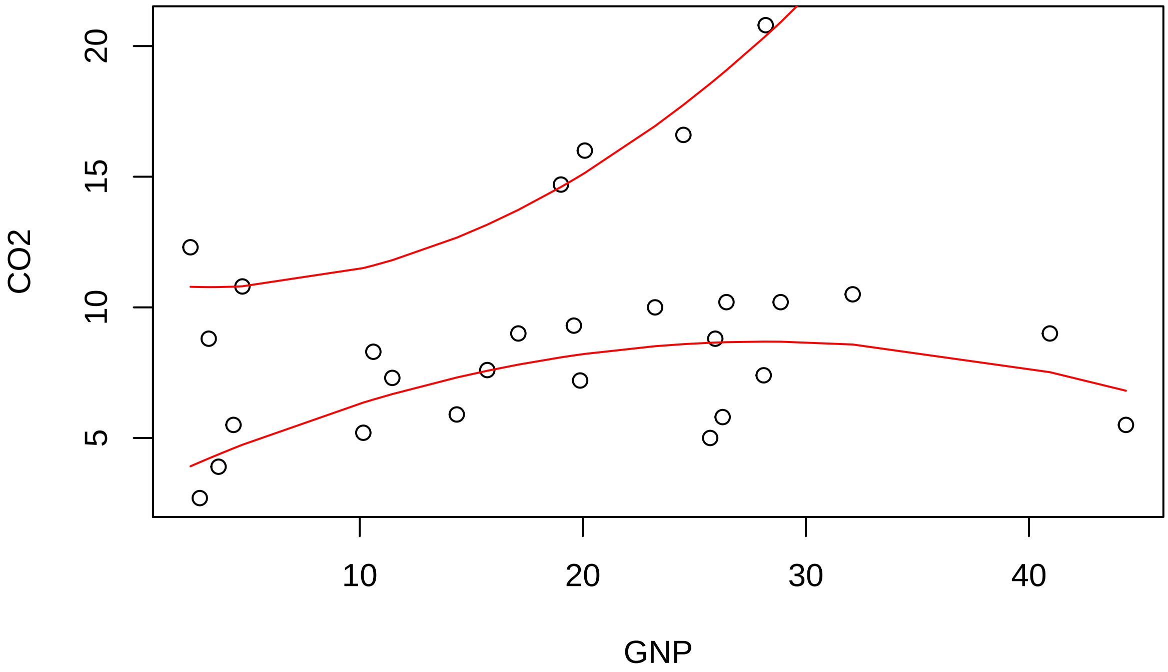


We can also specify 2nd order polynomial lines.

```
mx3 <- mixtureReg(
  regData = CO2data,
  formulaList = list(formula(CO2 ~ GNP + I(GNP^2)),
                     formula(CO2 ~ GNP + I(GNP^2)))
  )
```

```
## diff =  0.01458757
## iter =  45
## restart =  15
## log-likelihood =  -65.42241
```

```
plot(mx3, yName = "CO2", xName = "GNP", which = 1)
```



Comparison with mixtools

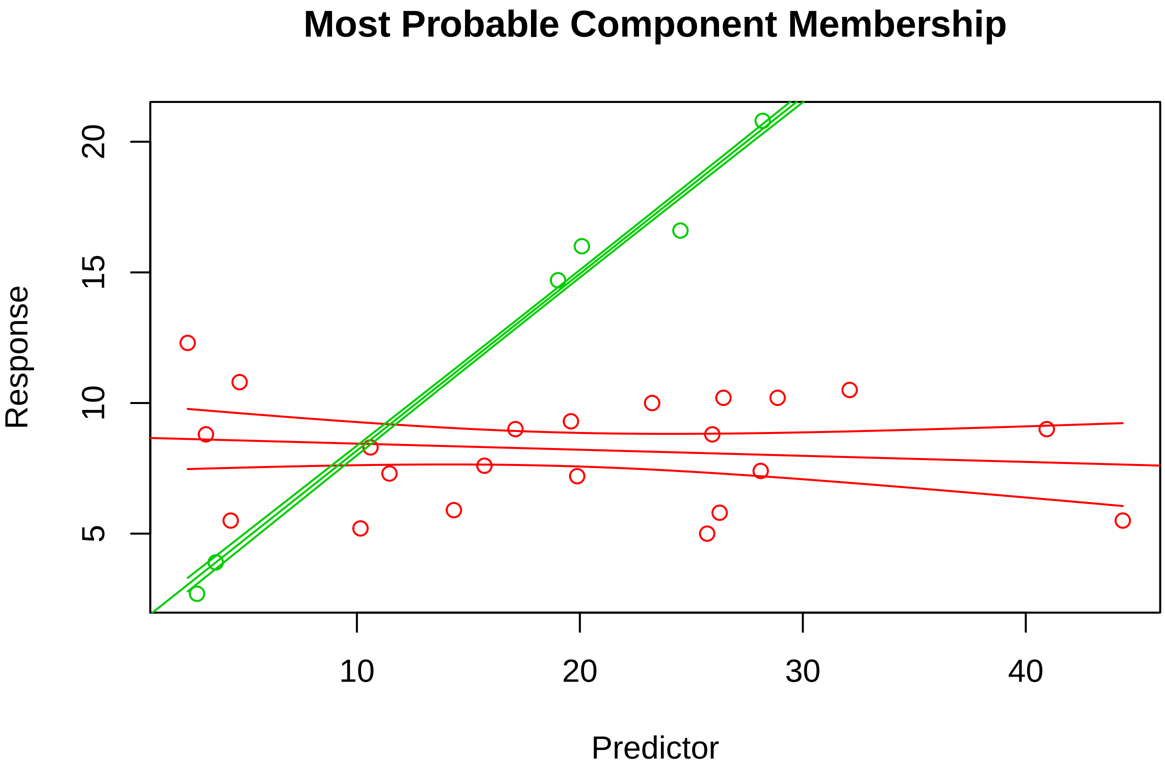
The main shortcoming of **mixtools** is that it doesn't provide easy to use options to restrict model coefficients like we do in model 2.

The following is an example from Tatiana Benaglia, Didier Chauveau, David R. Hunter, Derek Young (2009). This example produces similar results with our model 1.

```
compare1 <- mixtools::regmixEM(
  CO2data$CO2, CO2data$GNP,
  lambda = c(1/4, 3/4),
  beta = matrix(c(2, 0, 0, 1), 2, 2),
  sigma = c(1,1)
  )
```

```
## number of iterations= 18
```

```
plot(compare1, whichplots = 2)
```



References

de Veaux RD (1989). "Mixtures of Linear Regressions." Computational Statistics and Data Analysis, 8, 227-245.

Tatiana Benaglia, Didier Chauveau, David R. Hunter, Derek Young (2009). mixtools: An R Package for Analyzing Finite Mixture Models. Journal of Statistical Software, 32(6), 1-29. URL http://www.jstatsoft.org/v32/i06/.