Flexible Regression and Smoothing

Model Selection

Bob Rigby Mikis Stasinopoulos

Dipartimento di Scienze Cliniche a de Comunita, Universita degli studi di Milano, Italy, January 2018

gamlss

gamlss

Models

Statistical models are built to:

- explore the data where no theory exists, exploratory models,
- explain or verify a theory, explanatory models,
- predict future values, predictive models

or any combination of the above situations.

gamlss

GAMLSS componets

Let $\mathcal{M} = \{\mathcal{D}, \mathcal{G}, \mathcal{T}, \boldsymbol{\lambda}\}$ represent the GAMLSS model

- $\mathcal{D}$: distribution
- $\mathcal{G}$: the link function for distributional parameters
- $\mathcal{T}$: predictor terms for ($\boldsymbol{\eta}$' s) i.e. $\boldsymbol{\eta} = \mathbf{X}\boldsymbol{\beta} + \sum_j h_j(\mathbf{x}_j)$
- $\boldsymbol{\lambda}$: the hyperparameters

gamlss

Problems

Problems:

- which distribution
- which the link function for distributional parameters
- which x-variables for μ
- which x-variables for σ
- which x-variables for ν
- which x-variables for τ
- choosing the smoothing hyper parameters for terms in μ, σ, ν and τ
- selection between different (GAMLSS or not) models

gamlss

Questions to answer

- Do we need the extra complexity of GAMLSS?
- How we compare the different models? This depends also on the purpose of the study
  - is it for prediction (forecasting)?
  - are we exploring relationship between variables?
- Do we choose one or average between models?

gamlss

GAMLSS and GLM which to choose?

- is GAMLSS models are an improvement on the standard GLM?
- Bohl et al., (2013) compared GLM and GAMLSS estimators and concluded that "the GLM gamma was the most consistent" in their simulations.
- All their simulations involved a single explanatory categorical variable with two levels and equal sample sizes.

gamlss

GAMLSS statistical properties

For a parametric GAMLSS model ML estimation is used:

- if the model is correct
  - parameter estimators are (weakly) consistent, with correct asymptotic standard errors,
  - correct confidence interval coverage and test size,
  - parameter estimators are robust to outliers in specific cases
- if the model is not correct
  - the parameter estimators may not be consistent estimators of true population parameters.

gamlss

GLM statistical properties

- if the mean model is correct
  - the estimators of the mean model parameters are always strongly consistent (property of the exponential family)
- if the mean model is correct but the variance and/or distribution model is wrong
  - mean model parameter estimators are (strongly) consistent but asymptotically inefficient
  - estimated SE are in general, asymptotically incorrect unless robust SE are used,
- GLM mean model parameter estimators are not robust to outliers,

gamlss

Simulation studies

- Simulation 1 :
  - $Y \sim GA(100, 0.5)$, for $x = 0$
  - $Y \sim GA(130, 1)$ for $x = 1$
- simulation 2
  - $Y \sim GIG(100, 5, -2)$ for $x = 0$
  - $Y \sim GIG(130, 0.5, 2)$ for $x = 1$

gamlss

Simulation study 1

Table: Simulation 1 results: $Y \sim GA(100, 0.5)$, for $x = 0$, $Y \sim GA(130, 1)$ for $x = 1$.

sample size	Method	MSE [a]	% CC
(a) [1000,1000]	GLM	1.20	95.7
	GAMLSS	1.20	95.9
	GAMLSS (profile CI)	-	95.8
(b) [500, 1500]	GLM	1.21	**98.9**
	GAMLSS	1.21	94.8
	GAMLSS (profile CI)	-	94.8
(c) [1500, 500]	GLM	2.16	**85.1**
	GAMLSS	2.16	95.2
	GAMLSS (profile CI)	-	95.5

gamlss

Simulation study 2

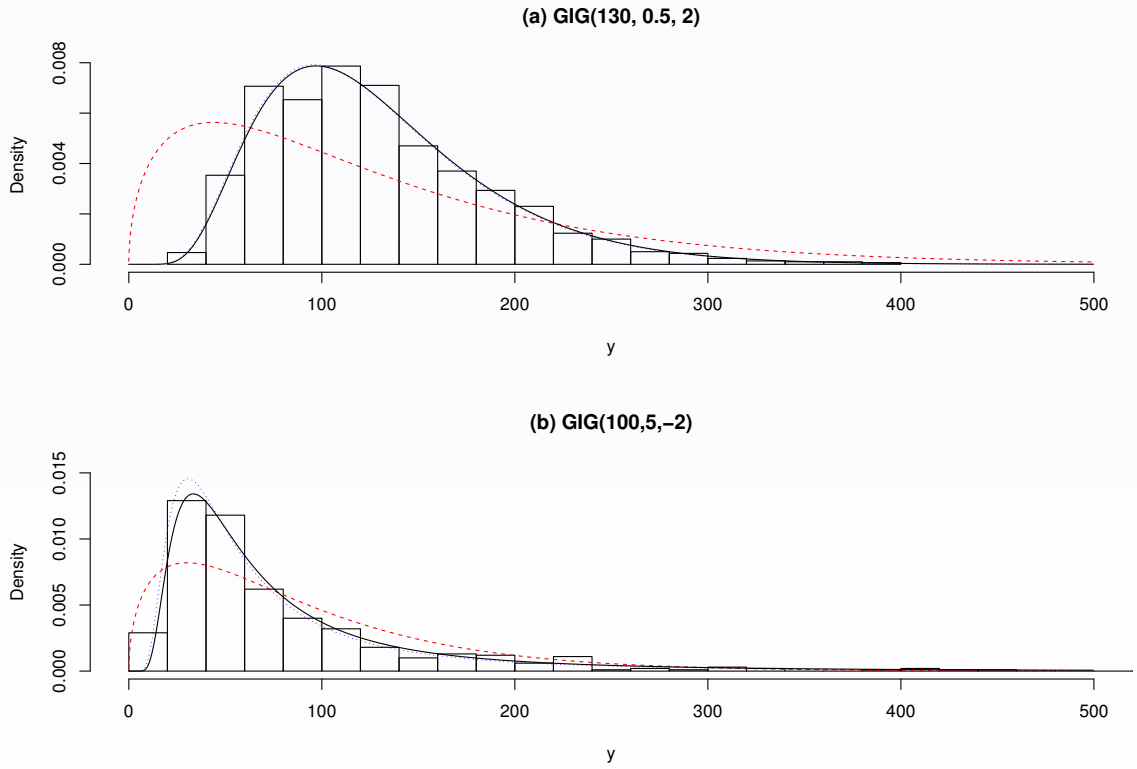
Table: Simulation 2: $Y \sim GIG(100, 5, -2)$ for $x = 0$, $Y \sim GIG(130, 0.5, 2)$ for $x = 1$.

sample size	Method	MSE[a]	% CC
(a) [1000,1000]	GLM	3.77	92.0
	GAMLSS	3.11	89.5
	GAMLSS (profile CI)	-	94.9
(b) [500, 1500]	GLM	6.98	**72.3**
	GAMLSS	5.24	89.0
	GAMLSS (profile CI)	-	94.8
(c) [1500, 500]	GLM	2.72	**98.4**
	GAMLSS	2.51	90.1
	GAMLSS (profile CI)	-	93.1

a Mean squared error multiplied by 1000

gamlss

Fitted distributions: GLM (dashed red), GAMLSS (dotted blue), true (black)



Conclusions

- GAMLSS model generally outperformed the GLM gamma model in terms of mean square error and coverage (the % of confidence intervals including the true parameter) for the parameter of interest.
- with different sample sizes the GLM coverage was very poor.
- robust GLM SE for mean model parameters should be used when the GLM incorrectly specifies the true variance model
- misspecification of the model can lead to seriously misleading standard errors, confidence intervals and tests for mean model parameters.

gamlss

Nested models

Nested models :

- parametric: use the likelihood ratio test
- Non-parametric: use likelihood ratio test as a guide

The function `LR.test()` can be used for example:

```
LR.test(m1, m2)
```

gamlss

Non nested models

Non-nested : use the Generalise Akaike Information Criterion

- use the functions `AIC(..., k)` or `GAIC(..., k)` where k is the penalty
- `GAIC(..., k=2)` is the standard AIC
- `GAIC(..., k=log(length(y)))` is the SBC or BIC

gamlss

Validation and Cross validation methods

K-fold Cross Validation

In data rich situations

- Training data set
- Validation data set
- Test data test

Use the "predictive" global deviance for selecting a model

gamlss

Validation/Test Global Deviance

$$\hat{\boldsymbol{\eta}}_i = \mathbf{X}_i\hat{\boldsymbol{\beta}}_i + \mathbf{Z}_{i1}\hat{\boldsymbol{\gamma}}_{i1} + \ldots + \mathbf{Z}_{ik_1}\hat{\boldsymbol{\gamma}}_{iJ_i}$$

and $\hat{\boldsymbol{\theta}}_i = g_i(\hat{\boldsymbol{\eta}})$ for $i = 1, 2, 3, 4$:

$$GDEV = -2\hat{\ell}(\mathbf{y}, |\boldsymbol{\lambda}_o).$$

$$\tilde{\boldsymbol{\eta}}_i = \tilde{\mathbf{X}}_i\hat{\boldsymbol{\beta}}_i + \tilde{\mathbf{Z}}_{i1}\hat{\boldsymbol{\gamma}}_{i1} + \ldots + \tilde{\mathbf{Z}}_{ik_1,}\hat{\boldsymbol{\gamma}}_{iJ_i}$$

$\tilde{\boldsymbol{\theta}}_i = g_i(\tilde{\boldsymbol{\eta}})$ for $i = 1, 2, 3, 4$.
The validation (or test) global deviance is defined as:

$$VDEV = TDEV = -2\tilde{\ell}(\tilde{\mathbf{y}}, |\boldsymbol{\lambda}_o).$$

gamlss

The different function for model selection

Comp.	All data	K-fold CV	Val./Test

$\mathcal{D}$	`GAIC()`
	`wp()`
		`gamlssCV()`,
		`CV()`
			`gamlssVGD()`,
			`VGD()`
			`getTGV()`
			`TGD()`

$\mathcal{G}$	`deviance()` *	`gamlssCV()`	as above

$\mathcal{T}$	`drop1()`, `add1()`,
	`add1ALL()`,
	`drop1ALL()`,
	`stepGAIC()`
	`stepGAICAll.A()`
	`stepGAICAll.B()`
		`gamlssCV()`
		`CV()`
			`drop1TGD()`
			`add1TGD()`
			
			`stepTGD()`

Λ global	`findhyper()`	`optim()`*	`optim()`*

gamlss

Choosing the distribution

How to select a distributeion?

	NO	*PE*	*TF*	*SHASH*	*SEP3*	*SST*
μ	✓	✓	✓	✓	✓	✓
σ	✓	✓	✓	✓	✓	✓
ν		✓	✓	✓	✓	✓
τ				✓	✓	✓

- Start with a simple appropriate distribution e.g. NO
- Check the residuals
- Increase the complexity until the residuals looking good

gamlss

The US pollution data

Data summary: US pollution data

- `y` : sulpher dioxide concentration in air in mgs. per c.m.
- `x1` : average annual temperature in degrees F
- `x2` : number of manufacturers employing $>$ 20 workers
- `x3` : population size in thousands
- `x4` : average annual wind speed in miles per hour
- `x5` : average annual rainfall in inches
- `x6` : average number of days rainfall per year

gamlss

drop1()

```
data(usair)
mod1 <- gamlss(y ~ ., data = usair, family = GA)
drop1(mod1)
drop1(mod1,  parallel="snow", ncpus=4 )
```

gamlss

drop1()

```
Single term deletions for
mu
Model:
y ~ x1 + x2 + x3 + x4 + x5 + x6
       Df    AIC     LRT  Pr(Chi)
<none>    319.16
x1      1 327.58 10.4245 0.001244 **
x2      1 326.92  9.7557 0.001788 **
x3      1 321.39  4.2299 0.039717 *
x4      1 324.08  6.9247 0.008501 **
x5      1 320.57  3.4141 0.064642 .
x6      1 317.16  0.0017 0.966960
---
Signif. codes:  0 "***" 0.001 "**" 0.01 "*Õ"0.05 "." 0.1 Ô " 1
```

gamlss

add1()

```
add1(mod1, scope = ~(x1 + x2 + x3 + x4 + x5 + x6)^2)
add1(mod1, scope = ~(x1 + x2 + x3 + x4 + x5 + x6)^2,
             parallel="snow", ncpus=4)
```

gamlss

add1()

```
 Single term additions for
mu
Model:
y ~ x1 + x2 + x3 + x4 + x5 + x6
       Df    AIC     LRT   Pr(Chi)
<none>    319.16
x1:x2   1 320.09  1.0689 0.3012045
x1:x3   1 319.40  1.7626 0.1843028
...
x4:x5   1 307.07 14.0870 0.0001745 ***
x4:x6   1 320.33  0.8346 0.3609322
x5:x6   1 318.74  2.4188 0.1198894
---
```

gamlss

stepGAIC()

```
mod2 <- stepGAIC(mod1,  parallel="snow", ncpus=4)
mod21 <- stepGAIC(mod1, k=length(usair$y))
mod3 <- stepGAIC(mod1, scope=list(lower=~1,
                upper=~(x1+x2+x3+x4+x5+x6)^2))
mod2$anova
mod4 <- stepGAIC(mod1, parameter="sigma",
                      scope=~x1+x2+x3+x4+x5+x6)
```

gamlss

Different Strategies

How to select explanatory variables?

	x_1	x_2	x_3	x_4	x_5	x_6
μ						
σ						
ν						
τ						

- Strategy A
- Strategy B
- Other strategies?
- Boosting

gamlss

Strategy A

Strategy A:

- It starts with a forward stepwise selection using GAIC.
- Each x variables is set for selection first for μ then for σ, ν and τ
- then it does a backward elimination for ν, σ and μ.

	x_1	x_2	x_3	x_4	x_5	x_6
μ		✓	✓	✓		✓
σ			✓	✓		
ν	✓		✓			
τ				✓		

gamlss

stepGAICAll.A()

```
m1 <- gamlss(y~1, data=usair, famly=GA,
                      trace=FALSE)
m2<- stepGAICAll.A(m1, scope=list(lower=~1,
         upper=~x1+x2+x3+x4+x5+x6))
m3 <- stepGAICAll.A(m1, scope=list(lower=~1,
          upper=~pb(x1)+pb(x2)+pb(x3)+pb(x4)+pb(x5)
          +pb(x6)), k=log(41))
```

gamlss

Strategy B

Strategy B: forward stepwise selection using GAIC in which an x variable is selected for all the parameters

Table: selecting explanatory variables

	x_1	x_2	x_3	x_4	x_5	x_6
μ	✓		✓			✓
σ	✓		✓			✓
ν	✓		✓			✓
τ	✓		✓			✓

gamlss

stepGAICAll.B()

```
m4<- stepGAICAll.B(m1, scope=list(lower=~1,
                    upper=~x1+x2+x3+x4+x5+x6),
                     k=log(41))
```

gamlss

Boosting

GAMLSS for high-dimensional data – a flexible approach based on boosting

Andreas Mayr*[1], Nora Fenske[2], Benjamin Hofner[1], Thomas Kneib[3], Matthias Schmid[1]

[1] Institut für Medizininformatik, Biometrie und Epidemiologie, Friedrich-Alexander-Universität Erlangen-Nürnberg, Germany
[2] Institut für Statistik, Ludwig-Maximilians-Universität München, Germany
[3] Institut für Mathematik, Carl von Ossietzky Universität Oldenburg, Germany

gamlss

gamlssCV()

```
rand1 <- sample (10 , 610, replace=TRUE)
# detacting how many cores exist in the machine
nC <- detectCores()
# no parallel
g1 <- gamlssCV(y~pb(x,df=2), sigma.formula=~pb(x,df=1),
       data=abdom, family=NO, rand=rand1)
# using multicore
g2 <- gamlssCV(y~pb(x,df=2), sigma.formula=~pb(x,df=1),
         data=abdom, family=LO, rand=rand1,
         parallel = "multicore", ncpus = nC )
# using snow
g3 <- gamlssCV(y~pb(x,df=2), sigma.formula=~pb(x,df=1),
        data=abdom, family=TF, rand=rand1,
        parallel = "snow", ncpus = nC )
CV(g1,g2,g3)
```

gamlss

gamlssVGD()

```
rand <- sample(2, 610, replace=TRUE, prob=c(0.6,0.4))
#-------------------------------------------------------------------
# using the argument rand
v1 <- gamlssVGD(y~pb(x,df=2),sigma.fo=~pb(x,df=1),
                  data=abdom, family=NO, rand=rand)
v2 <- gamlssVGD(y~pb(x,df=2),sigma.fo=~pb(x,df=1),
                   data=abdom, family=LO, rand=rand)
v3 <- gamlssVGD(y~pb(x,df=2),sigma.fo=~pb(x,df=1),
                   data=abdom, family=TF, rand=rand)
VGD(v1,v2,v3)
```

gamlss

getTGD()

```
# fit models first
g1 <- gamlss(y~pb(x,df=2),sigma.fo=~pb(x,df=1),
               data=olddata, family=NO,)
g2 <- gamlss(y~pb(x,df=2),sigma.fo=~pb(x,df=1),
              data=olddata, family=LO)
g3 <- gamlss(y~pb(x,df=2),sigma.fo=~pb(x,df=1),
              data=olddata, family=TF)
# and then use
gg1 <-getTGD(g1, newdata=newdata)
gg2 <-getTGD(g2, newdata=newdata)
gg3 <-getTGD(g3, newdata=newdata)
TGD(gg1,gg2,gg3)
```

gamlss

stepVGD()

```
# complete model
v1 <- gamlss(R~pb(Fl)+pb(A)+H+loc,
                    sigma.fo=~pb(Fl)+pb(A)+H+loc,
                data=oldrent, family=GA, trace=FALSE)
# drop1TGDP
nC <- detectCores()
v2<- drop1TGD(v1, newdata=newrent,  parallel="snow",
     ncpus=nC)
v4<- stepTGD(v0, scope=~pb(Fl)+pb(A)+H+loc,
             newdata=newrent,
             parallel="snow", ncpus=nC)
```

gamlss

Methods of choosing the smoothing parameters

Global	**Method**	**Reference**
Global	ML /REML (e.g. Laplace)	Rigby and Stasinopoulos (2005)
Global	GAIC (e.g. SBC)	Rigby and Stasinopoulos (2004,2006)
Global	Validation Global Deviance (VGD)	Stasinopoulos and Rigby (2007)
Local	ML	Rigby and Stasinopoulos (2013) i) alternate method ii) The Q-function
Local	GAIC	Rigby and Stasinopoulos (2013)
Local	Generalized Cross	Wood (2006)

gamlss

The AIDS data

Data summary: The AIDS data cases in the U.K. from January 1983 to March 1994

`y` : the number of quarterly aids cases in England and Wales

`x` : time in quarters from January 1983

`qrt` : a factor for the quarterly seasonal effect

gamlss

find.hyper()

- find.hyper() function
- Needs a model
- mod1 <- quote(gamlss(y cs(x, df = p[1]) + qrt, family = NBI, data = aids, trace = FALSE))
- op <- find.hyper(model = mod1, par = c(3), lower = c(1), steps = c(0.1), pen = 2.8)

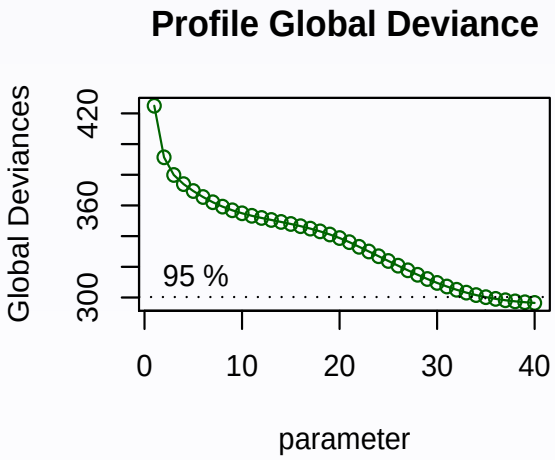
gamlss

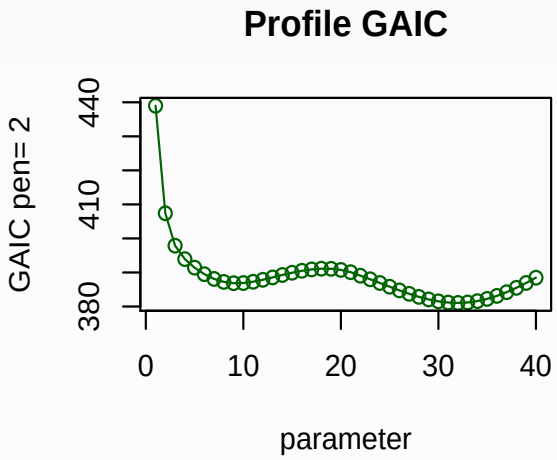
find.hyper()

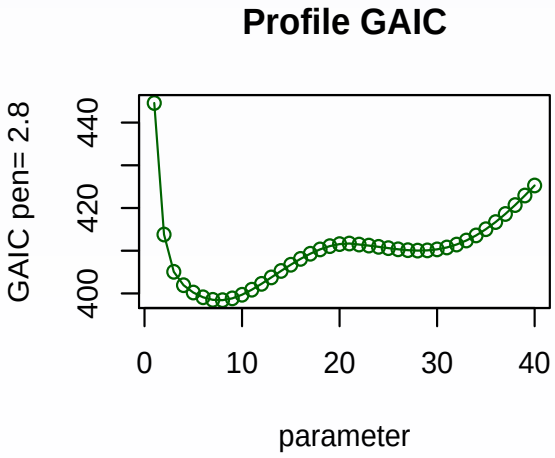
```
op
$par
[1] 7.652983
$value
[1] 398.3985
$counts
function gradient
      12       12
$convergence
[1] 0
$message
[1] "CONVERGENCE: REL_REDUCTION_OF_F <= FACTR*EPSMCH"
```

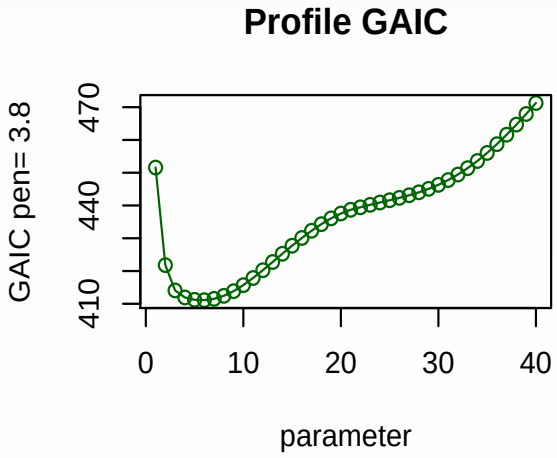
gamlss

find.hyper()



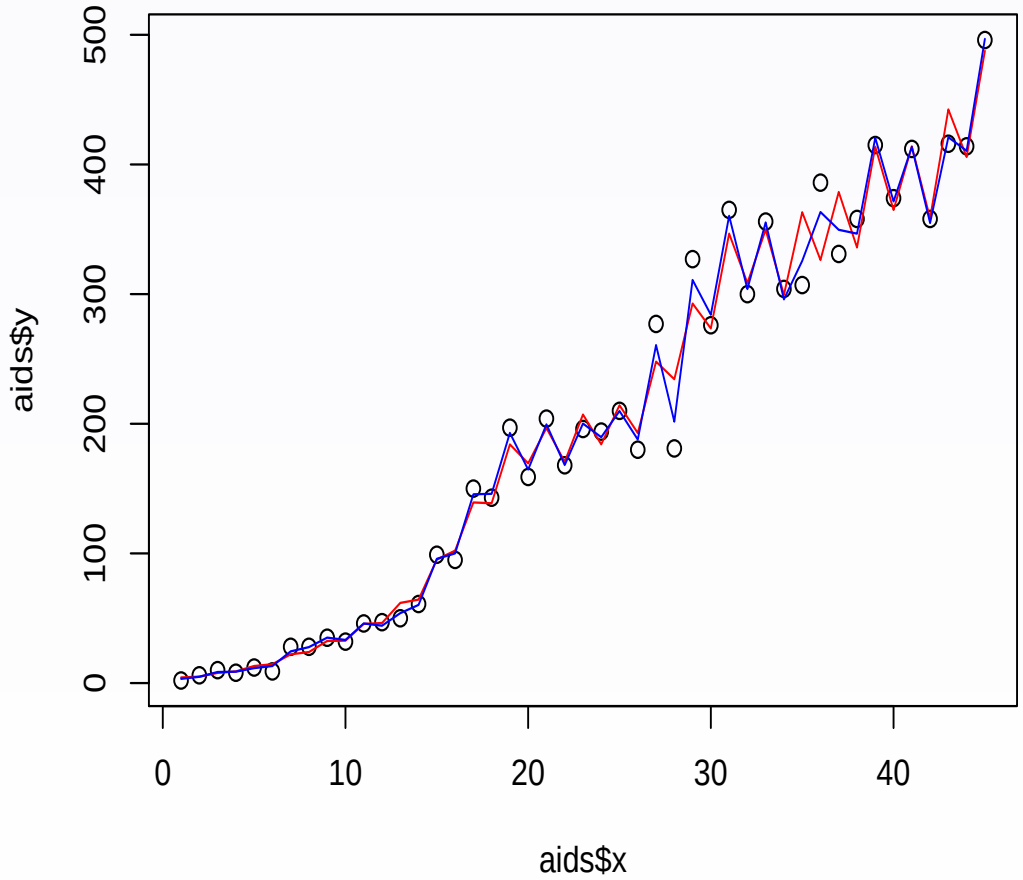






gamlss

find.hyper()



gamlss

Local ML

```
> data(aids)
> m1 <- gamlss(y~pb(x)+qrt, data=aids, family=NBI)
GAMLSS-RS iteration 1: Global Deviance = 373.1785
...
GAMLSS-RS iteration 5: Global Deviance = 366.9258
> edf(m1)
Effective df for mu model
   pb(x)
7.588861
```

gamlss

Local GCV

```
> m2 <- gamlss(y~pb(x, method="GCV")+qrt, data=aids, family=NBI)
GAMLSS-RS iteration 1: Global Deviance = 365.5964
...
GAMLSS-RS iteration 5: Global Deviance = 360.6098
> edf(m2)
Effective df for mu model
pb(x, method = "GCV")
             9.252291
```

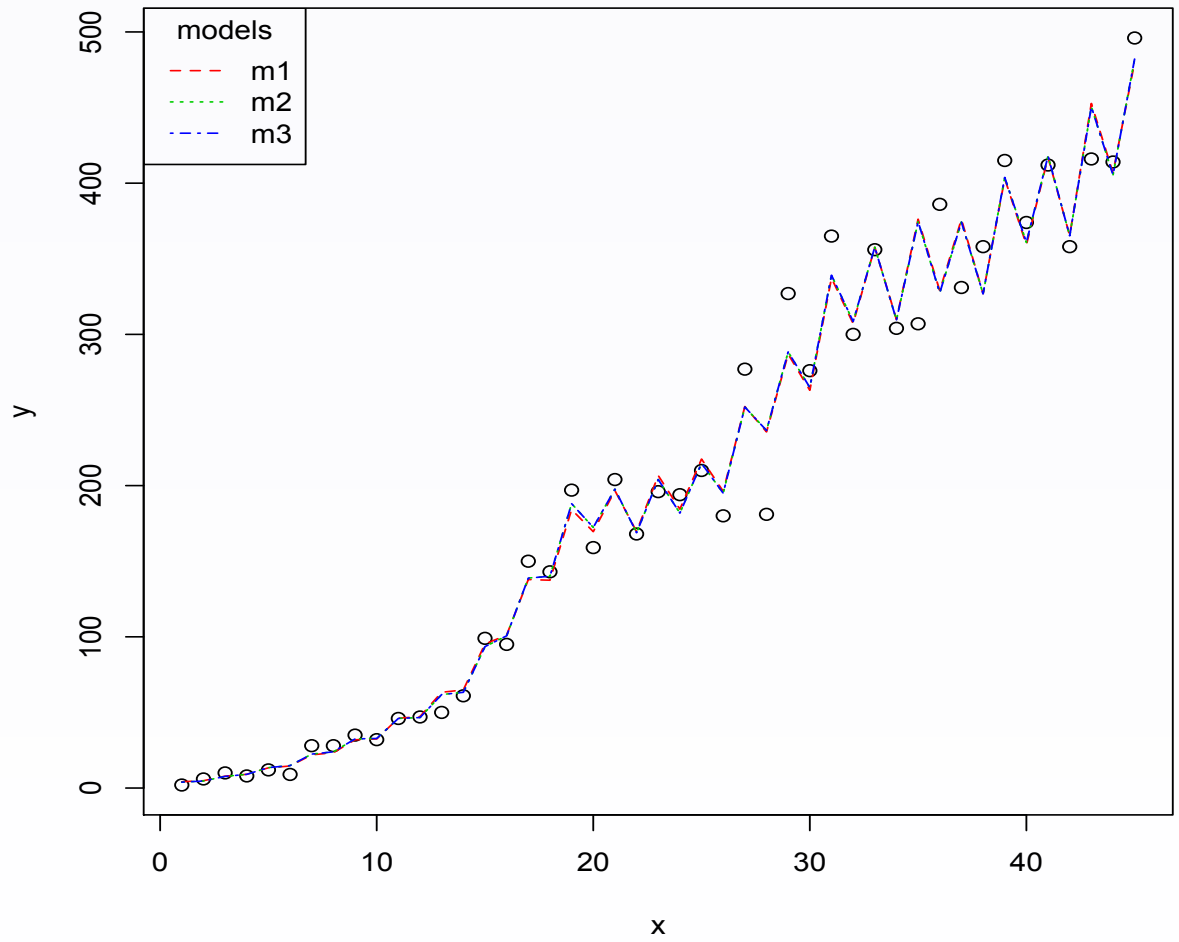
gamlss

Local GAIC

```
> m3 <- gamlss(y~pb(x, method="GAIC", k=2)+qrt, data=aids, family=NBI)
GAMLSS-RS iteration 1: Global Deviance = 395.9969
...
GAMLSS-RS iteration 7: Global Deviance = 358.7914
GAMLSS-RS iteration 8: Global Deviance = 358.791
> edf(m3)
Effective df for mu model
pb(x, method = "GAIC", k = 2)
                    9.990218
```

gamlss

Local methods



gamlss

Conclusions

- Selecting a GAMLSS model is not easy
- selecting using GAIC seems to be a reasonable solution (but which k?)
- selecting a distributions is not trivial
- selecting terms in a model for large data sets can be slow
- selecting smoothing parameters using local methods seems OK

gamlss

END

for more information see

www.gamlss.com

gamlss