universität innsbruck

Distributional Regression

Computation, Model Choice and Variable Selection

Nikolaus Umlauf

https://eeecon.uibk.ac.at/~umlauf/

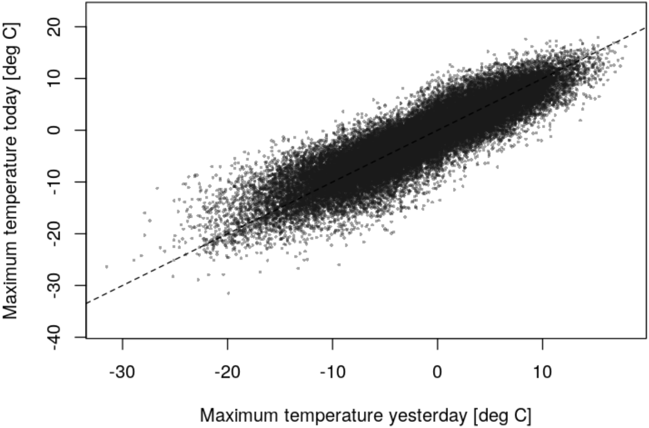
Overview

1. Introduction
2. Model specification
3. Model fitting
4. L1-type penalization
5. R package *bamlss*
6. An application on the Bird Breeding Survey

Introduction

Zugspitze daily maximum temperature data (1900/08-2016/12)

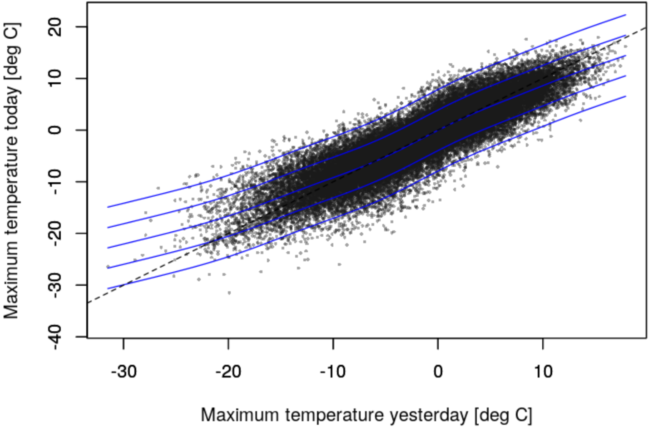
$$\mathtt{T} \sim N(\mu, \sigma^2).$$



Introduction

Zugspitze daily maximum temperature data (1900/08-2016/12)

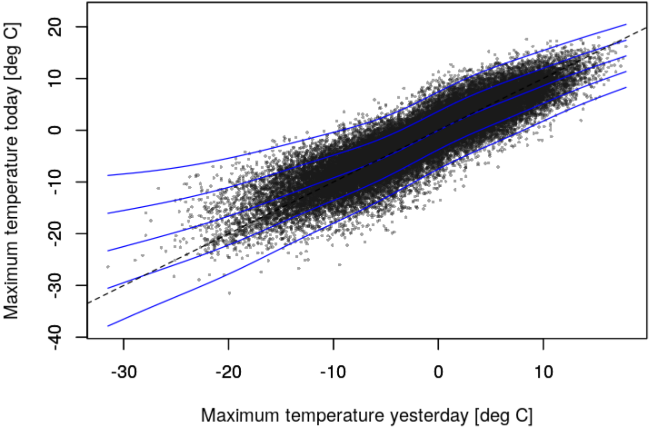
$$\mathtt{T} \sim N(\mu = f(\mathtt{T}_{t-1}), \, log(\sigma^2) = \beta_0).$$



Introduction

Zugspitze daily maximum temperature data (1900/08-2016/12)

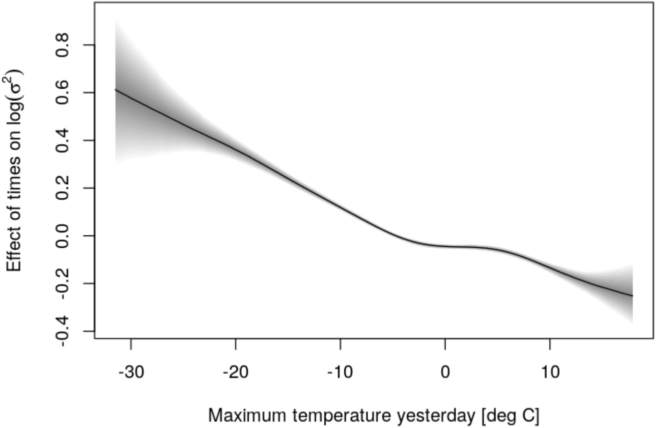
$$\mathtt{T} \sim N(\mu = f(\mathtt{T}_{t-1}),\ log(\sigma^2) = f(\mathtt{T}_{t-1})).$$



Introduction

Zugspitze daily maximum temperature data (1900/08-2016/12)

$$\mathtt{T} \sim N(\mu = f(\mathtt{T}_{t-1}),\ log(\sigma^2) = f(\mathtt{T}_{t-1})).$$



Model specification

Any parameter of a population distribution $\mathcal{D}$ may be modeled by explanatory variables

$$y \sim \mathcal{D}\left(h_1(\theta_1) = \eta_1,\ h_2(\theta_2) = \eta_2, \dots,\ h_K(\theta_K) = \eta_K\right),$$

Each parameter is linked to a structured additive predictor

$$h_k(\theta_k) = \eta_k = \eta_k(\mathbf{x}; \boldsymbol{\beta}_k) = f_{1k}(\mathbf{x}; \boldsymbol{\beta}_{1k}) + \ldots + f_{J_k k}(\mathbf{x}; \boldsymbol{\beta}_{J_k k}),$$

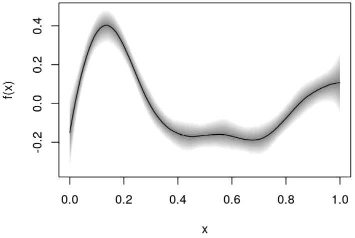
$j = 1, \dots, J_k$ and $k = 1, \dots, K$ and $h_k(\cdot)$ are known monotonic link functions.

Vector of function evaluations $\mathbf{f}_{jk} = (f_{jk}(\mathbf{x}_1; \boldsymbol{\beta}_{jk}), \dots, f_{jk}(\mathbf{x}_n; \boldsymbol{\beta}_{jk}))^\top$

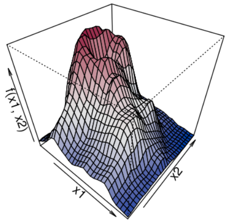
$$\mathbf{f}_{jk} = \begin{pmatrix} f_{jk}(\mathbf{x}_1; \boldsymbol{\beta}_{jk}) \\ \vdots \\ f_{jk}(\mathbf{x}_n; \boldsymbol{\beta}_{jk}) \end{pmatrix} = f_{jk}(\mathbf{X}_{jk}; \boldsymbol{\beta}_{jk}).$$

Model specification

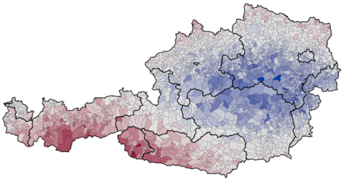
Nonlinear effects of continuous covariates



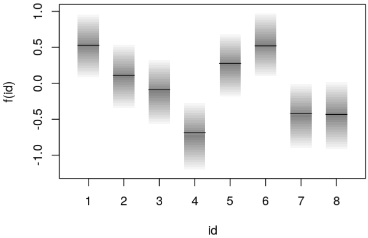
Two-dimensional surfaces



Spatially correlated effects f(x) = f(s)



Random intercepts f(x) = f(id)



Model specification

For simple linear effects $\mathbf{X}_{jk}\boldsymbol{\beta}_{jk}$: $p_{jk}(\boldsymbol{\beta}_{jk}) \propto const$.

For the smooth terms:

$$p_{jk}(\boldsymbol{\beta}_{jk}; \boldsymbol{\tau}_{jk}, \boldsymbol{\alpha}_{jk}) \propto d_{\boldsymbol{\beta}_{jk}}(\boldsymbol{\beta}_{jk} | \boldsymbol{\tau}_{jk}; \boldsymbol{\alpha}_{\boldsymbol{\beta}_{jk}}) \cdot d_{\boldsymbol{\tau}_{jk}}(\boldsymbol{\tau}_{jk} | \boldsymbol{\alpha}_{\boldsymbol{\tau}_{jk}}).$$

Using a basis function approach a common choice is

$$d_{\boldsymbol{\beta}_{jk}}(\boldsymbol{\beta}_{jk} | \boldsymbol{\tau}_{jk}, \boldsymbol{\alpha}_{\boldsymbol{\beta}_{jk}}) \propto |\mathbf{P}_{jk}(\boldsymbol{\tau}_{jk})|^{\frac{1}{2}} \exp\left(-\frac{1}{2}\boldsymbol{\beta}_{jk}^{\top}\mathbf{P}_{jk}(\boldsymbol{\tau}_{jk})\boldsymbol{\beta}_{jk}\right).$$

Precision matrix $\mathbf{P}_{jk}(\boldsymbol{\tau}_{jk})$ derived from prespecified penalty matrices $\boldsymbol{\alpha}_{\boldsymbol{\beta}_{jk}} = \{\mathbf{K}_{1jk}, \ldots, \mathbf{K}_{Ljk}\}$.

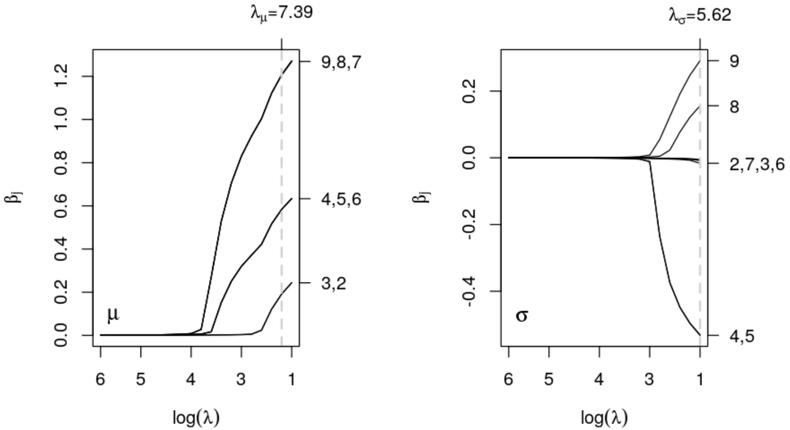
The variances parameters $\boldsymbol{\tau}_{jk}$ are equivalent to the inverse smoothing parameters in a frequentist approach.

Regularization in the GAMLSS framework

- A gradient boosting approach is provided by Mayr et al. (2012).
- Allows for variable selection within GAMLSS framework.
- Corresponding R-package *gamboostLSS* (Hofner et al., 2015).
- Provides a large number of pre-specified distributions.
- **New:** an alternative *gradient boosting* approach is implemented in the R-package *bamlss* (Umlauf et al., 2018b):
  - embeds many different approaches suggested in literature and software,
  - serves as unified conceptional "Lego toolbox" for complex regression models.

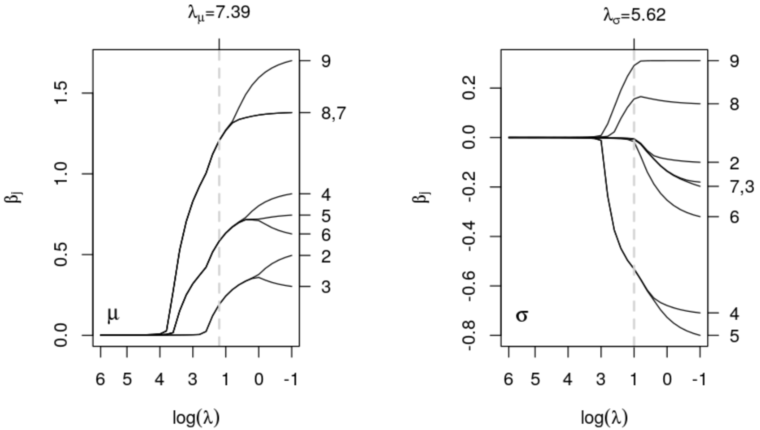
Regularization in the GAMLSS framework

New model terms $f_{jk}(\mathbf{x};\beta_{jk})$ with LASSO-type penalties.



Regularization in the GAMLSS framework

New model terms $f_{jk}(\mathbf{x}; \beta_{jk})$ with LASSO-type penalties.



Model fitting

The main building block of regression model algorithms is the probability density function $d_y(\mathbf{y}|\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_K)$.

Estimation typically requires to evaluate

$$\begin{aligned}\ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X}) \; = \; & \sum_{i=1}^{n} \log \, d_y(y_i; \theta_{i1} = h_1^{-1}(\eta_{i1}(\mathbf{x}_i, \boldsymbol{\beta}_1)), \ldots \\ & \ldots, \theta_{iK} = h_K^{-1}(\eta_{iK}(\mathbf{x}_i, \boldsymbol{\beta}_K))),\end{aligned}$$

with $\boldsymbol{\beta} = (\boldsymbol{\beta}_1^\top, \ldots, \boldsymbol{\beta}_K^\top)^\top$ and $\mathbf{X} = (\mathbf{X}_1, \ldots, \mathbf{X}_K)$.

The log-posterior

$$\log \, \pi(\boldsymbol{\beta}, \boldsymbol{\tau}; \mathbf{y}, \mathbf{X}, \boldsymbol{\alpha}) \propto \ell(\boldsymbol{\beta}; \mathbf{y}, \mathbf{X}) + \sum_{k=1}^{K} \sum_{j=1}^{J_k} \left[\log \, p_{jk}(\boldsymbol{\beta}_{jk}; \boldsymbol{\tau}_{jk}, \boldsymbol{\alpha}_{jk})\right],$$

where $\boldsymbol{\tau} = (\boldsymbol{\tau}_1^\top, \ldots, \boldsymbol{\tau}_K^\top)^\top = (\boldsymbol{\tau}_{11}^\top, \ldots, \boldsymbol{\tau}_{J_1 1}^\top, \ldots, \boldsymbol{\tau}_{1K}^\top, \ldots, \boldsymbol{\tau}_{J_K K}^\top)^\top$ (frequentist, penalized log-likelihood).

Model fitting

Posterior mode estimation, fortunately, partitioned updating is possible

$$\begin{aligned}
\beta_1^{(t+1)} &= U_1(\beta_1^{(t)}, \beta_2^{(t)}, \ldots, \beta_K^{(t)}) \\
\beta_2^{(t+1)} &= U_2(\beta_1^{(t+1)}, \beta_2^{(t)}, \ldots, \beta_K^{(t)}) \\
&\vdots \\
\beta_K^{(t+1)} &= U_K(\beta_1^{(t+1)}, \beta_2^{(t+1)}, \ldots, \beta_K^{(t)}),
\end{aligned}$$

E.g., Newton-Raphson type updating

$$\beta_k^{(t+1)} = U_k(\beta_k^{(t)}, \cdot) = \beta_k^{(t)} - \mathbf{H}_{kk}\left(\beta_k^{(t)}\right)^{-1}\mathbf{s}\left(\beta_k^{(t)}\right).$$

Can be further partitioned for each function within parameter block k. Moreover, using a basis function approach yields IWLS updates

$$\beta_{jk}^{(t+1)} = (\mathbf{X}_{jk}^\top \mathbf{W}_{kk}\mathbf{X}_{jk} + \mathbf{G}_{jk}(\boldsymbol{\tau}_{jk}))^{-1}\mathbf{X}_{jk}^\top \mathbf{W}_{kk}(\mathbf{z}_k - \boldsymbol{\eta}_{k,-j}^{(t)}).$$

Model fitting

A simple generic algorithm for distributional regression models:

```
while(eps > ε & t < maxit) {
    for(k in 1:K) {
        for(j in 1:J[k]) {
            Compute η̃ = η_k − f_jk.
            Obtain new (β*_jk, τ*_jk)^⊤ = U_jk(X_jk, y, η̃, β^[t]_jk, τ^[t]_jk).
            Update η_k.
        }
    }
    t = t + 1
    Compute new eps.
}
```

Functions $\mathtt{U}_{\mathtt{jk}}(\cdot)$ could either return updates from an optimizing algorithm or proposals from a MCMC sampler.

L1-type penalization

Idea: depending on the type of covariate effects, subtract a combination of (parts of) the following penalty terms $\tau^{-1}J(\boldsymbol{\beta})$ from the log-likelihood.

Classical LASSO (Tibshirani, 1996): For a metric covariate x_{jk} use

$$J_m(\beta_{jk}) = |\beta_{jk}|\,.$$

Group LASSO (Meier et al., 2008): For a (dummy-encoded) categorical covariate $\mathbf{x}_{jk}$ use

$$J_g(\boldsymbol{\beta}_{jk}) = ||\boldsymbol{\beta}_{jk}||_2\,,$$

with vector $\boldsymbol{\beta}_{jk}$ collecting all corresponding coefficients.

L1-type penalization

Alternatively, for categorical covariates often *clustering* of categories with implicit *factor selection* is desirable.

Fused LASSO (Gertheiss and Tutz, 2010): Depending on the *nominal* (left) or *ordinal* scale level (right) of the covariate, use

$$J_f(\boldsymbol{\beta}_{jk}) = \sum_{l>m} w_{lm}^{(jk)} \left|\beta_{jkl} - \beta_{jkm}\right| \text{ or } J_f(\boldsymbol{\beta}_{jk}) = \sum_{l=1}^{c_{jk}} w_{l}^{(jk)} \left|\beta_{jkl} - \beta_{jk,l-1}\right|$$

where c_{jk} is the number of levels of categorical predictor $\mathbf{x}_{jk}$ and $w_{lm}^{(jk)}, w_{l}^{(jk)}$ denote suitable weights. Choosing $l = 0$ as the reference, $\beta_{jk0} = 0$ is fixed.

L1-type penalization

Quadratic approximations of the penalties (compare Oelker & Tutz, 2017)

$$J_{jk}(\boldsymbol{\beta}_{jk}) \approx J_{jk}(\boldsymbol{\beta}_{jk}^{(t)}) + \frac{1}{2}\left(\boldsymbol{\beta}_{jk}^{\top}\mathbf{P}_{jk}(\boldsymbol{\beta}_{jk})\boldsymbol{\beta}_{jk} + (\boldsymbol{\beta}_{jk}^{(t)})^{\top}\mathbf{P}_{jk}(\boldsymbol{\beta}_{jk}^{(t)})\boldsymbol{\beta}_{jk}^{(t)}\right),$$

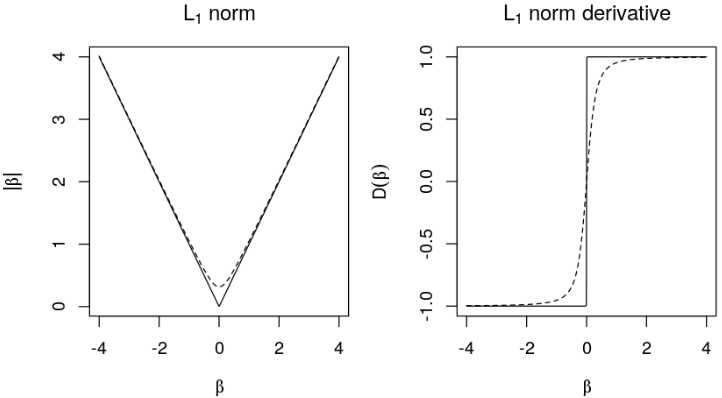
with

$$\mathbf{P}_{jk}(\boldsymbol{\beta}_{jk}^{(t)}) = q'_{jk}\left(\left\|\mathbf{a}_{jk}^{\top}\boldsymbol{\beta}_{jk}^{(t)}\right\|_{N_{jk}}\right) \cdot \frac{D_{jk}(\mathbf{a}_{jk}^{\top}\boldsymbol{\beta}_{jk}^{(t)})}{\mathbf{a}_{jk}^{\top}\boldsymbol{\beta}_{jk}^{(t)}} \cdot \mathbf{a}_{jk}\mathbf{a}_{jk}^{\top}.$$

E.g., $\|\beta\|_1 = |\beta|$ is approximated by $\sqrt{\beta^2 + c}$, hence, IWLS based updating functions $\mathtt{U_{jk}}(\cdot)$ are relatively easy to implement.

L1-type penalization

Example of the approximation of the L_1 norm.



Usually setting the constant to $c \approx 10^{-5}$ works well.

R package *bamlss*

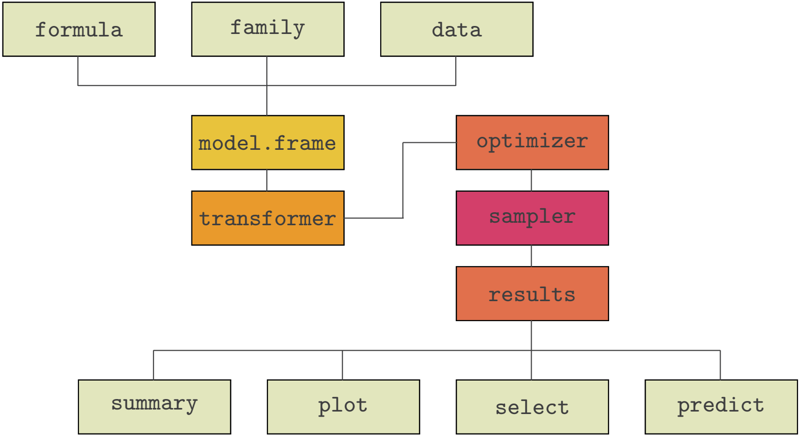
The package is available at

https://CRAN.R-project.org/package=bamlss

Development version, in R simply type

```
R> install.packages("bamlss",
+    repos = "http://R-Forge.R-project.org")
```

R package *bamlss*



In principle, the setup does not restrict to any specific type of engine (Bayesian or frequentist).

R package *bamlss*

Type	Function
Parser	`bamlss.frame()`
Transformer	`bamlss.engine.setup()`, `randomize()`
Optimizer	`bfit()`, `opt()`, `cox.mode()`, `jm.mode()`
	`boost()`, `lasso()`
Sampler	`GMCMC()`, `JAGS()`, `STAN()`, `BayesX()`,
	`cox.mcmc()`, `jm.mcmc()`
Results	`results.bamlss.default()`

To implement new engines, only the building block functions have to be exchanged.

R package *bamlss*

Work in progress ...

Function	Distribution
`beta_bamlss()`	Beta distribution
`binomial_bamlss()`	Binomial distribution
`cnorm_bamlss()`	Censored normal distribution
`cox_bamlss()`	Continuous time Cox-model
`gaussian_bamlss()`	Gaussian distribution
`gamma_bamlss()`	Gamma distribution
`gpareto_bamlss()`	Generalized Pareto distribution
`jm_bamlss()`	Continuous time joint-model
`multinomial_bamlss()`	Multinomial distribution
`mvn_bamlss()`	Multivariate normal distribution
`poisson_bamlss()`	Poisson distribution
...	

New families only require density, distribution, random number generator, quantile, score and hess functions.

R package *bamlss*

Wrapper function:

```
R> f <- list(y ~ la(id,fuse=2), sigma ~ la(id,fuse=1))
R> b <- bamlss(f, family = "gaussian", sampler = FALSE,
+    optimizer = lasso, criterion = "BIC", multiple = TRUE)
```

Standard extractor and plotting functions:

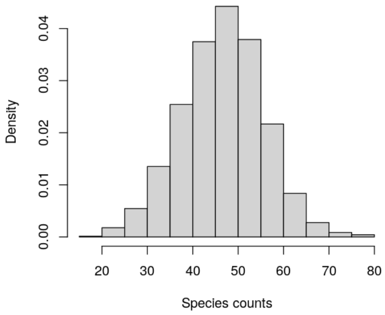
`summary()`, `plot()`, `fitted()`, `residuals()`, `predict()`, `coef()`, `logLik()`, `DIC()`, `samples()`, ...

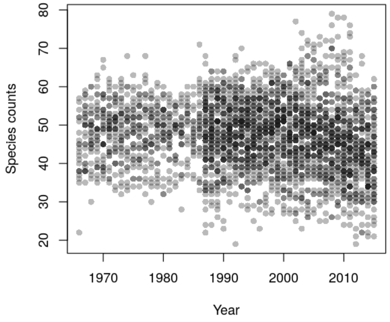
Bird Breeding Survey

- Pardieck et. al (2017) *North American Breeding Bird Survey Dataset 1966–2016*, version 2016.0. U.S. Geological Survey, Patuxent Wildlife Research Center. url:`https://www.pwrc.usgs.gov/bbs/RawData/`
- Long-term, large-scale, international avian monitoring program initiated in 1966 to track the status and trends of North American bird populations.
- Each year during the height of the avian breeding season, participants skilled in avian identification collect bird population data along roadside survey routes.
- At each stop, a 3-minute point count is conducted. During the count, every bird seen within a 0.25-mile radius or heard is recorded.

Bird Breeding Survey

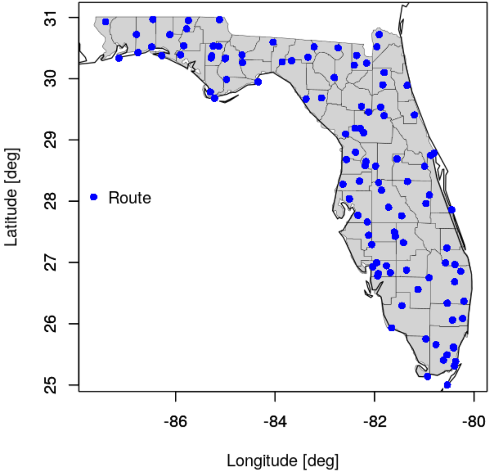
Change of average richness over time?





Bird Breeding Survey

Route specific effects?



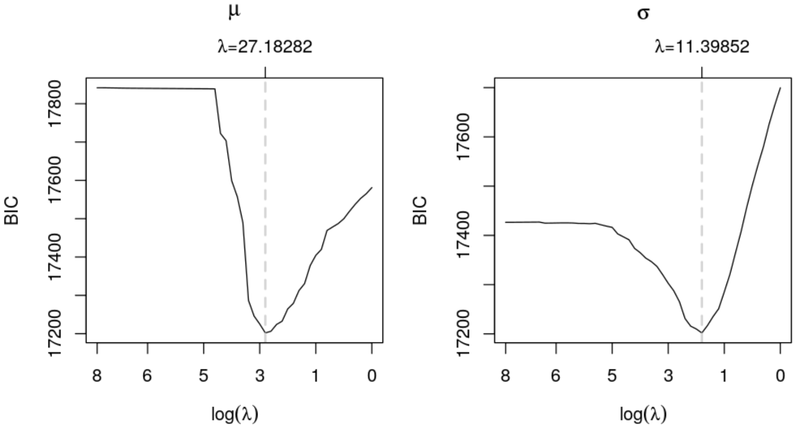
Bird Breeding Survey

Model in R (potentially 344 parameters):

```
R> f <- list(
+   counts ~ la(year,fuse=2) + la(route,fuse=1) + s(lon,lat,k=50),
+   sigma ~ la(year,fuse=2) + la(route,fuse=1) + s(lon,lat,k=50)
+ )

R> b <- bamlss(f, data = bbs, sampler = FALSE, optimizer = lasso,
+   criterion = "BIC", multiple = TRUE, nlambda = 50)

R> lasso.stop(b)

[1] 1781
attr(,"stats")
      logLik      logPost          BIC          edf    lambda.mu
 -8326.43632 -13856.38433  17201.90386     69.45302     27.18282
lambda.sigma
    11.39852
```

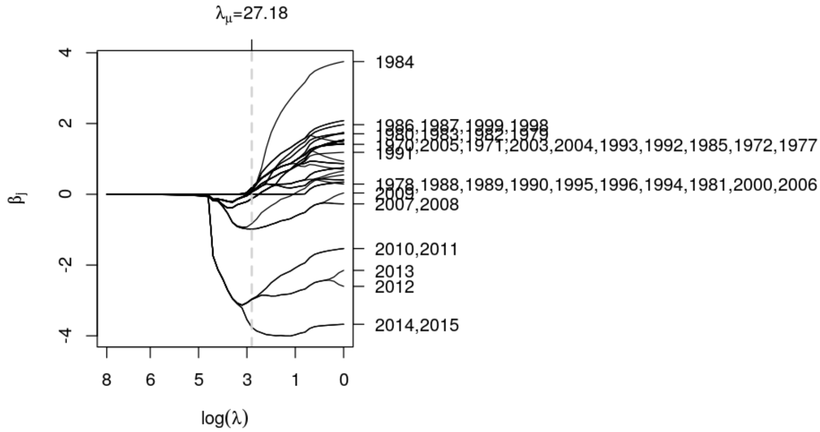
Bird Breeding Survey

```
R> lasso.plot(b, which = "criterion")
```



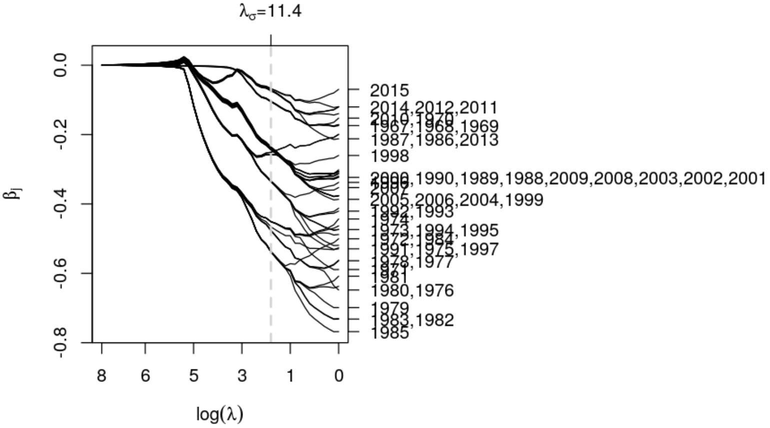
Bird Breeding Survey

```
R> lasso.plot(b, which = "parameters", model = "mu")
```



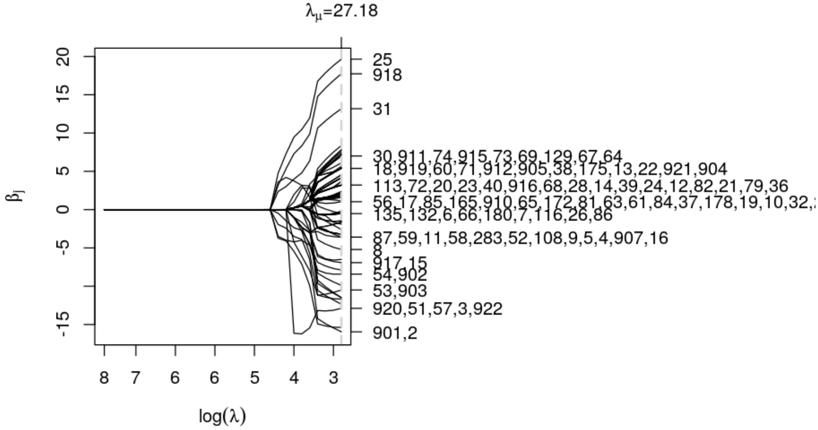
Bird Breeding Survey

```
R> lasso.plot(b, which = "parameters", model = "sigma")
```



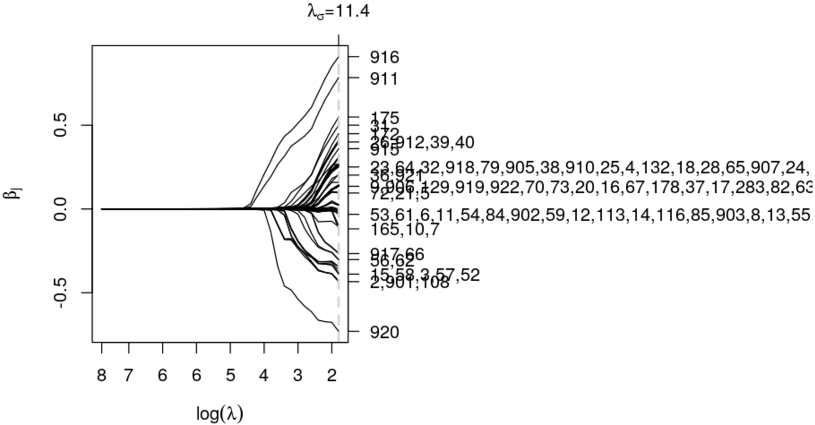
Bird Breeding Survey

```
R> lasso.plot(b, which = "parameters", model = "mu")
```



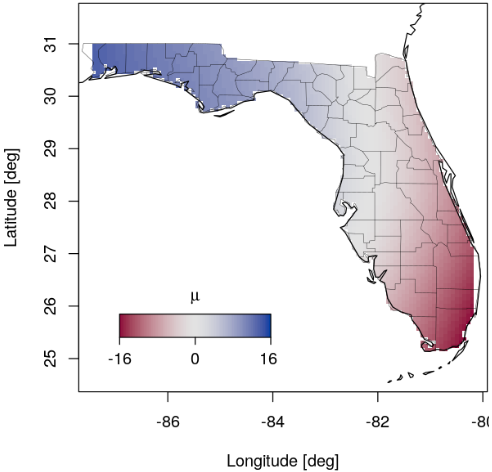
Bird Breeding Survey

```
R> lasso.plot(b, which = "parameters", model = "sigma")
```



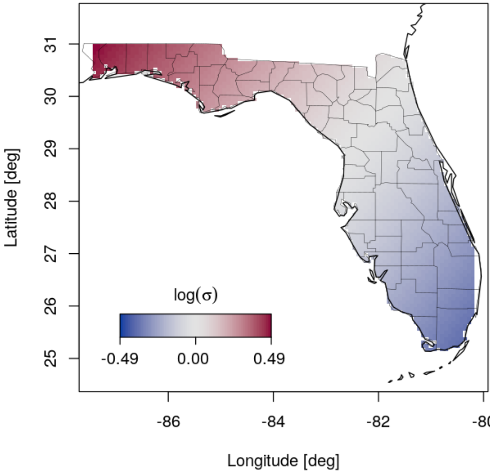
Bird Breeding Survey

```
R> p <- predict(b, newdata = nd, model = "mu",
+    term = "s(lon,lat)", mstop = lasso.stop(b))
```



Bird Breeding Survey

```
R> p <- predict(b, newdata = nd, model = "sigma",
+    term = "s(lon,lat)", mstop = lasso.stop(b))
```



References & Software

- Rigby, R. A. and D. M. Stasinopoulos (2005). Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* **54(3)**, 507 – 554.
- Stasinopoulos, D.M. & Rigby, R.A. (2007) Generalized additive models for location scale and shape (GAMLSS) in R. *Journal of Statistical Software* **23**(7).
- Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B 58* 267 – 288.
- Umlauf, N., Klein, N., & Zeileis, A. (2018a). BAMLSS: Bayesian additive models for location, scale and shape (and beyond). *Journal of Computational and Graphical Statistics*, doi:10.1080/10618600.2017.1407325.
- Umlauf, N., Klein, N., Zeileis, A. & Köhler, M. (2018b). **bamlss**: Bayesian additive models for location, scale and shape (and beyond). R package version 0.1-2. url:http://cran.r-project.org/package=bamlss

Thank you for your attention!

Nikolaus Umlauf

https://eeecon.uibk.ac.at/~umlauf/