Censored autoregressive regression models with Student-t innovations

Fernanda Lang Schumacher

Division of Biostatistics - CPH, The Ohio State University

Joint work with: Katherine Valeriano, Christian E. Galarza, Larissa A. Matos, and Victor Hugo Lachos

COMPSTAT 2022

Introduction

- Observations collected over time are often autocorrelated rather than independent.

- A stochastic model that has been useful in many real-world applications to deal with serial correlation in the error term is the AR model:

$$y_t = \phi_0 + \phi_1 y_{t-1} + \ldots + \phi_p y_{t-p} + \eta_t.$$

- In general, it is assumed that the disturbance η_t is normally distributed. However, distributions with heavier-than-normal tails, such as the Student-t distribution, might be more appropriate in real applications since they are less affected by outliers.

- An additional complication arises when observations are subject to upper or lower detection limits, below and above which they are not quantifiable.

Motivation: Total phosphorus concentration data

- Phosphorus is one of the two nutrients of main concern in Iowa river water, as excessive phosphorus in river water can result in eutrophication.

- Phosphorus concentration data (measured in mg/l) of West Fork Cedar River at Finchford, Iowa, USA, that were collected under the ambient water quality program conducted by the Iowa Department of Natural Resources (Iowa DNR) are available in the R package **carx**.

- The phosphorus concentration measurement was subject to a LOD of 0.02, 0.05, or 0.10, depending on the time, and therefore 15.47% observations are left-censored.

- A gap from 09/2008 to 03/2009 in the data is due to program suspension caused by a temporary lack of funding.

Motivation: Total phosphorus concentration data

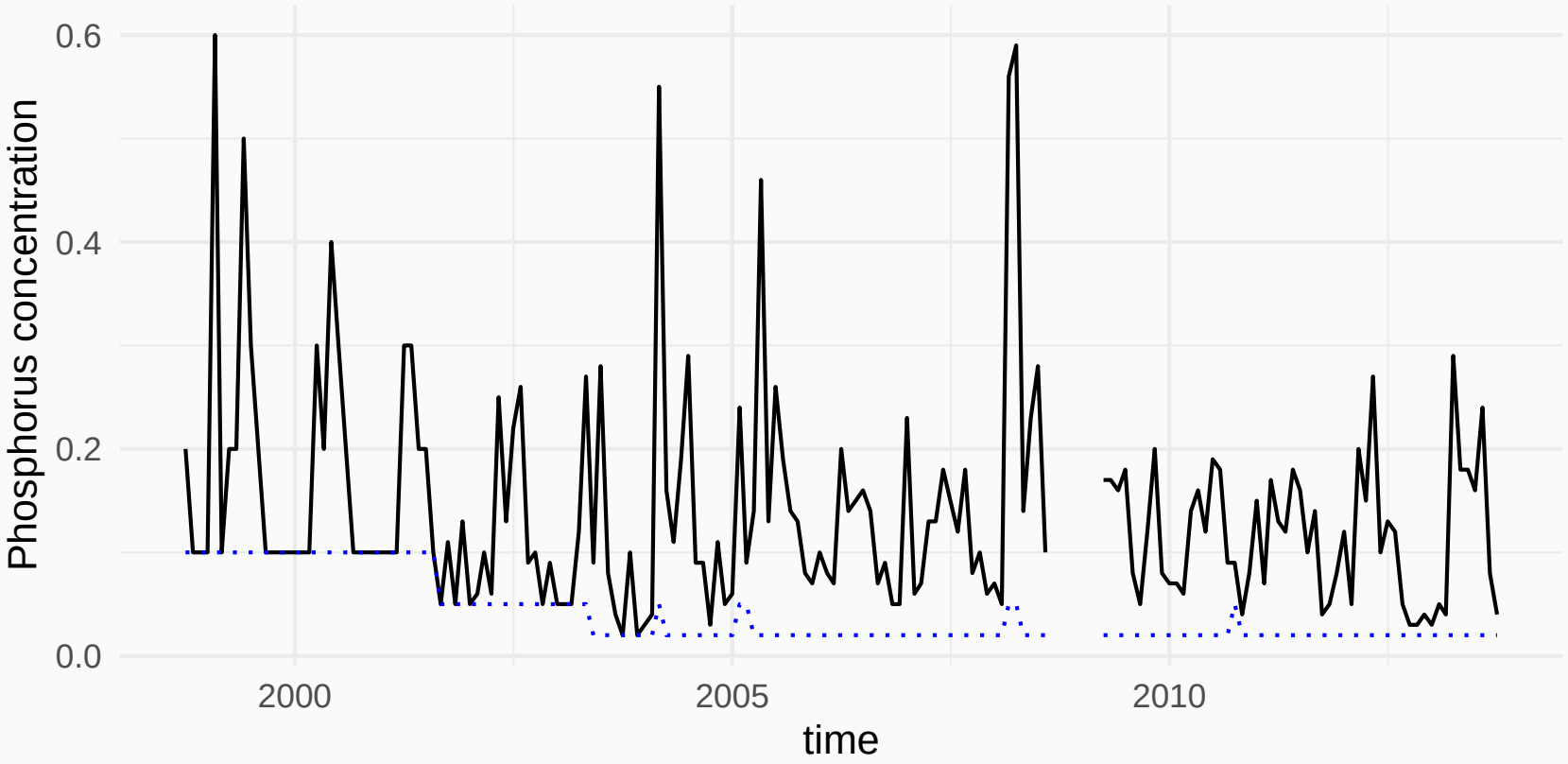


Figure 1: Phosphorus concentration time series and its limit of detection.

Motivation: Total phosphorus concentration data

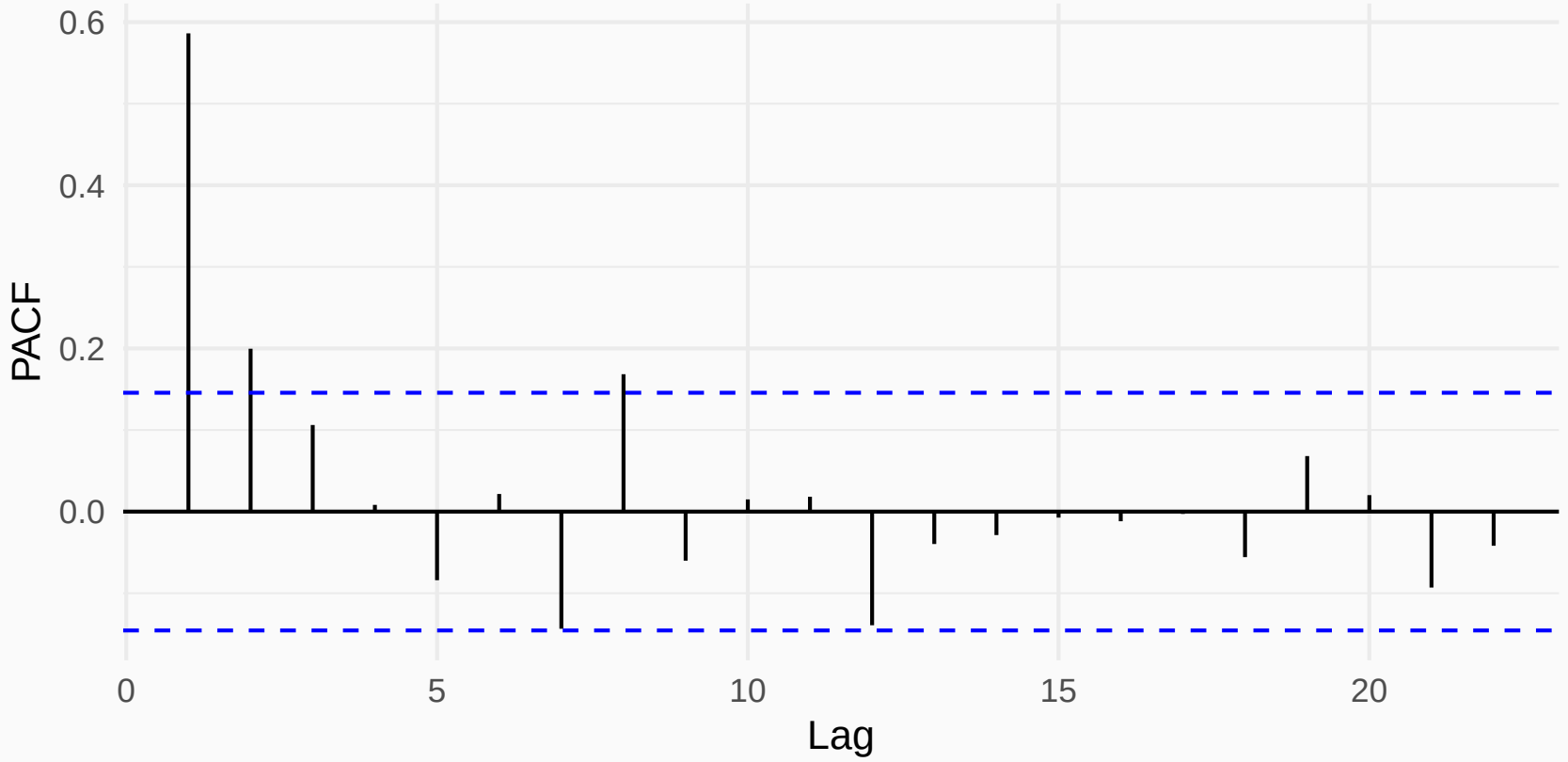


Figure 2: PACF from martingale residuals (Garay et al., 2017) for the censored model with independent errors.

Introduction

- Censored time series data are frequently encountered in environmental monitoring, medicine, and economics.
- Schumacher et al. (2017) considered an autoregressive censored linear model (CAR) estimation via an efficient stochastic approximation of the EM (SAEM) algorithm.
- Liu et al. (2019) proposed a coupled MCMC-SAEM algorithm to fit an AR(p) regression model with Student-t innovations accounting for missing data.
- Valeriano et al. (2022+) proposed an EM-type algorithm to estimate a CARt(p) model.
- The R package **ARCensReg** provide tools for fitting and evaluating CAR(p) and CARt(p) models.

Preliminaries

The autoregressive regression model of order p

A linear regression model with errors that are autocorrelated as a discrete-time autoregressive process of order p can be written as

$$\begin{aligned} y_t &= \mathbf{x}_t^\top \boldsymbol{\beta} + \xi_t, \\ \xi_t &= \phi_1 \xi_{t-1} + \ldots + \phi_p \xi_{t-p} + \eta_t, \quad \eta_t \overset{iid}{\sim} F(\cdot), \end{aligned}$$

for $t = 1, \ldots, n$, where

- y_t represents the dependent variable observed at time t,
- $\mathbf{x}_t = (x_{t1}, \ldots, x_{tq})^\top$ is a $q \times 1$ vector of known covariables,
- $\boldsymbol{\beta}$ is a $q \times 1$ vector of unknown regression parameters, and
- ξ_t is the regression error and follows an autoregressive model, with $\boldsymbol{\phi} = (\phi_1, \ldots, \phi_p)^\top$ being a $p \times 1$ vector of autoregressive coefficients and η_t being a shock of disturbance with distribution $F(\cdot)$.

The autoregressive regression model of order p

- Now, suppose that $\eta_t \sim t(0, \sigma^2, \nu)$. Then, the censored regression model with autoregressive errors will be called the *autoregressive regression t model of order p* (AR$t(p)$).

The autoregressive regression model of order p

- Now, suppose that $\eta_t \sim t(0, \sigma^2, \nu)$. Then, the censored regression model with autoregressive errors will be called the *autoregressive regression t model of order p* (AR$t(p)$).

- The distribution of η_t might be written as $\eta_t = Z_t/\sqrt{U_t}$, with $Z_t \overset{iid}{\sim} \mathcal{N}(0, \sigma^2) \perp U_t \overset{iid}{\sim} \text{Gamma}(\nu/2, \nu/2)$. Consequently, we have the following hierarchical representation:

$$\begin{aligned} \eta_t|(U_t = u_t) &\overset{ind}{\sim} \mathcal{N}(0, \sigma^2/u_t), \\ U_t &\overset{iid}{\sim} \text{Gamma}(\nu/2, \nu/2), \quad t = 1, \ldots, n. \end{aligned}$$

CAR(p) and CARt(p) models

The censored AR$t(p)$ model

- Assume that Y_t is not fully observed for all times. Instead, we observe (V_t, C_t), where V_t represents either an observed value or the LOD of a censored variable, and C_t is the censoring indicator defined as

$$C_t = \begin{cases} 1 & \text{if} \quad V_{t1} \leq Y_t \leq V_{t2}, \quad \text{(censored)} \\ 0 & \text{if} \quad V_t = Y_t. \quad \text{(observed)} \end{cases}$$

  The AR$t(p)$ model with censored observations will be called *censored autoregressive regression t model of order p* (CAR$t(p)$).

The censored AR$t(p)$ model

- Assume that Y_t is not fully observed for all times. Instead, we observe (V_t, C_t), where V_t represents either an observed value or the LOD of a censored variable, and C_t is the censoring indicator defined as

$$C_t = \begin{cases} 1 & \text{if} \quad V_{t1} \leq Y_t \leq V_{t2}, \quad \text{(censored)} \\ 0 & \text{if} \quad V_t = Y_t. \quad \text{(observed)} \end{cases}$$

  The AR$t(p)$ model with censored observations will be called *censored autoregressive regression t model of order p* (CAR$t(p)$).

- Left-censored if $C_t = 1$ and $V_t = (-\infty, V_{t2}]$;
- Right-censored if $C_t = 1$ and $V_t = [V_{t1}, +\infty)$;
- Interval censored if $C_t = 1$ and $V_t = [V_{t1}, V_{t2}]$; and
- Missing value if $C_t = 1$ and $V_t = (-\infty, +\infty)$.

The SAEM algorithm

Delyon et al. (1999) proposed splitting the E-step from the EM algorithm into a simulation step and an integration step (using a stochastic approximation):

- **E-step**:

  1. Simulation: Draw M samples of the missing data $\mathbf{y}_m^{(l,k)}, l = 1, \ldots, M$ from the conditional distribution $f(\mathbf{y}_m; \widehat{\boldsymbol{\theta}}^{(k)}, \mathbf{y}_o)$.

  2. Stochastic approximation: Update $Q_k(\boldsymbol{\theta})$ by

$$\widehat{Q}_k(\boldsymbol{\theta}) = \widehat{Q}_{k-1}(\boldsymbol{\theta}) + \delta_k \left(\frac{1}{M} \sum_{l=1}^{M} \ell_c(\boldsymbol{\theta}; \mathbf{y}_m^{(k,l)}, \mathbf{y}_o) - \widehat{Q}_{k-1}(\boldsymbol{\theta}) \right),$$

  where δ_k is a decreasing sequence of positive numbers such that $\sum_{k=1}^{\infty} \delta_k = \infty$ and $\sum_{k=1}^{\infty} \delta_k^2 < \infty$, also known as the smoothness parameter (Kuhn & Lavielle, 2005).

Likelihood function - CAR(p)

From Schumacher et al. (2017), for the CAR(p) model we have

$$L(\boldsymbol{\theta} \mid \mathbf{y}_o) = \phi_{n^o}(\mathbf{y}_o; \mathbf{X}^o\beta, \boldsymbol{\Sigma}^{oo})\Phi_{n^c}(\mathbf{V}^c; \boldsymbol{\mu}, \mathbf{S}).$$

Likelihood function - CAR(p)

From Schumacher et al. (2017), for the CAR(p) model we have

$$L(\boldsymbol{\theta} \mid \mathbf{y}_o) = \phi_{n^o}(\mathbf{y}_o; \mathbf{X}^o\beta, \boldsymbol{\Sigma}^{oo})\Phi_{n^c}(\mathbf{V}^c; \boldsymbol{\mu}, \mathbf{S}).$$

Letting the censored values be missing data, we have $\mathbf{y}_c = \left(\mathbf{y}^\top, \mathbf{V}^\top, \mathbf{C}^\top\right)^\top$, and an EM-type algorithm can be applied to $\ell(\boldsymbol{\theta}|\mathbf{y}_c)$:

$$\hat{Q}_k(\boldsymbol{\theta}) = E[\ell(\boldsymbol{\theta}|\mathbf{y}_c)|\mathbf{V}, \mathbf{C}, \hat{\boldsymbol{\theta}}^{(k)}] = -\frac{1}{2}\left[n\log\sigma^2 + \log|\mathbf{M}_p(\boldsymbol{\phi})| + \frac{1}{\sigma^2}\gamma^{(k)}\right],$$

where $\gamma^{(k)} = \text{tr}(\widehat{\mathbf{y}^{2(k)}}\mathbf{M}_n^{-1}(\boldsymbol{\phi})) - 2\widehat{\mathbf{y}^{\top(k)}}\mathbf{M}_n^{-1}(\boldsymbol{\phi})\mathbf{X}\beta + \beta^\top\mathbf{X}^\top\mathbf{M}_n^{-1}(\boldsymbol{\phi})\mathbf{X}\beta.$

Log-likelihood function - CARt(p)

Now, letting $\mathbf{y}_{(t,p)} = (y_{t-1}, \ldots, y_{t-p})^\top$, we have
$Y_t|\boldsymbol{\theta}, \mathcal{F}_{t-1} \overset{d}{=} Y_t|\boldsymbol{\theta}, \mathbf{y}_{(t,p)} \sim t(\mu_t, \sigma^2, \nu)$, where
$\mu_t = \mathbf{x}_t^\top \boldsymbol{\beta} + (\mathbf{y}_{(t,p)} - \mathbf{X}_{(t,p)}\boldsymbol{\beta})^\top \boldsymbol{\phi}$ and $\boldsymbol{\theta} = (\boldsymbol{\beta}^\top, \boldsymbol{\phi}^\top, \sigma^2, \nu)^\top$.

From Valeriano et al (2022+), conditionally on the first p observations, the observed log-likelihood function can be computed by

$$\ell(\boldsymbol{\theta}; \mathbf{y}_o) = \log\left(\int_{\mathbb{R}^m} \prod_{t=p+1}^{n} f(y_t|\boldsymbol{\theta}, y_{t-1}, y_{t-2}, \ldots, y_{t-p}) \mathrm{d}\mathbf{y}_m\right).$$

Log-likelihood function - CARt(p)

Now, letting $\mathbf{y}_{(t,p)} = (y_{t-1}, \ldots, y_{t-p})^\top$, we have
$Y_t|\boldsymbol{\theta}, \mathcal{F}_{t-1} \overset{d}{=} Y_t|\boldsymbol{\theta}, \mathbf{y}_{(t,p)} \sim t(\mu_t, \sigma^2, \nu)$, where
$\mu_t = \mathbf{x}_t^\top \boldsymbol{\beta} + (\mathbf{y}_{(t,p)} - \mathbf{X}_{(t,p)}\boldsymbol{\beta})^\top \boldsymbol{\phi}$ and $\boldsymbol{\theta} = (\boldsymbol{\beta}^\top, \boldsymbol{\phi}^\top, \sigma^2, \nu)^\top$.

From Valeriano et al (2022+), conditionally on the first p observations, the observed log-likelihood function can be computed by

$$\ell(\boldsymbol{\theta}; \mathbf{y}_o) = \log\left(\int_{\mathbb{R}^m} \prod_{t=p+1}^{n} f(y_t|\boldsymbol{\theta}, y_{t-1}, y_{t-2}, \ldots, y_{t-p}) \mathrm{d}\mathbf{y}_m\right).$$

Letting $\mathbf{y} = (y_{p+1}, \ldots, y_n)^\top$ and $\mathbf{u} = (u_{p+1}, \ldots, u_n)^\top$ be hypothetical missing data, we have $\mathbf{y}_c = \left(\mathbf{y}^\top, \mathbf{u}^\top, \mathbf{V}^\top, \mathbf{C}^\top\right)^\top$, and therefore

$$\ell_c(\boldsymbol{\theta}; \mathbf{y}_c) = g(\nu|\mathbf{u}) - \frac{n-p}{2}\log\sigma^2 - \frac{1}{2\sigma^2}\sum_{i=p+1}^{n} u_i\left(\tilde{y}_i - \mathbf{w}_i^\top\boldsymbol{\phi}\right)^2 + \text{ cte},$$

with $g(\nu|\mathbf{u}) = \frac{n-p}{2}\left[\nu\log\left(\frac{\nu}{2}\right) - 2\log\Gamma\left(\frac{\nu}{2}\right)\right] + \frac{\nu}{2}\left(\sum_{i=p+1}^{n}\log u_i - \sum_{i=p+1}^{n} u_i\right)$,
$\tilde{y}_t = y_t - \mathbf{x}_t^\top\boldsymbol{\beta}$, and $\mathbf{w}_t = \mathbf{y}_{(t,p)} - \mathbf{X}_{(t,p)}\boldsymbol{\beta}$.

Residual analysis

- To perform residual analysis for the CAR(p) and CARt(p) models, we first impute the censored or missing observations.

- Then, we consider quantile residuals computed as

$$\widehat{r}_i = \Phi^{-1}\left(T_1\left(y_i; \widehat{\mu}_i, \widehat{\sigma}^2, \widehat{\nu}\right)\right), \quad i = p+1, \ldots n,$$

  where $\widehat{\mu}_i = \mathbf{x}_i^\top \widehat{\boldsymbol{\beta}} + (\mathbf{y}_{(i,p)} - \mathbf{X}_{(i,p)}\widehat{\boldsymbol{\beta}})^\top \widehat{\boldsymbol{\phi}}$, with $\widehat{\boldsymbol{\theta}}$ referring to the ML estimates of $\boldsymbol{\theta}$ obtained through the SAEM algorithm.

- For the CARt(p) model, the quantile residual is calculated only for the latest $n - p$ observations.

The package ARCensReg

The R package ARCensReg

- The package **ARCensReg** implemets an EM-type algorithm in R using S3 class, containing methods for estimating and predicting CAR(p) and CARt(p) models.
- It has generic R functions `print`, `summary`, `plot`, `residuals` and `predict` implemented.
- The main functions in the package are `ARCensReg()` and `ARtCensReg()`, which fit a CAR(p) and a CARt(p), respectively.

The R package ARCensReg

```
# devtools::install_github("fernandalschumacher/ARCensReg")
library(ARCensReg)
ARCensReg(cc, lcl = NULL, ucl = NULL, y, x, p = 1)
ARtCensReg(cc, lcl = NULL, ucl = NULL, y, x, p = 1)
```

where

- `cc` is a vector of censoring indicators.
- `lcl` and `ucl` are the lower and upper bounds on the censoring interval, with `NULL` indicating no-censored. Missing data can be handled by setting `lcl=-Inf` and `ucl=Inf`.
- `y` is the vector of responses.
- `x` is the design matrix.
- `p` is the autoregressive order.

Application

Application: Phosphorus concentration data

Following Wang & Chan (2018), to make the phosphorous concentration (P) and the water discharge (Q) relation more linear we considered the logarithmic transformation of P and Q and fitted the following model:

$$\log(P_t) = \sum_{j=1}^{4} [\beta_{0j} S_{jt} + \beta_{1j} S_{jt} \log(Q_t)] + \xi_t, \quad t = p+1, \ldots, n,$$

where the regression error ξ_t follows an autoregressive model and S_j is a dummy seasonal variable for quarters $j = 1, 2, 3,$ and 4.

CloudCeiling data

```
data(phosphorus)
phosphorus %>% glimpse()
```

```
## Rows: 181
## Columns: 5
## $ lP   <dbl> -1.6094379, -2.3025851, -2.3025851, -2.3025851, -0.5108256, -2.30~
## $ lQ   <dbl> 6.709060, 6.673930, 6.222576, 5.825115, 6.504288, 6.340007, 7.376~
## $ cc   <int> 0, 1, 1, 1, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0,~
## $ lcl  <dbl> -2.302585, -2.302585, -2.302585, -2.302585, -2.302585, -2.302585,~
## $ time <fct> 1998-10, 1998-11, 1998-12, 1999-01, 1999-02, 1999-03, 1999-04, 19~
```

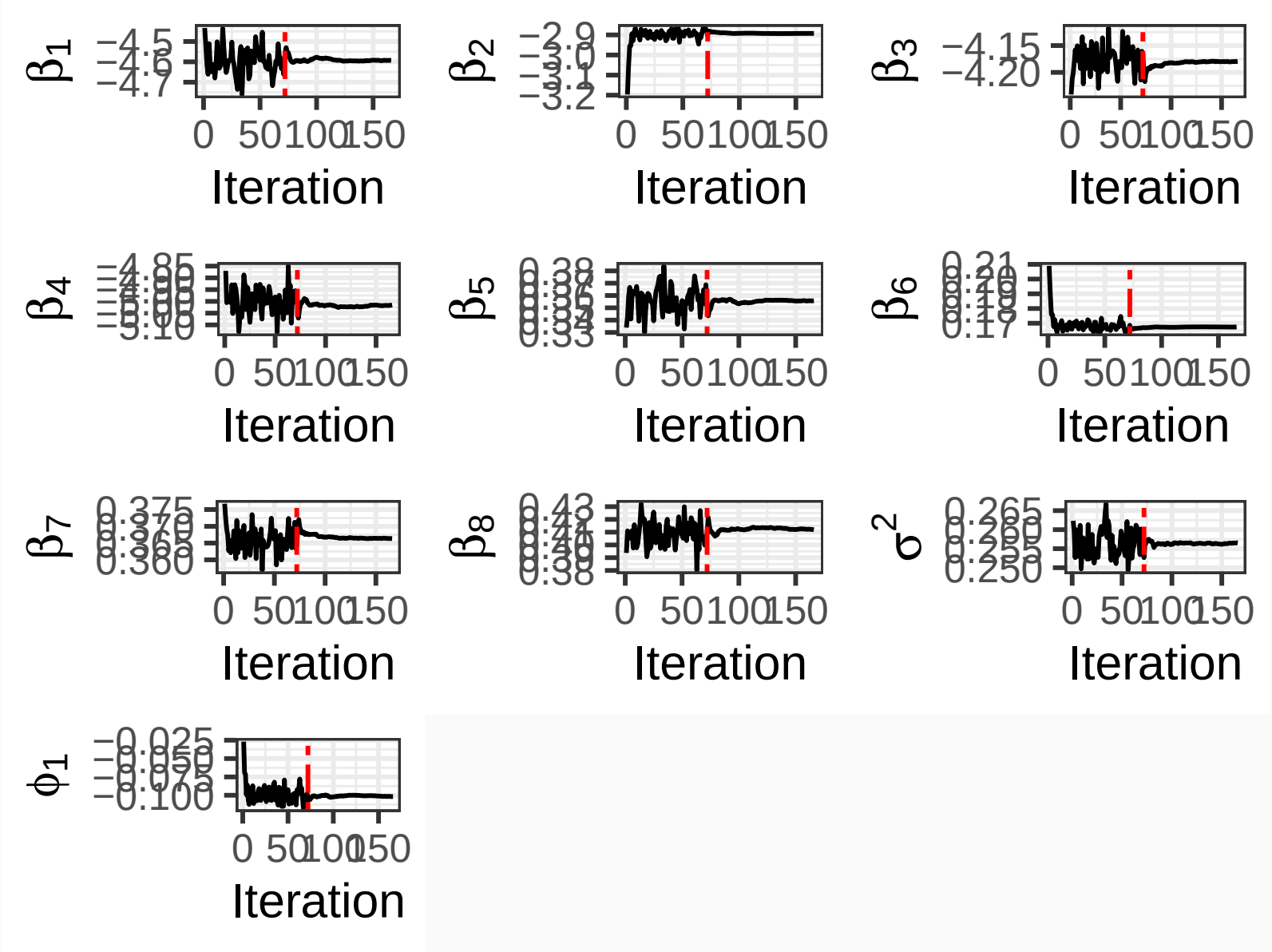
```
#
n <- nrow(phosphorus)
ucl <- phosphorus$lcl
# hanlding the missing data
miss <- which(is.na(phosphorus$lP))
phosphorus$cc[miss] <- 1
ucl[miss] <- Inf
phosphorus <- phosphorus %>% transform(month = substr(time, 6, 7))
phosphorus <- phosphorus %>% transform(quart = case_when(
                            month %in% c("01","02","03") ~ "1",
                            month %in% c("04","05","06") ~ "2",
                            month %in% c("07","08","09") ~ "3",
                            TRUE ~ "4"))
```

Example: CloudCeiling data

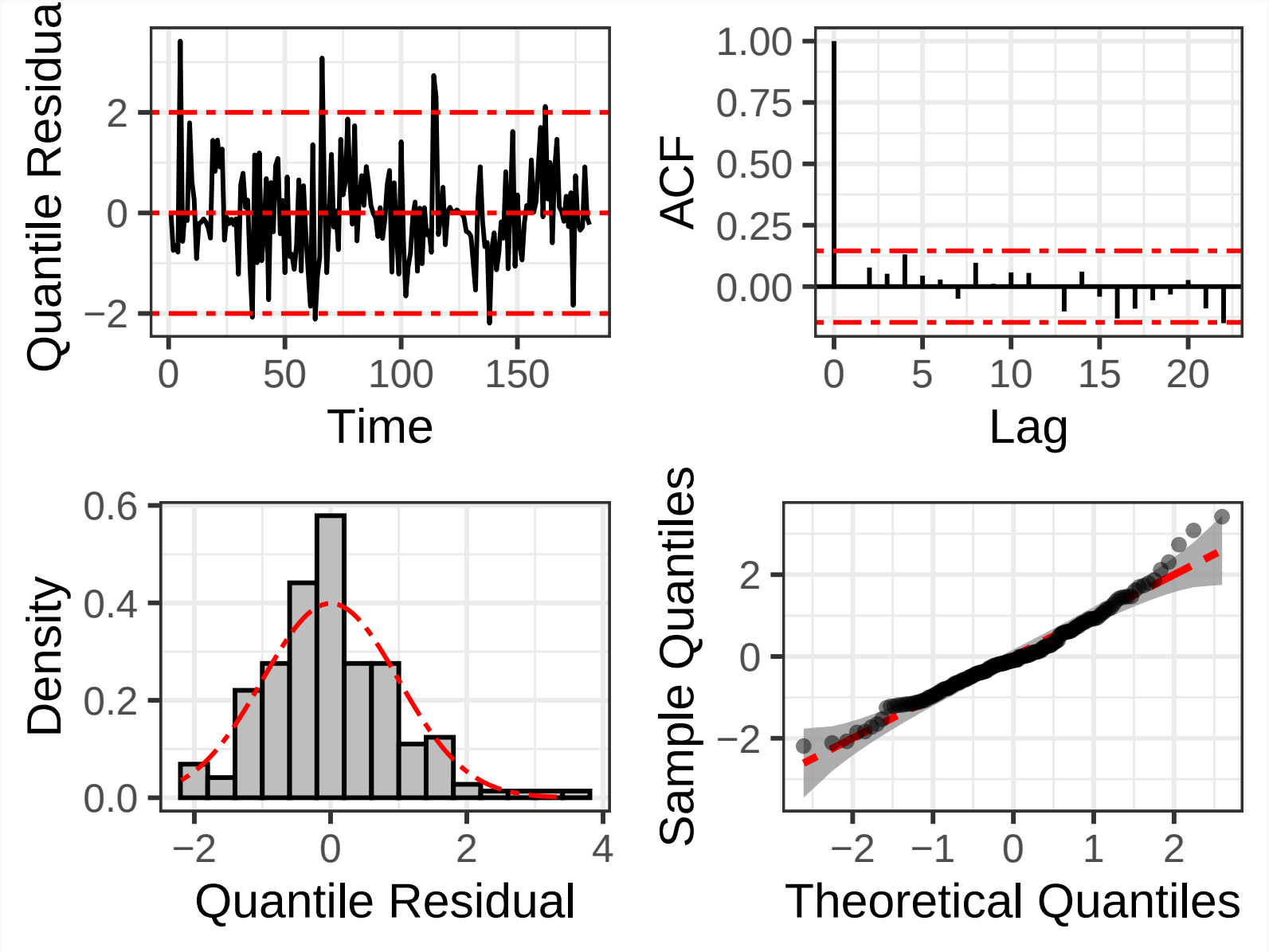
```
#fitting a CAR(1)
set.seed(8765)
car1<- ARCensReg(cc = phosphorus$cc, lcl = rep(-Inf, n), ucl = ucl, y = phosphorus$lP,
                 x = model.matrix(~quart+quart:lQ-1, data=phosphorus),
                 tol=.001, quiet = TRUE)
car1
```

```
## ------------------------------------------------
##   Censored Linear Regression Model with AR Errors
## ------------------------------------------------
## Call:
## ARCensReg(cc = phosphorus$cc, lcl = rep(-Inf, n), ucl = ucl,
##     y = phosphorus$lP, x = model.matrix(~quart + quart:lQ - 1,
##         data = phosphorus), tol = 0.001, quiet = TRUE)
##
## Estimated parameters:
##        beta1   beta2   beta3   beta4  beta5  beta6  beta7  beta8 sigma2    phi1
##      -4.5929 -2.8926 -4.1796 -5.0155 0.3557 0.1674 0.3663 0.4119 0.2565 -0.1017
## s.e.  0.4511  0.7252  0.4325  0.4829 0.0768 0.1011 0.0727 0.0882 0.0302  0.0858
##
## Details:
## Type of censoring: left
## Number of missing values: 7
## Convergence reached?:
## Iterations: 166 / 400
## MC sample: 10
## Cut point: 0.18
## Processing time: 52.69403 secs
```

```
plot(car1)
```



```
residuals(car1) %>% plot()
```

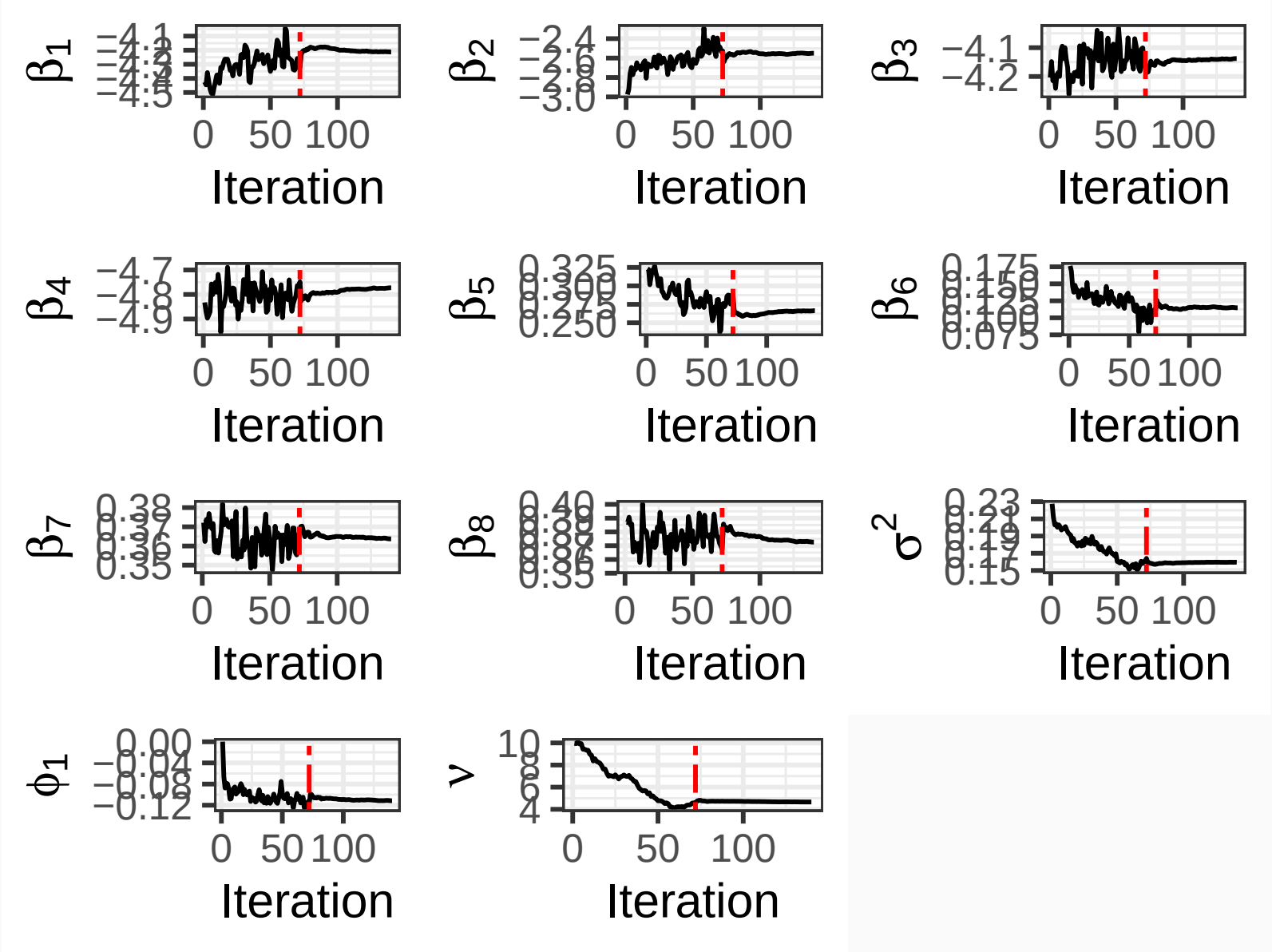


Example: CloudCeiling data

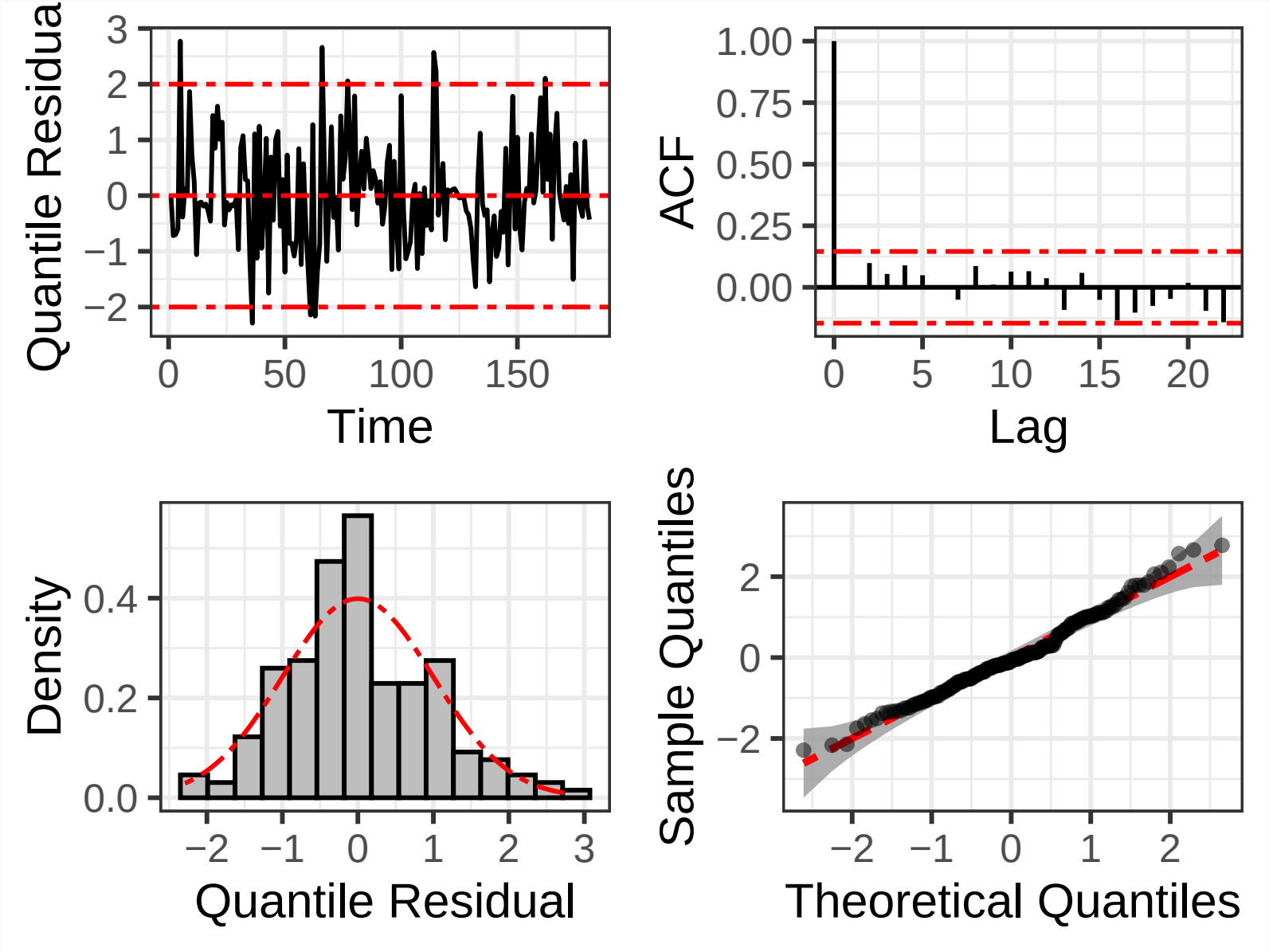
```r
#fitting a CARt(1)
set.seed(287399)
cart1<- ARtCensReg(cc = phosphorus$cc, lcl = rep(-Inf, n),ucl = ucl, y = phosphorus$lP,
                   x = model.matrix(~quart+quart:lQ-1, data=phosphorus),
                   tol=.001, M=15, quiet = TRUE)
cart1
```

```
## ------------------------------------------------
##   Censored Linear Regression Model with AR Errors
## ------------------------------------------------
## Call:
## ARtCensReg(cc = phosphorus$cc, lcl = rep(-Inf, n), ucl = ucl,
##     y = phosphorus$lP, x = model.matrix(~quart + quart:lQ - 1,
##         data = phosphorus), M = 15, tol = 0.001, quiet = TRUE)
##
## Estimated parameters:
##        beta1   beta2   beta3   beta4  beta5  beta6  beta7  beta8 sigma2    phi1      nu
##      -4.2125 -2.5502 -4.1377 -4.7728 0.2663 0.1148 0.3637 0.3726 0.1595 -0.1114  4.6598
## s.e.  0.4637  0.6959  0.3975  0.4180 0.0818 0.0967 0.0658 0.0769 0.0301  0.0846  1.7398
##
## Details:
## Type of censoring: left
## Number of missing values: 7
## Convergence reached?: TRUE
## Iterations: 140 / 400
## MC sample: 15
## Cut point: 0.18
## Processing time: 24.81226 secs
```

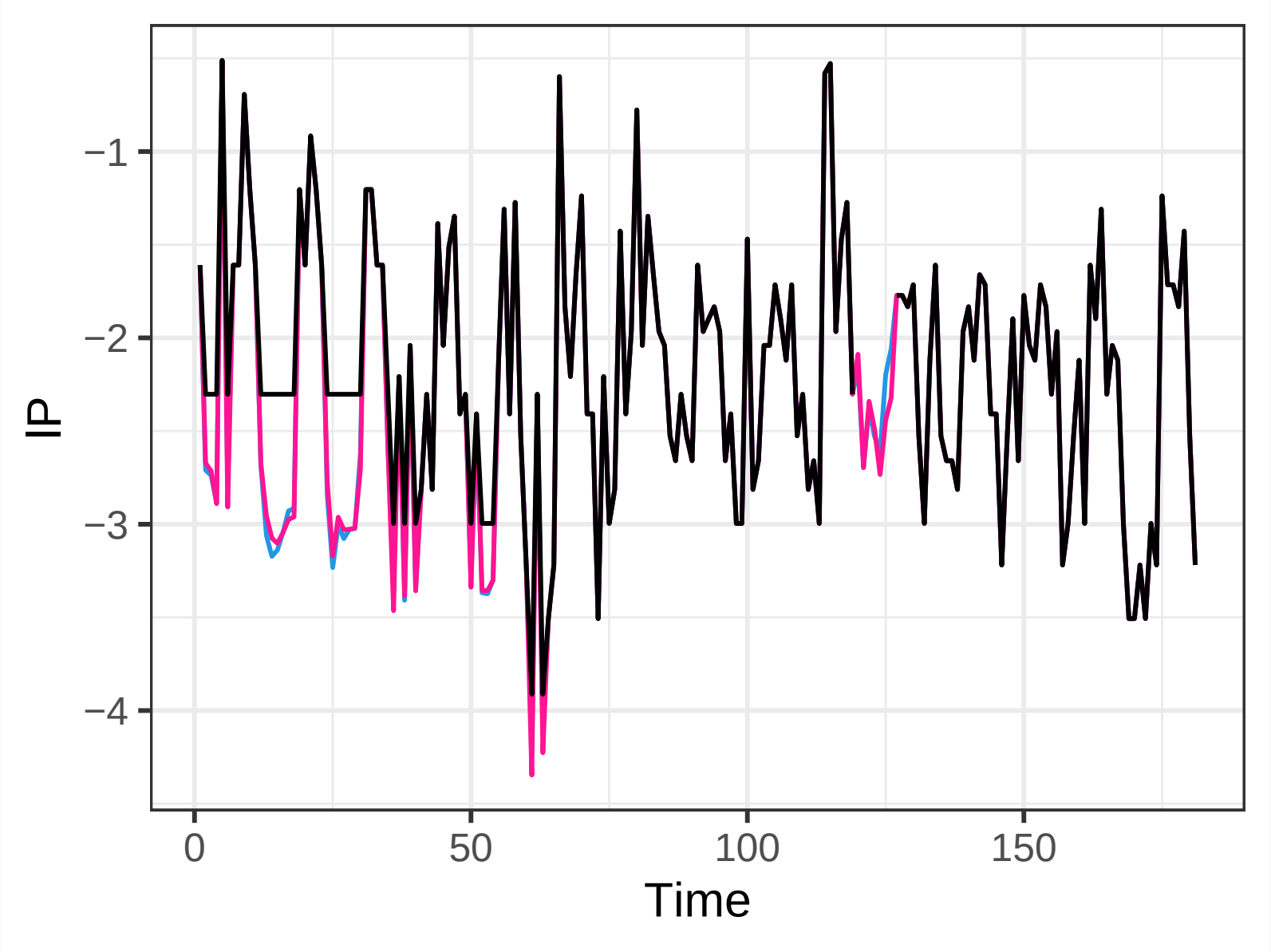
```
plot(cart1)
```



```
residuals(cart1) %>% plot()
```



Imputed censored and missing values



Prediction of future observations

```
xnew <- cart1$x[c(170,171),] %>% as.data.frame()
xnew$`quart4:lQ` <- c(4.7, 4.8)
predict(cart1, x_pred = xnew)
```

```
##            [,1]
## [1,] -3.002591
## [2,] -2.986267
```

Concluding remarks

Concluding remarks

- It is important to remark that we assumed the dropout/censoring mechanism to be missing at random (MAR). A possible venue for future work is to consider scenarios with informative censoring.

- Another interesting extension is to tackle the limitation of assuming that the first p observations are fully observed for the CARt(p) model.

References

- Delyon, B., Lavielle, M., & Moulines, E. (1999). Convergence of a stochastic approximation version of the EM algorithm. *The Annals of Statistics*, 27(1), 94–128.
- Garay, A., Lachos, V.H., Bolfarine, H., & Cabral, C. (2017). Linear Censored Regression Models with Scale Mixture of Normal Distributions. *Statistical Papers*, 58:247–278.
- Kuhn, E. & Lavielle, M. (2005). Maximum likelihood estimation in nonlinear mixed effects models. *Computational Statistics & Data Analysis*, 49(4), 1020–1038.
- Liu, J., Kumar, S., & Palomar, D.P. (2019). Parameter estimation of heavy-tailed AR model with missing data via stochastic EM. *IEEE Transactions on Signal Processing*, 67(8), 2159–2172.
- Schumacher, F.L., Lachos, V.H., & Dey, D.K. (2017). Censored regression models with autoregressive errors: A likelihood-based perspective. *Canadian Journal of Statistics*, 45(4), 375–392.
- Schumacher, F.L., Valeriano, K., Lachos, V.H., Galarza, C.E., & Matos, L.A. (2022). Package 'ARCensReg'. R package version 3.0.0.
- Valeriano, K.A., Schumacher, F.L., Galarza, C.E., Matos, L.A. (2022+). Censored autoregressive regression models with Student-t innovations. *arXiv preprint* arXiv:2110.00224.
- Wang, C. & Chan, K.-S. (2018). Quasi-likelihood estimation of a censored autoregressive model with exogenous variables. *Journal of the American Statistical Association*, 113(523), 1135–1145.

GitHub repository

Fernanda Lang Schumacher
Assistant Professor - Biostatistics
CPH - The Ohio State University
schumacher.313@osu.edu