VGAM Family Functions for Time Series

Beta Version 0.5-15

© Thomas W. Yee

Department of Statistics,
University of Auckland,
New Zealand
yee@stat.auckland.ac.nz
http://www.stat.auckland.ac.nz/∼yee

January 16, 2004

Contents

[Important note: This document and code is not yet finished, but should be completed one day ...]

1 Introduction

This document describes in detail VGAM family functions for time series. Classical time series analysis is a big topic and there is a huge literature. It is based on normal errors. Software is plentiful for the classical case.

This document describes two non-standard time series models. Firstly, in Section 2, software for fitting time series models to data coming from the exponential family. This is a large increase in generality. As implied below, the estimation of GLM-type time series naturally fits in within the VGAM framework. Secondly, in Section 3, reduced-rank vector autoregressive models are described.

Many of VGAM's features come from `glm()` and `gam()` so that readers unfamiliar with these functions are referred to Chambers and Hastie (1993). Additionally, the VGAM *User Manual* should be consulted for general instructions about the software.

2 The GARMA Model

This section draws heavily on Benjamin et al. (1998). Other references are Benjamin et al. (2003), Zeger and Qaqish (1988) and Li (1994). Li (1994) proposed the moving average version, while Zeger and Qaqish (1988) proposed the autoregressive version.

For $y_t,\ t = 1, \ldots, n$, the previous information set $\boldsymbol{D}_t = \{\boldsymbol{x}_t, \ldots, \boldsymbol{x}_1, y_{t-1}, \ldots, y_1, \mu_{t-1}, \ldots, \mu_1\}$ is assumed to belong to the exponential family

$$f(y_t|\boldsymbol{D}_t) \;=\; \exp\left\{\frac{y_t\theta_t - b(\theta_t)}{\phi/A_t} + c(y_t, \phi/A_t)\right\} \tag{1}$$

where θ_t and ϕ are the canonical and scale parameters respectively, and A_t are known prior weights. The mean $\mu_t \;=\; E(Y_t|\boldsymbol{D}_t) = b'(\theta_t)$ is related to the linear predictor η_t by the link function g.

A general model is given by

$$g(\mu_t) \;=\; \eta_t \;=\; \boldsymbol{x}_t^T\boldsymbol{\beta} + \tau_i$$

where the additional component τ_i allows autoregressive and moving average terms:

$$\tau_i \;=\; \sum_{k=1}^{p} \phi_k \mathcal{A}(y_{t-k}, \boldsymbol{X}_{t-k}, \boldsymbol{\beta}) + \sum_{k=1}^{q} \theta_k \mathcal{M}(y_{t-k}, \mu_{t-k}) \tag{2}$$

where $\mathcal{A}$ and $\mathcal{M}$ are functions representing the autoregressive and moving average terms respectively.

Model (2) is too general for practical applications. The model that the VGAM family function `garma()` fits is

$$g(\mu_t) \;=\; \eta_t \;=\; \boldsymbol{x}_t^T \boldsymbol{\beta} + \sum_{k=1}^{p} \phi_k \left\{ g(y_{t-k}) - \boldsymbol{x}_{t-k}^T \boldsymbol{\beta} \right\} + \sum_{k=1}^{q} \theta_k \left\{ g(y_{t-k}) - \eta_{t-k} \right\}. \tag{3}$$

Equations (1) and (3) together define the GARMA(p, q) model (GARMA stands for "generalized autoregressive moving-average models.")

Estimation and inference details are outlined in Benjamin et al. (1998). Here are some notes about the VGAM family function `garma()`:

1. For certain link functions g it may be necessary to replace y_{t-k} with some other value y^*_{t-k} to avoid numerical problems with $g(y_{t-k})$, e.g., $\log(0)$. The argument `constant` is for the log link: if $y = 0$ then y is replaced by some value c, where $0 < c < 1$ is `constant`.

2. Recursion is needed for the moving average term.This makes things a lot more complicated! For this reason, `garma()` currently only fits $q = 0$ models, i.e., autoregressive models only.

3. For binomial data the links are `logit`, `probit` and `cloglog`. For Poisson data the link is `loge` (for the natural logarithm.)

2.1 Other Topics

2.1.1 Input

The response is assumed continuous, counts or binary. The 'right' link function is chosen to reflect this. Input for the binary case is the same as `Binomial()`.

2.1.2 Convergence

Currently, initialization is quite poor, and could be improved. A limited amount of experience has shown that half-stepsizing is often needed for convergence, therefore choosing `crit="coef"` is not recommended.

Successful convergence often requires very good initial values. The argument `coefstart` in `garma()` can be used for this purpose; the argument `coefstart` in `vglm()` cannot be used because of technical reasons.

`garma()` should be used with `vglm()` rather than `vgam()` because of the step-sizing feature.

2.1.3 Over-dispersion

Scale parameters have not been implemented yet.

3 Reduced Rank Regression for Time Series

Consider the multivariate autoregressive AR(L) model

$$\boldsymbol{Y}_t \;=\; \sum_{j=1}^{L} \boldsymbol{\Phi}_j \boldsymbol{Y}_{t-j} + \boldsymbol{\varepsilon}_t, \qquad \boldsymbol{\varepsilon}_t \sim (\boldsymbol{0}, \boldsymbol{\Omega}) \text{ independently}, \quad t = 1, \ldots, n, \tag{4}$$

where $\boldsymbol{Y}_t$ is $M \times 1$ and $\boldsymbol{\Phi}_j$ is $M \times M$ and to be estimated. As the number of lags L increases the number of parameters involved grows rapidly. Ahn and Reinsel (1988) proposed the *nested reduced-rank autoregressive model* where $\boldsymbol{\Phi}_j$ is replaced by a matrix of rank r_j. One has the r_js being a nonincreasing sequence and $\boldsymbol{\Phi}_j = \mathbf{A}_j \mathbf{C}_j^T$ and $\mathrm{range}(\mathbf{A}_j) \supset \mathrm{range}(\mathbf{A}_{j+1})$. It is nested because with $r_j \equiv r < M$ and $\mathbf{A}_j = \mathbf{A}$ the special model $\boldsymbol{Y}_t = \mathbf{A} \sum_{j=1}^{L} \mathbf{C}_j^T \boldsymbol{Y}_{t-j} + \boldsymbol{\varepsilon}_t$ is obtained. Ahn and Reinsel (1988) gave a canonical form, computational details. and showed that a Newton-Raphson-like algorithm could be implemented using standard software for generalized least squares regression.

Model (4) lies out of the RR-VGLM framework because *each* $\boldsymbol{\Phi}_j$ is of reduced rank, not the combined matrix $(\boldsymbol{\Phi}_1^T, \boldsymbol{\Phi}_2^T, \ldots, \boldsymbol{\Phi}_L^T)^T$ ($= \boldsymbol{\Phi}_*$, say). Nevertheless, the VGAM family function `rrar()` has been written to implement this model. It takes in a $n \times M$ matrix response and the explanatory variables should just be an intercept term. The argument `Ranks` in `rrar()` specifies the ranks and must be of length L. See the tutorial example below. For more information about VGAM family functions for reduced-rank regression, and notations, see the other documentation.

In practice $\boldsymbol{\Omega}$ has to be estimated. We use

$$n^{-1} \sum_{t=1}^{n} \hat{\boldsymbol{\varepsilon}}_t \, \hat{\boldsymbol{\varepsilon}}_t^T.$$

3.1 Other Topics

3.1.1 Notation

Ahn and Reinsel (1988) use $\mathbf{B}_j$ instead of $\mathbf{C}_j^T$ here.

3.1.2 Input

The response is a $n \times M$ matrix. Any explanatory variables are ignored (except for the intercept term).

3.1.3 Convergence

Currently, initialization uses random numbers and is quite poor. For reproducibility of results use `set.seed()`.

Convergence is slow (much slower than a second order rate) because $\boldsymbol{\Omega}$ is estimated. The default of `rrar()` is to take a half-step instead of an ordinary full Newton-Raphson step—we found this necessary for the grain data below.

4 Tutorial Examples

4.1 GARMA

We look at the interspike data described in Zeger and Qaqish (1988). It concerns the interspike times collected from neurons in the motor cortex of a monkey. Zeger and Qaqish (1988) used an inverse link function but we will use a log link; the results are similar.

```
> interspike
  [1]  68  41  82  66 101  66  57  41  27  78  59  73   6  44  72  66  59  60
 [19]  39  52  50  29  30  56  76  55  73 104 104  52  25  33  20  60  47   6
 [37]  47  22  35  30  29  58  24  34  36  34   6  19  28  16  36  33  12  26
 [55]  36  39  24  14  28  13   2  30  18  17  28   9  28  20  17  12  19  18
 [73]  14  23  18  22  18  19  26  27  23  24  35  22  29  28  17  30  34  17
 [91]  20  49  29  35  49  25  55  42  29  16
> plot(interspike, ylim=c(0,120), las=1, font=1, xlab="Spike Number",
+    ylab="Inter-Spike Time ( ms )")
> fit = vglm(interspike ~ 1, garma("loge", p.ar.lag=2, coef=c( 4, 0.3, 0.4)),
+            trace=TRUE, crit="c")
> lines(spikenum[-(1:fit@misc$plag)], fitted(fit))
> abline(h=mean(interspike))
> summary(fit)

Call:
vglm(formula = interspike ~ 1, family = garma("loge", p.ar.lag = 2, coef = c(4,
```

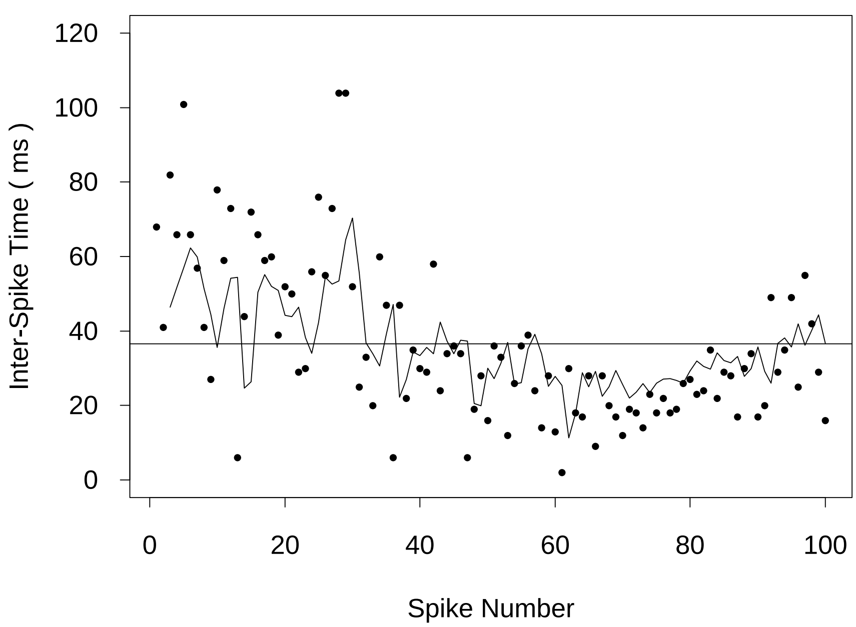

Figure 1: *Observed and predicted interspike times from a second-order Markov model. The horizontal line is the predicted level from a GLM without time dependence.*

```
        0.3, 0.4)), trace = TRUE, crit = "c")

Pearson Residuals:
            Min      1Q   Median    3Q    Max
log(mu) -6.5604 -2.1546 -0.45264 1.231 8.8819

Coefficients:
              Value Std. Error t value
(Intercept) 3.71331   0.041170 90.1944
     (lag1) 0.33723   0.030762 10.9626
     (lag2) 0.24180   0.029874  8.0939

Number of linear predictors:  1

Name of linear predictor: log(mu)

Dispersion Parameter for garma family:   1

Log-likelihood: 9330.662 on 95 degrees of freedom

Number of Iterations: 19
```

The plot of the data and fitted values is given in Fig. 1. This looks similar to Figure 4 of Zeger and Qaqish (1988). In comparison, Zeger and Qaqish (1988) obtained 0.0217, 0.336 and 0.283 as their regression parameters. They had t-statistics of 8.22, 3.97 and 3.51 respectively.

4.2 Nested AR

We mimic the analysis presented in Ahn and Reinsel (1988) who considered data consisting of monthly averages of grain prices in the United States for wheat flour, corn, wheat and rye for the period January 1961 through October 1972. The units are dollars per 100 pound sack for wheat flour, and per bushel for corn, wheat and rye. The model

$$\boldsymbol{Y}_t \;=\; \boldsymbol{\Phi}_1 \boldsymbol{Y}_{t-1} + \boldsymbol{\Phi}_2 \boldsymbol{Y}_{t-2} + \boldsymbol{\varepsilon}_t$$

was fitted, where $\mathrm{rank}(\boldsymbol{\Phi}_1) = 4$ and $\mathrm{rank}(\boldsymbol{\Phi}_2) = 1$ was chosen. This lag $L = 2$ model can be parameterized

$$\boldsymbol{Y}_t \;=\; \mathbf{A}_{k_1} \left(\mathbf{C}_1^T \boldsymbol{Y}_{t-1} + \mathbf{D}_2 \mathbf{C}_2^T \boldsymbol{Y}_{t-2} \right) + \boldsymbol{\varepsilon}_t.$$

We have

```
> usagrain = read.table("../data/usagrain.txt", header=T)
> year = seq(1961+1/12, 1972+10/12, by=1/12)
> par(mar=c(4,4,2,2)+0.1, mfrow=c(4,1))
> for(i in 1:4) {
+     plot(year, usagrain[,i], main=names(usagrain)[i], type="l", xlab="", ylab="")
+     points(year, usagrain[,i], pch="*")
+ }
```

This produces Fig. 2. Now the results of Ahn and Reinsel (1988) can be obtained by as follows.

```
> apply(usagrain, 2, mean)     # mu vector
 wheat.flour   corn wheat    rye
      6.8498 1.2513 1.543 1.1641
> cgrain = scale(usagrain, scale=F)   # Center the time series only
> fit = vglm(cgrain ~ 1, rrar(Ranks=c(4,1)))
> summary(fit)

Call:
vglm(formula = cgrain ~ 1, family = rrar(Ranks = c(4, 1)))

Pearson Residuals:
       Min     1Q Median    3Q  Max
[1,] -3.03 -0.348 0.1269 0.506 3.85
[2,] -2.80 -0.609 0.0702 0.643 2.75
[3,] -3.48 -0.419 0.0348 0.623 2.61
[4,] -2.98 -0.504 0.1214 0.566 3.38

Coefficients:
     Value Std. Error t value
 1 -0.0169     0.0630  -0.267
 2  0.3269     0.1120   2.920
 3  0.1912     0.0813   2.350
 4  0.9854     0.0721  13.667
 5 -0.4117     0.2823  -1.458
 6  0.5766     0.1741   3.311
 7 -0.4539     0.2457  -1.847
 8  0.0270     0.0591   0.457
 9  0.7948     0.0540  14.720
10  0.0558     0.0337   1.652
11 -0.0170     0.0531  -0.320
12 -0.2885     0.1049  -2.749
13 -0.1184     0.0959  -1.235
14  0.8235     0.0599  13.737
15  0.0462     0.0943   0.490
16 -0.1999     0.0762  -2.622
17 -0.0995     0.0697  -1.428
18  0.0449     0.0436   1.030
19  0.8089     0.0685  11.808
20 -0.0797     0.0698  -1.142
21  0.9261     0.2743   3.376
22 -0.6211     0.1689  -3.676
23  0.2350     0.2259   1.041

Number of linear predictors:  4

Dispersion Parameter for rrar family:   1

Number of Iterations: 20

> print(fit@misc$Ak1, dig=2)
       [,1] [,2] [,3] [,4]
[1,]  1.000    0    0    0
[2,] -0.017    1    0    0
```

```
[3,]  0.325    0    1    0
[4,]  0.191    0    0    1
> print(fit@misc$Cmatrices, dig=3)
[[1]]:
       [,1]    [,2]    [,3]    [,4]
[1,]  0.986  0.0341 -0.2778 -0.1961
[2,] -0.415  0.7989 -0.1078 -0.0968
[3,]  0.575  0.0560  0.8300  0.0458
[4,] -0.449 -0.0187  0.0499  0.8110

[[2]]:
        [,1]
[1,] -0.0812
[2,]  0.9275
[3,] -0.6207
[4,]  0.2262

> print(fit@misc$Dmatrices, dig=3)
[[1]]:
     [,1] [,2] [,3] [,4]
[1,]    1    0    0    0
[2,]    0    1    0    0
[3,]    0    0    1    0
[4,]    0    0    0    1

[[2]]:
     [,1]
[1,]    1
[2,]    0
[3,]    0
[4,]    0
> print(fit@misc$omegahat, dig=3)
            wheat.flour     corn    wheat       rye
wheat.flour    2.49e-02 0.000185 0.003836 -7.54e-05
       corn    1.85e-04 0.002213 0.000989  7.77e-04
      wheat    3.84e-03 0.000989 0.006890  1.51e-03
        rye   -7.54e-05 0.000777 0.001511  2.77e-03
> print(fit@misc$Phimatrices, dig=2)
[[1]]:
        [,1]  [,2]  [,3]    [,4]
[1,]  0.9864 -0.42 0.576 -0.4566
[2,]  0.0105  0.80 0.047 -0.0099
[3,]  0.0385 -0.25 1.020 -0.1010
[4,] -0.0082 -0.17 0.160  0.7236

[[2]]:
        [,1]   [,2]  [,3]    [,4]
[1,] -0.0800  0.929 -0.62  0.2365
[2,]  0.0013 -0.015  0.01 -0.0039
[3,] -0.0260  0.302 -0.20  0.0769
[4,] -0.0153  0.177 -0.12  0.0451
```

The results differ slightly from Ahn and Reinsel (1988), possibly because we have used $t = 1 + L, \ldots, n$

instead of $t = 1, \ldots, n$. That is, we have ignored the first L observations for simplicity.

Finally, the canonical variables are also returned. This transformed series $\boldsymbol{Z}_t$ whose components Z_{it} are arranged such that with increasing values of the index i, less past information on $\boldsymbol{Z}_t$ is necessary to explain the present value of subvector Z_{it}. They may be plotted as follows.

```
par(mar=c(4,4,2,2)+0.1, mfrow=c(4,1))
for(i in 1:4) {
    plot(year, fit@misc$Z[,i], main=paste("Z", i, sep=""),
         type="l", xlab="", ylab="")
    points(year, fit@misc$Z[,i], pch="*")
}
```

This gives Fig. 3. The y-limits do not correspond to the original series because it was centered. The y-limits corresponding to the original series can be obtained by using `(usagrain %*% t(solve(fit@misc$Ak1)))[,i]` instead. See Ahn and Reinsel (1988) for more information.

Exercises

1. Zeger and Qaqish (1988) fit spike number as a covariate (they called it “Model B”). Try doing this, both with a log link and inverse link.

2. [Hard] Extend `garma()` to handle moving average terms. Note that this will require recursion.

3. For the grain data, the full model can be specified by `rrar(Ranks=c(4,4))`. Fit this model. Is there much agreement between this full-rank model and the reduced-rank model? Show that the standard errors of the elements of $\hat{\boldsymbol{\Phi}}_1$ and $\hat{\boldsymbol{\Phi}}_2$ are approximately 25% smaller than those of the full model. Hint: use the result that $\sqrt{n}(\hat{\boldsymbol{\phi}} - \boldsymbol{\phi})$ converges in distribution to $N(\mathbf{0}, \mathbf{H}\,\mathcal{I}(\boldsymbol{\beta})^{-1}\mathbf{H}^T)$ where $\mathbf{H}$ is the matrix returned in `@misc$Hmatrix` and $\boldsymbol{\phi} = (\boldsymbol{\phi}_1^T, \ldots, \boldsymbol{\phi}_L^T)^T$ with $\boldsymbol{\phi}_j = \mathrm{vec}(\boldsymbol{\Phi}_j^T)$. Hint: use `mean(se)/mean(sef)` where `sef` is the vector of standard errors of the full model etc.

References

Ahn, S., Reinsel, G. C., 1988. Nested reduced-rank autoregressive models for multiple time series. Journal of the American Statistical Association 83, 849–856.

Benjamin, M. A., Rigby, R. A., Stasinopoulos, M. D., 1998. Fitting non-Gaussian time series models. In: Payne, R., Green, P. J. (Eds.), Proceedings in Computational Statistics COMPSTAT 1998. Physica-Verlag, pp. 191–196.

Benjamin, M. A., Rigby, R. A., Stasinopoulos, M. D., 2003. Generalized autoregressive moving average models. Journal of the American Statistical Association 98, 214–223.

Chambers, J. M., Hastie, T. J. (Eds.), 1993. Statistical Models in S. Chapman & Hall, New York.

Li, W. K., 1994. Time series models based on generalized linear models: some further results. Biometrics 50, 506–511.

Zeger, S. L., Qaqish, B., 1988. Markov regression models for time series: a quasi-likelihood approach. Biometrics 44, 1019–1031.

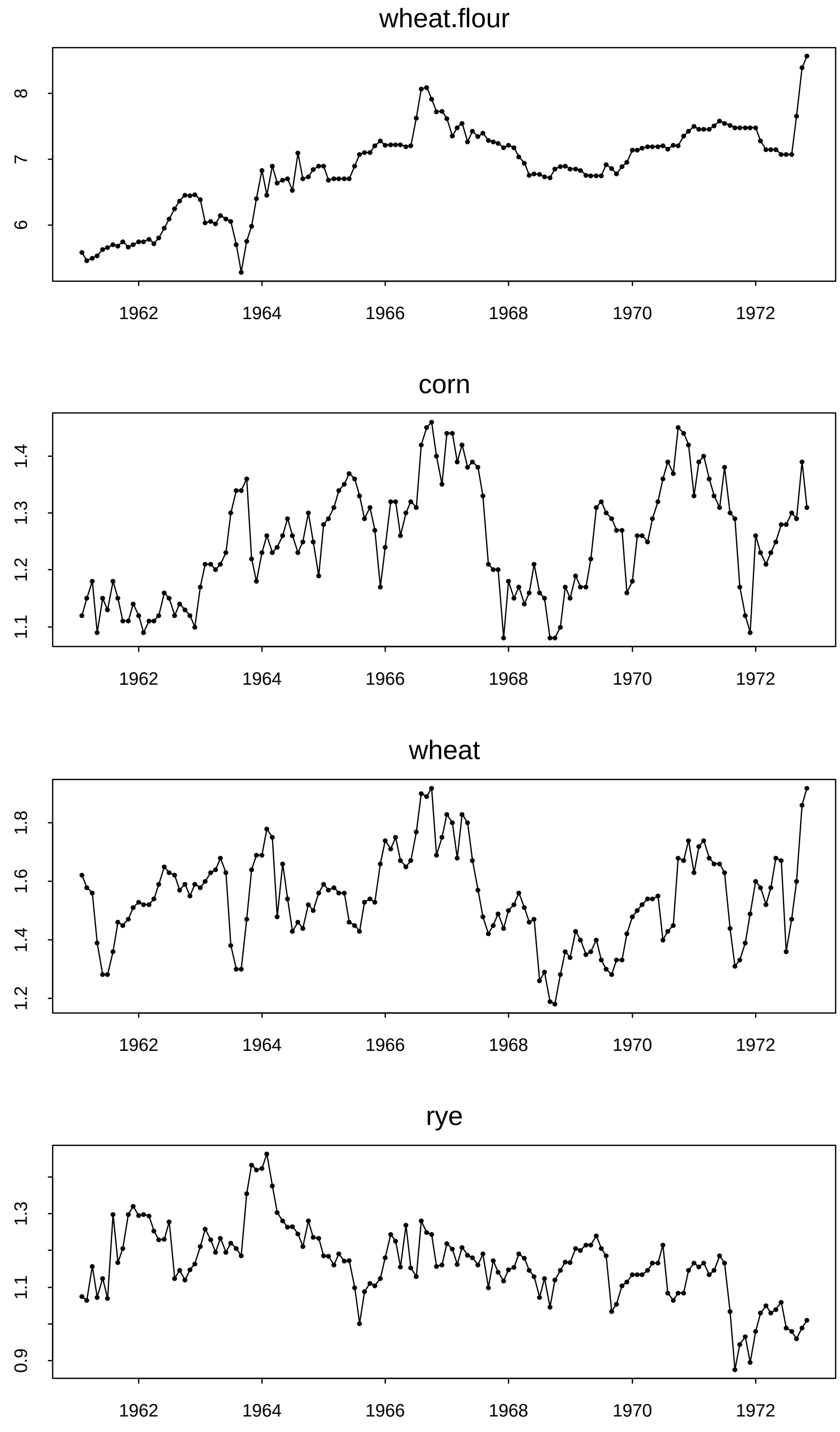

Figure 2: *Monthly average prices of Grain series, January 1961–October 1972.*

Figure 3: *Canonical variables of the Grain Price series, January 1961–October 1972.*