Application 4: Automated Model Selection

Dave Lorenz

July 26, 2017

This example illustrates the the automated model selection function. The example calibration data set include 4 constituents that include uncensored, Atra (atrazine); slightly censored, Alach (alachlor); highly censored, Buty (butlyate); and data having missing observations, SuspSed (suspended sediment).

The data used in this application were collected from the White River at Hazleton, Indiana. Forty five water-quality samples collected between October 1992 and September 1994 are used in conjunction with observed streamflow data to build the regression model.

Part 2 illustrates the seasonal-wave approach to modeling the seasonal pattern of pesticides. The user may wish to explore the seasonal-wave analysis for atrazine and alachor, both of which indicate a strong periodic seasonal lack of fit when using only the sine and cosine seasonl terms.

```
> # Load the rloadest package and the data
> library(rloadest)
> data(app4.calib)
> head(app4.calib)
```

```
       DATES TIMES  FLOW Buty.rmk  Buty Atra Alach.rmk Alach SuspSed
1 1992-10-19  1200  4290          0.004 0.28           0.016    90.5
2 1992-11-18  1145 31400          0.004 0.34           0.047   161.0
3 1992-12-21  1130  7770        < 0.002 0.18           0.026      NA
4 1993-01-12  1145 39700          0.009 0.23           0.047      NA
5 1993-02-18  1300  9310        < 0.002 0.10           0.013      NA
6 1993-03-30  1400 20900        < 0.002 0.15           0.034    98.0
```

The censored data in `app4.calib` are stored in 2 columns—the recorded value in the column identified by the constituent abbreviation and the remark code, "<" indicating a less-than value in the corresponding column with the .rmk suffix. In order to take advantage of the of `loadReg` function to automatically recognize censored data, the censored data in the example dataset should be converted to type "water-quality." This can be done using the `convert2qw` function in the `smwrQW` package, which is required by

rloadest. The naming convention in app.calib is consistent with the "partial" scheme for convert2qw. The conversion is accomplished in the R code below. The app4.calib is overwritten with the new data.

```
> # Convert Buty and Alach to class "qw"
> app4.calib <- convert2qw(app4.calib, scheme="partial")
> head(app4.calib)
```

```
       DATES TIMES  FLOW Atra SuspSed   Buty  Alach
1 1992-10-19  1200  4290 0.28    90.5  0.004  0.016
2 1992-11-18  1145 31400 0.34   161.0  0.004  0.047
3 1992-12-21  1130  7770 0.18      NA <0.002  0.026
4 1993-01-12  1145 39700 0.23      NA  0.009  0.047
5 1993-02-18  1300  9310 0.10      NA <0.002  0.013
6 1993-03-30  1400 20900 0.15    98.0 <0.002  0.034
```

1 Build the Model

The `selBestModel` function can be used to select the predefined model with the smallest AIC value. It also records the SPPC value, also known as BIC, which has a greater penalty for additional terms and will generally choose a more parsimonious model. The requirements for `selBestModel` are similar to `loadReg`, except that the name of the constituent is requires instead of the formula.

```
> # Create the "best" load model.
> app4.lr <- selBestModel("Buty", data = app4.calib, flow = "FLOW",
+                     dates = "DATES", conc.units="ug/L",
+                     station="White River at Hazleton, Ind.")
> # Print the warning in the vignette
> warnings()

NULL

> # Print the results
> app4.lr

*** Load Estimation ***

Station: White River at Hazleton, Ind.
Constituent: Buty

           Number of Observations: 45
Number of Uncensored Observations: 22
           Center of Decimal Time: 1993.8
                  Center of ln(Q): 9.4824
                Period of record: 1992-10-19 to 1994-09-12

Model Evaluation Criteria Based on AMLE Results
-----------------------------------------------

  model   AIC  SPCC  AICc
1     1 119.1 124.5 119.6
2     2 119.8 127.0 120.8
3     3 118.9 126.1 119.9
4     4 112.5 121.5 114.0
5     5 120.3 129.3 121.8
6     6 114.5 125.3 116.7
7     7 110.7 121.5 112.9
8     8 111.9 124.6 115.0
9     9 113.4 127.8 117.4
Model # 7 selected
```

```
Selected Load Model:
--------------------

Buty ~ model(7)

Model coefficients:
            Estimate Std. Error z-score p-value
(Intercept)  -2.9624     0.3186  -9.298  0.0000
lnQ           1.8007     0.3561   5.057  0.0000
DECTIME      -0.9432     0.5333  -1.769  0.0501
sin.DECTIME   0.6485     0.3617   1.793  0.0706
cos.DECTIME  -0.9413     0.4083  -2.305  0.0098

AMLE Regression Statistics
Residual variance: 1.76
Generalized R-squared: 52.56 percent
G-squared: 33.56 on 4 degrees of freedom
P-value: <0.0001
Prob. Plot Corr. Coeff. (PPCC):
  r = 0.9377
  p-value = 0.0168
Serial Correlation of Residuals: 0.4562

Variance Inflation Factors:
              VIF
lnQ         1.434
DECTIME     1.077
sin.DECTIME 1.100
cos.DECTIME 1.330

Comparison of Observed and Estimated Loads
------------------------------------------
      Summary Stats: Loads in kg/d
-------------------------------------------------
    Min  25%  50%  75%  90%  95%  Max
Est   0 0.04 0.16 0.49 0.89 0.95 1.88
Obs   0 0.02 0.06 0.40 0.93 1.03 2.11

Bias Diagnostics
----------------
 Bp: 23.46 percent
PLR: 1.235
  E: 0.01777
```

The model with the lowest AIC value is 7, which includes linear flow,

linear time and the seasonal time terms. Model 7 also has the smallest SPCC value, but model 4 is only slightly larger (requires 2 decimal digits to see the difference recorded in the `model.eval` component in the output object).

2 Diagnostic Plots

The rloadest package contains a `plot` function that creates diagnostic plots of the load model. Most often the user will just enter `plot(app4.lr)` (for this example) in the R Console window to generate the full suite of plots, but this example application will generate each plot individually. And, in general, the user will not need to set up a graphics device. But for this vignette, the graphics device must be set up for each graph.

Figure 1 shows the response versus the fitted values, censored observations are shown by open circles. It is perhaps a bit disconcerting that there are no censored observations below the dashed regression line for fitted values less than about -3, but that is not inconceivable. That discrepancy in the location of censored values also drives the appearance of greater scatter in larger fitted values; plots 2 and 3 are not shown.

```
> # setSweave is required for the vignette.
> setSweave("app4_01", 5, 5)
> plot(app4.lr, which=1, set.up=FALSE)
> graphics.off()
```

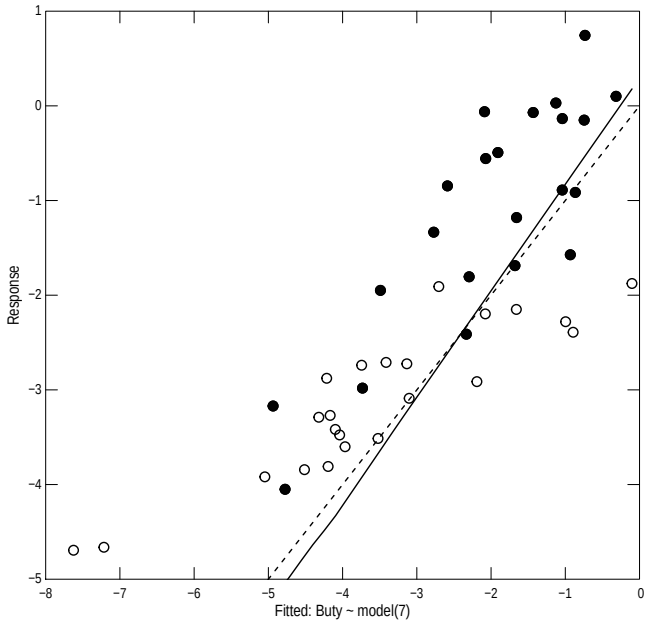

Figure 1. The rating-curve regression model.

Figure 2 is the correlogram. It suggests a seasonal lack of fit, although the shape of the smooth line is not conclusive.

```
> # setSweave is required for the vignette.
> setSweave("app4_02", 5, 5)
> plot(app4.lr, which=4, set.up=FALSE)
> graphics.off()
```

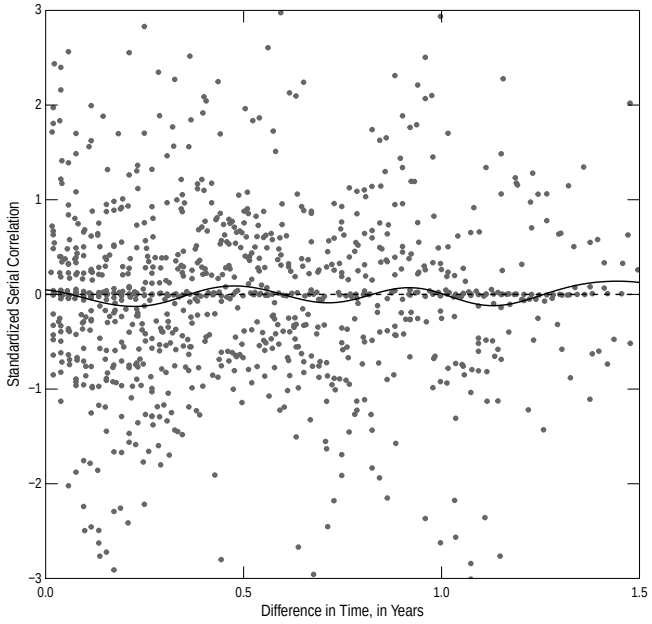

Figure 2. The correlogram.

Figure 3 is q-normal plot and shows the standardized residuals, which are assumed to have a standard deviation of 1. The solid line is the theoretical fit of mean of 0 and standard deviation of 1. The visual appearance of figure 5 confirms the results of the PPCC test in the printed output—the residuals show deviation from the normal distribution. The open circles are censored observations and the plotted value is the expected value returned by the `residuals` functions using the argument `type` set to "working."

```
> # setSweave is required for the vignette.
> setSweave("app4_03", 5, 5)
> plot(app4.lr, which=5, set.up=FALSE)
> graphics.off()
```

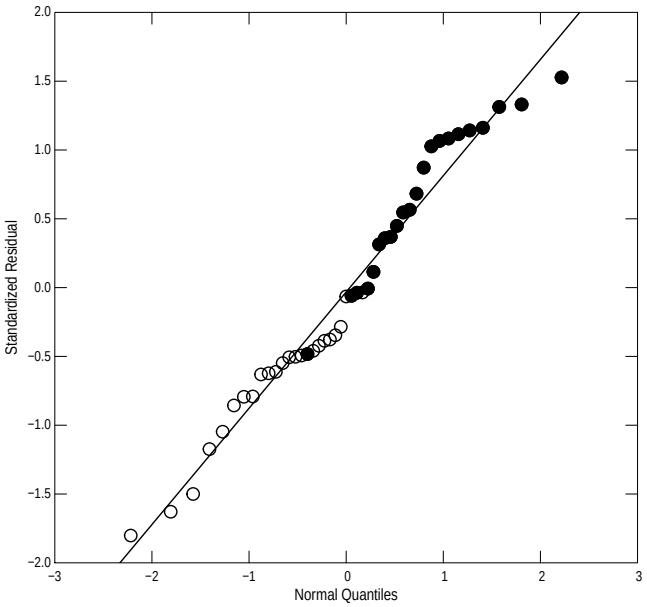

Figure 3. The Q-normal plot of the residuals.

Figure 4 is an extended box plot—a truncated box plot, at the 5 and 95 percentiles that shows the individual values larger than the 95th percentile and smaller than the 5th percentile. The box plots in figure 6 show the distributions of the actual and estimated loads. The appearance of these box plots helps to understand the printed bias diagnostics, which show a general over estimate. The upper and lower ends of both boxes are similar, but the box plot of the estimates shows a median value which is much larger than the median value for the observed values. Therefore, the middle range is over estimated.

```
> # setSweave is required for the vignette.
> setSweave("app4_04", 5, 5)
> plot(app4.lr, which=6, set.up=FALSE)
> graphics.off()
```

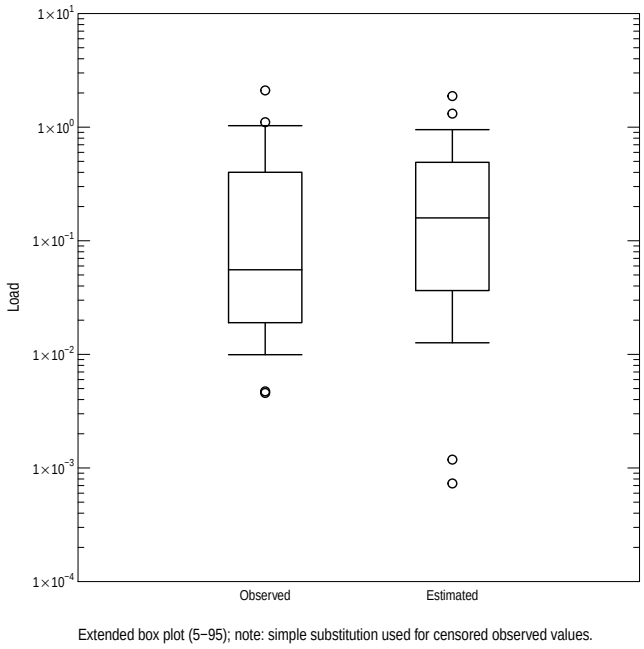

Figure 4. Box plot comparing estimated and observed values.

3 Part 2, Building a Seasonal-wave Load Model

All of the diagnostic plots in the previous section indicated a cause for concern about the validity of the periodic regression model. Vecchia and others (2008) describe a seasonal-wave function that often works well for pesticide models.

The smwrStats package contains the tools necessary to construct a seasonal-wave model. Building a good regression model is a multi-step process, required identifying the timing of the peak concentration and the other parameters of the seasonal-wave model.

The first step in constructing the seasonal-wave model is the identify the peak and potential values for the other parameters of the model. That involves building a regression model without any seasonal terms, and using the `seasonalPeak` function on the residuals to construct a first guess on those parameters. In this case, because there are censored values, we must use `censReg`. Note that it does not matter whether we use load or concentration because the residuals are the same.

```
> # Create the limited regression model.
> app4.cr <- censReg(Buty ~ center(log(FLOW)) + dectime(DATES),
+                    data = app4.calib, dist="lognormal")
> app4.sp <- seasonalPeak(dectime(app4.calib$DATES), residuals(app4.cr))
> app4.sp

Unconfirmed seasonal peak:
Default value: 0.393
Alternate values: 0.397 0.911
```

The next step in constructing the seasonal-wave model is to confirm the peak. This step requires the `confirm` function, which is interactive and cannot be demonstrated in a vignette. In this case, we can accept the default selection and estimated parameters. The user should step through the interactive process.

```
> # Show the plot for this example
> setSweave("app4_05", 5, 5)
> confirm(app4.sp, plot.only=TRUE)
> graphics.off()
> # Confirm the seasonalPeak analysis for a single peak.
> app4.sp <- confirm(app4.sp, all=TRUE)
> app4.sp

Confirmed seasonal peak:
Number of peaks: 1
Time of peak: 0.393
```

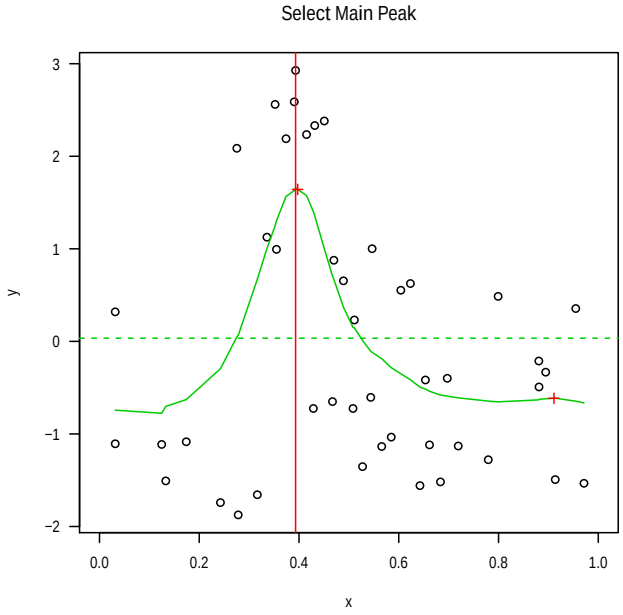

Figure 5. The seasonal peak graph.

The `selBestWave` function can be used to select the "best" parameters for the seasonal-wave model. It requires a column in decimal time format. The following code adds the column `Dectime` and executes `selBestWave`. The results from `selBestWave` are simply printed, but could be saved. Even though the timing of the peak is pretty clear from the graph, we'll take advantage of the exhaustive search to find the "best" peak too.

```
> # Add Dectime.
> app4.calib <- transform(app4.calib, Dectime=dectime(DATES))
> # Find the best model
> selBestWave(Buty ~ center(log(FLOW)) + dectime(DATES),
+             data = app4.calib,
+             "Dectime", app4.sp, exhaustive=TRUE, Regression=censReg,
+             dist="lognormal")

    Cmax Loading Hlife     Test
1 0.3932       1     1 87.09191
2 0.4032       1     1 88.46142
```

The "best" model has the timing of the peak at about 0.393, a pesticide loading period of 1 months and a decay rate indicated by a half-life of 1 month

(the fastest decay rate among the default choices). We are now ready to build and evaluate the seasonal-wave load model.

```
> # Create and print the seasonal-wave load model.
> # Note that we can use Dectime directly in this model
> app4.lrsw <- loadReg(Buty ~ center(log(FLOW)) + Dectime +
+                      seasonalWave(Dectime, 0.393, 1, 1),
+                      data = app4.calib, flow = "FLOW",
+                      dates = "DATES", conc.units="ug/L",
+                      station="White River at Hazleton, Ind.")
> app4.lrsw

*** Load Estimation ***

Station: White River at Hazleton, Ind.
Constituent: Buty

           Number of Observations: 45
Number of Uncensored Observations: 22
           Center of Decimal Time: 1993.8
                  Center of ln(Q): 9.4824
                Period of record: 1992-10-19 to 1994-09-12

Selected Load Model:
--------------------

Buty ~ center(log(FLOW)) + Dectime + seasonalWave(Dectime, 0.393,
    1, 1)

Model coefficients:
                                   Estimate Std. Error z-score
(Intercept)                        1579.573   712.8812   2.216
center(log(FLOW))                     1.632     0.2102   7.766
Dectime                              -0.793     0.3576  -2.218
seasonalWave(Dectime, 0.393, 1, 1)    3.464     0.5279   6.562
                                   p-value
(Intercept)                         0.0173
center(log(FLOW))                   0.0000
Dectime                             0.0172
seasonalWave(Dectime, 0.393, 1, 1)  0.0000

AMLE Regression Statistics
Residual variance: 0.7679
Generalized R-squared: 70.63 percent
G-squared: 55.13 on 3 degrees of freedom
P-value: <0.0001
```

```
Prob. Plot Corr. Coeff. (PPCC):
  r = 0.9643
  p-value = 0.1274
Serial Correlation of Residuals: 0.1422

Variance Inflation Factors:
                                      VIF
center(log(FLOW))                   1.084
Dectime                             1.072
seasonalWave(Dectime, 0.393, 1, 1) 1.014

Comparison of Observed and Estimated Loads
------------------------------------------
      Summary Stats: Loads in kg/d
------------------------------------------
    Min  25%  50%  75%  90%  95%  Max
Est   0 0.03 0.10 0.34 1.15 1.20 1.67
Obs   0 0.02 0.06 0.40 0.93 1.03 2.11

Bias Diagnostics
----------------
 Bp: 9.952 percent
PLR: 1.1
  E: 0.602
```

The Bias Diagnostics indicate a much better fit for this model than the seasonal model originally fit. The diagnostic plots confirm the improvement in the fit.

Figure 6 shows the response versus the fitted values, censored observations are shown by open circles. It is still a bit disconcerting that there is only one censored observation below the dashed regression line for fitted values less than about -3. That discrepancy in the location of censored values also drives the appearance of greater scatter in larger fitted values; plots 2 and 3 are not shown.

```
> # setSweave is required for the vignette.
> setSweave("app4_06", 5, 5)
> plot(app4.lrsw, which=1, set.up=FALSE)
> graphics.off()
```

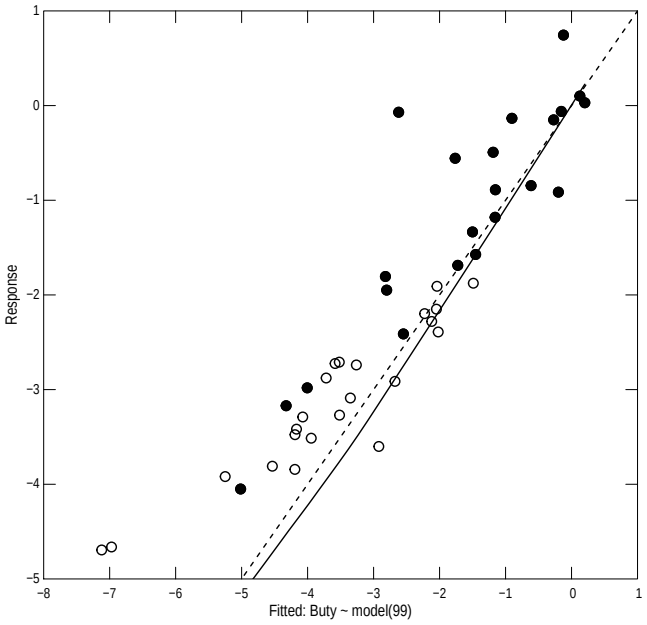

Figure 6. The rating-curve regression model using a seasonal wave.

Figure 7 is the correlogram. It does not suggest a seasonal lack of fit, but the smooth line is not as flat as one would like.

```
> # setSweave is required for the vignette.
> setSweave("app4_07", 5, 5)
> plot(app4.lrsw, which=4, set.up=FALSE)
> graphics.off()
```

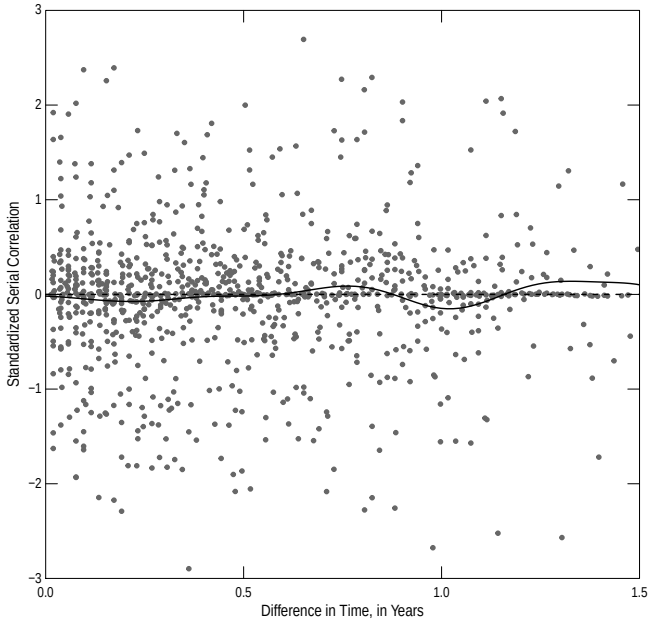

Figure 7. The correlogram for the model using a seasonal wave.

Figure 8 is q-normal plot and shows the standardized residuals. The visual appearance of figure 8 confirms the results of the PPCC test in the printed output—the residuals show little deviation from the normal distribution, but there is one outlying observation. The open circles are censored observations and the plotted value is the expected value returned by the `residuals` functions using the argument `type` set to "working."

```
> # setSweave is required for the vignette.
> setSweave("app4_08", 5, 5)
> plot(app4.lrsw, which=5, set.up=FALSE)
> graphics.off()
```

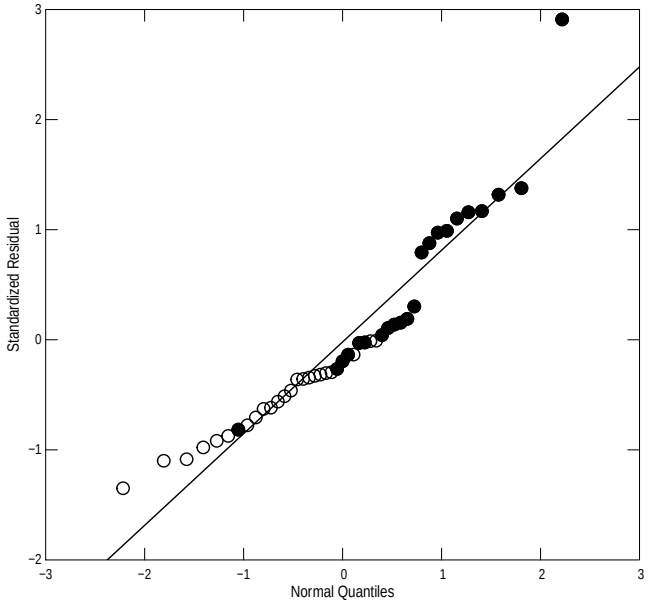

Figure 8. The Q-normal plot of the residuals for the model using a seasonal wave.

Figure 9 is an extended box plot—a truncated box plot, at the 5 and 95 percentiles that shows the individual values larger than the 95th percentile and smaller than the 5th percentile. The box plots in figure 6 show the distributions of the actual and estimated loads. The appearance of these box plots helps to understand the printed bias diagnostics, which show a small over estimate. The upper and lower ends of of the estimated values extend beyond the observed values and the box plot of the estimates shows a median value which is a bit larger than the median value for the observed values. Therefore, the middle and upper ranges are slightly over estimated. But the overall over estimate is less than 10 percent.

```
> # setSweave is required for the vignette.
> setSweave("app4_09", 5, 5)
> plot(app4.lrsw, which=6, set.up=FALSE)
> graphics.off()
```

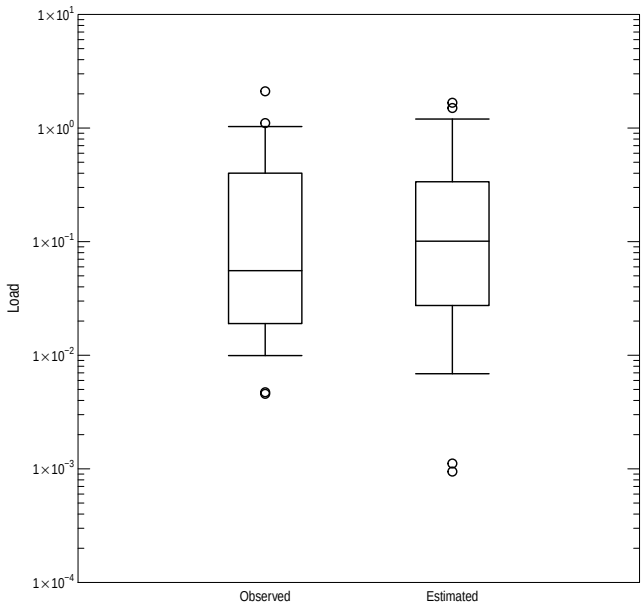

Figure 9. Box plot comparing estimated and observed values for the model using a seasonal wave.