Trend Analysis of Censored Metals

Dave Lorenz

October 30, 2013

This example illustrates the data manipulations for the Tobit analysis of uncensored data. Metals are often left-censored in natural waters and provide a useful example. This example also uses a common time frame for all of the trend tests. The common time frame facilitates comparing trends among the stations. Most often users will want to divide trend analyses into similar groups of analytes like metals, major ions, nutrients, and so forth because they will be analyzed in similar ways and will have common sampling time frames.

The data used in this application are a small subset of the data used by Schertz and others (1991). The data are samples taken from water year 1969 (October, 1968) through water year 1989 (September, 1989). Nineteen stations were selected and only copper and iron were selected for the metals.

```
> # Load the restrend package and the data
> library(restrend)
> data(EstrendSub)
> head(EstrendSub)
```

```
     STAID      DATES    QI QD RN.organic PN.organic RAmmonia PAmmonia
1 07227500 1968-10-01   7.6 NA                    NA                NA
2 07227500 1968-10-03   5.3 NA                    NA                NA
3 07227500 1968-10-16 532.0 NA                    NA                NA
4 07227500 1968-10-19  17.0 NA                    NA                NA
5 07227500 1968-11-01  17.0 NA                    NA                NA
6 07227500 1968-12-01   6.6 NA                    NA                NA
  RKjeldahl PKjeldahl RTotal.P PTotal.P RCopper PCopper RIron PIron Calcium
1                NA                NA              NA          NA      95
2                NA                NA              NA          NA      NA
3                NA                NA              NA          NA      42
4                NA                NA              NA          NA     121
5                NA                NA              NA          NA     150
6                NA                NA              NA          NA     138
  Chloride
1      280
2       NA
```

3	106
4	435
5	512
6	510

1 Summarize the Sample Data

In general, it is desirable, but not necessary, to subset the data before proceeding with the analysis of a subset of the constituents. Before these data are subsetted, the FLOW column must be created. The flow data are in two columns `QI`, the flow at the time of the sample; and `QD`, the mean flow on the day of the sample. The `coalesce` function in the `USGSwsBase` package can used to select the non-missing value for flow.

For censored data, which includes left- and multiply-cenosred data, the response variable must be converted to class "qw." The use of this class facilitates censored data analysis. The `convert2qw` function in the `USGSwsQW` package can be used to convert these data. The conversion requires at least 2 columns, one for the value and one for the remark code. For these data, columns beginning with "P" contain the value and columns beginning with "R" contain the remark code; the matching suffixes define the pair. This naming scheme is known as the Booker convention. Note that USGS data retrieved using the `importQW` function have much more meta information and do not need conversion.

```
> # Compute FLOW.
> EstrendSub <- transform(EstrendSub, FLOW=coalesce(QI, QD))
> # Convert, the default scheme is "booker"
> EstrendSub.qw <- convert2qw(EstrendSub)
> # Create the subset, the Pcolumn name is preserved
> Metals <- subset(EstrendSub.qw, select=c("STAID", "DATES", "FLOW",
+                                          "PIron", "PCopper"))
> # Rename metals to remove the leading P
> names(Metals)[4:5] <- c("Iron", "Copper")
> # Show the first few rows of the data
> head(Metals)

     STAID      DATES  FLOW Iron Copper
1 07227500 1968-10-01   7.6   NA     NA
2 07227500 1968-10-03   5.3   NA     NA
3 07227500 1968-10-16 532.0   NA     NA
4 07227500 1968-10-19  17.0   NA     NA
5 07227500 1968-11-01  17.0   NA     NA
6 07227500 1968-12-01   6.6   NA     NA
```

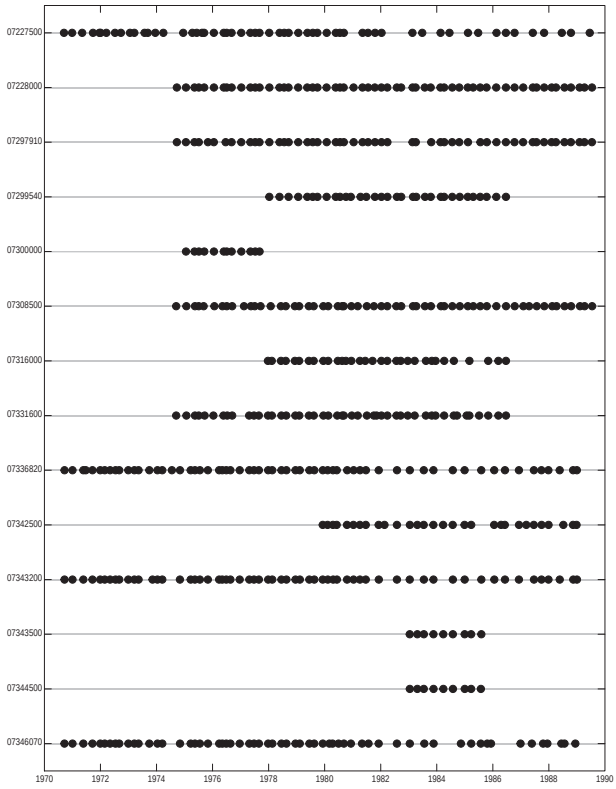
The `sampReport` function creates a simple PDF file that contains a report of the sample date ranges and graph of samples for each site. It can be used to help define the starting and ending date ranges for the trend tests as well as identifying sample gaps and other sampling issues. However, `sampReport` only reports the dates in the data set, it does not know about any missing samples. To get an accurate count of the samples, missing values

across all metal columns need to be removed. The `na.omit` function cannot be used becuase it would remove rows where there were any missing values.

```
> # Subset the data and show first few lines
> Metals <- subset(Metals, !(is.na(Iron) & is.na(Copper)))
> head(Metals)

       STAID      DATES  FLOW Iron Copper
116 07227500 1970-10-14   7.4   38      4
123 07227500 1971-01-28  38.0   34      6
137 07227500 1971-06-08  31.0   30      4
156 07227500 1971-10-27 104.0   40      5
167 07227500 1972-01-19  84.0   20      4
169 07227500 1972-02-03  30.0  720     25

> # Create the report
> sampReport(Metals, DATES="DATES", STAID="STAID", file="MetalsSampling")
```

The call to `sampReport` returns the file name invisibly (MetalsSampling.pdf). Because it is a full-size portrait PDF file, it is inserted here with compressed pages. The report gives the actual begin and end dates for sampling and the graph shows the sampling dates for each station. It is easy to see that only 14 stations were sampled for metals within the analysis period and the actual sampling at each station varied widely.

	STAID	FirstSamp	LastSamp	NumSamp
1	07227500	1970-10-14	1989-07-18	58
2	07228000	1974-10-24	1989-08-16	59
3	07297910	1974-10-24	1989-08-15	54
4	07299540	1978-02-10	1986-07-22	33
5	07300000	1975-02-20	1977-10-06	12
6	07308500	1974-10-15	1989-08-18	59
7	07316000	1978-01-23	1986-07-22	32
8	07331600	1974-10-15	1986-07-22	48
9	07336820	1970-10-21	1989-01-31	63
10	07342500	1980-01-07	1989-01-27	30
11	07343200	1970-10-21	1989-02-01	61
12	07343500	1983-02-14	1985-09-04	9
13	07344500	1983-02-15	1985-08-29	9
14	07346070	1970-10-20	1989-01-11	61


07227500
07228000
07297910
07299540
07300000
07308500
07316000
07331600
07336820
07342500
07343200
07343500
07344500
07346070
1970
1972
1974
1976
1978
1980
1982
1984
1986
1988
1990

2 Set up the Project

The user must balance the need to include as many stations as possible and the targeted time frame for the trend estimation. For these data, 4 stations have complete record beginning in October, 1970, but 3 additional stations have complete records beginning in October, 1974. This example will use the analysis period beginning in October, 1974 and ending in September, 1989.

The `setProj` function sets up the trend estimation project. There are many arguments to `setProj`, see the documentation for details. The constituent names or response variable names are referred to as `Snames` in keeping with the names used in the original ESTREND.

After projects have been set up, the user can get a list of the projects by using `lsProj` or can specify a project to use with `useProj`. The function `useProj` must be used to continue working on a project after the user quits from the R session.

```
> # Set up the project
> setProj("metals", Metals, STAID="STAID", DATES="DATES",
+         Snames=c("Iron", "Copper"), FLOW="FLOW",
+         type="tobit", Start="1974-10-01", End="1989-10-01")

[1] "metals"
```

The `setProj` function creates a folder in the users workspace with that name. That folder contains `R` data that are updated after each successful call to an analysis function in `restrend`. Table 1 describes the data created in this example's call to `setProj`. Any object of class "matrix" or "by" are indexed by station and sname.

Table 1. The data created by `setProj`.

Name	Class	Description
estrend.cl	list	A record of the calls to analysis functions.
estrend.cn	matrix	A description of the censoring. May be "none," "left," or "multiple."
estrend.cp	matrix	The percent of observations that are left-censored.
estrend.df	by	The dataset, contains STAID, DATES, FLOW, and the response variable.
estrend.in	list	Information about the project, such as the start and end dates and the names of columns in each dataset.
estrend.st	matrix	The status for each station and sname. Must be "OK" to continue with the trend analysis.

It is useful to verify which stations and snames will be analyzed. The user need only enter the name of the R data object in the console. The stations listed as "OK" matches what we expect from the sample report.

```
> # Which are OK?
> estrend.st
```

```
          snames
stations   Iron           Copper
  07227500 "OK"           "OK"
  07228000 "OK"           "OK"
  07297910 "OK"           "OK"
  07299540 "short record" "short record"
  07300000 "too few data" "too few data"
  07308500 "OK"           "OK"
  07316000 "short record" "short record"
  07331600 "short record" "short record"
  07336820 "OK"           "OK"
  07342500 "short record" "short record"
  07343200 "OK"           "OK"
  07343500 "too few data" "too few data"
  07344500 "too few data" "too few data"
  07346070 "OK"           "OK"
```

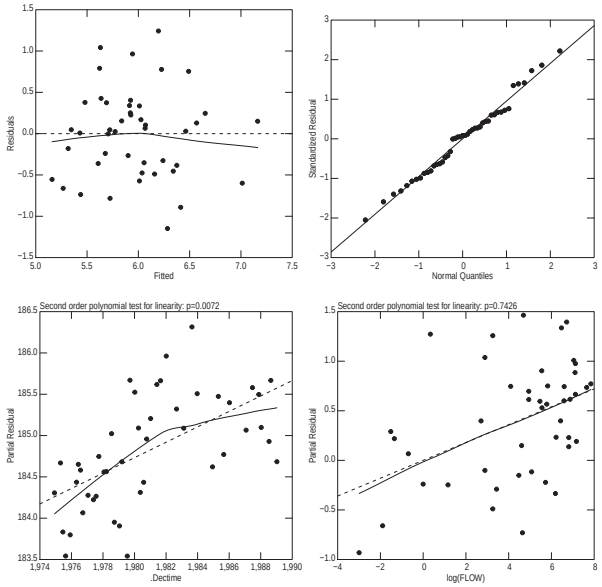
3 Tobit Trend Test

These data are now ready for the Tobit trend test. The function `tobitTrends` executes the trend test on all valid combinations of stations and snames. It can also execute the test on subsets if some changes need to be made. By default, the data are log-transformed and flow (also log-transformed) and first-order Fourier terms for seasonality are included in the regression analysis. The variable the describe the annual trend is called .Dectime and is always the last variable in the report.

The `tobitTrends` function also creates a PDF file that contains the result of the analysis and diagnostic graphs on each page. Most trends are very small for these data; only the report for Iron at 07346070 is shown. That is the only trend significant at the 0.05 level. The partial residual plot of trend shows some nonlinearity, but the reported slope is a good estimate of the average trned over the analysis period.

```
> # Trend tests, accepting default
> tobitTrends()

[1] "metals_tb.pdf"
```

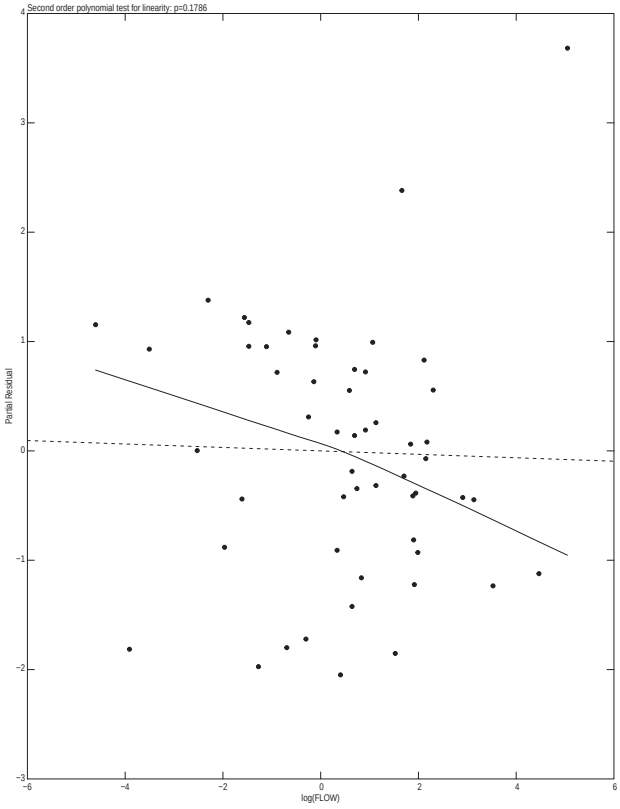
```
07346070  Iron
Call:
censReg(formula = log(Iron) ~ log(FLOW) + fourier(.Dectime, 1) +
    .Dectime)

Coefficients:
                              Estimate Std. Error z-score p-value
(Intercept)                 -179.27196   40.80412 -4.3935  0.0000
log(FLOW)                      0.08999    0.04539  1.9824  0.0400
fourier(.Dectime, 1)sin(k=1)  -0.12182    0.18443 -0.6606  0.4847
fourier(.Dectime, 1)cos(k=1)  -0.43195    0.12535 -3.4458  0.0006
.Dectime                       0.09330    0.02058  4.5328  0.0000

Estimated residual standard error (Unbiased) = 0.56

Distribution: normal
Number of observations = 45, number censored = 0 (0 percent)

Loglik(model) = -35.11 Loglik(intercept only) = -46.63
  Chi-square = 23.05, degrees of freedom = 4, p-value = 0.0001

Computation method: AMLE
```

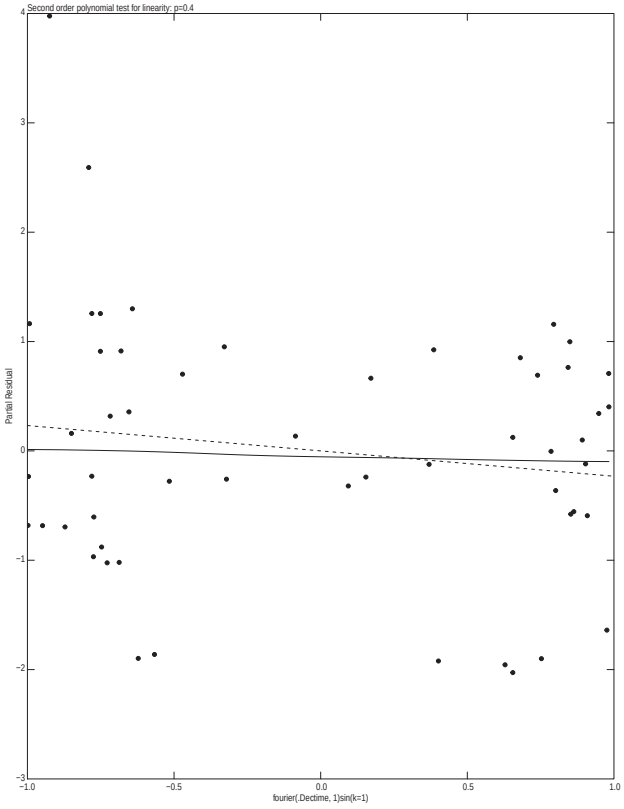


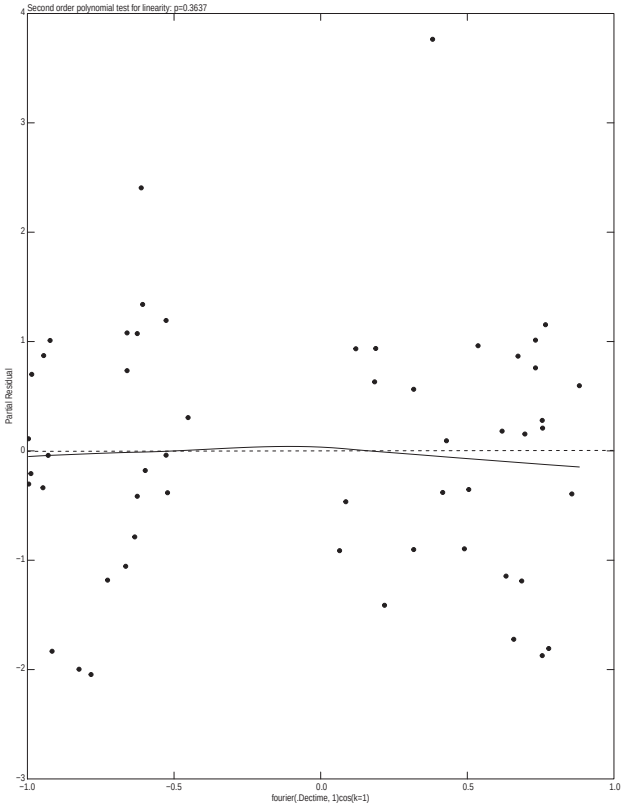
The diagnostic plots should be reviewed for verify the basic assumptions of linear regression–linearity of fit, uniformity of residuals, and normality of residuals. Note that the linearity and uniformity of residuals can be deceptive in the residuals vs. fit graph becuase of discrete values and censoring.

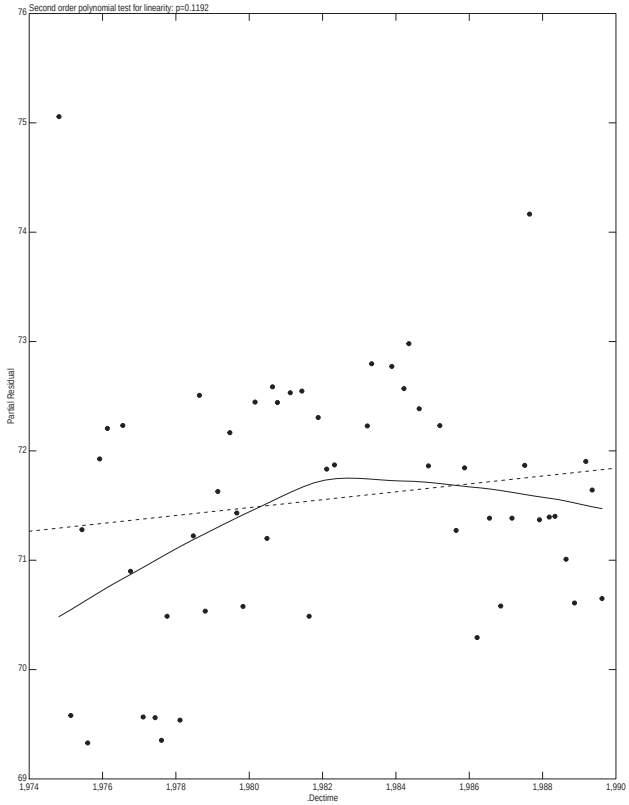
A more complete suite of diagnostic plots can be obtained using the `plotTT` function. For this example, there appears to be a highly influential observation in the analysis for Iron at station 07297910. The diagnostic plots reveal nothing unusual, except that it occurs very early in the record and at the largest flow.

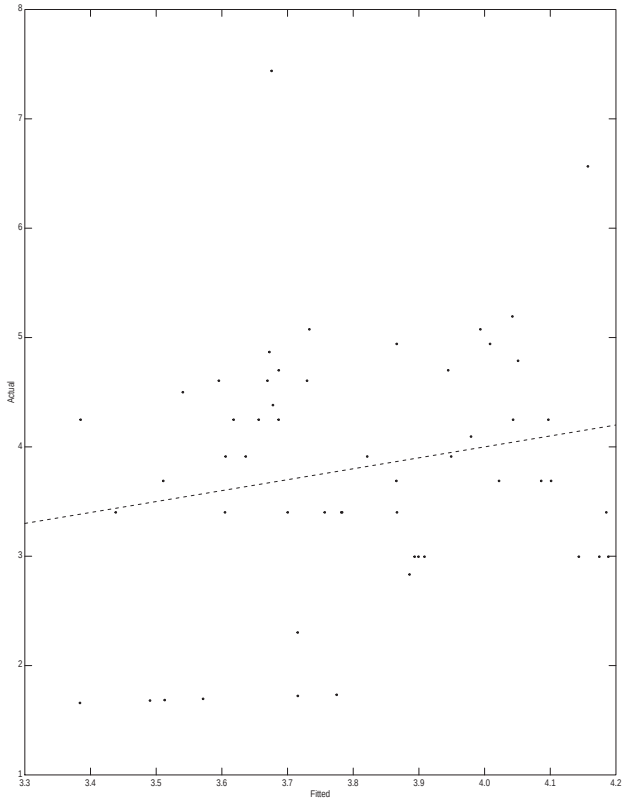
```
> # Trend tests, accepting default
> plotTT(Station="07297910", Sname="Iron", device="pdf")

[1] "X07297910_Iron.pdf"
```

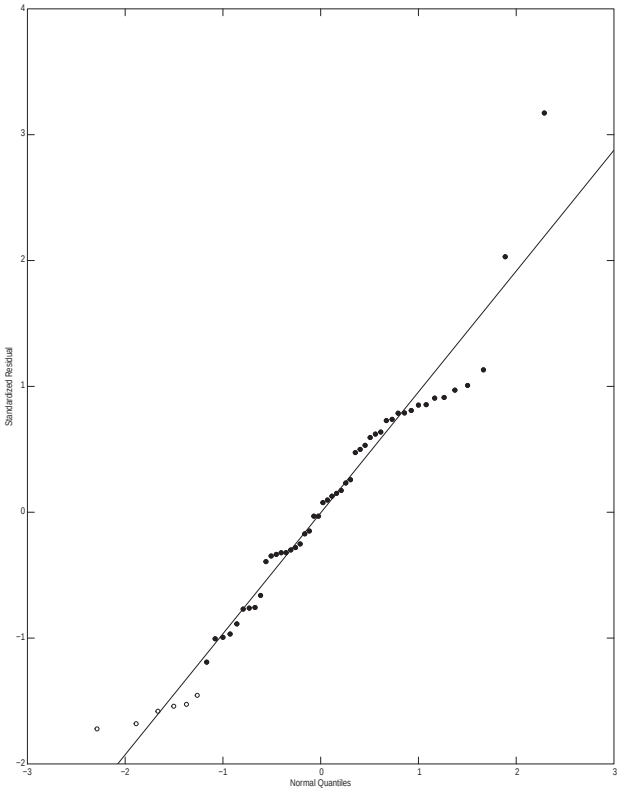

Second order polynomial test for linearity: p=0.1786
Partial Residual
log(FLOW)
4
3
2
1
0
−1
−2
−3
−6
−4
−2
0
2
4
6

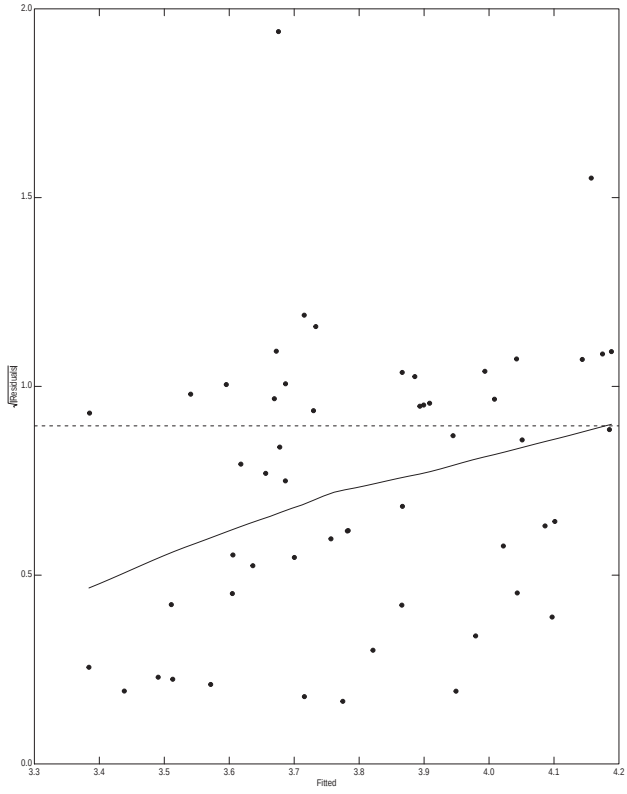

Second order polynomial test for linearity: p=0.4
Partial Residual
fourier(.Dectime, 1)sin(k=1)
4
3
2
1
0
−1
−2
−3
−1.0
−0.5
0.0
0.5
1.0

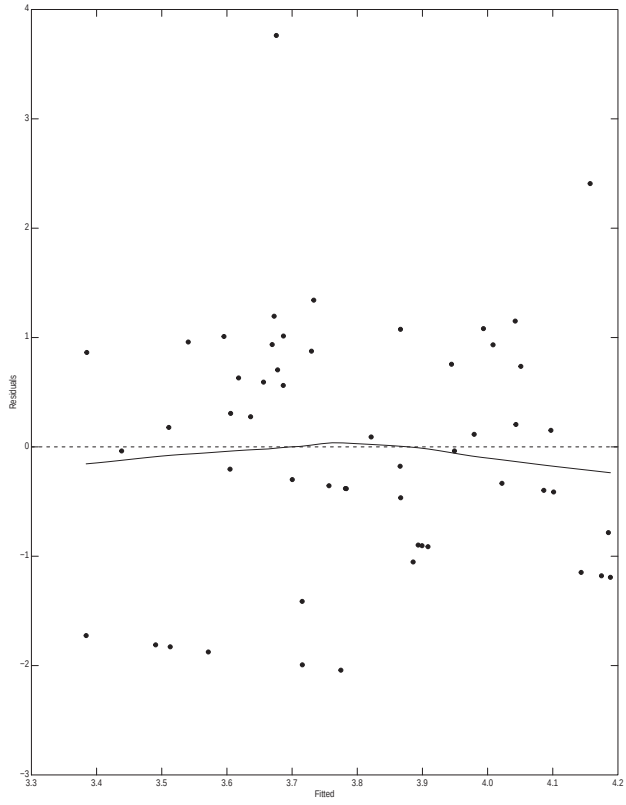

Second order polynomial test for linearity: p=0.3637
Partial Residual
fourier(.Dectime, 1)cos(k=1)
4
3
2
1
0
−1
−2
−3
−1.0
−0.5
0.0
0.5
1.0

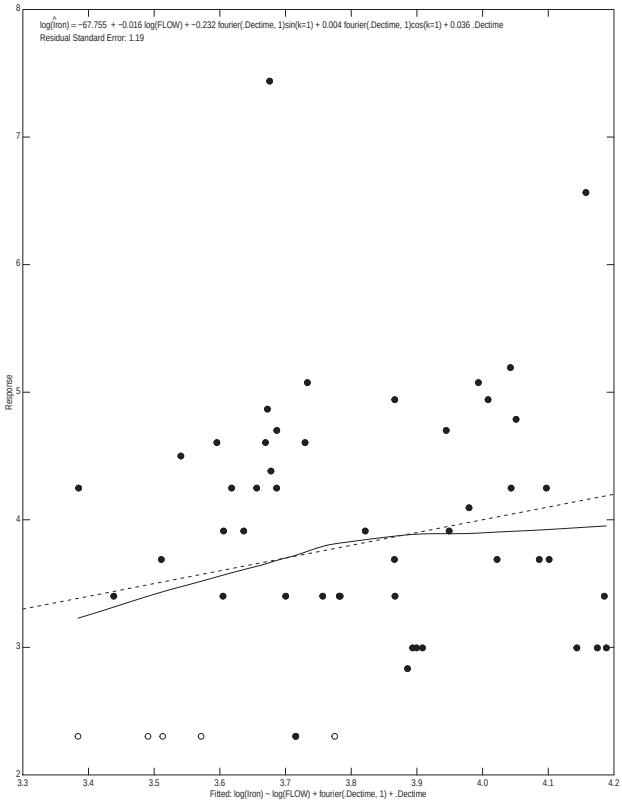

Second order polynomial test for linearity: p=0.1192
Partial Residual
.Dectime
76
75
74
73
72
71
70
69
1,974
1,976
1,978
1,980
1,982
1,984
1,986
1,988
1,990


Actual
Fitted
8
7
6
5
4
3
2
1
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4.0
4.1
4.2


Standardized Residual
Normal Quantiles
4
3
2
1
0
−1
−2
−3
−2
−1
0
1
2
3


2.0
1.5
1.0
0.5
0.0
√|Residuals|
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4.0
4.1
4.2
Fitted


Residuals
Fitted
4
3
2
1
0
−1
−2
−3
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4.0
4.1
4.2


log(Iron) = −67.755 + −0.016 log(FLOW) + −0.232 fourier(.Dectime, 1)sin(k=1) + 0.004 fourier(.Dectime, 1)cos(k=1) + 0.036 .Dectime
Residual Standard Error: 1.19
Response
8
7
6
5
4
3
2
3.3
3.4
3.5
3.6
3.7
3.8
3.9
4.0
4.1
4.2
Fitted: log(Iron) ~ log(FLOW) + fourier(.Dectime, 1) + .Dectime

4 Trend Results

When completed, or to check on intermediate results, the estimated trends can be extracted using the `getTrends` function. By default, all stations and snames are extracted. The output dataset is explained in the documentation for `getTrends`. The user has the option to set a significance level to determine whether there is a significant trend, the default level is 0.05.

```
> # get the trends
> metals.tnd <- getTrends()
> print(metals.tnd)
```

```
    Station Response  Type NumYears NumSeas Nobs RepValue        Trend
1  07227500     Iron Tobit 14.50513      NA   42       20 -0.540824909
2  07227500   Copper Tobit 14.50513      NA   42        2  0.001973088
3  07228000     Iron Tobit 14.81177      NA   58       20  0.563684368
4  07228000   Copper Tobit 14.81177      NA   58        1  0.034454725
5  07297910     Iron Tobit 14.80903      NA   54       40  1.470451671
6  07297910   Copper Tobit 14.80903      NA   54        1  0.011041926
7  07308500     Iron Tobit 14.84189      NA   55       30 -0.019554571
8  07308500   Copper Tobit 14.84189      NA   55        2  0.068046237
9  07336820     Iron Tobit 14.16016      NA   46       30 -0.518956445
10 07336820   Copper Tobit 14.16016      NA   46        1  0.044836056
11 07343200     Iron Tobit 14.16564      NA   46       30  0.936588462
12 07343200   Copper Tobit 14.16564      NA   46        2  0.045044124
13 07346070     Iron Tobit 14.10541      NA   45      400 39.116044095
14 07346070   Copper Tobit 14.10541      NA   46        2  0.114358859
     Trend.pct      P.value Trend.dir
1  -2.70412454 5.357736e-01      none
2   0.09865438 9.953258e-01      none
3   2.81842184 3.580891e-01      none
4   3.44547250 1.344894e-01      none
5   3.67612918 3.281969e-01      none
6   1.10419259 7.361143e-01      none
7  -0.06518190 9.846199e-01      none
8   3.40231183 2.899362e-01      none
9  -1.72985482 5.292452e-01      none
10  4.48360563 7.178072e-02      none
11  3.12196154 4.079293e-01      none
12  2.25220621 3.068992e-01      none
13  9.77901102 1.567344e-05        up
14  5.71794294 2.269600e-02        up
```

5 Further Remarks

Because trend analysis is not necessarily a straightforward process, but requires user assessments at several points in the process, it is not necessarily a good idea to simply create scripts and run them without any user interaction. To overcome recording the steps in a script, the functions in restrend record all changes to the projects data in a list called `estrend.cl`. It can be viewed at any time simply by entering estrend.cl in the console window. It can be saved with the data to ensure that the trend analysis is reproducible.

```
> # get the history
> estrend.cl

[[1]]
setProj(project = "metals", data = Metals, STAID = "STAID", DATES = "DATES",
    Snames = c("Iron", "Copper"), FLOW = "FLOW", type = "tobit",
    Start = "1974-10-01", End = "1989-10-01")

[[2]]
tobitTrends()
```

References

[1] Lorenz, D.L., in preparation, restrend: U.S. Geological Survey Open File Report, ? p.

[2] Schertz, T.L., Alexander, R.B., and Ohe, D.J., 1991, The computer program EStimate TREND (ESTREND), a system for the detection of trends in water-quality data: U.S. Geological Survey Water Resources Investigations Report 91-4040, 72 p.