Back transformation theory

Willem Vervoort, Michaela Dolk & Floris van Ogtrop

2017-09-22

```
# root dir
knitr::opts_knit$set(root.dir =
                        "C:/Users/rver4657/ownCloud/Virtual Experiments/VirtExp")
knitr::opts_chunk$set(echo = TRUE)
# LOAD REQUIRED PACKAGES # #####
library(pander)
library(tidyr)
library(zoo)
library(mgcv)
library(ggplot2)
```

This rmarkdown document and the resulting pdf are stored on github. All directories (apart from the root working directory) refer to the directories in this repository.

Introduction

This document is related to the manuscript "Disentangling climate change trends in Australian streamflow" (vervoort et al.), submitted to Journal of Hydrology.

Back transformation

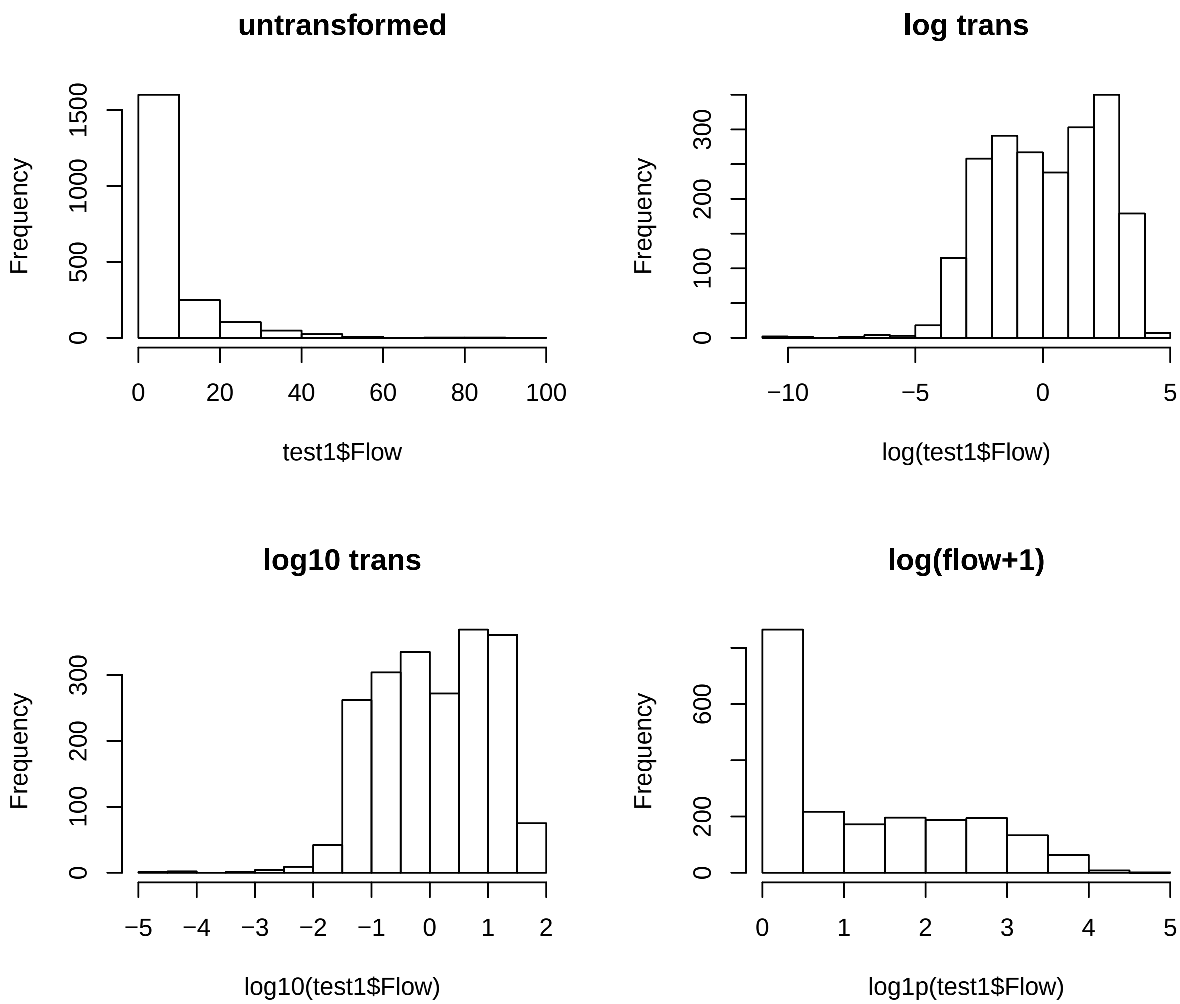
This document is related to the *3.GAMmodelTest.pdf* document and is to explain the back transformation used in the `gamm()` and `gls()` regressions, which are essentially the following type of regression equation
$log(y) = \beta_1 * trend$
How do we interpret β_1?
Assume a 1 unit change in trend in time, this results in:
$log(y_{t=1}) - log(y_{t=0}) = \beta_1$ or
$log(\frac{y_{t=1}}{y_{t=0}}) = \beta_1$
raising both sides to a power results in:
$\frac{y_{t=1}}{y_{t=0}} = exp(\beta_1)$
Or:
$y_{t=1} = exp(\beta_1) * y_{t=0}$
This means the actual slope of the trend is:
$exp(\beta_1) - 1$

```
load("data/ClimCh_project_MD.Rdata")

test <- flow_rain_maxT_weekly[flow_rain_maxT_weekly$Station=="DOMB",]
# add a vector of trend values
test$trend <- 1:nrow(test)

# remove 0 values
test1 <- test[test$Flow>0,]
```

```r
# plot distributions
par(mfrow=c(2,2))
hist(test1$Flow, main="untransformed")
hist(log(test1$Flow), main="log trans")
hist(log10(test1$Flow), main ="log10 trans")
hist(log1p(test1$Flow), main = "log(flow+1)")
```




```r
# log10 seems best
# modelling
plot(log10(Flow)~trend, data = test1)
mod1 <- lm(log10(Flow)~trend, data = test1)
summary(mod1)
```

```
##
## Call:
## lm(formula = log10(Flow) ~ trend, data = test1)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -4.6196 -0.7850  0.0180  0.8196  1.9875
```

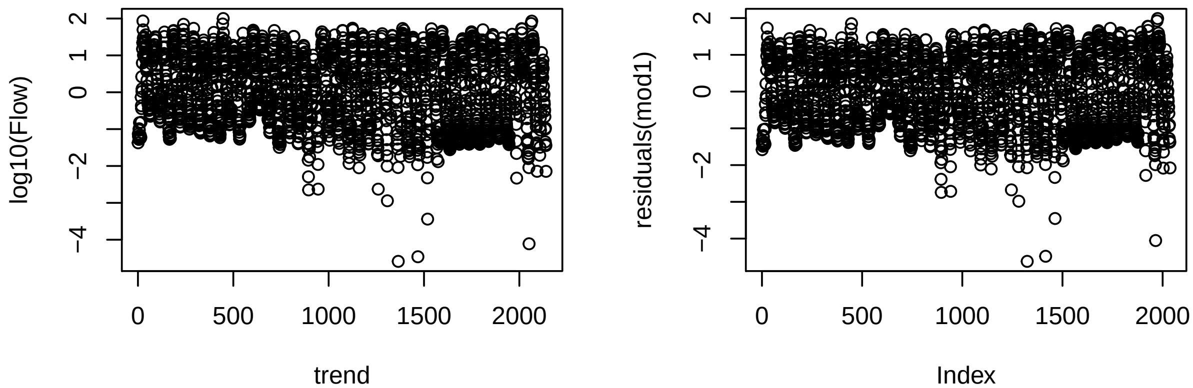
```
##
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  2.093e-01  4.146e-02   5.048 4.86e-07 ***
## trend       -1.296e-04  3.409e-05  -3.803 0.000147 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.951 on 2035 degrees of freedom
## Multiple R-squared:  0.007056,   Adjusted R-squared:  0.006568
## F-statistic: 14.46 on 1 and 2035 DF,  p-value: 0.0001473
```

```r
plot(residuals(mod1))
# interpret the slope
slope <- exp(coef(mod1)[2]) -1
slope
```

```
##         trend
## -0.0001296303
```

```r
# annual slope
slope*52
```

```
##        trend
## -0.006740773
```



comparison with non-transformed regression (for order of magnitude)

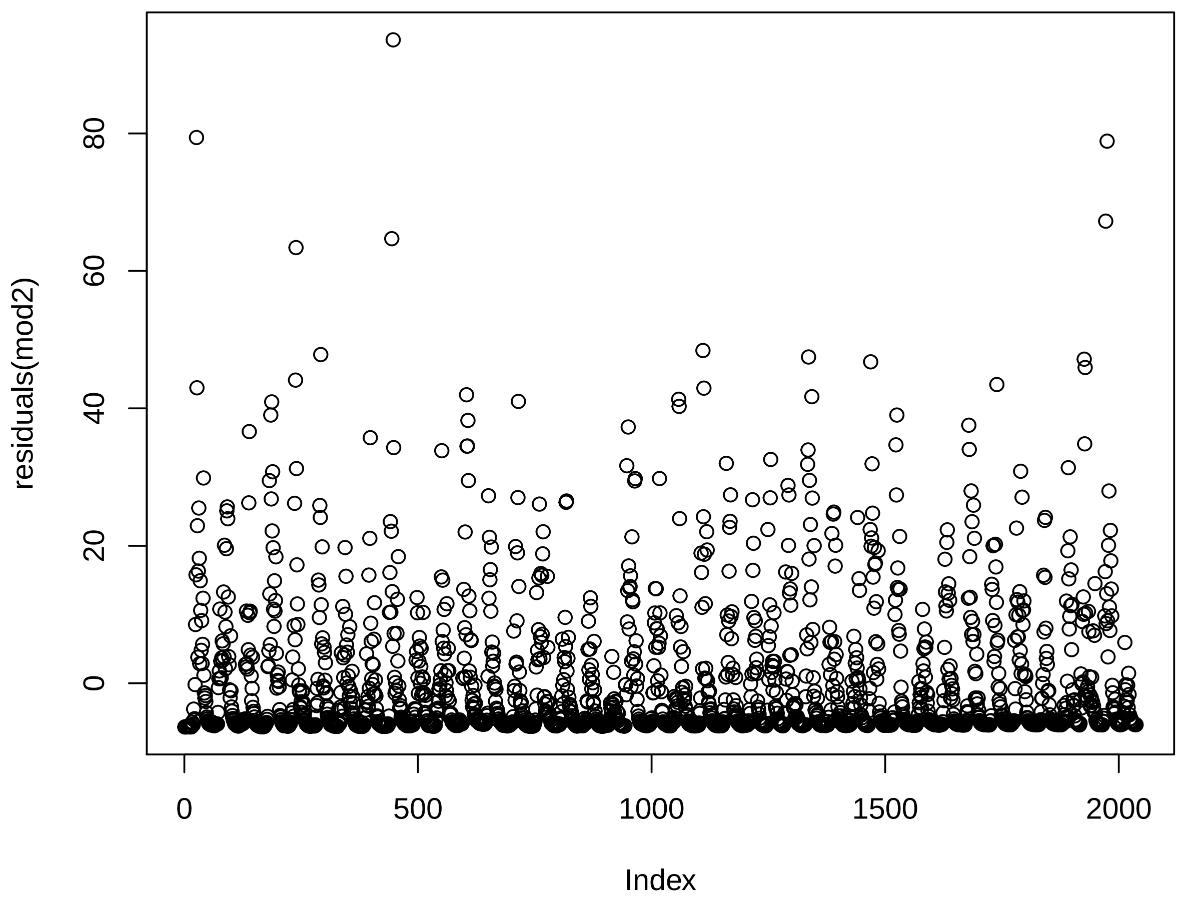
To compare the order of magnitude of the slope from the log transformed flow data, the simple regression also run. While this is statistically flawed it should at least indicate the order of magnitude of the actual slope in the data.

```
mod2 <- lm(Flow~trend, data = test1)
summary(mod2)
```

```
##
## Call:
## lm(formula = Flow ~ trend, data = test1)
##
## Residuals:
##    Min     1Q Median     3Q    Max
## -6.362 -6.009 -4.969  1.724 93.616
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)
## (Intercept)  6.4041901  0.4501729  14.226   <2e-16 ***
## trend       -0.0001718  0.0003702  -0.464    0.643
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.33 on 2035 degrees of freedom
## Multiple R-squared:  0.0001059,  Adjusted R-squared:  -0.0003855
## F-statistic: 0.2154 on 1 and 2035 DF,  p-value: 0.6426
```

```r
plot(residuals(mod2))
```




```r
# interpret the slope
(slope <- coef(mod2)[2])
```

```
##        trend
## -0.000171819
```

```r
# annual slope
slope*52
```

```
##        trend
## -0.008934589
```

This indicates the same order of magnitude as the log transformed analysis.