Statistical Modelling of Missing Discharge Data

PAZ

25 octobre 2016

Purpose

This document compares three methods to:

Input missing data in cleaned discharge data:

- Simple interpolation (inputting data)

Smooth-out the time series. - Exponential Weighted Moving Averages (EWMA) - i.e. parameter alpha is adjusted manually - Double Exponential Smoothing - i.e. R finds optimal parameters automatically

The input file is:

- **hydroAlteck2016_NAs_R.csv**

The file stems from *CleanDischargeDat_hydroAlteck2016_NAs.Rmd*, which removed aberrant values from the flow meter data.

The generated output file is:

- **hydroAlteck2016_smooth_R.csv**.

Required packages

```
# Plotting functions
library("ggplot2")
library("scales")
library("tidyr")

# Interpolation packages
library("zoo")
library("forecast")
```

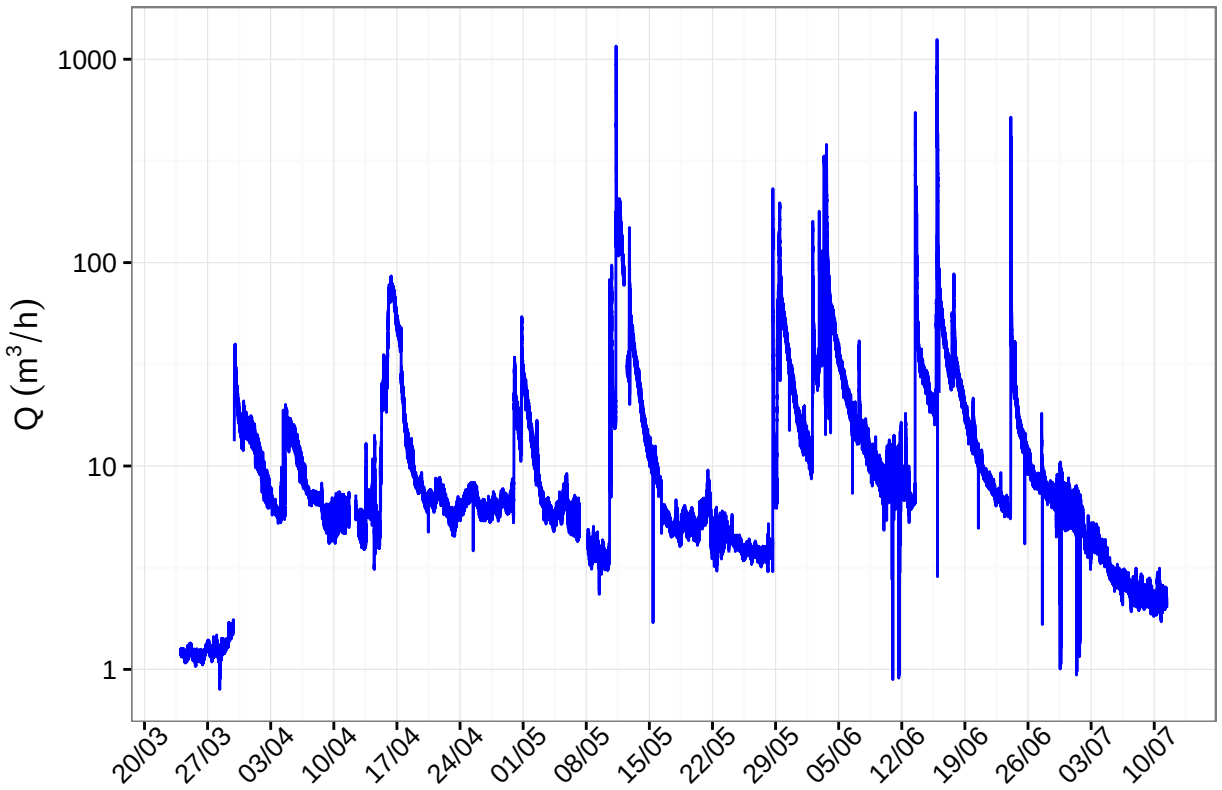
Import "clean" discharge data

```
hydroAlteck2016_na = read.csv2("Data/hydroAlteck2016_NAs_R.csv")
hydroAlteck2016_na$Date = as.POSIXct(strptime(hydroAlteck2016_na$Date, "%Y-%m-%d %H:%M", tz="CET"))

head(hydroAlteck2016_na)
```

```
##                  Date        DateCheck Q.m3Hrs   Qna
## 1 2016-03-25 00:00:00 25/03/2016 00:00   1.256 1.256
## 2 2016-03-25 00:02:00 25/03/2016 00:02   1.219 1.219
## 3 2016-03-25 00:04:00 25/03/2016 00:04   1.192 1.192
## 4 2016-03-25 00:06:00 25/03/2016 00:06   1.212 1.212
## 5 2016-03-25 00:08:00 25/03/2016 00:08   1.195 1.195
## 6 2016-03-25 00:10:00 25/03/2016 00:10   1.219 1.219
```

```
altp <- ggplot(hydroAlteck2016_na, aes(x=Date, y=Qna))
altp <- altp + geom_line(colour = "blue") +
  theme_bw() +
  scale_x_datetime(breaks = date_breaks("weeks"), labels = date_format("%d/%m")) +
  theme(axis.text.x=element_text(angle = 45, hjust = 0.75)) +
  xlab("") +
  ylab(expression(paste("Q ",({m}^"3"/h)))) +
  scale_y_continuous(trans=log_trans(), breaks=c(1,10,100,1000))

altp
```

```
## Warning: Removed 30 rows containing missing values (geom_path).
```




```
# + coord_cartesian(xlim = c(as.POSIXct("2016-05-08 23:00:00 CET"),
#                            as.POSIXct("2016-07-12 23:00:00 CET"))
#                   , ylim = c(0, 100)
```

```
#                   ) # no.1
  #scale_x_datetime(breaks = date_breaks("weeks"), labels = date_format("%d/%m"))
```

1st Discharge Set - Approximating Missing Data via the Zoo package

The **Zoo** package is one of the few packages (i.e. also **forecast**) where inputing data to univariate time series is possible [@Moritz2015]. Functions include:

- na.aggregate()
- na.StructTS()
- na.locf()
- na.approx()
- na.spline()

na.approx() function

Missing values (NAs) are replaced by linear interpolation using the na.approx function.

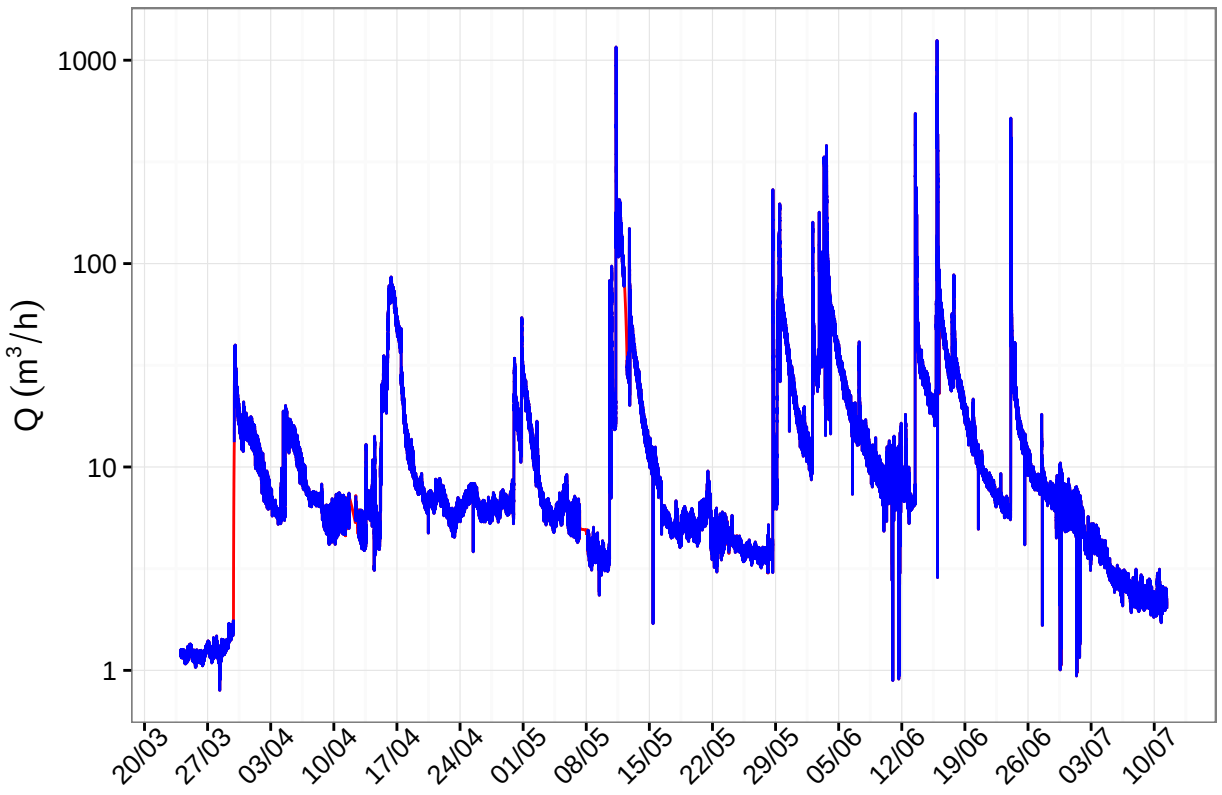
```
hydroAlteck2016_na$Qapprox = na.approx(hydroAlteck2016_na$Qna)
head(hydroAlteck2016_na)
```

```
##                   Date        DateCheck Q.m3Hrs   Qna Qapprox
## 1 2016-03-25 00:00:00 25/03/2016 00:00   1.256 1.256   1.256
## 2 2016-03-25 00:02:00 25/03/2016 00:02   1.219 1.219   1.219
## 3 2016-03-25 00:04:00 25/03/2016 00:04   1.192 1.192   1.192
## 4 2016-03-25 00:06:00 25/03/2016 00:06   1.212 1.212   1.212
## 5 2016-03-25 00:08:00 25/03/2016 00:08   1.195 1.195   1.195
## 6 2016-03-25 00:10:00 25/03/2016 00:10   1.219 1.219   1.219
```

```
interpol <- ggplot(hydroAlteck2016_na, aes(Date)) +
  theme_bw() +
  scale_x_datetime(breaks = date_breaks("weeks"), labels = date_format("%d/%m")) +
  theme(axis.text.x=element_text(angle = 45, hjust = 0.75)) +
  xlab("") +
  ylab(expression(paste("Q ",({m}^"3"/h)))) +
  scale_y_continuous(trans=log_trans(), breaks=c(1,10,100,1000)) +
  geom_line(aes(y = hydroAlteck2016_na$Qapprox), color="red") +
  geom_line(aes(y = hydroAlteck2016_na$Qna), color="blue") # +
  # coord_cartesian(xlim = c(as.POSIXct("2016-03-29 23:00:00 CET"), as.POSIXct("2016-04-05 00:00:00 CET
  #                  , ylim = c(0, 100)
  #                  )
interpol
```

```
## Warning: Removed 30 rows containing missing values (geom_path).
```

```
## Warning: Removed 30 rows containing missing values (geom_path).
```



na.StructTS() function (not working, can't convert to ts object with freq.)

```
# Code for na.StructTS
```

na.interp() function

This function shows no improvement over the na.approx() method.

```
hydroAlteck2016_na$Qinterp = na.interp(hydroAlteck2016_na$Qna)
head(hydroAlteck2016_na)
```

```
##                  Date        DateCheck Q.m3Hrs   Qna Qapprox Qinterp
## 1 2016-03-25 00:00:00 25/03/2016 00:00   1.256 1.256   1.256   1.256
## 2 2016-03-25 00:02:00 25/03/2016 00:02   1.219 1.219   1.219   1.219
## 3 2016-03-25 00:04:00 25/03/2016 00:04   1.192 1.192   1.192   1.192
## 4 2016-03-25 00:06:00 25/03/2016 00:06   1.212 1.212   1.212   1.212
## 5 2016-03-25 00:08:00 25/03/2016 00:08   1.195 1.195   1.195   1.195
## 6 2016-03-25 00:10:00 25/03/2016 00:10   1.219 1.219   1.219   1.219
```

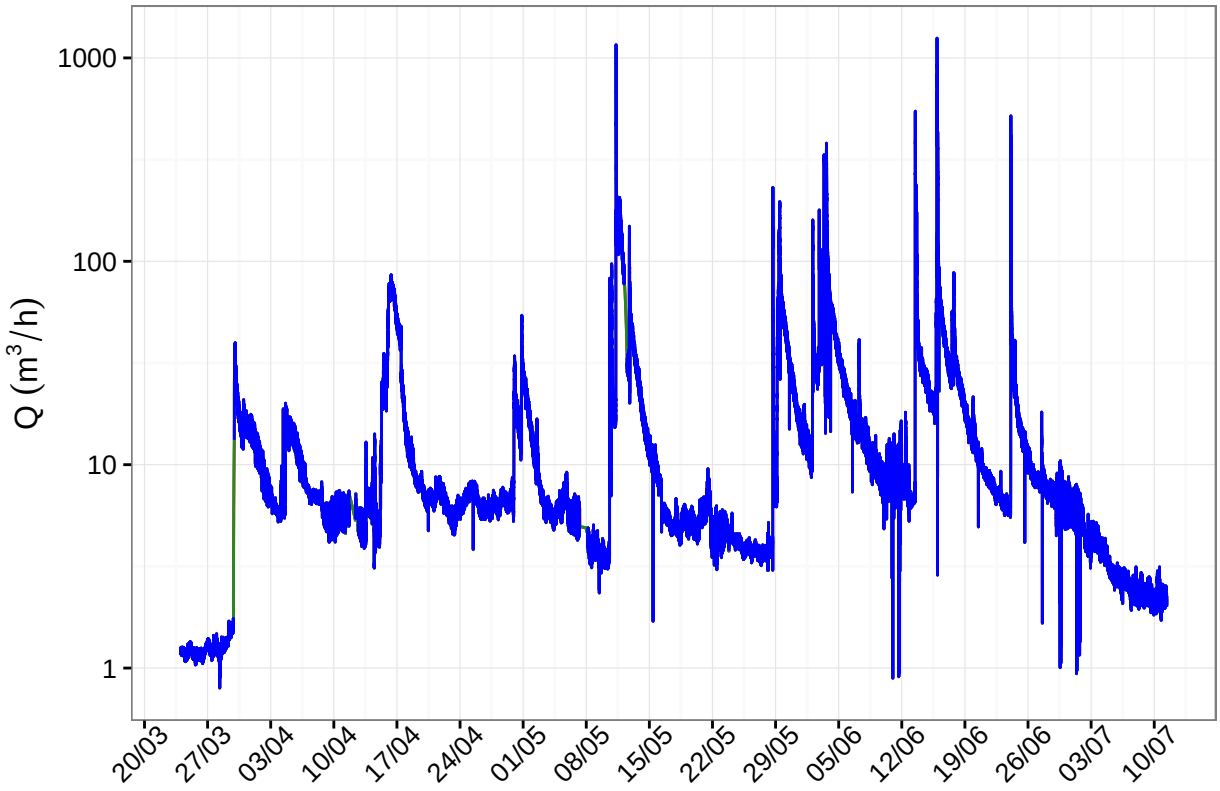
```
interpol <- ggplot(hydroAlteck2016_na, aes(Date)) +
  theme_bw() +
  scale_x_datetime(breaks = date_breaks("weeks"), labels = date_format("%d/%m")) +
```

```
  theme(axis.text.x=element_text(angle = 45, hjust = 0.75)) +
  xlab("") +
  ylab(expression(paste("Q ",({m}^"3"/h)))) +
  scale_y_continuous(trans=log_trans(), breaks=c(1,10,100,1000)) +
  geom_line(aes(y = hydroAlteck2016_na$Qapprox), color="red") +
  geom_line(aes(y = hydroAlteck2016_na$Qinterp), color="forestgreen") +
  geom_line(aes(y = hydroAlteck2016_na$Qna), color="blue") #  +
  # coord_cartesian(xlim = c(as.POSIXct("2016-03-29 23:00:00 CET"), as.POSIXct("2016-04-05 00:00:00 CET
  #                  , ylim = c(0, 100)
  #                  )
interpol
```

```
## Warning: Removed 30 rows containing missing values (geom_path).
```

```
## Warning: Removed 30 rows containing missing values (geom_path).
```

```
## Warning: Removed 30 rows containing missing values (geom_path).
```



Smoothing Data

Holt Winters 1 - Exponential Weighted Moving Averages (EWMA)

This approach manually adjusts the value of alpha.

```
# plot.ts(hydroAlteck2016_na$Qinter)
Q.HW1mean <- HoltWinters(hydroAlteck2016_na$Qinter,
                         alpha = 0.2, # If larger, less damping (i.e. more reactive).
                         beta = FALSE, # Controls how the trend adapts
                         gamma = FALSE # Controls adaptation of seasonal values
                         )

# Note:
# beta=False and gamma=FALSE gives Exponential Weighted Moving Averages (EWMA)

Q.HW1mean
```

```
## Holt-Winters exponential smoothing without trend and without seasonal component.
##
## Call:
## HoltWinters(x = hydroAlteck2016_na$Qinter, alpha = 0.2, beta = FALSE,     gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.2
##  beta : FALSE
##  gamma: FALSE
##
## Coefficients:
##       [,1]
## a 2.120646
```

```
# Removing the first entry of the original data to merge model
hydroAlteck2016 = hydroAlteck2016_na[2:nrow(hydroAlteck2016_na),]
hydroAlteck2016$Q.HW1 = Q.HW1mean$fitted[,1]

head(hydroAlteck2016)
```

```
##                   Date        DateCheck Q.m3Hrs   Qna Qapprox Qinterp
## 2 2016-03-25 00:02:00 25/03/2016 00:02   1.219 1.219   1.219   1.219
## 3 2016-03-25 00:04:00 25/03/2016 00:04   1.192 1.192   1.192   1.192
## 4 2016-03-25 00:06:00 25/03/2016 00:06   1.212 1.212   1.212   1.212
## 5 2016-03-25 00:08:00 25/03/2016 00:08   1.195 1.195   1.195   1.195
## 6 2016-03-25 00:10:00 25/03/2016 00:10   1.219 1.219   1.219   1.219
## 7 2016-03-25 00:12:00 25/03/2016 00:12   1.217 1.217   1.217   1.217
##      Q.HW1
## 2 1.256000
## 3 1.248600
## 4 1.237280
## 5 1.232224
## 6 1.224779
## 7 1.223623
```

Holt Winters 2 - Double Exponential Smoothing

This approach manually adjusts the value of alpha.

```
Q.HW2mean <- HoltWinters(hydroAlteck2016_na$Qinter,
                         gamma = FALSE)

Q.HW2mean
```

```
## Holt-Winters exponential smoothing with trend and without seasonal component.
##
## Call:
## HoltWinters(x = hydroAlteck2016_na$Qinter, gamma = FALSE)
##
## Smoothing parameters:
##  alpha: 0.9242677
##  beta : 0
##  gamma: FALSE
##
## Coefficients:
##        [,1]
## a  2.008201
## b -0.037000
```

```
# Note:
# gamma=FALSE gives Double Exponential Smoothing

# Shorten the data set by one more observation
hydroAlteck2016 = hydroAlteck2016[2:nrow(hydroAlteck2016),]
hydroAlteck2016$Q.HW2 = Q.HW2mean$fitted[,1]

head(hydroAlteck2016)
```

```
##                   Date        DateCheck Q.m3Hrs   Qna Qapprox Qinterp
## 3 2016-03-25 00:04:00 25/03/2016 00:04   1.192 1.192   1.192   1.192
## 4 2016-03-25 00:06:00 25/03/2016 00:06   1.212 1.212   1.212   1.212
## 5 2016-03-25 00:08:00 25/03/2016 00:08   1.195 1.195   1.195   1.195
## 6 2016-03-25 00:10:00 25/03/2016 00:10   1.219 1.219   1.219   1.219
## 7 2016-03-25 00:12:00 25/03/2016 00:12   1.217 1.217   1.217   1.217
## 8 2016-03-25 00:14:00 25/03/2016 00:14   1.230 1.230   1.230   1.230
##      Q.HW1    Q.HW2
## 3 1.248600 1.182000
## 4 1.237280 1.154243
## 5 1.232224 1.170626
## 6 1.224779 1.156154
## 7 1.223623 1.177241
## 8 1.222299 1.176989
```

Plotting the two smoothing methods

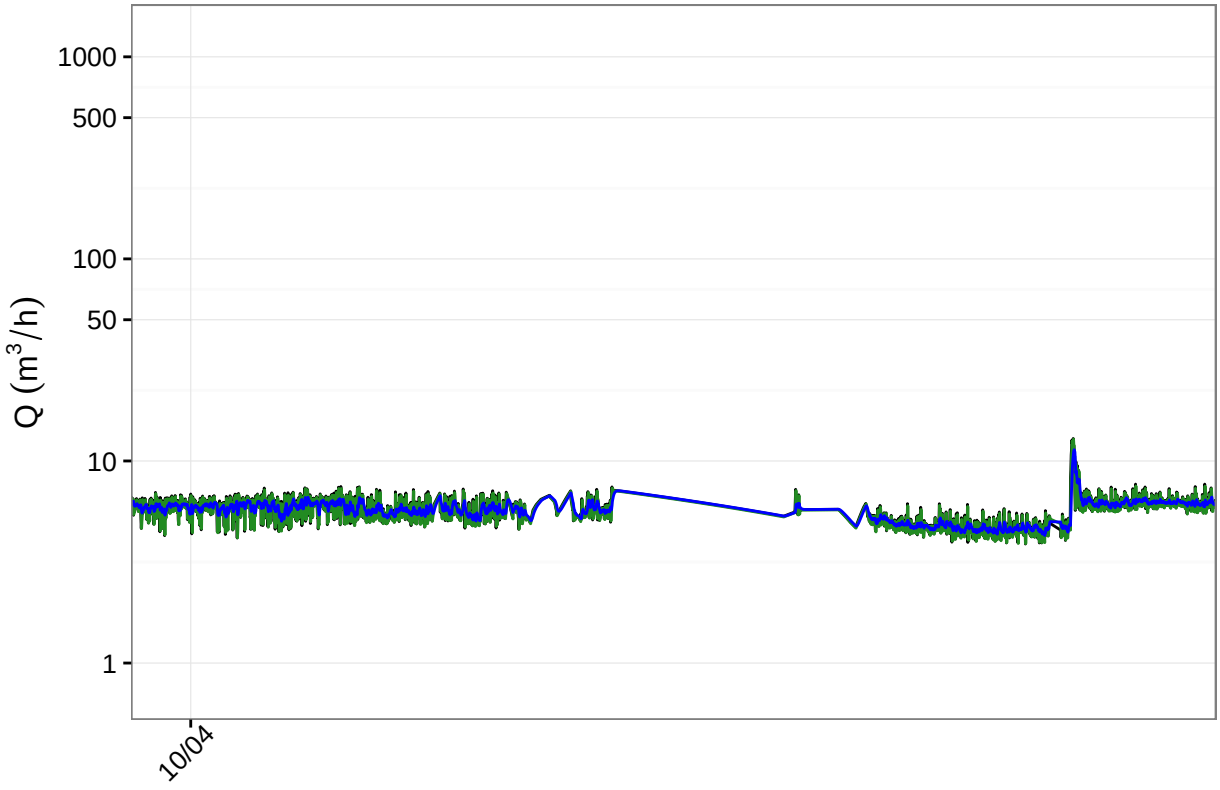
```
Qsmooth <- ggplot(hydroAlteck2016, aes(Date)) +
  theme_bw() +
  scale_x_datetime(breaks = date_breaks("weeks"), labels = date_format("%d/%m")) +
  theme(axis.text.x=element_text(angle = 45, hjust = 0.75)) +
  xlab("") +
```

```
  ylab(expression(paste("Q ",({m}^"3"/h)))) +
  scale_y_continuous(trans=log_trans(), breaks=c(1,10, 50, 100, 500,1000)) +
  geom_line(aes(y = hydroAlteck2016$Qinter), color="black") +
  geom_line(aes(y = hydroAlteck2016$Q.HW2), color="forestgreen") +
  geom_line(aes(y = hydroAlteck2016$Q.HW1), color="blue")  +
   coord_cartesian(xlim = c(as.POSIXct("2016-04-10 23:00:00 CET"), as.POSIXct("2016-04-15 00:00:00 CET"))
  #                  , ylim = c(0, 100)
                   )
Qsmooth
```

```
## Warning: Removed 30 rows containing missing values (geom_path).
```

```
## Warning: Removed 30 rows containing missing values (geom_path).
```

```
## Warning: Removed 30 rows containing missing values (geom_path).
```



Approximating missing values via subset prediction (trends)

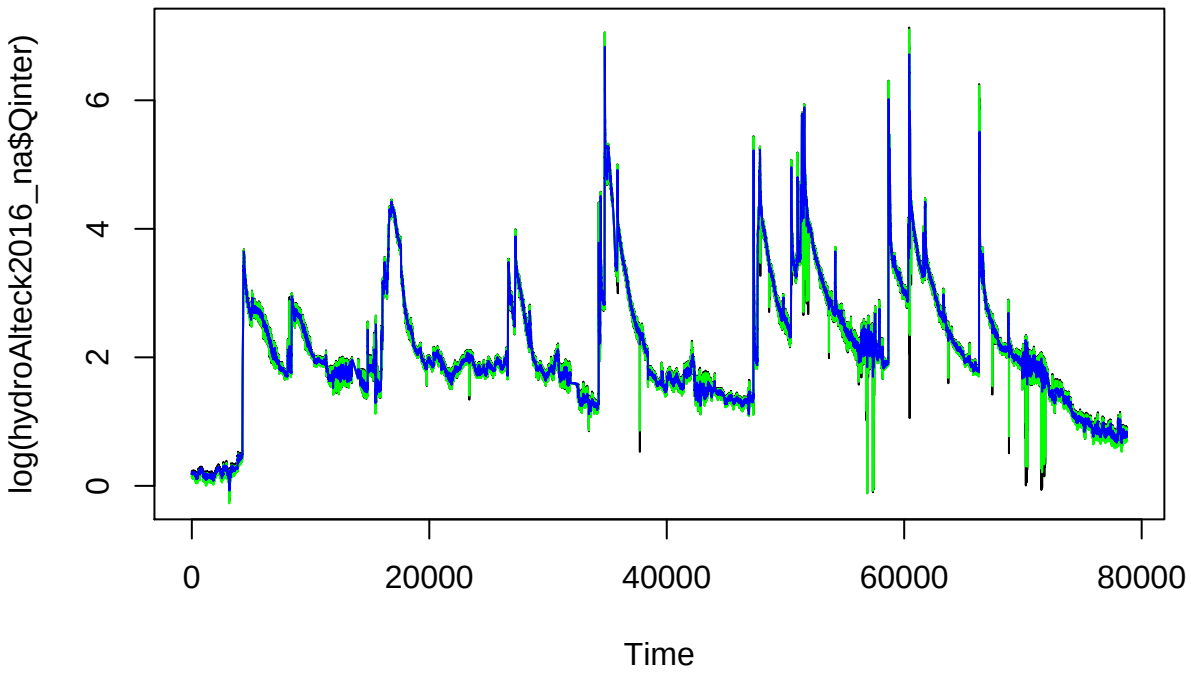
This section needs to subset the missing data and treated separetely.

```
Q1.predict <- predict(Q.HW1mean,
                      n.ahead = 10,
                      prediction.interval = TRUE)
```

```
Q1.predict
```

```
## Time Series:
## Start = 78762
## End = 78771
## Frequency = 1
##            fit      upr       lwr
## 78762 2.120646 20.76006 -16.51877
## 78763 2.120646 21.12919 -16.88790
## 78764 2.120646 21.49129 -17.25000
## 78765 2.120646 21.84675 -17.60546
## 78766 2.120646 22.19591 -17.95462
## 78767 2.120646 22.53910 -18.29781
## 78768 2.120646 22.87662 -18.63533
## 78769 2.120646 23.20874 -18.96744
## 78770 2.120646 23.53570 -19.29441
## 78771 2.120646 23.85775 -19.61646
```

```
# Q1.mean$fitted
plot.ts(log(hydroAlteck2016_na$Qinter))
lines(log(Q.HW2mean$fitted[,1]), col="green")
lines(log(Q.HW1mean$fitted[,1]), col="blue")
```



Approximating Missing Data - Local Level Model

The local level model assumes that we observe a time series, y_t, and that time series is the sum of another time series, μ_t, and random, corrupting noise, e_t. We would prefer to directly observe μ_t, a latent variable, but cannot due to the noise.

Establish the model

```
struct1 <- StructTS(hydroAlteck2016_na$Qinter, type="level")
if (struct1$code != 0) stop("optimizer did not converge")
print(struct1$coef)
```

```
##     level   epsilon
## 29.800690  2.643909
```

```
cat("Transitional variance:", struct1$coef["level"],
    "\n", "Observational variance:", struct1$coef["epsilon"],
    "\n", "Initial level:", struct1$model0$a, "\n")
```

```
## Transitional variance: 29.80069
##  Observational variance: 2.643909
##  Initial level: 1.256
```

Filter the with the StrucTS Model created

```
filt <- KalmanRun(hydroAlteck2016_na$Qinter, struct1$model)
#plot(unlist(filt))
```

Stuck trying to filter the data base donthe model...

Saving

```
write.csv2(hydroAlteck2016, "Data/hydroAlteck2016_smooth_R.csv", row.names = FALSE)
```