Joining data sets

Hydrologists often need to join time series which do not have values at the same time intervals. Fortunately, the dplyr package has functions which can help. The following is a simple example of how you can align time series.

We will be loading time series of annual peaks at two stations on the same river. We will align them in time, so that we can plot one against the other and do a regression.

We can load the data using the **tidyhydat** package function hy\_annual\_instant\_peaks.

library(tidyhydat)  
station1ID <- "05EF001" # North Saskatchewan at Deer Creek  
station2ID <- "05GG001" # North Saskatchewan at Prince Albert  
  
station1\_peaks <- hy\_annual\_instant\_peaks(station1ID)  
station2\_peaks <- hy\_annual\_instant\_peaks(station2ID)

We can now selecte only the annual maximum peak flows are selected, and the dates. The variable names are changed to the station IDs so that they can be identified when combining the time series.

station1\_max\_peak\_flows <-   
 station1\_peaks[(station1\_peaks$Parameter == "Flow") &   
 (station1\_peaks$PEAK\_CODE == "MAX") ,   
 c("Date", "Value")]  
  
names(station1\_max\_peak\_flows)[2] <- station1ID  
  
station2\_max\_peak\_flows <-   
 station2\_peaks[(station2\_peaks$Parameter == "Flow") &   
 (station2\_peaks$PEAK\_CODE == "MAX") ,   
 c("Date", "Value")]   
  
names(station2\_max\_peak\_flows)[2] <- station2ID

We also need to add the year of each peak, to the time series to be joined.

station1\_max\_peak\_flows$year <- as.numeric(format(station1\_max\_peak\_flows$Date, format = "%Y"))  
  
station2\_max\_peak\_flows$year <- as.numeric(format(station2\_max\_peak\_flows$Date, format = "%Y"))

Checking the data sets show that the time series span differing years.

summary(station1\_max\_peak\_flows)

## Date 05EF001 year   
## Min. :1944-06-17 Min. : 323.0 Min. :1944   
## 1st Qu.:1977-09-12 1st Qu.: 747.0 1st Qu.:1977   
## Median :1990-01-20 Median : 989.5 Median :1990   
## Mean :1988-06-21 Mean :1343.8 Mean :1988   
## 3rd Qu.:2002-11-21 3rd Qu.:1597.5 3rd Qu.:2002   
## Max. :2017-06-13 Max. :3960.0 Max. :2017

summary(station2\_max\_peak\_flows)

## Date 05GG001 year   
## Min. :1912-07-14 Min. : 477 Min. :1912   
## 1st Qu.:1963-11-24 1st Qu.: 813 1st Qu.:1964   
## Median :1981-08-02 Median :1140 Median :1981   
## Mean :1978-03-25 Mean :1409 Mean :1978   
## 3rd Qu.:1999-01-14 3rd Qu.:1670 3rd Qu.:1998   
## Max. :2016-08-31 Max. :5660 Max. :2016

We can now align the data sets, using the inner\_join function from the package **dplyr**. This function only selects peaks which have mathcing key values, in this case the years. Because the Date variable was in both data frames, both are in the joined data frame.

library(dplyr)  
common\_flows <- inner\_join(station1\_max\_peak\_flows, station2\_max\_peak\_flows, by = "year")  
  
summary(common\_flows$year)

## Min. 1st Qu. Median Mean 3rd Qu. Max.   
## 1952 1978 1992 1989 2002 2016

Although we could plot the flows against each other, we are interested in peaks occurring at boith gauges. So we can select only events where the downstream peak is a few days after the upstream peak.

common\_flows$delta\_t <- common\_flows$Date.y - common\_flows$Date.x  
  
common\_events <- common\_flows[(common\_flows$delta\_t >= 3) & (common\_flows$delta\_t <= 12),]

Having peaks which are at both gauges, we can plot the relationship between the peaks using **ggplot2**.

library(ggplot2)  
p <- ggplot(common\_events, aes(`05EF001`, `05GG001`)) +  
 geom\_point() +  
 xlab("Upstream (05EF001) peak discharge (m³/s)") +  
 ylab("Downstream (05GG001) peak discharge (m³/s)") +  
 geom\_smooth(method = lm)  
p

