Nutrient loads: plots with secondary axis

Introduction

Plots with secondary axis can be useful when illustrating the covariation of two variables that are measured in different units. Generally, it is advisable to be careful when using secondary axis as they can be misleading as to the magnitude of the two variables. Here we show how to construct plots with two y-axes using nutrient loads data (in tons) that are often closely related to the amount of total or average discharge.

Before you start, make sure you have installed and loaded the `tidyverse` and `readxl` packages.

```
library(tidyverse)
library(readxl)
```

Read the nutrient loads data set

This dataset contains nitrogen, phosphorus and TOC loads in river outlets in Sweden[1], as well a average discharge and some other catchment area characteristics.

```
Loads <- read_excel("Nutrient loads.xlsx")
```

Select data from a single river

First, recall that we can use `filter` to filter the data so that only observations from the Dalälven river are kept:

```
Loads %>%
 filter(River=="Dalälven")
```

Make a plot with secondary axis using ggplot2

Usually, for concentration series a line plots is used. Discharge could be plotted as line or as bars - we use bars in this example:

```
Loads %>%
 filter(River=="Dalälven") %>%
  ggplot(aes(x = Year)) +
  geom_line(aes(y = `Tot_N (tons)`)) +
  geom_col(aes(y=`Average discharge`), fill="lightblue")
```

Note that we do not get an axis for discharge - only the first variable used (total nitrogen) is shown on the y-axis. Also any co-variation is difficult to see as the blue bars are so small, because they are shown on the same scale as the total nitrogen measurements.

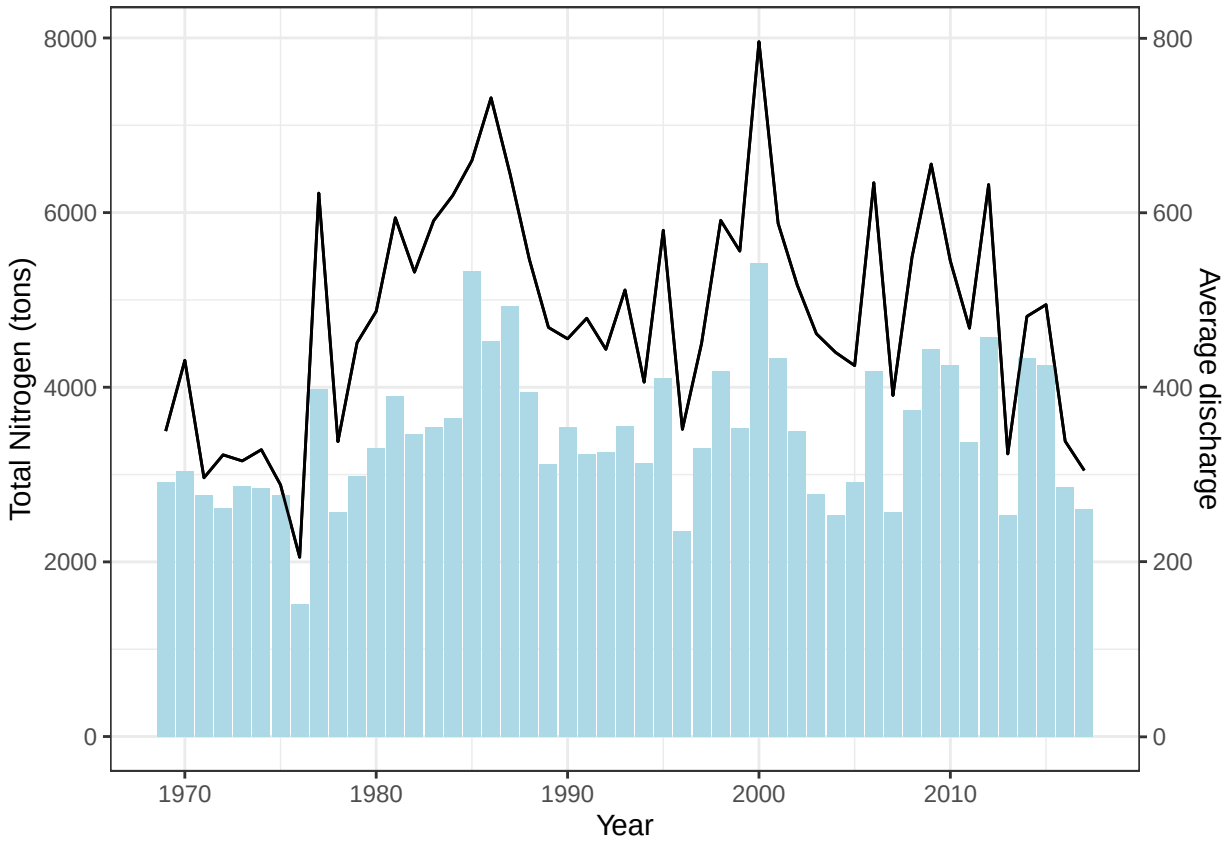
GGplot does not allow too flexible use of secondary plots due to the risk of misuse. What can be done is to adjust the secondary series with a multiplier (10 in this case) to get the series on a similar level in the plot. At the same time the secondary axis values must be adjusted to show a 10^{th} of the first axis to be correct.

[1] This data is part of the Swedish environmental monitoring for river outlets accessible for everyone through SLU's open data, https://miljodata.slu.se/mvm/

```
Loads %>%
 filter(River=="Dalälven") %>%
  ggplot(aes(x = Year)) +
  geom_line(aes(y = `Tot_N (tons)`)) +
  geom_col(aes(y=`Average discharge`*10),fill="lightblue") +
  scale_y_continuous(sec.axis = sec_axis(~./10, name = "Average discharge") )
```

To improve the plot further we might want to plot the black line in front of the blue bars. This can be done by simply redrawing it after the bars are added. Also the axis label for the first axis needs to be adjusted.

```
Loads %>%
 filter(River=="Dalälven") %>%
  ggplot(aes(x = Year)) +
  geom_line(aes(y = `Tot_N (tons)`)) +
  ylab("Total Nitrogen (tons)")+
  geom_col(aes(y=`Average discharge`*10),fill="lightblue") +
  scale_y_continuous(sec.axis = sec_axis(~./10, name = "Average discharge") )+
  geom_line(aes(y = `Tot_N (tons)`)) +
  theme_bw()
```



Exercise

1. Choose one of the other variables in the data set (TOC or total phosphorus) and make a similar plot for this variable together with average discharge. Note that the multiplication factor probably needs to be adjusted.

Make a plot with secondary axis using the general plot function

Using the general `plot` function from base R (i.e. not ggplot2) we can create plots with a secondary axis more easily. For instance, the axis scale is adjusted automatically. Instead we need to fix with the axis details to get everything right. Also, we need to create a data frame containing only Dalälven data first.

```
Loads_Dal <- Loads %>%
 filter(River=="Dalälven")
```

We start by creating a plot of the loads series without any axes (`axes=F`) or axis text (`xlab="", ylab=""`).. The range of the x-axis is given in case the two series to be plotted are unequally long (`xlim=c(1986, 2017)`). The plot should be a thin line (`type="l", lwd=1`).

```
plot(Loads_Dal$Year, Loads_Dal$`Tot_N (tons)`, axes=F, xlab="", ylab="",
     xlim=c(1968, 2017), type="l", lwd=1)
```

We can also create the plot containing the average discharge, also without axes. We use some kind of bars by specifying `type="h"` and adjust the width of the bars by `lwd=7`. The color of the bars can be adjusted by the statement `col=`.

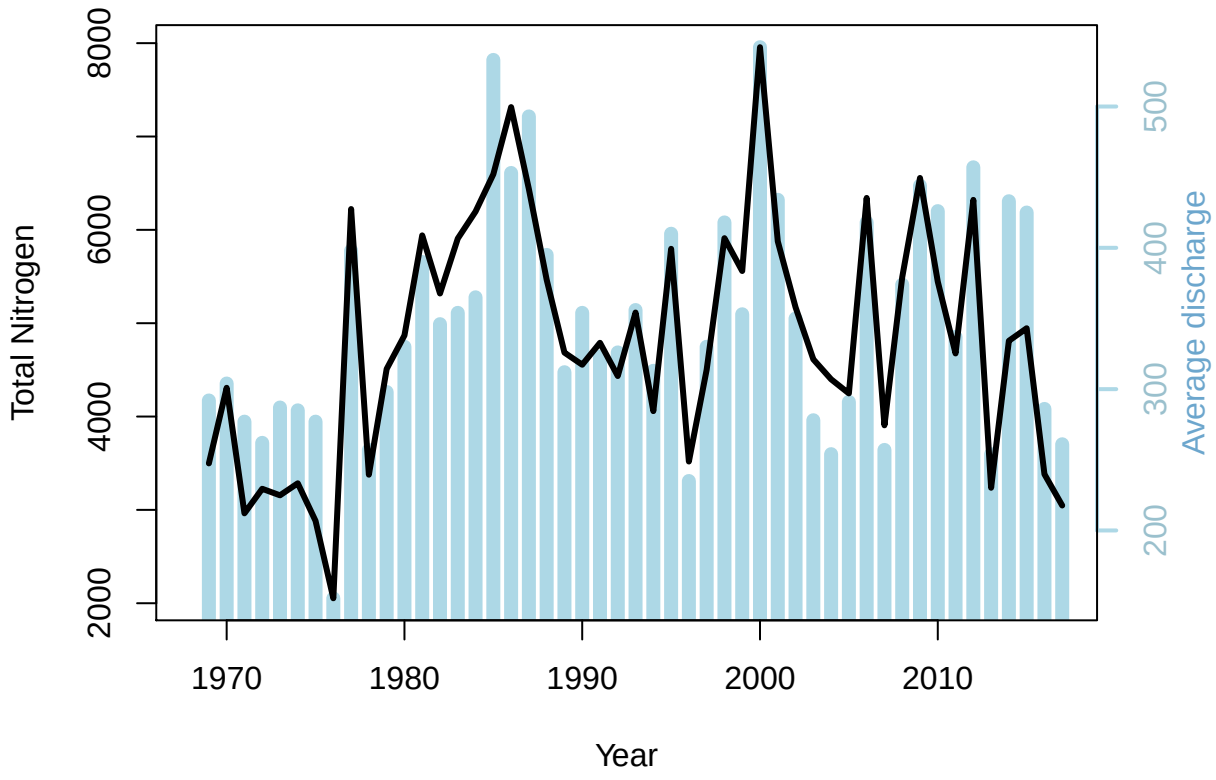
```
 plot(Loads_Dal$Year, Loads_Dal$`Average discharge`, axes=F, xlab="", ylab="",
      type="h", lwd=7, xlim=c(1968,2017), col="lightblue")
```

To combine these two plots in an R Markdown code chunk we need to add brackets {} around the code and put the cursor in the first line to make everything run at the same time. You can also use the green arrow in the upper right corner of the code chunk. If you use R SCript and not R Markdown there is no need to do this.

```
{par(mar=c(5,4,2,4))
plot(Loads_Dal$Year, Loads_Dal$`Tot_N (tons)`, axes=F, xlab="", ylab="",
     xlim=c(1968, 2017), type="l", lwd=1)
par(new=T)
plot(Loads_Dal$Year, Loads_Dal$`Average discharge`, axes=F, xlab="",ylab="",
     type="h", lwd=7, xlim=c(1968,2017), col="lightblue")}
```

Now we need to add the axes. For the secondary axis we use the `axis` and `mtext` statements. For the first axis we set the axis titles when creating the nitrogen loads series again (so that it is on top and can be seen better). The line starting with `par` needed to make enough space for margins for axis texts.

```
{par(mar=c(5,4,2,4))
plot(Loads_Dal$Year, Loads_Dal$`Tot_N (tons)`, axes=F, xlab="", ylab="",
     xlim=c(1968, 2017), type="l", lwd=1)
par(new=T)
 plot(Loads_Dal$Year, Loads_Dal$`Average discharge`, axes=F, xlab="", ylab="",
      type="h", lwd=7, xlim=c(1968,2017), col="lightblue")
axis(side = 4, line=0, col="lightblue", lwd=2, col.axis="lightblue3")
mtext(side = 4, line=2, "Average discharge", col="skyblue3" )
par(new=T)
plot(Loads_Dal$Year, Loads_Dal$`Tot_N (tons)`, xlab="Year", ylab="Total Nitrogen",
     xlim=c(1968, 2017), type="l", lwd=3)}
```



Solutions to selected exercises

1.

```
Loads %>%
 filter(River=="Dalälven") %>%
  ggplot(aes(x = Year)) +
  geom_line(aes(y = `TOC (tons)`)) +
  ylab("TOC (tons)")+
  geom_col(aes(y=`Average discharge`*200),fill="lightblue") +
  scale_y_continuous(sec.axis = sec_axis(~./200, name = "Average discharge") )+
  geom_line(aes(y = `TOC (tons)`))
```