Translation-pdf

SK

11/16/2020

First you need to download the whole folder containing the code (.R), the project (.Rproj) and the adiitonal information sheets. Once you have your local repository, you can first ofen the .Rproj file and then slect the Translation.R script from the *Files* tab.

Read input flow record

The data is saved in q__tt and number of days the discharge is recorded is 731

```
q_tt <- read_delim("Vermigliana_1996_1997.csv", delim=";",
                   col_names = c("Date", "Monthly_flow"), col_types = "cd")
nq <- length(q_tt[["Monthly_flow"]]) #length of the flow record
head(nq)
```

```
## [1] 731
```

Chose the eflow method that you want to apply by setting i__efty to;

- 1: constant for the whole year
- 2: for variable on monthly basis
- 3: for proportional input

```
 i_efty <- 2
```

For the scenario to be applied, choose between 1-5;

- 1: VMF, 2014
- 2: Tessman, 1980
- 3: Smakhtin et al., 2004
- 4: Tennant, 1976
- 5: Q90-Q50, 2014

```
i_mvef <- 1
```

For the water consumption properties, chose 1 for consumptive, 0 for non-consumptive use

```
icons <- 1
```

Other parameters;

- vrmax: max storage volume
- vrmin: minimum storage volume
- damhei: dam height (max available head)
- power: power generated by the power plant

```
rvmax   <- 20 * 10^6 #(10^6m3) #to keep it line with Guido's notations
rvmin <- 1.5* 10^6 #(10^6m3)
damhei <- 100 #(m)
power <- 0.1 * 10^6 #(MW)
```

Pre-processing of inflows and downstream water uses

Here we organize the variables so that it's easier to use in the future. Now we have the day month and year as separate variables

```
q_tt[["Date"]]<- mdy(q_tt[["Date"]])
q_tt <- q_tt%>%
  mutate(year= year(Date),
         month=month(Date),
         day=day(Date))
```

Then we calculate the monthly and yearly mean flow by grouping the observations by month or year. MMF_tt and MAF_tt has the mean flow of each month and year respectively.

```
MMF_tt <- q_tt %>% group_by(year, month) %>%
  summarise(mean_monthly_flow=mean(Monthly_flow))

MAF_tt <- q_tt %>% group_by(year) %>%
  summarise(mean_yearly_flow=mean(Monthly_flow))

ny <- nrow(MAF_tt) # number of years in the dataset

MMF <- MMF_tt %>% group_by(month) %>%           # Mean monthly flow vector for the whole record
  summarise(month_average=mean(mean_monthly_flow))

MAF <- mean(MMF[["month_average"]])             # Mean annual flow

head(MMF_tt)
```

```
## # A tibble: 6 x 3
## # Groups:   year [1]
##    year month mean_monthly_flow
##   <dbl> <dbl>             <dbl>
## 1  1996     1             1.08
## 2  1996     2             1.76
## 3  1996     3             0.929
## 4  1996     4             1.37
## 5  1996     5             2.94
## 6  1996     6             3.64
```

```r
head(MMF)
```

```
## # A tibble: 6 x 2
##   month month_average
##   <dbl>         <dbl>
## 1     1          1.10
## 2     2          1.39
## 3     3          1.14
## 4     4          1.41
## 5     5          3.05
## 6     6          4.75
```

50%, 90% and 97 % quantile calculation (there might be an error here but we kept it as Guido's)

```r
q50 <- q_tt %>% summarise(fifty_percentile = quantile(Monthly_flow, 0.50))
q90 <- q_tt %>% summarise(ninetth_percentile = quantile(Monthly_flow, 0.10))
q97 <- q_tt %>% summarise(ninetyseventh_percentile = quantile(Monthly_flow, 0.03))
```

Eflow scenarios

```r
qefm <-vector("numeric", 1)
LIHflo <- vector("numeric", 1)

if (i_efty==1){ #case 1:constant eflow throughout the whole year

  qefm <- rep(q90, 12)

   } else if (i_efty==2){ #case 2: variable eflow on a monthly basis

      if (i_mvef==1){                         # ----------------- VMF (2014)
        for (i in 1:nrow(MMF)) {
          if (MMF[i, 2] <= 0.4*MAF) {
            qefm[i] <- (0.6*MAF)
            LIHflo[i] <- 1    #i denotes a low flow month
            } else if(MMF[i, 2] > 0.4*MAF && MMF[i, 2] <= 0.8*MAF) {
              qefm[i] <- as.numeric(0.45*MMF[i, 2]) #SK:I added as.numeric() here,
              #otherwise it produces a list of different classes. There might be a more efficient solut
              LIHflo[i] <- 2    #i denotes an intermediate flow month
              } else {
                qefm[i] <-(0.3*MAF)
                LIHflo[i] <- 3 #i denotes a high flow month
              }
          }
        }
     else if (i_mvef==2){                     # --------------- Tessmann (1980)
       for (i in 1:nrow(MMF)) {
        if (MMF[i, 2] <= 0.4*MAF) {
         qefm[i] <- as.numeric(MMF[i, 2]) #sk:see note above
         LIHflo[i] <- 1 #i denotes a low flow month
         } else if(MMF[i, 2] > 0.4*MAF && 0.4*MMF[i, 2] <= 0.4*MAF) {
           qefm[i] <- 0.4*MAF
```

```
          LIHflo[i] <- 2 #i denotes an intermediate flow month
          } else {
            qefm[i] <- 0.4*MAF
            LIHflo[i] <- 3 #i denotes a high flow month
      }
    }
  }
     else if (i_mvef==3) {                  #--------------- Smakhtin et al. (2004b)
       for (i in 1:nrow(MMF)) {
         if (MMF[i, 2] <= MAF) {
         qefm[i] <- q90[[1]] #sk:also for quantile data, instead of as.numeric,
                                # I just indicated the data to extract, need to test later on
         LIHflo[i] <- 1 #i denotes a low flow month
         } else {
           qefm[i] <- 0.2*MAF # check this rule
           LIHflo[i] <- 3 #i denotes a high flow month
      }
    }
  }
     else if (i_mvef==4) {                  #--------------- Tennant (1976)
      for (i in 1:nrow(MMF)) {
       if (MMF[i, 2] <= MAF) {
        qefm[i] <- 0.2*MAF
        LIHflo[i] <- 1 #i denotes a low flow month
        } else {
         qefm[i] <- 0.4*MAF
         LIHflo[i] <- 3 #i denotes a high flow month
      }
    }
  }
     else if (i_mvef==5) {                  #--------------- Q90-Q50 (2014)
      for (i in 1:nrow(MMF)) {
       if (MMF[i, 2] <= MAF) {
        qefm[i] <-q90[[1]]
        LIHflo[i] <- 1 #i denotes a low flow month
        } else {
         qefm[i] <- q50[[1]]
         LIHflo[i] <- 3 #i denotes a high flow month
      }
    }
  }
} else { #case 3
  #---------------- eflow proportional to the incoming flow -----------
   for (i in 1:nrow(MMF)) {
    if (MMF[i,2] < MAF) {
      qefm[i] <- q90[[1]]
      } else {
       qefm[i] <- q50[[1]]
    }
  }
}
```

Conversion of monthly data to daily data

```r
ndays <- c(31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31)
qefy <-vector("numeric", 365)
nsum <- 0
for (i in seq_along(ndays)){
  qefy[(nsum+1):(nsum+(ndays[i]))] <- rep(qefm[[i]], ndays[i])
  nsum <- ndays[i]+nsum
}

qef <- rep(qefy, ny) #GZ: Check for leap years
#SK: leap years is doable with lubridate package
#This gives a vector of 730 which makes sense, Guido's give 731,
#did he manually added the leap year? maybe with q90 at the end?
qef <- c(qef[1:32], qef[32], qef[33:730])
```

Reservoir operation module

We kept the original code from Guido can also be done differently. The parameters are as follows;

- qhp: average discharge for the power production
- qcons: consumptive water use time series assumed as MAF
- qnconc: non-consumptive water use tume series mased on input dam power

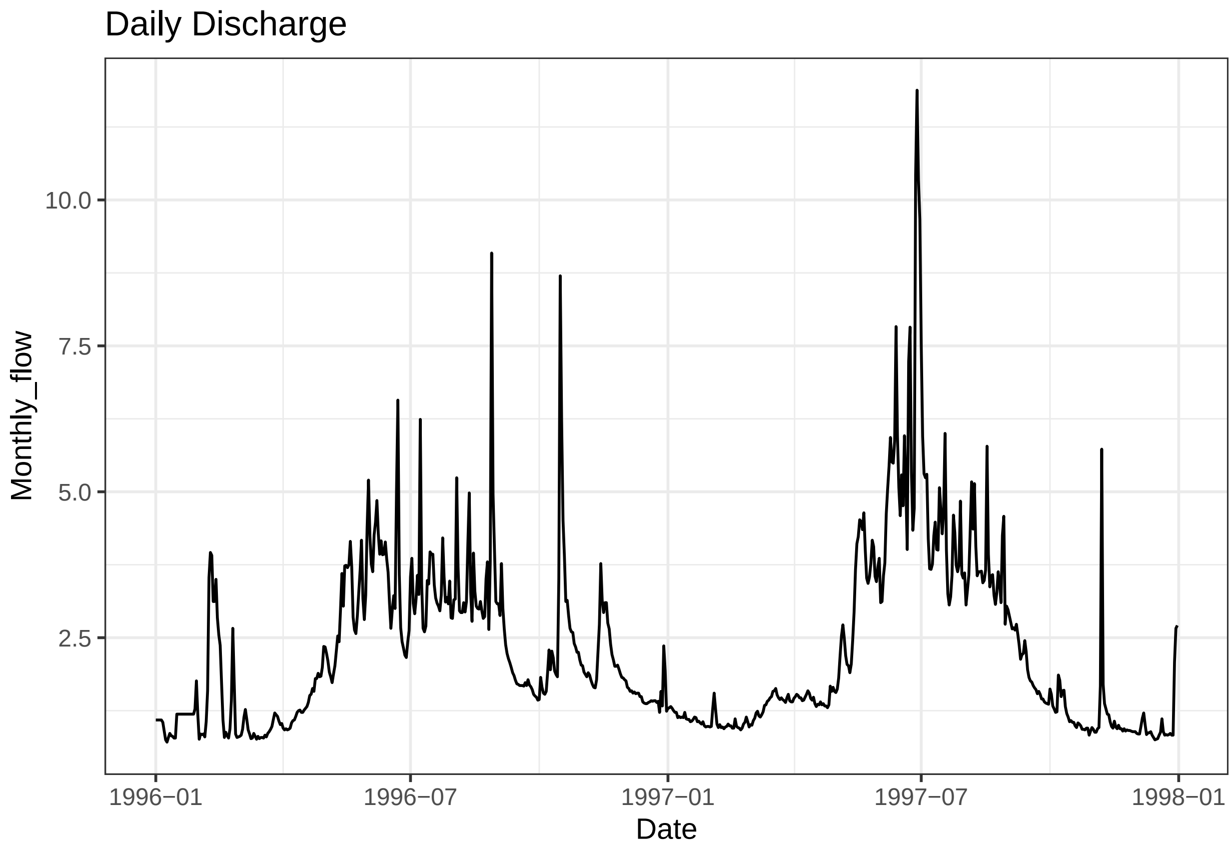
```r
qhp <- power/damhei/9810              #(m3/s)
qcons <- rep(q90[[1,1]], nq)          #(m3/s) F
qncons <- rep (qhp,nq)                #(m3/s)
qspill <- rep(0,nq)
qdown <- rep(0,nq)
qout <- rep(0, nq)
qnet <- rep(0, nq)
unmdem <- 0
unmdays <- 0
rvol <- rep(0, nq)                    # initialization
rvol[1] <- rvmax/2                    # initial reservoir volume
dt <- 86400                           # seconds in a day
```

The dam operation

```r
for (i in 1:(nq-1)){
  qdown[i] <- qspill[i]+icons*qef[i]+(1-icons)*max(qef[i], qncons[i])
  qout[i] <- qdown[i]+icons*qcons[i]
  qnet[i] <- q_tt[["Monthly_flow"]][[i]]-qout[i]
  rvol[i+1] <- rvol[i]+qnet[i]*dt
  if (rvol[i]>rvmax){
    qspill[i+1] <- rvol[i]-rvmax
    rvol[i] <- rvmax
  } else if (rvol[i]<rvmin){
    unmdem <- unmdem+qcons[i]+qncons[i]      # total unmet demand (consumpt + non consumpt)
    unmdays <- unmdays+1                     # n. of days in which demand is unmet
    qncons[i] <- 0
    qcons[i] <- 0
    rvol[i] <- rvmin
  }
}
```

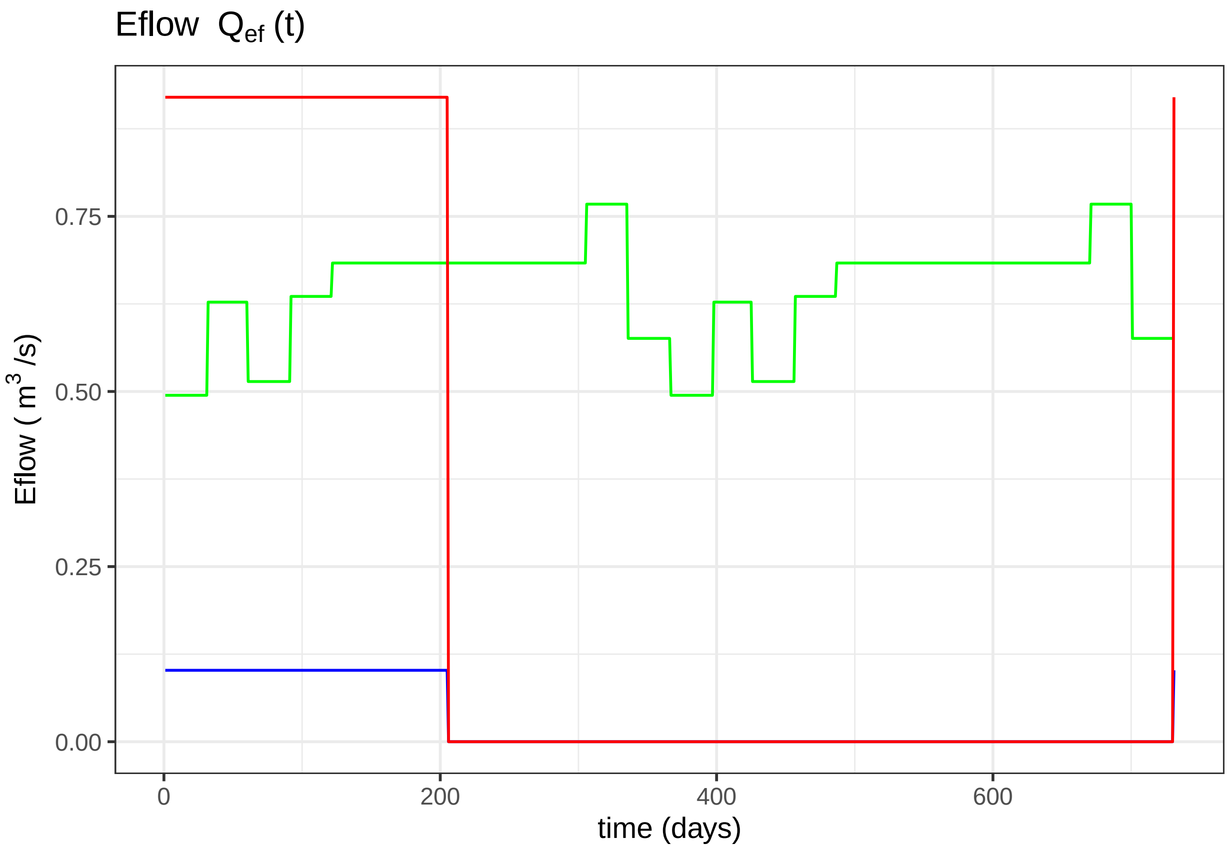
Plotting

```
flow <- ggplot(q_tt,aes(Date, Monthly_flow)) +
  geom_line()+theme_bw()+ggtitle("Daily Discharge")
flow
```

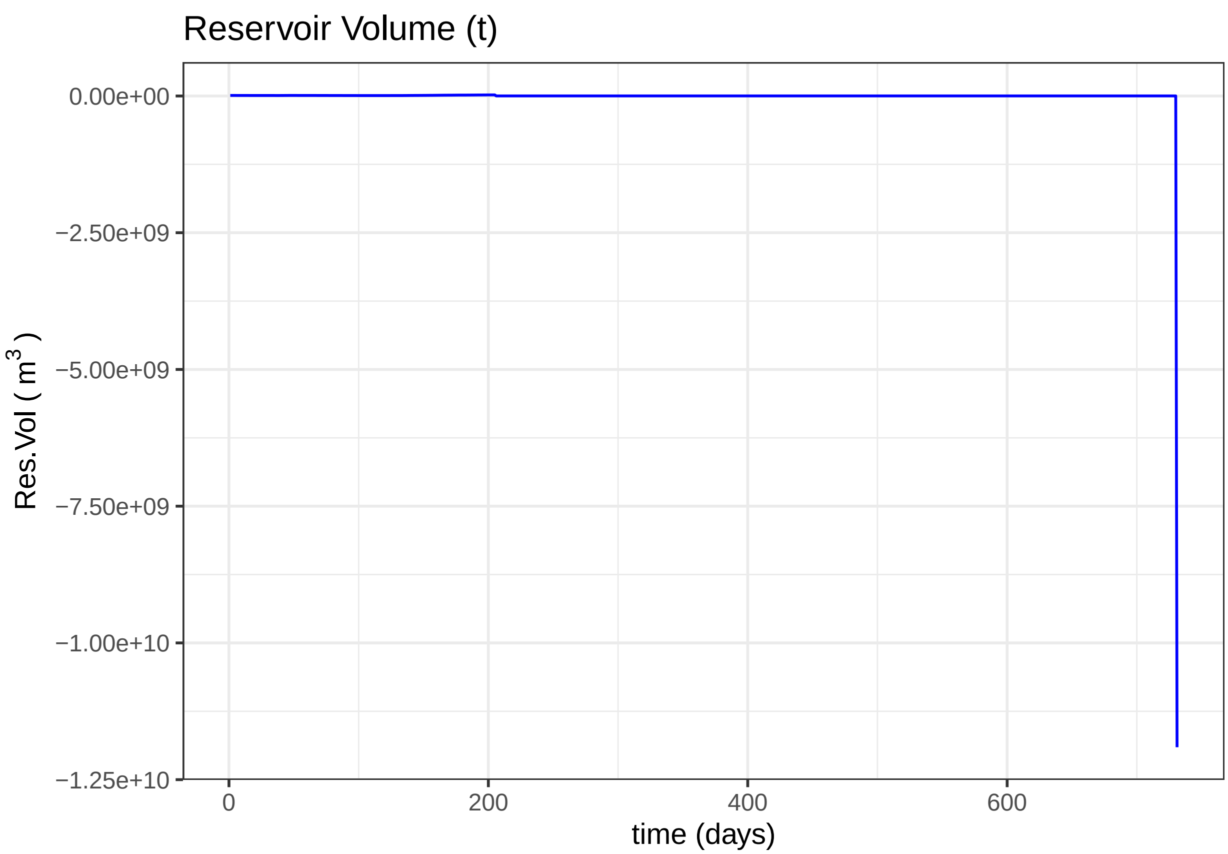


Calculated eflow plot

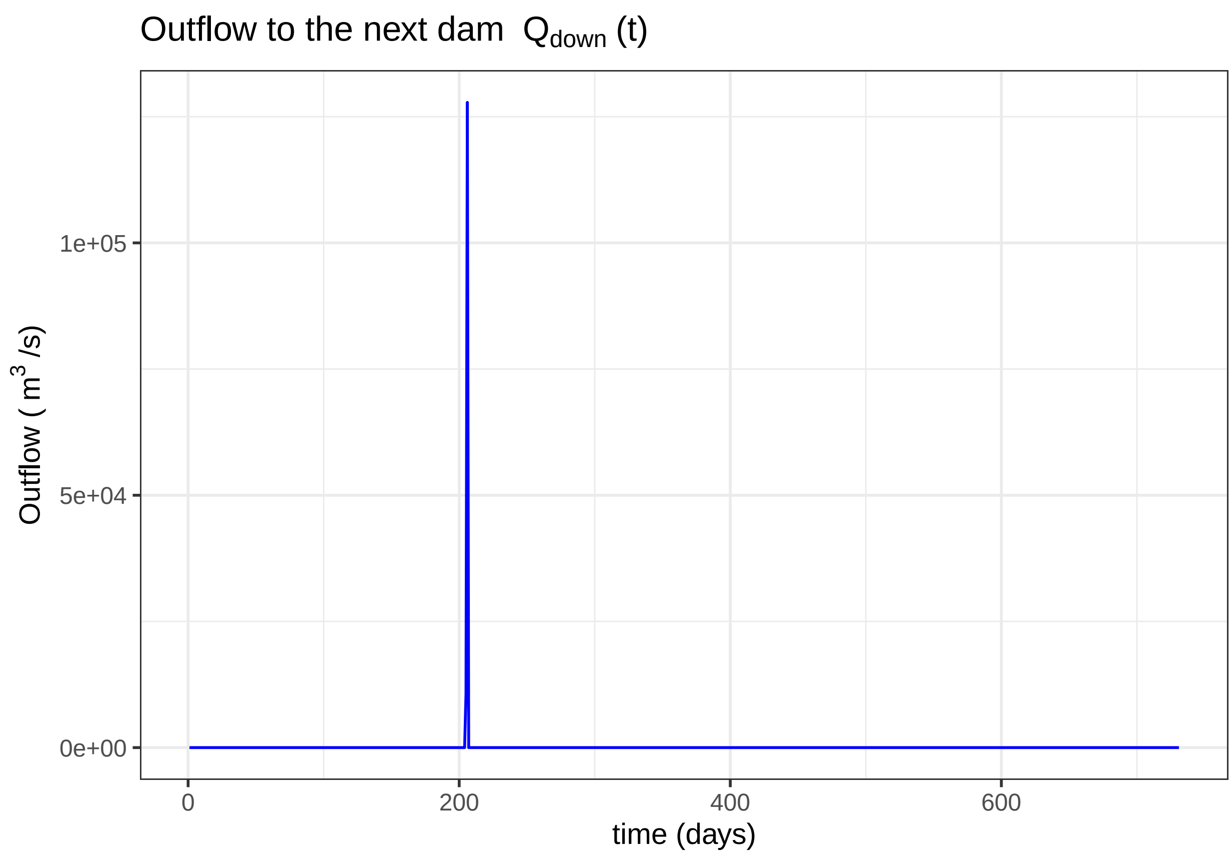
```
time_series <- ggplot()+ geom_line(aes(x=(1:nq), y=qef), color="green")+
  geom_line(aes(x=(1:nq), y=qncons), color="blue")+
  geom_line(aes(x=(1:nq), y=qcons), color="red")+
  theme_bw()+ ggtitle(bquote('Eflow ' ~Q[ef]~'(t)')) +
  xlab("time (days)") + ylab( bquote("Eflow ("~m^3~'/s)'))
time_series
```



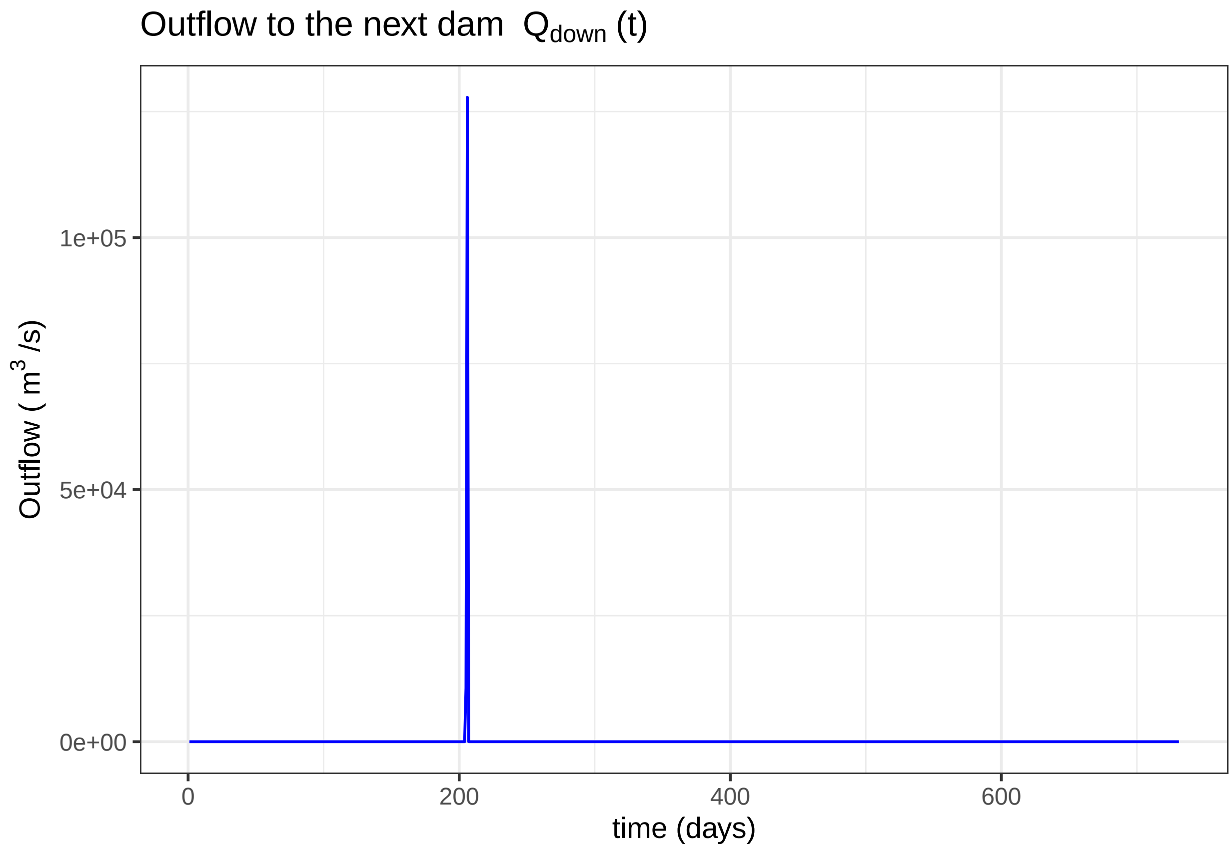

```
# Reservoir volume
reservoir_volume <- ggplot(q_tt,aes(x=(1:nq), rvol)) +
  geom_line(color="blue")+theme_bw()+ggtitle("Reservoir Volume (t)") +
  xlab("time (days)") + ylab( bquote("Res.Vol ("~m^3~')'))
reservoir_volume
```




```
# Outflow to the next dam
outflow_next_dam <- ggplot(q_tt,aes(x=(1:nq), qdown)) +
  geom_line(color="blue")+theme_bw()+ggtitle('Outflow to the next dam ' ~Q[down]~'(t)') +
  xlab("time (days)") + ylab( bquote("Outflow ("~m^3~'/s)'))
outflow_next_dam
```




```r
# Outflow to the next dam
excess_flow <- ggplot(q_tt,aes(x=(1:nq), qspill)) +
  geom_line(color="blue")+theme_bw()+ggtitle('Excess flow ' ~Q[spill]~'(t)') +
  xlab("time (days)") + ylab( bquote("Excess flow ("~m^3~'/s)'))
outflow_next_dam
```



Outflow to the next dam Q_down (t)
Outflow (m3 /s)
1e+05
5e+04
0e+00
0
200
400
600
time (days)