R-scripts

Water levels and water discharge values at Elverum

```
# Info from flood monument and systematic measurements:

floods <- c(1789,1995,1934,1916,1966,1967)                  # floods
hfloods <- c(3900,3239,2963,2892,2600,2533)                 # discharges
waterlevels <- c(266.5,198.5,155,142.5,99,96)               # height of flood mark (cm)
MyRating<-splinefun(waterlevels,hfloods)
MyRatingII<-approxfun(waterlevels,hfloods)

x = c(96:297)
y = MyRating(x)
yII = MyRatingII(x)

plot(y,x,type='l',ylab= 'Water level (cm)' ,xlab='Discharge(m3/s)')
points(hfloods,waterlevels,col=2)

# Plotting rating curve using a linear interpolating between the points:
plot(yII,x,type='l',main="Rating curve (Elverum flood monument)",ylab= 'Water level (cm)'
,xlab='Discharge(m3/s)')
points(hfloods,waterlevels,col=2)
```

Similarity measures

```
# Analyze_floods.R
# Set work directory
setwd('C:/Users/Anna/Desktop/R. Analyze floods')

# import files
Age_model <- read.csv("C:/Users/Anna/Desktop/R. Analyze floods/Age_model.csv", sep=";")
Flooddata <- read.csv("C:/Users/Anna/Desktop/R. Analyze floods/Flooddata.csv", sep=";")
SedimentdataNEW <- read.csv("C:/Users/Anna/Desktop/R. Analyze
floods/SedimentdataNEWthreshold.csv", header=FALSE, sep=";")

# make matrix named "floodinfo"
[1]  = Year
[2]  = Uncertainty-Age
[3]  = Roc > P94
[4]  = RoC
[5]  = RoC > P90
[6]  = RoC
[7]  = Grey > P94
[8]  = Grey
[9]  = Grey > P90
[10] = Grey
[11]      = RoC > P93
[12]      = RoC
```

[13]	= RoC > P92
[14]	= RoC
[15]	= RoC > P91
[16]	= RoC
[17]	= RoC > P95
[18]	= RoC
[19]	= RoC > P96
[20]	= RoC
[21]	= RoC > P97
[22]	= RoC
[23]	= RoC > P98
[24]	= RoC
[25]	= Nfloods = number of floods
[26]	= Volume = transferred water amount
[27]	= Duration
[28]	= Peak = max water discharge
[29]	= Season (spring/autumn)
NA

# "Year"	is year.
# "Uncertainty-Age"	is the uncertainty in year for the sediments
# "Roc94"	is the number of sediment layers where the RoC-value is higher than 94% percentile for the given year.
# "RoC"	is the biggest RoC-value for the given year.
# "RoC90"	is the number of sediment layers where the RoC-value is higher than 90% percentile for the given year.
# "RoC"	is the biggest RoC-value for the given year.
# "Grey94"	is the number of sediment layers where the Grey-value is higher than 94% percentile for the given year.
# "Grey"	is the biggest Grey-value for the given year.
# "Grey90"	is the number of sediment layers where the Grey- value is higher than 90% percentile for the given year.
# "Grey"	is the biggest Grey value for the given year.
# "Nfloods"	is number of floods with bifurcation.
# "Volume"	is flood volume (10^6 m3).
# "Duration"	is duration of the flood (in days).
# "Peak"	is the flood peak.
# "Season"	is season where the biggest flood appear.

```
# Load data:
source('Load_dataNEW2threshold.R')
source('Analyze_functions.R')

# Load_dataNEW2threshold.R
# Load_dataNEW2.R

# read the flood data
flooddata<-read.table('Flooddata.csv',sep=";",header=TRUE,row.names=NULL)
```

```
# Sequence of years
floodyears<-seq(1945,2013,by=1)

#get the month for each event
fmonth<-as.integer(substr(flooddata[,6],4,5))

#get the day in month for each event
fday<-as.integer(substr(flooddata[,6],1,2))

# get the day number in year for each date
ml<-c(0,31,28,31,30,31,30,31,31,30,31,30)
ml<-cumsum(ml)
dayinyear<-ml[fmonth]+fday

# get the date at decimal-year
ftimes<-as.integer(flooddata[,1])+(dayinyear)/365

# Now we have a flood data matrix with only flood events
flooddata[,1]<-ftimes


# Read the sediment data
sediments<-read.table('SedimentdataNEW.csv',sep=";",header=TRUE)
sed_roc_94<-sediments[sediments[,7]==1,]
sed_roc_90<-sediments[sediments[,8]==1,]
sed_grey_94<-sediments[sediments[,9]==1,]
sed_grey_90<-sediments[sediments[,10]==1,]
sed_roc_93<-sediments[sediments[,11]==1,]
sed_roc_92<-sediments[sediments[,12]==1,]
sed_roc_91<-sediments[sediments[,13]==1,]


# Read the uncertainty model
agemodel<-read.table('Age_model.csv',sep=";",header=TRUE)
ai<-dim(agemodel)[1]-1
syears<-sort(na.omit(floor(sediments[,5])))
myears<-unique(floodyears)
uncertainty<-rep(NA,length(myears))

for(i in 1 : ai){

  uncertainty[myears > agemodel[i+1,2] & myears <= agemodel[i,2]] <-agemodel[i,3]

}

uncertainty[myears<=agemodel[ai,2]]<-agemodel[ai,3]
ucmodel<-cbind(myears,uncertainty)

# select only one event per year
imax<-by(flooddata,as.integer(flooddata[,1]),function(x) x[which.max(x[,2]),],simplify=TRUE)
amax<-matrix(unlist(imax),ncol=6,byrow=TRUE)

# Now make a matrix with the annual occurrence of floods.
```

```
# Make a matrix with flood sequences, i.e. one value for each year
floodsequence<-matrix(0,nrow=length(floodyears),ncol=dim(amax)[2])

# Need to find in which row to put the floods
fi<-na.omit(match(floor(amax[,1]),floodyears))

# number of floods for each year
nfloods<-as.integer(by(flooddata[,2]>0,floor(flooddata[,1]),sum))

floodsequence[]<-matrix(0,nrow=length(floodyears),ncol=dim(amax)[2])

# Write the flood information to the complete flood matrix
floodsequence[fi,1]<-nfloods
floodsequence[fi,2]<-amax[,2]
floodsequence[fi,3]<-amax[,3]
floodsequence[fi,4]<-amax[,4]
floodsequence[fi,5]<-amax[,5]
floodsequence[fi,6]<-amax[,6]


# Extract the different flood information. Calculate number of sediment flood events per year.

s1<-as.integer(by(sediments[,7],syears,sum))
s2<-as.integer(by(sediments[,8],syears,sum))
s3<-as.integer(by(sediments[,9],syears,sum))
s4<-as.integer(by(sediments[,10],syears,sum))
s5<-as.integer(by(sediments[,11],syears,sum))
s6<-as.integer(by(sediments[,12],syears,sum))
s7<-as.integer(by(sediments[,13],syears,sum))


s1<-c(rep(0,min(syears)-min(floodyears)),s1)
s2<-c(rep(0,min(syears)-min(floodyears)),s2)
s3<-c(rep(0,min(syears)-min(floodyears)),s3)
s4<-c(rep(0,min(syears)-min(floodyears)),s4)
s5<-c(rep(0,min(syears)-min(floodyears)),s5)
s6<-c(rep(0,min(syears)-min(floodyears)),s6)
s7<-c(rep(0,min(syears)-min(floodyears)),s7)


# Sed roc 94
imax<-by(sed_roc_94,as.integer(sed_roc_94[,5]),function(x) x[which.max(x[,4]),],simplify=TRUE)
sed_roc_94_max<-matrix(as.numeric(unlist(imax)),ncol=dim(sed_roc_94)[2],byrow=TRUE)
si1<-rep(0,length(floodyears))
si1[na.omit(match(floor(sed_roc_94_max[,5]),floodyears))]<-sed_roc_94_max[,4]

# Sed roc 90
imax<-by(sed_roc_90,as.integer(sed_roc_90[,5]),function(x) x[which.max(x[,4]),],simplify=TRUE)
sed_roc_90_max<-matrix(as.numeric(unlist(imax)),ncol=dim(sed_roc_90)[2],byrow=TRUE)
```

```
si2<-rep(0,length(floodyears))
si2[na.omit(match(floor(sed_roc_90_max[,5]),floodyears))]<-sed_roc_90_max[,4]

# sed grey 94
imax<-by(sed_grey_94,as.integer(sed_grey_94[,5]),function(x) x[which.max(x[,3]),],simplify=TRUE)
sed_grey_94_max<-matrix(as.numeric(unlist(imax)),ncol=dim(sed_grey_94)[2],byrow=TRUE)
si3<-rep(0,length(floodyears))
si3[na.omit(match(floor(sed_grey_94_max[,5]),floodyears))]<-sed_grey_94_max[,3]

# sed grey 90
imax<-by(sed_grey_90,as.integer(sed_grey_90[,5]),function(x) x[which.max(x[,3]),],simplify=TRUE)
sed_grey_90_max<-matrix(as.numeric(unlist(imax)),ncol=dim(sed_grey_94)[2],byrow=TRUE)
si4<-rep(0,length(floodyears))
si4[na.omit(match(floor(sed_grey_90_max[,5]),floodyears))]<-sed_grey_90_max[,3]

# sed roc 93
imax<-by(sed_roc_93,as.integer(sed_roc_93[,5]),function(x) x[which.max(x[,4]),],simplify=TRUE)
sed_roc_93_max<-matrix(as.numeric(unlist(imax)),ncol=dim(sed_roc_93)[2],byrow=TRUE)
si5<-rep(0,length(floodyears))
si5[na.omit(match(floor(sed_roc_93_max[,5]),floodyears))]<-sed_roc_93_max[,4]

# sed roc 92
imax<-by(sed_roc_92,as.integer(sed_roc_92[,5]),function(x) x[which.max(x[,4]),],simplify=TRUE)
sed_roc_92_max<-matrix(as.numeric(unlist(imax)),ncol=dim(sed_roc_92)[2],byrow=TRUE)
si6<-rep(0,length(floodyears))
si6[na.omit(match(floor(sed_roc_92_max[,5]),floodyears))]<-sed_roc_92_max[,4]

# sed roc 91
imax<-by(sed_roc_91,as.integer(sed_roc_91[,5]),function(x) x[which.max(x[,4]),],simplify=TRUE)
sed_roc_91_max<-matrix(as.numeric(unlist(imax)),ncol=dim(sed_roc_91)[2],byrow=TRUE)
si7<-rep(0,length(floodyears))
si7[na.omit(match(floor(sed_roc_91_max[,5]),floodyears))]<-sed_roc_91_max[,4]


floodinfo<-data.frame(cbind(ucmodel,s1,si1,s2,si2,s3,si3,s4,si4,s5,si5,s6,si6,s7,si7,floodsequence))

colnames(floodinfo)<-c("Year","Uncertainty-Age","Roc94","RoC","RoC90","RoC","Grey94","Grey","Grey90","Grey","RoC 93","RoC","RoC 92","RoC","RoC 91","RoC","Nfloods","overført mengde","Duration","max vannføring","Season","something")

rownames(floodinfo)<-floodyears


# Analyze_functions.R

# Plotting data.
plot_floods_sediments<-function(sinfo,finfo,ucmodel,vindex=4,tindex=5,ubars=TRUE,ytxt="RoC"){

  windows(12,6)
  xll<-c(ucmodel[1,1], ucmodel[dim(ucmodel)[1],1])
```

```
  par(mar=c(5, 4, 4, 4) + 0.1)
  plot(sinfo[,tindex],sinfo[,vindex],xlab="Year",ylab=ytxt,xlim=xll,col="red",cex=1.5,pch=19)
  lines(sinfo[,tindex],sinfo[,vindex],xlim=xll,col="red",type="h")
  epsilon = max(sinfo[,vindex])/100
  siy<-floor(sinfo[,tindex])

  tci<-ucmodel[match(siy,ucmodel[,1]),2]
  segments((sinfo[,tindex]-tci), sinfo[,vindex],(sinfo[,tindex]+tci), sinfo[,vindex],col=2)

  segments((sinfo[,tindex]-tci),sinfo[,vindex]-epsilon,(sinfo[,tindex]-tci),sinfo[,vindex]+epsilon,col=2)
  segments((sinfo[,tindex]+tci),sinfo[,vindex]-epsilon,(sinfo[,tindex]+tci),sinfo[,vindex]+epsilon,col=2)
  par(new=TRUE)

  plot(finfo[,1],finfo[,2],axes=F,col="blue",pch=18,ylim=c(0,30),cex=1.5,xlab="",ylab="",xlim=xll)
  par(new=TRUE)


plot(finfo[,1],finfo[,2],axes=F,col="blue",pch=18,ylim=c(0,30),cex=1.5,xlab="",ylab="",xlim=xll,type="h"
)

  axis(4)
  mtext("Volume (M m3)", side = 4, line = 3)
}


# make a contingecy table
make_ctable<-function(x,y){
  xx<-sort(unique(x))
  yy<-sort(unique(y))
  nx<-length(xx)
  ny<-length(yy)
  ct<-matrix(NA,nx,ny)
  colnames(ct)<-paste("Flood_",xx,sep="")
  rownames(ct)<-paste("Sediment_",yy,sep="")
#  dimnames(ct)<-list("Sediment","Streamflow")
  for(i in 1: nx){
    for(j in 1 : ny){
      ct[j,i]<-sum(as.integer(x)==as.integer(xx[i])&as.integer(y)==as.integer(yy[j]))
    }
  }
    out<-list()
  out$kulzinsky <-ct[2,2]/(ct[1,2]+ct[2,1])
  out$correlation <- (ct[2,2]*ct[1,1]-
ct[1,2]*ct[2,1])/(sqrt((ct[2,1]+ct[2,2])*(ct[1,2]+ct[1,1])*(ct[2,2]+ct[1,2])*(ct[1,1]+ct[2,1])))
  out$dice <-ct[2,2]/(ct[1,2]+ct[2,1]+2*ct[2,2])
  out$jaccardneedham<-ct[2,2]/(ct[1,2]+ct[2,1]+ct[2,2])
  out$ct <-ct
  return(out)
}
```

Additional, the following matrix are made:
matrix with the biggest floods

```
sed_roc_94_max          # 21 values
sed_roc_92_max          # 24 values
sed_roc_90_max          # 31 values
sed_grey_94_max         # 16 values
sed_grey_90_max         # 26 values
```

Graphs with the different thresholds & instrumental known bifurcations

```
plot_floods_sediments(sed_roc_94_max,flooddata,ucmodel,vindex=4,tindex=5,ubars=TRUE,ytxt="RoC 94")

plot_floods_sediments(sed_roc_92_max,flooddata,ucmodel,vindex=4,tindex=5,ubars=TRUE,ytxt="RoC 92")

plot_floods_sediments(sed_roc_90_max,flooddata,ucmodel,vindex=4,tindex=5,ubars=TRUE,ytxt="RoC 90")

plot_floods_sediments(sed_grey_94_max,flooddata,ucmodel,vindex=3,tindex=5,ubars=TRUE,ytxt="Grey scale 94")

plot_floods_sediments(sed_grey_90_max,flooddata,ucmodel,vindex=3,tindex=5,ubars=TRUE,ytxt="Grey scale 90")
```

Flood-info from R-studio

```
> sed_roc_94_max
           [,1] [,2]    [,3]     [,4]     [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
 [1,] 178217.00  226 20894.5 23057.94 1954.728   NA    1    1    1     1     0     0     1
 [2,] 174617.00  216 21268.9 33720.30 1956.393   NA    1    1    1     1     1     1     1
 [3,] 161656.00  180 32616.3 35337.79 1962.225   NA    1    1    1     1     1     1     1
 [4,] 157695.00  169 22340.7 39833.58 1963.956   NA    1    1    1     1     1     1     1
 [5,] 150855.00  150 22238.5 38996.50 1966.889   NA    1    1    1     1     0     1     1
 [6,] 142214.00  126 20638.7 18094.47 1970.492   NA    1    1    0     1     0     0     0
 [7,] 126732.00  428 19485.3 19933.48 1976.663   NA    1    1    0     0     0     0     0
 [8,] 116291.00  462 20672.5 17852.07 1980.617   NA    1    1    0     1     0     0     0
 [9,] 110891.00   85 20773.7 27765.03 1982.597   NA    1    1    1     1     0     0     0
[10,] 102610.00   62 18654.0 17521.61 1985.546   NA    1    1    0     0     0     0     0
[11,]  97929.50  417 19294.7 18981.60 1987.167   NA    1    1    0     0     1     1     1
[12,]  95409.30  410 20855.3 16948.05 1988.026   NA    1    1    1     1     0     0     0
[13,]  88208.40  394 19379.0 23424.55 1990.426   NA    1    1    0     0     0     0     0
[14,]  83167.90  381 19892.5 17161.73 1992.059   NA    1    1    0     1     0     0     0
[15,]  73807.00  440 22515.7 42577.90 1994.989   NA    1    1    1     1     1     1     1
[16,]  64446.70  337 17561.9 19909.89 1997.785   NA    1    1    0     0     1     1     1
[17,]  57246.00  449 20265.2 26176.55 1999.844   NA    1    1    0     1     0     0     0
[18,]  55805.80  318 21545.4 28550.88 2000.247   NA    1    1    1     1     0     0     0
[19,]  41764.50  281 20046.5 26453.10 2004.004   NA    1    1    0     1     0     0     0
[20,]  25202.80  245 18805.6 20218.62 2008.049   NA    1    1    0     0     0     0     0
[21,]   5760.81   17 20982.5 35032.85 2012.262   NA    1    1    1     1     0     0     0
```

```
> sed_roc_92_max

           [,1] [,2]    [,3]      [,4]     [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
 [1,] 178577.00  227 17029.2  1083.224 1954.560   NA    0    0    0     0     1     1     1
 [2,] 174617.00  216 21268.9 33720.298 1956.393   NA    1    1    1     1     1     1     1
```

```
 [3,] 163096.00  184 19957.9 16154.639 1961.590   NA    0    1    0     1     1     1     1
 [4,] 161656.00  180 32616.3 35337.789 1962.225   NA    1    1    1     1     1     1     1
 [5,] 157695.00  169 22340.7 39833.579 1963.956   NA    1    1    1     1     1     1     1
 [6,] 150855.00  150 22238.5 38996.498 1966.889   NA    1    1    1     1     0     1     1
 [7,] 141134.00  123 20521.8 17427.850 1970.935   NA    1    1    0     1     1     1     1
 [8,] 116651.00  458 18288.0  4637.133 1980.483   NA    0    0    0     0     1     1     1
 [9,] 112691.00  462 17686.5  9208.929 1981.942   NA    0    0    0     0     1     1     1
[10,] 105490.00   70 17405.1  6674.055 1984.532   NA    0    0    0     0     0     1     1
[11,]  97929.50  417 19294.7 18981.597 1987.167   NA    1    1    0     0     1     1     1
[12,]  88568.60  395 16610.5 19529.530 1990.308   NA    1    1    0     0     1     1     1
[13,]  83528.10  382 17911.8  6333.372 1991.944   NA    0    0    0     0     1     1     1
[14,]  79207.50  371 18623.1 15337.995 1993.315   NA    0    1    0     0     1     1     1
[15,]  73807.00  440 22515.7 42577.897 1994.989   NA    1    1    1     1     1     1     1
[16,]  64446.70  337 17561.9 19909.892 1997.785   NA    1    1    0     0     1     1     1
[17,]  61206.30  328 18226.3  8266.155 1998.721   NA    0    0    0     0     1     1     1
[18,]  54725.60  315 17772.2 14481.841 2000.547   NA    0    1    0     0     1     1     1
[19,]  42124.40  282 17598.3  8771.494 2003.912   NA    0    0    0     0     1     1     1
[20,]  36723.60  269 17283.2 13188.890 2005.280   NA    0    1    0     0     0     1     1
[21,]  33483.60  262 17479.5 11093.672 2006.079   NA    0    0    0     0     1     1     1
[22,]  25562.90  246 17117.6 10247.796 2007.966   NA    0    0    0     0     1     1     1
[23,]  21242.20  234 17749.8  5905.105 2008.954   NA    0    0    0     0     0     1     1
[24,]   3960.83   12 19094.5 23437.323 2012.623   NA    1    1    0     0     1     1     1
```

```
> sed_roc_90_max
          [,1] [,2]    [,3]     [,4]     [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13] [,14] [,15] [,16] [,17] [,18]
 [1,] 178217.00  226 20894.5 23057.94 1954.728   NA    1    1    1     1     0     0     1     0     0     0     0     0
 [2,] 174617.00  216 21268.9 33720.30 1956.393   NA    1    1    1     1     1     1     1     1     1     1     0     0
 [3,] 163096.00  184 19957.9 16154.64 1961.590   NA    0    1    0     1     1     1     1     1     1     0     0     0
 [4,] 161656.00  180 32616.3 35337.79 1962.225   NA    1    1    1     1     1     1     1     1     0     0     0     0
 [5,] 157695.00  169 22340.7 39833.58 1963.956   NA    1    1    1     1     1     1     1     0     0     0     0     0
 [6,] 156975.00  167 21834.5 16700.29 1964.269   NA    0    1    1     1     0     0     0     0     0     0     0     0
 [7,] 150855.00  150 22238.5 38996.50 1966.889   NA    1    1    1     1     0     1     1     0     0     0     0     0
 [8,] 142214.00  126 20638.7 18094.47 1970.492   NA    1    1    0     1     0     0     0     0     0     0     0     0
 [9,] 126732.00  428 19485.3 19933.48 1976.663   NA    1    1    0     0     0     0     0     0     0     0     0     0
[10,] 116291.00  462 20672.5 17852.07 1980.617   NA    1    1    0     1     0     0     0     0     0     0     0     0
[11,] 114491.00  462 19287.5 14403.40 1981.282   NA    0    1    0     0     0     0     0     0     0     0     0     0
[12,] 110891.00   85 20773.7 27765.03 1982.597   NA    1    1    1     1     0     0     0     0     0     0     0     0
[13,] 105130.00   69 19063.3 13012.58 1984.660   NA    0    1    0     0     0     0     0     0     0     0     0     0
[14,] 102610.00   62 18654.0 17521.61 1985.546   NA    1    1    0     0     0     0     0     0     0     0     0     0
[15,]  98649.40  419 19218.9 13966.74 1986.920   NA    0    1    0     0     0     0     0     0     0     0     0     0
[16,]  97929.50  417 19294.7 18981.60 1987.167   NA    1    1    0     0     1     1     1     1     1     0     0     0
[17,]  95409.30  410 20855.3 16948.05 1988.026   NA    1    1    1     1     0     0     0     0     0     0     0     0
[18,]  88208.40  394 19379.0 23424.55 1990.426   NA    1    1    0     0     0     0     0     0     0     0     0     0
[19,]  83167.90  381 19892.5 17161.73 1992.059   NA    1    1    0     1     0     0     0     0     0     0     0     0
[20,]  78487.60  369 21997.3 15444.76 1993.541   NA    0    1    1     1     0     0     0     0     0     0     0     0
[21,]  73807.00  440 22515.7 42577.90 1994.989   NA    1    1    1     1     1     1     1     1     1     1     1     0
[22,]  64446.70  337 17561.9 19909.89 1997.785   NA    1    1    0     0     1     1     1     0     0     0     0     0
[23,]  60846.40  327 19833.9 15601.35 1998.824   NA    0    1    0     1     0     0     0     0     0     0     0     0
[24,]  57246.00  449 20265.2 26176.55 1999.844   NA    1    1    0     1     0     0     0     0     0     0     0     0
[25,]  55805.80  318 21545.4 28550.88 2000.247   NA    1    1    1     1     0     0     0     0     0     0     0     0
[26,]  41764.50  281 20046.5 26453.10 2004.004   NA    1    1    0     1     0     0     0     0     0     0     0     0
[27,]  36004.00  451 19228.1 15661.19 2005.459   NA    0    1    0     0     0     0     0     0     0     0     0     0
[28,]  27723.00  436 18055.2 13636.93 2007.461   NA    0    1    0     0     0     0     0     0     0     0     0     0
[29,]  25202.80  245 18805.6 20218.62 2008.049   NA    1    1    0     0     0     0     0     0     0     0     0     0
[30,]  20882.50  233 18802.6 12994.32 2009.035   NA    0    1    0     0     0     0     0     0     0     0     0     0
[31,]   5760.81   17 20982.5 35032.85 2012.262   NA    1    1    1     1     0     0     0     0     0     0     0     0
```

```
> sed_grey_94_max
          [,1] [,2]    [,3]       [,4]     [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
 [1,] 178217.00  226 20894.5 23057.9410 1954.728   NA    1    1    1     1     0     0     1
 [2,] 173897.00  214 22051.0  7183.9239 1956.724   NA    0    0    1     1     0     0     0
 [3,] 172097.00  209 23123.7   525.3619 1957.547   NA    0    0    1     1     0     0     0
 [4,] 162376.00  182 22704.5  7708.8905 1961.908   NA    0    0    1     1     1     1     1
 [5,] 161656.00  180 32616.3 35337.7895 1962.225   NA    1    1    1     1     1     1     1
 [6,] 157695.00  169 22340.7 39833.5788 1963.956   NA    1    1    1     1     1     1     1
 [7,] 156975.00  167 21834.5 16700.2863 1964.269   NA    0    1    1     1     0     0     0
 [8,] 150855.00  150 22238.5 38996.4978 1966.889   NA    1    1    1     1     0     1     1
```

```
 [9,] 140774.00   122 21709.4  8076.8470 1971.082   NA    0    0    1     1     0     0     0
[10,] 110891.00    85 20773.7 27765.0298 1982.597   NA    1    1    1     1     0     0     0
[11,]  95409.30   410 20855.3 16948.0523 1988.026   NA    1    1    1     1     0     0     0
[12,]  78487.60   369 21997.3 15444.7628 1993.541   NA    0    1    1     1     0     0     0
[13,]  73807.00   440 22515.7 42577.8974 1994.989   NA    1    1    1     1     1     1     1
[14,]  73087.00   442 22590.6  9685.4513 1995.209   NA    0    0    1     1     0     0     0
[15,]  55805.80   318 21545.4 28550.8793 2000.247   NA    1    1    1     1     0     0     0
[16,]   5760.81    17 20982.5 35032.8486 2012.262   NA    1    1    1     1     0     0     0
```

```
> sed_grey_90_max
           [,1] [,2]    [,3]       [,4]     [,5] [,6] [,7] [,8] [,9] [,10] [,11] [,12] [,13]
 [1,] 178217.00   226 20894.5 23057.9410 1954.728   NA    1    1    1     1     0     0     1
 [2,] 173897.00   214 22051.0  7183.9239 1956.724   NA    0    0    1     1     0     0     0
 [3,] 172097.00   209 23123.7   525.3619 1957.547   NA    0    0    1     1     0     0     0
 [4,] 162376.00   182 22704.5  7708.8905 1961.908   NA    0    0    1     1     1     1     1
 [5,] 161656.00   180 32616.3 35337.7895 1962.225   NA    1    1    1     1     1     1     1
 [6,] 157695.00   169 22340.7 39833.5788 1963.956   NA    1    1    1     1     1     1     1
 [7,] 156975.00   167 21834.5 16700.2863 1964.269   NA    0    1    1     1     0     0     0
 [8,] 150855.00   150 22238.5 38996.4978 1966.889   NA    1    1    1     1     0     1     1
 [9,] 148334.00   143 19611.5  8672.5420 1967.952   NA    0    0    0     1     0     0     0
[10,] 142214.00   126 20638.7 18094.4705 1970.492   NA    1    1    0     1     0     0     0
[11,] 140774.00   122 21709.4  8076.8470 1971.082   NA    0    0    1     1     0     0     0
[12,] 116291.00   462 20672.5 17852.0669 1980.617   NA    1    1    0     1     0     0     0
[13,] 110891.00    85 20773.7 27765.0298 1982.597   NA    1    1    1     1     0     0     0
[14,]  97569.40   416 20295.5  8116.4026 1987.290   NA    0    0    0     1     0     0     0
[15,]  95409.30   410 20855.3 16948.0523 1988.026   NA    1    1    1     1     0     0     0
[16,]  83167.90   381 19892.5 17161.7332 1992.059   NA    1    1    0     1     0     0     0
[17,]  78487.60   369 21997.3 15444.7628 1993.541   NA    0    1    1     1     0     0     0
[18,]  73807.00   440 22515.7 42577.8974 1994.989   NA    1    1    1     1     1     1     1
[19,]  73087.00   442 22590.6  9685.4513 1995.209   NA    0    0    1     1     0     0     0
[20,]  66606.90   343 19832.6 16588.9903 1997.151   NA    0    1    0     1     0     0     0
[21,]  60846.40   327 19833.9 15601.3532 1998.824   NA    0    1    0     1     0     0     0
[22,]  57246.00   449 20265.2 26176.5463 1999.844   NA    1    1    0     1     0     0     0
[23,]  55805.80   318 21545.4 28550.8793 2000.247   NA    1    1    1     1     0     0     0
[24,]  52925.40   310 20164.8 11472.4070 2001.042   NA    0    0    0     1     0     0     0
[25,]  41764.50   281 20046.5 26453.1033 2004.004   NA    1    1    0     1     0     0     0
[26,]   5760.81    17 20982.5 35032.8486 2012.262   NA    1    1    1     1     0     0     0
```

Estimate correlation between bifurcation / flood events and flood layers in core
(floodinfo[,27] (= Nfloods = number of know floods/bifurcation events) is the not-shifting value/factor, and the other one is changing from RoC94, RoC92 etc.)

Contingency table
the following tables are in this form;

	No bifurcations	Bifurcations
No sediment layer	Correct negative	Miss
Sediment layer	False alarm	Hit

following is a CT-table and a CSI-value estimated and calculated for the thresholds; RoC94, RoC92, RoC90, Grey94 and Grey90:

Estimate similarity measures between bifurcations and RoC P94

```
corr_RoC94<-make_ctable(floodinfo[,27]>0,floodinfo[,3]>0)
> corr_RoC94
$kulzinsky[1] 0.1025641
$correlation      [1] -0.2569177
$dice             [1] 0.08510638
$jaccardneedham   [1] 0.09302326

$ct
```

```
             Flood_FALSE Flood_TRUE
Sediment_FALSE        26        18
Sediment_TRUE         21         4
```

```
# Estimate similarity measures between bifurcations and RoC 92
corr_RoC92<-make_ctable(floodinfo[,27]>0,floodinfo[,13]>0)
```

```
> corr_RoC92
$kulzinsky[1] 0.1351351
$correlation        [1] -0.1922194
$dice               [1] 0.106383
$jaccardneedham     [1] 0.1190476

$ct
             Flood_FALSE Flood_TRUE
Sediment_FALSE        27        17
Sediment_TRUE         20         5
```

```
# Estimate similarity measures between bifurcations and RoC P90
corr_RoC90<-make_ctable(floodinfo[,27]>0,floodinfo[,5]>0)
```

```
> corr_RoC90
$kulzinsky[1] 0.2702703
$correlation        [1] -0.07212009
$dice               [1] 0.1754386
$jaccardneedham     [1] 0.212766

$ct
             Flood_FALSE Flood_TRUE
Sediment_FALSE        22        12
Sediment_TRUE         25        10
```

```
# Estimate similarity measures between bifurcations and Grey scale P94
corr_Grey94<-make_ctable(floodinfo[,27]>0,floodinfo[,7]>0)
> corr_Grey94
$kulzinsky[1] 0.1034483
$correlation        [1] -0.09105444
$dice               [1] 0.08571429
$jaccardneedham     [1] 0.09375

$ct
             Flood_FALSE Flood_TRUE
Sediment_FALSE        37        19
Sediment_TRUE         10         3
```

```
# Estimate similarity measures between bifurcations and Grey scale P90
corr_Grey90<-make_ctable(floodinfo[,27]>0,floodinfo[,9]>0)
```

```
> corr_Grey90
$kulzinsky[1] 0.1764706
$correlation       [1] -0.107878
$dice              [1] 0.1304348
$jaccardneedham    [1] 0.15

$ct
              Flood_FALSE Flood_TRUE
Sediment_FALSE        29         16
Sediment_TRUE         18          6
```

```
# Estimate number of floods in x-years running windows

# installing package "zoo"
install.packages("zoo")


# Estimate sum for 3 years running window

n3floods<-rollapply(zoo(floodinfo[,27]),3,sum,align='left')

n3roc94<-rollapply(zoo(floodinfo[,3]),3,sum,align='left')
n3roc92<-rollapply(zoo(floodinfo[,13]),3,sum,align='left')
n3roc90<-rollapply(zoo(floodinfo[,5]),3,sum,align='left')
n3grey94<-rollapply(zoo(floodinfo[,7]),3,sum,align='left')
n3grey90<-rollapply(zoo(floodinfo[,9]),3,sum,align='left')


# Estimate correlation between Nfloods (known bifurcations) and the different thresholds

cor(n3floods,n3roc94)        # -0.05036244
cor(n3floods,n3roc92)        # 0.1996111
cor(n3floods,n3roc90)        # 0.244691
cor(n3floods,n3grey94)       # 0.1893286
cor(n3floods,n3grey90)       # 0.178834

# RoC 90 has the highest correlation (0.245)

# Estimate p-value for the best correlation
# Correct for number of independent observations (calculating effective number of independent
observations):

n3eff<-length(n3floods)/3                       # [1] 22.33333
# Estimate significance;
cor3roc90<-cor(n3floods,n3roc90)                # [1] 0.244691
# Calculate t-test
t3test<-cor3roc90 / sqrt((1- cor3roc90^2)/(n3eff-2))
# [1] 1.137966
pvaluecor3roc90<-1-pt(t3test,n3eff-2)           # [1] 0.1341832
```

Estimate sum for 4 years running window

```
n4floods<-rollapply(zoo(floodinfo[,27]),4,sum,align='left')
n4roc94<-rollapply(zoo(floodinfo[,3]),4,sum,align='left')
n4roc92<-rollapply(zoo(floodinfo[,13]),4,sum,align='left')
n4roc90<-rollapply(zoo(floodinfo[,5]),4,sum,align='left')
n4grey94<-rollapply(zoo(floodinfo[,7]),4,sum,align='left')
n4grey90<-rollapply(zoo(floodinfo[,9]),4,sum,align='left')
```

Estimate correlation between Nfloods (known bifurcations) and the different thresholds

```
cor(n4floods,n4roc94)       # 0.01643868
cor(n4floods,n4roc92)                 # 0.2819891
cor(n4floods,n4roc90)       # 0.3401985
cor(n4floods,n4grey94)      # 0.219318
cor(n4floods,n4grey90)      # 0.2137205
```

the threshold with highest correlation: RoC90 (0.34)

Estimate p-value for the best correlation
Correct for number of independent observations:

```
n4eff<-length(n4floods)/4                          # [1] 16.5
# Estimate sigificans;
cor4roc90<-cor(n4floods,n4roc90)                   # [1] 0.3401985
t4test<-cor4roc90/ sqrt((1- cor4roc90^2)/(n4eff-2))
# [1] 1.377606
pvaluecor4roc90<-1-pt(t4test,n4eff-2)              # [1] 0.09460858
```

Estimate sum for 5 years running window

```
n5floods<-rollapply(zoo(floodinfo[,27]),5,sum,align='left')
n5roc94<-rollapply(zoo(floodinfo[,3]),5,sum,align='left')
n5roc92<-rollapply(zoo(floodinfo[,13]),5,sum,align='left')
n5roc90<-rollapply(zoo(floodinfo[,5]),5,sum,align='left')
n5grey94<-rollapply(zoo(floodinfo[,7]),5,sum,align='left')
n5grey90<-rollapply(zoo(floodinfo[,9]),5,sum,align='left')
```

Estimate correlation between Nfloods (known bifurcations) and the different thresholds

```
cor(n5floods,n5roc94)       # 0.08037397
cor(n5floods,n5roc92)       # 0.3384541
cor(n5floods,n5roc90)       # 0.4138701
cor(n5floods,n5grey94)      # 0.2183799
cor(n5floods,n5grey90)      # 0.20408
```

The threshold with the highest correlation: RoC 90 (0.4138701)

Estimate p-value for the best correlation
Correct for number of independent observations:

```
n5eff<-length(n5floods)/5                    # [1] 13
# Estimate sigificans;
cor5roc90<-cor(n5floods,n5roc90)             # [1] 0.4138701
t5test<-cor5roc90 / sqrt((1- cor5roc90^2)/(n5eff-2))
# [1] 1.507852
pvaluecor5roc90<-1-pt(t5test,n5eff-2)
# [1] 0.07988181


# Estimate sum for 10 years running window

n10floods<-rollapply(zoo(floodinfo[,27]),10,sum,align='left')
n10roc94<-rollapply(zoo(floodinfo[,3]),10,sum,align='left')
n10roc92<-rollapply(zoo(floodinfo[,13]),10,sum,align='left')
n10roc90<-rollapply(zoo(floodinfo[,5]),10,sum,align='left')
n10grey94<-rollapply(zoo(floodinfo[,7]),10,sum,align='left')
n10grey90<-rollapply(zoo(floodinfo[,9]),10,sum,align='left')


# Estimate correlation between Nfloods (known bifurcations) and the different thresholds

cor(n10floods,n10roc94)      # 0.06362349
cor(n10floods,n10roc92)      # 0.3697143
cor(n10floods,n10roc90)      # 0.4430773
cor(n10floods,n10grey94)     # -0.0005549942
cor(n10floods,n10grey90)     # -0.0171576

# Threshold with highest correlation: RoC 90 (0.4430773)


# Estimate p-value for the best correlation

# Correct for number of independent observations:
n10eff<-length(n10floods)/10                 # [1] 6
# Estimate sigificans;
cor10roc90<-cor(n10floods,n10roc90)               # [1] 0.4430773
t10test<-cor10roc90 / sqrt((1- cor10roc90^2)/(n10eff-2))
# [1] 0.9884784
pvaluecor10roc90 <-1-pt(t10test,n10eff-2)
# [1] 0.189438
```

Return levels for 2.604 Elverum

Period before regulations (1871-1936)

```
# set work directory
setwd("C:/Users/Anna/Desktop/R. Elverum og Nors Bru/regulationsElverum")

# import file elverum1
myfile_elverum1 <-"C:/Users/Anna/Desktop/R. Elverum og Nors
Bru/regulationsElverum/elverum1.txt"
elverum1 <- read.table("C:/Users/Anna/Desktop/R. Elverum og Nors
Bru/regulationsElverum/elverum1.txt", quote="\"")

# read file, but excludes the first 6 lines
dailydat_elverum1 <- read.table(myfile_elverum1,sep=" ",skip=6)
colnames(dailydat_elverum1) <- c("orig_date", "vf")

# estimate the date
dailydat_elverum1$date <- as.Date(dailydat_elverum1$orig_date, format = "%Y%m%d")

# estimate which year each day belong to
dailydat_elverum1$year <-  as.numeric(format(as.Date(dailydat_elverum1$date), "%Y"))

# set "-9999" to NA
dailydat_elverum1[dailydat_elverum1 == -9999] <- NA

# estimate for which years there exist data
daily_years_elverum1 <- as.numeric(na.omit(unique(dailydat_elverum1$year)))

sum_nona<-function(xx){
  sum(!is.na(xx))
  }

# estimate number of observations each year
nna_elverum1 <-
(as.numeric(unlist(by(dailydat_elverum1[,2],dailydat_elverum1[,4],sum_nona,simplify=TRUE))))

# Use only years with more than 0 data
length_OK <- nna_elverum1> 0

# take out the number of values for these years
nna_elverum1<-nna_elverum1[length_OK]

# take out the year for the years with more than 0 data
daily_years_elverum1 <-daily_years_elverum1[length_OK]

# Take out daily data only for years with more than 0 data
dailydat_elverum1 <-dailydat_elverum1[is.element(dailydat_elverum1$year,daily_years_elverum1),]

# take out the daily maximum data only for years with more than 0 data
daily_ams_elverum1 <-
as.numeric(unlist(by(dailydat_elverum1[,2],dailydat_elverum1[,4],max,na.rm=TRUE)))
get_max_date_elverum1<-function(myind){
```

```
   dat_temp<-dailydat_elverum1[dailydat_elverum1$year==daily_years_elverum1[myind],]
   mdate<-dat_temp[head(which(dat_temp[,2]==daily_ams_elverum1[myind]),1),3]
   mdate
  }

# nr of years with data:
nmax<-length(daily_ams_elverum1)       # nmax = 66 (years with data)

# estimate date for yearly maxima values
daily_ams_dates_elverum1 <-as.Date(sapply(c(1:nmax),get_max_date_elverum1),origin="1970-01-
01")

# put all the info in a data frame
ddd_elverum1 <-data.frame(daily_years_elverum1,daily_ams_dates_elverum1,daily_ams_elverum1)


# Bayesian MCMC
# install package nsRFA
install.packages("nsRFA")

# do a MCMC-adaptation using only the ams-serie from the station
myprior_elverum1 <- function (x) {
     # x = vector of parameter values: c(location, scale, shape)
     # I assume the shape parameter only has a prior with mean zero and standard deviation 0.2
     dnorm(x[3], 0, 0.2)
   }

 # Bayesian MCMC-adaptation
Myfit_elverum1<-BayesianMCMC (xcont=daily_ams_elverum1, confint = c(0.05, 0.95), dist = "GEV",
apriori = myprior_elverum1)


# Adding information from historical events
# seuil = threshold (m^3/s) that is not exceeded in historical times, except those with historical
information. The threshold chosen is 2533 m^3/s, which is the estimated flood size of 1967-flood.
# nbans = the period (before systematic measurements started) containing all relevant historical
information we have (ca 1650-1850).
# infhist = vector containing the historical flood events, and the lowest possible values for these.
# nfloods = number of floods that have exceeded seuil during nbans. 9 years had floods exceeding the
1967-flood-threshold; 1789, 1675, 1773, 1717, 1724, 1749, 1850, 1827, 1846. The latter flood are
marked with the same elevation as the 1967-flood, and is included.

seuil = 2533
t = seuil
nbans = 200
hperiod = nbans
nfloods = 9
infhist = rep(t,nfloods)

# do the extreme value analysis one more time, with this additional (historical) information
Myfit_hist_elverum1 <-BayesianMCMC (xcont=daily_ams_elverum1, confint = c(0.05, 0.95), dist =
"GEV", apriori = myprior_elverum1,nbans=hperiod,seuil=t,infhist=rep(t,nfloods))
```

# Adding estimated discharge from Elverum
infhist_elverum1 = rep(t,nfloods)

threshold flood (1967-flood) was exceeded the following years;
(1789,1675,1773,1717,1724,1749,1850,1827,1846)

# the discharge these 9 times where exceeded (the same order as the years in the vector above);
xhist = c(3900,3090,3090,3034,3034,3034,2963,2794,2533)

Myfit_est_elverum1 <-BayesianMCMC (xcont=daily_ams_elverum1, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum1,nbans=hperiod,seuil=t, xhist = xhist)

Adding floods from paleohydrologic information
Use all floods that occurred between 1200 and 1870, e.i. 670 years (nbans = 670). 155 floods happened in this period (from flood counts in long core, FLP213).

Lowest threshold at Elverum: 1300 (infhist) m3/s (based on the regression line). Highest threshold: 3900 (suphist) m3/s.

Myfit_paleo_elverum1 <-BayesianMCMC (xcont=daily_ams_elverum1, confint = c(0.05, 0.95), dist = "G EV", apriori = myprior_elverum1,nbans=670,seuil=1300,infhist=rep(1300,155),suphist=rep(3900,155), nbchaines=3)

# Adding flood information from systematic + historical + paleohydrological sources
# Make a vector containing the lower threshold. Adds a small uncertainty interval around the historical floods.
infhist2<-c(rep(1300,107),c(3900,3090,3090,3034,3034,3034,2963,2794,2533)*0.95)
suphist2<-c(rep(3900,107),c(3900,3090,3090,3034,3034,3034,2963,2794,2533)*1.05)

# Number of years the floods are representable for. The vector need the same length as in infhist2 and suphist2. Number of years in the first row.
nbans2<-c(470,rep(0,106),200,rep(0,8))

# The thresholds must also be included in a vector:
seuil2<-c(rep(1300,107),rep(2500,9))

Myfit_paleo_hist_elverum1 <-BayesianMCMC (xcont=daily_ams_elverum1, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum1,nbans=nbans2,seuil=seuil2,infhist=infhist2,suphist=suphist2,nbchaines=3)

Return level plot
1) Systematic data: Myfit_elverum1
2) Historical information: Myfit_hist_elverum1
3) Estimated discharge: Myfit_est_elverum1
4) Paleofloods: Myfit_paleo_elverum1
5) Syst + hist + paleo: Myfit_paleo_hist_elverum1

```
# Making return level graph with info (1), (2) and (3):
# plot estimated discharge values

plot(Myfit_est_elverum1,ylab="Discharge (m3/s)",main="Return levels at 2.604 Elverum (before regulations)", xlab="Return period T (years) ",ylim=c(0,5000))

# adding systematic data – return period
lines(Myfit_elverum1$returnperiods,Myfit_elverum1$intervals[1,],lty="dashed",col=1)
lines(Myfit_elverum1$returnperiods,Myfit_elverum1$quantilesML,col=1)
lines(Myfit_elverum1$returnperiods,Myfit_elverum1$intervals[2,],lty="dashed",col=1)

# adding estimated data – return period
lines(Myfit_est_elverum1$returnperiods,Myfit_est_elverum1$intervals[1,],lty="dashed",col=2)
lines(Myfit_est_elverum1$returnperiods,Myfit_est_elverum1$quantilesML,col=2)
lines(Myfit_est_elverum1$returnperiods,Myfit_est_elverum1$intervals[2,],lty="dashed",col=2)

# adding historical data – return period
lines(Myfit_hist_elverum1$returnperiods,Myfit_hist_elverum1$intervals[1,],lty="dashed",col=3)
lines(Myfit_hist_elverum1$returnperiods,Myfit_hist_elverum1$quantilesML,col=3)
lines(Myfit_hist_elverum1$returnperiods,Myfit_hist_elverum1$intervals[2,],lty="dashed",col=3)

legend("bottomright",col=c(1,1,3,3,2,2),lty=c(1,2,1,2,1,2),legend=c("Systematic data","90% CI","Hist. data (threhsold)","90% CI","Hist. data (est. discharge)","90% CI"))



# Making return-level graph with info (1), (4) and (5)
plot(Myfit_elverum1,ylab="Discharge (m3/s)",main="Return levels at 2.604 Elverum (before regulations)", xlab="Return period T (years) ",ylim=c(0,5000))
# adding paleoflood information – return period
lines(Myfit_paleo_elverum1$returnperiods,Myfit_paleo_elverum1$intervals[2,],type="l",lty="dashed",col=4)
lines(Myfit_paleo_elverum1$returnperiods,Myfit_paleo_elverum1$intervals[1,],lty="dashed",col=4)
lines(Myfit_paleo_elverum1$returnperiods,Myfit_paleo_elverum1$quantilesML,col=4)

# adding syst+hist+paleo flood information – return period
lines(Myfit_paleo_hist_elverum1$returnperiods,Myfit_paleo_hist_elverum1$intervals[2,],type="l",lty="dashed",col=5)
lines(Myfit_paleo_hist_elverum1$returnperiods,Myfit_paleo_hist_elverum1$intervals[1,],lty="dashed",col=5)
lines(Myfit_paleo_hist_elverum1$returnperiods,Myfit_paleo_hist_elverum1$quantilesML,col=5)

legend("bottomright",col=c(1,1,4,4,5,5),lty=c(1,2,1,2,1,2),legend=c("Systematic data","90% CI","Paleofloods","90% CI","Syst + hist + paleofloods","90% CI"))

# Making graph with all info together:
# plot estimated discharge values
plot(Myfit_elverum1,ylab="Discharge (m3/s)",main="Return levels at 2.604 Elverum (before regulations)", xlab="Return period T (years) ",ylim=c(0,5000))


# adding estimated data – return period
```

```
lines(Myfit_est_elverum1$returnperiods,Myfit_est_elverum1$quantilesML,col=2)

# adding historical data – return period
lines(Myfit_hist_elverum1$returnperiods,Myfit_hist_elverum1$quantilesML,col=3)

# adding paleoflood information – return period
lines(Myfit_paleo_elverum1$returnperiods,Myfit_paleo_elverum1$quantilesML,col=4)

# adding syst+hist+paleo flood information – return period
lines(Myfit_paleo_hist_elverum1$returnperiods,Myfit_paleo_hist_elverum1$quantilesML,col=5)

# adding legend
legend("bottomright",col=c(1,1,3,2,4,5),lty=c(1,2,1,1,1,1),legend=c("Systematic data","90% CI","Hist. data (threshold)","Hist. data (est. discharge)","Paleofloods","Syst + hist + paleofloods"))
```

Sensitivity test for the estimation – using Storofsen

Since Storofsen (1789) is an extreme case, I want to test how sensitive the estimation with this flood. How much does Storofsen influence the flood estimation? Storofsen was characterized as a 1000-year flood, but there are reasons to believe that this flood was actually even rarer. From Holocene sediment records, there is no flood layers that can compares with Storofsen, so Storofsen might be a 5000- year flood, or even a 10 000-year flood!

By changing the first number in “nbans3”, I can change the length of the period Storofsen represents.

```
# Storofsen – 100-year flood?
xhist3<-c(3900,3090,3090,3034,3034,3034,2963,2794,2533)
nbans3<-c(100,180,rep(0,8))
seuil3<-c(3890,rep(2500,9))
Myfit_storofsen100 <-BayesianMCMC (xcont=daily_ams_elverum1, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum1,nbans=nbans3,seuil=seuil3,xhist=xhist3,nbchaines=3)


# Storofsen – a 1000-year flood?
xhist4<-c(3900,3090,3090,3034,3034,3034,2963,2794,2533)
nbans4<-c(1000,180,rep(0,8))
seuil4<-c(3890,rep(2500,9))

Myfit_storofsen1000 <-BayesianMCMC (xcont=daily_ams_elverum1, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum1,nbans=nbans4,seuil=seuil4,xhist=xhist4,nbchaines=3)


# Storofsen – a 5000-year flood?
xhist5<-c(3900,3090,3090,3034,3034,3034,2963,2794,2533)
nbans5<-c(5000,180,rep(0,8))
seuil5<-c(3890,rep(2500,9))

Myfit_storofsen5000 <-BayesianMCMC (xcont=daily_ams_elverum1, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum1,nbans=nbans5,seuil=seuil5,xhist=xhist5,nbchaines=3)
# Storofsen – a 10 000-year flood?
```

```
xhist6<-c(3900,3090,3090,3034,3034,3034,2963,2794,2533)
nbans6<-c(10000,180,rep(0,8))
seuil6<-c(3890,rep(2500,9))

Myfit_storofsen10000 <-BayesianMCMC (xcont=daily_ams_elverum1, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum1,nbans=nbans6,seuil=seuil6,xhist=xhist6,nbchaines=3)


# Excluding Storofsen from the estimations, to see how this influence the estimations;

xhist7<-c(3900,3090,3090,3034,3034,3034,2963,2794,2533)
nbans7<-180
seuil7<-2500
Myfit_excluded <-BayesianMCMC (xcont=daily_ams_elverum1, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum1,nbans=nbans7,seuil=seuil7,xhist=xhist7,nbchaines=3)


# Making return-level graph
# plot Storofsen as 100-yr flood

plot(Myfit_storofsen100,ylab="Discharge (m3/s)",main="Sensitivity test - Storofsen", xlab="Return period T (years) ",ylim=c(0,5000))

# adding Storofsen as 1000-yr flood
lines(Myfit_storofsen1000$returnperiods,Myfit_storofsen1000$quantilesML,col=2)


# adding Storofsen as 5000-yr flood
lines(Myfit_storofsen5000$returnperiods,Myfit_storofsen5000$quantilesML,col=3)


# adding Storofsen as 10000-yr flood
lines(Myfit_storofsen10000$returnperiods,Myfit_storofsen10000$quantilesML,col=4)


# adding excluded Storofsen
lines(Myfit_excluded$returnperiods,Myfit_excluded$quantilesML,col=5)


legend("bottomright",col=c(1,1,2,3,4,5),lty=c(1,2,1,1,1,1),legend=c("100-yr flood","90% CI","1000-yr flood","5000-yr flood","10000-yr flood","excluded"))
```

Period after regulations (1937-2015)

```
# set work directoy
setwd("C:/Users/Anna/Desktop/R. Elverum og Nors Bru/regulationsElverum")

# import file elverum2
myfile_elverum2 <-"C:/Users/Anna/Desktop/R. Elverum og Nors Bru/regulationsElverum/elverum2.txt"
elverum2 <- read.table("C:/Users/Anna/Desktop/R. Elverum og Nors Bru/regulationsElverum/elverum2.txt", quote="\"")
```

```
# read file, but excludes the first 6 lines
dailydat_elverum2 <- read.table(myfile_elverum2,sep=" ",skip=6)
colnames(dailydat_elverum2) <- c("orig_date", "vf")

# estimate the date
dailydat_elverum2$date <- as.Date(dailydat_elverum2$orig_date, format = "%Y%m%d")

# estimate which year each day belong to
dailydat_elverum2$year <-  as.numeric(format(as.Date(dailydat_elverum2$date), "%Y"))

# set "-9999" to NA
dailydat_elverum2[dailydat_elverum2 == -9999] <- NA

# estimate for which years there exist data
daily_years_elverum2 <- as.numeric(na.omit(unique(dailydat_elverum2$year)))

sum_nona<-function(xx){
  sum(!is.na(xx))
  }

# estimate number of observations each year
nna_elverum2 <-
(as.numeric(unlist(by(dailydat_elverum2[,2],dailydat_elverum2[,4],sum_nona,simplify=TRUE))))

# use only years with more than 0 data
length_OK <- nna_elverum2> 0

# take out the number of values for these years
nna_elverum2<-nna_elverum2[length_OK]

# take out the year for the years with more than 0 data
daily_years_elverum2 <-daily_years_elverum2[length_OK]

# take out daily data only for years with more than 0 data
dailydat_elverum2 <-dailydat_elverum2[is.element(dailydat_elverum2$year,daily_years_elverum2),]

# take out the daily maximum data only for years with more than 0 data
daily_ams_elverum2 <-
as.numeric(unlist(by(dailydat_elverum2[,2],dailydat_elverum2[,4],max,na.rm=TRUE)))

get_max_date_elverum2<-function(myind){
   dat_temp<-dailydat_elverum2[dailydat_elverum2$year==daily_years_elverum2[myind],]
   mdate<-dat_temp[head(which(dat_temp[,2]==daily_ams_elverum2[myind]),1),3]
   mdate
  }

# nr of years with data:
nmax<-length(daily_ams_elverum2)      # nmax = 79 (years with data)

# estimate date for yearly maxima values
daily_ams_dates_elverum2 <-as.Date(sapply(c(1:nmax),get_max_date_elverum2),origin="1970-01-
01")
```

```
# put all the info in a data frame
ddd_elverum2 <-data.frame(daily_years_elverum2,daily_ams_dates_elverum2,daily_ams_elverum2)

# Bayesian MCMC
# install package nsRFA
install.packages("nsRFA")

# do a MCMC-adaptation using only the ams-serie from the station
myprior_elverum2 <- function (x) {
     # x = vector of parameter values: c(location, scale, shape)
     # I assume the shape parameter only has a prior with mean zero and standard deviation 0.2
     dnorm(x[3], 0, 0.2)
   }

Myfit_elverum2<-BayesianMCMC (xcont=daily_ams_elverum2, confint = c(0.05, 0.95), dist = "GEV",
apriori = myprior_elverum2)


# Adding information from historical events
# seuil = threshold (m3/s) that is not exceeded in historical times, except those with historical
information. The threshold chosen is 2533 m3/s, which is the estimated flood size of 1967-flood.
# nbans = the period (before systematic measurements started) containing all relevant historical
information we have (ca 1650-1850).
# infhist = vector containing the historical flood events, and the lowest possible values for these.
# nfloods = number of floods that have exceeded seuil during nbans. 9 years had floods exceeding the
1967-flood-threshold; 1789, 1675, 1773, 1717, 1724, 1749, 1850, 1827, 1846. The latter flood are
marked with the same elevation as the 1967-flood, and is included.

seuil = 2533
t = seuil
nbans = 200
hperiod = nbans
nfloods = 9
infhist = rep(t,nfloods)


# do the extreme value analysis one more time, with this additional (historical) information
Myfit_hist_elverum2 <-BayesianMCMC (xcont=daily_ams_elverum2, confint = c(0.05, 0.95), dist =
"GEV", apriori = myprior_elverum2,nbans=hperiod,seuil=t,infhist=rep(t,nfloods))

# Comparing results
# graph showing return period on x-axis and discharge on y-axis (with- and without historical
information).

ymax=max(Myfit_elverum2$intervals[2,1:31],Myfit_elverum2$intervals[2,1:31],Myfit_elverum2$inter
vals[2,1:31])
plot(Myfit_elverum2$returnperiods,Myfit_elverum2$intervals[2,],type="l",lty="dashed",log="x",xlim=c
(1,1100),ylim=c(0,ymax),main="Return period with and without historical information",xlab="Return
period (years)",ylab="Discharge (m3/s)")
lines(Myfit_elverum2$returnperiods,Myfit_elverum2$intervals[1,],lty="dashed")
lines(Myfit_elverum2$returnperiods,Myfit_elverum2$quantilesML)
```

lines(Myfit_hist_elverum2$returnperiods,
Myfit_hist_elverum2$intervals[2,],type="l",lty="dashed",col=2)
lines(Myfit_hist_elverum2$returnperiods, Myfit_hist_elverum2 $intervals[1,],lty="dashed",col=2)
lines(Myfit_hist_elverum2 $returnperiods, Myfit_hist_elverum2 $quantilesML,col=2)
legend("bottomright",col=c(1,1,2,2),lty=c(1,2,1,2),legend=c("Without historical data","90% CI","With historical data","90% CI"))

Adding estimated discharges from Nor

infhist_elverum2 = rep(t,nfloods)

threshold flood (1967-flood) was exceeded the following years;
(1789,1675,1773,1717,1724,1749,1850,1827,1846)

# the discharge these 9 times where exceeded (the same order as the years in the vector above);
xhist = c(3900,3090,3090,3034,3034,3034,2963,2794,2533)

Myfit_est_elverum2 <-BayesianMCMC (xcont=daily_ams_elverum2, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum2,nbans=hperiod,seuil=t, xhist = xhist)

Adding floods from paleohydrologic information
Use all floods that occurred between 1200 and 1870, e.i. 670 years (nbans = 670). 155 floods happened in this period (from flood counts in long core, FLP213).

Lowest threshold at Elverum: 1300 (infhist) m3/s (based on the regression line). Highest threshold: 3900 (suphist) m3/s.

Myfit_paleo_elverum2 <-BayesianMCMC (xcont=daily_ams_elverum2, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum2,nbans=670,seuil=1300,infhist=rep(1300,155),suphist=rep(3900,155), nbchaines=3)

Adding flood information from syst + hist + paleo
Include systematic data, historical information AND paleohydrological information in the estimation;

# Make a vector containing the lower threshold. Adds a small uncertainty interval around the historical floods.
infhist2<-c(rep(1300,107),c(3900,3090,3090,3034,3034,3034,2963,2794,2533)*0.95)
suphist2<-c(rep(3900,107),c(3900,3090,3090,3034,3034,3034,2963,2794,2533)*1.05)

# Number of years the floods are representable for. The vector need the same length as in infhist2 and suphist2. Number of years in the first row.
nbans2<-c(470,rep(0,106),200,rep(0,8))

# The thresholds must also be included in a vector:
seuil2<-c(rep(1300,107),rep(2500,9))

Myfit_paleo_hist_elverum2 <-BayesianMCMC (xcont=daily_ams_elverum2, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum2,nbans=nbans2,seuil=seuil2,infhist=infhist2,suphist=suphist2,nbchaines=3)

```
# Return level plot
# 1) Systematic data:           Myfit_elverum2
# 2) Historical information:    Myfit_hist_elverum2
# 3) Estimated discharge:       Myfit_est_elverum2
# 4) Paleofloods:               Myfit_paleo_elverum2
# 5) Syst + hist + paleo:       Myfit_paleo_hist_elverum2
```

```
# Making return-level graph
# plot estimated discharge values
plot(Myfit_elverum2,ylab="Discharge (m3/s)",main="Return levels at 2.604 Elverum (after regulations)
", xlab="Return period T (years) ",ylim=c(0,5000))

lines(Myfit_est_elverum2$returnperiods,Myfit_est_elverum2$quantilesML,col=2)
lines(Myfit_hist_elverum2$returnperiods,Myfit_hist_elverum2$quantilesML,col=3)
lines(Myfit_paleo_elverum2$returnperiods,Myfit_paleo_elverum2$quantilesML,col=4)
lines(Myfit_paleo_hist_elverum2$returnperiods,Myfit_paleo_hist_elverum2$quantilesML,col=5)
legend("bottomright",col=c(1,1,3,2,4,5),lty=c(1,2,1,1,1,1),legend=c("Systematic data","90% CI","Hist. d
ata (threshold)","Hist.data (est.discharge)","Paleofloods","Syst + hist + paleofloods"))
```

Whole period (1871-2015)

```
# set work directory
setwd("C:/Users/Anna/Desktop/R. Elverum og Nors Bru/regulationsElverum")

# import file Elverum
myfile_elverum <-"C:/Users/Anna/Desktop/R. Elverum og Nors Bru/regulationsElverum/elverum-
helePerioden.txt"

elverum <- read.table("C:/Users/Anna/Desktop/R. Elverum og Nors Bru/regulationsElverum/elverum-
helePerioden.txt", quote="\"")

# read file, but excludes the first 6 lines
dailydat_elverum <- read.table(myfile_elverum,sep=" ",skip=6)
colnames(dailydat_elverum) <- c("orig_date", "vf")

# estimate the date
dailydat_elverum$date <- as.Date(dailydat_elverum$orig_date, format = "%Y%m%d")

# estimate which year each day belong to
dailydat_elverum$year <-  as.numeric(format(as.Date(dailydat_elverum$date), "%Y"))

# set "-9999" to NA
dailydat_elverum[dailydat_elverum == -9999] <- NA

# estimate for which years there exist data
daily_years_elverum <- as.numeric(na.omit(unique(dailydat_elverum$year)))

sum_nona<-function(xx){

  sum(!is.na(xx))
```

```
  }
# estimate number of observations each year
nna_elverum <-
(as.numeric(unlist(by(dailydat_elverum[,2],dailydat_elverum[,4],sum_nona,simplify=TRUE))))

# use only years with more than 0 data
length_OK <- nna_elverum> 0

# take out the number of values for these years
nna_elverum<-nna_elverum[length_OK]

# take out the year for the years with more than 0 data
daily_years_elverum <-daily_years_elverum[length_OK]

# Take out daily data only for years with more than 0 data
dailydat_elverum <-dailydat_elverum[is.element(dailydat_elverum$year,daily_years_elverum),]

# take out the daily maximum data only for years with more than 0 data
daily_ams_elverum <-
as.numeric(unlist(by(dailydat_elverum[,2],dailydat_elverum[,4],max,na.rm=TRUE)))

get_max_date_elverum<-function(myind){

   dat_temp<-dailydat_elverum[dailydat_elverum$year==daily_years_elverum[myind],]

   mdate<-dat_temp[head(which(dat_temp[,2]==daily_ams_elverum[myind]),1),3]

   mdate

  }

# nr of years with data
nmax<-length(daily_ams_elverum)                    # nmax = 145 = 145 years with data.

# estimate date for yearly maxima values
daily_ams_dates_elverum <-as.Date(sapply(c(1:nmax),get_max_date_elverum),origin="1970-01-01")

# put all the info in a data frame
ddd_elverum <-data.frame(daily_years_elverum,daily_ams_dates_elverum,daily_ams_elverum)


# Bayesian MCMC
# install package nsRFA
install.packages("nsRFA")

# then do a MCMC-adaptation using only the ams-serie from the station
myprior_elverum <- function (x) {

    # x = vector of parameter values: c(location, scale, shape)
    # I assume the shape parameter only has a prior with mean zero and standard deviation 0.2

    dnorm(x[3], 0, 0.2)

   }
```

Bayesian MCMC-method/analysis

Myfit_elverum<-BayesianMCMC (xcont=daily_ams_elverum, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum)

# plot the estimated flood sizes
ymax=max(Myfit_elverum$intervals[2,1:31],Myfit_elverum$intervals[2,1:31],Myfit_elverum$intervals[2,1:31])
plot(Myfit_elverum$returnperiods,Myfit_elverum$intervals[2,],type="l",lty="dashed",log="x",xlim=c(1,1100),ylim=c(0,ymax),xlab="Return period (years)",ylab="Discharge (m3/s)")
lines(Myfit_elverum$returnperiods,Myfit_elverum$intervals[1,],lty="dashed")
lines(Myfit_elverum$returnperiods,Myfit_elverum$quantilesML)

Adding information from historical events
seuil = threshold (m^3/s) that is not exceeded in historical times, except those with historical information. The threshold chosen is 2533 m^3/s, which is the estimated flood size of 1967-flood.

nbans = the period (before systematic measurements started) containing all relevant historical information we have (ca 1650-1850).

infhist = vector containing the historical flood events, and the lowest possible values for these.

nfloods = number of floods that have exceeded seuil during nbans. 9 years had floods exceeding the 1967-flood-threshold; 1789, 1675, 1773, 1717, 1724, 1749, 1850, 1827, 1846. The latter flood are marked with the same elevation as the 1967-flood, and is included.

seuil = 2533
t = seuil
nbans = 200
hperiod = nbans
nfloods = 9
infhist = rep(t,nfloods)

# do the extreme value analysis one more time, with this additional (historical) information
Myfit_hist_elverum <-BayesianMCMC (xcont=daily_ams_elverum, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum,nbans=hperiod,seuil=t,infhist=rep(t,nfloods))

# Adding estimated discharge values for the floods exceeding the threshold-flood
infhist_elverum = rep(t,nfloods)

threshold flood (1967-flood) was exceeded the following years;
(1789,1675,1773,1717,1724,1749,1850,1827,1846)

# the discharge these 9 times where exceeded (the same order as the years in the vector above);
xhist = c(3900,3090,3090,3034,3034,3034,2963,2794,2533)

Myfit_est_elverum <-BayesianMCMC (xcont=daily_ams_elverum, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum,nbans=hperiod,seuil=t, xhist = xhist)

Adding floods from paleohydrological information
Use all floods that occurred between 1200 and 1870, e.i. 670 years (nbans = 670). 155 floods happened in this period (from flood counts in long core, FLP213).

Lowest threshold at Elverum: 1300 (infhist) m3/s (based on the regression line). Highest threshold: 3900 (suphist) m3/s.

Myfit_paleo_elverum <-BayesianMCMC (xcont=daily_ams_elverum, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum,nbans=670,seuil=1300,infhist=rep(1300,155),suphist=rep(3900,155),nbchaines=3)

# Include systematic data, historical information AND paleohydrological information in the estimation;
# Make a vector containing the lower threshold. Adds a small uncertainty interval around the historical floods.
infhist2<-c(rep(1300,107),c(3900,3090,3090,3034,3034,3034,2963,2794,2533)*0.95)
suphist2<-c(rep(3900,107),c(3900,3090,3090,3034,3034,3034,2963,2794,2533)*1.05)

# Number of years the floods are representable for. The vector need the same length as in infhist2 and suphist2. Number of years in the first row.
nbans2<-c(470,rep(0,106),200,rep(0,8))

# The thresholds must also be included in a vector:
seuil2<-c(rep(1300,107),rep(2500,9))

Myfit_paleo_hist_elverum <-BayesianMCMC (xcont=daily_ams_elverum, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum,nbans=nbans2,seuil=seuil2,infhist=infhist2,suphist=suphist2,nbchaines=3)

Making return level plots

1) Systematic data Myfit_elverum
2) Historical information: Myfit_hist_elverum
3) Estimated discharge: Myfit_est_elverum
4) Paleofloods: Myfit_paleo_elverum
5) Syst + hist + paleo: Myfit_paleo_hist_elverum

Making return level graph

plot(Myfit_elverum,ylab="Discharge (m3/s)",main="Return levels at 2.604 Elverum", xlab="Return period T (years) ",ylim=c(0,5000))
lines(Myfit_est_elverum$returnperiods,Myfit_est_elverum$quantilesML,col=2)
lines(Myfit_hist_elverum$returnperiods,Myfit_hist_elverum$quantilesML,col=3)
lines(Myfit_paleo_elverum$returnperiods,Myfit_paleo_elverum$quantilesML,col=4)
lines(Myfit_paleo_hist_elverum$returnperiods,Myfit_paleo_hist_elverum$quantilesML,col=5)

```
legend("bottomright",col=c(1,1,3,2,4,5),lty=c(1,2,1,1,1,1),legend=c("Systematic data","90 % CI","Hist. data (threshold)","Hist. data (est. discharge)","Paleofloods","Syst + hist + paleofloods"))
```

Elverum – the last 60 years (1956-2015)

```
# set work directory
setwd("C:/Users/Anna/Desktop/R. Elverum og Nors Bru/regulationsElverum")


# import file elverum
myfile_elverum60 <-"C:/Users/Anna/Desktop/R. Elverum og Nors Bru/regulationsElverum/elverum60yr.txt"

elverum60 <- read.table("C:/Users/Anna/Desktop/R. Elverum og Nors Bru/regulationsElverum/elverum60yr.txt", quote="\"")

# read file, but excludes the first 6 lines
dailydat_elverum60 <- read.table(myfile_elverum60,sep=" ",skip=6)
colnames(dailydat_elverum60) <- c("orig_date", "vf")

# estimate the date
dailydat_elverum60$date <- as.Date(dailydat_elverum60$orig_date, format = "%Y%m%d")

# estimate which year each day belong to
dailydat_elverum60$year <-  as.numeric(format(as.Date(dailydat_elverum60$date), "%Y"))

# set "-9999" to NA
dailydat_elverum60[dailydat_elverum60 == -9999] <- NA

# estimate for which years there exist data
daily_years_elverum60 <- as.numeric(na.omit(unique(dailydat_elverum60$year)))

sum_nona<-function(xx){

  sum(!is.na(xx))

  }

# estimate number of observations each year
nna_elverum60 <-
(as.numeric(unlist(by(dailydat_elverum60[,2],dailydat_elverum60[,4],sum_nona,simplify=TRUE))))

# use only years with more than 0 data
length_OK <- nna_elverum60> 0

# take out the number of values for these years
nna_elverum60<-nna_elverum60[length_OK]

# take out the year for the years with more than 0 data
daily_years_elverum60 <-daily_years_elverum60[length_OK]
```

```
# Take out daily data only for years with more than 0 data
dailydat_elverum60 <-
dailydat_elverum60[is.element(dailydat_elverum60$year,daily_years_elverum60),]

# take out the daily maximum data only for years with more than 0 data
daily_ams_elverum60 <-
as.numeric(unlist(by(dailydat_elverum60[,2],dailydat_elverum60[,4],max,na.rm=TRUE)))

get_max_date_elverum60<-function(myind){

   dat_temp<-dailydat_elverum60[dailydat_elverum60$year==daily_years_elverum60[myind],]

   mdate<-dat_temp[head(which(dat_temp[,2]==daily_ams_elverum60[myind]),1),3]

   mdate

  }

# nr of years with data
nmax<-length(daily_ams_elverum60)

# estimate date for yearly maxima values
daily_ams_dates_elverum60 <-as.Date(sapply(c(1:nmax),get_max_date_elverum60),origin="1970-01-
01")

# put all the info in a data frame
ddd_elverum60 <-
data.frame(daily_years_elverum60,daily_ams_dates_elverum60,daily_ams_elverum60)


# Bayesian MCMC
# install package nsRFA
install.packages("nsRFA")

# then do a MCMC-adaptation using only the ams-serie from the station
myprior_elverum60 <- function (x) {

    # x = vector of parameter values: c(location, scale, shape)
    # I assume the shape parameter only has a prior with mean zero and standard deviation 0.2

    dnorm(x[3], 0, 0.2)

   }


# Bayesian MCMC-method/analysis
Myfit_elverum60 <-BayesianMCMC (xcont=daily_ams_elverum60, confint = c(0.05, 0.95), dist = "GEV",
apriori = myprior_elverum60)


# Adding information from historical events
# seuil = threshold (m3/s) that is not exceeded in historical times, except those with historical
information. The threshold chosen is 2533 m3/s, which is the estimated flood size of 1967-flood.
```

nbans = the period (before systematic measurements started) containing all relevant historical information we have (ca 1650-1850).

infhist = vector containing the historical flood events, and the lowest possible values for these.

nfloods = number of floods that have exceeded seuil during nbans. 9 years had floods exceeding the 1967-flood-threshold; 1789, 1675, 1773, 1717, 1724, 1749, 1850, 1827, 1846. The latter flood are marked with the same elevation as the 1967-flood, and is included.

seuil = 2533
t = seuil
nbans = 200
hperiod = nbans
nfloods = 9
infhist = rep(t,nfloods)

# do the extreme value analysis one more time, with this additional (historical) information
Myfit_hist_elverum60 <-BayesianMCMC (xcont=daily_ams_elverum60, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum60,nbans=hperiod,seuil=t,infhist=rep(t,nfloods))

# Adding estimated discharge from Elverum
infhist_elverum60 = rep(t,nfloods)

threshold flood (1967-flood) was exceeded the following years; (1789,1675,1773,1717,1724,1749,1850,1827,1846)

# the discharge these 9 times where exceeded (the same order as the years in the vector above);
xhist = c(3900,3090,3090,3034,3034,3034,2963,2794,2533)

Myfit_est_elverum60 <-BayesianMCMC (xcont=daily_ams_elverum60, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum60,nbans=hperiod,seuil=t, xhist = xhist)

Adding floods from paleohydrologic information
Use all floods that occurred between 1200 and 1870, e.i. 670 years (nbans = 670). 155 floods happened in this period (from flood counts in long core, FLP213).

Lowest threshold at Elverum: 1300 (infhist) m3/s (based on the regression line). Highest threshold: 3900 (suphist) m3/s.

Myfit_paleo_elverum60 <-BayesianMCMC (xcont=daily_ams_elverum60, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum60,nbans=670,seuil=1300,infhist=rep(1300,155),suphist=rep(3900,155),nbchaines=3)

Adding flood information from systematic + historical + paleohydrological
Make a vector containing the lower threshold. Adds a small uncertainty interval around the historical floods.

infhist2<-c(rep(1300,107),c(3900,3090,3090,3034,3034,3034,2963,2794,2533)*0.95)
suphist2<-c(rep(3900,107),c(3900,3090,3090,3034,3034,3034,2963,2794,2533)*1.05)

Number of years the floods are representable for. The vector need the same length as in infhist2 and suphist2. Number of years in the first row.

nbans2<-c(470,rep(0,106),200,rep(0,8))

# The thresholds must also be included in a vector:
seuil2<-c(rep(1300,107),rep(2500,9))

Myfit_paleo_hist_elverum60 <-BayesianMCMC (xcont=daily_ams_elverum60, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum60,nbans=nbans2,seuil=seuil2,infhist=infhist2,suphist=suphist2,nbchaines=3)

Return level plot
1) Systematic data: Myfit_elverum60
2) Historical information: Myfit_hist_elverum60
3) Estimated discharge: Myfit_est_elverum60
4) Paleofloods: Myfit_paleo_elverum60
5) Syst + hist + paleo: Myfit_paleo_hist_elverum60

# Making return-level graph
# plot estimated discharge values
plot(Myfit_elverum60,ylab="Discharge (m3/s)",main="Return levels at 2.604 Elverum, 1956-2015", xlab="Return period T (years) ",ylim=c(0,5000))

# adding estimated data – return period
lines(Myfit_est_elverum60$returnperiods,Myfit_est_elverum60$quantilesML,col=2)

# adding historical data – return period
lines(Myfit_hist_elverum60$returnperiods,Myfit_hist_elverum60$quantilesML,col=3)

# adding paleoflood information – return period
lines(Myfit_paleo_elverum60$returnperiods,Myfit_paleo_elverum60$quantilesML,col=4)

# adding syst+hist+paleo flood information – return period
lines(Myfit_paleo_hist_elverum60$returnperiods,Myfit_paleo_hist_elverum60$quantilesML,col=5)

# adding legend
legend("bottomright",col=c(1,1,3,2,4,5),lty=c(1,2,1,1,1,1),legend=c("Systematic data","90% CI","Hist. data (threshold)","Hist. data (est. discharge)","Paleofloods","Syst + hist + paleofloods"))

Elverum – the last 30 years (1986-2015)

# set work directory
setwd("C:/Users/Anna/Desktop/R. Elverum og Nors Bru/regulationsElverum")

```
# import file elverum
myfile_elverum30 <-"C:/Users/Anna/Desktop/R. Elverum og Nors Bru/regulationsElverum/elverum30yr.txt"

elverum30 <- read.table("C:/Users/Anna/Desktop/R. Elverum og Nors Bru/regulationsElverum/elverum30yr.txt", quote="\"")

# read file, but excludes the first 6 lines
dailydat_elverum30 <- read.table(myfile_elverum30,sep=" ",skip=6)
colnames(dailydat_elverum30) <- c("orig_date", "vf")

# estimate the date
dailydat_elverum30$date <- as.Date(dailydat_elverum30$orig_date, format = "%Y%m%d")

# estimate which year each day belong to
dailydat_elverum30$year <-  as.numeric(format(as.Date(dailydat_elverum30$date), "%Y"))

# set "-9999" to NA
dailydat_elverum30[dailydat_elverum30 == -9999] <- NA

# estimate for which years there exist data
daily_years_elverum30 <- as.numeric(na.omit(unique(dailydat_elverum30$year)))

sum_nona<-function(xx){

  sum(!is.na(xx))

  }

# estimate number of observations each year
nna_elverum30 <-
(as.numeric(unlist(by(dailydat_elverum30[,2],dailydat_elverum30[,4],sum_nona,simplify=TRUE))))

# use only years with more than 0 data
length_OK <- nna_elverum30> 0

# take out the number of values for these years
nna_elverum30<-nna_elverum30[length_OK]

# take out the year for the years with more than 0 data
daily_years_elverum30 <-daily_years_elverum30[length_OK]

# Take out daily data only for years with more than 0 data
dailydat_elverum30 <-
dailydat_elverum30[is.element(dailydat_elverum30$year,daily_years_elverum30),]

# take out the daily maximum data only for years with more than 0 data
daily_ams_elverum30 <-
as.numeric(unlist(by(dailydat_elverum30[,2],dailydat_elverum30[,4],max,na.rm=TRUE)))


get_max_date_elverum30<-function(myind){

   dat_temp<-dailydat_elverum30[dailydat_elverum30$year==daily_years_elverum30[myind],]
```

```
   mdate<-dat_temp[head(which(dat_temp[,2]==daily_ams_elverum30[myind]),1),3]

   mdate

  }

# nr of years with data
nmax<-length(daily_ams_elverum30)

# estimate date for yearly maxima values
daily_ams_dates_elverum30 <-as.Date(sapply(c(1:nmax),get_max_date_elverum30),origin="1970-01-
01")

# put all the info in a data frame
ddd_elverum30 <-
data.frame(daily_years_elverum30,daily_ams_dates_elverum30,daily_ams_elverum30)


# Bayesian MCMC
# install package nsRFA
install.packages("nsRFA")

# then do a MCMC-adaptation using only the ams-serie from the station

myprior_elverum30 <- function (x) {

    # x = vector of parameter values: c(location, scale, shape)
    # I assume the shape parameter only has a prior with mean zero and standard deviation 0.2

    dnorm(x[3], 0, 0.2)

   }


# Bayesian MCMC-method/analysis
Myfit_elverum30 <-BayesianMCMC (xcont=daily_ams_elverum30, confint = c(0.05, 0.95), dist = "GEV",
apriori = myprior_elverum30)


# Adding information from historical events
# seuil = threshold (m^3/s) that is not exceeded in historical times, except those with historical
information. The threshold chosen is 2533 m^3/s, which is the estimated flood size of 1967-flood.

# nbans = the period (before systematic measurements started) containing all relevant historical
information we have (ca 1650-1850).

# infhist = vector containing the historical flood events, and the lowest possible values for these.

# nfloods = number of floods that have exceeded seuil during nbans. 9 years had floods exceeding the
1967-flood-threshold; 1789, 1675, 1773, 1717, 1724, 1749, 1850, 1827, 1846. The latter flood are
marked with the same elevation as the 1967-flood, and is included.

seuil = 2533
t = seuil
```

nbans = 200
hperiod = nbans
nfloods = 9
infhist = rep(t,nfloods)

# do the extreme value analysis one more time, with this additional (historical) information
Myfit_hist_elverum30 <-BayesianMCMC (xcont=daily_ams_elverum30, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum30,nbans=hperiod,seuil=t,infhist=rep(t,nfloods))

# Adding estimated discharge from Elverum
infhist_elverum30 = rep(t,nfloods)

threshold flood (1967-flood) was exceeded the following years;
(1789,1675,1773,1717,1724,1749,1850,1827,1846)

# the discharge these 9 times where exceeded (the same order as the years in the vector above);
xhist = c(3900,3090,3090,3034,3034,3034,2963,2794,2533)

Myfit_est_elverum30 <-BayesianMCMC (xcont=daily_ams_elverum30, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum30,nbans=hperiod,seuil=t, xhist = xhist)

Adding floods from paleohydrologic information
Use all floods that occurred between 1200 and 1870, e.i. 670 years (nbans = 670). 155 floods happened in this period (from flood counts in long core, FLP213).

Lowest threshold at Elverum: 1300 (infhist) m3/s (based on the regression line). Highest threshold: 3900 (suphist) m3/s.

Myfit_paleo_elverum30 <-BayesianMCMC (xcont=daily_ams_elverum30, confint = c(0.05, 0.95), dist = "GEV", apriori = myprior_elverum30,nbans=670,seuil=1300,infhist=rep(1300,155),suphist=rep(3900,155),nbchaines=3)

Adding flood information from systematic + historical + paleohydrological
Make a vector containing the lower threshold. Adds a small uncertainty interval around the historical floods.

infhist2<-c(rep(1300,107),c(3900,3090,3090,3034,3034,3034,2963,2794,2533)*0.95)
suphist2<-c(rep(3900,107),c(3900,3090,3090,3034,3034,3034,2963,2794,2533)*1.05)

# Number of years the floods are representable for. The vector need the same length as in infhist2 and suphist2. Number of years in the first row.
nbans2<-c(470,rep(0,106),200,rep(0,8))

# The thresholds must also be included in a vector:
seuil2<-c(rep(1300,107),rep(2500,9))

```
Myfit_paleo_hist_elverum30 <-BayesianMCMC (xcont=daily_ams_elverum30, confint = c(0.05, 0.95),
dist = "GEV", apriori =
myprior_elverum30,nbans=nbans2,seuil=seuil2,infhist=infhist2,suphist=suphist2,nbchaines=3)


# Return level plot
# 1) Systematic data:          Myfit_elverum30
# 2) Historical information:   Myfit_hist_elverum30
# 3) Estimated discharge:      Myfit_est_elverum30
# 4) Paleofloods:              Myfit_paleo_elverum30
# 5) Syst + hist + paleo:      Myfit_paleo_hist_elverum30


# Making return-level graph
# plot estimated discharge values
plot(Myfit_elverum30,ylab="Discharge (m3/s)",main="Return levels at 2.604 Elverum, 1986-2015",
xlab="Return period T (years) ",ylim=c(0,5000))

 # adding estimated data – return period
lines(Myfit_est_elverum30$returnperiods,Myfit_est_elverum30$quantilesML,col=2)


# adding historical data – return period
lines(Myfit_hist_elverum30$returnperiods,Myfit_hist_elverum30$quantilesML,col=3)


 # adding paleoflood information – return period
lines(Myfit_paleo_elverum30$returnperiods,Myfit_paleo_elverum30$quantilesML,col=4)

# adding syst+hist+paleo flood information – return period
lines(Myfit_paleo_hist_elverum30$returnperiods,Myfit_paleo_hist_elverum30$quantilesML,col=5)

# add legend
legend("bottomright",col=c(1,1,3,2,4,5),lty=c(1,2,1,1,1,1),legend=c("Systematic data","90% CI","Hist.
data (threshold)","Hist. data (est. discharge)","Paleofloods","Syst + hist + paleofloods"))
```