

Flood Interval Analysis

EAo3o

April 6, 2025 ver. 0.32

This handout provides an overview of hydrology, focusing on the USGS streamgaging network and the analysis of flood intervals. It covers the basics of hydrology, the significance of stream flow hydrographs, and the concept of flood return intervals. The handout also includes practical examples and visualizations to illustrate these concepts.

Hydrology Basics

Hydrology is the scientific study of the movement, distribution, and quality of water on Earth. Its roots trace back to ancient civilizations such as Mesopotamia, Egypt, India, and China, where early societies managed water for agriculture, transportation, and flood control using observational knowledge and basic engineering like canals and embankments.

Hydrology began to formalize as a science in the 17th and 18th centuries with pioneers like Pierre Perrault and Edmund Halley, who quantified rainfall and streamflow. By the 19th century, the rise of civil engineering demanded more precise understanding of water systems for designing infrastructure. Today, hydrology is an interdisciplinary field integrating tools like GIS, remote sensing, and computer models to address modern challenges such as flood risk, water security, and climate change.

USGS Stations and Methods

The United States Geological Survey (USGS) plays a central role in monitoring water resources across the country. As part of its mission to provide impartial scientific information about the natural environment, the USGS operates one of the worldâŽs most extensive and long-standing networks of streamgages, which are used to monitor river and streamflow conditions in real time.

The USGS streamgaging network consists of over 10,000 active stations located throughout the United States, including Alaska, Hawaii, and U.S. territories. These stations collect continuous data on stream discharge, stage (water height), temperature, and sometimes sediment or water quality. Many stations transmit data via satellite or radio every 15 to 60 minutes, making real-time information publicly available through the USGS National Water Information System (NWIS).

This network supports a wide range of applications, including flood forecasting, drought monitoring, water resource management, ecological studies, infrastructure design, and emergency response. Federal, state, and local agencies, as well as private sector users and researchers, rely heavily on USGS data for planning and decision-making.

Stream Flow (Discharge) Hydrograph

A hydrograph displays how river discharge responds to a rainfall event. Its key components include:

- **Rising Limb:** The initial increase in discharge as rainfall runoff enters the river system.
- **Peak Discharge:** The maximum flow rate reached during the event, often corresponding to flood risk.
- **Lag Time:** The delay between peak rainfall and peak discharge, influenced by land use and basin characteristics.
- **Falling Limb:** The decline in flow as water drains away and returns to normal levels.
- **Base Flow:** The underlying groundwater-fed flow present before and after the storm.

Flood Return Interval

lood intervals, also known as return periods, estimate how often extreme flood events are likely to occur. These estimates are critical for evaluating flood risk and informing infrastructure design, emergency planning, and land-use decisions. By understanding flood frequency, communities can build safer systems and reduce vulnerability to damaging events.

Return periods are used in engineering and environmental planning. Civil engineers design infrastructure like bridges and dams based on specific flood intervals (e.g., 100-year floods). Urban planners and insurance programs such as FEMA's National Flood Insurance Program use them to guide development and define hazard zones. Scientists also apply flood interval data to study ecosystems and assess climate-related changes in flood behavior.

Methods for Estimating Flood Intervals

There are several methods used in hydrology to estimate flood intervals or return periods. These methods vary in complexity and



Figure 1: Gauge Height

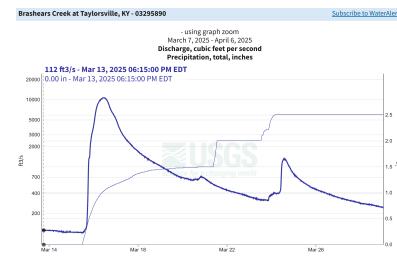


Figure 2: Discharge (cubic feet per second) at USGS streamgage 03295890, Brashears Creek at Taylorsville, KY. The hydrograph shows the response of streamflow to a rainfall event on April 5, 2025. The rising limb indicates increasing discharge as runoff enters the river system, while the peak discharge represents the maximum flow rate reached during the event. The falling limb shows the decline in flow as water drains away and returns to normal levels. Base flow is the underlying groundwater-fed flow present before and after the storm. The gauge height is also shown in Figure 1.

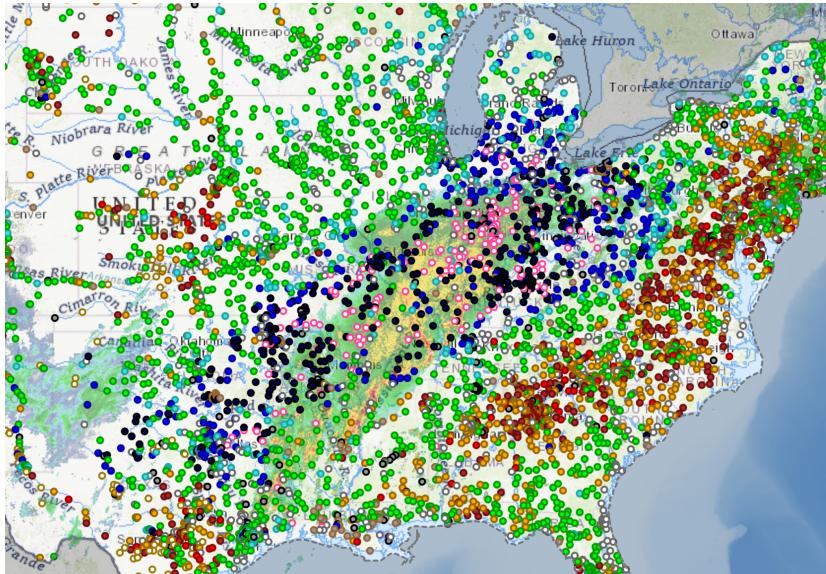


Figure 3: USGS National Water Dashboard – Realtime Streamflow, April 5, 2025. Back dots signify highest discharge on record as of access date. Open pink circles signify flood risks. Source: <https://dashboard.waterdata.usgs.gov/app/nwd/en/>.

applicability depending on the available data and purpose. Below are some of the most common approaches:

1. Empirical Method (Plotting Position Formula)

This is the simplest method and involves ranking annual peak discharges and applying a plotting position, where the return period T is defined as:

$$T = \frac{n+1}{m}$$

Where:

- n is the number of years of data
- m is the rank ($1 = \text{largest}$)

This method does not assume any underlying distribution.

```
# test implementaton in R --- TBA
```

2. Gumbel Distribution (Extreme Value Type I)

A commonly used theoretical distribution for modeling annual maxima is the Gumbel distribution. The return period is related to the probability of exceedance P as:

$$P = \frac{1}{T}$$

The Gumbel cumulative distribution function (CDF) is:

$$F(x) = \exp \left[-\exp \left(-\frac{x-\mu}{\beta} \right) \right]$$

Where μ is the location parameter and β is the scale parameter.

These can be estimated using method of moments or maximum likelihood.

3. Log-Pearson Type III Distribution

This method is widely recommended in the United States for flood frequency analysis. It applies a Pearson Type III distribution to the log-transformed discharge data. The procedure involves:

- Taking the logarithm of the discharge data
- Calculating the mean, standard deviation, and skewness
- Using these to estimate quantiles for different return periods

```
# Load necessary library
if (!require(e1071)) install.packages("e1071", dependencies = TRUE)

## Loading required package: e1071

library(e1071) # for skewness function

# Step 1: Input annual peak discharge data (in m^3/s)
discharge <- c(400, 520, 610, 700, 650, 720, 800, 750, 680, 670)

# Step 2: Log-transform the discharge data
logQ <- log10(discharge)

# Step 3: Compute statistics of log-values
mean_logQ <- mean(logQ)
sd_logQ <- sd(logQ)
skew_logQ <- skewness(logQ)

# Step 4: Define return periods and calculate exceedance probabilities
T <- c(2, 5, 10, 25, 50, 100) # Return periods
P <- 1 - 1 / T # Exceedance probability
z <- qnorm(P) # Standard normal variate

# Step 5: Apply Log-Pearson Type III quantile formula
K <- z + ( (z^2 - 1) * skew_logQ ) / 6 # Pearson III frequency factor

logQ_est <- mean_logQ + K * sd_logQ # Estimated logQ
```

```

Q_est <- 10^logQ_est                                # Convert back to original scale

# Combine results in a table
lp3_results <- data.frame(
  Return_Period = T,
  Exceedance_Prob = round(P, 4),
  Discharge_Estimate = round(Q_est, 2)
)

## % latex table generated in R 4.2.2 by xtable 1.8-4 package
## % Sun Apr  6 12:58:10 2025
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrr}
##   & Return\_Period & Exceedance\_Prob & Discharge\_Estimate \\
##   \hline
##   1 & 2.00 & 0.50 & 663.81 \\
##   2 & 5.00 & 0.80 & 766.28 \\
##   3 & 10.00 & 0.90 & 808.48 \\
##   4 & 25.00 & 0.96 & 842.30 \\
##   5 & 50.00 & 0.98 & 857.23 \\
##   6 & 100.00 & 0.99 & 865.72 \\
##   \hline
## \end{tabular}
## \end{table}

```

4. Bayesian Methods

Bayesian approaches allow incorporating prior information and uncertainty in parameter estimates. They are especially useful for short or regional datasets and are implemented via MCMC simulations.

5. Regional Frequency Analysis (RFA)

Instead of analyzing one site in isolation, RFA uses data from multiple similar sites in a region to improve flood frequency estimates. The Index Flood method is a popular RFA technique.

6. Generalized Extreme Value (GEV) Distribution

GEV unifies the Gumbel, Fréchet, and Weibull families under one formulation:

$$F(x) = \exp \left\{ - \left[1 + \xi \left(\frac{x - \mu}{\sigma} \right) \right]^{-1/\xi} \right\}$$

Where:

- μ is the location
- σ is the scale
- ξ is the shape parameter

These methods are often evaluated based on goodness-of-fit tests, diagnostic plots, and expert judgment.

Using R to Estimate Flood Return Intervals

Annual Peak Flow

```
# Eel River Max Discharge for USGS Record
library(fields)
# Use Stream_Hydrography_Analysis to create Eel.df

# Find Yearly Max flow...

Eel$year <- as.Date(Eel$date, "%Y")
Eel$maxq <- NA

aggregate(Eel$Discharge, list(Eel$year), max) -> tmp
names(tmp) <- c("year", "Discharge")
par(mar=c(2, 4.3, 1, 0) + 0.1, cex=1.8)
barplot(tmp$Discharge, names.arg=format(tmp$year, "%Y"), tcl= -0.3, xaxt="n", ylim=c(0,20000), ylab="", las=2)
mtext("Discharge (m³/s)", 2, line=3.5, cex=1.8)
mtext("Year", 1, line=0, cex=1.5)

arrow.plot(50, 15600, 90, 12000, arrow.ex=14, length=.1, col='red', lwd=3)
text(36, 15200, "1964", col='red', cex=1)
```

Basic Example

Below is an R code chunk that demonstrates how to calculate flood intervals using annual peak discharge data.

```
# Sample annual peak discharge data (in cubic meters per second)
discharge <- c(400, 520, 610, 700, 650, 720, 800, 750, 680, 670)
```

```
# Sort in descending order
sorted_discharge <- sort(discharge, decreasing = TRUE)

# Rank and return period
rank <- 1:length(sorted_discharge)
n <- length(discharge)
return_period <- (n + 1) / rank

# Create a data frame
flood_data <- data.frame(
  Rank = rank,
  Discharge = sorted_discharge,
  Return_Period = return_period
)
```

```
# Print table
print(flood_data)

##      Rank Discharge Return_Period
## 1      1     800    11.000000
## 2      2     750     5.500000
## 3      3     720     3.666667
## 4      4     700     2.750000
## 5      5     680     2.200000
## 6      6     670     1.833333
## 7      7     650     1.571429
## 8      8     610     1.375000
## 9      9     520     1.222222
## 10    10     400     1.100000
```

Flood Data Table

Below is the table of calculated return periods for each ranked discharge.

```
## % latex table generated in R 4.2.2 by xtable 1.8-4 package
## % Sun Apr  6 12:58:10 2025
## \begin{table}[ht]
## \centering
## \begin{tabular}{rrrr}
##   & Rank & Discharge & Return\_Period \\
## \hline
## 
```

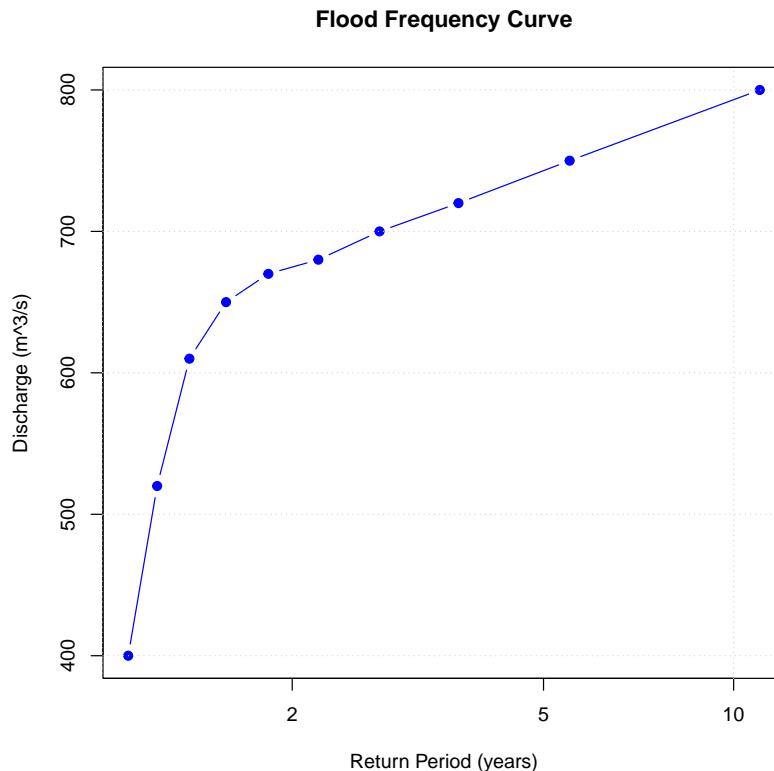
```

## 1 & 1 & 800.00 & 11.00 \\
## 2 & 2 & 750.00 & 5.50 \\
## 3 & 3 & 720.00 & 3.67 \\
## 4 & 4 & 700.00 & 2.75 \\
## 5 & 5 & 680.00 & 2.20 \\
## 6 & 6 & 670.00 & 1.83 \\
## 7 & 7 & 650.00 & 1.57 \\
## 8 & 8 & 610.00 & 1.38 \\
## 9 & 9 & 520.00 & 1.22 \\
## 10 & 10 & 400.00 & 1.10 \\
## \hline
## \end{tabular}
## \caption{Flood Return Period Table}
## \end{table}

```

Flood Frequency Plot

The following plot shows the discharge against its return period on a semi-logarithmic scale.



Using USGS Data

Load Required Packages

```
### STEP 1

### STEP 2
### Loading two specific packages into RWater
library(dataRetrieval)
library(xts)

## Loading required package: zoo
##
## Attaching package: 'zoo'
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

Load Data

```
### STEP 3
### Get the Peak Annual Discharge
mysite <- "07061000"
startDate = ""
endDate = ""
annualpeak<-readNWISpeak(mysite, startDate, endDate)
Qmax_2025 = 180000 #4/5/2025 11 CDT

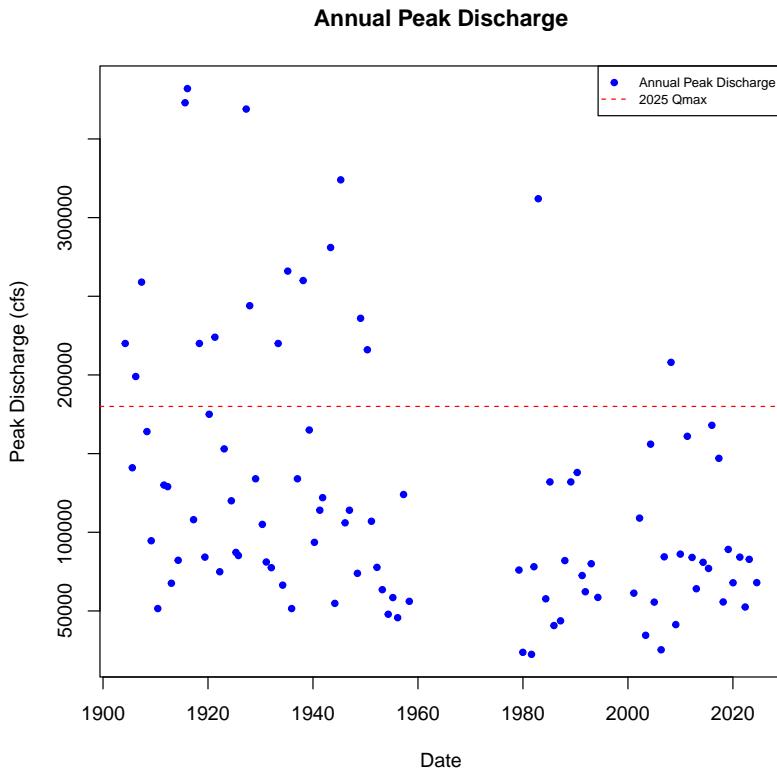
names(annualpeak)[1:5]

## [1] "agency_cd" "site_no"    "peak_dt"    "peak_tm"
## [5] "peak_va"
```

Plot Data

```
### STEP 4
### Plot the data
par(mfrow=c(1,1))
plot(annualpeak$peak_dt, annualpeak$peak_va, type="p", pch=19, cex=.7, col="blue",
      xlab="Date", ylab="Peak Discharge (cfs)", main="Annual Peak Discharge")
abline(h=Qmax_2025, col="red", lty=2)
```

```
legend("topright", legend=c("Annual Peak Discharge", "2025 Qmax"),
      col=c("blue", "red"), pch=c(19, NA), cex = .7, lty=c(NA, 2))
```



Determine Station Records Ranges ()

```
annualpeak = annualpeak[complete.cases(annualpeak$peak_va), ]
annualpeak = annualpeak[complete.cases(annualpeak$peak_dt), ]
dt_range = range(annualpeak$peak_dt); dt_range

## [1] "1904-03-27" "2024-07-18"

va_range = range(annualpeak$peak_va); va_range

## [1] 22400 382000
```

Split Data into Two Periods

```

### STEP 4
### Split the downloaded data into two periods
period1 <-subset(annualpeak, peak_dt>=dt_range[1] & peak_dt<="2010-12-31")
period2 <-subset(annualpeak, peak_dt>="2010-12-31" & peak_dt<=dt_range[2])

## NOT RUN -- just an example
### STEP 5
### Split the plot window in two columns
par(mfrow=c(1,2))

plot(period1$peak_dt, period1$peak_va, type="p", pch=19,
      col="blue", xlab="Date", ylab="Peak Discharge (cfs)", main="Period 1")
plot(period2$peak_dt, period2$peak_va, type="p", pch=19,
      col="red", xlab="Date", ylab="Peak Discharge (cfs)", main="Period 2")
### Add a legend
legend("topright", legend=c("Period 1", "Period 2"),
       col=c("blue", "red"), pch=19)

```

Fit Peak Flow with 'Method of Moments'

```

### STEP 6
### Perform Flood Frequency Analysis
### Locate the column of your data set that has the peak discharges
### You can see that peak discharges are stored in the 6th column (peak_va)

Q = period1$peak_va
# extract year from date range??
# Q$peak_dt = as.Date(period1$peak_dt, format="%Y")

# format(as.Date(period1$peak_dt, format="%d/%m/%Y"), "%Y")
graphlab = paste(range(format(as.Date(period1$peak_dt, format="%d/%m/%Y"), "%Y"))), collapse = "-")

# Generate plotting positions
n = length(Q); n

## [1] 81

r = n + 1 - rank(Q); head(r) # highest Q has rank r = 1

## [1] 14 24 18 9 21 42

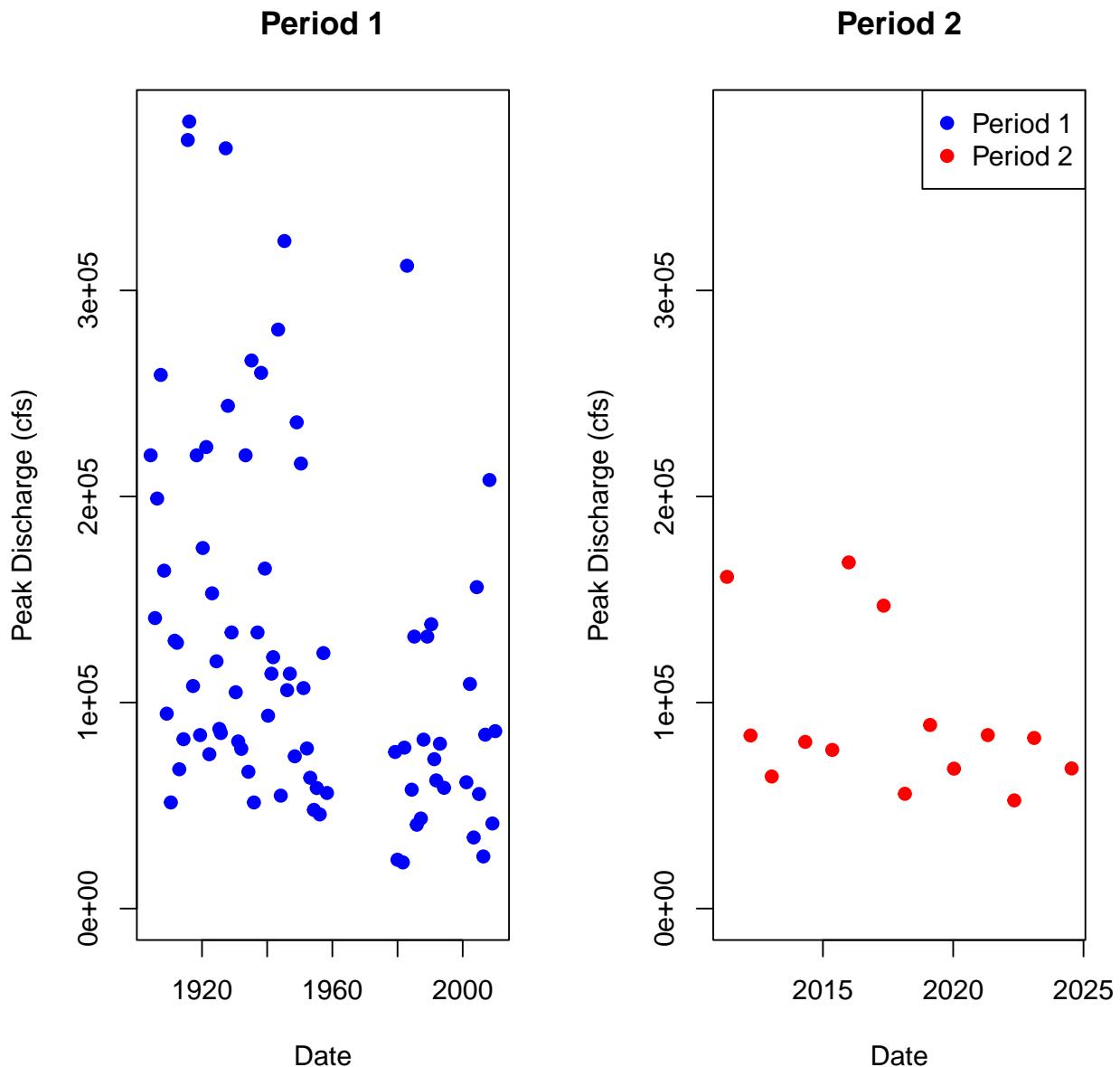
T = (n + 1)/r; head(T)

## [1] 5.857143 3.416667 4.555556 9.111111 3.904762 1.952381

```

```
# Added ylim for the figures to match
par(mfrow=c(1,2))

plot(period1$peak_dt, period1$peak_va, type="p", pch=19,
      col="blue", xlab="Date", ylab="Peak Discharge (cfs)",
      main="Period 1", ylim=c(0, va_range[2]))
plot(period2$peak_dt, period2$peak_va, type="p", pch=19,
      col="red", xlab="Date", ylab="Peak Discharge (cfs)",
      main="Period 2", ylim=c(0, va_range[2]))
### Add a legend
legend("topright", legend=c("Period 1", "Period 2"),
       col=c("blue", "red"), pch=19)
```

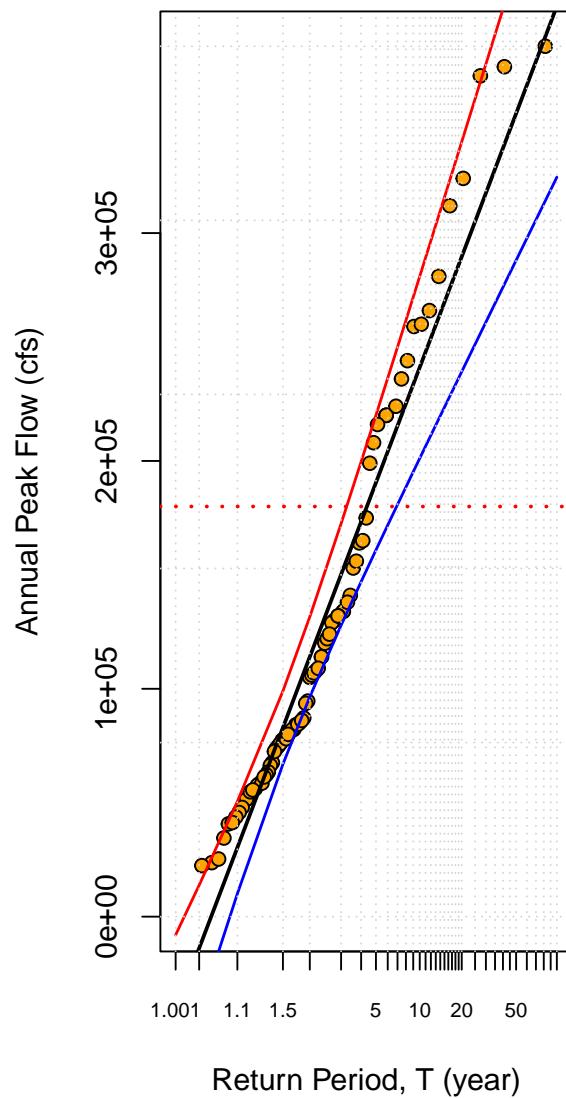
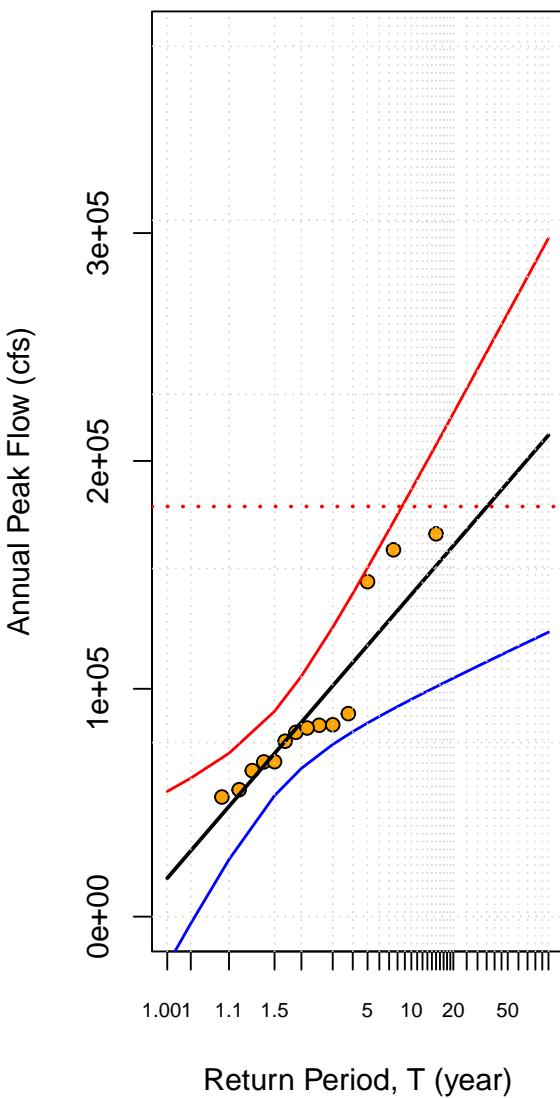


```

# Set up x axis tick positions and labels
Ttick = c(1.001,1.01,1.1,1.5,2,3,4,5,6,7,8,9,10,11,12,13,
         14,15,16,17,18,19,20,25,30,35,40,45,50,60,70,80,90,100)
xtlab = c(1.001,1.01,1.1,1.5,2,NA,NA,5,NA,NA,NA,NA,NA,10,NA,
          NA,NA,NA,15,NA,NA,NA,NA,20,NA,30,NA,NA,NA,50,NA,NA,NA,NA,100)
y = -log(-log(1 - 1/T))
ytick = -log(-log(1 - 1/Ttick))
xmin = min(min(y),min(ytick))
xmax = max(ytick)

# Fit a line by method of moments, along with 95% confidence intervals
KTtick = -(sqrt(6)/pi)*(0.5772 + log(log(Ttick/(Ttick-1))))
QTtick = mean(Q) + KTtick*sd(Q)
nQ = length(Q)
se = (sd(Q)*sqrt((1+1.14*KTtick + 1.1*KTtick^2)))/sqrt(nQ)
LB = QTtick - qt(0.975, nQ - 1)*se
UB = QTtick + qt(0.975, nQ - 1)*se
max = max(UB)
Qmax = max(QTtick)

```

07061000 1904–2009**07061000 2011–2024**

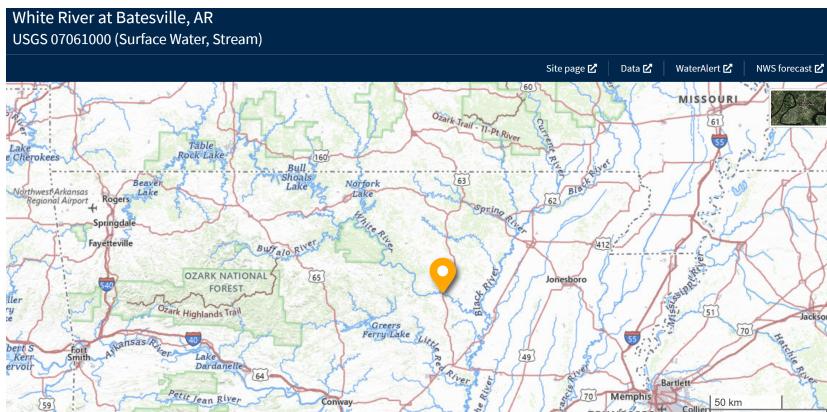


Figure 4: White River, Arkansas, USGS gage 07052500. Annual peak flow frequency analysis for the period 2000-2025 (left) and 1980-2000 (right). The red dashed line indicates the 2025 projected peak flow.

Using Functions to Plot Flood Return Intervals

We have three functions prepared for today's activities – which is partially based on the current flooding situation in the US.

First, we define a site based on the USGS gage number.

```
mysite <- "03287500"
startDate = ""
endDate = ""
Qmax_2025 = 113000 #4/6/2025 1 pm CDT
splitdate = "2000-01-01"
```

Using a function, we'll create a data frame with the site number in the name so we can track the file easier.

```
site03287500 <- getAnnualPeak.fun(mysite, startDate, endDate)

## Some dates could not be converted to a valid date, and
## were returned as NA
```

As always, let's check the data frame to make sure it looks correct.

```
head(site03287500)

##   agency_cd site_no    peak_dt peak_tm peak_va peak_cd
## 6      USGS 03287500 1895-01-19     <NA>    41600    <NA>
## 7      USGS 03287500 1896-03-20     <NA>    47800    <NA>
## 8      USGS 03287500 1897-02-24     <NA>    76900    <NA>
## 9      USGS 03287500 1898-01-23     <NA>    59800    <NA>
## 10     USGS 03287500 1899-03-08     <NA>    76400    <NA>
## 11     USGS 03287500 1900-03-22     <NA>    29600    <NA>
##   gage_ht gage_ht_cd year_last_pk ag_dt ag_tm ag_gage_ht
## 6     16.7       <NA>           NA    <NA>    <NA>        NA
## 7     19.0       <NA>           NA    <NA>    <NA>        NA
## 8     32.2       <NA>           NA    <NA>    <NA>        NA
## 9     25.1       <NA>           NA    <NA>    <NA>        NA
## 10    29.6       <NA>           NA    <NA>    <NA>        NA
## 11    12.7       <NA>           NA    <NA>    <NA>        NA
##   ag_gage_ht_cd
## 6             <NA>
## 7             <NA>
## 8             <NA>
## 9             <NA>
## 10            <NA>
## 11            <NA>

str(site03287500)
```

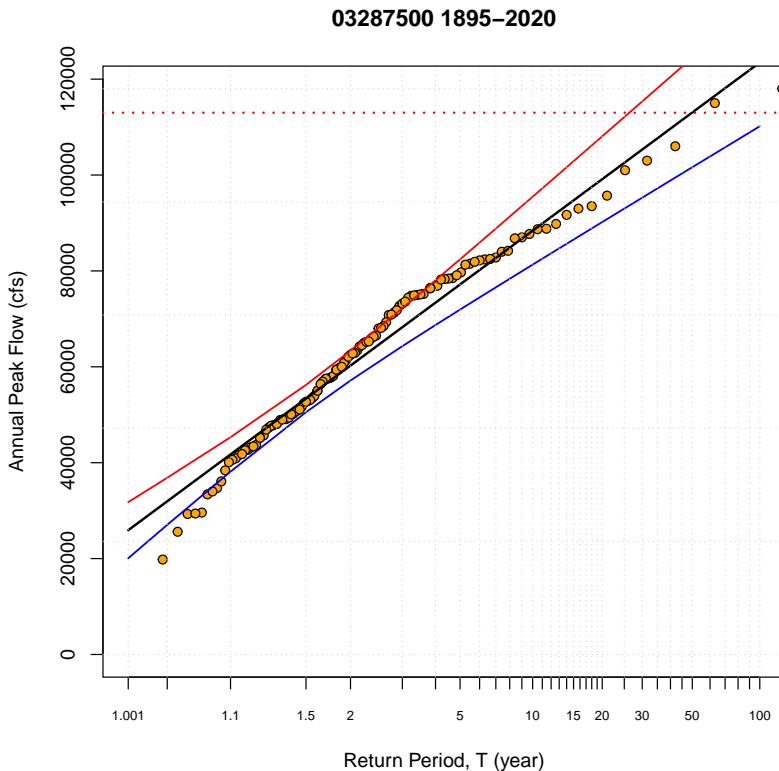
```

## 'data.frame': 125 obs. of 13 variables:
## $ agency_cd      : chr "USGS" "USGS" "USGS" "USGS" ...
## $ site_no        : chr "03287500" "03287500" "03287500" "03287500" ...
## $ peak_dt        : Date, format: "1895-01-19" ...
## $ peak_tm        : chr NA NA NA NA ...
## $ peak_va        : num 41600 47800 76900 59800 76400 29600 57600 68000 70800 33400 ...
## $ peak_cd        : chr NA NA NA NA ...
## $ gage_ht        : num 16.7 19 32.2 25.1 29.6 12.7 24 27 28.5 14.5 ...
## $ gage_ht_cd     : chr NA NA NA NA ...
## $ year_last_pk   : logi NA NA NA NA NA NA ...
## $ ag_dt          : Date, format: NA ...
## $ ag_tm          : chr NA NA NA NA ...
## $ ag_gage_ht     : num NA NA NA NA NA NA NA NA NA ...
## $ ag_gage_ht_cd: chr NA NA NA NA ...
## - attr(*, "queryTime")= POSIXct[1:1], format: "2025-04-06 12:58:14"
## - attr(*, "url")= chr "https://nwis.waterdata.usgs.gov/usa/nwis/peak/?site_no=03287500&range_selection=0&start_date=1895-01-01T00:00:00Z&end_date=2025-04-06T12:58:14Z"
## - attr(*, "headerInfo")=List of 8
##   ..$ date              : chr "Sun, 06 Apr 2025 19:58:13 GMT"
##   ..$ server            : chr "Apache"
##   ..$ strict-transport-security : chr "max-age=31536000"
##   ..$ amfplus-ver       : chr "1.3.2.2"
##   ..$ x-ua-compatible    : chr "IE=edge,chrome=1"
##   ..$ access-control-allow-origin: chr "*"
##   ..$ transfer-encoding   : chr "chunked"
##   ..$ content-type        : chr "text/plain; charset=UTF-8"
##   ..- attr(*, "class")= chr [1:2] "insensitive" "list"
## - attr(*, "comment")= chr [1:72] "#" "# U.S. Geological Survey" "# National Water Information System"
## - attr(*, "siteInfo")='data.frame': 1 obs. of 42 variables:
##   ..$ agency_cd      : chr "USGS"
##   ..$ site_no        : chr "03287500"
##   ..$ station_nm     : chr "KENTUCKY RIVER AT LOCK 4 AT FRANKFORT, KY"
##   ..$ site_tp_cd     : chr "ST"
##   ..$ lat_va         : num 381241
##   ..$ long_va        : num 845227
##   ..$ dec_lat_va     : num 38.2
##   ..$ dec_long_va    : num -84.9
##   ..$ coord_meth_cd  : chr "N"
##   ..$ coord_acy_cd   : chr "1"
##   ..$ coord_datum_cd : chr "NAD83"
##   ..$ dec_coord_datum_cd: chr "NAD83"
##   ..$ district_cd    : chr "21"
##   ..$ state_cd       : chr "21"
##   ..$ county_cd      : chr "073"

```

Now let's see what the flood return interval looks like for the flooding this week.

```
par(mfrow=c(1,1))  
plot_floodreturn.fun(site03287500, Qmax_2025)
```

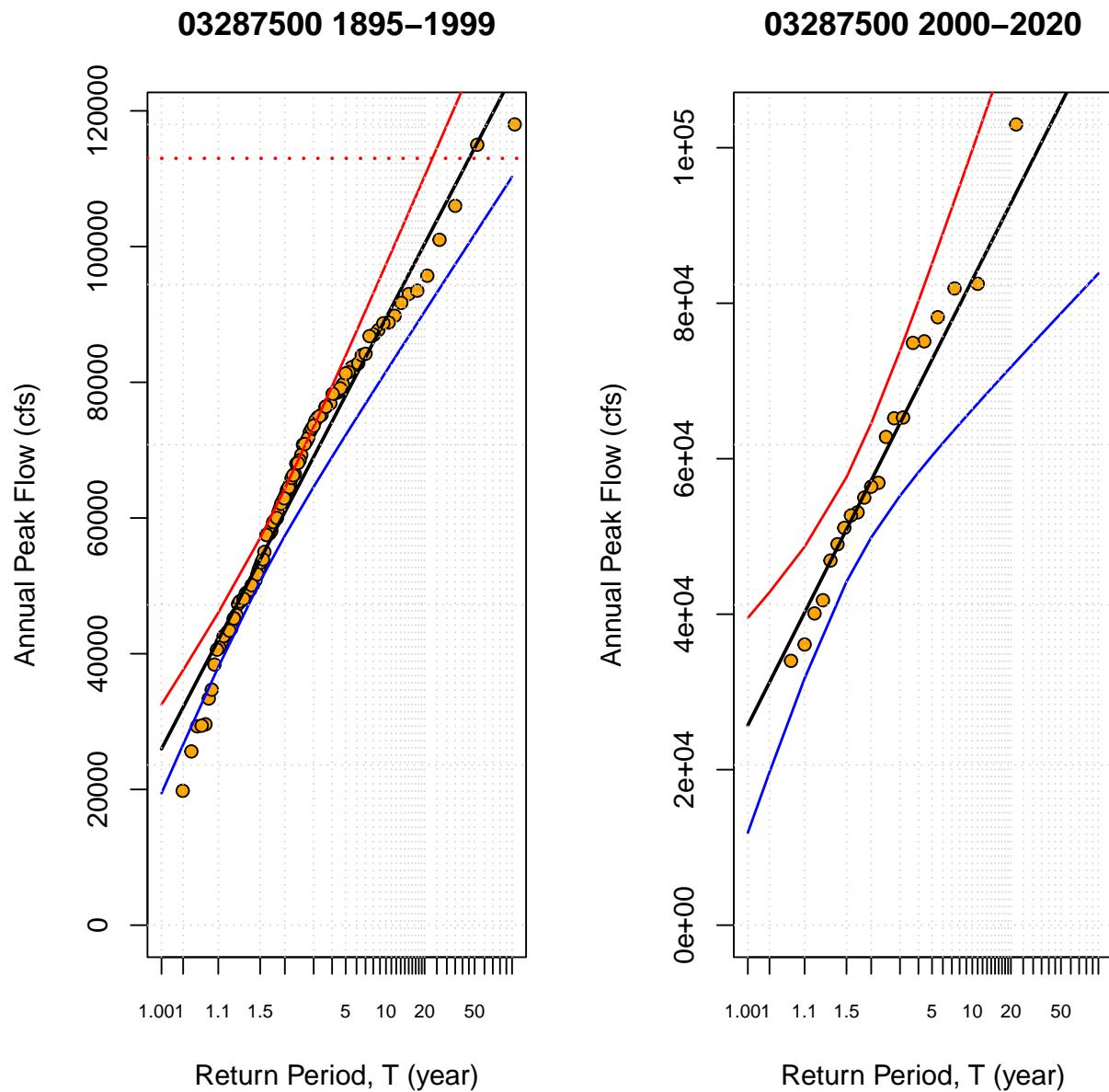


Finally, we'll split the data and look at how the record changes based on historic and recent data. I created a "list" of two dataframes, but then had to separate them so they would work with my function. See if you can spot the problem with this when you look at the y-axis.

```
# Split Data
periodlist = splitdata.fun(site03287500, splitdate)
period1 = periodlist[[1]]
period2 = periodlist[[2]]
```

We often use plotted data to evaluate the flood risk, relative to the record. In each case, we can determine the relative severity of flood based on the historic record and relative to potential changes in landuse.

```
plot_floodreturn.fun(period1, Qmax_2025)
plot_floodreturn.fun(period2, Qmax_2025)
```



Guide Function Details

Get Annual Peak Flow Data Function

This function retrieves annual peak flow data from the USGS NWIS database. It requires the USGS gage number as input and can optionally take start and end dates to filter the data. The function returns a data frame containing the annual peak flow data.

The function also removes any rows with missing values in the peak flow or date columns to ensure that the data is clean and ready for analysis.

usage: (getAnnualPeak.fun(site, startDate="", endDate=""))

Site is the USGS gage number, startDate and endDate are optional arguments. If not provided, the function will return all available data.

```
## getAnnualPeak.fun <- function (site, startDate = "", endDate = "") {
## {
##   library(dataRetrieval)
##   library(ggplot2)
##   suppressMessages(library(dplyr))
##   annualpeak <- readNWISpeak(site, startDate, endDate)
##   annualpeak = annualpeak[complete.cases(annualpeak$peak_va),
##     ]
##   annualpeak = annualpeak[complete.cases(annualpeak$peak_dt),
##     ]
##   return(annualpeak)
## }
```

Split Data Function

```
splitdata.fun<- function(annualpeak, splitdate){

  period1 = annualpeak[annualpeak$peak_dt < splitdate,]
  period2 = annualpeak[annualpeak$peak_dt >= splitdate,]
  periodlist = list(period1, period2)
  return(periodlist)
}
```

Plot Flood Return Interval

usage: (plot_floodreturn.fun(annualpeak.df, Q_max)

```

## plot_floodreturn.fun <- function (annualpeak, Qmax_2025)
## {
##   graphlab = paste(range(format(as.Date(annualpeak$peak_dt,
##                                     format = "%d/%m/%Y"), "%Y")), collapse = "-")
##   Q = annualpeak$peak_va
##   va_range = range(annualpeak$peak_va)
##   site_no = annualpeak$site_no[1]
##   n = length(Q)
##   r = n + 1 - rank(Q)
##   T = (n + 1)/r
##   Ttick = c(1.001, 1.01, 1.1, 1.5, 2, 3, 4, 5, 6, 7, 8, 9,
##            10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 25, 30, 35,
##            40, 45, 50, 60, 70, 80, 90, 100)
##   xtlab = c(1.001, 1.01, 1.1, 1.5, 2, NA, NA, 5, NA, NA, NA,
##            NA, 10, NA, NA, NA, NA, 15, NA, NA, NA, NA, 20, NA, 30,
##            NA, NA, NA, 50, NA, NA, NA, NA, 100)
##   y = -log(-log(1 - 1/T))
##   ytick = -log(-log(1 - 1/Ttick))
##   xmin = min(min(y), min(ytick))
##   xmax = max(ytick)
##   KTTick = -(sqrt(6)/pi) * (0.5772 + log(log(Ttick/(Ttick -
##             1))))
##   QTTick = mean(Q) + KTTick * sd(Q)
##   nQ = length(Q)
##   se = (sd(Q) * sqrt((1 + 1.14 * KTTick + 1.1 * KTTick^2)))/sqrt(nQ)
##   LB = QTTick - qt(0.975, nQ - 1) * se
##   UB = QTTick + qt(0.975, nQ - 1) * se
##   max = max(UB)
##   Qmax = max(QTTick)
##   plot(y, Q, ylab = expression("Annual Peak Flow (cfs)"), xaxt = "n",
##        xlab = "Return Period, T (year)", ylim = c(0, va_range[2]),
##        xlim = c(xmin, xmax), pch = 21, bg = "orange", main = paste(site_no,
##                      graphlab))
##   par(cex = 0.65)
##   axis(1, at = ytick, labels = as.character(xtlab))
##   lines(ytick, QTICK, col = "black", lty = 1, lwd = 2)
##   lines(ytick, LB, col = "blue", lty = 1, lwd = 1.5)
##   lines(ytick, UB, col = "red", lty = 1, lwd = 1.5)
##   abline(v = ytick, lty = 3, col = "light gray")
##   abline(h = seq(0, va_range[2], length.out = 6), lty = 3,
##          col = "light gray")
##   par(cex = 1)
##   abline(h = Qmax_2025, lty = 3, lwd = 2, col = "red")

```

```
## }
## <bytecode: 0x5bc2788>
```

References