

# Univariate Extreme Value Analysis of Daily Clemson Rainfall Data

Whitney Huang

November 05, 2024

## Contents

Load the USHCN dataset . . . . .	1
Extract the time series from a station near Clemson . . . . .	1
Extract Annual Maximum Values . . . . .	2
Block-Maxima Method . . . . .	3
Step 1: Determine the block size and compute the maxima for the blocks . . . . .	3
Step 2: Fit a GEV to the maxima and assess the fit . . . . .	4
Step 3: Perform inference for return levels . . . . .	6
Peak-Over-Threshold Method . . . . .	8
Step 1: Pick a threshold and extract the threshold exceedances . . . . .	8
Step 2: Fit a GPD to threshold excesses and assess the fit . . . . .	11
Step 3: Perform inference for return levels . . . . .	12

These data were taken from United States Historical Climatology Network (<http://cdiac.ornl.gov/epubs/ndp/ushcn/ushcn.html>). These data were compiled by Will Kleiber from CU-Boulder, thanks Will!

## Load the USHCN dataset

```
# Note: you should place the data in the same folder as the R code
load("USHCNprcpSetup.rdata")
ls()
```

```
## [1] "days"      "elev"       "lon.lat"    "ns"         "PRCP"       "stn.name"   "years"
```

## Extract the time series from a station near Clemson

```
Clemson.lon.lat <- c(-82.8374, 34.6834)
library(fields)
dist2Clemson <- rdist.earth(matrix(Clemson.lon.lat, 1, 2), lon.lat, miles = F)
id <- which.min(dist2Clemson)
lon.lat[id,] # near Lamaster Dairy Center
```

```
##      lon      lat
## -82.8236 34.6603
```

```
stn.name[id]
```

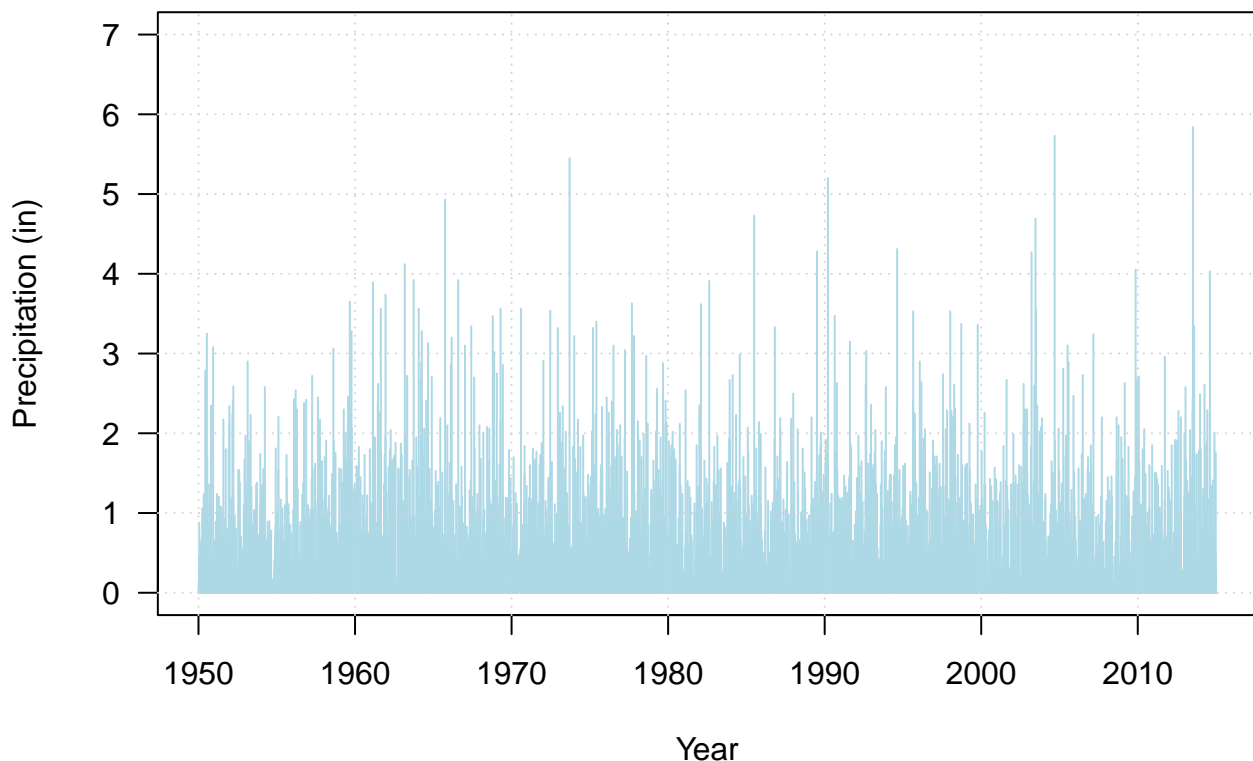
```
## [1] "CLEMSON UNIV, SC, 381770"
```

```
## Use data from 1950/1/1-2014/12/31
dat <- PRCP[(50 * 365 + 1):41975, id]
```

## Extract Annual Maximum Values

```
# Reformat the data into [day, year] and convert rainfall to inches
rain <- array(dat, dim = c(365, 65)) / 100
# Extract annual maxima
annmx <- apply(rain, 2, max, na.rm = T)
# Extract the timing of annual maxima
annmxT <- apply(rain, 2, which.max)
# Plot the daily time series
plot(1950 + 1:23725/365, c(rain),
     xlab = "Year", ylab = "Precipitation (in)",
     main = "Daily Precip in Clemson", type = "h",
     pch = 19, cex = 0.5, col = "lightblue",
     ylim = c(0, 7), las = 1)
grid()
```

## Daily Precip in Clemson



## Block-Maxima Method

We are going to conduct an extreme value analysis using the `extRemes` package, developed and maintained by Eric Gilleland.

### Step 1: Determine the block size and compute the maxima for the blocks

```
# Setting up the figure configuration
old.par <- par(no.readonly = TRUE)
mar.default <- par("mar")
mar.left <- mar.default
mar.right <- mar.default
mar.left[2] <- 0
mar.right[4] <- 0

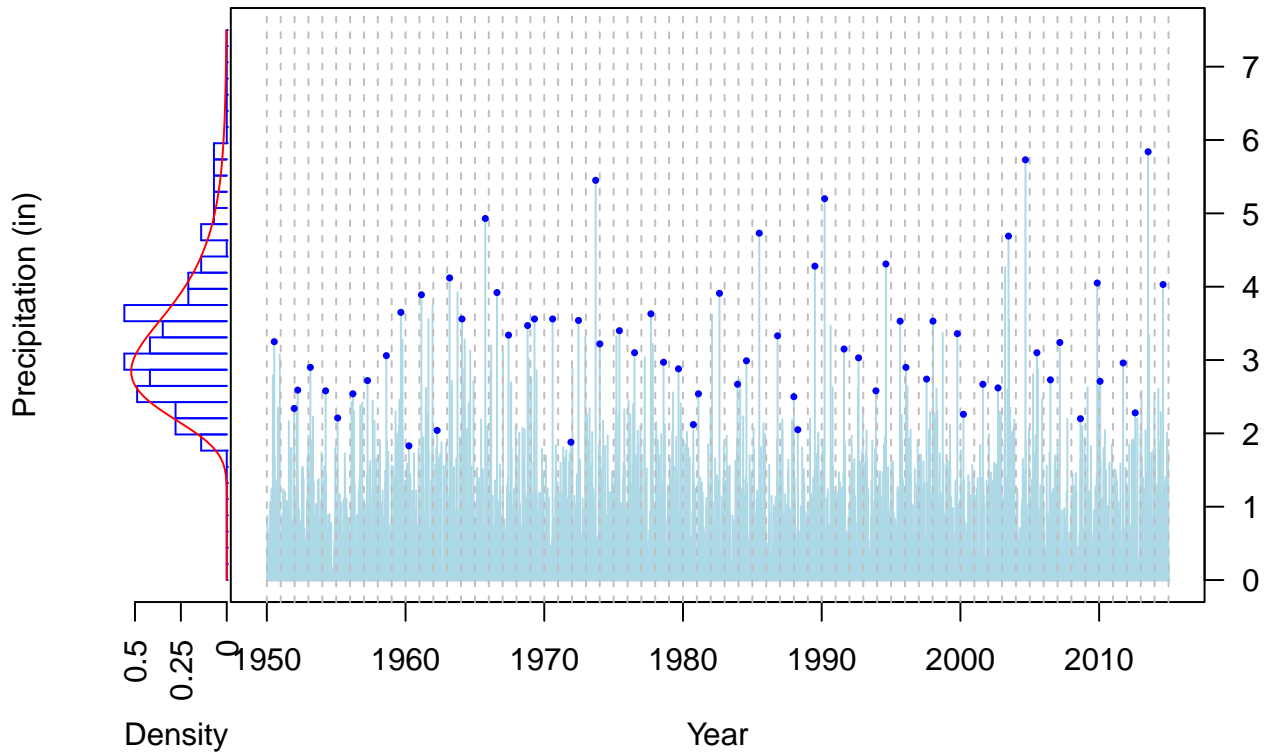
# Time series plot
par(fig = c(0.2, 1, 0, 1), mar = mar.left)
plot(1950 + 1:23725/365, c(rain),
     xlab = "Year", ylab = "",
     main = "Daily Precip in Clemson",
     type = "h", pch = 19, cex = 0.5, col = "lightblue",
     ylim = c(0, 7.5), yaxt = "n")
par(las = 2)
axis(4, at = 0:8)
par(las = 0)
mtext("Precipitation (in)", side = 2, line = 5)
abline(v = 1950:2015, col = "gray", lty = 2)
points(1950:2014 + annmxT / 365, annmx,
       pch = 16, col = "blue", cex = 0.5)

# Histogram
hs <- hist(annmx,
           breaks = seq(0, 7.5, length.out = 35),
           plot = FALSE)

par(fig = c(0, 0.2, 0, 1.0), mar = mar.right, new = T)
plot(NA, type = 'n', axes = FALSE, yaxt = 'n',
     col = rgb(0, 0, 0.5, alpha = 0.5),
     xlab = "Density", ylab = NA, main = NA,
     xlim = c(-0.55, 0),
     ylim = c(0, 7.5))
axis(1, at = c(-0.5, -0.25, 0), c(0.5, 0.25, 0), las = 2)
arrows(rep(0, length(hs$breaks[-35])), hs$breaks[-35],
       -hs$density, hs$breaks[-35], col = "blue",
       length = 0, angle = 0, lwd = 1)
arrows(rep(0, length(hs$breaks[-1])), hs$breaks[-1],
       -hs$density, hs$breaks[-1], col = "blue",
       length = 0, angle = 0, lwd = 1)
arrows(-hs$density, hs$breaks[-35], -hs$density,
       hs$breaks[-1], col = "blue", angle = 0,
       length = 0)
library(extRemes)
mle <- fevd(annmx)$results$par
```

```
xg <- seq(0, 7.5, length.out = 100)
library(ismev)
lines(-gev.dens(mle, xg), xg, col = "red")
```

## Daily Precip in Clemson



```
par(old.par)
```

### Step 2: Fit a GEV to the maxima and assess the fit

We assume the annual maxima  $m_1, \dots, m_t$  follows a GEV distribution  $GEV(\mu_n, \sigma_n, \xi)$  and we maximize the log-likelihood

$$-\sum_{i=1}^t m_i^{-\frac{1}{\xi}} - t \log(\sigma) - \left(\frac{1}{\xi} + 1\right) \sum_{i=1}^t \log(m_i)$$

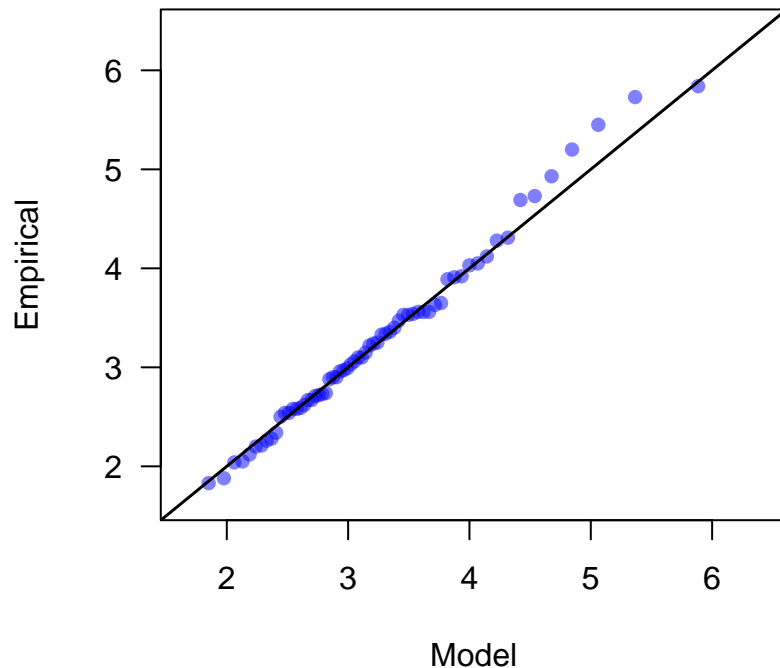
```
# Fit a GEV to annual maximum daily precip using MLE
library(scales)
gevfit <- fevd(annmx)
# Print the results
gevfit
```

```
##
## fevd(x = annmx)
##
## [1] "Estimation Method used: MLE"
```

```
##
##
## Negative Log-Likelihood Value: 80.60481
##
##
## Estimated parameters:
## location      scale      shape
## 2.85348346 0.70628902 0.01229713
##
## Standard Error Estimates:
## location      scale      shape
## 0.09985002 0.07338167 0.10043952
##
## Estimated parameter covariance matrix.
##          location      scale      shape
## location 0.009970026 0.003021057 -0.003803483
## scale    0.003021057 0.005384870 -0.002348046
## shape    -0.003803483 -0.002348046 0.010088098
##
## AIC = 167.2096
##
## BIC = 173.7328
```

```
#QQ plot
nmax <- length(annmx)
p <- 1:nmax / (nmax + 1)
qm <- gevq(mle, 1 - p)
rg <- c(0.9 * min(annmx), 1.1 * max(annmx))
plot(qm, sort(annmx), xlim = rg, ylim = rg, pch = 16,
     cex = 1, col = alpha("blue", 0.5), xlab = "Model",
     ylab = "Empirical", main = "Quantile Plot", las = 1)
abline(0, 1, lwd = 1.5)
```

## Quantile Plot



### Step 3: Perform inference for return levels

Suppose we are interested in estimating the 50-year return level

```
RL50 <- return.level(gevfit, return.period = 50) # Estimate of the 100-year event
RL50
```

```
## fevd(x = annmx)
## get(paste("return.level.fevd.", newcl, sep = ""))(x = x, return.period = return.period)
##
## GEV model fitted to annmx
## Data are assumed to be stationary
## [1] "Return Levels for period units in years"
## 50-year level
##      5.676568
```

```
# Quantify the estimate uncertainty
## Delta method
CI_delta <- ci(gevfit, return.period = 50, verbose = T)
```

```
##
## Preparing to calculate 95 % CI for 50-year return level
##
## Model is fixed
##
## Using Normal Approximation Method.
```

```
CI_delta
```

```
## fevd(x = annmx)
##
## [1] "Normal Approx."
##
## [1] "50-year return level: 5.677"
##
## [1] "95% Confidence Interval: (4.6309, 6.7223)"
```

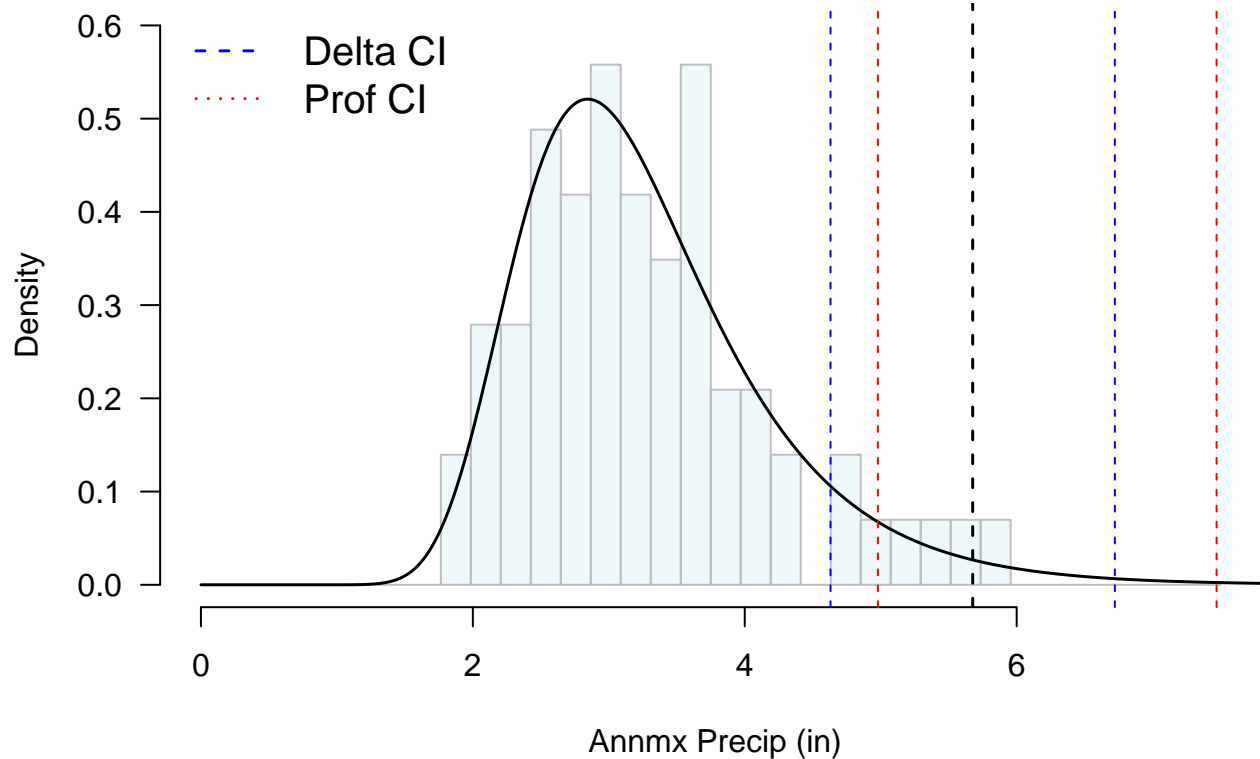
```
## Profile likelihood method
```

```
CI_prof <- ci(gevfit, method = "proflik", xrange = c(4, 10),
  return.period = 50, verbose = F)
CI_prof
```

```
## fevd(x = annmx)
##
## [1] "Profile Likelihood"
##
## [1] "50-year return level: 5.677"
##
## [1] "95% Confidence Interval: (4.9791, 7.4705)"
```

```
hist(annmx, breaks = seq(0, 7.5, length.out = 35),
  col = alpha("lightblue", 0.2), border = "gray",
  xlim = c(0, 7.5), prob = T, ylim = c(0, 0.6),
  xlab = " Annmx Precip (in)",
  main = "95% CI for 50-yr RL",
  las = 1)
xg <- seq(0, 10, len = 1000)
mle <- gevfit$results$par
lines(xg, gev.dens(mle, xg), lwd = 1.5)
for (i in c(1, 3)) abline(v = CI_delta[i], lty = 2, col = "blue")
for (i in c(1, 3)) abline(v = CI_prof[i], lty = 2, col = "red")
abline(v = RL50, lwd = 1.5, lty = 2)
legend("topleft", legend = c("Delta CI", "Prof CI"), col = c("blue", "red"),
  lty = c(2, 3), bty = "n", cex = 1.25, lwd = 1.5)
```

## 95% CI for 50-yr RL



## Peak-Over-Threshold Method

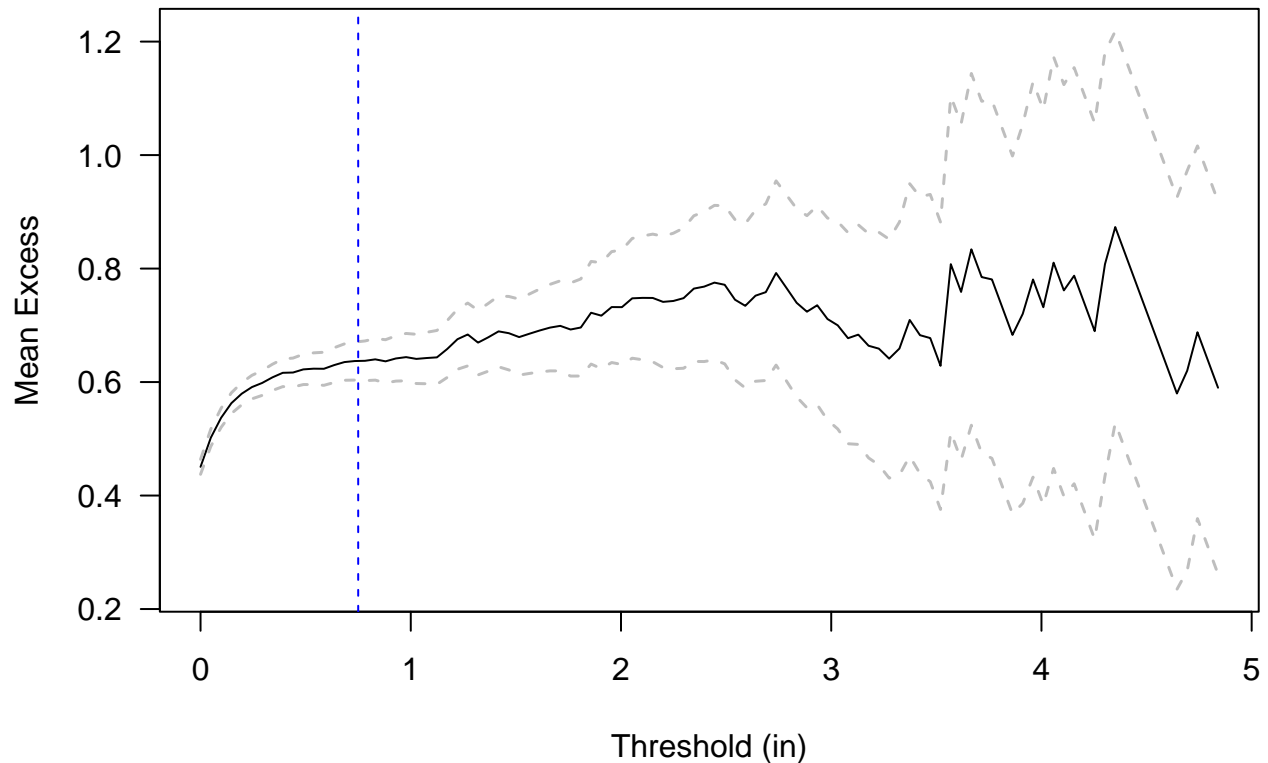
### Step 1: Pick a threshold and extract the threshold exceedances

I choose 0.75 as the threshold, but note that the ‘straightness’ of the mean residual life plot is difficult to assess in general

```
mrlplot(dat[!is.na(dat)] / 100, main = "Mean Residual Life", xlab = "Threshold (in)", las = 1)
# I choose 0.75 as the threshold but note that the "straightness"
# is difficult to assess
abline(v = 0.75, col = "blue", lty = 2)
```



## Mean Residual Life



```
old.par <- par(no.readonly = TRUE)
mar.default <- par('mar')
mar.left <- mar.default
mar.right <- mar.default
mar.left[2] <- 0
mar.right[4] <- 0
# Time series plot
par(fig = c(0.2, 1, 0, 1), mar = mar.left)
plot(1950 + 1:23725/365, c(rain),
     xlab = "Year", ylab = "",
     main = "Daily Precip in Clemson",
     type = "h", pch = 19, cex = 0.5, col = "lightblue",
     ylim = c(0, 7.5), yaxt = "n")
par(las = 2)
axis(4, at = 0:8)
par(las = 0)
mtext("Precipitation (in)", side = 2, line = 4)
#Threshold exceedances
thres <- 0.75
ex_id <- which(rain > thres)
ex <- rain[ex_id]
nex <- length(ex)
#Extract the timing of POT
abline(h = thres, col = "blue", lty = 2)
points(1950 + ex_id / 365, ex, col = alpha("blue", 0.5), pch = 16,
       cex = 0.75)
```

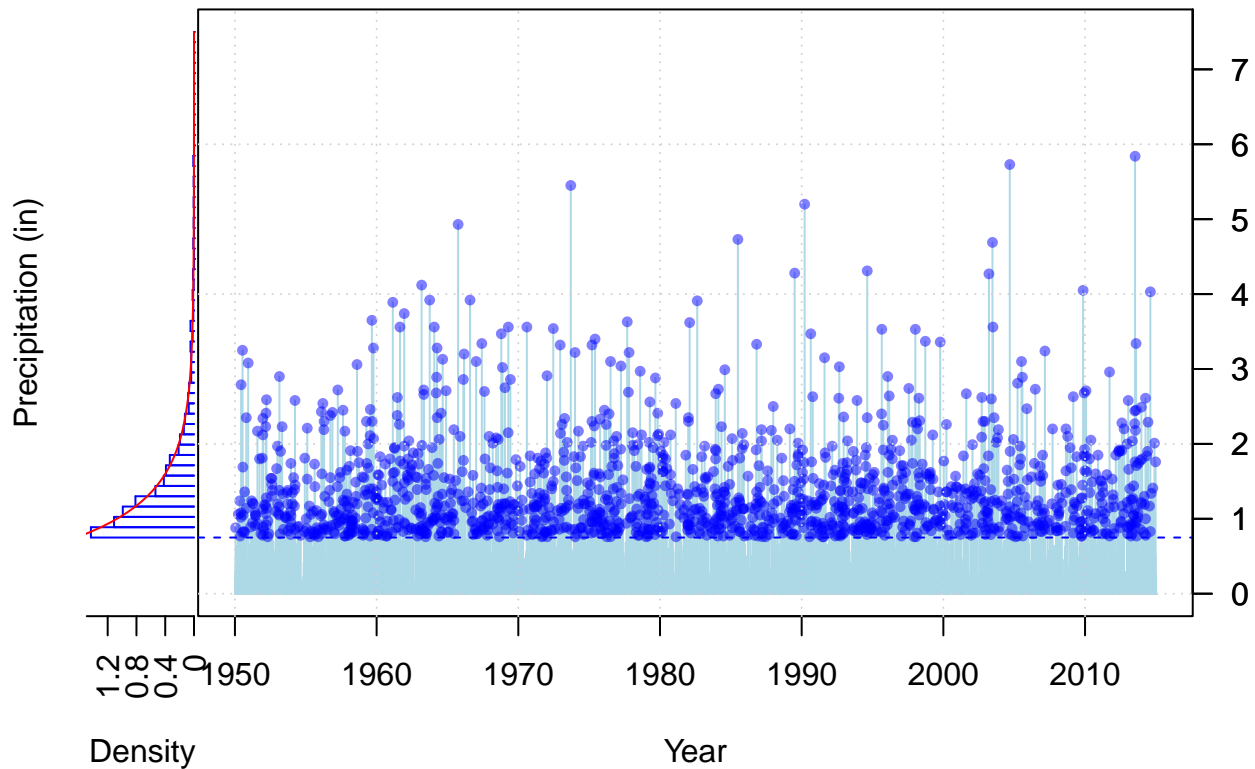
```

par(las = 2)
axis(4, at = 0:8)
par(las = 0)
grid()
hs <- hist(ex, seq(thres, 7.5, len = 50), plot = FALSE)
par(fig = c(0, 0.2, 0, 1.0), mar = mar.right, new = T)
plot(NA, type = 'n', axes = FALSE, yaxt = 'n',
     col = rgb(0,0,0.5, alpha = 0.5),
     xlab = "Density", ylab = NA, main = NA,
     xlim = c(-max(hs$density), 0),
     ylim = c(0, 7.5))
axis(1, at = c(-1.6, -1.2, -0.8, -0.4, 0),
     c(1.6, 1.2, 0.8, 0.4, 0), las = 2)
#abline(h = 21, col = "red", lty = 5)
arrows(rep(0, length(hs$breaks[-50])), hs$breaks[-50],
       -hs$density, hs$breaks[-50], col = "blue",
       length = 0, angle = 0, lwd = 1)
arrows(rep(0, length(hs$breaks[-1])), hs$breaks[-1],
       -hs$density, hs$breaks[-1], col = "blue",
       length = 0, angle = 0, lwd = 1)
arrows(-hs$density, hs$breaks[-50], -hs$density,
       hs$breaks[-1], col = "blue", angle = 0,
       length = 0)

mle <- fevd(c(rain[!is.na(rain)]), threshold = thres, type = "GP")$results$par
xg <- seq(thres, 7.5, length.out = 100)
lines(-gpd.dens(mle, thres, xg), xg, col = "red")

```

## Daily Precip in Clemson



```
par(old.par)
```

Step 2: Fit a GPD to threshold excesses and assess the fit

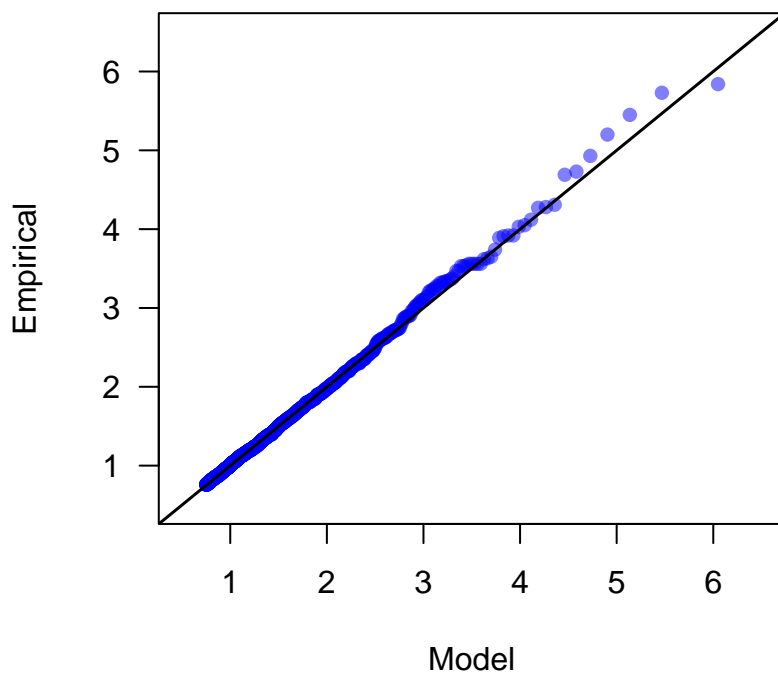
```
# Fit a GPD for threshold exceeances using MLE
gpdfit1 <- fevd(c(rain[!is.na(rain)]), threshold = thres, type = "GP")
# Print the results
gpdfit1
```

```
##
## fevd(x = c(rain[!is.na(rain)]), threshold = thres, type = "GP")
##
## [1] "Estimation Method used: MLE"
##
##
## Negative Log-Likelihood Value: 825.0139
##
##
## Estimated parameters:
##      scale      shape
## 0.61188988 0.04527592
##
## Standard Error Estimates:
##      scale      shape
```

```
## 0.02290870 0.02704591
##
## Estimated parameter covariance matrix.
##          scale          shape
## scale  0.0005248087 -0.0004281417
## shape -0.0004281417  0.0007314815
##
## AIC = 1654.028
##
## BIC = 1664.64
```

```
# QQ plot
p <- 1:nex / (nex + 1)
qm <- gpdq(mle, 0.75, 1 - p)
plot(qm, sort(ex), xlim = c(0.5, 6.5), ylim = c(0.5, 6.5),
     pch = 16, cex = 1, col = alpha("blue", 0.5),
     xlab = "Model", ylab = "Empirical", main = "Quantile Plot",
     las = 1)
abline(0, 1, lwd = 1.5)
```

## Quantile Plot



### Step 3: Perform inference for return levels

Again we are interested in estimating 50-year return level

```
RL50 <- return.level(gpdfit1, return.period = 50 * 365 / 365.25)
RL50
```

```
## fevd(x = c(rain[!is.na(rain)]), threshold = thres, type = "GP")
```

```
## get(paste("return.level.fevd.", newcl, sep = ""))(x = x, return.period = return.period)
##
## GP model fitted to c(rain[!is.na(rain)])
## Data are assumed to be stationary
## [1] "Return Levels for period units in years"
## 49.9657768651608-year level
## 5.829943
```

```
CI_delta <- ci(gpfit1, return.period = 50 * 365 / 365.25,
               verbose = F)
CI_delta
```

```
## fevd(x = c(rain[!is.na(rain)]), threshold = thres, type = "GP")
##
## [1] "Normal Approx."
##
## [1] "49.9657768651608-year return level: 5.83"
##
## [1] "95% Confidence Interval: (5.0405, 6.6194)"
```

```
CI_prof <- ci(gpfit1, method = "proflik", xrange = c(4, 10),
              return.period = 50 * 365 / 365.25, verbose = F)
CI_prof
```

```
## fevd(x = c(rain[!is.na(rain)]), threshold = thres, type = "GP")
##
## [1] "Profile Likelihood"
##
## [1] "49.9657768651608-year return level: 5.83"
##
## [1] "95% Confidence Interval: (5.2632, 6.7866)"
```

```
hist(ex, breaks = seq(thres, 6, 0.1), col = alpha("lightblue", 0.2), border = "gray",
     xlim = c(thres, 7.5), prob = T, ylim = c(0, 2),
     xlab = "Threshold excess (in)",
     main = "95% CI for 50-yr RL", las = 1)
xg <- seq(thres, 10, len = 1000)
mle <- gpfit1$results$par
lines(xg, gpd.dens(mle, thres, xg), lwd = 1.5)
for (i in c(1, 3)) abline(v = CI_delta[i], lty = 2, col = "blue")
for (i in c(1,3)) abline(v = CI_prof[i], lty = 2, col = "red")
abline(v = RL50, lwd = 1.5, lty = 2)
legend("topleft", legend = c("Delta CI", "Prof CI"),
      col = c("blue", "red"), lty = c(2, 3), bty = "n", cex = 1.25,
      lwd = 1.5)
```

### 95% CI for 50-yr RL

