MATH 8090: Extreme Value Analysis

Whitney Huang, Clemson University

10/31-11/2/2023

Contents

These data were taken from United States Historical Climatology Network. 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 at 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
```

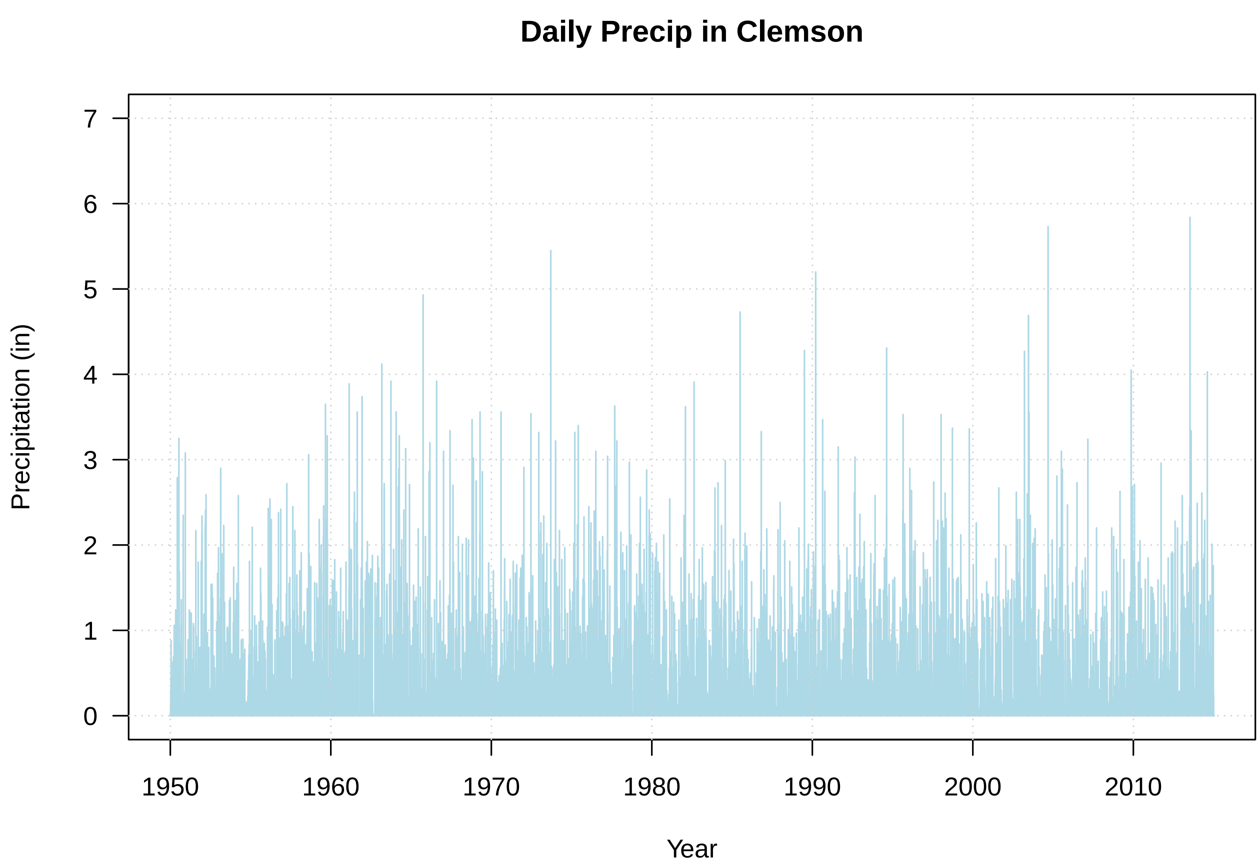
```
##      lon      lat
## -82.8236  34.6603
```

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

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

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

```r
# 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()
```



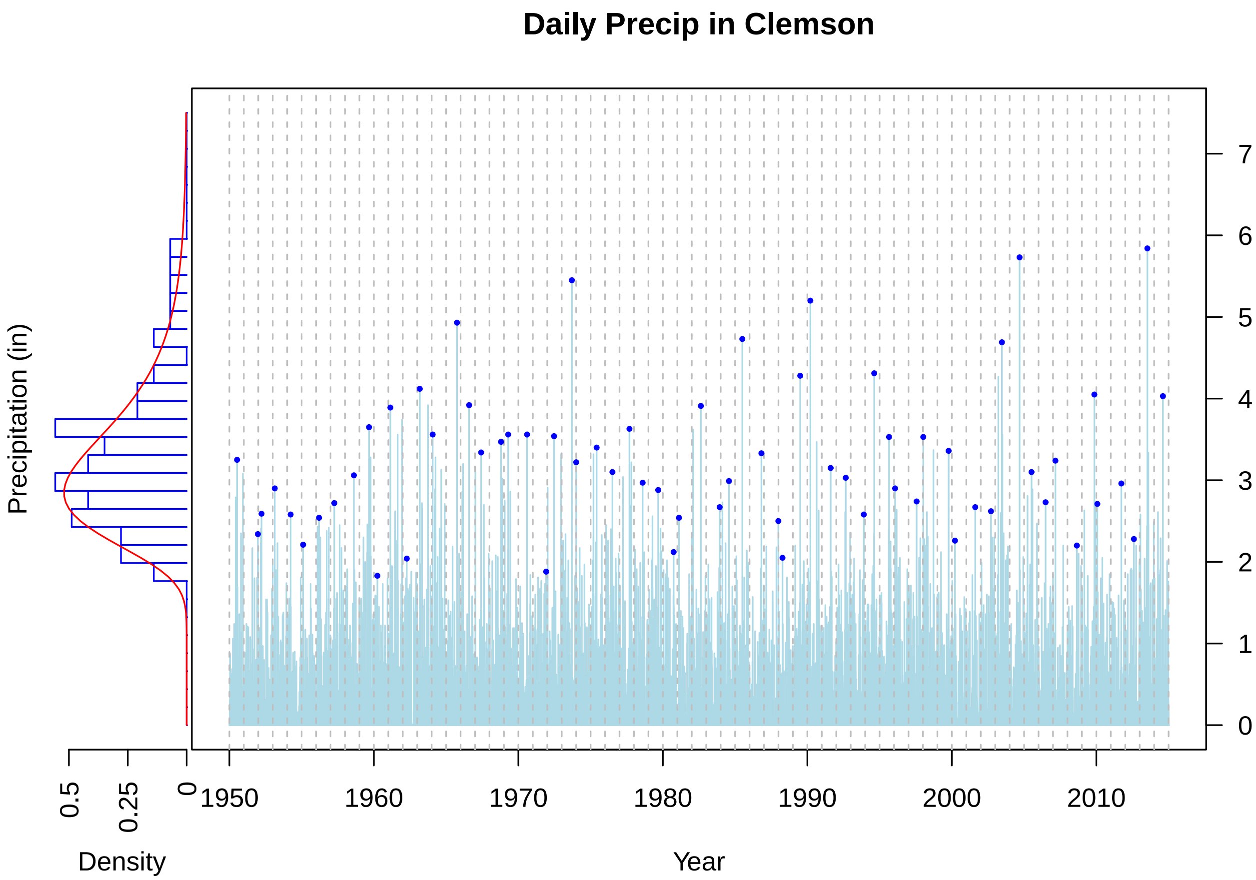
Block-Maxima Method

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

Step I: Determine the block size and compute maxima for blocks

```r
# 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
```

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




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

Step II: Fit a GEV to the maxima and assess the fit

We assume the annual maxima $m_1, \cdots, m_k$ follows a GEV distribution $GEV(\mu_k, \sigma_k, \xi)$ and we maximize the log-likelihood

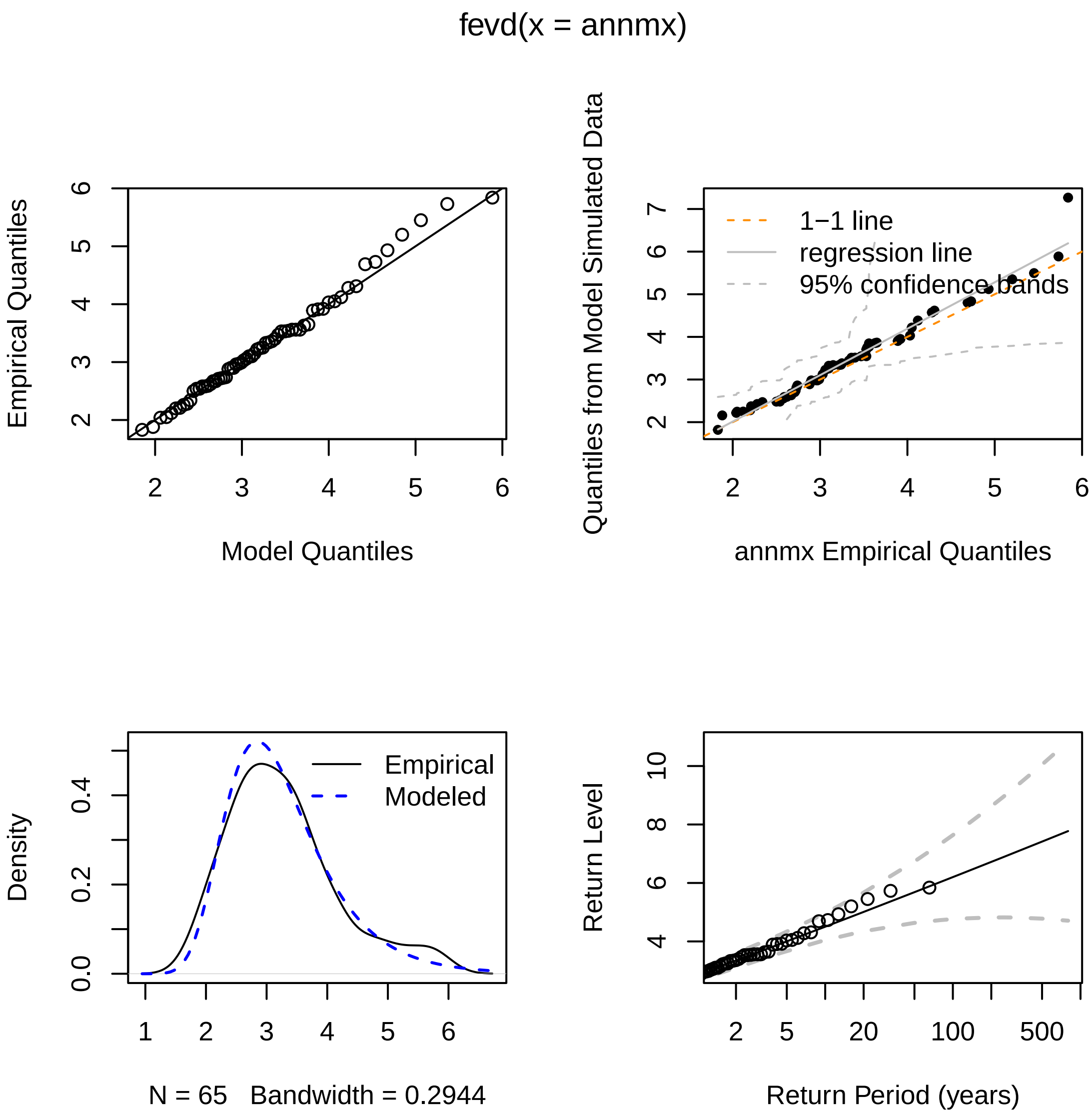
$$-\sum_{i=1}^{k} m_i^{-\frac{1}{\xi}} - k\log(\sigma) - (\frac{1}{\xi}+1)\sum_{i=1}^{k} log(m_i)$$

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

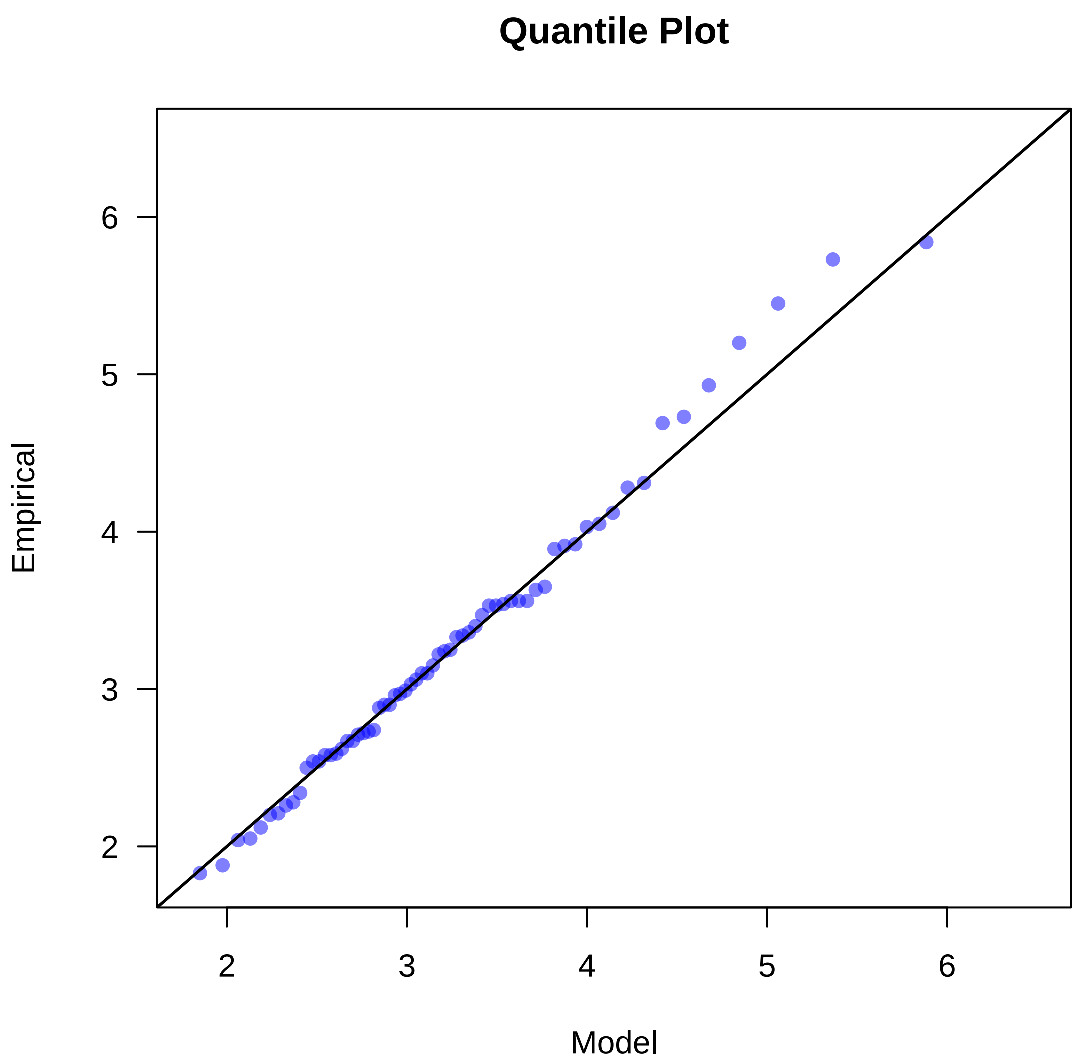
```
##
## fevd(x = annmx)
##
```

```
## [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
```

```r
plot(gevfit)
```




```
#QQ plot
p <- 1:65 / 66
qm <- gevq(mle, 1 - p)
plot(qm, sort(annmx), xlim = c(1.8, 6.5), ylim = c(1.8, 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)
```



Step III: Perform inference for return levels

Suppose we are interested in estimating 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.
```

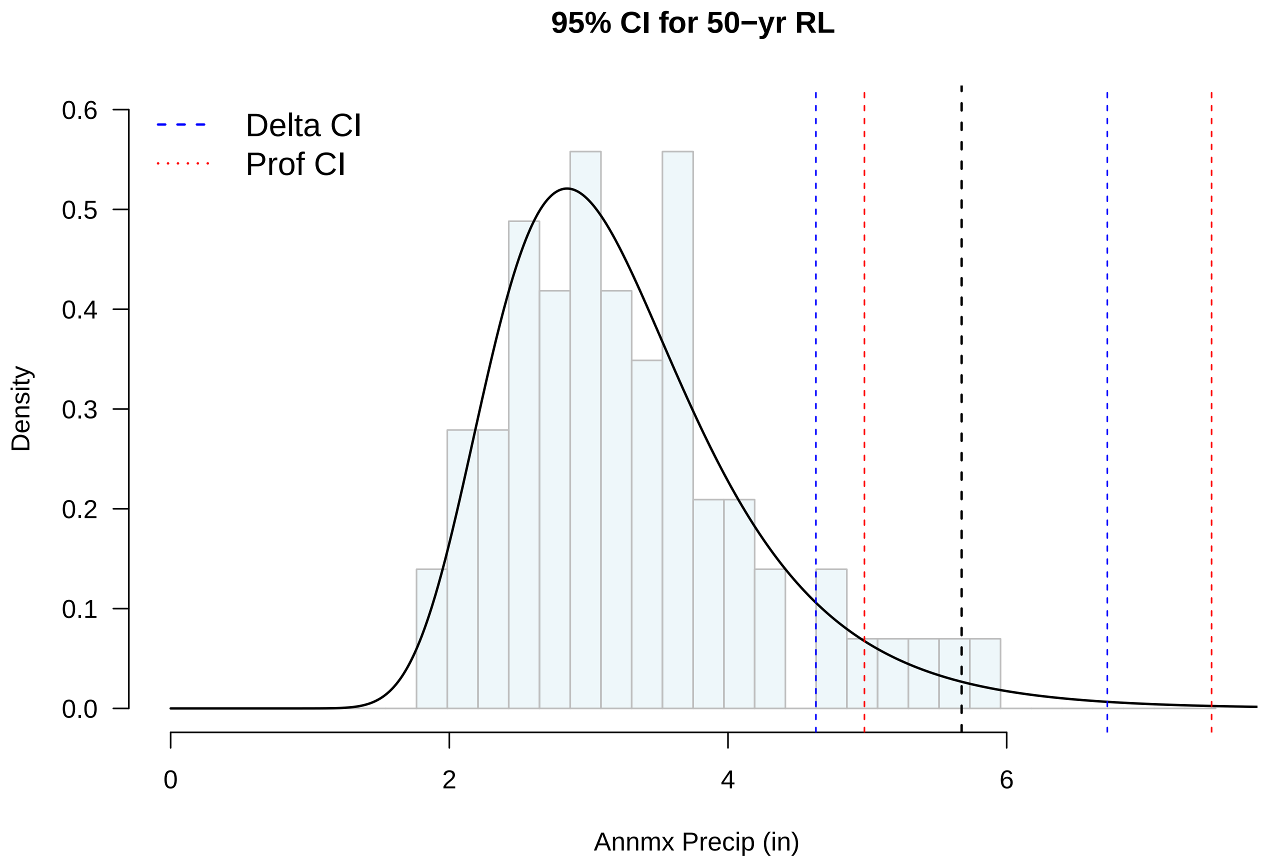
```
CI_delta
```

```
## 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)
```



Peak-Over-Threshold Method

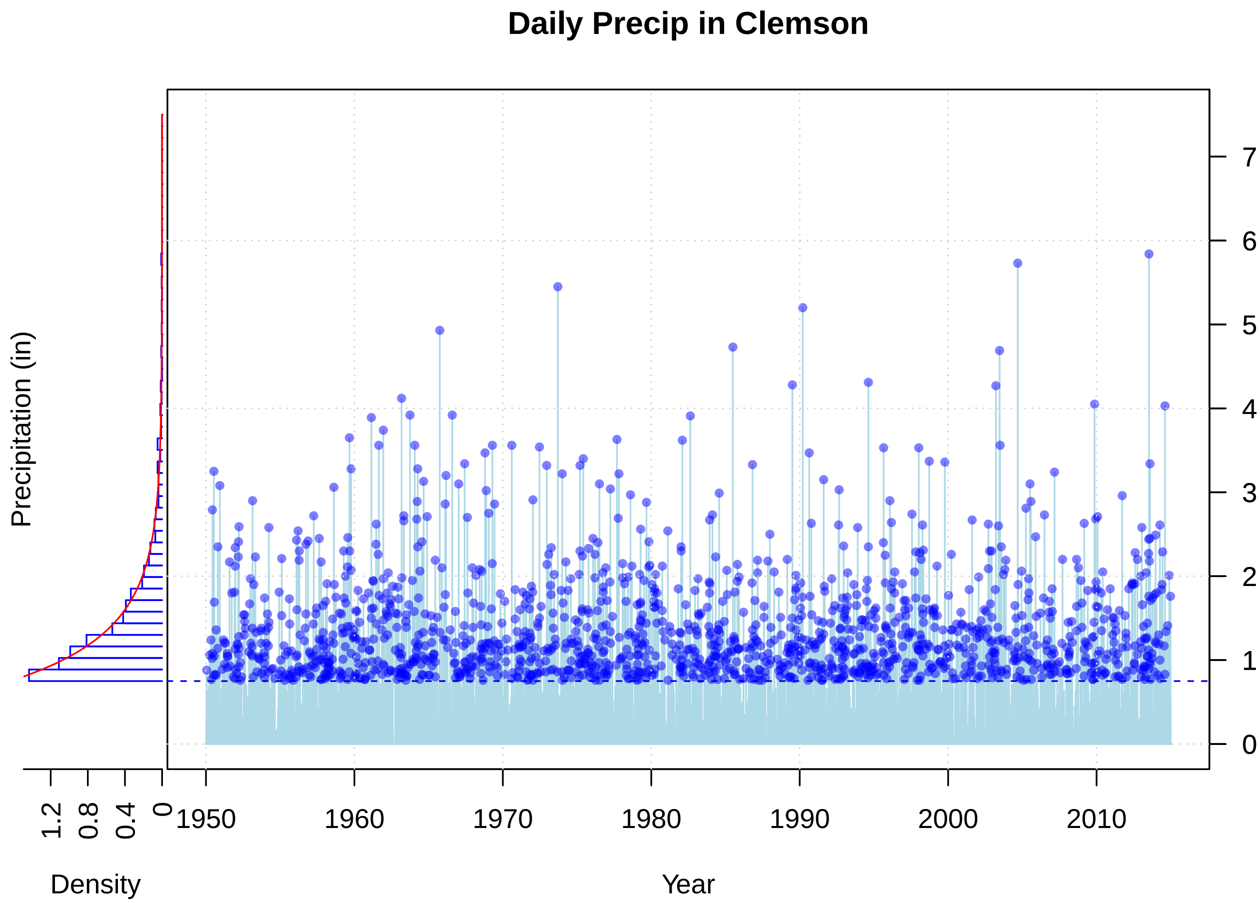
Step I: Pick a threshold and extract the threshold exceedances

```
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)
```

```r
ex <- rain[ex_id]
length(ex)
```

```
## [1] 1489
```

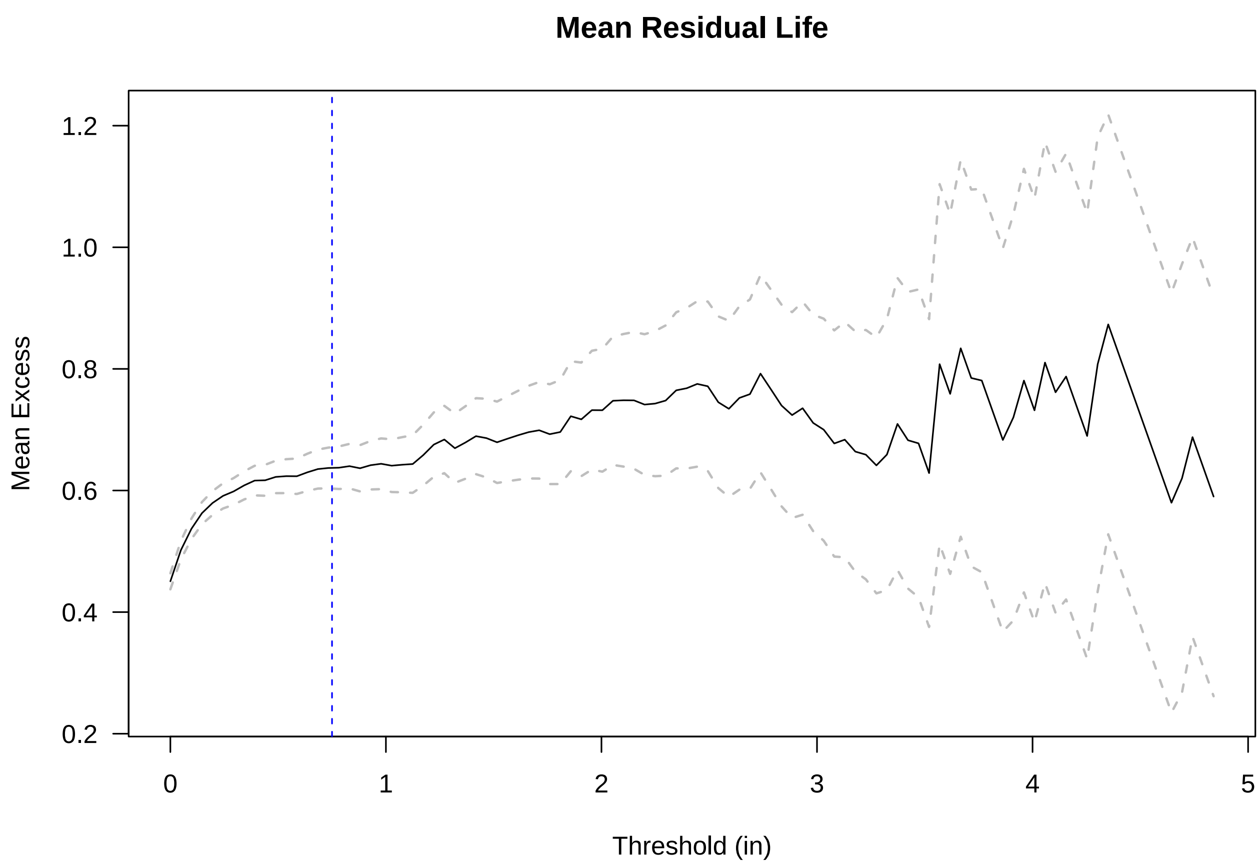
```r
#Extract the timing of POT
abline(h = thres, col = "blue", lty = 2)
points(1950 + ex_id / 365, ex, col = alpha("blue", 0.5), pch = 16,
       cex = 0.75)
par(las = 2)
axis(4, at = 0:8)
par(las = 0)
#mtext("Precipitation (in)", side = 2, line = 5)
grid()
hs <- hist(ex, seq(thres, 7.5, len = 50), 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")
```




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

How to choose the "right" threshold

```
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)
```



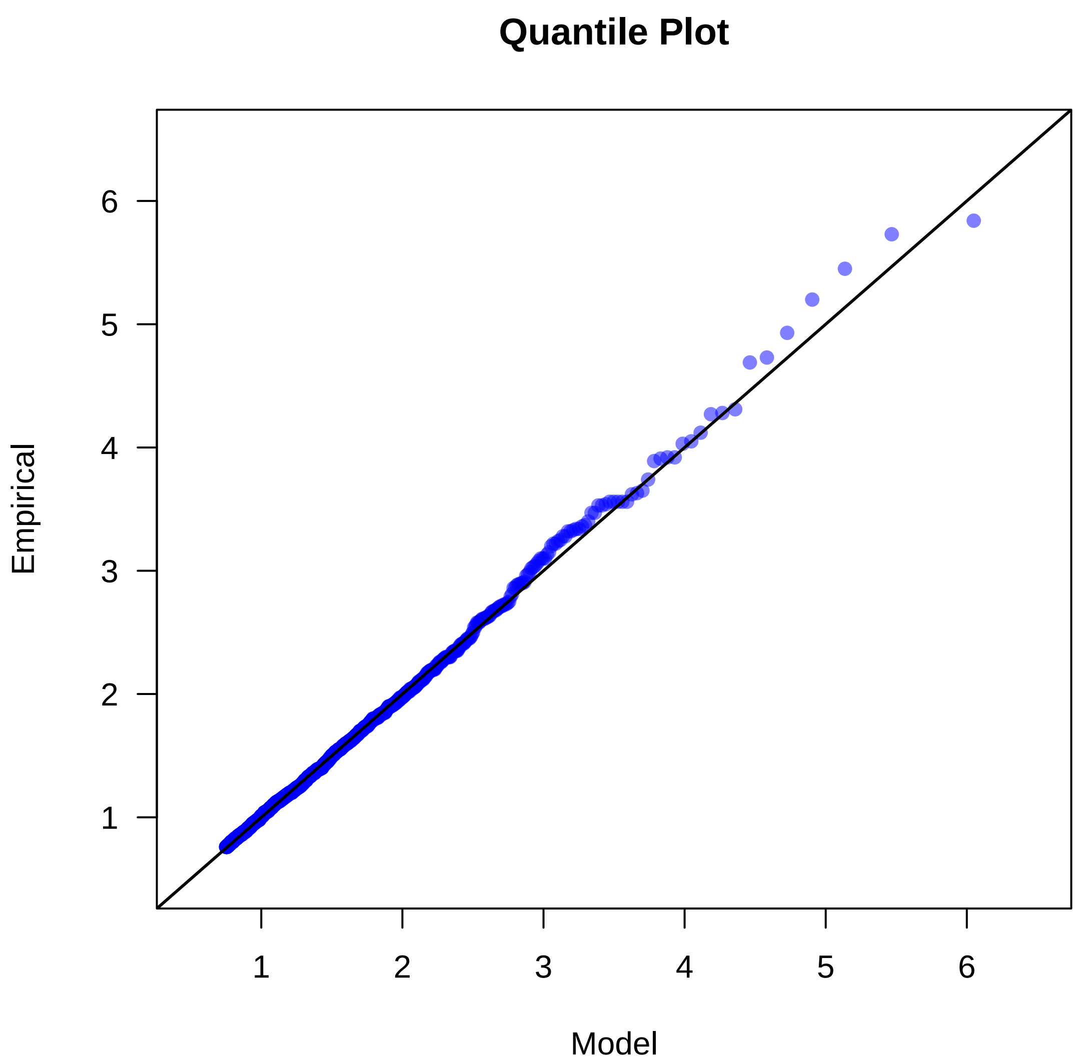
Step II: Fit a GPD to threshold excesses and assess the fit

```
# Fit a GPD for threshold exceenances 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
```

```r
# QQ plot
p <- 1:1489 / 1490
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)
```



Step III: Perform inference for return levels

Again we are interested in estimating 50-year return level

```r
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
```

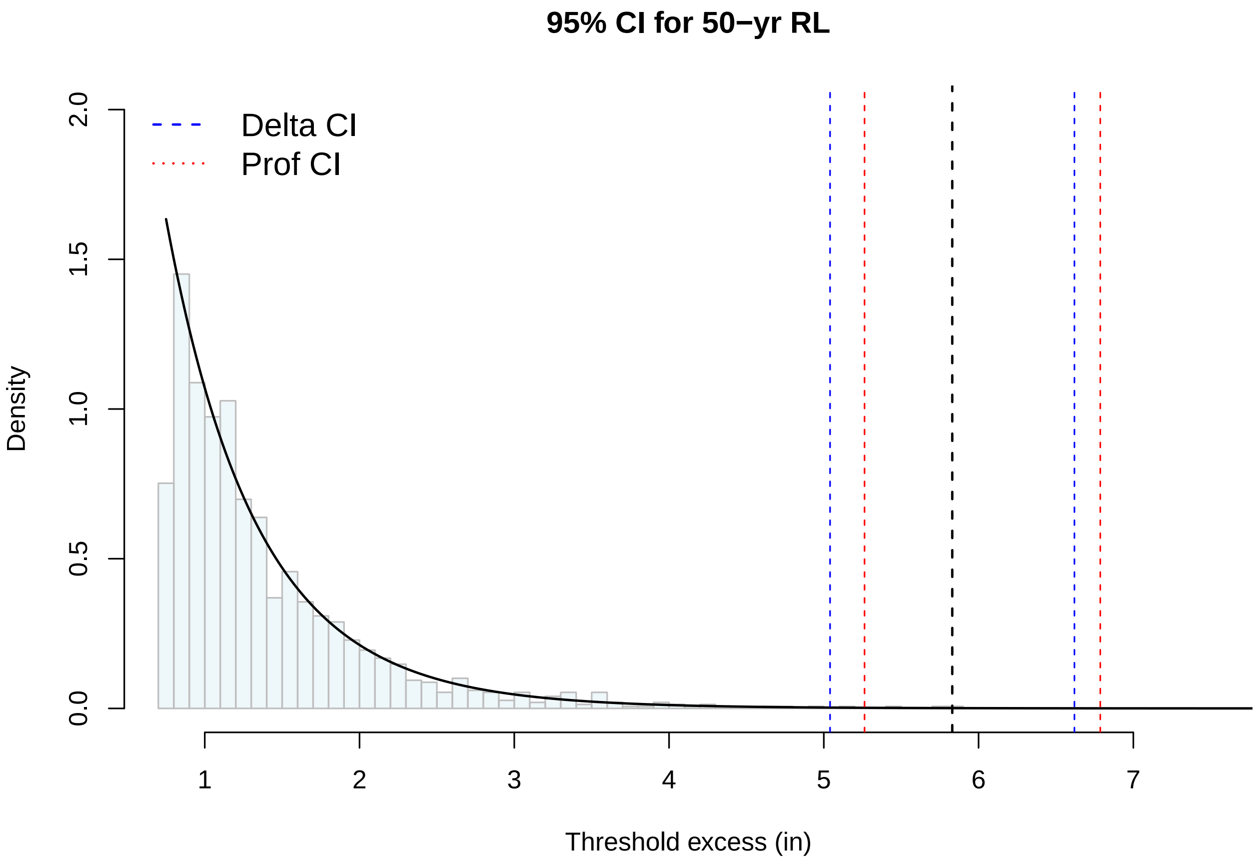
```r
CI_delta <- ci(gpdfit1, return.period = 50 * 365 / 365.25,
               verbose = F)
CI_delta
```

```
## 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)"
```

```r
CI_prof <- ci(gpdfit1, 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)"
```

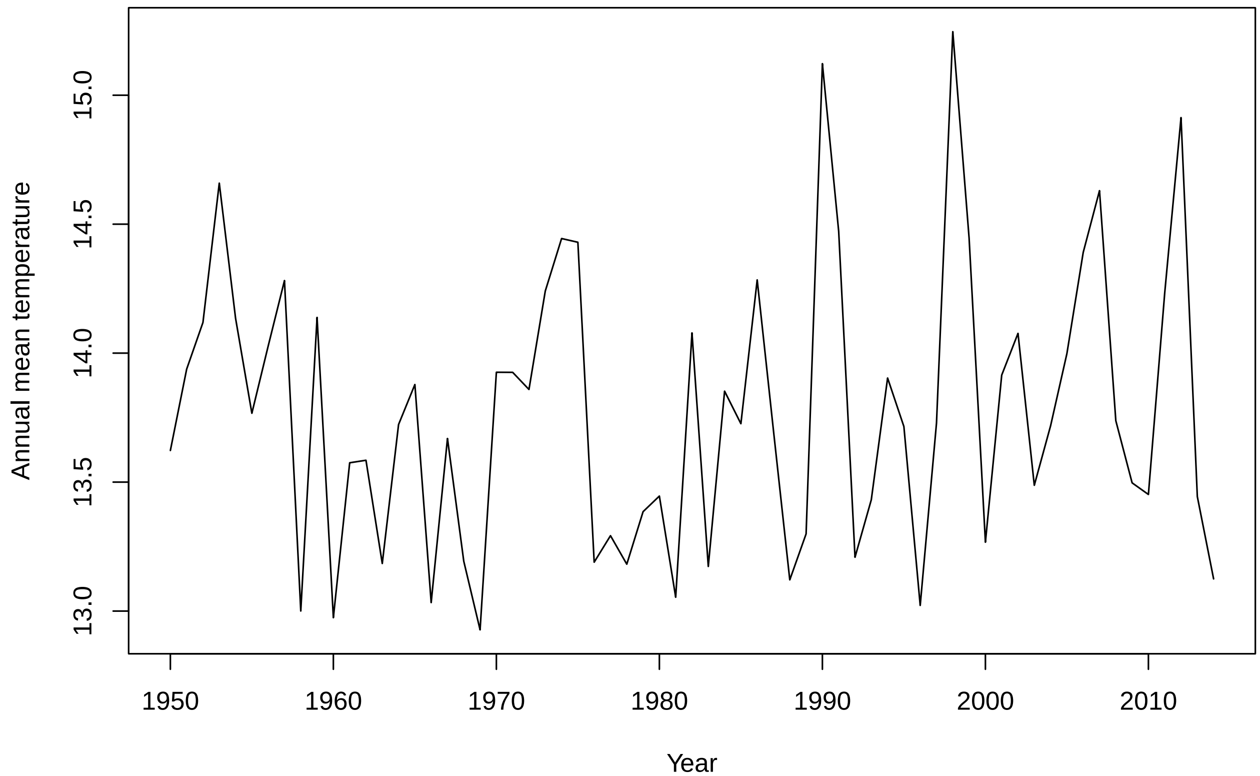
```r
hist(ex, 40, col = alpha("lightblue", 0.2), border = "gray",
     xlim = c(thres, 7.5), prob = T, ylim = c(0, 2),
     xlab = "Threshold excess (in)",
     main = "95% CI for 50-yr RL")
xg <- seq(thres, 10, len = 1000)
mle <- gpdfit1$results$par
lines(xg, gpd.dens(mle, thres, xg), lwd = 1.5)
for (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)
```



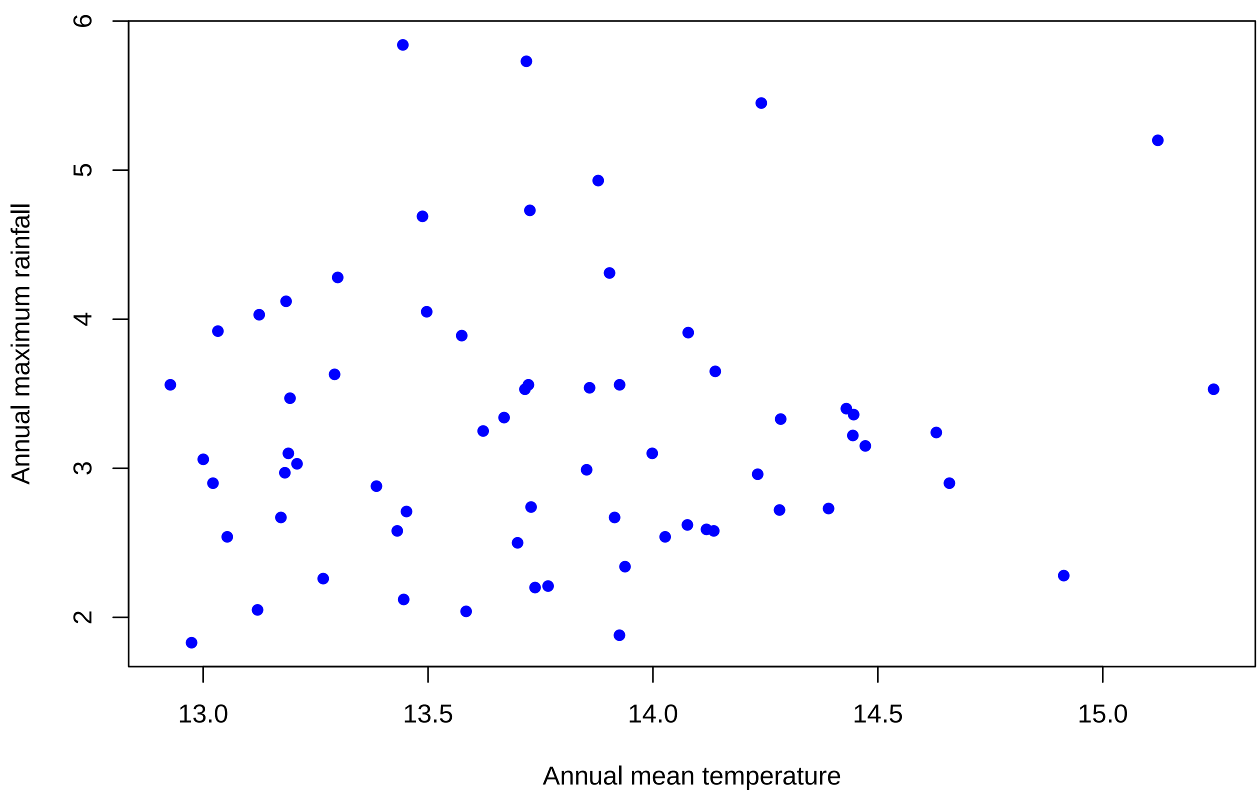
Non-stationary GEV Fitting

Here, I perform non-stationary GEV modeling of annual maximum rainfall using annual mean temperature data from the Clemson grid cell, computed from the NCEP reanalysis data product, as the covariate. I model the GEV location parameter μ as a linear function of annual mean temperature. A likelihood ratio test is used for comparison with the stationary GEV, which does not include any covariates.

```
load("SC_-82.5_35_NCEP.RData")
annmeanT <- unlist(lapply(SC_t, mean))
plot(1950:2014, annmeanT, type = "l",
     xlab = "Year", ylab = "Annual mean temperature")
```




```r
plot(annmeanT, annmx, pch = 16, col = "blue",
     xlab = "Annual mean temperature",
     ylab = "Annual maximum rainfall")
```



```r
gevfit1 <- fevd(annmx, location.fun = ~ annmeanT, scale.fun = ~ annmeanT, use.phi = T)
gevfit1
```

```
##
## fevd(x = annmx, location.fun = ~annmeanT, scale.fun = ~annmeanT,
##     use.phi = T)
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  79.94048
##
##
##  Estimated parameters:
##         mu0         mu1        phi0        phi1       shape
##  0.83026373  0.14636606  0.87964795 -0.09093870  0.03768321
##
##  Standard Error Estimates:
##       mu0       mu1      phi0      phi1     shape
## 2.2809840 0.1648066 2.4754592 0.1798972 0.1052161
##
##  Estimated parameter covariance matrix.
##                mu0           mu1        phi0         phi1         shape
## mu0    5.202887793 -3.755756e-01  1.85314930 -0.133890183 -4.388317e-03
## mu1   -0.375575615  2.716122e-02 -0.13459480  0.009748896  2.944438e-05
## phi0   1.853149305 -1.345948e-01  6.12789843 -0.444920914  1.972351e-02
## phi1  -0.133890183  9.748896e-03 -0.44492091  0.032362986 -1.685069e-03
## shape -0.004388317  2.944438e-05  0.01972351 -0.001685069  1.107042e-02
##
##  AIC = 169.881
##
##  BIC = 180.7529
```

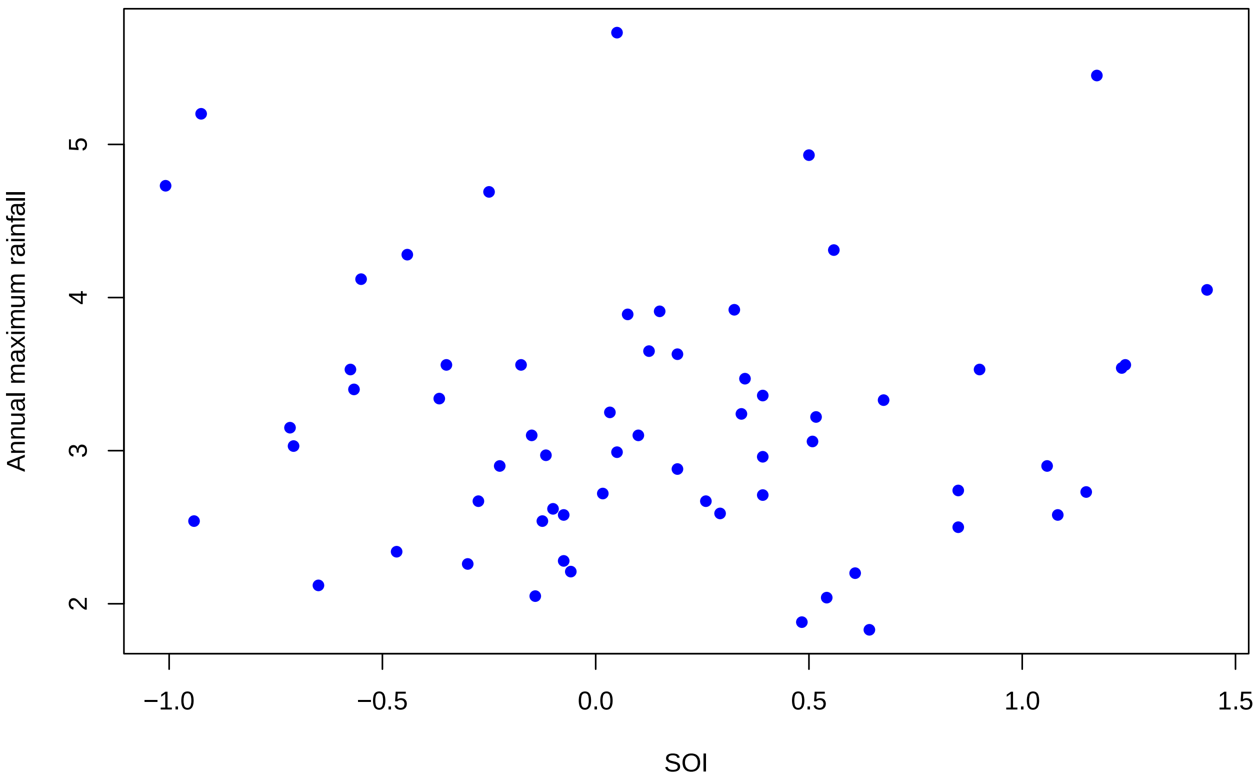
```r
lr.test(gevfit1, gevfit)
```

```
##
##  Likelihood-ratio Test
##
## data:  annmxannmx
## Likelihood-ratio = 1.3287, chi-square critical value = 5.9915, alpha =
## 0.0500, Degrees of Freedom = 2.0000, p-value = 0.5146
## alternative hypothesis: greater
```

I also include the SOI index as a covariate and model the location parameter of GEV as a linear function of the covariate. To facilitate model comparison with the stationary GEV, which does not include any covariates, I conduct a likelihood ratio test.

```r
library(rsoi)
enso <- download_enso()
order <- order(enso$Date)
ENSO <- enso[order,]
## 1952 -2019
```

```
soi_temp <- ENSO$SOI[12:827]
SOI_mon <- array(soi_temp, dim = c(12, 68))
SOI_yr <- apply(SOI_mon, 2, mean)

plot(SOI_yr[1:63], annmx[1:63], pch = 16, col = "blue", xlab = "SOI",
     ylab = "Annual maximum rainfall")
```




```
gevfit2 <- fevd(annmx[1:63], location.fun = ~ SOI_yr[1:63], scale.fun = ~ SOI_yr[1:63], use.phi = T)
gevfit2
```

```
##
## fevd(x = annmx[1:63], location.fun = ~SOI_yr[1:63], scale.fun = ~SOI_yr[1:63],
##     use.phi = T)
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  75.09131
##
##
##  Estimated parameters:
##          mu0          mu1         phi0          phi1         shape
##  2.834847198 -0.037061658 -0.383112674 -0.005096873 -0.008379923
##
##  Standard Error Estimates:
##       mu0       mu1      phi0      phi1     shape
## 0.1006719 0.1660822 0.1099295 0.1620517 0.1017207
##
```

```
##  Estimated parameter covariance matrix.
##                 mu0           mu1         phi0          phi1         shape
## mu0    0.0101348262 -0.0039321697  0.004133988 -0.0009320272 -0.0037509020
## mu1   -0.0039321697  0.0275833127  0.000355157  0.0000992694 -0.0005955417
## phi0   0.0041339879  0.0003551570  0.012084484 -0.0054053162 -0.0040754360
## phi1  -0.0009320272  0.0000992694 -0.005405316  0.0262607577  0.0023656379
## shape -0.0037509020 -0.0005955417 -0.004075436  0.0023656379  0.0103471050
##
##  AIC = 160.1826
##
##  BIC = 170.8983
```

```
lr.test(gevfit2, gevfit)
```

```
##
##  Likelihood-ratio Test
##
## data:  annmxannmx[1:63]
## Likelihood-ratio = 11.027, chi-square critical value = 5.9915, alpha =
## 0.0500, Degrees of Freedom = 2.0000, p-value = 0.004032
## alternative hypothesis: greater
```