Example

Whitney Huang

April 29, 2021

Contents

Load the daily values of wind speed and precitation

```
load("NW_pr_w.RData")
```

Define seasons

```
DJF = c(1:59, 335:365)
MAM = 60:151
JJA = 152:243
SON = 244:334
```

Split the data by season

```
pr_DJF <- nw_pr[DJF,,]; pr_MAM <- nw_pr[MAM,,]
pr_JJA <- nw_pr[JJA,,]; pr_SON <- nw_pr[SON,,]
w_DJF <- nw_wind[DJF,,]; w_MAM <- nw_wind[MAM,,]
w_JJA <- nw_wind[JJA,,]; w_SON <- nw_wind[SON,,]
```

Conditional Percentiles

```
p <- seq(0.5, 0.9, 0.1)
```

Compute the unconditional seasonal percentiles for wind speed

```
q_w <- array(dim = c(5, 4))
q_w[, 1] = quantile(w_DJF, probs = p)
q_w[, 2] = quantile(w_MAM, probs = p)
q_w[, 3] = quantile(w_JJA, probs = p)
q_w[, 4] = quantile(w_SON, probs = p)
```

Extract seasonal annual maxima and their timings

```
annMx_pr_season <- annMx_pr_season_t <- array(dim = c(50, 35, 4))
annMx_w_season <- annMx_w_season_t <- array(dim = c(50, 35, 4))
for (i in 1:4){
  annMx_pr_season[,, 1] = apply(pr_DJF, 2:3, max)
  annMx_pr_season[,, 2] = apply(pr_MAM, 2:3, max)
  annMx_pr_season[,, 3] = apply(pr_JJA, 2:3, max)
  annMx_pr_season[,, 4] = apply(pr_SON, 2:3, max)
  annMx_pr_season_t[,, 1] = apply(pr_DJF, 2:3, which.max)
  annMx_pr_season_t[,, 2] = apply(pr_MAM, 2:3, which.max)
  annMx_pr_season_t[,, 3] = apply(pr_JJA, 2:3, which.max)
  annMx_pr_season_t[,, 4] = apply(pr_SON, 2:3, which.max)
  annMx_w_season[,, 1] = apply(w_DJF, 2:3, max)
  annMx_w_season[,, 2] = apply(w_MAM, 2:3, max)
  annMx_w_season[,, 3] = apply(w_JJA, 2:3, max)
  annMx_w_season[,, 4] = apply(w_SON, 2:3, max)
  annMx_w_season_t[,, 1] = apply(w_DJF, 2:3, which.max)
  annMx_w_season_t[,, 2] = apply(w_MAM, 2:3, which.max)
  annMx_w_season_t[,, 3] = apply(w_JJA, 2:3, which.max)
  annMx_w_season_t[,, 4] = apply(w_SON, 2:3, which.max)
}
```

Extract concomitants of annual maxima

```
pr_wMx_season <- w_prMx_season <- array(dim = c(50, 35, 4))
for (j in 1:50){
  for (k in 1:35){
    pr_wMx_season[j, k, 1] <- pr_DJF[annMx_w_season_t[j, k, 1], j, k]
    pr_wMx_season[j, k, 2] <- pr_MAM[annMx_w_season_t[j, k, 2], j, k]
    pr_wMx_season[j, k, 3] <- pr_JJA[annMx_w_season_t[j, k, 3], j, k]
    pr_wMx_season[j, k, 4] <- pr_SON[annMx_w_season_t[j, k, 4], j, k]
    w_prMx_season[j, k, 1] <- w_DJF[annMx_pr_season_t[j, k, 1], j, k]
    w_prMx_season[j, k, 2] <- w_MAM[annMx_pr_season_t[j, k, 2], j, k]
```

```
    w_prMx_season[j, k, 3] <- w_JJA[annMx_pr_season_t[j, k, 3], j, k]
    w_prMx_season[j, k, 4] <- w_SON[annMx_pr_season_t[j, k, 4], j, k]
  }
}
```

GEV Fittings

```
library(ismev)
```

```
## Loading required package: mgcv
```

```
## Loading required package: nlme
```

```
## This is mgcv 1.8-33. For overview type 'help("mgcv-package")'.
```

```
gevfitPrMLE <- apply(annMx_pr_season, 3, function(x) gev.fit(c(x), show = F)$mle)
gevfitPrSE <- apply(annMx_pr_season, 3, function(x) gev.fit(c(x), show = F)$se)
gevfitWMLE <- apply(annMx_w_season, 3, function(x) gev.fit(c(x), show = F)$mle)
gevfitWrSE <- apply(annMx_w_season, 3, function(x) gev.fit(c(x), show = F)$se)
```

Estimating return levels

```
RP <- c(2, 5, 10, 20, 50)
RL_Pr <- apply(gevfitPrMLE, 2, function(x) gevq(a = x, p = 1 / RP))
RL_W <- apply(gevfitWMLE, 2, function(x) gevq(a = x, p = 1 / RP))
```

Constrcuting CI for 50-yr return level

```
library(extRemes)
```

```
## Loading required package: Lmoments
```

```
## Loading required package: distillery
```

```
##
## Attaching package: 'extRemes'
```

```
## The following objects are masked from 'package:stats':
##
##     qqnorm, qqplot
```

```r
CI_50_prof <- array(dim = c(4, 2))
dat <- annMx_pr_season
RL50 <- RL_Pr[5,]

for (j in 1:4){
  fit <- fevd(c(dat[,, j]))
  CI_50_prof[j,] <- ci(fit, method = "proflik", xrange = c(RL50[j] * 0.8, RL50[j] * 1.2),
                       return.period = 50, verbose = F)[c(1, 3)]
}
```

Fitting conditional upper quantiles

```r
source("FitMCQRNN.R")
source("FitCEV.R")
```

```
## Loading required package: mvtnorm

## Loading required package: ggplot2

## Registered S3 methods overwritten by 'texmex':
##   method            from
##   plot.declustered  extRemes
##   print.declustered extRemes

##
## Attaching package: 'texmex'

## The following object is masked from 'package:ismev':
##
##     gpd.prof

##
## Attaching package: 'evd'

## The following objects are masked from 'package:texmex':
##
##     dgev, dgpd, dgumbel, pgev, pgpd, pgumbel, portpirie, qgev, qgpd,
##     qgumbel, rgev, rgpd, rgumbel, rl

## The following objects are masked from 'package:extRemes':
##
##     fbvpot, mrlplot

##
## Attaching package: 'rmutil'

## The following object is masked from 'package:stats':
##
##     nobs
```

```
## The following objects are masked from 'package:base':
##
##     as.data.frame, units
```

```
## X grids from 0.6 precentile to max
xg_pr <- apply(annMx_pr_season, 3,
               function(x) seq(quantile(c(x), 0.6), max(c(x)), len = 1000))

CEV_Fit_w_prMx_season <- array(dim = c(5, 1000, 4))
for (j in 1:4){
  xg <- xg_pr[, j]
  CEV_Fit_w_prMx_season[,, j] <-
    CEVFit(cbind(c(w_prMx_season[,, j]),
                 c(annMx_pr_season[,, j])),
           x_pred = xg, p = p, type = 8)$y
}

MCQRNN_Fit_w_prMx <- array(dim = c(1000, 5, 4))
for (j in 1:4){
  xg <- xg_pr[, j]
  MCQRNN_Fit_w_prMx[,, j] <- FitMCQRNN(annMx_pr_season[,, j], w_prMx_season[,, j], x_Pred = xg)$pred
}
```

```
## tau = 0.5 0.6 0.7 0.8 0.9
## 1/1
## 1 0.3014148
## * 0.3014148
##
## tau = 0.5 0.6 0.7 0.8 0.9
## 1/1
## 1 0.3075451
## * 0.3075451
##
## tau = 0.5 0.6 0.7 0.8 0.9
## 1/1
## 1 0.2729226
## * 0.2729226
##
## tau = 0.5 0.6 0.7 0.8 0.9
## 1/1
## 1 0.2937563
## * 0.2937563
```

```
col = rev(rainbow(6)[1:5])
par(mar = c(3.5, 3.5, 1.5, 0.6))
plot(c(annMx_pr_season[,, 1]),
     c(w_prMx_season[,, 1]),
     pch = 16, col = "gray", cex = 0.5,
     xlab = "", ylab = "", main = "", las = 1)
mtext("Wind speed (m/s)", line = 2, side = 2)
abline(v = RL_Pr[5, 1])
rug(CI_50_prof[1,], side = 3, lwd = 1.2, ticksize = 0.03)
mtext(expression(RL50), line = 0, at = RL_Pr[5, 1])
```

```
for (i in 1:5){
  lines(xg_pr[, 1], MCQRNN_Fit_w_prMx[, i, 1],
        col = col[i], lty = 1, lwd = 1.5)
  lines(xg_pr[, 1], CEV_Fit_w_prMx_season[i,, 1],
        col = col[i], lty = 2, lwd = 1.5)
  abline(h = q_w[i, 1], col = alpha(col[i], 0.65), lty = 1, lwd = 1)
}
legend("topleft",
       legend = rev(c(seq(0.5, 0.9, 0.1))), col = rev(col), lty = 1, bty = "n",
       cex = 0.8, title = expression(tau))
legend(min(annMx_pr_season[,, 1] + 8), max(w_prMx_season[,, 1] + 0.25),
       legend = c("MCQRNN", "CEV", "Uncond"), bty = "n", lty = c(1, 2, 1),
       lwd = c(1.5, 1.5, 1))
mtext("DJF max daily precip (mm)", line = 2, side = 1)
```