

# Einfuehrung\_Nextreme

Dr.-Ing. Bora Shehu

## Einfuehrung

In diesem Beispiel werden praktische Uebungen in R gezeigt, wie das Nextreme-Paket, welches die Inhalte des Arbeitsblattes DWA-A 531 umsetzt, verwendet werden kann. Es werden im Beispiel die 5-minuetigen Niederschlagsdaten der Wetterstation Goerlitz verwendet, die frei im Climate Data Center (CDC) des Deutschen Wetterdienstes (DWD) verfuegbar sind.

Ein Ueberblick ueber die enthaltenen Ziele und Schritte ist im Folgenden aufgefuehrt:

1. **Datenerfassung:** Format der 5-min-Niederschlagsreihen
2. **Datenaufbereitung:** Bildung von jaerlichen Serien fuer bestimmte Dauer
3. **Extremwertstatistische Auswertung:** Schaetzung von Parametern, Wiederkehrperioden und Designregenfaellen
4. **Vergleiche zwischen lokaler Extremwertstatistik und KOSTRA-DWD-Werten**

## Benoetigte Software

Zunaechst ist es erforderlich, R auf dem Rechner zu installieren. Es ist empfohlen, auch R Studio zu installieren. Das ist eine Programmierumgebung (IDE), die das Arbeiten mit R erleichtert. Ein Tutorial zur Installation beider Programme (fuer alle gaengigen Betriebssysteme) stehen hier zu Verfuegung.

Nach der Installation der beiden Programme ein neues R-Skript oeffnen (File > New File > R Script). In diesem Dokument (praktisch eine Textdatei) wird nun nach und nach Programmiercode hinzugefuegt, um die Ergebnisse der lokalen Extremwertanalyse, die fuer die Station Goerlitz durchgefuehrt wurde, zu analysieren und grafisch darzustellen. Die angegebenen Codebeispiele koennen in das geoeffnete R-Skript eingefuegt und gespeichert werden.

Als erstes sollten Sie sicherstellen, dass alle von Nextreme verwendeten Pakete in Ihrem Rstudio / R installiert sind:

```
install.packages("lmomco")
install.packages("lubridate")
install.packages("terra")
install.packages("evd")
install.packages("scales")
```

Als naechstes ist es erforderlich, dass fuer das Arbeitsblatt DWA-A 531 entwickelte R-Paket „Nextreme“ zu installieren und einzubinden.

- a) Das Paket kann aus der bereitgestellten Archiv-Datei (Nextreme\_1.2.1.tar.gz) von <https://github.com/DWA-A-531/Nextreme/releases/tag/v1.2.1> mit folgenden Schritten installiert werden: Tools > Install Packages > Install from: Package Archive File (.zip; .tar.gz) > Browse to the package location > Install.
- b) Das Paket kann auch von installiert werden:

```
install.packages("pak")
pak::pak("DWA-A-531/Nextreme")
```

Um die notwendigen Pakete in R-Studio zu laden, verwenden Sie den folgenden Befehl:

```
library("lubridate")
#>
#> Attache Paket: 'lubridate'
#> Die folgenden Objekte sind maskiert von 'package:base':
#>
#>      date, intersect, setdiff, union
library("lmomco")
#> # Package lmomco (2.5.1) loaded
library("terra")
#> terra 1.7.83
library("Nextreme")
```

## Anwendungsbeispiele

### 1. Datenerfassung: Format der 5-min-Niederschlagsreihen

Die hier beschriebenen Beispiele basieren auf der 5min-Zeitreihe der Station Goerlitz, die vom Deutschen Wetterdienst zur Verfügung steht. Diese Zeitreihe ist bereits im Paket *Nextreme* enthalten und kann mit dem folgenden Befehl aufgerufen werden:

```
data("Regendaten_01684")
# die ersten 10 Zeilen der Regendaten_01684 zeigen
head(Regendaten_01684, 10)
#>      Datum RH
#> 1 1991-01-01 00:00:00 NA
#> 2 1991-01-01 00:05:00 NA
#> 3 1991-01-01 00:10:00 NA
#> 4 1991-01-01 00:15:00 NA
#> 5 1991-01-01 00:20:00 NA
#> 6 1991-01-01 00:25:00 NA
#> 7 1991-01-01 00:30:00 NA
#> 8 1991-01-01 00:35:00 NA
#> 9 1991-01-01 00:40:00 NA
#> 10 1991-01-01 00:45:00 NA
# die letzten 10 Zeilen der Regendaten_01684 zeigen
tail(Regendaten_01684, 10)
#>      Datum RH
#> 3155895 2020-12-31 23:10:00 0
#> 3155896 2020-12-31 23:15:00 0
```

```
#> 3155897 2020-12-31 23:20:00 0
#> 3155898 2020-12-31 23:25:00 0
#> 3155899 2020-12-31 23:30:00 0
#> 3155900 2020-12-31 23:35:00 0
#> 3155901 2020-12-31 23:40:00 0
#> 3155902 2020-12-31 23:45:00 0
#> 3155903 2020-12-31 23:50:00 0
#> 3155904 2020-12-31 23:55:00 0
```

Zu beachten ist, dass die 5-Minuten-Zeitreihe ein `data.frame()`-Format mit zwei Spalten sein sollte: *Datum* und *RH*. *Datum* - ist das angegebene Datum als `as.POSIXct()` date-type, *RH* - ist die gemessene Regenhoehe bei jedem 5min-Zeitschritt in mm/5min. Es ist darauf zu achten, dass die Datumsdaten kontinuierlich sein sollten und kein Datum fehlen sollte. Falls ein Datum fehlt, muss es in die Zeitreihe aufgenommen und der entsprechende *RH* als fehlender Wert gesetzt werden. Alle fehlenden Werte fuer *RH* sollten als *NA* angegeben werden. Vor der Verwendung ist sicherzustellen, dass die *RH*-Spalten keine negativen Werte enthalten und die Werte plausibel sind (keine sehr hohen Werte).

```
# Ueberpruefung der Struktur der 5-Minuten-Zeitreihe
str(Regendaten_01684)
#> 'data.frame': 3155904 obs. of 2 variables:
#> $ Datum: POSIXct, format: "1991-01-01 00:00:00" "1991-01-01 00:05:00" ...
#> $ RH : num NA NA NA NA NA NA NA NA NA NA ...
# Ueberpruefung, ob die Datumdaten kontinuierlich sind
unique(diff(Regendaten_01684$Datum))
#> [1] 5
# Ueberpruefung des Wertebereichs der RH
range(Regendaten_01684$RH, na.rm=T)
#> [1] 0.000 15.938
```

## 2. Datenaufbereitung: Bildung von jaerlichen Serien fuer bestimmte Dauer (von 5 Minuten bis 7 Tagen)

Fuer die 5-Minuten-Niederschlagsreihen in Goerlitz werden fuer jede Dauer jaehrliche Maxima-Serien abgeleitet. Dies erfolgt hier nach den im Arbeitsblatt DWA-A 531 beschriebenen Schritten, die aus den folgenden beschrieben sind.

### 2.1 Ermittlung von Regenhoeihen bestimmter Dauer aus den Grunddaten

Ziel ist die Berechnung der maximalen jaehrlichen Maxima-Serien, die bei bestimmten Dauern (hier 5, 10, 15, 30, 60, 120, 360, 720, 1440, 2880, 4320, 10080 Minuten) beobachtet wurden. Zu diesem Zweck kann die Funktion `jaehrliche_maxSerie()` verwendet werden. Hier wird eine Dauer (*DSDmin*) von 4 Stunden verwendet, um unabhengige Regenereignisse zu trennen, und da die Regendaten in 5-Minuten-Zeitschritten gegeben sind, wird das Intervallargument (*Intervall*) auf 5 gesetzt. Um die Parameter der Extremwertverteilung zu ermitteln, sind die maximalen jaehrlichen Regensintensitaeten fuer jede Dauer erforderlich. Aus diesem Grund geben wir `SerieTyp=„INT“` an, um die maximalen Intensitaeten als Ausgabe zu erhalten.

```
Goerlitz_maxIntSerie = jaehrliche_maxSerie(Regendaten_01684,
                                           Dauern = c(5, 10, 15, 30, 60, 120, 360, 720,
                                                         1440, 2880, 4320, 10080),
                                           DSDmin=240, Intervall = 5,
                                           SerieTyp = "INT")
```

```
print(head(Goerlitz_maxIntSerie,10))
#>           5          10          15          30          60          120          360          720          1440          2880
#> 1991  59.304  46.644  38.524  21.626  12.601   6.326  3.695  2.576  1.823  0.914
#> 1992  37.464  37.170  27.596  16.270  10.300   9.343  5.463  2.745  1.401  0.815
#> 1993  56.400  40.800  45.600  28.200  21.900  11.850  6.169  4.512  3.618  1.890
#> 1994  76.800  52.200  40.000  23.148  18.258  17.061  8.084  4.192  2.104  1.096
#> 1995  82.800  81.600  62.800  36.200  19.600   9.900  5.438  3.793  2.383  1.240
#> 1996  74.400  42.600  30.800  16.400  10.400   7.700  4.083  2.767  2.404  1.219
#> 1997  84.000  65.394  52.316  27.820  15.986   8.253  4.167  2.725  2.092  1.071
#> 1998  60.000  48.000  38.800  26.000  14.500   9.000  4.233  2.292  1.675  0.927
#> 1999 110.400  75.600  60.400  40.800  22.700  11.350  3.783  2.142  1.358  0.773
#> 2000 191.256 118.338  88.852  50.998  25.499  12.799  5.333  3.042  1.708  1.104
#>      4320 10080
#> 1991  0.622  0.301
#> 1992  0.561  0.270
#> 1993  1.300  0.829
#> 1994  0.736  0.349
#> 1995  0.865  0.561
#> 1996  0.812  0.571
#> 1997  0.807  0.452
#> 1998  0.701  0.564
#> 1999  0.581  0.295
#> 2000  0.736  0.401
```

Die Variable *Goerlitz\_maxIntSerie* ist eine Tabelle mit der maximalen Intensitaet (mm/h), die bei verschiedenen Dauern (in Spalten) und Jahren (in Zeilen) ermittelt wurde.

## 2.2 Beruecksichtigung der Intervalllaenge

Da die Niederschlagsreihe in 5-Minuten-Blocken verfuegbar ist, werden die jaehrliche Maxima-Serien fuer die Dauern von 5, 10, 15 und 20 Minuten korrigiert (Intervallkorrektur, siehe Tabelle 1 in DWA-A 531). Zu diesem Zweck kann die Funktion *Intervallkorrektur()* verwendet werden. Da die Regendaten in 5-Minuten-Zeitschritten gegeben sind, wird das Intervallargument (*Intervall*) auf 5 gesetzt und damit Dauerstufen bis 20 min bei der Korrektur beruecksichtigt.

```
Goerlitz_maxIntSerie = Intervallkorrektur(Goerlitz_maxIntSerie, Intervall = 5)
print(head(Goerlitz_maxIntSerie,10))
#>           5          10          15          30          60          120          360          720          1440
#> 1991  67.60656  49.90908  40.06496  21.626  12.601   6.326  3.695  2.576  1.823
#> 1992  42.70896  39.77190  28.69984  16.270  10.300   9.343  5.463  2.745  1.401
#> 1993  64.29600  43.65600  47.42400  28.200  21.900  11.850  6.169  4.512  3.618
#> 1994  87.55200  55.85400  41.60000  23.148  18.258  17.061  8.084  4.192  2.104
#> 1995  94.39200  87.31200  65.31200  36.200  19.600   9.900  5.438  3.793  2.383
#> 1996  84.81600  45.58200  32.03200  16.400  10.400   7.700  4.083  2.767  2.404
#> 1997  95.76000  69.97158  54.40864  27.820  15.986   8.253  4.167  2.725  2.092
#> 1998  68.40000  51.36000  40.35200  26.000  14.500   9.000  4.233  2.292  1.675
#> 1999 125.85600  80.89200  62.81600  40.800  22.700  11.350  3.783  2.142  1.358
#> 2000 218.03184 126.62166  92.40608  50.998  25.499  12.799  5.333  3.042  1.708
#>      2880  4320 10080
#> 1991  0.914  0.622  0.301
#> 1992  0.815  0.561  0.270
#> 1993  1.890  1.300  0.829
```

```
#> 1994 1.096 0.736 0.349
#> 1995 1.240 0.865 0.561
#> 1996 1.219 0.812 0.571
#> 1997 1.071 0.807 0.452
#> 1998 0.927 0.701 0.564
#> 1999 0.773 0.581 0.295
#> 2000 1.104 0.736 0.401
```

## 2.3 Stationaritaet und Sprungkorrektur bei kleinen Dauerstufen

Da der Sensortyp an der Station Goerlitz sich ueber die Zeit aendert, wird ueberprueft, ob die jaehrlichen Maximalreihen der kurzen Dauern (5, 10, 15 und 30 Minuten) eine sprunghafte Instationaritaet aufweisen, die dem Datum des Sensorwechsels entspricht. Wenn es eine solche sprunghafte Instationaritaet gibt, wird der Sprung eliminiert.

**HINWEIS:** Dieser Schritt wird hier nur durchgefuehrt, um ein Beispiel fuer eine solche Anwendung zu zeigen. Laut Arbeitsblatt DWA-A 531 sollte eine Sprungkorrektur getestet werden, wenn ein Sensor vom analogen (Regenschreiber) zum digitalen Typ gewechselt wurde. An der Station Goerlitz sind in diesem Beispiel alle Sensoren digital, so dass eine Sprungpruefung und -korrektur eigentlich nicht erforderlich waere. Hier wird dieser Schritt nur gezeigt, um einen vollstaendigen Arbeitsablauf darzulegen!

Zu diesem Zweck kann die Funktion *Sprung\_Korrektur()* verwendet werden. Zusaetzlich zur Eingabe der jaehrlichen Serie (*Goerlitz\_maxIntSerie*) benoetigt die Funktion das Datum des Sensorwechsels in einer Variablen vom Typ *Date()*.

```
WechselDatum = as.Date("1991-01-01", format=c("%Y-%m-%d"))
Goerlitz_maxSerie_korrigiert = Sprung_Korrektur(Goerlitz_maxIntSerie,
                                                WechselDatum)

# Die Differenz vor und nach der Funktion sprungKorrektur() ueberpruefen.
print(head(round(Goerlitz_maxSerie_korrigiert-Goerlitz_maxIntSerie,0),10))
#>      5 10 15 30 60 120 360 720 1440 2880 4320 10080
#> 1991 0 0 0 0 0 0 0 0 0 0 0 0
#> 1992 0 0 0 0 0 0 0 0 0 0 0 0
#> 1993 0 0 0 0 0 0 0 0 0 0 0 0
#> 1994 0 0 0 0 0 0 0 0 0 0 0 0
#> 1995 0 0 0 0 0 0 0 0 0 0 0 0
#> 1996 0 0 0 0 0 0 0 0 0 0 0 0
#> 1997 0 0 0 0 0 0 0 0 0 0 0 0
#> 1998 0 0 0 0 0 0 0 0 0 0 0 0
#> 1999 0 0 0 0 0 0 0 0 0 0 0 0
#> 2000 0 0 0 0 0 0 0 0 0 0 0 0
```

Wie bereits erwaeht, gibt es in diesem Beispiel keine Spruenge in den Daten, und daher ist die von der Funktion *Sprung\_Korrektur()* zurueckgegebene Tabelle dieselbe wie die Eingabe.

## 3. Extremwertstatistische Auswertung: Extremwertanalyse fuer Dauern von 5 Minuten bis 7 Tagen durchfuehren

Im Folgenden werden Beispiele fuer die Extremwertanalyse an der Station Goerlitz gezeigt.

### 3.1 Schaetzung der Parameter, die die Verteilung der Extremwerte ueber alle Dauern beschreiben

Ziel ist die Anpassung der Generalisierten Extremwert Verteilung (GEV) an die erhaltenen jaehrlichen Serien der maximalen Regensintensitaet fuer die Station Goerlitz. Die Koutsoyiannis-Methode, wie auf den DWA-A 531 Arbeitsblatt beschrieben, wird zur Normalisierung der Intensitaeten ueber die gegebenen Dauern verwendet.

Zu diesem Zweck kann die Funktion *Parameter\_Schaetzung()* verwendet werden, um die Extremwertparameter aus den erhaltenen jaehrlichen Maxima-Serien zu schaetzen. In der Funktion sollte die Dauer fuer die Extremwertparameter angegeben werden, und die Art der Verteilung, die angepasst werden soll. Wie in DWA-A 531 Blatt wird hier die GEV-Verteilung mit einem festen Formparameter von -0,1 (Fréchet-Typ) verwendet. Deshalb lauten die Argumente *methGEV*="GEV", *formTyp*="FIX" und *Gamma*=-0,1. Da die jaehrlichen Maximalintensitaetsreihen (mm/h) als Eingabe verwendet werden, wird der *SerieTyp* entsprechend auf "INT" gesetzt. Werden die jaehrlichen Maxima-Serien (mm/Dauer) als Eingabe verwendet, so ist *SerieTyp*="VOL".

**HINWEIS:** Die Funktion *Parameter\_Schaetzung()* kann entweder eine GEV oder eine Gumbel-Verteilung (Formparameter = 0.0) anpassen. Fuer die Gumbel-Verteilung (*methGEV*="GUM") bleiben *formTyp* und *Gamma* unberuecksichtigt; fuer die GEV kann der Formparameter hingegen auch frei aus L-Momenten geschaezt werden (mit *formTyp*="CON", in diesem Fall wird *Gamma* nicht beruecksichtigt). In allen Faellen werden die maximalen Intensitaeten ueber die gegebene Dauer normalisiert. Die Funktion *Parameter\_Schaetzung()* kann nicht GEV-Verteilungen individuell an jede Dauer anpassen!

```
N_pars      = Parameter_Schaetzung(Goerlitz_maxIntSerie,
                                   Dauern = c(5, 10, 15,30,60,120,360,720,1440,
                                              2880,4320, 10080),
                                   methGEV="GEV", formTyp = "FIX", Gamma = -0.1,
                                   SerieTyp="INT")

print(N_pars)
#>      Mu      Sigma Gamma      Theta      Eta      KW
#> 1 16.88105 5.610762 -0.1 0.06259154 0.7499916 2.412742
```

Die zurueckgegebene Variable *N\_pars* ist ein einzeliger Dataframe, der die berechneten Parameter der Extremwertverteilung des Niederschlags an der Station Goerlitz enthaelt: *Mu* - der GEV- Lokationsparameter, *Sigma* - der GEV- Skalenparameter, *Gamma* - der GEV- Formparameter, *Theta* und *Eta* jeweils der erste und zweite Koutsoyiannis- Parameter, waehrend *KW* die Kruskal-Wallis Teststatistik ist, die bei der Optimierung der Parameter erreicht wurde.

### 3.2 Schaetzung der Regenhoehe fuer angegeben Dauer und Wiederkehrintervalle

Ziel ist die Berechnung der Jaehrlichkeit (Wiederkehrintervalle) der beobachteten Regenhoehe und bestimmten Dauerstufen auf der Grundlage der berechneten Extremwertparameter.

Zu diesem Zweck kann die Funktion *Quantil\_Schaetzung()* verwendet werden. Neben den Parametern (*N\_pars*), den Dauern (*Dauern*) und Wiederkehrintervalle (*Tn*), sollte auch die GEV-Methode (*methGEV*) angegeben werden. In diesem Fall wird der *SerieTyp* auf *VOL* gesetzt, um Regenhoehe (mm/Dauer) als Output zu erhalten. Ausserdem kann die Art der Ausgabe mit dem Argument *SerieTyp* gesteuert werden (*VOL* fuer Regenhoehe in mm/Dauer oder *INT* fuer Regenintensitaet in mm/h).

```
H_quas      = Quantil_Schaetzung(N_pars,
                                   Dauern = c(5, 10, 15,30,60,120,360,720,1440,
                                              2880, 4320, 10080),
                                   Tn =c(2,5,10,20,50,100),
```

```

methGEV="GEV",SerieTyp = "VOL")
print(H_quas)
#>      5      10      15      30      60      120      360      720      1440      2880      4320
#> 2      7.38 10.52 12.50 16.09 19.97 24.30 32.46 38.76 46.08 55.20 61.20
#> 5      9.42 13.42 15.96 20.54 25.50 31.00 41.46 49.44 59.04 70.08 77.76
#> 10     11.21 15.97 18.98 24.43 30.34 36.90 49.32 58.92 70.08 83.52 92.16
#> 20     12.81 18.25 21.70 27.93 34.67 42.16 56.34 67.32 80.16 95.52 105.84
#> 50     15.41 21.96 26.11 33.60 41.72 50.74 67.80 81.00 96.48 114.72 127.44
#> 100    17.52 24.98 29.69 38.22 47.44 57.70 77.10 92.04 109.68 130.56 144.72
#>      10080
#> 2      75.60
#> 5      95.76
#> 10     114.24
#> 20     131.04
#> 50     157.92
#> 100    178.08

```

Die zurueckgegebene Variable *H\_quas* ist eine Tabelle (Dataframe-Format) mit den Wiederkehrintervallen in verschiedenen Zeilen und der Dauer in verschiedenen Spalten. Da in diesem Fall der *SerieTyp*="VOL" ist, sind die Einheiten in mm/Dauer.

### 3.3 Schaetzung der Unsicherheitsbereiche

Ziel ist die Durchfuehrung eines Bootstrapps, um die Unsicherheit der Stichprobe bei der Anpassung der Extremwertparameter an die Serie der Station Goerlitz zu beruecksichtigen.

Zu diesem Zweck kann die Funktion *Unsicherheit\_Schaetzung()* implementiert werden. Neben den bisher angegebenen typischen Informationen zur Berechnung von Parametern und Extremwerten koennen drei weitere Argumente festgelegt werden: *nBoots* - zur Angabe der gewuenschten Anzahl von Realisierungen, *rSeed* - eine Zufallszahl, um sicherzustellen, dass die Ausgabe wiederholbar ist, und *Konfidenzgrenzen* - zur Angabe der Quantils der oberen (*Ko*) und unteren Konfidenzintervallgrenzen (*Ku*). Die geschaetzten jaehrlichen Maximalintensitaetsserien koennen *nBoots*-mal mit Ersatz neu gesampelt werden, um die Stichprobenunsicherheit zu beruecksichtigen. Anschliessend koennen fuer jede *nBoots*-Realisierung die Parameter geschaetzt werden. Auf diese Weise erhaelt man *nBoots* Realisierungen der Parameter, die als Grundlage fuer die Berechnung von Konfidenzintervallen nicht nur der Parameter, sondern auch der Quantile verwendet werden koennen.

**Dies kann mehrere Minuten dauern!**

```

KI      = Unsicherheit_Schaetzung(Goerlitz_maxIntSerie,
                                Tn =c(2,5,10,20,50,100),
                                Dauern = c(5, 10, 15,30,60,120,360,720,1440,
                                           2880, 4320, 10080),
                                methGEV="GEV",
                                formTyp = "FIX",
                                Gamma=-0.1,
                                nBoots = 100,
                                rSeed = 1232,
                                SerieTyp = "VOL",
                                Konfidenzgrenzen = c(0.025,0.975))

# Zugriff auf die Parameterinformationen
PAR_KI   = KI$PAR_INFO
# Zugriff auf die Quantils Informationen
HN_KI    = KI$QUA_INFO

```

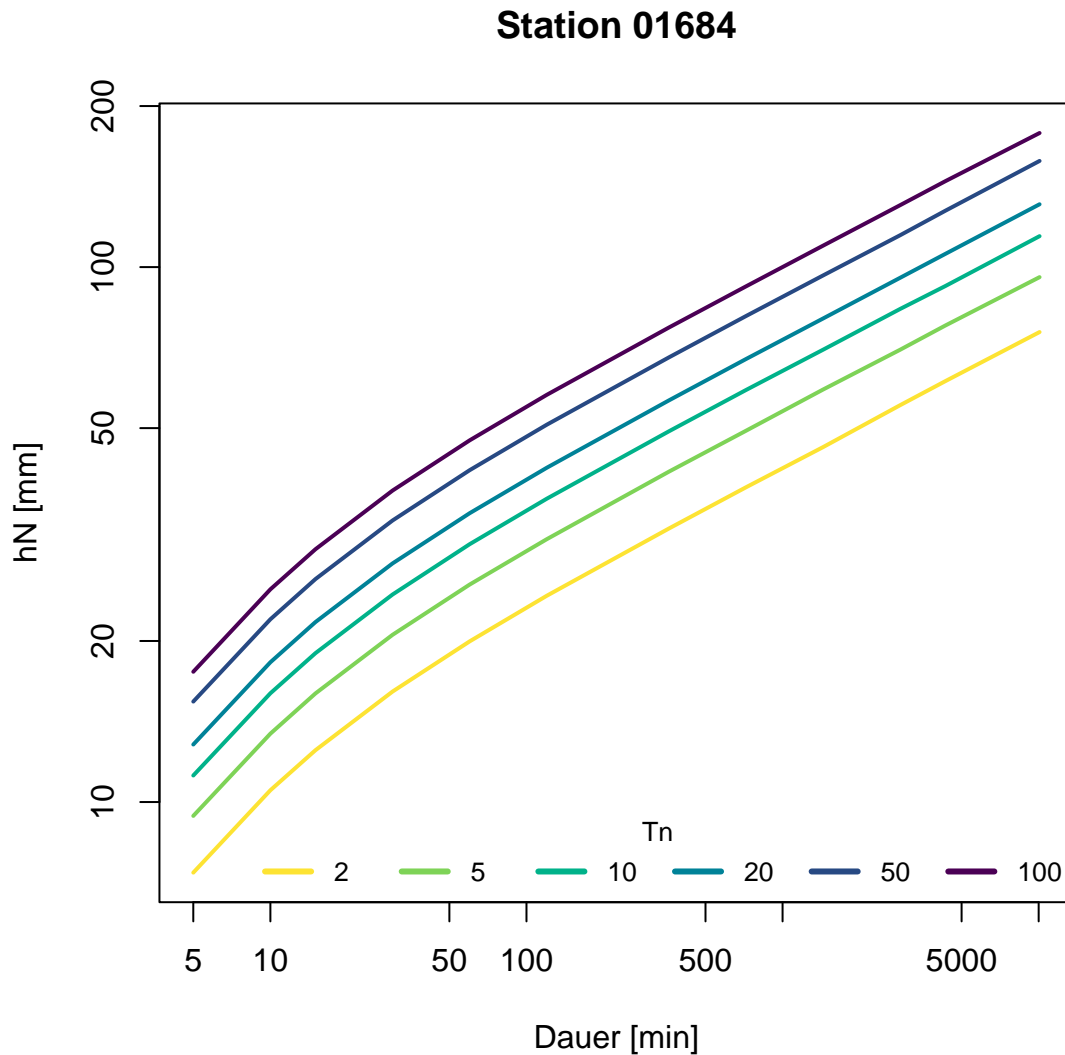
Die Funktion *Unsicherheit\_Schaetzung()* gibt eine Liste mit den Parameterinformationen *PAR\_INFO* und Quantils Informationen *QUA\_INFO* zurueck. Die beiden Eintraege in der Liste enthalten Informationen ueber die angegebenen Quantile fuer die Berechnung der Grenzen des Konfidenzintervalls, des Mittelwerts der Realisierungen und der relativen Unsicherheit  $(Ko-Ku)/K$ .

### 3.4 Darstellung der lokal geschaetzten Regenhoeen unter Beruecksichtigung des Unsicherheitsbereichs

Der folgende Code zeigt ein Beispiel fuer die Visualisierung verschiedener Regenhoeen an der Station Goerlitz, die mit verschiedenen Dauern (auf der x-Achse) und Wiederkehrintervallen (in verschiedenen Farben) einhergehen. Fuer beide Achsen wird eine logarithmische Darstellung verwendet.

```
Dauern = c(5, 10, 15,30,60,120,360,720,1440, 2880, 4320, 10080)
Tn_Farbe = rev(hcl.colors(6, palette = "viridis"))
plot(Dauern, H_quas["2",], type="l", lwd=2, lty=1, log="xy", col=Tn_Farbe[1],
     ylab="hN [mm]", ylim=range(H_quas), xlab="Dauer [min]",
     main = "Station 01684")
lines(Dauern, H_quas["5",], lwd=2, col=Tn_Farbe[2])
lines(Dauern, H_quas["10",], lwd=2, col=Tn_Farbe[3])
lines(Dauern, H_quas["20",], lwd=2, col=Tn_Farbe[4])
lines(Dauern, H_quas["50",], lwd=2, col=Tn_Farbe[5])
lines(Dauern, H_quas["100",],lwd=2, col=Tn_Farbe[6])
legend("bottomright", legend = c(2,5,10,20,50,100),lty=1, lwd=3, col=Tn_Farbe,
     cex=0.8, title="Tn", horiz=T, bty="n")
```

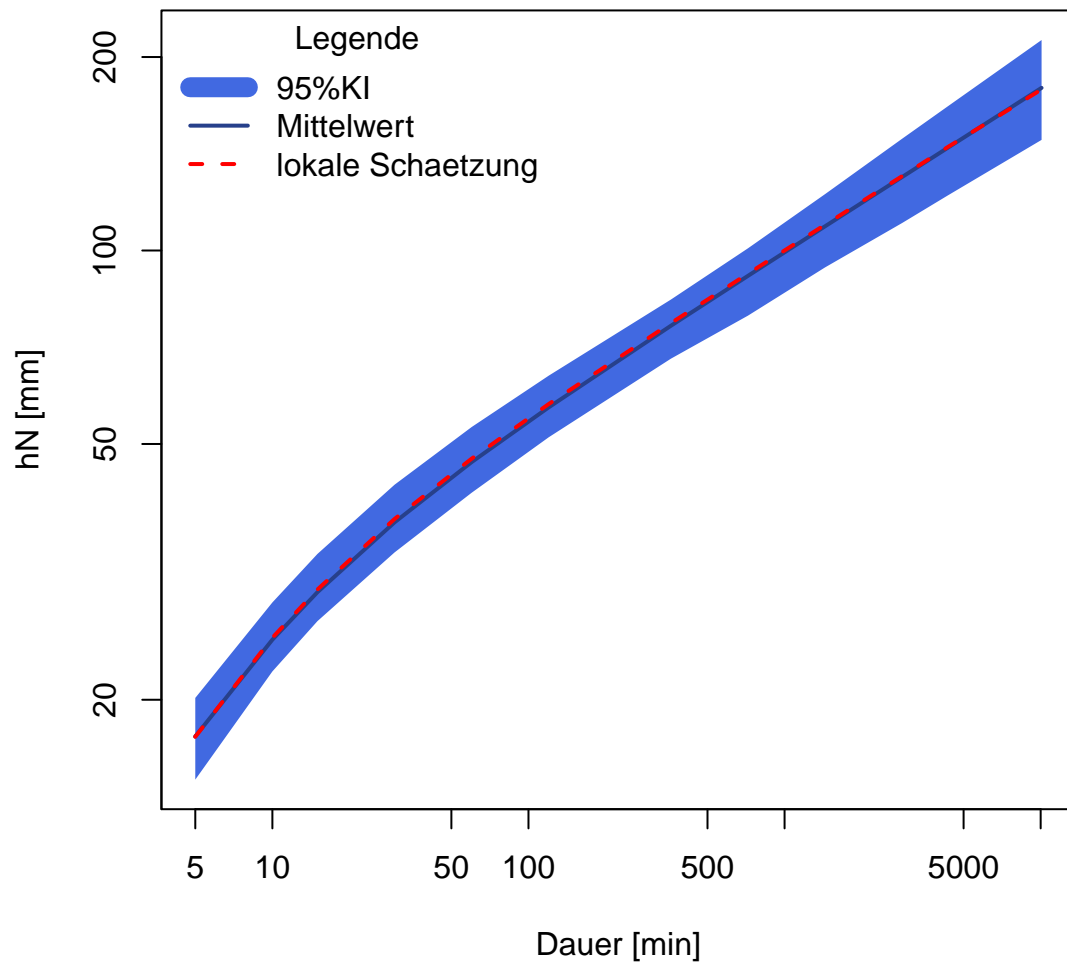




Im folgenden Beispiel wird ein Code-Teil zur Darstellung des 100-jährlichen Regenvolumens fuer die Station Goerlitz gegeben, wie es fuer die Dauer von 5 Minuten bis 7 Tagen aus der bisher durchgefuehrten Extremwertanalyse der 5-Minuten-Niederschlagsdaten des DWD berechnet wurde (in rot dargestellt). Die Konfidenzgrenze, die den 2.5 %- und 97.5 %-Quantils der 100 Realisierungen fuer die 100-jährliche Regenmenge entsprechen, sind ebenfalls in blau dargestellt.

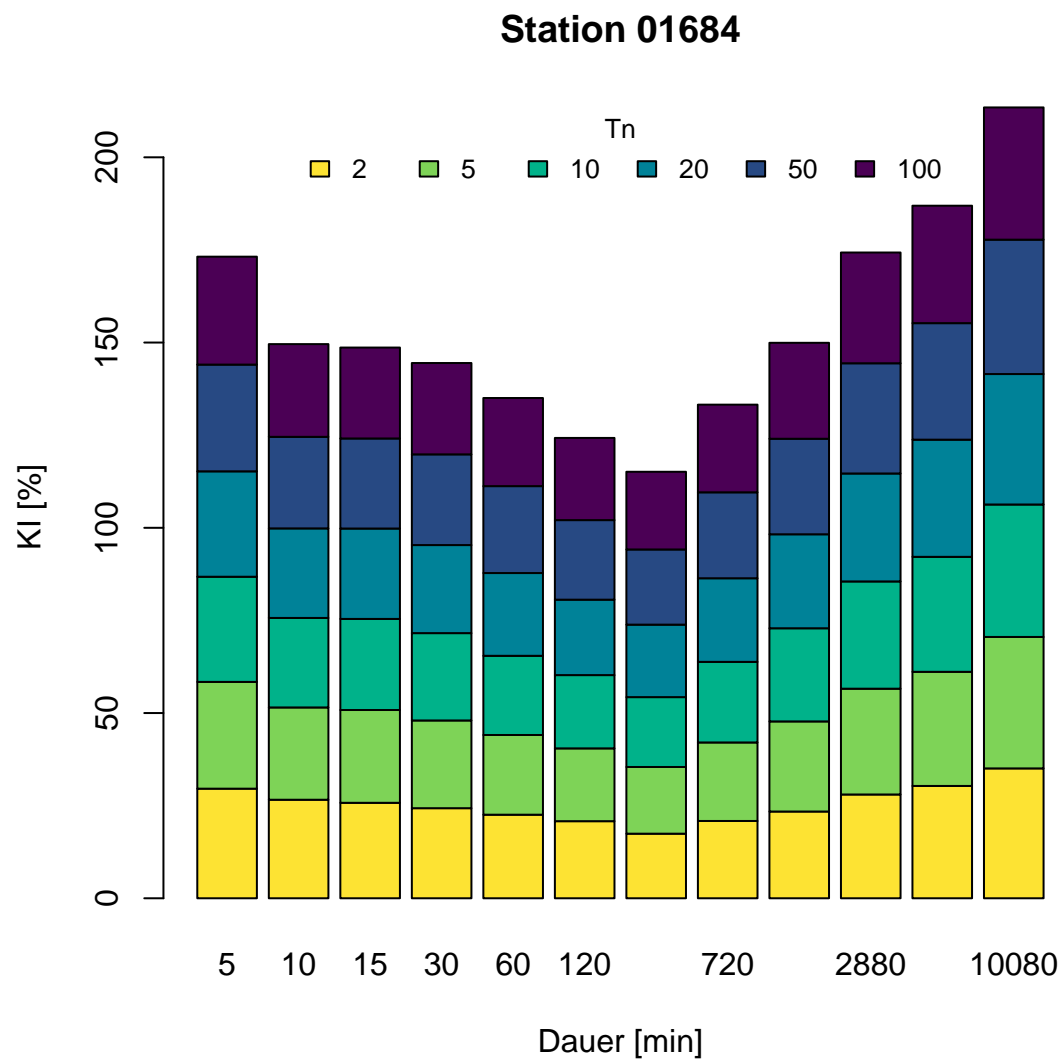
```
Dauern = c(5, 10, 15, 30, 60, 120, 360, 720, 1440, 2880, 4320, 10080)
plot(Dauern, H_quas["100",], type="l", lwd=2, lty=2, log="xy",
     ylim=range(HN_KI$`97.5%`["100",], HN_KI$`2.5%`["100",]), col="red",
     ylab="hN [mm]", xlab="Dauer [min]", main = "Station 01684")
polygon(c(Dauern, rev(Dauern)), c(HN_KI$`2.5%`["100",],
     rev(HN_KI$`97.5%`["100",])), col="royalblue", border=NA)
lines(Dauern, HN_KI$Mittelwert["100",], type="l", col="royalblue4", lwd=2)
lines(Dauern, H_quas["100",], type="l", col="red", lwd=2, lty=2)
legend("topleft", c("95%KI", "Mittelwert", "lokale Schaetzung"),
     col=c("royalblue", "royalblue4", "red"), lty=c(1, 1, 2), lwd=c(10, 2, 2),
     title = "Legende", bty="n")
```

## Station 01684

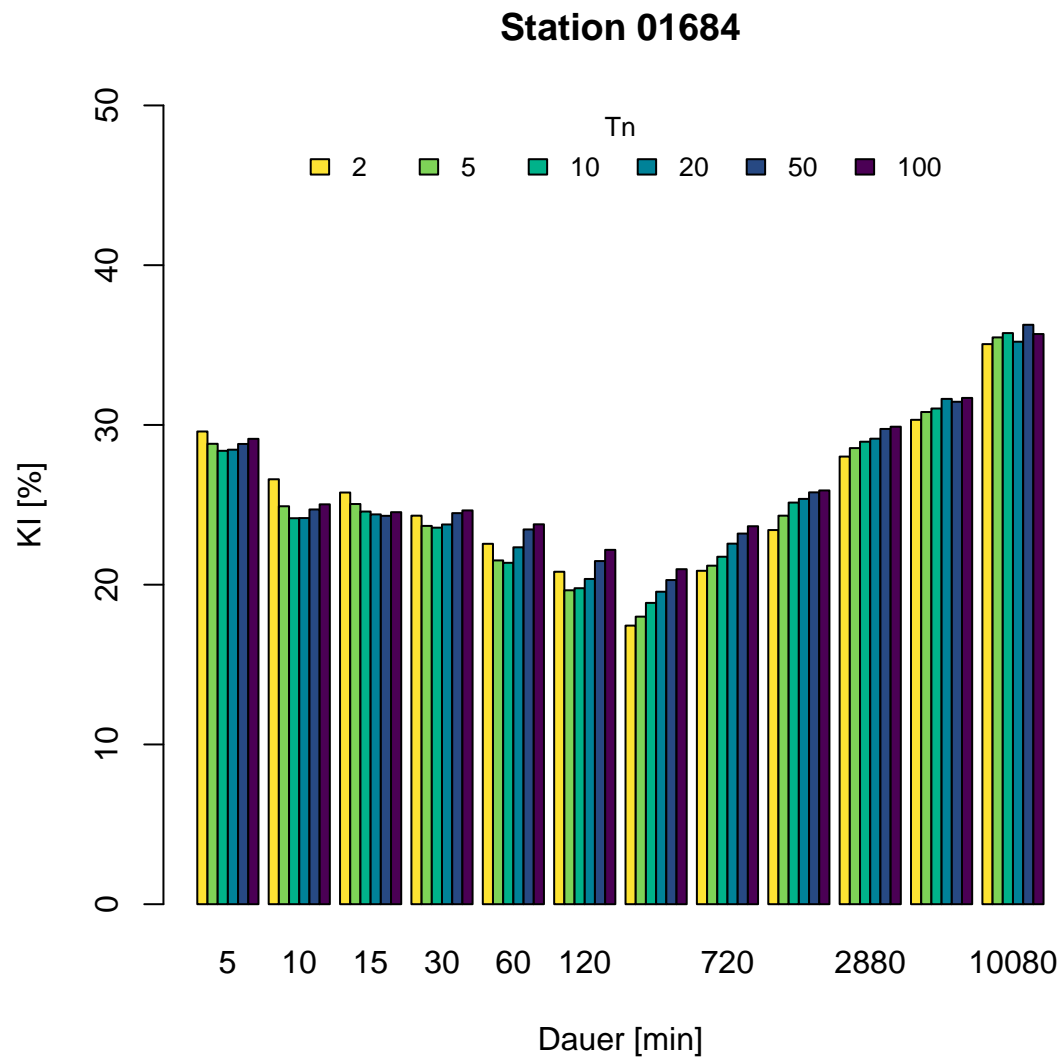


Der folgende Code zeigt drei verschiedene Beispiele fuer die Visualisierung der berechneten relativen Unsicherheiten ( $Ko-Ku/K$ ) aus dem *nBoots* Bootstrapping als eine Moeglichkeit, die Unsicherheit zu messen. Die Einheiten sind in Prozent ausgegeben.

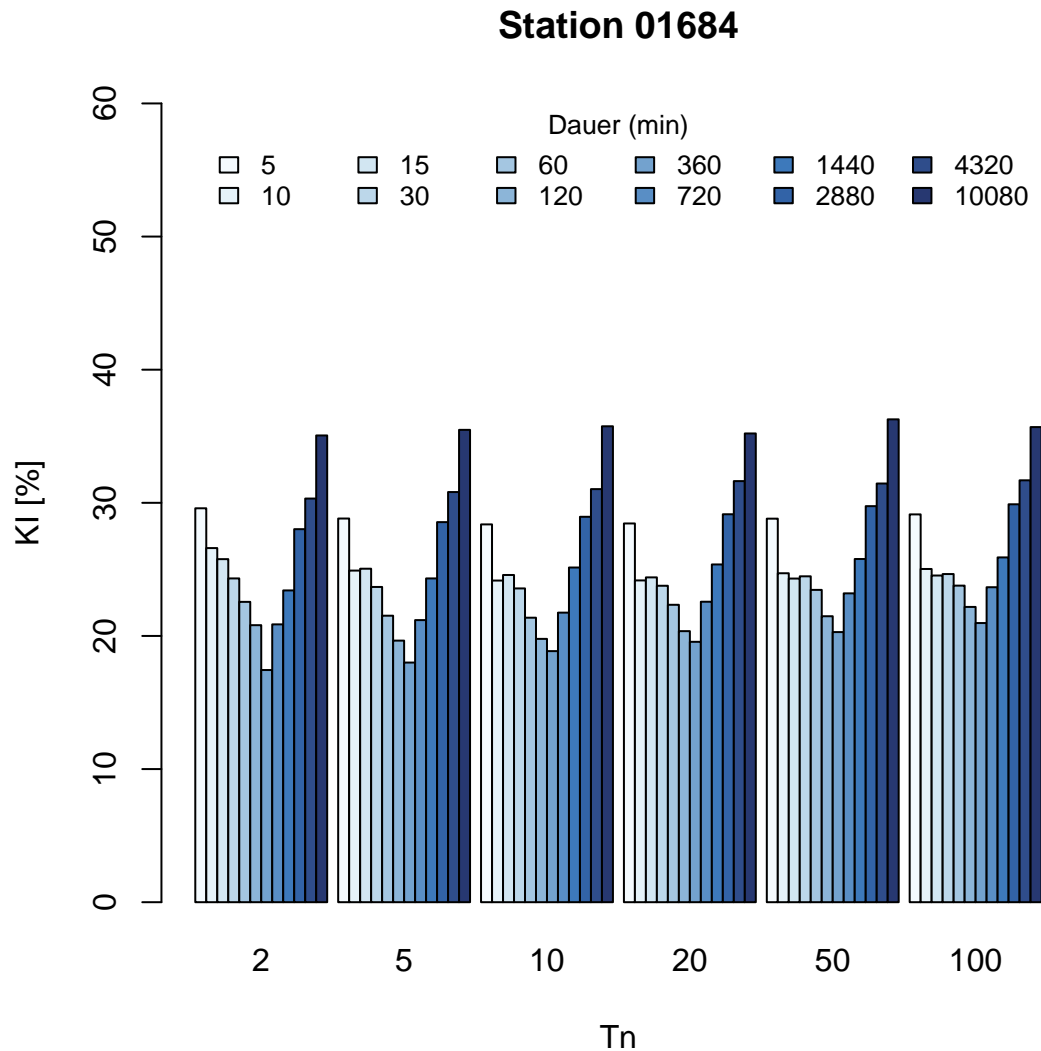
```
KI = round(HN_KI$rel.Unsicherheit,2)
# Option 1
barplot(as.matrix(KI), col=Tn_Farbe, ylab="KI [%]", xlab="Dauer [min]",
        main = "Station 01684")
legend("top", legend = c(2,5,10,20,50,100), fill=Tn_Farbe, cex=0.8,
       title="Tn", horiz=T, bty="n")
```



```
# Option 2
barplot(as.matrix(KI), col=Tn_Farbe, beside=T, ylim=c(0,50), ylab="KI [%]",
        xlab="Dauer [min]", main = "Station 01684")
legend("top", legend = c(2,5,10,20,50,100), fill=Tn_Farbe, cex=0.8,
        title="Tn", horiz=T, bty="n")
```



```
# Option 3
Dauer_Farbe = hcl.colors(dim(KI)[2], "blues", rev = T)
barplot(as.matrix(t(KI)), col=Dauer_Farbe, beside=T, ylim=c(0,60),
        ylab="KI [%]", xlab="Tn", main = "Station 01684")
legend("top", legend = Dauern, fill=Dauer_Farbe, cex=0.8, title="Dauer (min)"
      , bty="n", ncol=6)
```



### 3.5 Schaetzung der Wiederkehrintervallue fuer angegeben Dauer und Regenhoehe

Moeglicherweise ist es auch erwuenscht, dass nach der Beobachtung eines Regenereignisses die Jaehrlichkeit bei verschiedenen Dauern erfasst werden kann.

Zu diesem Zweck kann die Funktion `Tn_Schaetzung()` verwendet werden. Die Parameter `N_pars`, die die Extremwertstatistik beschreiben, sollten als einzeliger Dataframe angegeben (Benennung der Spalten ist wichtig!), die beobachtete Regenmenge `hN` in mm sollten bei verschiedenen Dauern `Dauern` als Vektor angegeben. Es ist wichtig, dass die Laenge der Vektoren `hN` und `Dauern` uebereinstimmen! Ausserdem sollte es in der Funktion angegeben ob die Gumbel- oder GEV-Parameter in `N_pars` verwendet sind.

Waehrend eines Starkregenereignisses am 18. Juli 2010 wurden an der Station Goerlitz 58,6 mm Niederschlagsvolumen in 6 Stunden gemessen. Basierend auf den geschaetzten Parametern betraegt die berechnete Wiederkehrperiode:

```
Ta_Ereignis = Tn_Schaetzung(N_pars, Dauern = 360, hN= 58.6, methGEV="GEV")
print(Ta_Ereignis)
```

```
#>      hN    D Tn
#> 1 58.6 360 23
```

Die Funktion `Tn_Schaetzung()` liefert eine Tabelle mit der beobachteten Regenhoehe (mm), der Dauer (min) und der geschaetzten Wiederkehrintervalle (Tn).

#### 4. Vergleiche zwischen lokaler Extremwertstatistik und KOSTRA-DWD-Werten

Fuer einen bestimmten Ort ist es auch moeglich, die oertlich geschaetzte Statistik mit den Bemessungsniederschlaegen nach KOSTRA-DWD-2020 zu vergleichen. In den folgenden Beispielen wird gezeigt, wie die Informationen ueber Extremniederschlaege aus den KOSTRA-DWD-2020-Datensatz extrahiert und mit den bisher berechneten Informationen verglichen werden koennen.

**Hinweis:** Das Beispiel wird hier nur fuer eine Station - Goerlitz - gezeigt. Es ist jedoch moeglich, mehrere Stationen auf einmal zu berechnen. Weitere Informationen finden Sie in der Funktion `help(Kostra2020_hN_Schaetzung)` oder `help(Kostra2020_Parameter)`.

Die Funktion `Kostra2020_hN_Schaetzung()` liest fuer bestimmte Standortkoordinaten, Dauer und Wiederkehrintervalle die entsprechende extreme Regenhoehe (*hN*) und den Unsicherheitsbereich (wenn Argument *Unsicherheit=T*). Das folgende Beispiel liest die KOSTRA-DWD-2020 Regenhoehe fuer Goerlitz und die angegebenen Dauern und Wiederkehrintervalle.

```
Tn = c(2,5,10,20,50,100)
Dauern = c(5, 10, 15, 30, 60, 120, 360, 720, 1440, 2880, 4320, 10080)
Station = data.frame(Stations_id = 01684, geoBreite = 51.1621,
                      geoLaenge = 14.9506)
H_quas_Kostra = Kostra2020_hN_Schaetzung(Standorte = Station, Dauern=Dauern,
                                          Tn =Tn, Temp_Pfad = "./",
                                          Unsicherheit=T)

# Zugang zu den Regenhoeihen und formatiert sie so, dass sie der zuvor geschaetzten
# Tabelle H_quas entsprechen.
Hn_Kostra = H_quas_Kostra$Kostra_HN
print(Hn_Kostra[1,1:5])
#>      ID geoBreite geoLaenge D00005_T002 D00005_T005
#> 1 1684    51.1621    14.9506         8         10.4
Hn_Kostra = matrix(unlist(Hn_Kostra[1,-(1:3)]), nrow = 6, ncol = 12, byrow=F)
Hn_Kostra = as.data.frame(Hn_Kostra)
rownames(Hn_Kostra) = Tn
names(Hn_Kostra) = Dauern

# Zugang zu den Unsicherheiten und formatiert sie so, dass sie der zuvor
# geschaetzten Tabelle H_quas entsprechen.
UC_Kostra = H_quas_Kostra$Kostra_UC
print(UC_Kostra[1,1:5])
#>      ID geoBreite geoLaenge D00005_T002 D00005_T005
#> 1 1684    51.1621    14.9506         12         13
UC_Kostra = matrix(unlist(UC_Kostra[1,-(1:3)]), nrow = 6, ncol = 12, byrow=F)
UC_Kostra = as.data.frame(UC_Kostra)
rownames(UC_Kostra) = Tn
names(UC_Kostra) = Dauern

# Da die KOSTRA-DWD-2020 Unsicherheiten in Prozent angegeben sind, kann die
# obere und untere Konfidenzgrenze wie folgt berechnet werden:
```

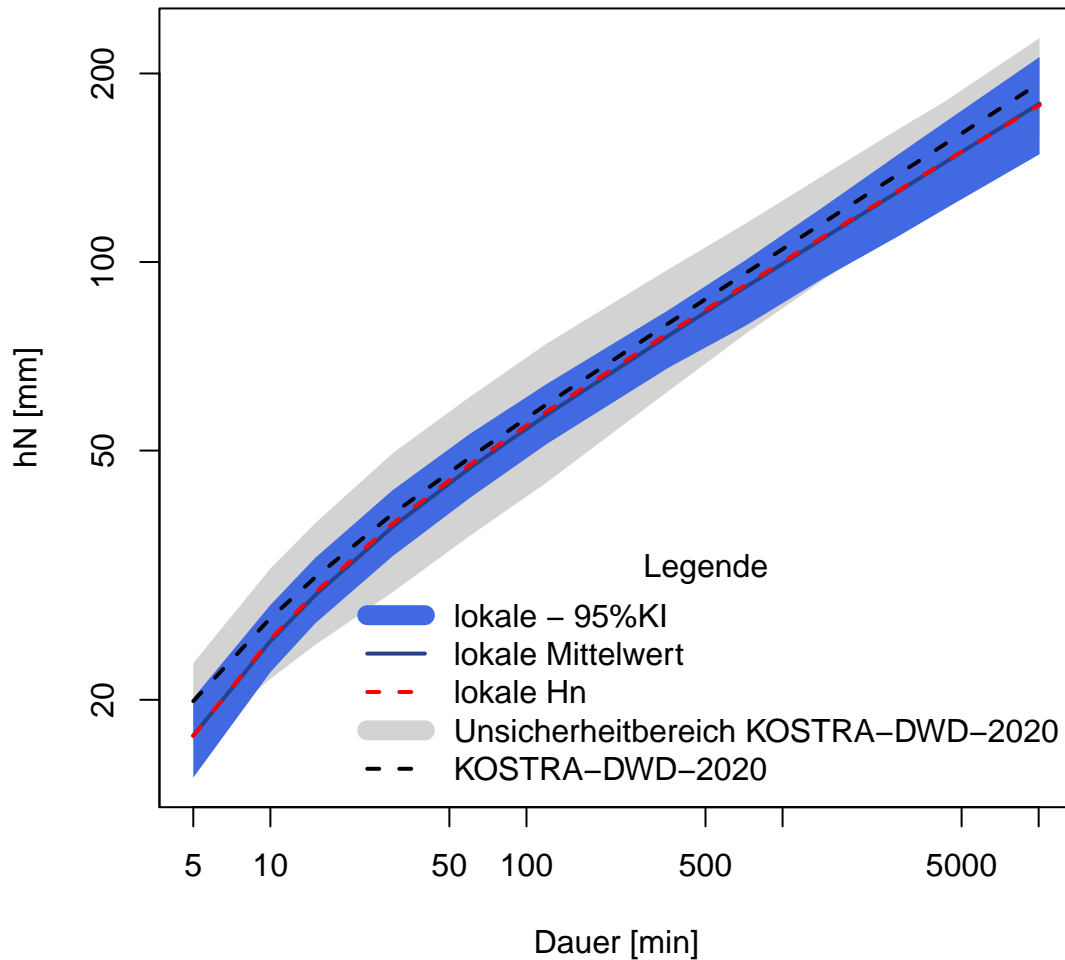
```
Hn_Kostra_Ku = Hn_Kostra - round(UC_Kostra*Hn_Kostra/100,2) # untere Grenze
Hn_Kostra_Ko = Hn_Kostra + round(UC_Kostra*Hn_Kostra/100,2) # obere Grenze
```

Das folgende Beispiel zeigt einen Codeausschnitt zur Darstellung der 100-jaehrlichen Regenhoeen fuer die Station Goerlitz, wie es fuer die Dauern von 5 Minuten bis 7 Tagen aus der bisher durchgefuehrten Extremwertanalyse der 5-Minuten-Niederschlagsdaten des DWD (rot dargestellt) und KOSTRA-DWD-2020 (schwarz dargestellt) berechnet wurde. Die 95% Konfidenzgrenzen von KOSTRA-DWD-2020 und der lokalen Analyse sind ebenfalls grau bzw. blau dargestellt.

```
Dauern = c(5, 10, 15,30,60,120,360,720,1440, 2880, 4320, 10080)
plot(Dauern, H_quas["100",], type="l", lwd=2, lty=2, log="xy",
     ylim=range(HN_KI$`97.5%`["100",], HN_KI$`2.5%`["100",],
               Hn_Kostra_Ku["100",], Hn_Kostra_Ko["100",]), col="red",
     ylab="hN [mm]", xlab="Dauer [min]", main = "Station 01684, Tn=100")
polygon(c(Dauern, rev(Dauern)), c(Hn_Kostra_Ku["100",],
                                   rev(Hn_Kostra_Ko["100",])),col="lightgrey", border=NA)
polygon(c(Dauern, rev(Dauern)), c(HN_KI$`2.5%`["100",],
                                   rev(HN_KI$`97.5%`["100",])),col="royalblue", border=NA)
lines(Dauern, HN_KI$Mittelwert["100",], type="l", col="royalblue4", lwd=2)
lines(Dauern, H_quas["100",], type="l", col="red", lwd=2, lty=2)
lines(Dauern, Hn_Kostra["100",], type="l", col="black", lwd=2, lty=2)

legend("bottomright", c("lokale - 95%KI", "lokale Mittelwert", "lokale Hn",
                       "Unsicherheitbereich KOSTRA-DWD-2020", "KOSTRA-DWD-2020"),
     col=c("royalblue", "royalblue4", "red", "lightgrey", "black"),
     lty=c(1, 1, 2, 1, 2), lwd=c(10,2,2, 10, 2), title = "Legende", bty="n")
```

## Station 01684, Tn=100

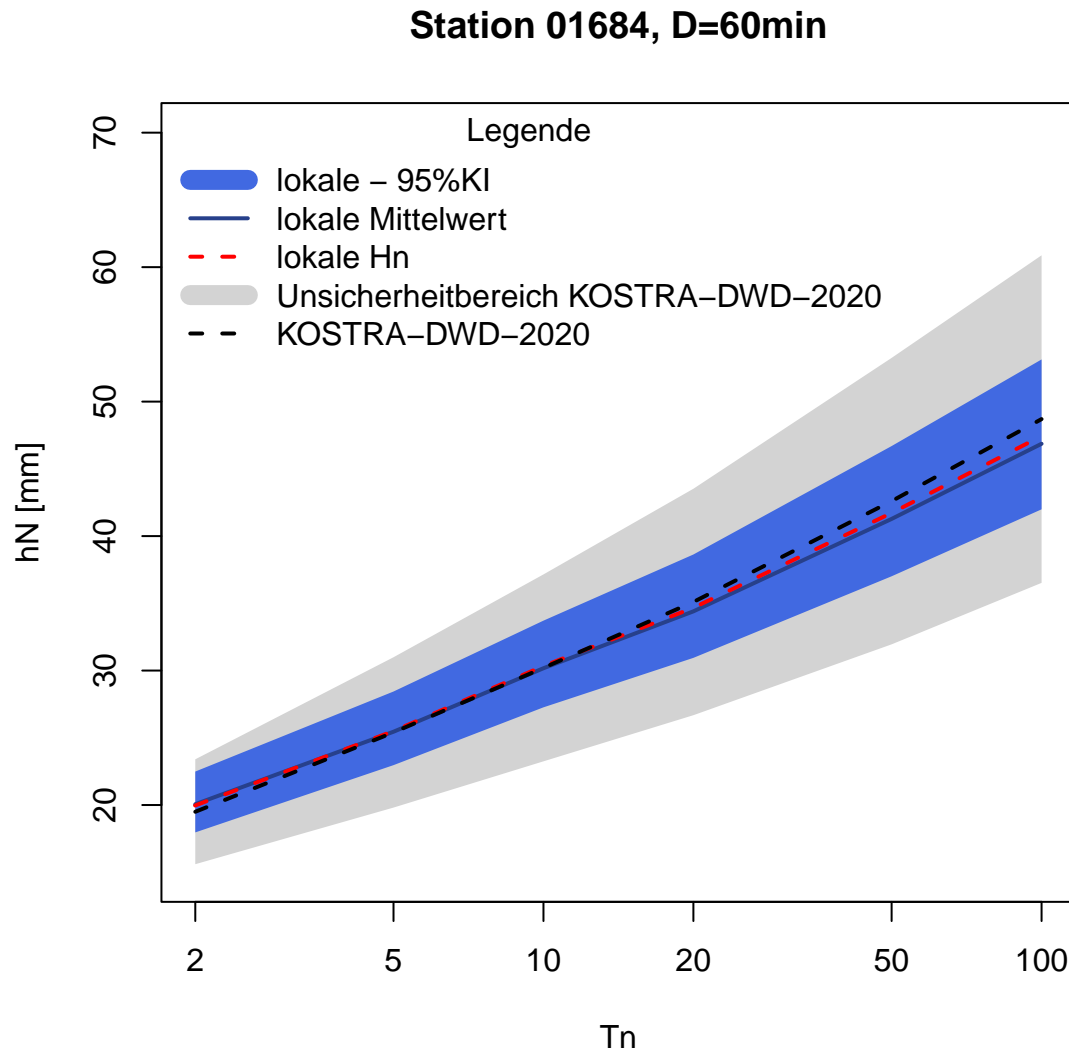


Das folgende Beispiel zeigt einen Codeausschnitt zur Darstellung der Regenhoeihen mit  $D=60\text{min}$  fuer die Station Goerlitz, wie es fuer die Wiederkehrintervalle von 2 bis 100 Jahren aus der bisher durchgefuehrten Extremwertanalyse der 5-Minuten-Niederschlagsdaten des DWD (rot dargestellt) und KOSTRA-DWD-2020 (schwarz dargestellt) berechnet wurde. Die 95% Konfidenzgrenzen von KOSTRA-DWD-2020 und der lokalen Analyse sind ebenfalls grau bzw. blau dargestellt.

```
plot(Tn, H_quas[, "60"], type="l", lwd=2, lty=2, log="x", ylim=range(15, 70),
     col="red", ylab="hN [mm]", xlab="Tn", main = "Station 01684, D=60min")
polygon(c(Tn, rev(Tn)), c(Hn_Kostra_Ku[, "60"], rev(Hn_Kostra_Ko[, "60"])),
       col="lightgrey", border=NA)
polygon(c(Tn, rev(Tn)), c(HN_KI$`2.5%`[, "60"], rev(HN_KI$`97.5%`[, "60"])),
       col="royalblue", border=NA)
lines(Tn, HN_KI$Mittelwert[, "60"], type="l", col="royalblue4", lwd=2)
lines(Tn, H_quas[, "60"], type="l", col="red", lwd=2, lty=2)
lines(Tn, Hn_Kostra[, "60"], type="l", col="black", lwd=2, lty=2)
legend("topleft", c("lokale - 95%KI", "lokale Mittelwert", "lokale Hn",
                    "Unsicherheitbereich KOSTRA-DWD-2020", "KOSTRA-DWD-2020"),
```



```
col=c("royalblue", "royalblue4", "red", "lightgrey", "black"),
lty=c(1, 1, 2, 1, 2), lwd=c(10,2,2, 10, 2), title = "Legende", bty="n")
```



Die Funktion *Koetra2020\_Parameter()* liest fuer bestimmte Standortkoordinaten die entsprechenden Parameter von KOSTRA-DWD-2020. Das folgende Beispiel liest die KOSTRA-DWD-2020 Parameter fuer Goerlitz. Dann wird die *Tn\_Schaetzung()* Funktion verwendet um die Jaehrlichkeit der beobachtete Regenmenge *hN* bei verschiedenen Dauern Dauern zu schaezen.

```
koetraParameter      = Koetra2020_Parameter(Standorte = Station )
Hn_Tn_Koetra        = Tn_Schaetzung(koetraParameter, Dauern = 360,
                                     hN   = 58.6, # values have to be in mm
                                     methGEV="GEV")

print(Hn_Tn_Koetra)
#>      hN    D Tn
#> 1 58.6 360 21
```

## Fazit

Dieses Vignette-Dokument zeigt, wie mit dem Paket **Nextreme** eine lokale Extremwertanalyse gemaess DWA-A 531 durchgefuehrt werden kann. Am Beispiel der Station Goerlitz wurde der gesamte Analyseprozess – von der Datenstruktur bis zum Vergleich mit KOSTRA-DWD-2020 – detailliert dargestellt. Das Paket bietet damit ein leistungsfaehtiges Werkzeug fuer die praxisnahe Starkregenanalyse.

## Weiterfuehrende Informationen

Weitere Informationen und Hilfestellungen zu den Funktionen finden Sie in der Paketdokumentation oder mit dem Befehl `?Funktionname` in R. Bei Fragen oder Feedback wenden Sie sich gerne an den Paketautor.