

Bivariate Extreme Value Analysis Tutorial in R

Namitha Viona Pais, Nalini Ravishanker and James O'Donnell

2024-02-19

Contents

1	Introduction	2
2	Input	2
2.1	Data Description	2
3	Goal	3
4	Peak-Over-Threshold by GPD approach	3
4.1	Selecting the threshold	4
4.1.1	Bivariate Threshold Choice Plot	4
4.1.2	Bayesian analysis of extreme events with threshold estimation for univariate data	5
4.2	Implementing R package <i>evd</i>	6
4.3	Implementing R package <i>evir</i>	8
4.4	Implementing R package <i>POT</i>	9
4.5	Implementing R package <i>extRemes</i>	9
4.6	Conclusion	10
5	Copula approach	11
5.1	Fitting the copula in R.	11
5.1.1	Implementing R package <i>evCopula</i>	11
5.1.2	Implementing R package <i>VineCopula</i>	11
5.2	Conclusion	12
6	Maxima approach	13
6.1	Implementing R package <i>fbvevd</i>	14
6.2	Conclusion	14
7	L-comoments—Multivariate Extensions of L-moments	15
7.1	Implementing R package <i>lmomco</i>	16
7.2	Conclusion	17

1 Introduction

Bivariate extreme value analysis is a statistical method used to model the joint behavior of two variables, assuming that they follow an extreme value distribution. This method is commonly used in the analysis of extreme events, such as floods or storms, where the occurrence of one extreme event may be related to the occurrence of another. By modeling the joint behavior of these events, bivariate extreme value analysis can provide valuable insights into the likelihood and severity of extreme events, and can help in the development of more effective risk management strategies.

This file is a tutorial that provides a practical guide on how to implement various bivariate extreme value approaches using the R programming language. The tutorial covers different methods for modeling the joint behavior of two variables, including the Peak-Over-Threshold by GPD approach, the copula-based approach,maxima approach and the L-comoments approach. The tutorial includes step-by-step instructions on how to fit these models using R, and provides examples and illustrations to help understand the underlying concepts. By following this tutorial, users can gain a deeper understanding of bivariate extreme value analysis and learn how to apply these methods to their own data.

2 Input

In bivariate extreme value analysis, the input consists of observed data on two variables. The analysis focuses on studying the joint distribution of these variables in the tail regions, where extreme events occur.

2.1 Data Description

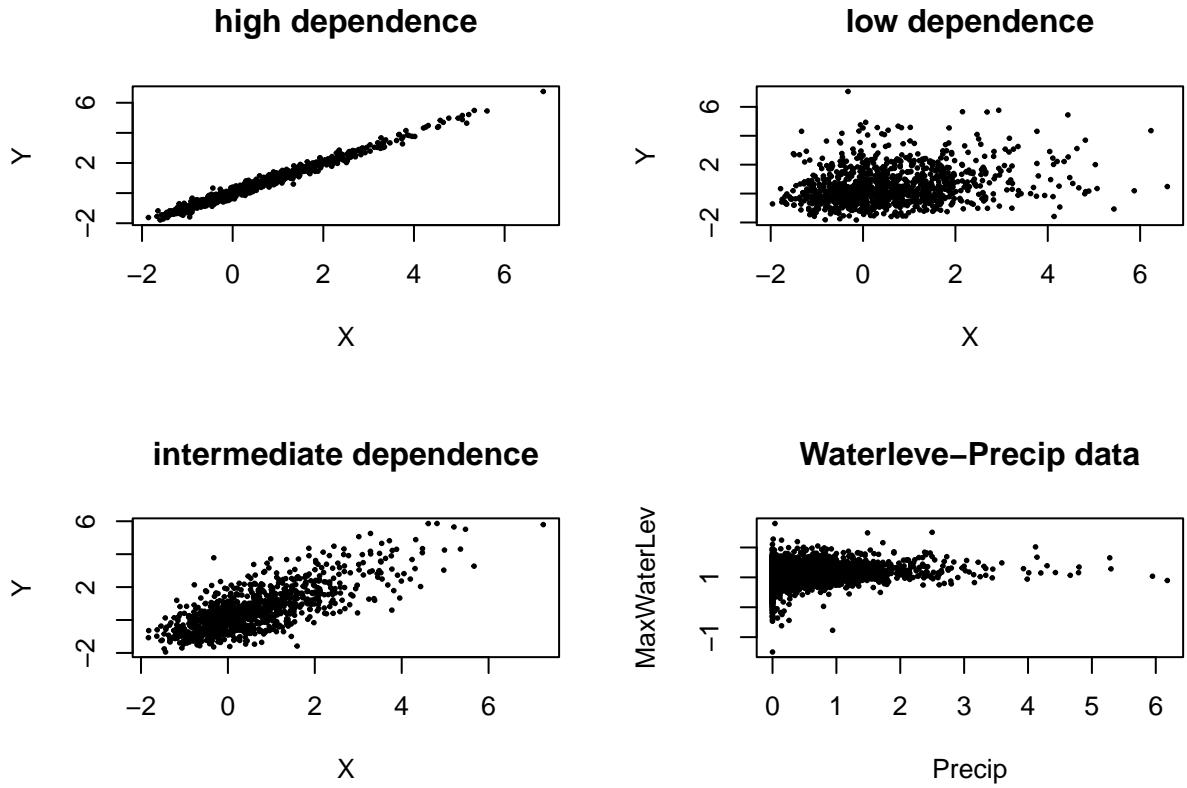
Using 3 simulated data sets (high dependence,low dependence,intermediate dependence) and bivariate sea level and precipitation data, we explore the following bivariate extreme value approaches.

We examine various bivariate extreme value approaches using a combination of simulated and real-world data. Specifically, we employ three simulated datasets that vary in their level of dependence - high, low, and intermediate - as well as real-world bivariate data on sea level and precipitation. Through these datasets, we aim to explore the effectiveness of different bivariate extreme value methods. By examining the results of these methods on both simulated and real-world data, we hope to gain insights into the strengths and limitations of each approach and their applicability in different contexts.

Here is the summary of the bivariate sea level and precipitation data.

```
##      Precip      MaxWaterLev
##  Min.   :0.00000  Min.   :-1.498
##  1st Qu.:0.00000  1st Qu.: 0.892
##  Median :0.00000  Median : 1.047
##  Mean   :0.12055  Mean   : 1.044
##  3rd Qu.:0.05118  3rd Qu.: 1.207
##  Max.   :6.18109  Max.   : 2.802
```

Here is the plot of the simulated datasets and the bivariate sea level and precipitation data.



3 Goal

The goal is to evaluate the dependence between the two variables in the tail region and obtain the return level curves for probability levels $p = 0.9, 0.95, 0.99$.

4 Peak-Over-Threshold by GPD approach

The Peak-Over-Threshold (POT) approach using the Generalized Pareto Distribution (GPD) is a popular method in extreme value analysis for modeling exceedances over a high threshold Coles [2001], Borsos [2021]. In R, there are several packages that can be used for this purpose, including *evd*, *evir*, *POT*, and *extRemes*. These packages provide functions for fitting GPD models to exceedances, estimating parameters, and making predictions. The choice of package and specific functions used will depend on the specific analysis needs and the characteristics of the data being analyzed. For a joint distribution $F(x, y)$ of (X, Y) , the tail of the marginal distribution function is approximated by a GPD distribution for suitable thresholds u_x and u_y .

By letting $\tilde{X} = -\left(\log \left\{1-\xi_x \left[1+\frac{\xi_x(X-u_x)}{\sigma_x}\right]^{-1/\xi_x}\right\}\right)^{-1}$ and $\tilde{Y} = -\left(\log \left\{1-\xi_y \left[1+\frac{\xi_y(Y-u_y)}{\sigma_y}\right]^{-1/\xi_y}\right\}\right)^{-1}$, the marginal distribution of \tilde{X} and \tilde{Y} is approximately standard Fr'echet for $X \geq u_x$ and $Y > u_y$. Then we have $F(x, y) \rightarrow G(x, y) = \exp\{-V(\tilde{x}, \tilde{y})\}$ for $x > u_x, y > u_y$ where

$$V(x, y) = 2 \int_0^1 \max\left(\frac{\omega}{x}, \frac{1-\omega}{y}\right) dH(\omega), \quad (1)$$

and H is a distribution function on $[0, 1]$ satisfying the mean constraint

$$\int_0^1 \omega dH(\omega) = 1/2 \quad (2)$$

There are a few choices of parametric families for H on $[0, 1]$ whose mean is equal to 0.5 for every value of the parameter which includes logistic family, bilogistic family, negative logistic family, Husler-Reiss etc. The most popular choice is logistic family $h(\omega)$ defined as,

$$h(\omega) = \frac{1}{2}(\alpha^{-1} - 1)\{\omega(1 - \omega)\}^{-1-1/\alpha}\{\omega^{-1/\alpha} + (1 - \omega)^{-1/\alpha}\}^{\alpha-2} \quad (3)$$

for $0 < \omega < 1$. Logistic family is popular due to the interpretation of α where $\alpha \rightarrow 1$ corresponds to independent variables and $\alpha \rightarrow 0$ corresponds to perfectly dependent variables.

Given the choice of the parametric family H , the likelihood function (censored) is given by,

$$L(\boldsymbol{\theta} | (x_1, y_1), \dots, (x_n, y_n)) = \prod_{i=1}^n \Psi(\boldsymbol{\theta} | (x_i, y_i)) \quad (4)$$

where,

$$\Psi(\boldsymbol{\theta} | (x, y)) = \begin{cases} \left. \frac{\delta^2 F}{\delta x \delta y} \right|_{(x,y)}, & (x, y) \in [u_x, \infty) \times [u_y, \infty) \\ \left. \frac{\delta F}{\delta x} \right|_{(x,u_y)}, & (x, y) \in [u_x, \infty) \times (-\infty, u_y) \\ \left. \frac{\delta F}{\delta y} \right|_{(u_x,y)}, & (x, y) \in (-\infty, u_x) \times [u_y, \infty) \\ F(u_x, u_y), & (x, y) \in (-\infty, u_x) \times (-\infty, u_y) \end{cases} \quad (5)$$

4.1 Selecting the threshold

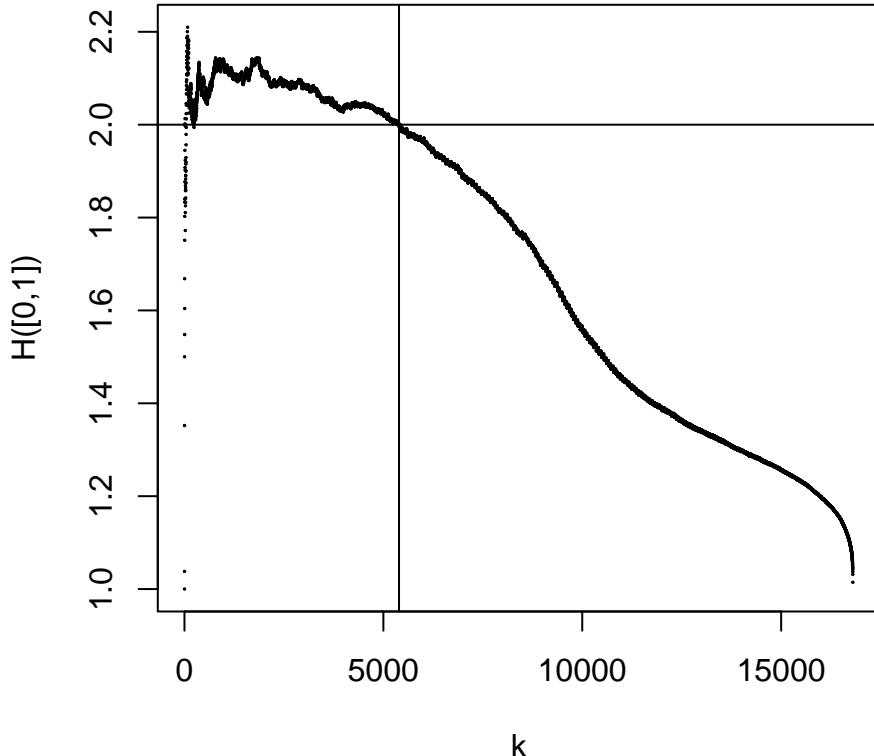
In order to model $F(z_1, z_2)$ for $z_1 > u_1$ and $z_2 > u_2$, it is necessary to select appropriate thresholds u_1 and u_2 . We discuss two methods to select the threshold values.

4.1.1 Bivariate Threshold Choice Plot

In their work Beirlant et al. [2006], the authors propose selecting a single threshold u^* based on the function $r(z)$, where $r(z) = x_1(z_1) + x_2(z_2)$ and $x_j(z_j) = \frac{-1}{\log F_j(z_j)}$ for $j = 1, 2$, where F_j is estimated empirically. One approach for selecting the threshold is to use a spectral measure plot, where integers $k = 1, 2, \dots, n-1$ are plotted against $(k/n)r_{(n-k)}$. The largest value of k , denoted as k_0 , for which $(k/n)r_{(n-k)}$ is close to 2 determines the pair of threshold values to be used.

The R package *evd* provides the function *bvtcpplot* for creating a bivariate threshold selection plot. This plot helps to identify appropriate threshold values for extreme value analysis of bivariate data. The function takes in the data for which the plot needs to be generated and the threshold selected is given as an output k_0 .

```
library(evd)
k4 <- bvtcpplot(Data4, pch=19, cex=0.1)$k0
```



```
#Threshold for Data4
thresh4<-round(apply(Data4, 2, sort, decreasing = TRUE)[(k4+5)/2,],2)
thresh4
```

```
##      Precip MaxWaterLev
##      0.19      1.29
```

The threshold estimated using this method is \$ 0.19 \$ for Precipitation and 1.29 for Water level.

4.1.2 Bayesian analysis of extreme events with threshold estimation for univariate data

The package *extrememix* [Behrens et al., 2004] enables fitting a Bayesian model to fit data characterized by extremal events where a threshold is directly estimated. The threshold is simply considered as another model parameter. This package performs inference by leveraging all available observations. It assumes that observations below the threshold are generated from a specific distribution, while observations above the threshold are generated from a GPD. The threshold parameter estimated from the two univariate models can also be considered for the bivariate GPD analysis.

```
#library(extrememix)
#p1<-fggpd(Data4[,1], it = 25000, burn = 5000, thin = 25)
#summary(p1)
#p2<-fggpd(Data4[,2], it = 25000, burn = 5000, thin = 25)
#summary(p2)
```

The threshold estimated using this method is 0.39 for Precipitation and 1.44 for Water level.

To fit the bivariate Generalized Pareto Distribution (GPD) data, we utilize the threshold obtained from the Bivariate Threshold Choice Plot. The analysis can be re-evaluated using the thresholds obtained from the *extrememix* package.

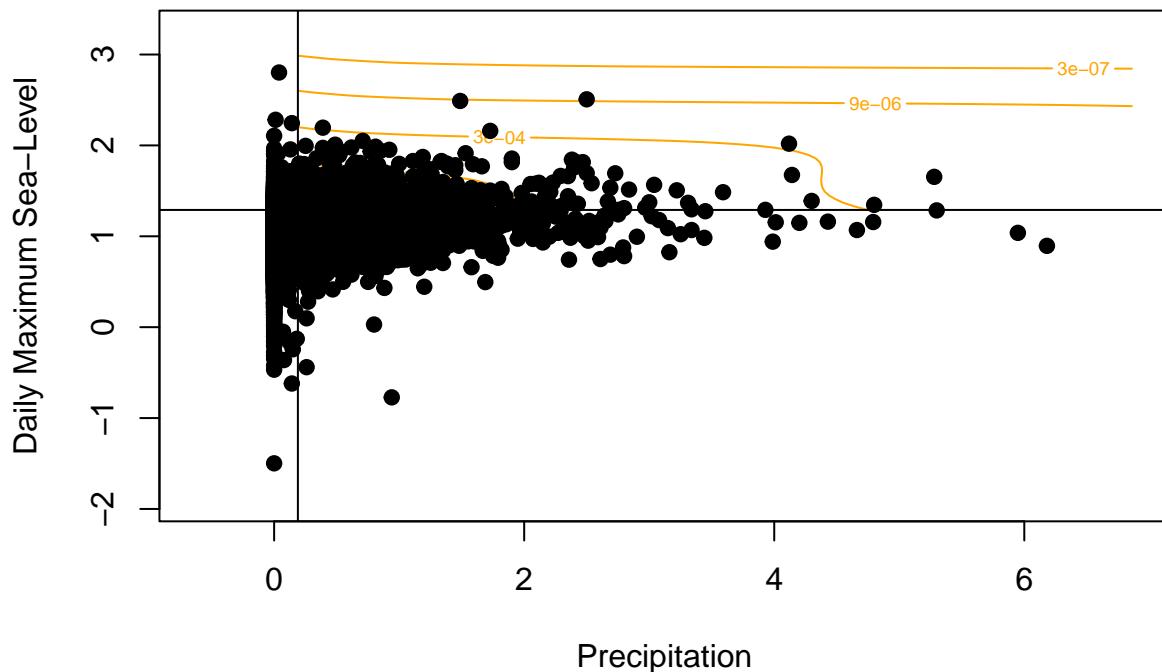
4.2 Implementing R package *evd*

The *fbvpot* function used to fit a bivariate POT model takes the data as its first argument and the threshold values for each variable as the second argument. The *model* argument specifies the parametric model to be used in the analysis. In the example given, *model = log* specifies that the logarithmic model should be used. The output of the *fbvpot* function is a list object containing various results from the analysis, such as the estimated parameters of the distribution, the standard errors of the parameters, and the value of the log-likelihood function. For instance, the command *param* gives the estimated parameters for the fitted bivariate GPD model. The *plot* function can be used to generate a bivariate POT model fit plot, with *which = 3* indicating the return level curves, and *p = c(0.9, 0.95, 0.99)* specifying the probability levels for the return levels.

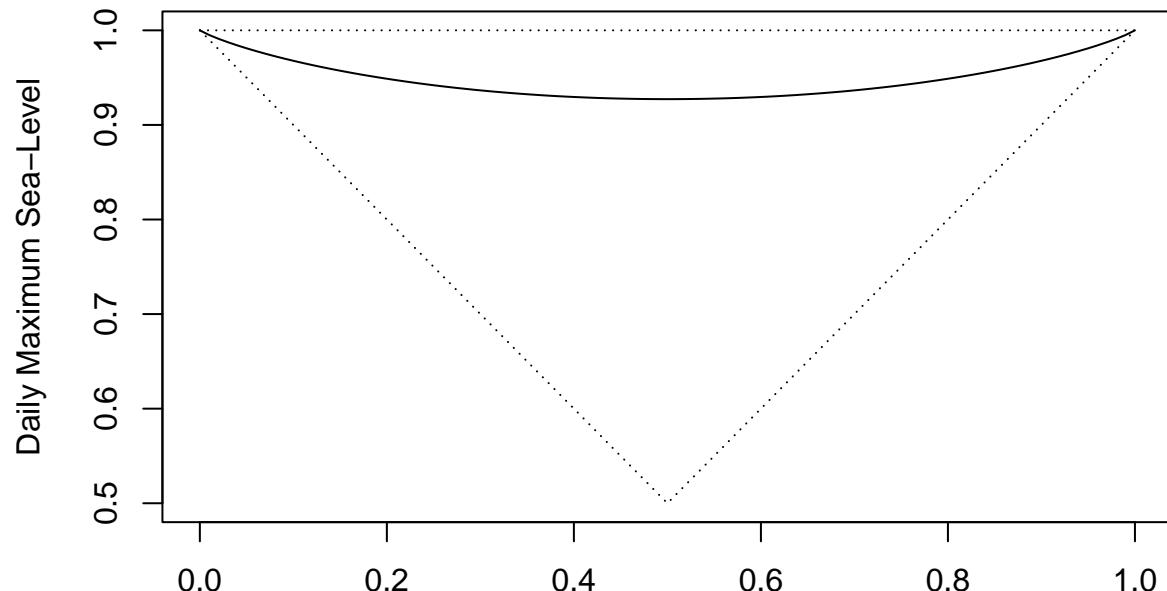
```
m4<-evd::fbvpot(Data4, thresh4, model = "log")
m4$param
```

```
##      scale1      shape1      scale2      shape2      dep
##  0.411246336  0.186592442  0.130663433 -0.002266491  0.891059540
plot(m4,which=,p=c(0.9,0.95,0.99),col="orange",pch=19,cex=0.3,
     ylab="Daily Maximum Sea-Level",xlab="Precipitation")
```

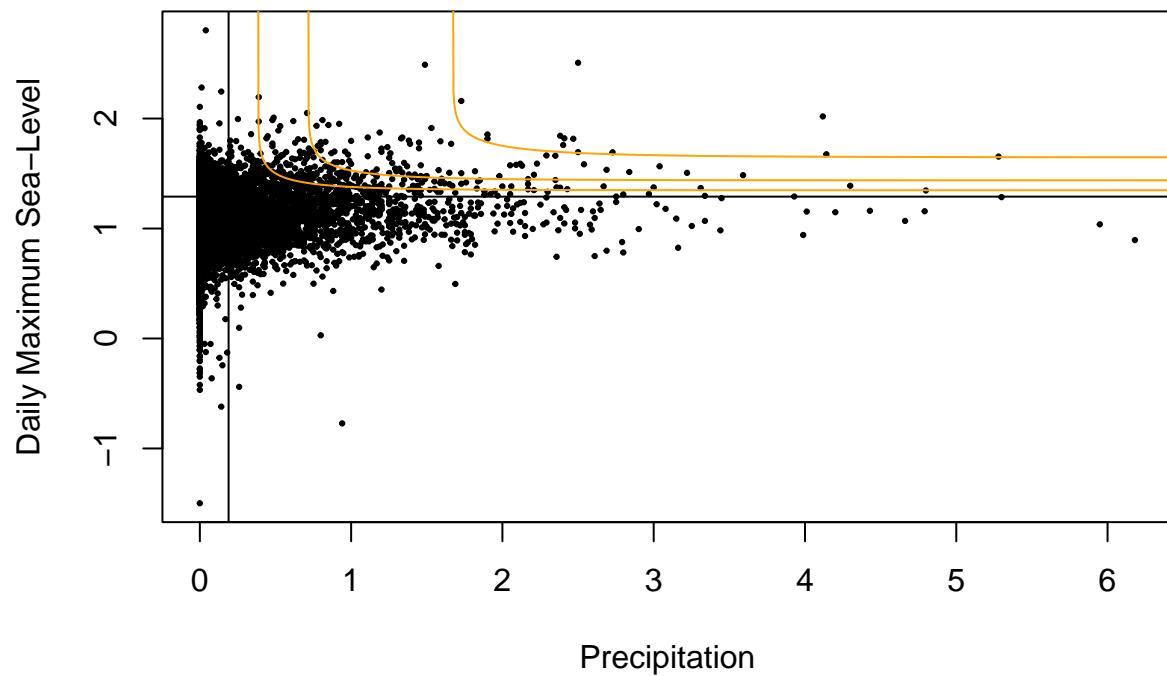
Density Plot



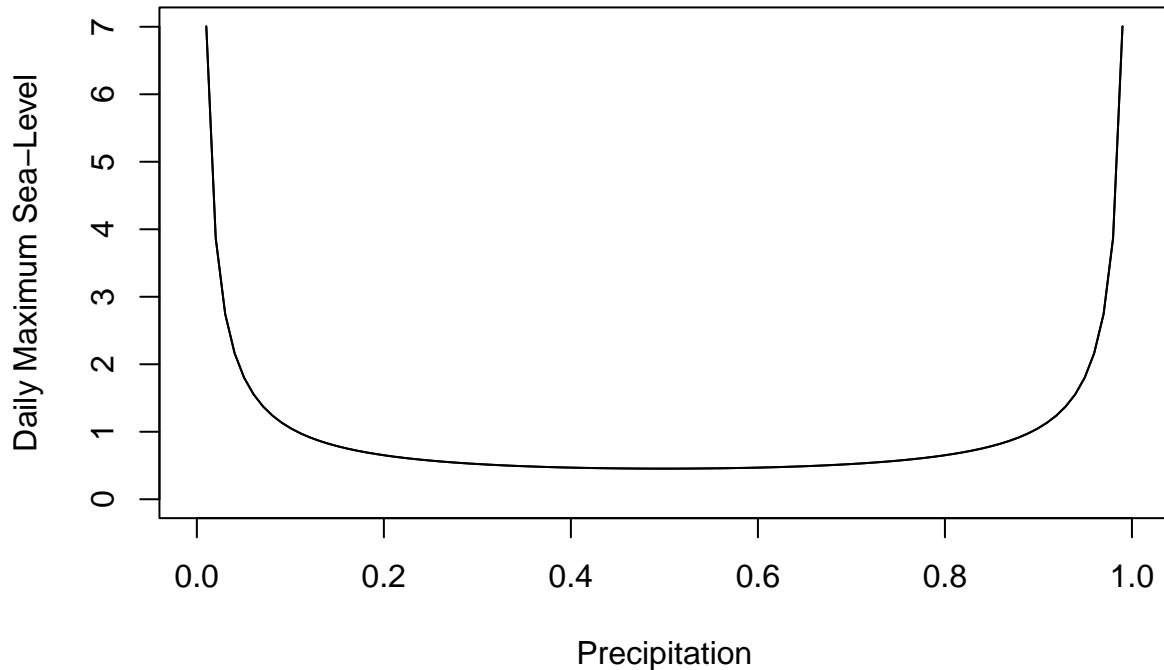
Dependence Function



Precipitation Quantile Curves



Spectral Density



4.3 Implementing R package *evir*

The function *gpdbiv* in the R package *evir* Smith [1994] used to fit a bivariate POT model requires input data along with *u1* and *u2* as arguments, representing the threshold values for the first and second variables in the analysis. After fitting the model, model estimates by calling *alpha*, *par.est1* and *par.est2*. These parameters are estimates of the parameters in the GPD distribution for each variable, which can be used to calculate return levels and other extreme value statistics.

```
library(evir)
head(Data4[,1])

## [1] 0.00000 0.00000 0.11811 0.00000 0.00000 0.00000

m14<-gpdbiv(Data4[,1],Data4[,2],u1=thresh4[1],u2=thresh4[2],global=TRUE)
m14$alpha

##
## 0.7964378

m14$par.est1

##           xi      beta
## 0.2344549 0.4099008

m14$par.est2

##           xi      beta
## 0.01882613 0.13304863
```

The function *interpret.gpdbiv* takes the fitted gpdbiv model, along with the 99% quantiles of the marginal distributions, and returns the 99% joint return level estimates for the two variables based on the fitted model. This allows us to estimate the value of both variables that we would expect to be exceeded simultaneously with a 1% chance of occurrence, which is a useful measure in the context of extreme value analysis.

```

q4<-quant99(Data4)
interpret.gpd(biv(m14,q4[1],q4[2]))

## Thresholds: 0.19 1.29
## Extreme levels of interest (x,y): 1.649603 1.6119
## P(X exceeds x) 0.01183836
## P(Y exceeds y) 0.01457128
## P(X exceeds x AND Y exceeds y) 0.003443994
## P(X exceeds x) * P(Y exceeds y) 0.0001725001
## P(Y exceeds y GIVEN X exceeds x) 0.2909181
## P(X exceeds x GIVEN Y exceeds y) 0.236355

```

4.4 Implementing R package *POT*

The *fitvgpd* Smith et al. [1997] used to fit a bivariate POT model takes the data and threshold values as inputs. In addition, logistic model is specified by providing the argument *log*. The *param* command returns the estimated parameters of the bivariate GPD model fit and the *plot* function generates a diagnostic plot of the fitted model, which can help to assess the adequacy of the model fit where *which* = 2 provides the return level plot.

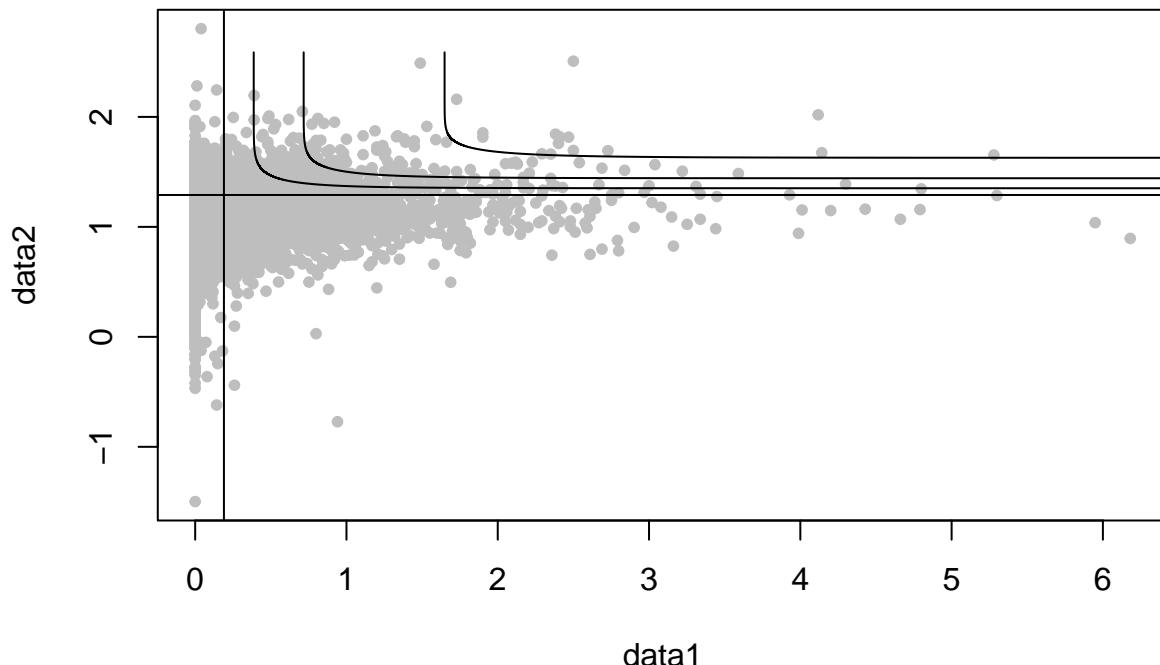
```

m24 <- fitvgpd(Data4, thresh4, "log")
m24$param

##      scale1      shape1      scale2      shape2      alpha
##  0.4164894  0.1669304  0.1428082 -0.1100797  0.7500000
plot(m24,which=2,pch=20,col="grey",p=c(0.9,0.95,0.99))

```

Bivariate Return Level Plot



4.5 Implementing R package *extRemes*

The function *fbvpot* from the *extRemes* package Beirlant et al. [2006] is used to fit a bivariate POT model with inputs as the data, threshold and dependence model which is chosen as logistic in this case.

```
library(extRemes)
m34 <- extRemes::fbvpot( x = Data4, threshold = thresh4,
                         dep.model = "logistic")
m34$fit$par

## parameter
## 0.7215404
```

4.6 Conclusion

The dependence parameter estimated using each of the packages is shown in Table 1

	Data (low dep)	Data (intermediate dep)	Data (high dep)	Real data
<i>evd</i>	0.8926	0.5865	0.1170	0.8910
<i>evir</i>	0.7801	0.5856	0.1191	0.7964
<i>POT</i>	0.8925	0.5863	0.1179	0.7500
<i>extRemes</i>	0.7267	0.5107	0.1160	0.7215

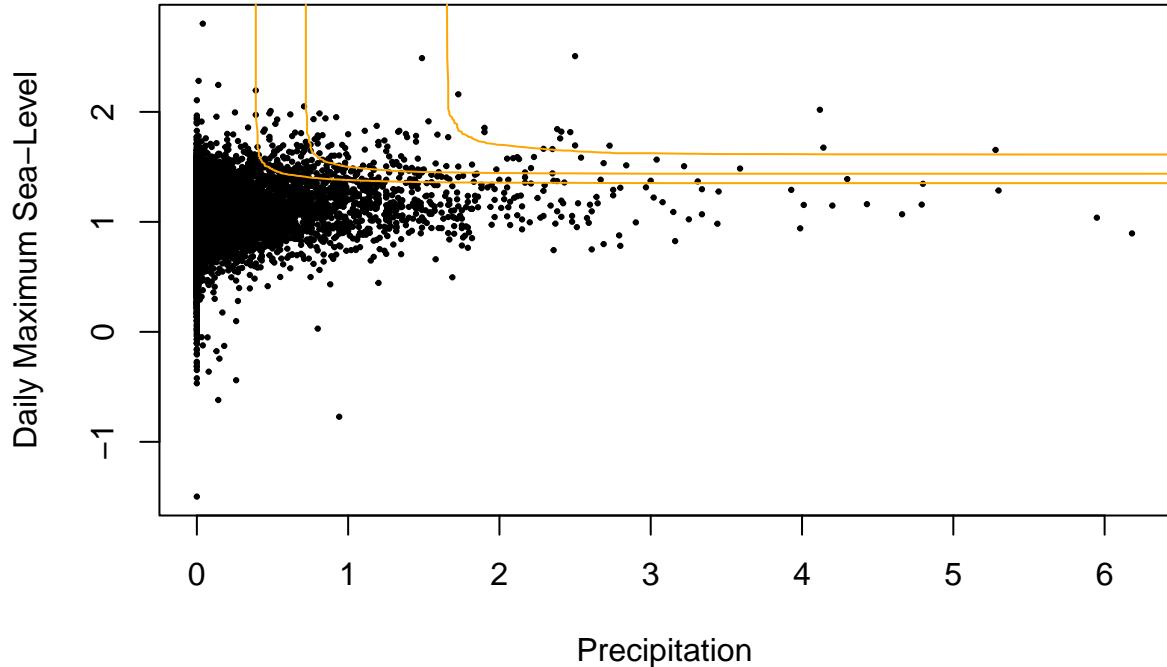
Table 1: Dependence parameter obtained from Peak-Over-Threshold by GPD approach

The variation in α values across different packages is due to the differences in numerical optimization settings. However, despite these variations, the values still lead to the same conclusion.

On the simulated data the interpretation of dependence parameter α matches with the true dependence. For the real data, the value of α indicates that the water level and precipitation are weakly dependent in the tail regions.

The plot based on the non-parametric estimates for quantile curves for $p = 0.90, 0.95, 0.99$ is shown below.

```
qcbvnonpar(p=c(0.9,0.95,0.99), data = Data4, epmar = TRUE,
            plot=TRUE,cex=0.3,pch=19,col="orange",
            ylab="Daily Maximum Sea-Level",xlab="Precipitation")
```



5 Copula approach

5.1 Fitting the copula in R.

When working with extreme values, copulas are a powerful tool for modeling the dependence between two or more variables. In R, the copula package provides a range of functions and methods for working with copulas, including estimation, simulation, and goodness-of-fit testing.

5.1.1 Implementing R package *evCopula*

The *evCopula* class in the copula package is a class of all extreme-value copulas. There are currently five subclasses that inherit from *evCopula*: *galambosCopula*, *huslerReissCopula*, *tawnCopula*, *tevCopula*, and *gumbelCopula*. Among these subclasses, *gumbelCopula* is an Archimedean copula, which is also documented on the page for the *archmCopula* class.

As an illustration we consider Gumbel copula fit to the bivariate water level precipitation data. The Gumbel copula (a.k.a. Gumbel-Hougaard copula) is an asymmetric Archimedean copula, exhibiting greater dependence in the positive tail than in the negative. This copula is given by:

$$C_\alpha(u, v) = \exp \left\{ - [(-\ln u)^\alpha + (-\ln v)^\alpha]^{\frac{1}{\alpha}} \right\} \quad (6)$$

where $\alpha \in [1, \infty)$

The relationship between Kendall's τ and the Gumbel copula parameter α is given by

$$\alpha = \frac{1}{1 - \tau} \quad (7)$$

```
library(copula)
library(VineCopula)

## 
## Attaching package: 'VineCopula'
```

```

## The following object is masked from 'package:copula':
##
##      pobs

gumbel.cop <- gumbelCopula(3, dim=2)
set.seed(500)
m <- pobs(as.matrix(Data4))
fit1 <- fitCopula(gumbel.cop,m,method='ml')
coef(fit1)

##      alpha
## 1.142687

BiCopPar2Tau(4,coef(fit1))

## [1] 0.1248699

```

To fit a copula to extreme value data in R, the first step is typically to transform the data to uniform marginals using the *pobs* function. The copula can then be estimated using the *fitCopula* function, which takes as input the copula object and the uniform data. Once a copula has been fitted, it can be used for simulations, to estimate dependence measure Kendall's tau, and to compute return levels.

5.1.2 Implementing R package *VineCopula*

One way to select the copula family is to use the function *BiCopSelect* from the package *VineCopula* which selects the copula family that best fits the observed dependence structure between u and v using the Bivariate Copula Selection method. The *familyset = NA* argument specifies that all available copula families should be considered. In this instance, a bivariate Gaussian copula is selected which is given by,

$$C_R(u_1, u_2) = \Phi_R(\Phi^{-1}(u_1), \Phi^{-1}(u_2)) \quad (8)$$

In this expression, $C_R(u_1, u_2)$ represents the Gaussian copula, Φ_R is the cumulative distribution function (CDF) of the standard bivariate normal distribution, and Φ^{-1} denotes the inverse CDF of the standard normal distribution. The variables u_1 and u_2 typically represent the marginal probabilities or quantiles associated with the two random variables in the copula.

Based on the copula family selected we use *fitCopula* to fit copula to the observed data using maximum likelihood estimation (MLE). The *coef* function provides the estimated correlation coefficient between the two variables. If we use a gumbel copula to fit the data the function *coef* provides the tail dependence parameter estimate. Then based on the copula and its estimated parameter, we can evaluate the theoretical Kendall's tau using the function *BiCopPar2Tau*.

```

library(VineCopula)
u<- pobs(as.matrix(Data4))[,1]
v <- pobs(as.matrix(Data4))[,2]
selectedCopula <- BiCopSelect(u,v,familyset=NA)

## Warning in cor(x[(x[, 1] < 0) & (x[, 2] < 0), ]): the standard deviation is
## zero

selectedCopula

## Bivariate copula: Gaussian (par = 0.27, tau = 0.17)
myCop <- normalCopula(dim = 2)
set.seed(500)
m <- pobs(as.matrix(Data4))
fit <- fitCopula(myCop,m,method='ml')
coef(fit)

```

```

##      rho.1
## 0.2696213
BiCopPar2Tau(1,coef(fit))

## [1] 0.1737971

```

5.2 Conclusion

The Copula fit for each of the data and the corresponding Kendals τ estimated is shown in Table 2

	Data (low dep)	Data (intermediate dep)	Data (high dep)	Real data
Copula Selected	Frank	Gumbel	Gumbel	Gaussian
Kendals τ	0.1528	0.5012	0.8994	0.1738

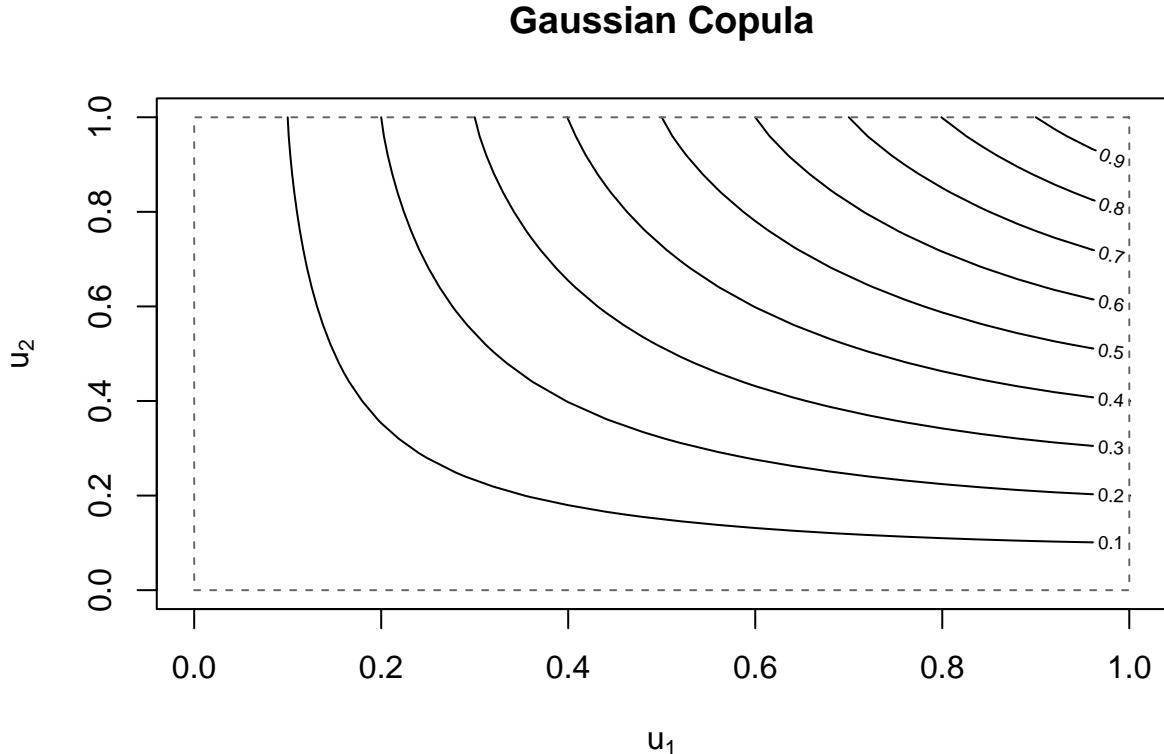
Table 2: Kendals τ obatined from the best copula fit

The Kendall's τ estimated from both the Gumbel copula is 0.1249 and from the Gaussian copula is 0.1738. This indicates low dependence between the two variables. The contour plots obtained from the copula fits is shown below.

```

myCop <- normalCopula(coef(fit),dim = 2)
contour(myCop, pCopula,main="Gaussian Copula")

```



6 Maxima approach

The maxima approach for bivariate extreme value analysis is a statistical method used to model the joint distribution of two variables based on their maximum values. The approach assumes that the variables follow a multivariate extreme value distribution, and that the joint distribution can be estimated by considering the maximum values of each variable across a number of observations.

We define $M_{x,n} = \max_{i=1,2,\dots,n} \{X_i\}$ and $M_{y,n} = \max_{i=1,2,\dots,n} \{Y_i\}$. Then, $\mathbf{M}_n = (M_{x,n}, M_{y,n})$ is the vector of componentwise maxima which need not be an observed vector in the original series. Our goal is to estimate the distribution of \mathbf{M}_n as $n \rightarrow \infty$.

Since, $M_{x,n} \sim GEV(\mu_x, \sigma_x, \xi_x)$ and $M_{y,n} \sim GEV(\mu_y, \sigma_y, \xi_y)$, by letting $\tilde{X} = \left[1 + \frac{(X - \mu_x)}{\sigma_x}\right]^{1/\xi_x}$ $\tilde{Y} = \left[1 + \frac{(Y - \mu_y)}{\sigma_y}\right]^{1/\xi_y}$ will ave the standard Fr'echet distribution. The following theorem can be used to estimate the joint distribution.

Theorem: Let $\mathbf{M}_n^* = (M_{\tilde{x},n}^*, M_{\tilde{y},n}^*)$ where $(\tilde{X}_i, \tilde{Y}_i)$ are independent vectors with standard Fr'echet marginal distribution. Then if

$$P\{M_{\tilde{x},n}^* \leq x, M_{\tilde{y},n}^* \leq y\} \xrightarrow{d} G(x, y), \quad (9)$$

where G is a non-degenerate distribution function, G has the form

$$G(x, y) = \exp\{-V(x, y)\}, x > 0, y > 0 \quad (10)$$

where

$$V(x, y) = 2 \int_0^1 \max\left(\frac{\omega}{x}, \frac{1-\omega}{y}\right) dH(\omega), \quad (11)$$

and H is a distribution function on $[0, 1]$ satisfying the mean constraint

$$\int_0^1 \omega dH(\omega) = 1/2 \quad (12)$$

6.1 Implementing R package *fbvevd*

The *fbvevd* function in the evd package in R is used to estimate the parametric fit of the bivariate extreme value distribution. It fits a bivariate EVD to the joint distribution of the maxima of two variables. We consider a (symmetric) logistic extreme value distribution by setting *model* = *log* and the function *fitted* contains information on the fit which has a single dependence parameter and three parameters on each of the GEV margins.

```
xx <- rep(1:2801, each = 6)
data_grouped <- cbind(tapply(Data4[,1], xx, max),
                      tapply(Data4[,2], xx, max))
colnames(data_grouped) <- colnames(Data4)
summary(data_grouped)

##      Precip          MaxWaterLev
##  Min.   :0.00000   Min.   :0.028
##  1st Qu.:0.07874   1st Qu.:1.103
##  Median :0.33858   Median :1.239
##  Mean   :0.53241   Mean   :1.252
##  3rd Qu.:0.75197   3rd Qu.:1.392
##  Max.   :6.18109   Max.   :2.802

model4 <- fbvevd(data_grouped, model = "log", std.err = FALSE)
fitted(model4)

##           loc1       scale1       shape1        loc2       scale2       shape2         dep
##  0.1597052  0.2213529  0.8162166  1.1271440  0.2823068 -0.1677235  0.7500000
```

6.2 Conclusion

The dependence parameter estimated using maxima approach is shown in Table 3

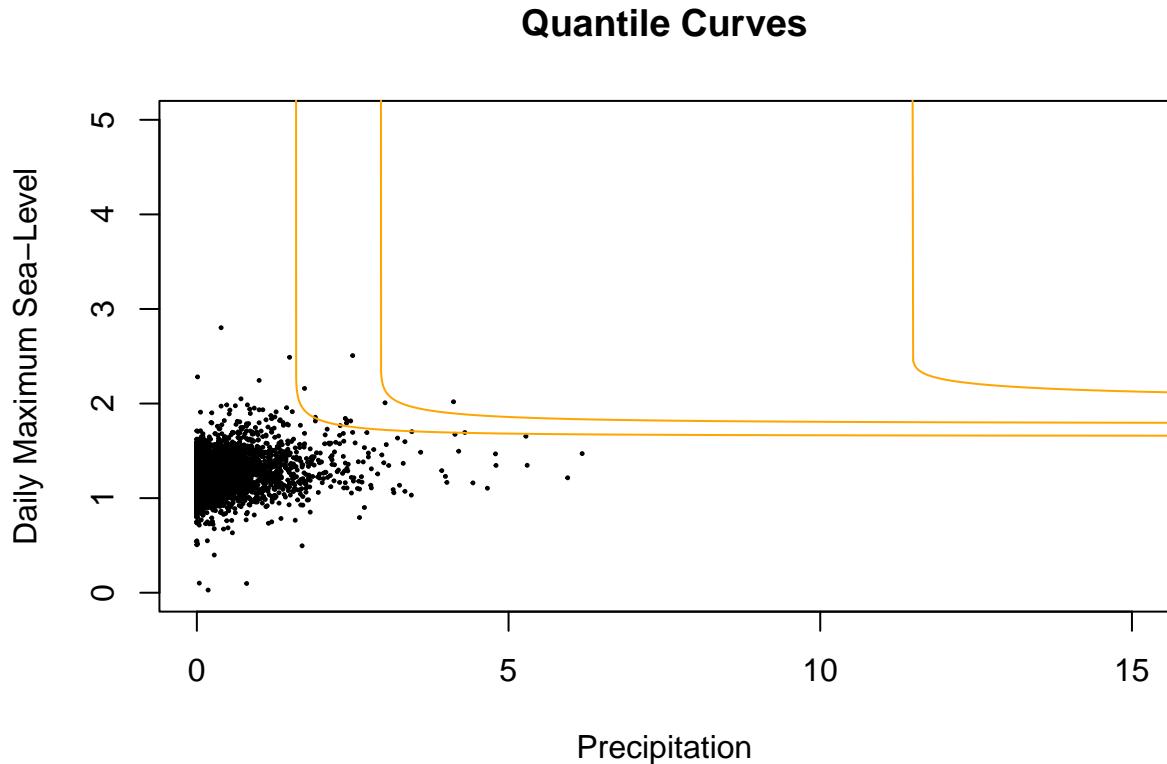
	Data (low dep)	Data (intermediate dep)	Data (high dep)	Real data
α	0.8916	0.5360	0.10519	0.7500

Table 3: Dependence parameter α obtained from maxima approach

On the simulated data the interpretation of dependence parameter α matches with the true dependence. For the real data, the value of α indicates that the water level and precipitation are weakly dependent in their maxima.

The plot on quantile curves for $p = 0.90, 0.95, 0.99$ is shown below.

```
plot(model4, which = 5, pch = 20, p=c(0.90, 0.95, 0.99),
      xlim=c(0,15), ylim=c(0,5),
      col="orange", ylab="Daily Maximum Sea-Level",
      xlab="Precipitation", cex=0.3)
```



7 L-comoments—Multivariate Extensions of L-moments

Multivariate L-moments, also known as L-comoments, extend the concept of L-moments to multivariate distributions. Introduced by Serfling and Xiao [2007], L-comoments provide a framework for analyzing the joint behavior, dependence structure, and shape characteristics of multiple variables simultaneously.

In the univariate setup, the r^{th} population L-moment is defined as,

$$\lambda_r = r^{-1} \sum_{k=0}^{r-1} (-1)^k \binom{r-1}{k} E(X_{r-k:r}) \quad (13)$$

where $X_{k:n}$ represents k^{th} order statistic in a sample of size n .

Similarly in the bivariate setup, the r^{th} population L-comoment of $X^{[1]}$ wrt $X^{[2]}$ is defined as,

$$\lambda_r^{[12]} = r^{-1} \sum_{k=0}^{r-1} (-1)^k \binom{r-1}{k} E(X_{r-k:r}^{[12]}) \quad (14)$$

where $X_{r-k:r}^{[12]}$ indicates the element of $\{X_1^{(1)}, X_2^{(1)}, \dots, X_n^{(1)}\}$ that is paired with $X_{r:n}^{(2)}$ and $E(X_{r-k:r}^{[12]}) = nE(X_1^{(1)} | X_1^{(2)} = X_{r:n}^{(2)})$.

Serfling and Xiao [2007] provides an unbiased sample estimator for $\lambda_r^{[12]}$ given by,

$$\hat{\lambda}_r^{[12]} = \frac{1}{n} \sum_{j=1}^n w_{j:n}^{(r)} x_{j:n}^{[12]} \quad (15)$$

where $x_{j:n}^{[12]}$ is the ordered sample and the weights $w_{j:n}^{(r)}$ are computed as,

$$w_{j:n}^{(r)} = \sum_{i=1}^{\min(j-1, r-1)} (-1)^{r-1-i} \binom{r-1}{i} \binom{r-1+i}{i} \binom{j-1}{i} / \binom{n-1}{i} \quad (16)$$

Similarly we can define the estimator for the r^{th} L-comoment of $X^{[2]}$ wrt $X^{[1]}$. An important characteristic of L-comoments is that they need not be symmetric.

The L-comoment ratios in the form of $\tau_r^{[12]} = \frac{\lambda_r^{[12]}}{\lambda_2^{[1]}}$ or $\tau_r^{[21]} = \frac{\lambda_r^{[21]}}{\lambda_2^{[2]}}$ for $r \geq 2$ can be seen as analogous to the univariate L-moment ratios, τ_r . Specifically, when $r = 2$, $\tau_2^{[12]}$ represents the L-correlation of $X^{[1]}$ with respect to $X^{[2]}$. On the other hand, when $r = 3$ and $r = 4$, $\tau_3^{[21]}$ and $\tau_3^{[21]}$ respectively represent the L-coskewness and L-cokurtosis of $X^{[2]}$ with respect to $X^{[1]}$. These L-comoment ratios [Asquith, 2011] provide insights into the multivariate relationships, dependencies, and higher-order moments between the different components of the observed data, extending the analysis beyond the univariate setup.

7.1 Implementing R package *lmomco*

We utilize the *lmomco* package in R to analyze the bivariate data consisting of precipitation and water level. The code generates a plot of the bivariate data and adds two rug plots in red color using the *rug* function. These rug plots display the marginal distributions of each variable along their respective axes.

The code then calculates the L-moment matrix of order 2 and stores it in the variable *L2*. By utilizing this matrix, the L-correlation between the two variables is computed using the *Lcomoment.correlation* function. The resulting L-correlations are found to be $\tau_2^{[12]} = 0.3160835$ and $\tau_2^{[21]} = 0.2628868$. These values indicate a weak association between the variables.

Furthermore, the code computes the L-moment matrix of order 3 and calculates the L-coskew between the two variables using the *Lcomoment.coefficients* function. The obtained values are $\tau_3^{[12]} = 0.07018642$ and $\tau_3^{[21]} = 0.0307913$. Although the L-coskews differ slightly, they are close to zero. This suggests a symmetry of sorts in the co-movement of X with respect to Y and Y with respect to X.

Overall, based on the L-correlations and L-coskews, the code indicates a weak association between the variables.

```

library(lmomco)

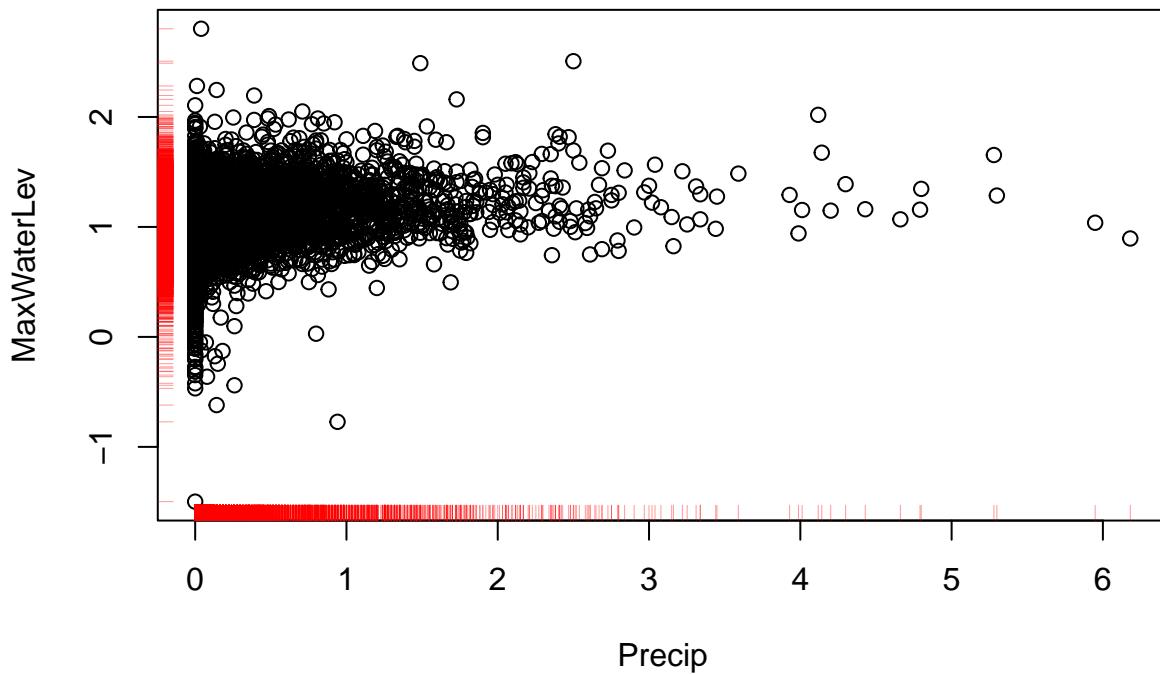
##
## Attaching package: 'lmomco'

## The following object is masked from 'package:POT':
##      pp

## The following object is masked from 'package:evd':
##      pp

plot(Data4)
rug(Data4$Precip, side=1, col=rgb(1,0,0,0.4))
rug(Data4$MaxWaterLev, side=2, col=rgb(1,0,0,0.4))

```



```

L2 <- Lcomoment.matrix(Data4, k=2) # order 2 matrix
Lcomoment.correlation(L2) # compute L-correlation

```

```

## $type
## [1] "Lcomoment.coefficients"
##
## $order
## [1] 2
##
## $matrix
## [,1]      [,2]
## [1,] 1.0000000 0.3160835
## [2,] 0.2628868 1.0000000

```

```

L3 <- Lcomoment.matrix(Data4, k=3)
Lcomoment.coefficients(L3,L2) # compute L-coskew

```

```

## $type

```

```

## [1] "Lcomoment.coefficients"
##
## $order
## [1] 3
##
## $matrix
##           [,1]      [,2]
## [1,] 0.7652255 0.07018642
## [2,] 0.0307913 -0.02543878

```

7.2 Conclusion

The L-correlation estimates for different datasets is shown in Table 4

	Data (low dep)	Data (intermediate dep)	Data (high dep)	Real data
$\tau_2^{[12]}$	0.2412	0.7403	0.9889	0.3161
$\tau_2^{[21]}$	0.2311	0.7316	0.9889	0.2628

Table 4: L-correlation estimates for different datasets

On the simulated data the interpretation of L-correlations matches with the true dependence. For the real data, the value of L-correlations indicates that the water level and precipitation indicates a weak association between the variables.

References

- William H Asquith. *Distributional analysis with L-moment statistics using the R environment for statistical computing*. CreateSpace Scotts Valley, CA, USA, 2011.
- Cibele N Behrens, Hedibert F Lopes, and Dani Gamerman. Bayesian analysis of extreme events with threshold estimation. *Statistical modelling*, 4(3):227–244, 2004.
- Jan Beirlant, Yuri Goegebeur, Johan Segers, and Jozef L Teugels. *Statistics of extremes: theory and applications*. John Wiley & Sons, 2006.
- Attila Borsos. Application of bivariate extreme value models to describe the joint behavior of temporal and speed related surrogate measures of safety. *Accident Analysis & Prevention*, 159:106274, 2021.
- Stuart Coles. *An Introduction to Statistical Modeling of Extreme Values*. Springer, 1 edition, 2001.
- Robert Serfling and Peng Xiao. A contribution to multivariate l-moments: L-comoment matrices. *Journal of Multivariate Analysis*, 98(9):1765–1781, 2007.
- Richard L Smith. Multivariate threshold methods. In *Extreme Value Theory and Applications: Proceedings of the Conference on Extreme Value Theory and Applications, Volume 1 Gaithersburg Maryland 1993*, pages 225–248. Springer, 1994.
- Richard L Smith, Jonathan A Tawn, and Stuart G Coles. Markov chain models for threshold exceedances. *Biometrika*, 84(2):249–268, 1997.