Conditional modelling of multivariate extreme value data using R

Harry Southworth and Janet E. Heffernan

September 28, 2016

1 Introduction

This document illustrates the use of the `texmex` package, [7] for performing extreme value analysis of multivariate data in `R`, [5]. Broadly speaking, the analysis proceeds in two steps: generalized Pareto distribution (GPD) modelling of the marginal variables followed by conditional multivariate extreme value modelling. The first step is covered in more detail in the `texmex` vignette `texmex1d`; here we describe briefly the stages of the univariate modelling and focus in more detail on the multivariate modelling.

To cite this vignette, refer to Vignette name: `texmexMultivariate` and use the package citation:

```
##
## To cite package 'texmex' in publications use:
##
##   Harry Southworth, Janet E. Heffernan and Paul D. Metcalfe
##   (2016). texmex: Statistical modelling of extreme values. R
##   package version 2.3.
##
## A BibTeX entry for LaTeX users is
##
##   @Manual{,
##     title = {texmex: Statistical modelling of extreme values},
##     author = {Harry Southworth and Janet E. Heffernan and Paul D. Metcalfe},
##     year = {2016},
##     note = {R package version 2.3},
##   }
```

1.1 Preliminaries

With `texmex` installed, use the `library` command to make the package available to the current session, set the colours used for graphics, and set the random seed so that results are reproducible on a given machine:

```
library(texmex)
#palette(c("black","purple","cyan","orange"))
set.seed(20130618)
```

1.2 Data

The dataset used in this example analysis is contained in the `texmex` package. This vignette reproduces some of the analysis presented in Heffernan and Tawn (2004) [2], describing the extremal behaviour of daily maxima of hourly means of five air pollutants. We focus on the `winter` data from the months November to February inclusive:

```
head(winter)

##    O3 NO2  NO SO2 PM10
## 1 27  50 112  13   34
## 2 27  51 126  13   29
## 3 15  43  90  21   33
## 4  9  71 470  44  101
## 5 20  51 167  48   30
## 6  8  50 211  16   44

summary(winter,digits=2)

##        O3          NO2            NO           SO2           PM10
##  Min.   : 1   Min.   : 19   Min.   : 10   Min.   :  1   Min.   :  7
##  1st Qu.:10   1st Qu.: 37   1st Qu.: 64   1st Qu.:  8   1st Qu.: 29
##  Median :22   Median : 43   Median :112   Median : 15   Median : 40
##  Mean   :20   Mean   : 44   Mean   :135   Mean   : 21   Mean   : 48
##  3rd Qu.:29   3rd Qu.: 51   3rd Qu.:166   3rd Qu.: 26   3rd Qu.: 60
##  Max.   :44   Max.   :104   Max.   :568   Max.   :200   Max.   :177
```

The response variables are

O3 Daily maximum ozone in parts per billion.

NO2 Daily maximum NO2 in parts per billion.

NO Daily maximum NO in parts per billion.

SO2 Daily maximum SO2 in parts per billion.

PM10 Daily maximum PM10 in micrograms/metre^3.

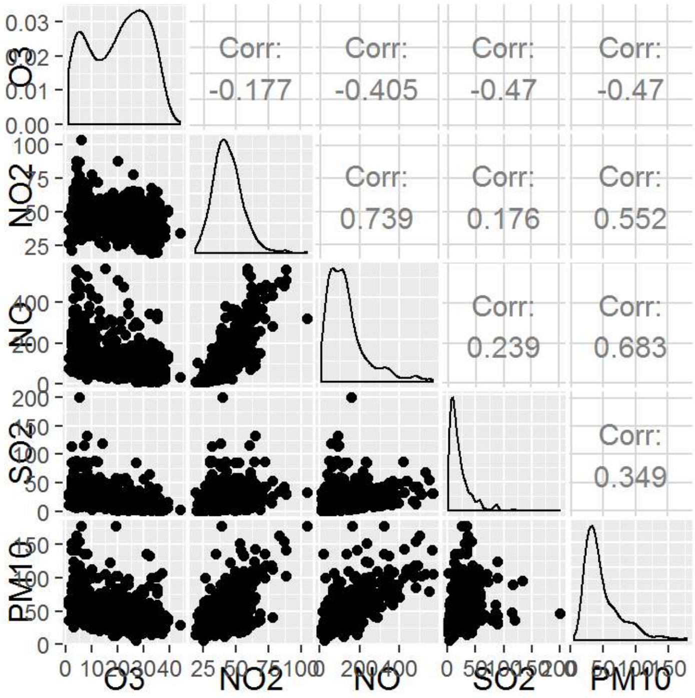
2 Exploratory multivariate modelling

Modelling of multivariate extreme values is more complicated than univariate modelling. An issue that quickly arises is how to define a multivariate extreme observation. If an observation has to be extreme in all components simultaneously, the amount of data to model quickly diminishes to numbers too small to do anything meaningful with. Moreover, dependencies between variables in the body of the data do not necessarily tell us anything at all about dependence in the extremes.

2.1 Exploratory plots

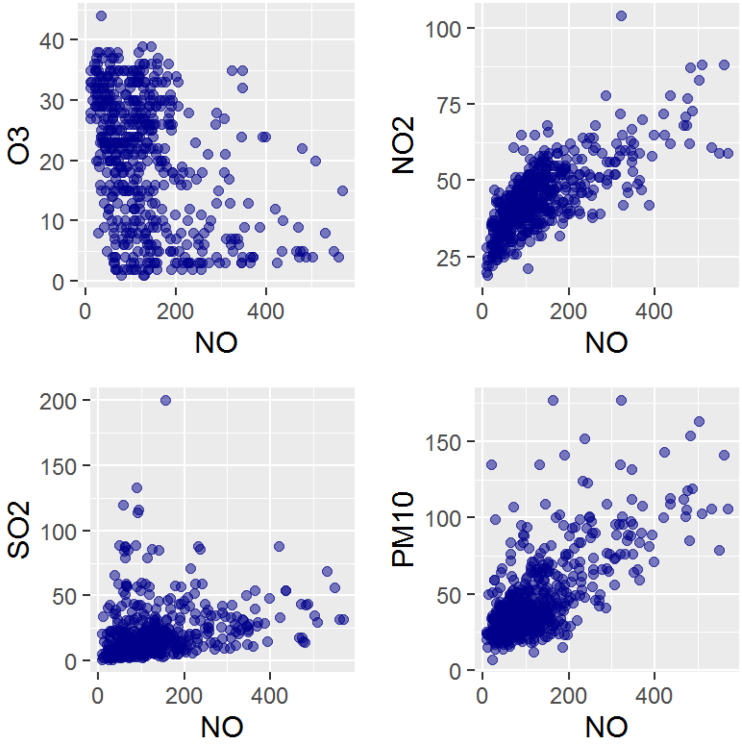
Firstly, we attempt to get a feel for the data by examining the pairwise dependence between variables. A pairwise scatterplot of the data shows some extremal dependence between the variables, the nature of which varies considerably between the pairs.

```
GGally::ggpairs(winter)
```



Next, we plot each of the other variables against NO; a full analysis would consider all pairs of variables.

```
p1 <- ggplot(winter,aes(NO,O3)) + geom_point(colour="darkblue",alpha=0.5)
p2 <- ggplot(winter,aes(NO,NO2)) + geom_point(colour="darkblue",alpha=0.5)
p3 <- ggplot(winter,aes(NO,SO2)) + geom_point(colour="darkblue",alpha=0.5)
p4 <- ggplot(winter,aes(NO,PM10)) + geom_point(colour="darkblue",alpha=0.5)
grid.arrange(p1,p2,p3,p4,ncol=2)
```

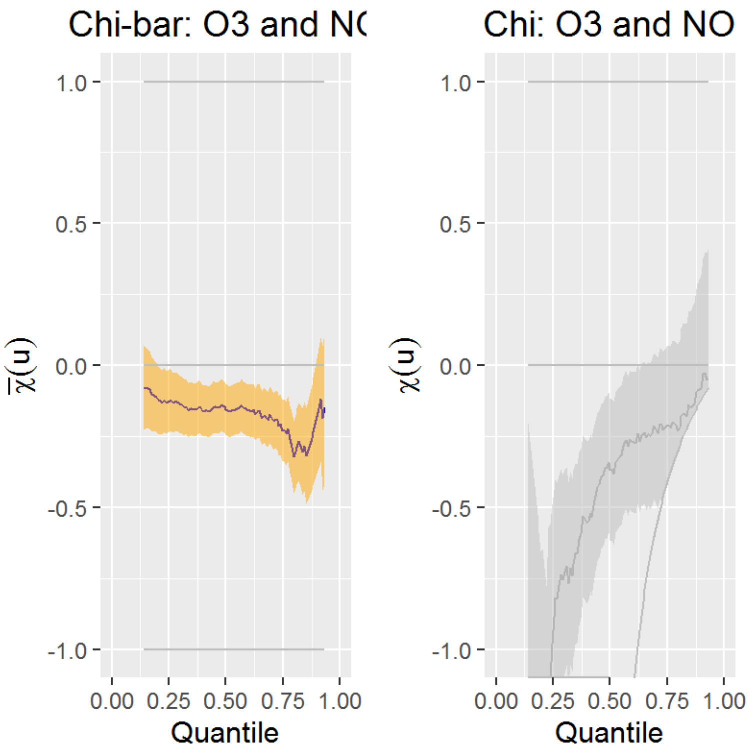


We see that the dependence between these variables differs markedly from one pair to another. Ozone (O3) appears to be negatively dependent on NO at high levels, whereas NO2 and PM10 are both clearly positively dependent at these levels, although the latter less strongly so than the former. Plotting the other pairs of variables is left as an exercise.

2.2 Exploring pairwise extremal dependence

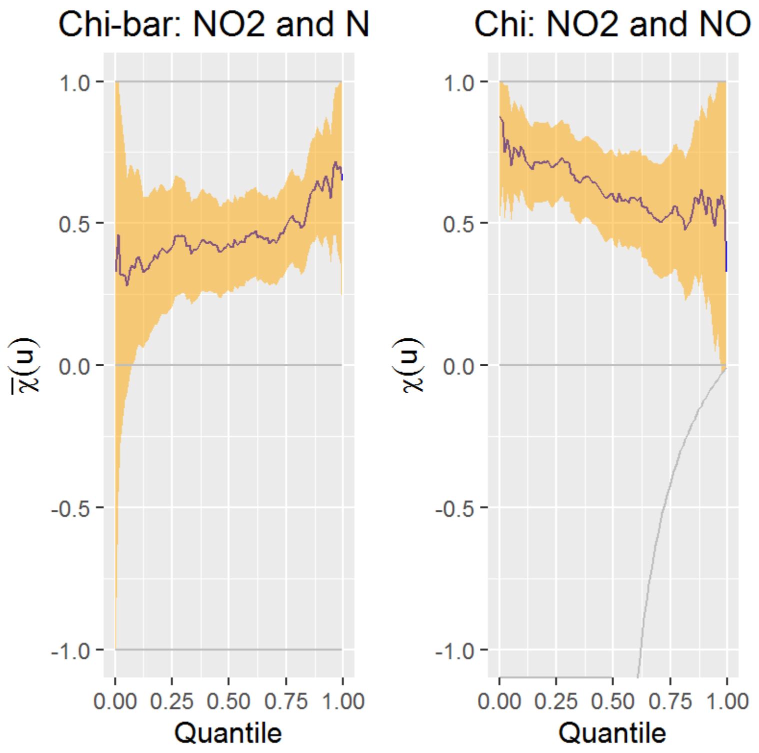
We can examine pairwise extremal dependence by plotting summary statistics χ and $\bar{\chi}$ as defined by Coles, Heffernan and Tawn [1]. Here we do so for associations only between O3 and NO, and between NO2 and NO.

```
chiO3 <- chi(winter[, c("O3", "NO")])
ggplot(chiO3, main=c("Chi"="Chi: O3 and NO",
                     "ChiBar"="Chi-bar: O3 and NO"))
```



Note that here the second plot is greyed out – this is done because the confidence interval for the limiting value of $\bar{\chi}$ as the quantile tends to 1 excludes 1. This is evidence of asymptotic independence, in which case the plot of χ is not relevant – since this shows the level of dependence only within the asymptotic dependence class.

```
chiNO2 <- chi(winter[, c("NO2", "NO")])
ggplot(chiNO2, main=c("Chi"="Chi: NO2 and NO",
                      "ChiBar"="Chi-bar: NO2 and NO"))
```



The plots are interpreted as follows:

a. Look at limiting value of $\bar{\chi}(u)$ plot as the quantile u tends to 1 . This gives a diagnostic as to whether the data exhibit asymptotic dependence (the very largest values of each variable tend to occur in the same observation). A limiting value of 1 is indicative of asymptotic dependence.

b. If limit in a. is equal to 1 examine plot of $\chi(u)$ for a measure of the strength of dependence within the asymptotic dependence class. The limiting value of this function as the quantile $u \to 1$ tells us about the strength of this dependence, with values closer to 1 indicating stronger dependence.

c. If limit in a. is less than 1 examine plot of $\bar{\chi}(u)$ for a measure of the strength of dependence within the asymptotic independence class. Although at asymptotic levels, the largest values of the variables tend not to occur in the same observation, at moderately extreme levels, dependence may still be relatively strong. The limiting value of this function as $u \to 1$ tells us about the strength of this dependence, with positive values closer to 1 indicating stronger positive dependence and negative values closer to -1 indicating stronger negative dependence. Values close to 0 indicate asymptotic near independence.

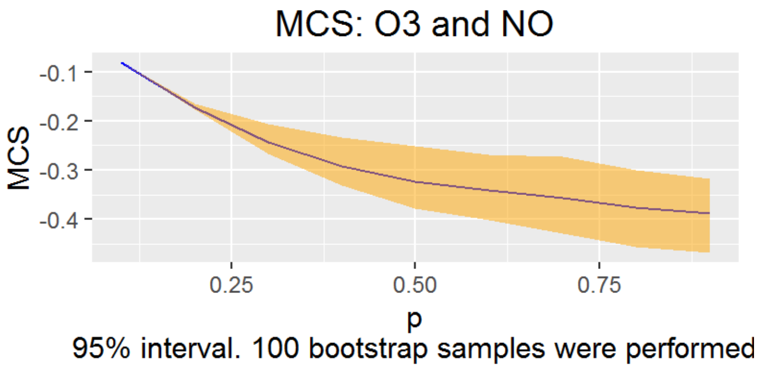
The $\bar{\chi}$ for O3 and NO shows that these variables are likely to be asymptotically independent, with weak negative dependence within this class. We do not examine the χ plot for this pair (and hence the χ plot is automatically greyed out). For NO2 and NO, the $\bar{\chi}$ plot rises towards the right, and includes 1 as a possble limit indicating possible asymptotic dependence. The χ plot indicates moderate positive dependence within this class.

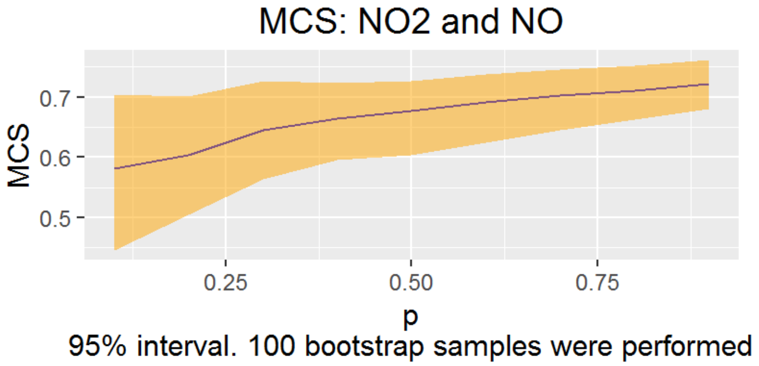
An alternative approach to examining pairwise extremal dependence is to examine the multivariate conditional Spearman's correlation coefficient across a sliding window of values of the variables, following Schmidt and Schmitt [6]. This is carried out as follows (output not shown):

```
mcsO3 <- MCS(winter[, c("O3", "NO")])
mcsNO2 <- MCS(winter[, c("NO2", "NO")])
g1 <- ggplot(mcsO3, main="MCS: O3 and NO")
g2 <- ggplot(mcsNO2, main="MCS: NO2 and NO")
gridExtra::grid.arrange(g1,g2,ncol=1)
```

Confidence intervals can be added to the MCS plots by using `bootMCS` and its associated plot method as follows:

```
bootmcsO3 <- bootMCS(winter[, c("O3", "NO")],trace=1000)
bootmcsNO2 <- bootMCS(winter[, c("NO2", "NO")],trace=1000)
g1 <- ggplot(bootmcsO3, main="MCS: O3 and NO")
g2 <- ggplot(bootmcsNO2, main="MCS: NO2 and NO")
gridExtra::grid.arrange(g1,g2,ncol=1)
```





The plots of the multivariate conditional Spearman's ρ do not have the same vertical axes, and tell a similar story to the plots of χ and $\bar{\chi}$. The exploratory summaries of this section suggest that when we come to the conditional multivariate extreme value modelling, we should expect to find a negative association between extreme O3 and extreme NO, and a possibly stronger positive association between NO2 and NO. The reader is left to check the other pairs of variables and to look at the analogous dependence in the `summer` dataset, which is not the same.

3 Conditional multivariate extreme value modelling

The conditional multivariate approach of Heffernan and Tawn proceeds by first fitting Generalised Pareto distribution (GPD) models to the marginal variables, then estimating the dependence structure. For more details on the marginal modelling by using the Generalised Pareto distribution, see the `texmex` vignette `texmex1d`. Like the GPD model for excesses above a threshold, the dependence component of the Heffernan and Tawn model also conditions on a variable exceeding a threshold. It then seeks to describe the conditional distribution of the remaining variables given the threshold excess by the first variable, using a regression type model. Uncertainty in the parameters in the dependence structure can be characterized via a bootstrap scheme.

3.1 Marginal transformation

The structure of the regression type dependence model is defined not on the original data scale, but after marginal transformation to standardised margins. In the original implementation, Heffernan and Tawn used a transformation to Gumbel margins but subsequent developments (see [3]) in this area show the structure of the regression model to be greatly simplified if Laplace margins are used instead. The package `texmex` implements both and correspondingly we describe both here. Let $\boldsymbol{X} = (X_1, \ldots, X_d)$ be a d dimensional random variable with arbitrary marginal distributions. Let $\hat{F}_i$ denote an estimate of the ith marginal distribution function $(i = 1, \ldots, d)$ and let G denote the distribution function of the standardised marginal distribution, to be determined. The original vector variable $\boldsymbol{X}$ is transfromed to $\boldsymbol{Y} = (Y_1, \ldots, Y_d)$, a variable having standardised marginal distributions by using the *probability integral transform* as follows:

$$Y_i \quad = \quad (G^{-1}(\hat{F}_i(X_i)), i = 1, \ldots, d. \tag{1}$$

In practice, the $\hat{F}_i$ can be the marginal empirical distribution functions of the data (in which case Equation (1) is also known as the *rank transform*), or the semi-parametric model using the empirical distributions below a threshold and the fitted GPD models for the tails of the distributions above the threshold.

3.1.1 Regression model structure

Let $Y_i, i \in \{1, \ldots, d\}$, be the variable on which we are to condition. Then $\boldsymbol{Y}_{-i}$ denotes the remainder of the vector $\boldsymbol{Y}$ excluding the ith component. The Heffernan and Tawn approach conditions on Y_i being above some high threshold t, and models the dependence of the remaining $\boldsymbol{Y}_{-i}$ conditional on the observed value of $Y_i > t$. The form of the regression model for the conditional dependence structure depends on the precise choice of G in Equation (1).

Laplace margins G is the Laplace distribution function and $\boldsymbol{Y}$ are marginally Laplace distributed. Conditional on variable Y_i exceeding a high threshold t, the Heffernan and Tawn model for the remaining variables $\boldsymbol{Y}_{-i}$ takes the form:

$$\boldsymbol{Y}_{-i} = \boldsymbol{\alpha}_{|i} Y_i + (Y_i)^{\boldsymbol{\beta}_{|i}} \boldsymbol{Z}_{|i} \tag{2}$$

where $\boldsymbol{Z}_{|i}$ is a vector residual and $(d-1)$ dimensional parameter vectors $\boldsymbol{\alpha}_{|i}$ and $\boldsymbol{\beta}_{|i}$ satisfy $(\boldsymbol{\alpha}_{|i}, \boldsymbol{\beta}_{|i}) \in [-1,1]^{d-1} \times (-\infty, 1)^{d-1}$. Here, $\alpha_{j|i}$, the $\boldsymbol{\alpha}_{|i}$ associated with $Y_j, (j \in \{1, \ldots, d\}, j \neq i)$, then $0 < \alpha_{j|i} \leq 1$ and $-1 \leq \alpha_{j|i} < 0$ correspond respectively to positive and negative association between Y_j and large values of Y_i.

Gumbel margins G is the Gumbel distribution function and $\boldsymbol{Y}$ are marginally Gumbel distributed. Conditional on variable Y_i exceeding a high threshold t, the Heffernan and Tawn model for the remaining variables $\boldsymbol{Y}_{-i}$ takes the form:

$$\boldsymbol{Y}_{-i} = \boldsymbol{\alpha}_{|i} Y_i + I_{\boldsymbol{\alpha}_{|i}=0, \boldsymbol{\beta}_{|i}<0} (\boldsymbol{c}_{|i} - \boldsymbol{d}_{|i} \log Y_i) + (Y_i)^{\boldsymbol{\beta}_{|i}} \boldsymbol{Z}_{|i} \tag{3}$$

where $\boldsymbol{Z}_{|i}$ is a vector residual and $(d-1)$ dimensional parameter vectors $\boldsymbol{\alpha}_{|i}$, $\boldsymbol{\beta}_{|i}$, $\boldsymbol{c}_{|i}$ and $\boldsymbol{d}_{|i}$ this time satisfy $(\boldsymbol{\alpha}_{|i}, \boldsymbol{\beta}_{|i}, \boldsymbol{c}_{|i}, \boldsymbol{d}_{|i}) \in [0,1]^{d-1} \times (-\infty, 1)^{d-1} \times (\infty, \infty)^{d-1} \times (0,1)^{d-1}$. Here positive association between Y_j and large Y_i is described by $\alpha_{j|i}$, when both $\alpha_{j|i} > 0$ and $\beta_{j|i} < 0$. The model structure changes in the case of negative dependence in which case $\alpha_{j|i} = 0$ and further parameters $c_{j|i}$ and $d_{j|i}$ are required.

The structure of the dependence model is greatly simplified under the use of Laplace margins, in which case a single model structure suffices to describe both positive and negative dependence. This makes inference considerably more straightforward, particularly in the case of weak dependence.

Note that in both Laplace and Gumbel cases, there is no parametric family of distributions assumed to describe the distribution of model residuals $\boldsymbol{Z}_{|i}$. Thus the Heffernan and Tawn conditional dependence model is semi-parametric. For a complete description of the dependence between conditioning variable Y_i and the remaining variables $\boldsymbol{Y}_{-i}$, we need both the parametric regression type model (either (2) or (3)) and the distribution of the model residuals $\boldsymbol{Z}_{|i}$, the latter being modelled by the empirical distribution of observed model residuals. These model residuals are calculated by using transformed data $\boldsymbol{Y}$ and estimates of model parameters $\hat{\boldsymbol{\alpha}}, \hat{\boldsymbol{\beta}}$ (and possibly also $\hat{\boldsymbol{c}}$ and $\hat{\boldsymbol{d}}$) in (2) or (3).

3.2 Constraints on parameter space

Recent developments to the Heffernan and Tawn method, [3] address the issue of validity of the fitted model. This work shows that in order for the fitted model to be valid, it is necessary impose tighter constraints on the parameters of the Heffernan and Tawn model than the originl box constraints described above. Constraints suggested by Keef *et al.* enforce the consistency of the fitted dependence model with the strength of extremal dependence exhibited by the data.

The effect of these constraints is to limit the shape of the dependence parameter space so that its boundary is curved. The constraint brings with it some performance issues for the optimiser used to estimate the dependence parameters, in particular sensitivity to choice of starting value.

In `texmex`, this constrained estimation is implemented for Laplace margins only. It is to be preferred to the use of unconstrained estimation which can result in invalid, inconsistent inferences and which can lead to misleading predictions particularly if extrapolation is to be made far into the tail of the modelled distribution. As such, the package defaults are to use Laplace margins and to constrain the parameters to give valid fitted models. Diagnostic plots to visualise this constrained parameter space are provided: see examples below in Section 4.4, page 16.

4 Conditional multivariate extreme value modelling using `texmex`

The whole conditional multivariate extreme value modelling algorithm is rather complicated. Fitted models are arguably most easily interpreted by using them to predict quantities of interest.

4.1 Model fitting

Now we fit the multivariate model to the winter dataset, conditioning on each of the five marginal variables in turn. Here, `mqu` specifies the marginal quantile which defines the threshold above which the marginal GPD models will be fitted.

```
mex.O3   <- mex(winter, mqu=.7, penalty="none", which="O3")
mex.NO2 <- mex(winter, mqu=.7, penalty="none", which="NO2")
mex.NO   <- mex(winter, mqu=.7, penalty="none", which="NO")
mex.SO2 <- mex(winter, mqu=.7, penalty="none", which="SO2")
mex.PM10 <- mex(winter, mqu=.7, penalty="none", which="PM10")
```

The function `mex` is a wrapper for the functions `migpd` and `mexDependence` which carry out the marginal and dependence modelling stages respectively. An equivalent way of carrying out the above, conditioning on O3 would be to use:

```
marg <- migpd(winter, mqu=0.7, penalty="none")
mex.O3 <- mexDependence(marg, which = "O3")
```

This would be a more efficient way to fit the above models, as it does the GPD estimation only once, whereas this was repeated for each of the different conditioning variables in the preceding code chunk. By default, if no dependence threshold is supplied, the threshold for fitting the dependence component of the model is taken to be equal to that used to fit the GPD model to the tail of the conditioning variable, and a warning message is given. There is, however, no reason why the thresholds employed for marginal and dependence modelling should be the same, and there is no required ordering on the two types of threshold. Different thresholds can be used for marginal and dependence modelling, by specifying the quantile `dqu` to be used for the dependence threshold:

```r
mexDependence(marg, which = "O3", dqu=0.8)
```

4.2 Marginal model diagnostics

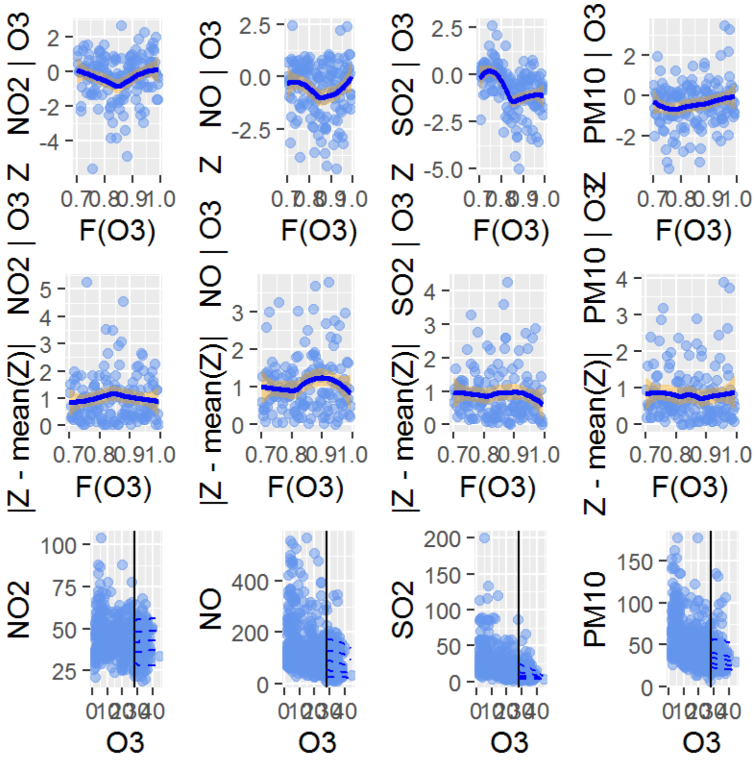
We can check the diagnostics for the fitted marginal models in the usual way. Use of `mrlPlot` and `gpdRangeFit` can also be informative at this stage (see `texmex1d` vignette for more details of these univariate methods - here output is suppressed since it is lengthy!).

```r
g <- ggplot(marg)

do.call("grid.arrange", c(g[[1]], list(ncol=2, nrow=2)))  # ... etc
do.call("grid.arrange", c(g[[2]], list(ncol=2, nrow=2)))  # ... etc

ggplot(gpdRangeFit(winter$O3))  # ... etc
ggplot(mrl(winter$O3))          # ... etc
```

4.3 Dependence model diagnostics

Plotting model diagnostics for the dependence component of the model is carried out as follows - first, for the model fitted by conditioning on the O3 variable:

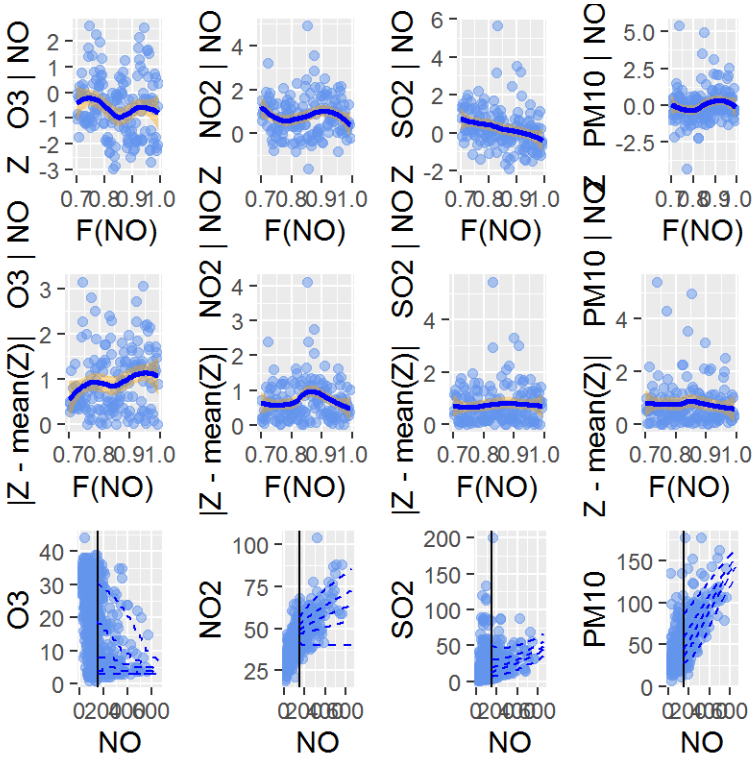
```r
ggplot(mex.O3)
```



The plots show (top to bottom): centred and scaled values of the dependence model residuals across the range of the extreme conditioning variable; absolute values of these; and the original untransformed data with contours showing quantiles of the fitted conditional model. If the model fits the data, the top and centre rows of the plots should show no structure with scatterplot smoothers being more or less horizontal. In the bottom row, the fitted quantiles should agree with the shape of the raw data distribution. Take care to note that the one dimensional conditional distribution of $(X_j \mid X_i)$ (whose estimated quantiles at each value of X_i are shown by the contours) is *not* the same thing as the (two dimensional) joint distribution of the (X_i, X_j), estimated by the scatterplot of the data points.

For the models fitted by conditioning on the NO variable, we do:

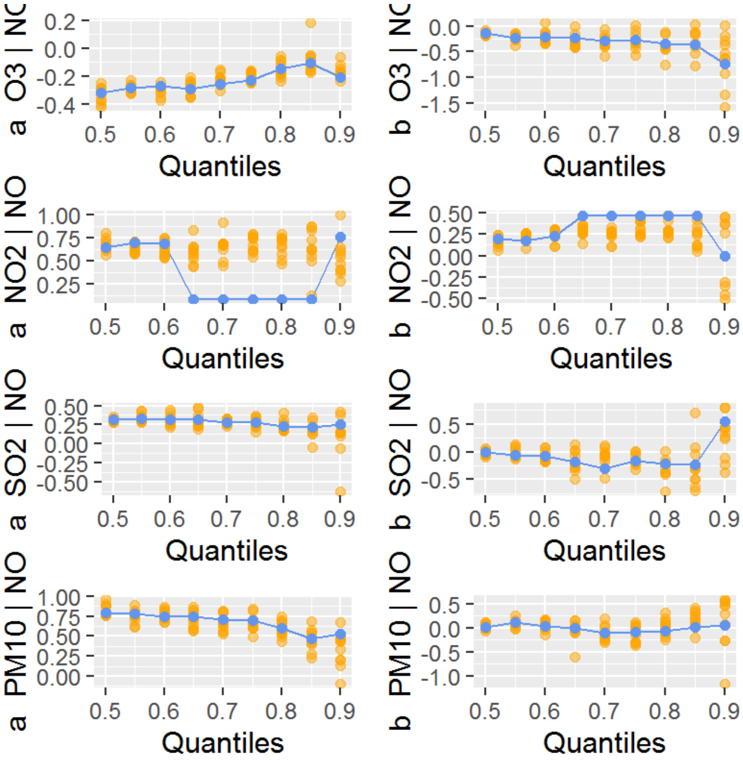
```
ggplot(mex.NO)
```



Most of the plots support the choice of threshold, however the top plot for SO2 given NO shows a decrease in location with increase in conditioning NO.

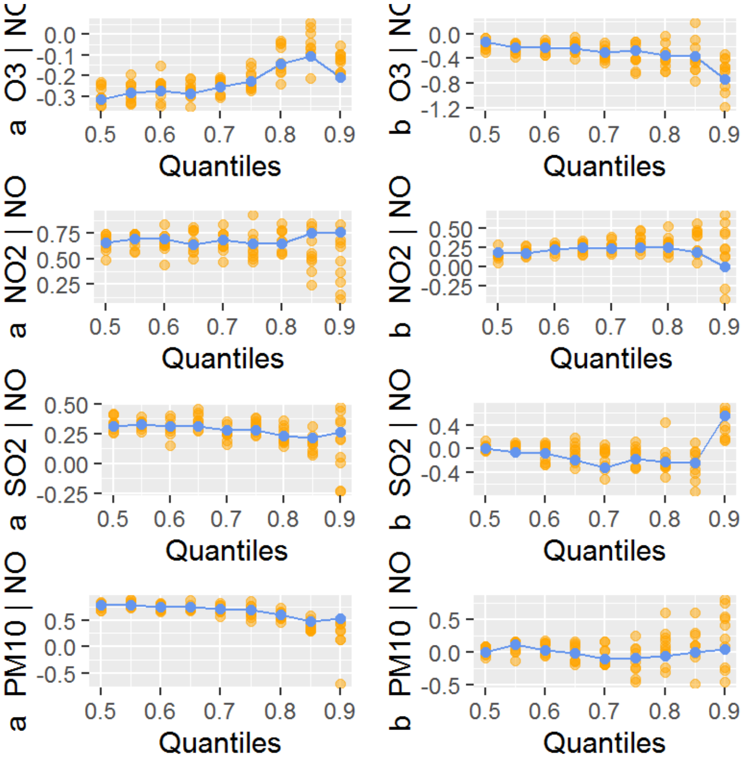
We can investigate further by plotting the dependence structure parameter estimates across a range of thresholds. Beyond a suitably high threshold, we should expect the parameters to be constant. To gain some feeling for the variability in the parameters, we perform 10 (by default) bootstrap samples. We set `trace=11` to suppress printing of progress reports in this document (the default is to report every ten replicates).

```
mrf <- mexRangeFit(marg, "NO", trace=11)
ggplot(mrf)
```



These plots suggest that there is an issue with the starting values in the NO2 model fit. We try using different starting values. There is an option for using a previously fitted dependence model as a starting point, see the documentation for `mexDependence`.

```
start <- coef(mex.NO$dependence)[1:2,] # alternative starting value

mrf <- mexRangeFit(marg, "NO", trace=11,start=c(0.1,0.1))
ggplot(mrf)
```



This does appear to have resolved the issue about the starting value which we had identified. We would need to take care in other fitting that the fitted model for NO2 given NO does not suffer from this issue.

The stability of the parameter estimates in the resulting plot provides some reassurance that the 70^{th} percentile is a suitable threshold.

4.4 Constrained parameter space

Before carrying on to examine our fitted models or to use them for prediction, we need to take some care to make sure our parameter estimates do correspond to the true maximum of the objective functions used for estimation. This is an issue since the performance of the optimiser can be sensitive to the choice of starting value. It is up to the user to check that the parameter estimates have converged to the true maximum likelihood estimates. This is carried out straightforwardly using simple visual diagnostics.

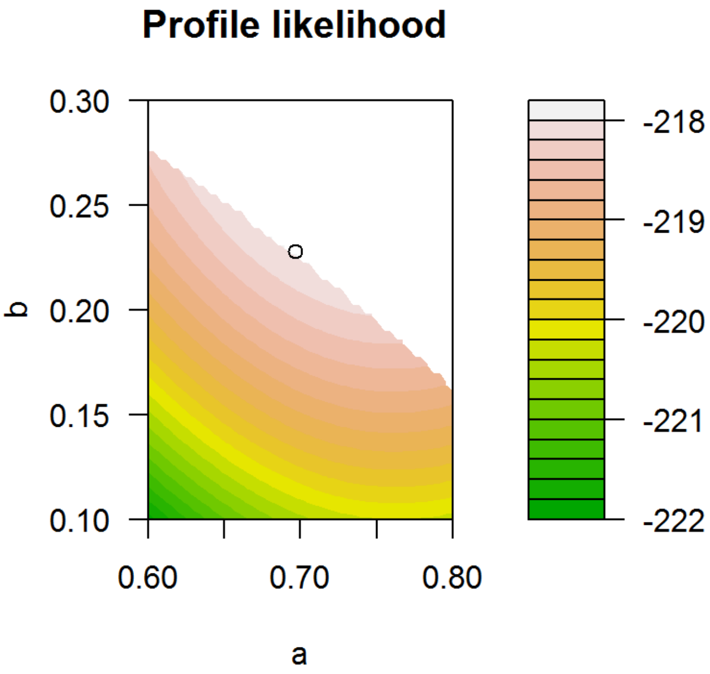
To reduce the amount of output produced, here we show the procedure only for NO2 given NO. We use `mexDependence` to plot the profile-likelihood surface which is maximised for estimation of the dependence model parameters.

```
par(mfrow=c(3,4), mar=par("mar")/2)
marg.NO2.NO <- migpd(winter[,c("NO2","NO")],mqu=0.7)
```

```
mex.NO2.NO <- mexDependence(marg.NO2.NO, which = "NO",
                            dqu=0.7, PlotLikDo=TRUE)
```

This plot shows the point estimate to lie on the edge of the permissible parameter space, and we can home in on the region containing this estimate to check that the surface has been successfully maximised:

```
par(mfrow=c(1,1))
mex.NO2.NO <- mexDependence(marg.NO2.NO, which = "NO",
                            dqu=0.7, PlotLikDo=TRUE,
                            PlotLikRange=list(a=c(0.6,0.8),b=c(0.1,0.3) ))
```



This plot reassures us that the point estimate does correspond to the maximum of the objective function. If this had not been the case, we should have tried a range of different starting values for the optimisation. More details are given in the documentation for `mexDependence`.

It is left as an exercise to produce plots for all of the conditional models fitted in this section here, for example:

```
mexDependence(marg,which="O3",dqu=0.7,PlotLikDo=TRUE)
```

4.5 Fitted model parameters

Now that we are satisfied with the fit of our model, we can examine the estimated model parameters. The parameters in the dependence structure are not straighforwardly interpretable, though values of `a` close to 1 (or -1) indicate strong positive (or negative) extremal dependence.

```
mex.O3

## mexDependence(x = marg, which = "O3")
##
##
## Marginal models:
##
## A collection of 5 generalized Pareto models.
## All models converged.
## Penalty to the likelihood: none
##
## Summary of models:
##                       O3      NO2         NO     SO2     PM10
## Threshold        28.0000  49.0000  149.00000 23.0000  53.0000
## P(X < threshold)  0.7000   0.7000    0.70000  0.7000   0.7000
## sigma             6.2303   9.3145  118.69843 19.6826  37.5644
## xi               -0.3693  -0.0279   -0.09549  0.1059  -0.2067
## Upper end point  44.8716 382.8796 1392.06236     Inf 234.7397
##
##
## Dependence model:
##
## Conditioning on O3 variable.
## Thresholding quantiles for transformed data: dqu = 0.7
## Using laplace margins for dependence estimation.
## Constrained estimation of dependence parameters using v = 10 .
## Log-likelihood = -257.702 -256.6681 -231.7684 -234.0916
##
## Dependence structure parameter estimates:
##       NO2       NO     SO2     PM10
## a 0.01301 -0.07278 -0.1683 -0.04719
## b 0.02020  0.03038 -0.1418  0.07142
```

It is clear from the values of the dependence parameters, that SO2 is the most strongly (negatively) dependent on large values of O3, with the other variables having only weak extremal dependence on ozone.

```
mex.NO

## mexDependence(x = marg, which = "NO", dqu = 0.7, start = c(0.1,
##     0.1))
##
##
## Marginal models:
##
## A collection of 5 generalized Pareto models.
## All models converged.
## Penalty to the likelihood: none
##
## Summary of models:
##                        O3      NO2         NO     SO2     PM10
## Threshold        28.0000  49.0000  149.00000 23.0000  53.0000
## P(X < threshold)  0.7000   0.7000    0.70000  0.7000   0.7000
## sigma             6.2303   9.3145  118.69843 19.6826  37.5644
## xi               -0.3693  -0.0279   -0.09549  0.1059  -0.2067
## Upper end point  44.8716 382.8796 1392.06236     Inf 234.7397
##
##
## Dependence model:
##
## Conditioning on NO variable.
## Thresholding quantiles for transformed data: dqu = 0.7
## Using laplace margins for dependence estimation.
## Constrained estimation of dependence parameters using v = 10 .
## Log-likelihood = -230.9793 -218.0096 -220.2708 -243.5154
##
## Dependence structure parameter estimates:
##        O3    NO2     SO2     PM10
## a -0.2558 0.6850  0.2851  0.70840
## b -0.2986 0.2335 -0.3101 -0.09802
```

The values of the estimated dependence parameters show that NO2, SO2 and PM10 all have positive extremal dependence on NO, the strongest being that of PM10 on NO. Ozone has fairly weak negative dependence on NO.

4.6 Prediction under the fitted model

The dependence between pairs of variables is described by a pair of parameters (a, b) and also the associated empirical distribution of the residuals $\boldsymbol{Z}_{|i}$. For this reason, the interpretation of the fitted models is arguably most straightforward via prediction of variables given extreme values of the conditioning variable, which we cover now.

Comparison of the plots of the remaining variables against NO reveals that the extremal dependence between the variables varies considerably (see plot on page 4).

We can obtain predictions under the fitted conditional multivariate model by importance sampling using the `predict` method. We tell the function to simulate values of the variables conditional on `NO` being above its 90^{th} percentile.

```
set.seed(20130619)
nsim <- 1000
pO3 <- predict(mex.O3, pqu=.9, nsim=nsim)
pNO2 <- predict(mex.NO2, pqu=.9, nsim=nsim)
pNO <- predict(mex.NO, pqu=.9, nsim=nsim)
pSO2 <- predict(mex.SO2, pqu=.9, nsim=nsim)
pPM10 <- predict(mex.PM10, pqu=.9, nsim=nsim)
```

The resulting conditional distributions are summarised as follows:

```
summary(pO3)

## predict.mex(object = mex.O3, pqu = 0.9, nsim = nsim)
##
## Conditioned on O3 being above its 90th percentile.
##
##
## Conditional Mean and Quantiles:
##
##      O3|O3>Q90 NO2|O3>Q90 NO|O3>Q90 SO2|O3>Q90 PM10|O3>Q90
## mean      36.6       42.7      90.8       12.4        37.6
## 5%        33.8       26.0      18.0        3.0        20.0
## 50%       36.1       43.0      80.0       10.0        31.0
## 95%       41.2       58.7     187.0       26.7        85.1
##
## Conditional probability of threshold exceedance:
##
##  P(O3>28|O3>Q90) P(NO2>49|O3>Q90) P(NO>149|O3>Q90) P(SO2>23|O3>Q90)
##                1            0.279            0.154            0.093
##  P(PM10>53|O3>Q90)
##               0.11
```

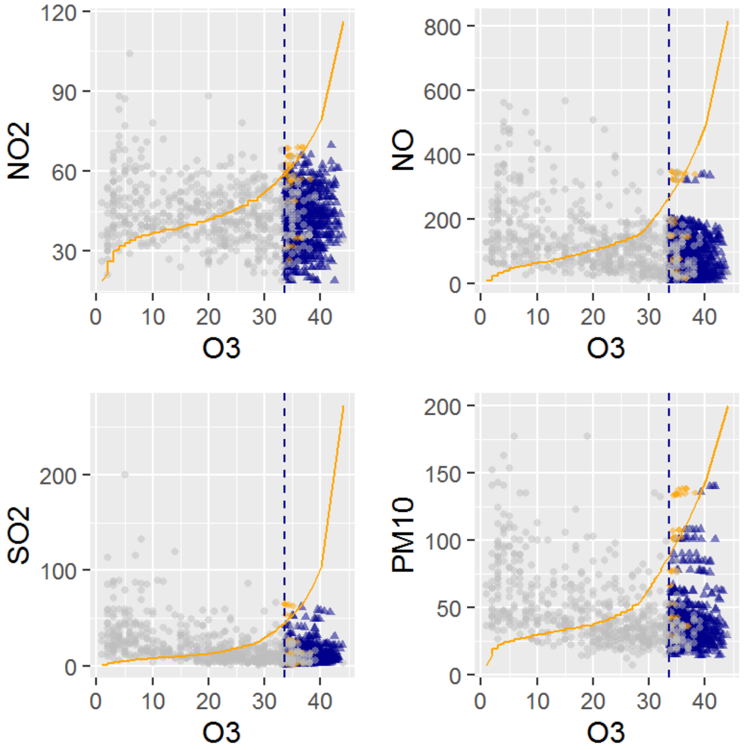
The thresholds cited in the final part of the output are by default taken to be the marginal thresholds used for fitting the GPD models (in this case these are the 0.7 quantiles of the marginal distributions). However, any value of threshold can be used for prediction by specifying the argument `mth` of the `summary` function, for example:

```
summary(pO3,mth=c(39,40,100,10,40))

## predict.mex(object = mex.O3, pqu = 0.9, nsim = nsim)
##
```

```
## Conditioned on O3 being above its 90th percentile.
##
##
## Conditional Mean and Quantiles:
##
##      O3|O3>Q90 NO2|O3>Q90 NO|O3>Q90 SO2|O3>Q90 PM10|O3>Q90
## mean      36.6       42.7      90.8       12.4        37.6
## 5%        33.8       26.0      18.0        3.0        20.0
## 50%       36.1       43.0      80.0       10.0        31.0
## 95%       41.2       58.7     187.0       26.7        85.1
##
## Conditional probability of threshold exceedance:
##
##  P(O3>39|O3>Q90) P(NO2>40|O3>Q90) P(NO>100|O3>Q90) P(SO2>10|O3>Q90)
##            0.177            0.599            0.426            0.469
##  P(PM10>40|O3>Q90)
##              0.266
```

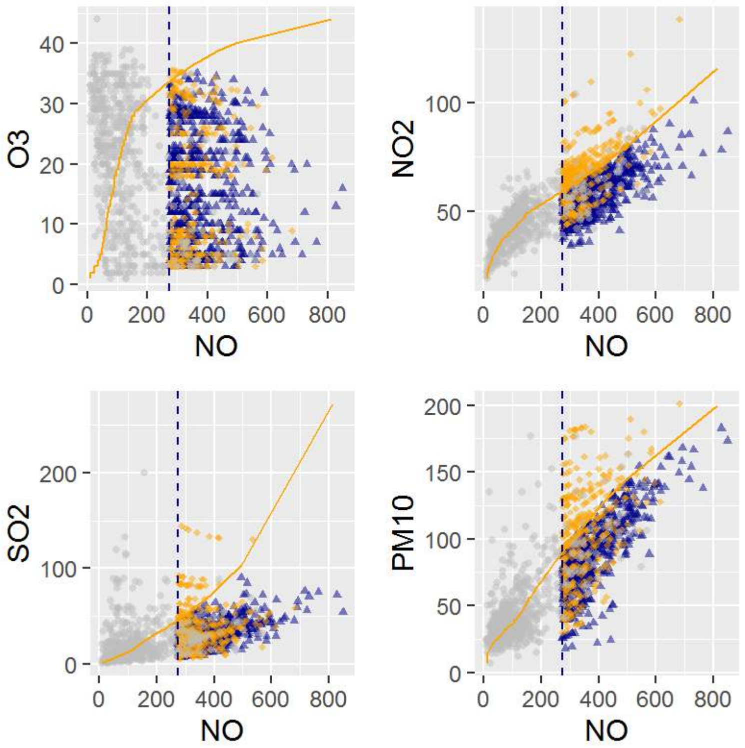
The plot method can be used to visualise the fitted conditional models using the importance samples as follows:

```
ggplot(pO3)
```



Plots show the original data (grey circles) and data importance sampled under the fitted model above the threshold for prediction (orange triangles and blue diamonds). Samples represented by a blue diamond are largest (on the common quantile scale) in the conditioning variable, orange diamonds are largest in a different variable. The solid curve in each plot is for reference and joins equal quantiles of the marginal distributions – perfectly dependent variables would lie exactly on this line (this line is analogous to the diagonal line on a QQ plot, but here since the two marginal distributions are not equal, the curve is not a straight line). We can compare the above output conditioning on O3 (which has weak or negative dependence) with that obtained when we condition on NO where the dependence is stronger:

```
ggplot(pNO)
```



The strong extremal dependence of winter PM10 on NO is evident here, with the sampled data following closely the curve of equal marginal quantiles. These plots show that the sampled points are a greater mixture of points that are largest in the conditioning variable and points that are not (there are many orange diamonds below the solid orange "diagonal" line).

The importance samples generated by the `predict` method can also be used to estimate probabilities of arbitrary tail regions falling above the threshold for the conditioning variable used for importance sampling, or to calculate functionals of the multidimensional variables. The precise implementation will depend on the application in question.

4.7 Building samples from multiple conditional models

In some applications, there is a requirement to sample from the whole of the joint distribution of the multivariate random variable, and not just from the conditional distribution given that a single component is large. This sampling approach could be taken for example for evaluating probabilities of events falling in failure sets located in arbitrary regions of the distribution's tails. The precise definition of any failulre regions will depend on the application in question. Here we show how to construct a large Monte Carlo sample from the whole of the modelled joint distribution defined by a collection of conditional models fitted by conditioning on each of the marginal variables in turn.

This process of collecting conditional models together is assumed to take place after these models have been fitted individually, including all the necessary threshold selection procedures that go with such model fitting.

We follow the example given in Heffernan and Tawn (2004) using the Winter air pollution data again. We assume that the chosen modelling thresholds for each variable have been chosen at appropriate values given in Table 4 (marginal thresholds) and page 519 (dependence thresholds) of that paper.

The winter dataset is 5 dimensional, giving 5 conditional dependence models and 5 fitted GPD models. We must use the same five fitted GPDs for each of the conditional models we fit to ensure consistency of the resulting combination. We gather our fitted models into a single R opbject:

```r
mAll <- mexAll(winter,mqu=0.7,dqu=rep(0.7,5))
```

We can then generate a Monte Carlo sample of the required size from the collection of fitted models. As in Heffernan and Tawn, we use the model that conditions on the ith component of the random vector to simulate values in that part of the sample space for which the ith component is the largest of all components (measured on a quantile scale). This is carried out as follows:

1. Generate a Monte Carlo sample from the original dataset, by sampling the required number of observations uniformly with replacement from the entire dataset;

2. Transform the Monte Carlo sample obtained in step 1. to the Laplace scale by using the fitted GPDs (here we can see why we must take care to use the same fitted GPD for all of the conditional model fits);

3. On the Laplace scale, identify which component of each transformed data point is the largest (since we are working on the common Laplace scale, this step calculates which component represents the highest marginal quantile);

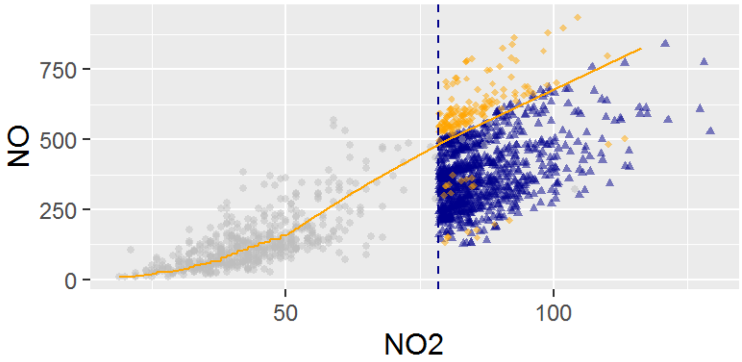
4. Identify which of our Monte Carlo sample identified in Step 3 additionally lie above the corresponding conditional dependence model threshold (for example, for all points whose ith component is the largest component, we find which of these have ith component above the dependence threshold used to fit the conditional model given the ith component is above a given threshold);

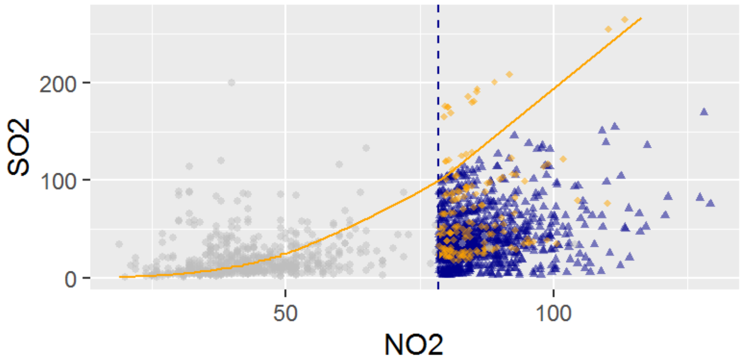
5. For each conditioning variable in turn, generate a large independent sample from the fitted conditional distribution, conditional upon being above the associated dependence model fitting threshold. This is carried out on the original scale of the data;

6. On the original scale of the data, for each variable in turn, replace those values in our Monte Carlo sample from step 1 which are both above their conditional model threshold and for which the conditioning variable is the

largest component (identified in Step 4) by a value generated from the appropriate conditional model (from step 5).

The following plots highlight the selection of points in step 4. Points fulfilling the requirements of step 4 are shown by blue dots:





The second of these plots shows the NO2 and SO2 values of all Monte Carlo samples above the conditional dependence threshold, conditioning on NO2. All of the orange diamonds and blue triangles are large in NO2, but only those shown by the blue triangles are largest in NO2 (assessed on a common quantile scale). Clearly the diamonds which lie above the solid blue line are larger in SO2 than in NO2. Those samples shown by orange diamonds *below* the solid blue line are largest in a different variable – neither NO2 nor SO2, but another variable not shown on the plot.

All the steps required for the simulation are carried out in the `texmex` function `mexMonteCarlo`. Here, we generate 5000 points from the original dataset (below the dependence thresholds) and the collection of conditional models above each of the dependence thresholds:

```
mexMC <- mexMonteCarlo(5000, mAll)
```

For each margin, the number of points from the original sample from the dataset that were replaced by points sampled parametrically from the corresponding conditional tail model is as follows:

```
mexMC$nR

##   O3  NO2   NO  SO2 PM10
## 1235  704  603  780  522
```

This shows that considerably more samples were replaced for points which had O3 as the most extreme component than for any other margin (O3 has around twice as many points replaced as any of the other margins). This corresponds to the fitted models which describe very weak or negative dependence between O3 and the remaining variables when O3 is large:

```
mAll$O3$dependence

## Conditioning on O3 variable.
## Thresholding quantiles for transformed data: dqu = 0.7
## Using laplace margins for dependence estimation.
## Constrained estimation of dependence parameters using v = 10 .
## Log-likelihood = -257.7152 -256.7165 -231.7644 -234.1655
##
## Dependence structure parameter estimates:
##       NO2       NO     SO2     PM10
## a 0.01444 -0.07238 -0.1678 -0.04756
## b 0.02122  0.03100 -0.1404  0.07132
```

A consequence of this is that when O3 is large, the other variables are not.

We can plot our large Monte Carlo sample and compare it with the original dataset which was plotted on page 3 (not shown here).

```
pairs(mexMC$MCsample)
```

There are clear limitations in using this approach to try to generate large samples from the required joint distribution:

1. The first and the most fundamental is that the taken together, the collection of conditional models do not give a consistent or even well defined joint distribution. This approach is entirey empirical and relies on the validity of the underlying joint distribution of the data which is used to estimate conditional models. We hope that these models – being estimated from the same underlying data – will reflect the underlying joint structure and therefore give approximately consistent distributions but there is no guarantee that this works in practice. Recent work by [4] has had some success in addressing this issue but is not yet implemented in `texmex`.

2. The importance of appropriate threshold choice is highlighted in this approach to combining different estimated models. Marginal and dependence

thresholds should be selected so that the transition from empirical model (e.g. below the GPD or conditional model threshold) to parametric model (above the respective thresholds) is smooth. Lack of the required continuity between the components of the resulting semi-parametric models will be revealed in Monte Carlo samples which have the appearance of failing to fit together at the joins between the component models, as indeed would be the case.

4.8 Joint exceedance curve estimation

In the univariate setting, *return level curves* show the way in which the marginal distribution of a variable extrapolates. It is useful to report the tail behaviour in terms of *return levels* associated with given *return periods*. Return periods have a simple interpretation of being the expected waiting time between events at or above the associated *return level*, or alternatively, the time interval during which we would expect to see exactly one exceedance of this level.

In two or more dimensions, there is no such simple curve. When we fix a *return period*, say 200 years, then this is equivalent to specifying a probability of observing the associated event in any one observation. For daily i.i.d. data, this probability would be $1/(365 \times 200)$. In the univarite setting, we can calculate the quantile of the fitted distribution with this exceedance probability, and this is the *return level* associated with the given return period. For two or more dimensions however, there are many joint events, involving both/all variables that occur with any given small probablility. Instead of a single value consituting our *return level*, we now need a curve that describes a set of events, all of which have the probability which was specified by our return period. For a given exceedance probability p, the *joint exceedance curve* is the set of points

$$\{(x_{1,p}, \ldots, x_{d,p}) : \Pr(X_1 > x_{1,p}, \ldots, X_d > x_{d,p}) = p\}.$$

In `texmex`, joint exceedance curve estimation is implemented for two-dimensional subsets of variables. The curve can be estimated in a variety of ways:

from the original data for relatively non-extreme curves only, within the range of observations seen in the data;

from a single fitted conditional model conditioning on one variable only being large: curves can therefore be estimated only in that part of the space in which this estimated model is defined;

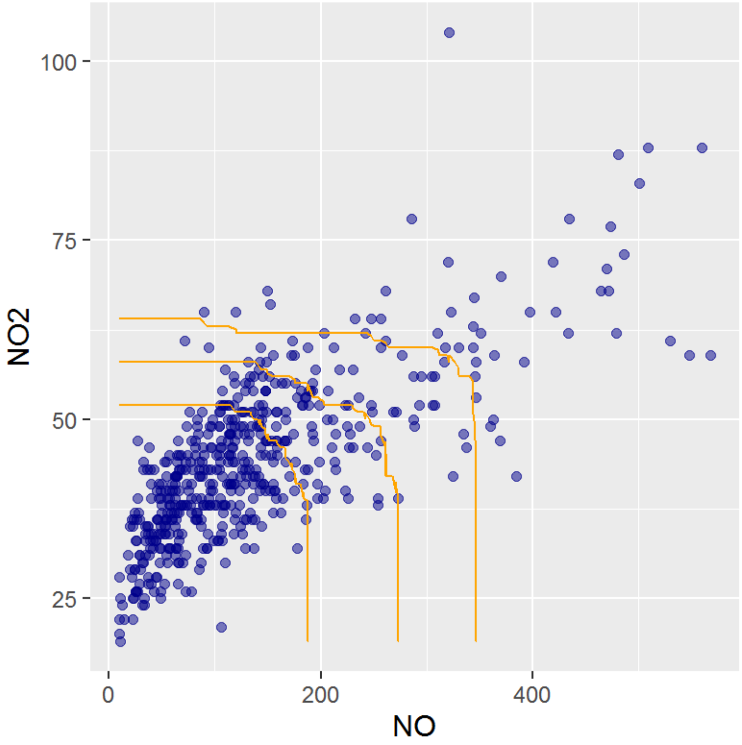
from a collection of conditional models each fitted conditional model is used to estimate that part of the joint exceedance curve in which that model's conditioning variable is largest.

We show how to do each of these types of estimation in the following sections, using the (NO2,NO) variables from the Winter dataset.

4.8.1 Joint exceedance curve directly from original data

Using the raw data only, we are not able to extrapolate beyond the levels observed in the data, as the following plot shows. The three return level curves shown correspond to constant joint exceedance probabilities of 0.2, 0.1 and 0.05. These curves are estimated empirically and so become increasingly badly estimated as we move to higher levels where there is less data.

```
WinterNO.NO2 <- winter[,3:2]
j1 <- JointExceedanceCurve(WinterNO.NO2,0.2)
j2 <- JointExceedanceCurve(WinterNO.NO2,0.1)
j3 <- JointExceedanceCurve(WinterNO.NO2,0.05)
ggplot(WinterNO.NO2,aes(NO,NO2)) +
    geom_point(colour="dark blue",alpha=0.5) +
    geom_jointExcCurve(j1,colour="orange") +
    geom_jointExcCurve(j2,colour="orange") +
    geom_jointExcCurve(j3,colour="orange")
```

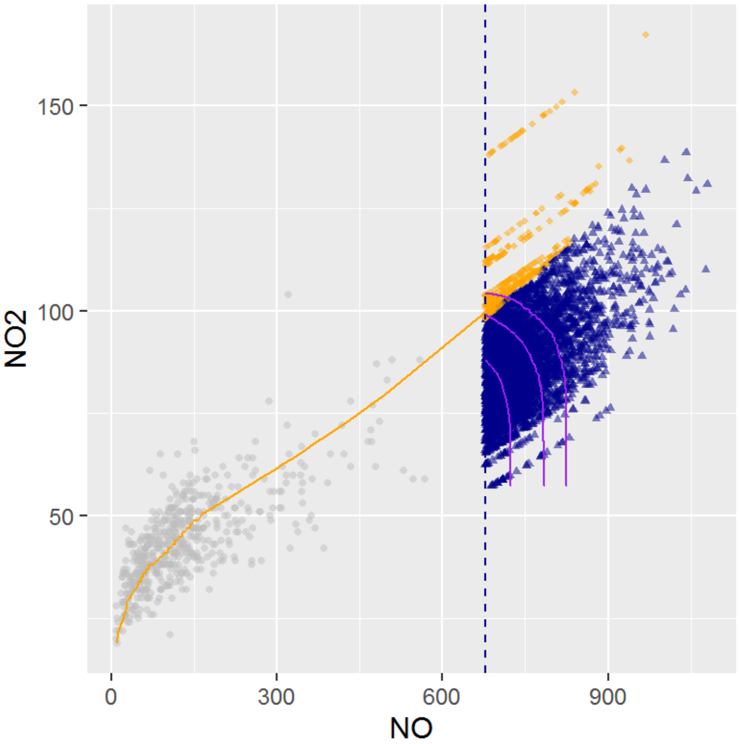


4.8.2 Joint exceedance curve from fitted conditional model: one conditioning variable only

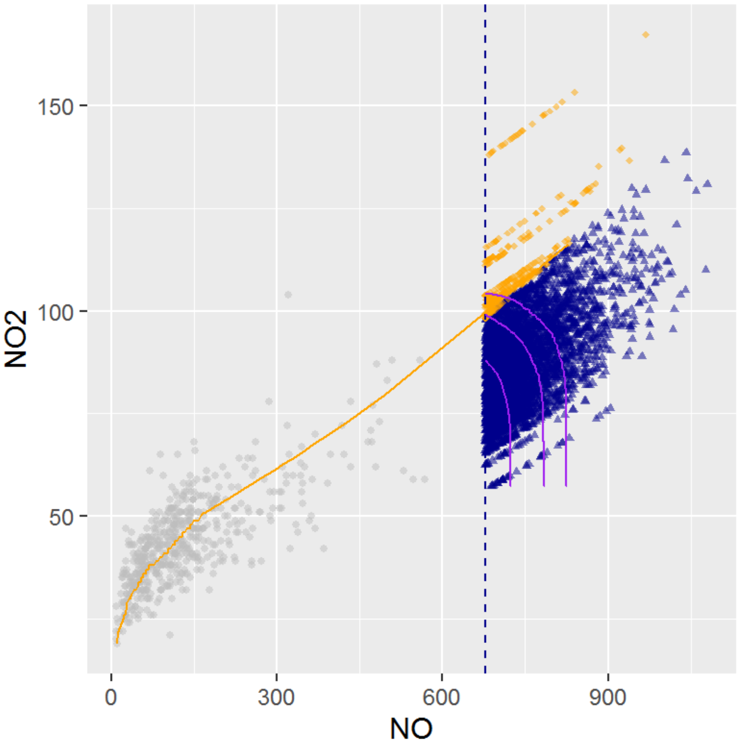
We can extrapolate further if we use the fitted conditional model, and generate importance samples from the joint tail region where we want to estimate our curve. Here we use the model for the conditional behvaiour of NO2 given extreme NO, from Section 4.4.

It is up to the user to ensure that the threshold used for generating importance samples (the argument `pqu` to the function `predict` in the following) is chosen suitably for the joint exceedance curve of interest (plot not shown):

```
p1 <- predict(mex.NO2.NO,nsim=5000,pqu=0.999)
g <- ggplot(p1,plot.=FALSE)
j4 <- JointExceedanceCurve(p1,0.0005,which=c("NO","NO2"))
j5 <- JointExceedanceCurve(p1,0.0002,which=c("NO","NO2"))
j6 <- JointExceedanceCurve(p1,0.0001,which=c("NO","NO2"))
pl <- g[[1]] +
    geom_jointExcCurve(j4,aes(NO,NO2),col="purple") +
    geom_jointExcCurve(j5,aes(NO,NO2),col="purple") +
    geom_jointExcCurve(j6,aes(NO,NO2),col="purple")
pl
```

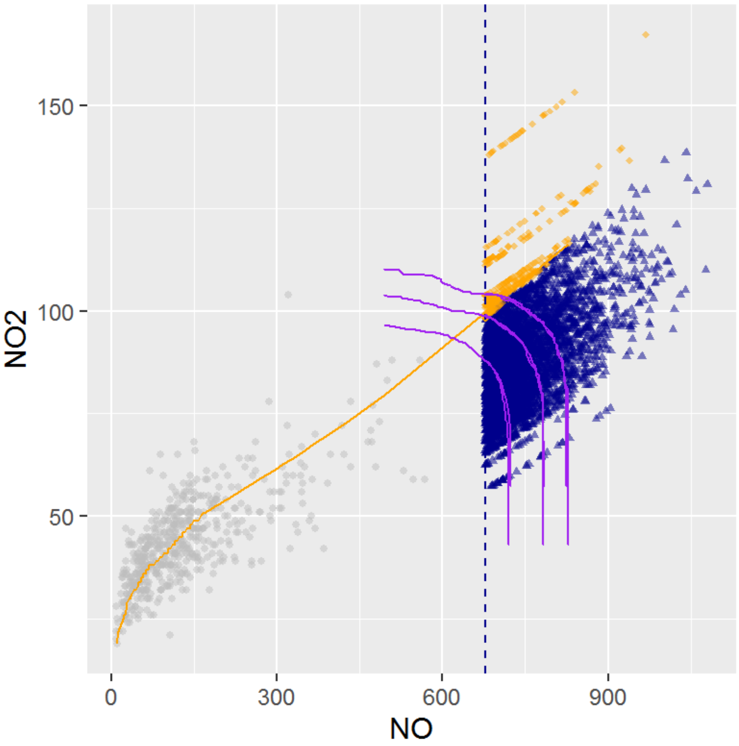


```
pl
```



If required, we can combine importance samples from more than one choice of threshold as follows:

```
p2 <- predict(mex.NO2.NO,nsim=5000,pqu=0.99)
j7 <- JointExceedanceCurve(p2,0.0005,which=c("NO","NO2"))
j8 <- JointExceedanceCurve(p2,0.0002,which=c("NO","NO2"))
j9 <- JointExceedanceCurve(p2,0.0001,which=c("NO","NO2"))

pl + geom_jointExcCurve(j7,aes(NO,NO2),col="purple") +
    geom_jointExcCurve(j8,aes(NO,NO2),col="purple") +
    geom_jointExcCurve(j9,aes(NO,NO2),col="purple")
```



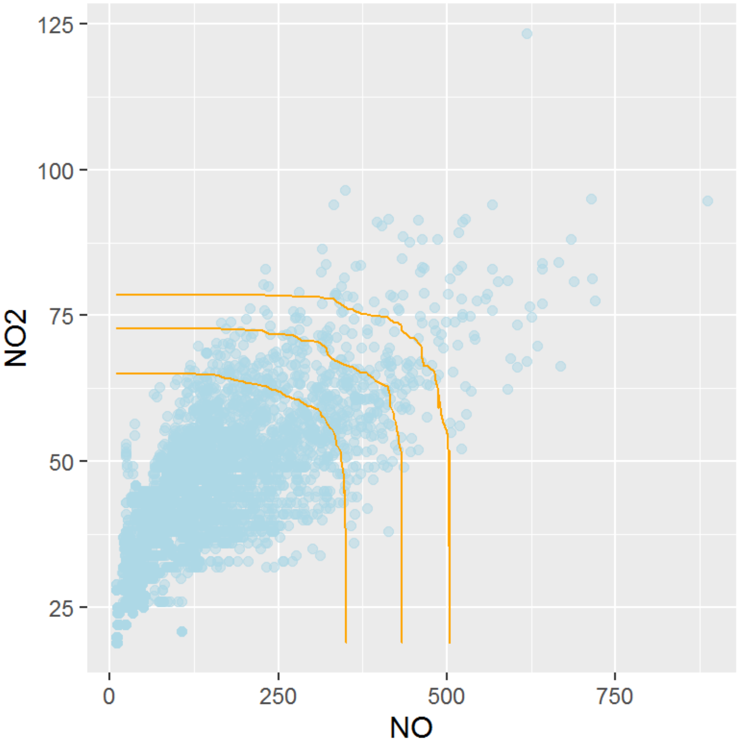
The calculated joint exceedance curves can be returned explicitly (optionally the user can specify the values of the first variable at which to report the curve, by using the argument `x` in the call to `JointExceedanceCurve` below):

```
Curve <- JointExceedanceCurve(p2,0.0005,which=c("NO","NO2"),x=seq(43,96,by=3))
Curve

##
##  Extimated curve with constant joint exceedance probability equal to 5e-04
##    NO      NO2
## 1  43 96.46665
## 2  46 96.46665
## 3  49 96.46665
## 4  52 96.46665
## 5  55 96.46665
## 6  58 96.46665
## 7  61 96.46665
## 8  64 96.46665
## 9  67 96.46665
## 10 70 96.46665
## 11 73 96.46665
## 12 76 96.46665
## 13 79 96.46665
## 14 82 96.46665
## 15 85 96.46665
## 16 88 96.46665
## 17 91 96.46665
## 18 94 96.46665
```

4.8.3 Joint exceedance curve from family of conditional models

If required, we can combine multiple conditional models fitted to each margin as conditioning variable in turn. The fitting of this set of models was demonstrated in Section 4.7, and we use the fitted models from that section, held in the object `mAll` here for sampling.

```
p3 <- mexMonteCarlo(nSample=5000,mexList=mAll)
j10 <- JointExceedanceCurve(p3,0.05,which=c("NO","NO2"))
j11 <- JointExceedanceCurve(p3,0.02,which=c("NO","NO2"))
j12 <- JointExceedanceCurve(p3,0.01,which=c("NO","NO2"))
ggplot(as.data.frame(p3$MCsample[,c("NO","NO2")]),aes(NO,NO2)) +
    geom_point(col="light blue",alpha=0.5) +
    geom_jointExcCurve(j10,aes(NO,NO2),col="orange") +
    geom_jointExcCurve(j11,aes(NO,NO2),col="orange") +
    geom_jointExcCurve(j12,aes(NO,NO2),col="orange")
```



For smaller exceedance probabilities, the size of the sample used to estimate these curves can be made arbitrarily large until the required accuracy is achieved.

Again, the precise values of points making up these curves are given in the objects returned by the call to `JointExceedanceCurve`, for example the object `j10` above gives the coordinates of the joint exceedance curve associated with an exceedance probability of 0.05. Alternatively, we can calculate the curve at user specified points `x` as follows:

```
JointExceedanceCurve(p3,0.05,which=c("NO","NO2"),x=seq(10,360,by=50))

##
##  Extimated curve with constant joint exceedance probability equal to 0.05
##     NO      NO2
## 1  10 65.08552
## 2  60 65.08559
## 3 110 65.08560
## 4 160 64.66921
## 5 210 63.37411
## 6 260 61.32037
## 7 310 58.52519
## 8 360 36.00000
```

4.8.4 Specifying return periods in terms of units of time

Throughout the package `texmex` the units of return period is the *observation*. This is because in some applications, observations may represent time periods but in others, they may represent individual patients in which case it makes no sense to talk about time scales. We have aimed to keep the package very general in its implementation.

However, in some settings it is useful to think about return levels as being associated with particular temporal return periods (or for joint exceedance curves to have a given *Annual Exceedance Probability* (AEP) of e.g. 1 in 10 years). It is trivial to convert from return periods stated in terms of numbers of observations (as implemented in the package) to years, by considering the numbers of observations in a year. For example in the `winter` air pollution example, there are 120 observations per winter (winter being defined here as November – February inclusive). So to calculate the 200 year joint exceedance curve, we carry out the following calculation:

```
ReturnPeriodInYears <- 200
NobsPerYear <- 120
ExceedanceProb <- 1/ (ReturnPeriodInYears * NobsPerYear)
ExceedanceProb

## [1] 4.166667e-05
```

```
j200 <- JointExceedanceCurve(p1,ExceedanceProb,which=c("NO","NO2"),
                             x=seq(700,by=50,len=5))
j200

##
##  Extimated curve with constant joint exceedance probability equal to 4.166667e-05
##    NO       NO2
## 1 700 110.00389
## 2 750 109.03533
## 3 800 105.95170
## 4 850  97.28181
## 5 900  76.27774
```

To plot the data and curve (not shown):

```
j200 <- JointExceedanceCurve(p1,ExceedanceProb,
                             which=c("NO","NO2"))
ggplot(WinterNO.NO2,aes(NO,NO2))+
    geom_point(colour="light blue",alpha=0.5) +
    geom_jointExcCurve(j200,aes(NO,NO2),col="purple") +
    labs(title="200 year joint exceedance curve")
```

References

[1] S. G. Coles, J. E. Heffernan, and J. A. Tawn. Dependence measures for extreme value analyses. *Extremes*, 2:339 – 365, 1999.

[2] J. E. Heffernan and J. Tawn. A conditional approach for multivariate extreme values. *Journal of the Royal Statistical Society Series B*, 56:497 – 546, 2004.

[3] C. Keef, I. Papastathopoulos, and J. A. Tawn. Estimation of the conditional distribution of a multivariate variable given that one of its components is large: Additional constraints for the heffernan and tawn model. *Journal of Multivariate Analysis*, 115:396–404, 2013.

[4] Y. Liu and J.A. Tawn. Self-consistent estimation of conditional multivariate extreme value distributions. *Journal of Multivariate Analysis*, 127:19–35, 2014.

[5] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.

[6] F. Schmid and R. Schmidt. Multivariate conditional versions of spearman's rho and related measures of tail dependence. *Journal of Multivariate Analysis*, 98:1123 – 1140, 2007.

[7] H. Southworth and J. E. Heffernan. *texmex: Threshold exceedences and multivariate extremes*, 2016. R package version 2.3.