margarita: making texmex more digestible

Harry Southworth

1 Introduction

This document illustrates some usage of the margarita package for R [1]. The purpose of the package is to provide `ggplot` [3] replacement functions for several of texmex's [2] `plot` functions, to enable simpler robust regression modelling, and to enable much easier prediction of return levels and threshold exceedance probabilities.

The functionality of margarita is illustrated by reanalysing the liver data that can be found in the texmex package. We begin by loading the package and producing the shiftplots.

```
set.seed(20140517)
library(margarita)
library(xtable)

shiftplot(liver, aes(ALT.B, ALT.M), by=~dose,
          xlab="Baseline ALT (U/L)",
          ylab="Maximum ALT (U/L)")
```

2 Robust regression

Robust regression can be performed using the `lmr` (linear model robust) function. This function is a wrapper to `rlm` in the MASS [4] package, setting the method to MM and efficiency to 85%. It also computes the coefficient covariance and attaches it to the returned object. A ggplot function for diagnostics is provided.

```
liver$ndose <- as.numeric(liver$dose)
mm <- lmr(log(ALT.M) ~ log(ALT.B) + ndose, data=liver)
ggplot(mm)
```

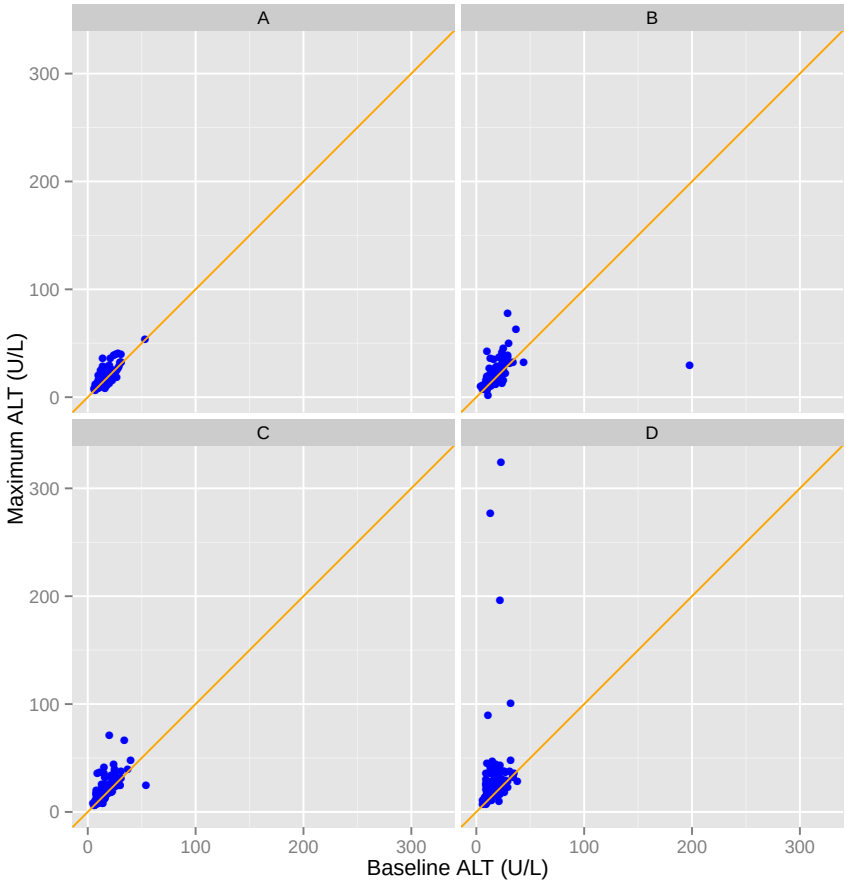
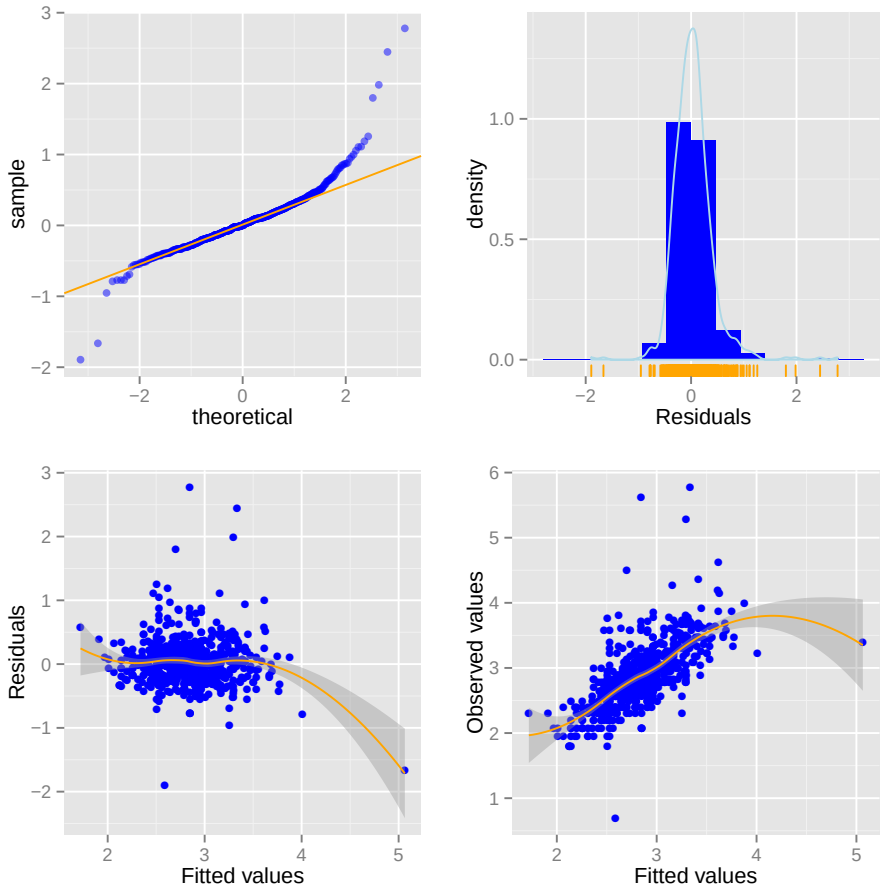
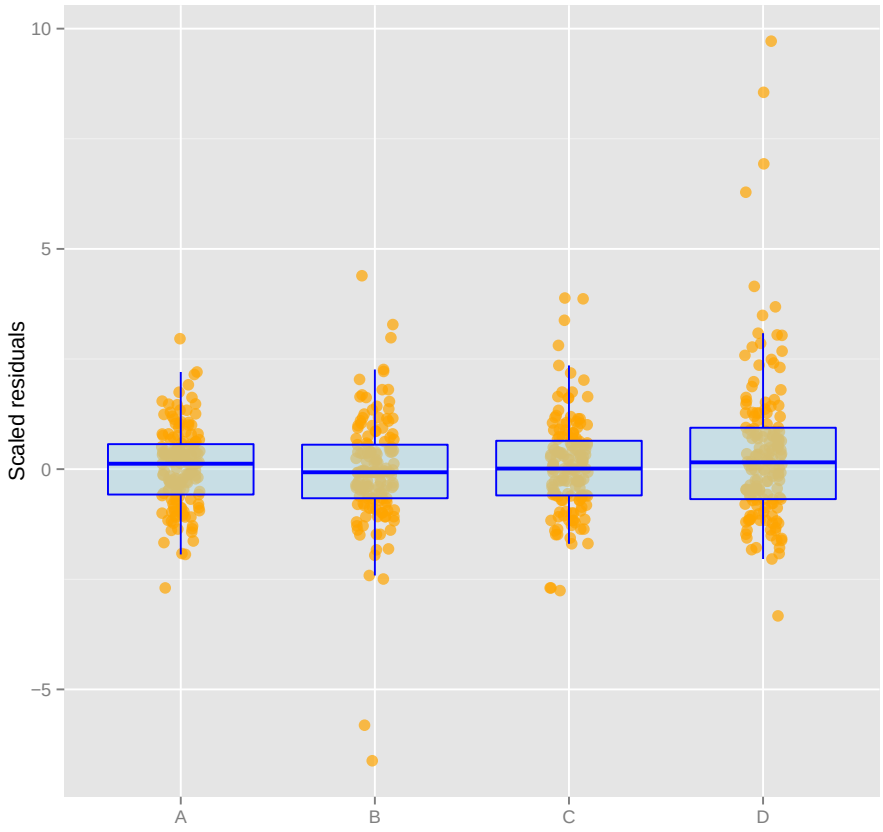


Figure 1: Shiftplots of the liver data.



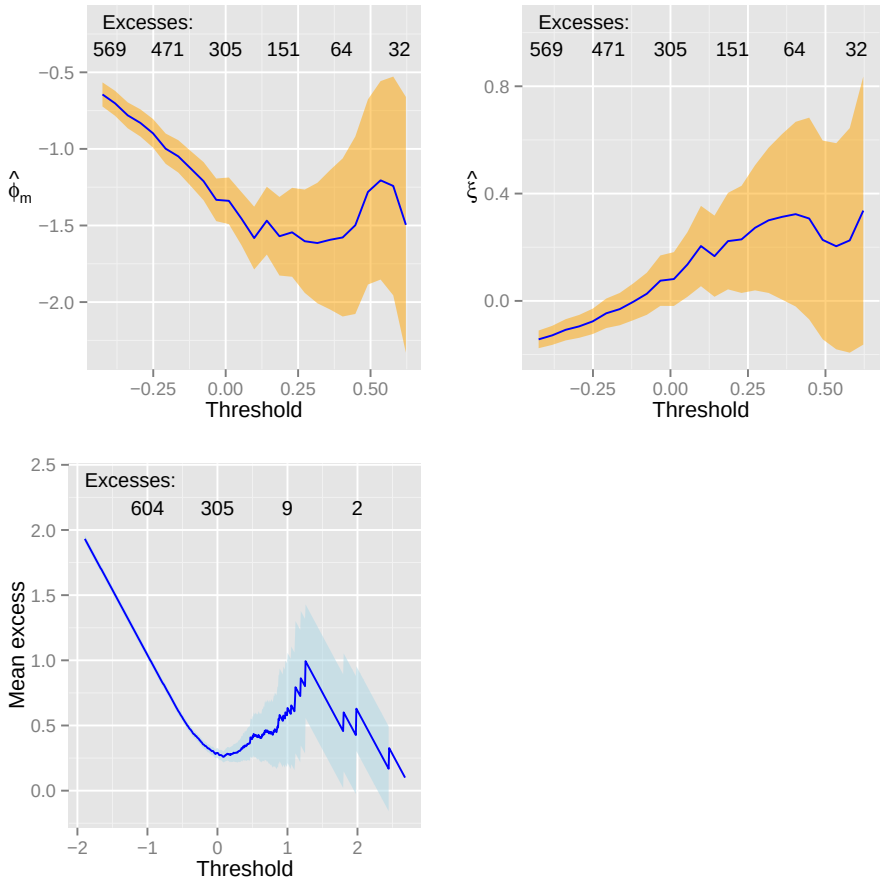

```
boxplot(mm, by="dose")
```



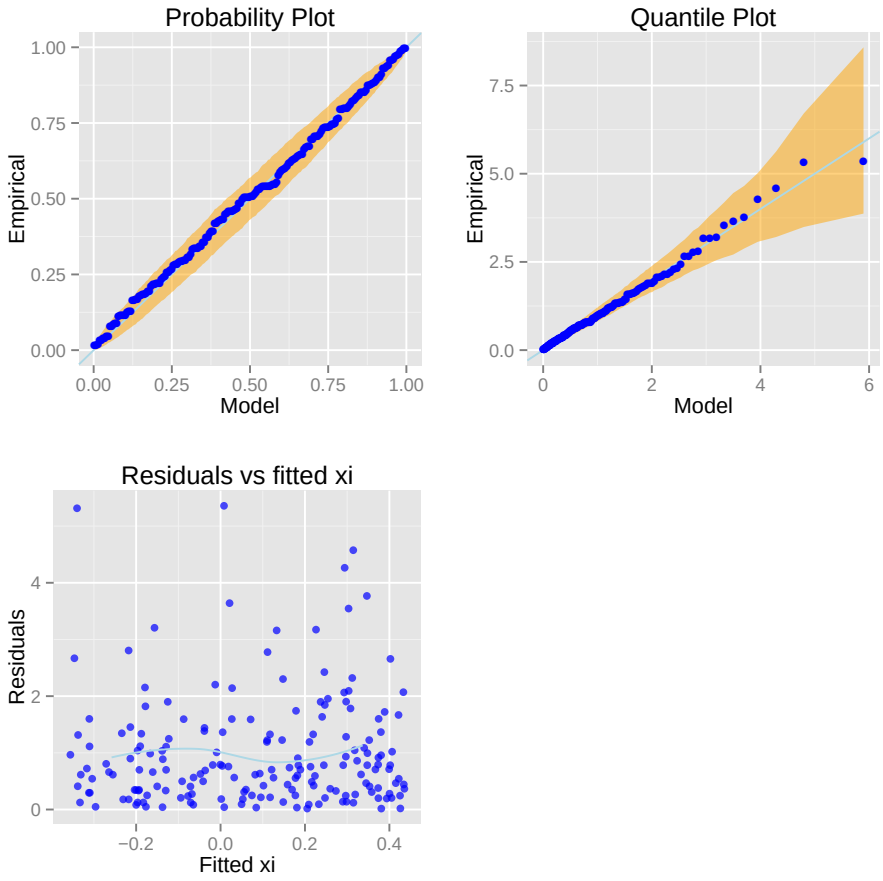
3 Extreme value modelling

We now move on to fit generalized Pareto distributions to the residuals from the robust regression. First we examine some threshold selection plots, then fit the model by maximum likelihood and look at diagnostic plots, then refit by MCMC and look at diagnostic plots of the Markov chains.

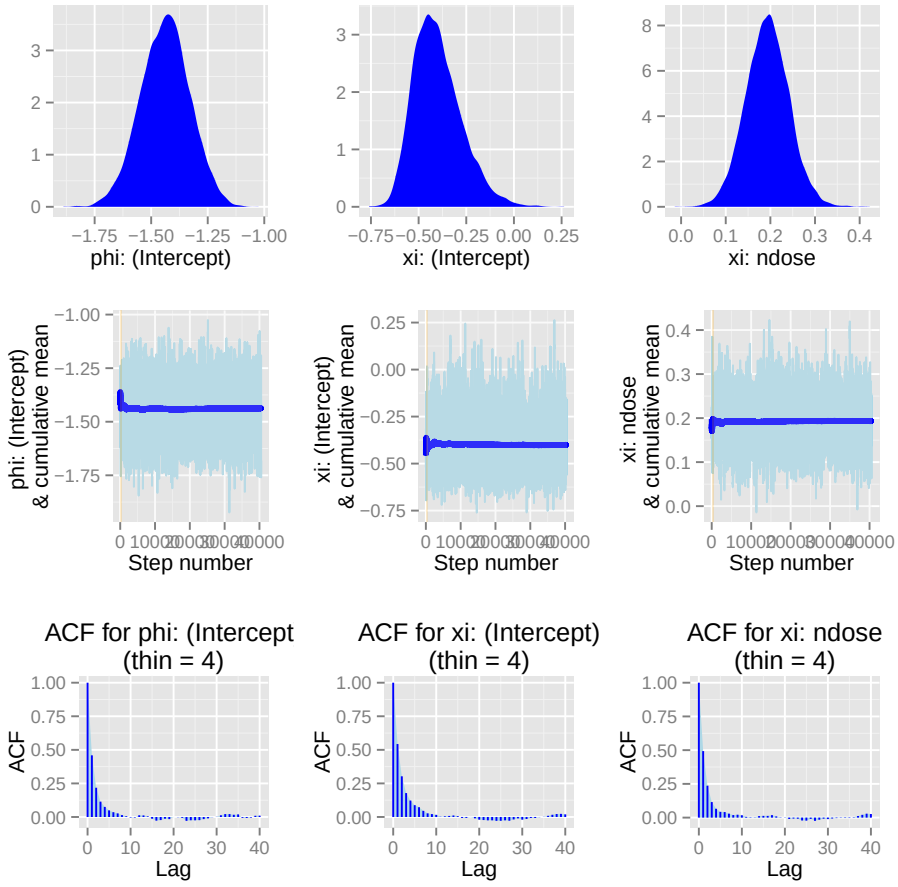
```
# Do threshold selection plots
liver$r <- resid(mm)

ggplot(gpdThresh(liver$r))

# Fit GPD model by penalized ML
mlmod <- evm(r, data=liver, qu=.7, xi=~ndose, penalty="none")
ggplot(mlmod)
```




```
## geom_smooth:  method="auto" and size of largest group is <1000,
so using loess.  Use 'method = x' to change the smoothing method.
```




```
# Refit using MCMC
gmod <- evm(r, data=liver, qu=.7, xi=~ndose,
            method="sim", verbose=FALSE)
ggplot(gmod)
```



4 `margarita`

Before simulating return levels and probabilities of threshold exceedance, we first create a new object that contains the robust regression model, the extreme value model, and other information to allow the simulations to be performed. The function we use is `margarita`.

```
nd <- data.frame(ndose=1:4)
mods <- margarita(mm, gmod, newdata=nd,
                  trans=log, invtrans=exp,
                  baseline="ALT.B")
```

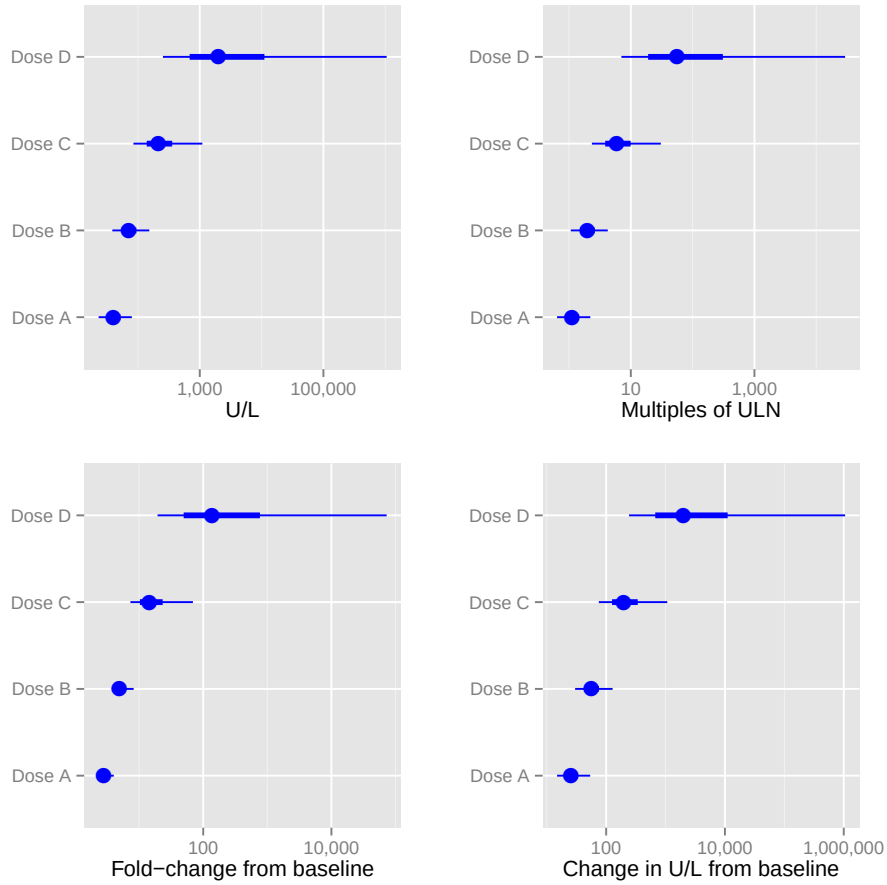
5 Return levels

Return levels for clinical trial data can be simulated using the `simulate` function with `type='rl'` (the default). The simulations for various return levels can be run in a single call by passing a vector argument into the function.

The first step before simulation is to create a new object of class 'margarita' that contains the robust linear model and the evmSim model. The `simulate` functions infer from the number of draws from the posterior distributions contained in the extreme value model how many cases to simulate.

If there are no covariates in the model, `simulate` doesn't need the `newdata` argument. However, if there are covariantes, in order to deal with all kinds of trial designs, the `simulate` function requires `newdata` The rows of `newdata` should be unique and it should *n*ot contain the baseline data. Baseline data are simulated internally by resampling from the data attached to the linear model.

```
rl1000 <- simulate(mods, type="rl", M=1000)

s <- summary(rl1000)
suln <- s/36
sFold <- summary(rl1000, scale="proportional")
sDiff <- summary(rl1000, scale="difference")

# s and the others are lists with an element for each element of M
rownames(s[[1]]) <- paste("Dose", LETTERS[1:4])
rownames(sFold[[1]]) <- rownames(sDiff[[1]]) <-
  rownames(suln[[1]]) <- rownames(s[[1]])
gs <- ggplot(s, xlab="U/L")
gsuln <- ggplot(suln, xlab="Multiples of ULN")
gsFold <- ggplot(sFold, xlab="Fold-change from baseline")
gsDiff <- ggplot(sDiff, xlab="Change in U/L from baseline")
grid.arrange(gs, gsuln, gsFold, gsDiff)
```



Notice that for return levels, because of the way they are calculated, it is straightforward to change the scale after the calculations have been done. It is achieved by providing optional arguments to `summary` or by manipulating the object returned by `summary` directly.

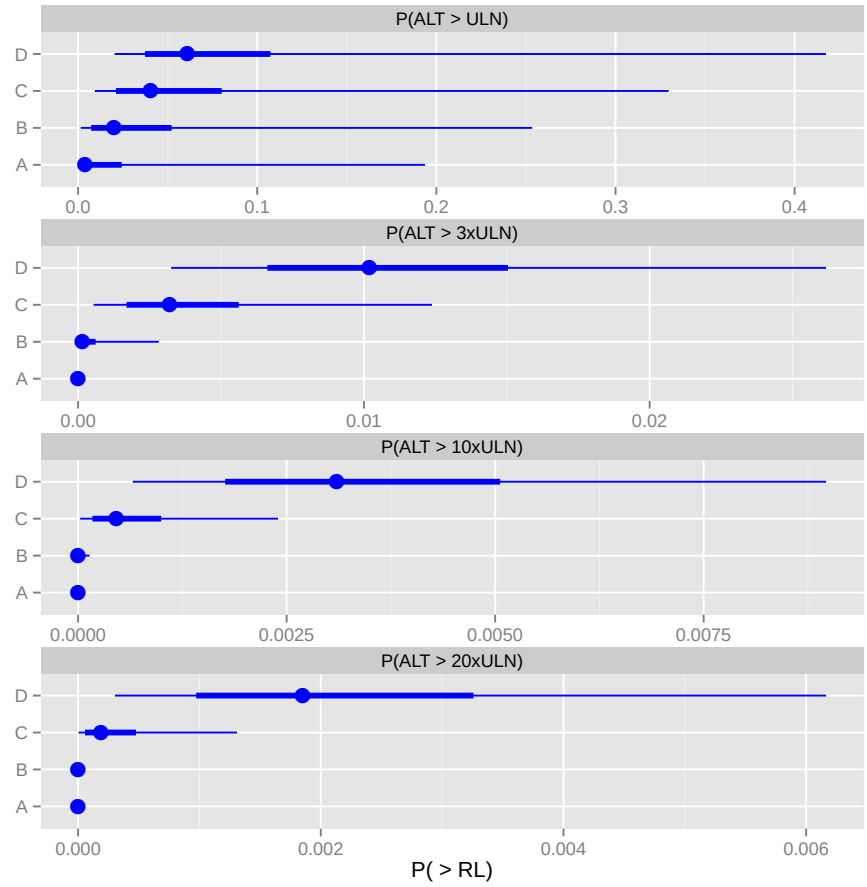
6 Threshold exceedance probabilities

When simulating threshold exceedance probabilities, the scale of the predicted values needs to be specifed in the call to `simulate` rather than `summary`.

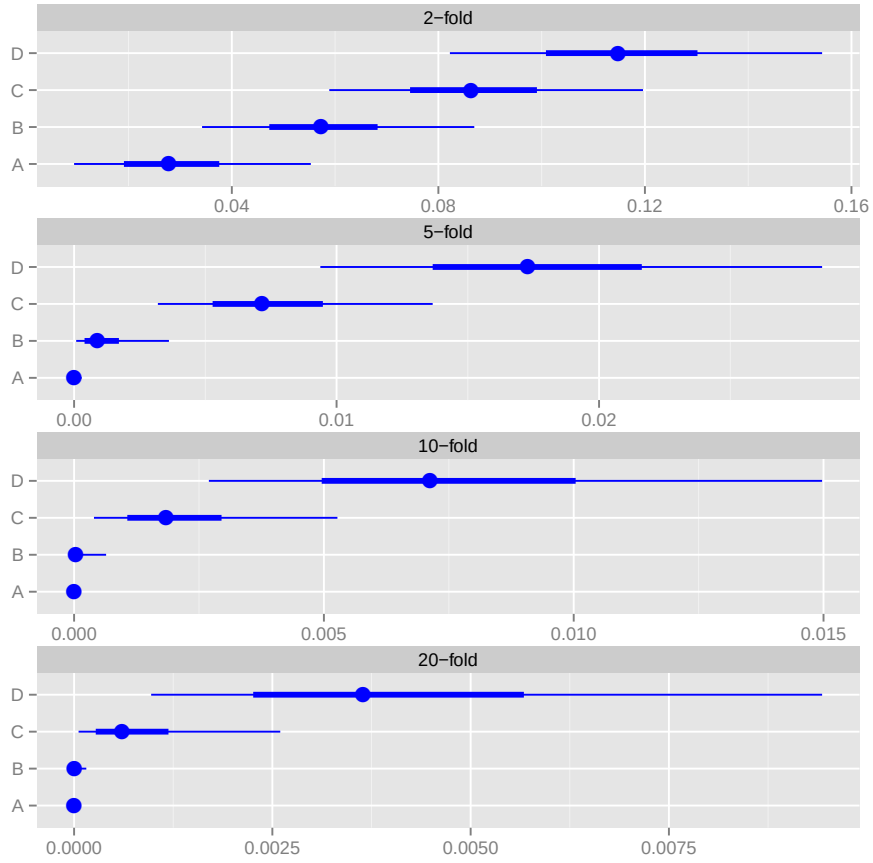
First we find probabilities of threshold exceedance on the scale of the raw data (i.e. multiples of ULN), then in terms of multiples of baseline.

```
pULNs <- simulate(mods, type="prob",
                  M=36*c(1, 3, 10, 20),
                  Mlabels=c("P(ALT > ULN)", "P(ALT > 3xULN)",
                            "P(ALT > 10xULN)", "P(ALT > 20xULN)"))
ps <- summary(pULNs)

# Manually set the treatment group names
names(ps) <- LETTERS[1:4]
```

```
ggplot(ps, ncol=1)
```




```
# Simulate fold-changes from baseline
pBase <- simulate(mods, type="prob", M=c(2, 5, 10, 20),
                  Mlabels=c("2-fold", "5-fold",
                            "10-fold", "20-fold"),
                  scale="proportional")
pbs <- summary(pBase)

# Manually change treatment group names
names(pbs) <- LETTERS[1:4]
ggplot(pbs, ncol=1, xlab="")
```



References

[1] R Core Team (2014). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/.

[2] Harry Southworth and Janet E. Heffernan. texmex: Statistical modelling of extreme values. R package version 2.2, 2013

[3] H. Wickham. ggplot2: elegant graphics for data analysis. Springer New York, 2009

[4] W. N. Venables and B. D. Ripley Modern Applied Statistics with S. Fourth Edition. Springer, New York, 2002. ISBN 0-387-95457-0