Comparison with mgcv

James T. Thorson

```
library(tinyVAST)
library(pdp)  # approx = TRUE gives effects for average of other covariates
#> Warning: package 'pdp' was built under R version 4.3.1
library(lattice)
library(visreg)
#> Warning: package 'visreg' was built under R version 4.3.1
set.seed(101)
```

`tinyVAST` is an R package for fitting vector autoregressive spatio-temporal (VAST) models using a minimal and user-friendly interface. We here show how it can replicate analysis using splines specified via `mgcv`

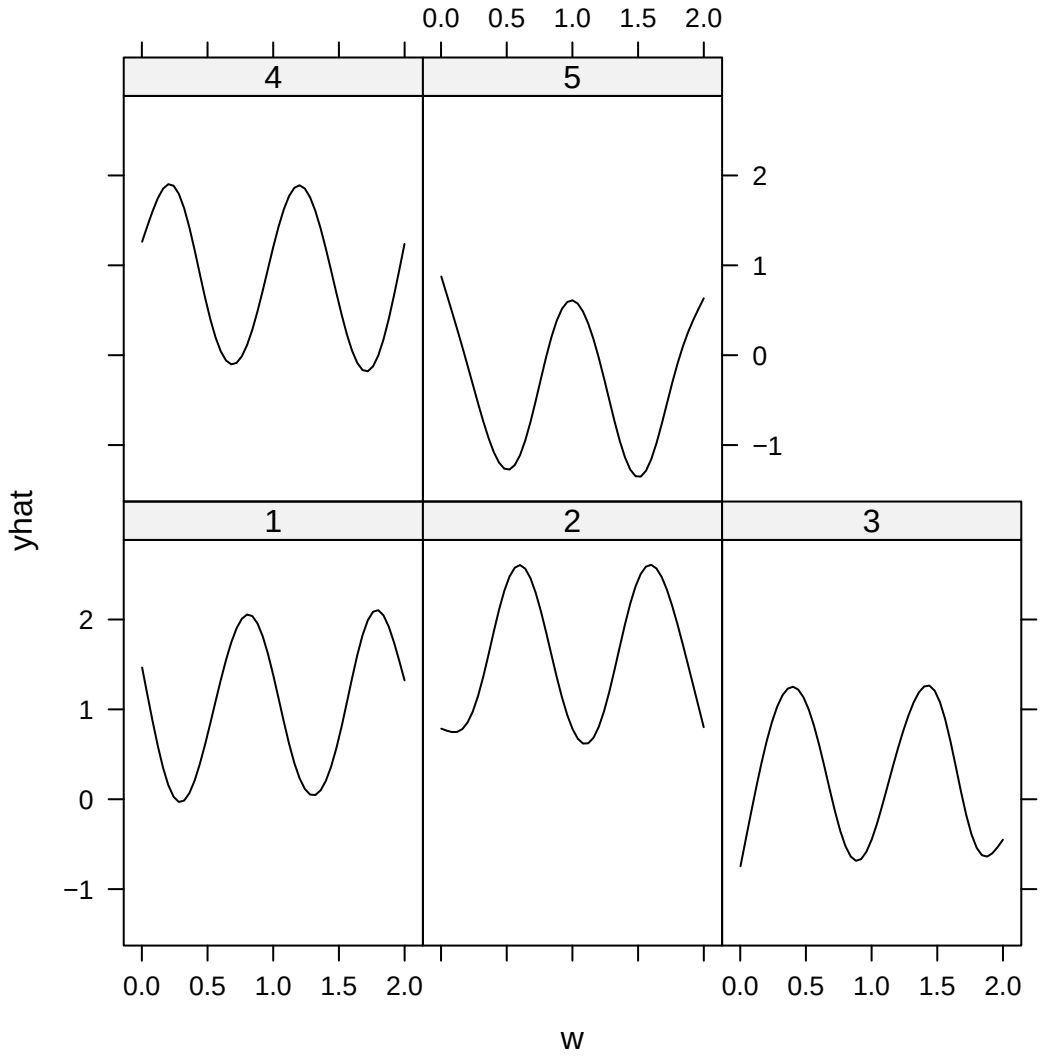
```
# Simulate
n_obs = 1000
x = rnorm(n_obs)
group = sample( x=1:5, size=n_obs, replace=TRUE )
w = runif(n_obs, min=0, max=2)
z = 1 + x^2 + cos((w+group/5)*2*pi) + rnorm(5)[group]
y = z + rnorm(n_obs, sd=0.2)
Data = data.frame( x=x, y=y, w=w, z=z, group=factor(group) )

# fit model
Formula = y ~ 1 + s(group, bs="re") + poly(x, 2, raw=TRUE) + s(w, by=group)
myfit = fit( data = Data,
           formula = Formula,
           quiet = TRUE )
```

`tinyVAST` then has a standard `predict` function:

```
predict(myfit, newdata=data.frame(x=0, y=1, w=0.4, group=2) )
#> [1] 1.852686
```

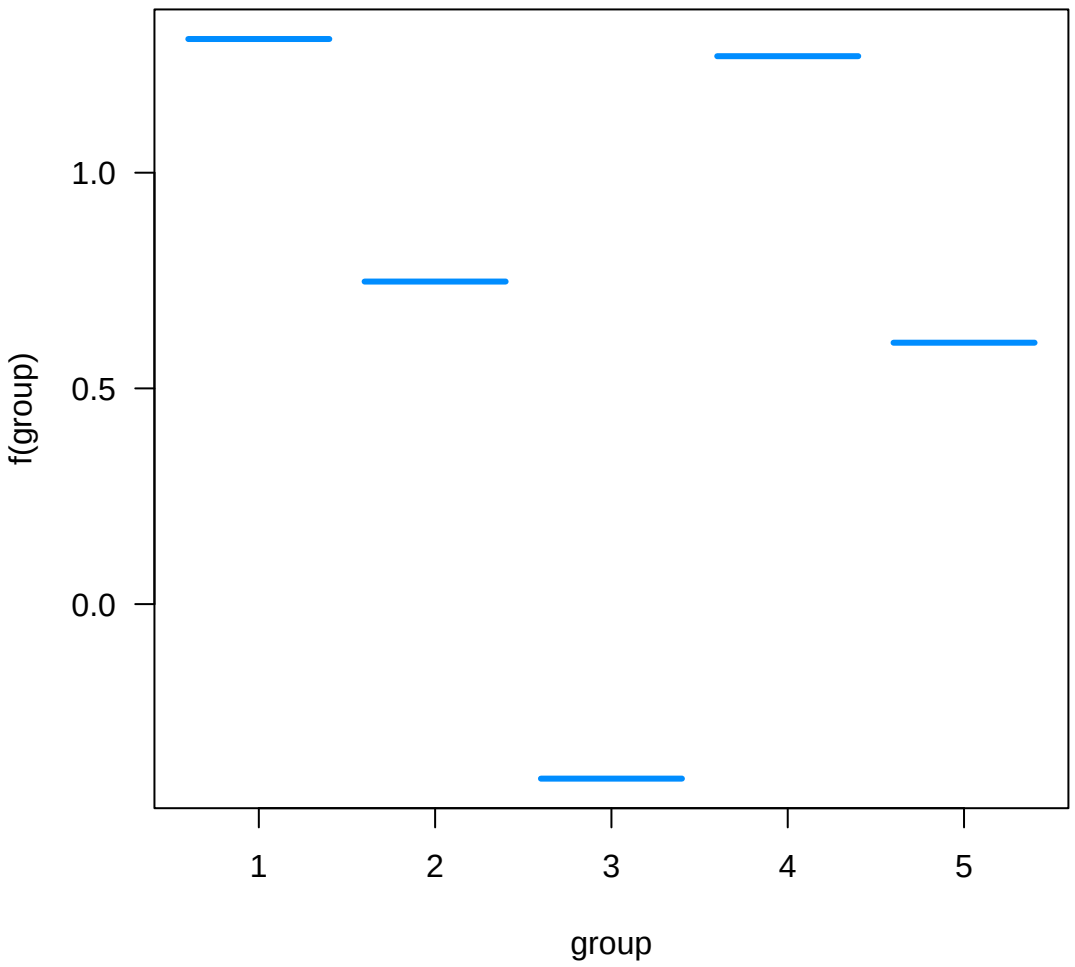
and this is used to compute partial-dependence plots using package `pdp`

```
# compute partial dependence plot
Partial = partial( object = myfit,
                   pred.var = c("w","group"),
                   pred.fun = \(object,newdata) predict(object,newdata),
                   train = Data,
                   approx = TRUE )

# Lattice plots as default option
plotPartial( Partial )
```

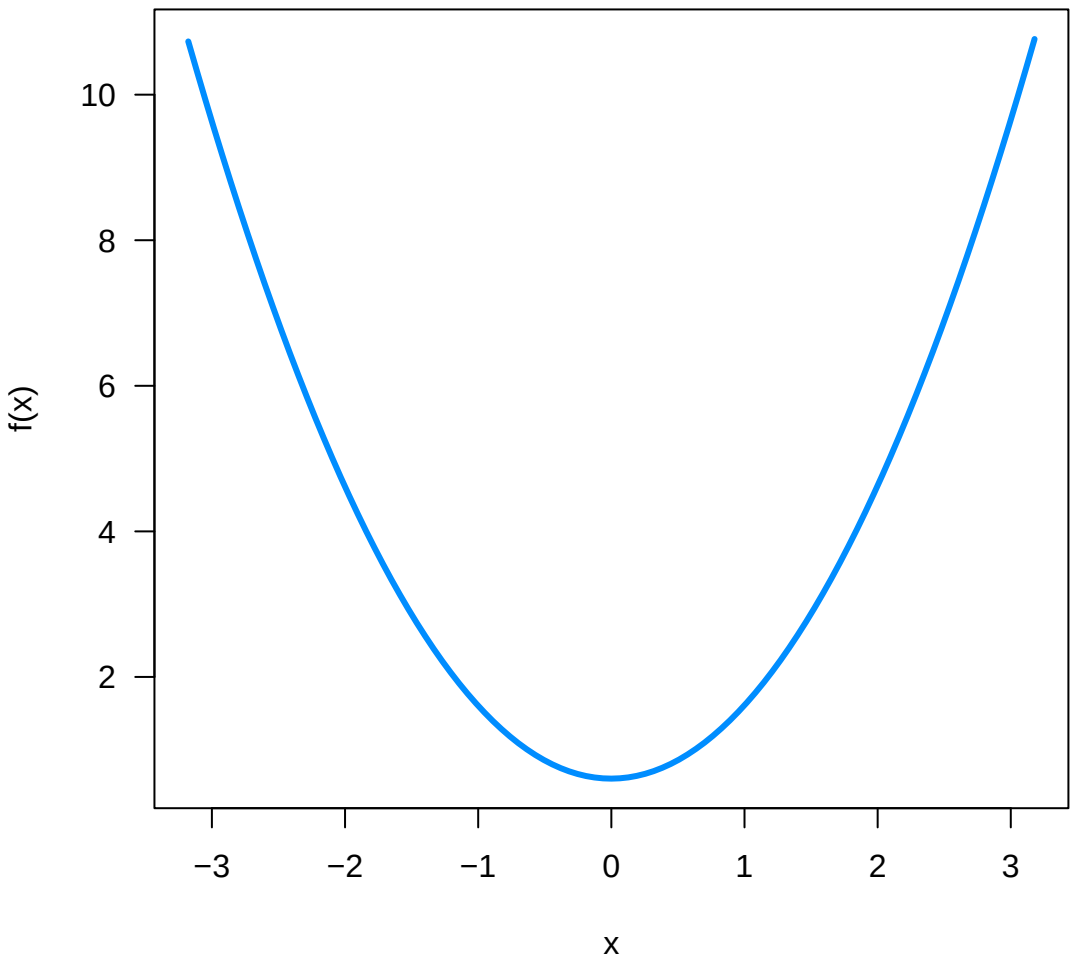


Alternatively, we can use `visreg` to visualize output:

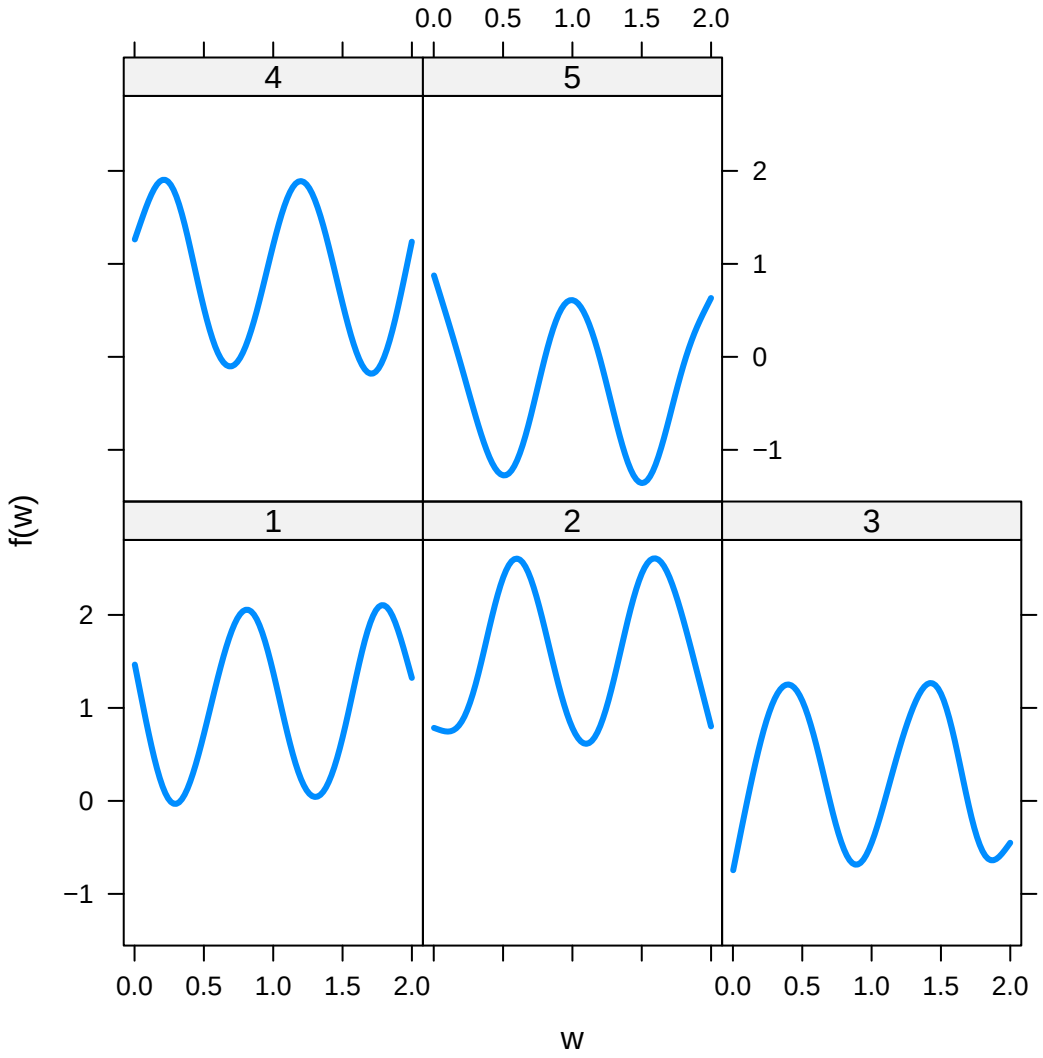
```
visreg(myfit, xvar="group")
#> Warning in plot.visreg(v, ...): The generic function residuals() is not set up for this type of mode
#> you will need to define your own residuals.tinyVAST() function.
```




```
visreg(myfit, xvar="x")
#> Warning in plot.visreg(v, ...): The generic function residuals() is not set up for this type of mode
#> you will need to define your own residuals.tinyVAST() function.
```



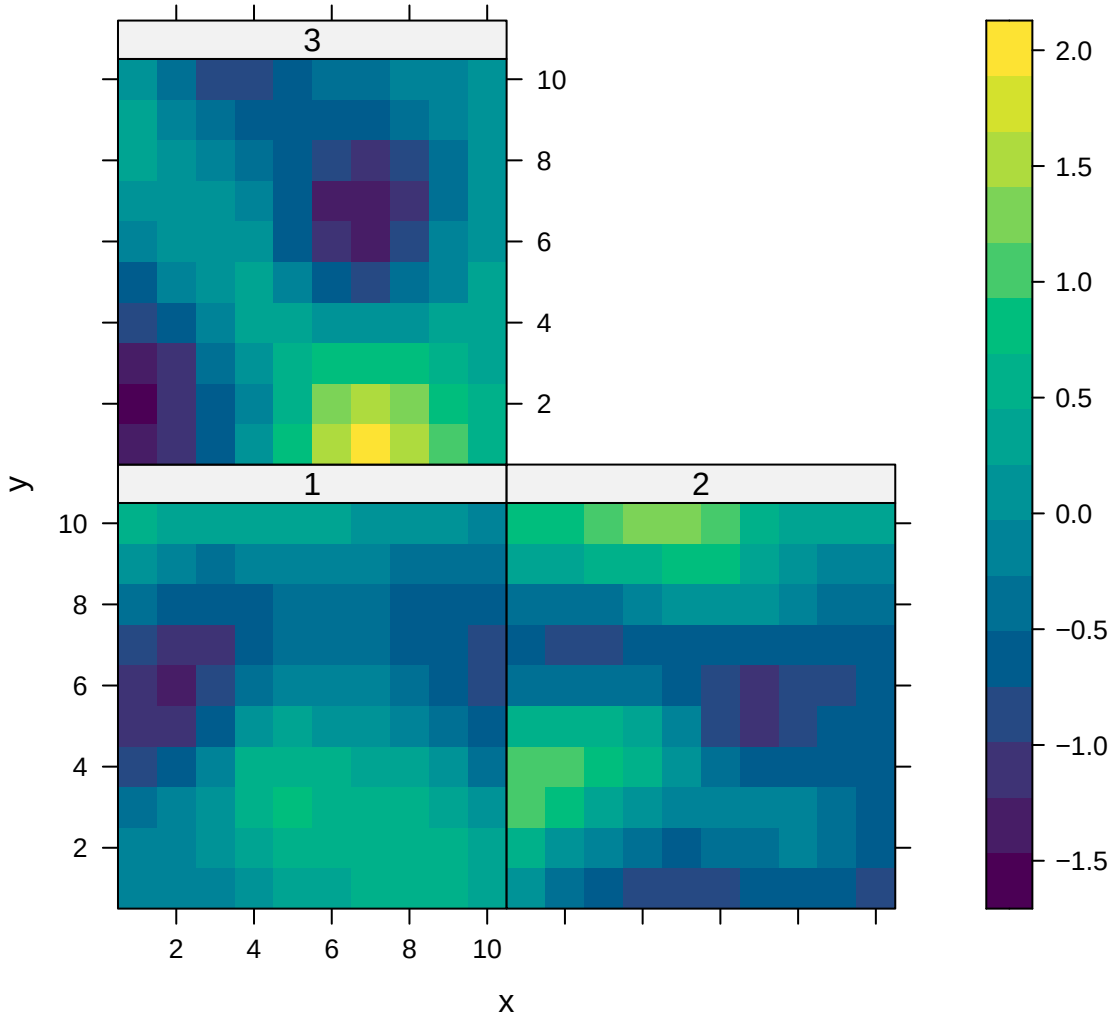

```
visreg(myfit, xvar="w", by="group")
#> Warning in plot.visreg(v, ...): The generic function residuals() is not set up for this type of mode
#> you will need to define your own residuals.tinyVAST() function.
```



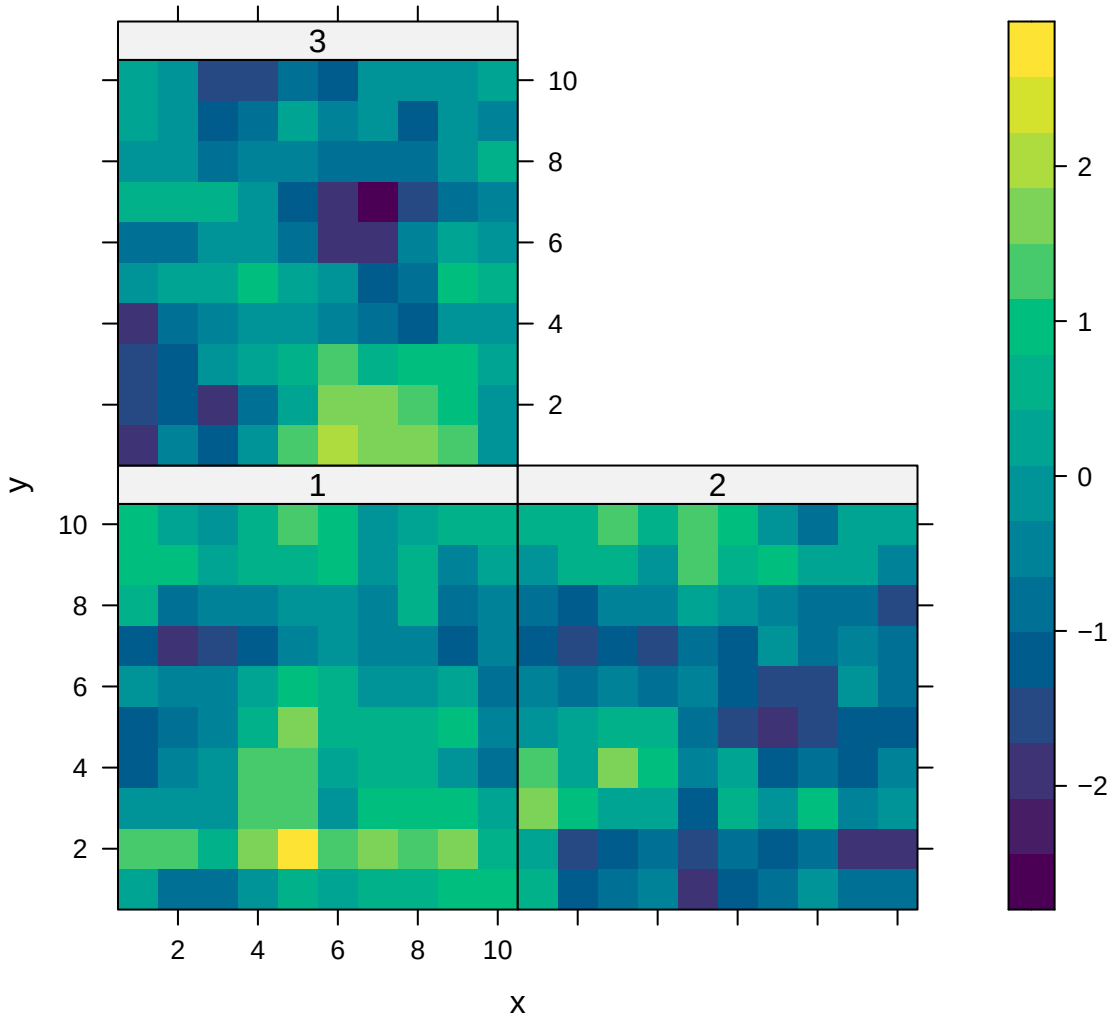
Similarly, we can fit a grouped 2D spline

```
# Simulate
R = exp(-0.4 * abs(outer(1:10, 1:10, FUN="-")) )
z = mvtnorm::rmvnorm(3, sigma=kronecker(R,R) )
Data = data.frame( expand.grid(x=1:10, y=1:10, group=1:3), z=as.vector(t(z)))
Data$n = Data$z + rnorm(nrow(Data), sd=0.1)
Data$group = factor(Data$group)

# fit model
Formula = n ~ s(x, y, by=group)
myfit = fit( data = Data,
             formula = Formula,
             quiet = TRUE )

# compute partial dependence plot
mypartial = partial( object = myfit,
```

```
                    pred.var = c("x","y","group"),
                    pred.fun = \(object,newdata) predict(object,newdata),
                    train = Data,
                    approx = TRUE )

# Lattice plots as default option
plotPartial( mypartial )
```

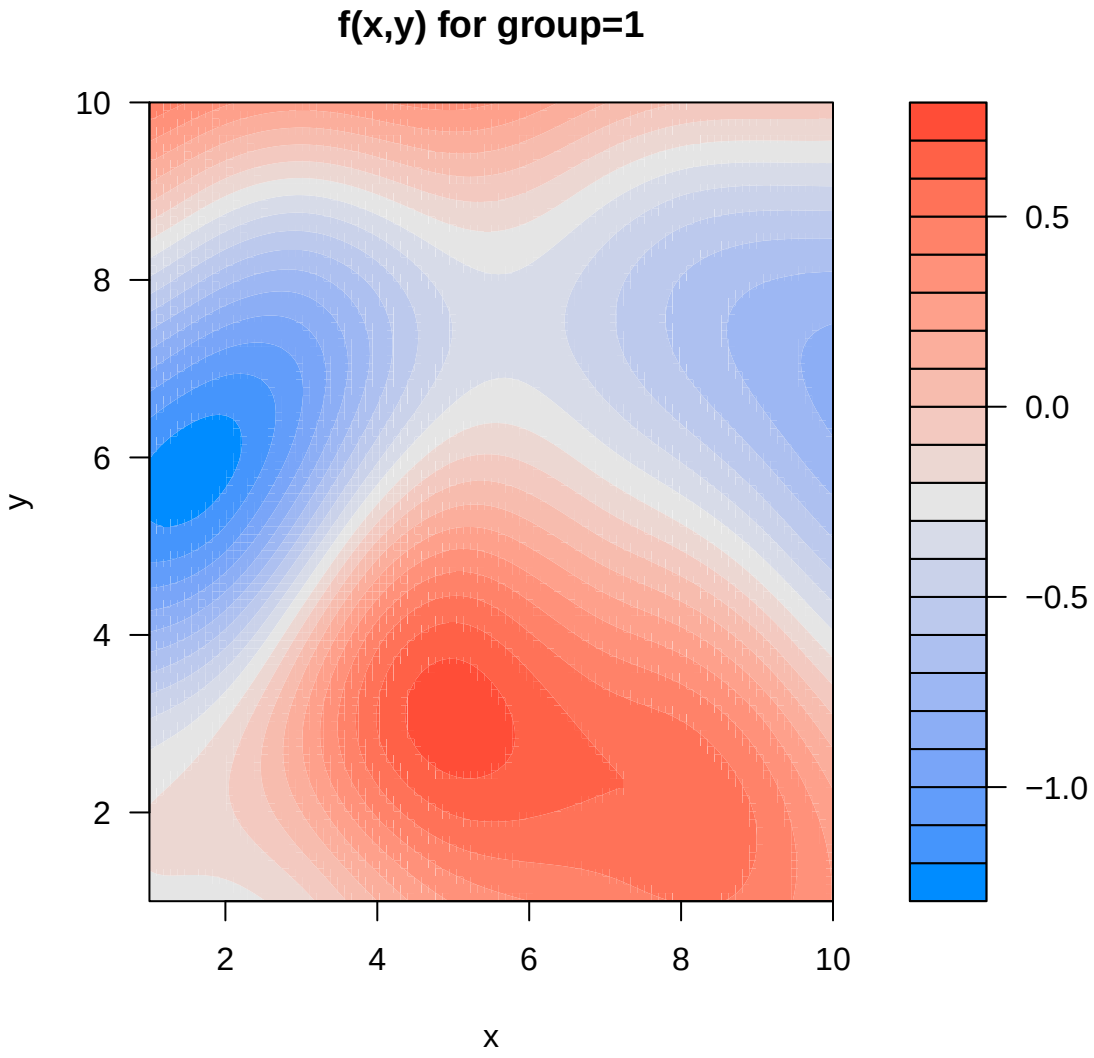



```
# Lattice plot of true values
mypartial$yhat = Data$z
plotPartial( mypartial )
```



We can again use `visreg` to visualize response surfaces, although it doesn't seem possible to extract a grouped spatial term, so we here show only a single term:

```
out = visreg2d( myfit, "x", "y", cond=list("group"=1), plot=FALSE )
plot( out, main="f(x,y) for group=1")
```


f(x,y) for group=1
10
8
6
4
2
y
2
4
6
8
10
x
0.5
0.0
−0.5
−1.0