vivid-R-journal

Alan n. Inglis

2022-10-19

Packages

```
# devtools::install_github("AlanInglis/vivid")
# devtools::install_github("cbhurley/condvis2")
if(!require(network)){
    install.packages("network")
}
if(!require(sp)){
    install.packages("sp")
}

library("vivid") # for visualisations
library("randomForest") # to create model
library('xgboost') # to create model
library("ggplot2") # for visualisations
library('MASS') # for data
```

Get data and fit models

```
set.seed(1701) # for reproducibility

# get data
data("Boston")

# Model fits ----------------------------------------------------------------

# random forest:
rf <- randomForest(medv ~ ., data = Boston, importance = TRUE)

# gbm:
gbst <- xgboost(
  data = as.matrix(Boston[,c(1:13)]),
  label =  as.matrix(Boston[,14]),
  nrounds = 100
)
```

Create vivid matrix

```
# vivi function ------------------------------------------------------------

# vivi for rf
set.seed(1701)

viviRf <- vivi(fit = rf,
               data = Boston,
               response = "medv",
               reorder = FALSE,
               normalized = FALSE,
               importanceType = 'agnostic',
               gridSize = 50,
               nmax = 500,
               class = 1,
               predictFun = NULL)



# predict function for gbm
pFun <- function(fit, data, prob = FALSE) predict(fit, as.matrix(data[,1:13]))

# vivi for GBM
set.seed(1701)
viviGBst <- vivi(fit = gbst,
                 data = Boston,
                 response = "medv",
                 reorder = FALSE,
                 normalized = FALSE,
                 predictFun = pFun)
```

vip2vivid function

```
# vip2vivid ----------------------------------------------------------------

library("vip")
# get model specific VImps using vip package
vipVImp <- vi(rf, method = 'model')
# get VInts using vip package
vipVInt <- vint(rf, feature_names = names(Boston[-14]))

# turn into vivi-matrix
vipViviMat <- vip2vivid(importance = vipVImp, interaction = vipVInt)
```

vivid matrix set up

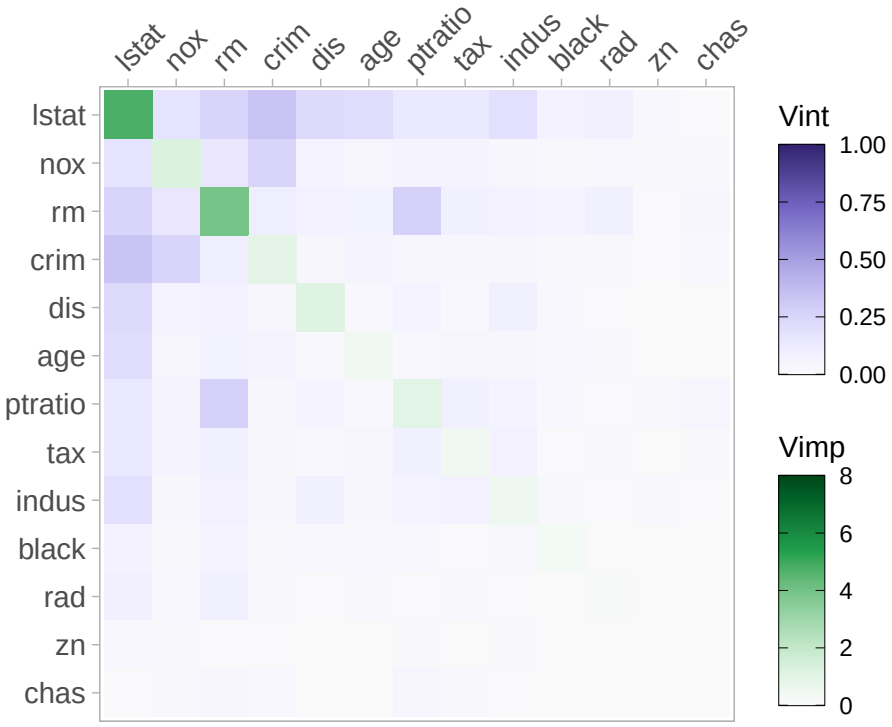
```
# vivi matrix --------------------------------------------------------------

# average over matrices and seriate to get common ordering
```

```
viviAvg <- (viviRf + viviGBst) / 2
viviAvgReorder <- vividReorder(viviAvg)

# reorder vivi-matrices
ord <- colnames(viviAvgReorder)
viviRf <- viviRf[ord,ord]
viviGBst <- viviGBst[ord,ord]
```

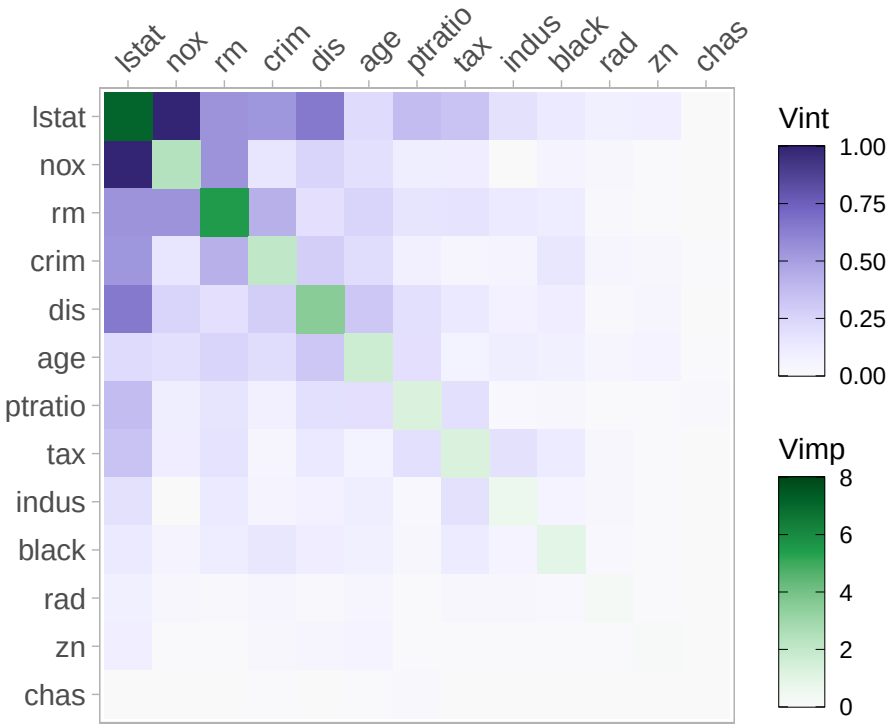
Heatmap plot

Figure 2

```
# heatmap for random forest
viviHeatmap(viviRf, angle = 45, intLims = c(0,1), impLims = c(0,8))
```



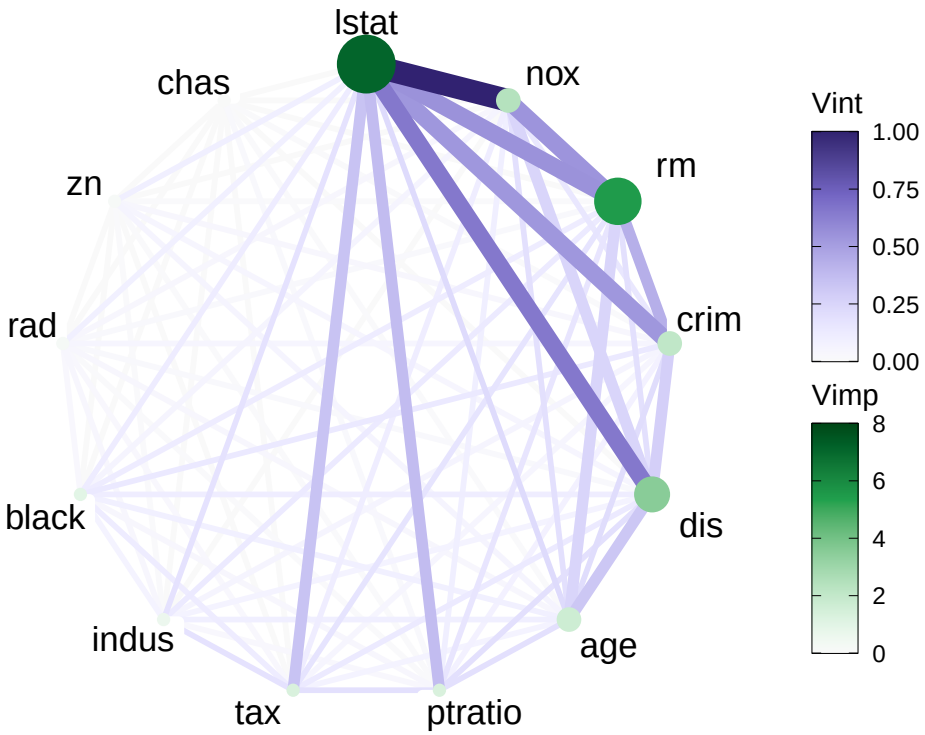

```
# heatmap for GBM
viviHeatmap(viviGBst, angle = 45, intLims = c(0,1), impLims = c(0,8))
```



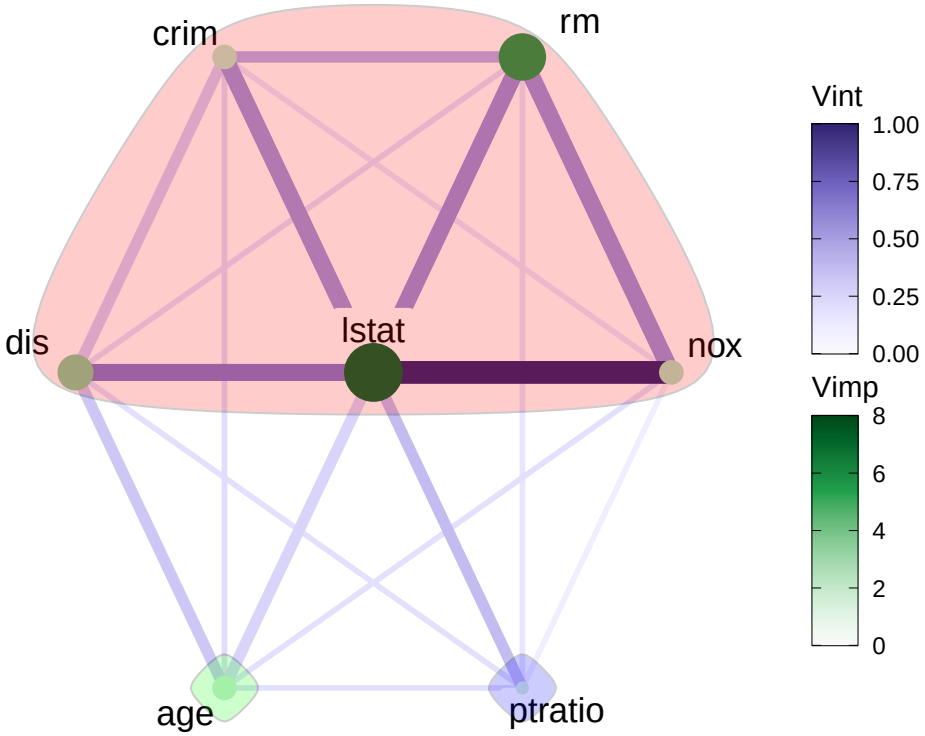
Network plot

Figure 3

```
# default network plot for GBM shown in Figure 2 (a)
viviNetwork(viviGBst)
```



```
# clustered and filtered network for GBM shown in Figure 2 (b)
intVals <- viviGBst
diag(intVals) <- NA

# select VIVI values above median value
sv <- which(diag(viviGBst) > median(diag(viviGBst)) |
              apply(intVals, 1, max, na.rm=TRUE) > median(apply(intVals, 1, max, na.rm=TRUE)))

# perform hierarchical clustering
h <- hclust(-as.dist(viviGBst[sv,sv]), method="single")

# plot graph clustering by high VIVI
viviNetwork(viviGBst[sv,sv],
            intLims = c(0,1),
            impLims = c(0,8),
            cluster = cutree(h, k = 3), # specify number of groups
            layout = igraph::layout_as_star)
```



vivid dataframe

```
head(as.data.frame(viviRf), 4)
```

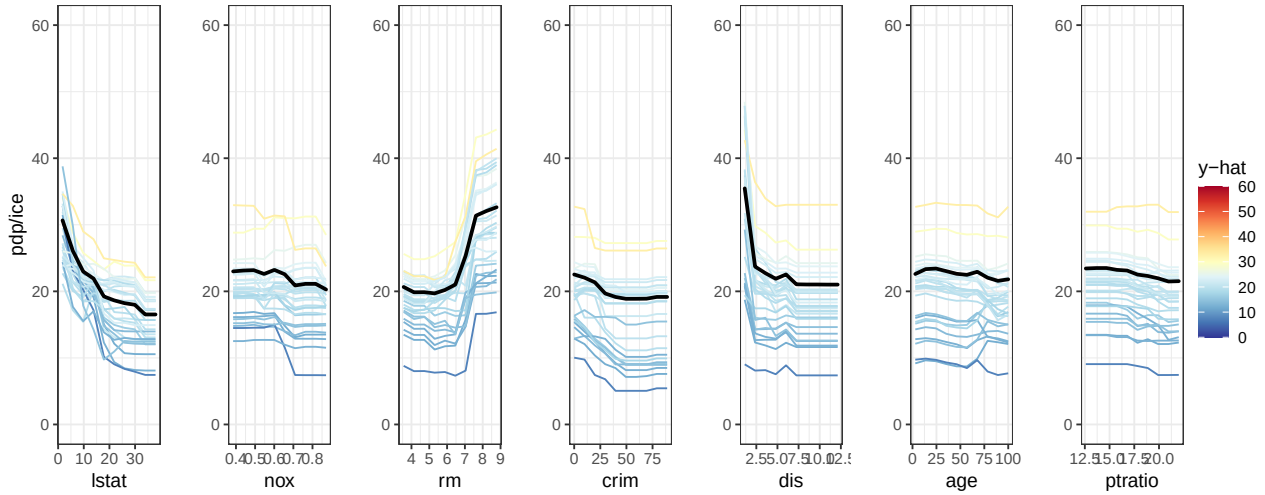
```
##           lstat       nox        rm      crim        dis        age    ptratio
## lstat 4.7720394 0.1789794 0.2562817 0.3371364 0.22668020 0.21651732 0.13766704
## nox   0.1789794 1.3070659 0.1558134 0.2500224 0.07291952 0.04626609 0.06121573
## rm    0.2562817 0.1558134 3.9296588 0.1070185 0.08261184 0.07595697 0.27238691
## crim  0.3371364 0.2500224 0.1070185 0.9506134 0.03999383 0.06023689 0.04538070
##              tax      indus      black        rad          zn        chas
## lstat 0.13711979 0.18967729 0.07459025 0.08754038 0.026026784 0.007842981
```

```
## nox   0.05786027 0.04290793 0.01929860 0.03231187 0.023965214 0.019481521
## rm    0.08928171 0.08253460 0.06791441 0.09102666 0.012697135 0.035426704
## crim  0.04378221 0.04490474 0.01675859 0.01530794 0.009210787 0.028336857
```

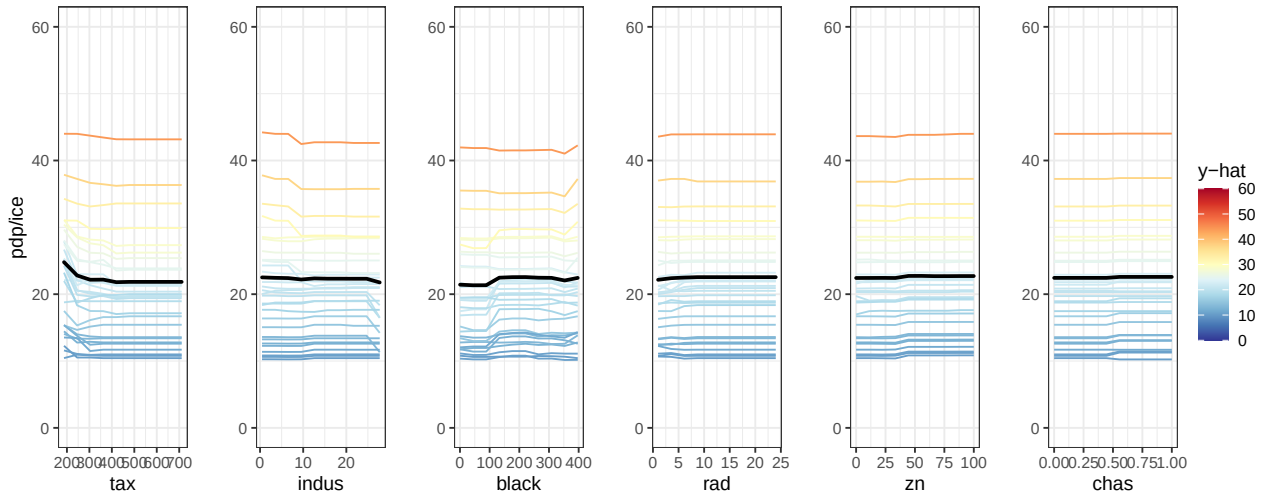
PDP

Figure 4

```
# create PDPs for GBM
pdpVars(data = Boston,
        fit = gbst,
        response = 'medv',
        vars = colnames(viviGBst)[1:7],
        predictFun = pFun)
```



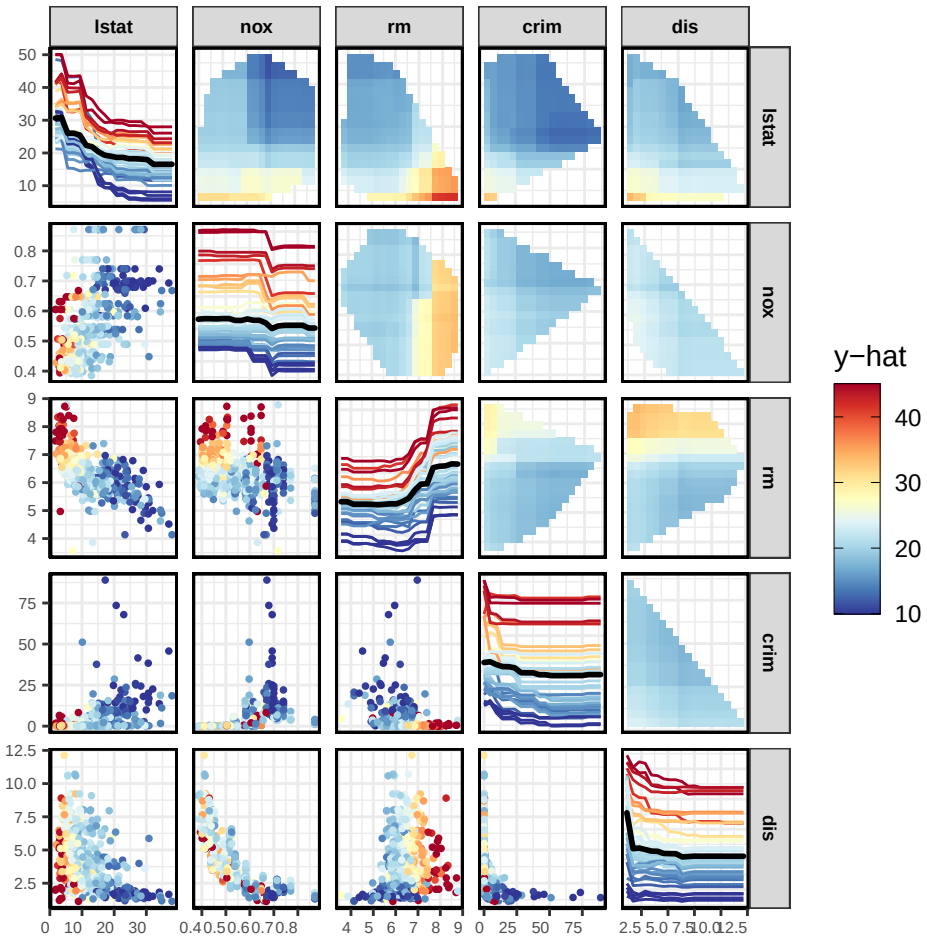

```
pdpVars(data = Boston,
        fit = gbst,
        response = 'medv',
        vars = colnames(viviGBst)[8:13],
        predictFun = pFun)
```



GPDP

Figure 5

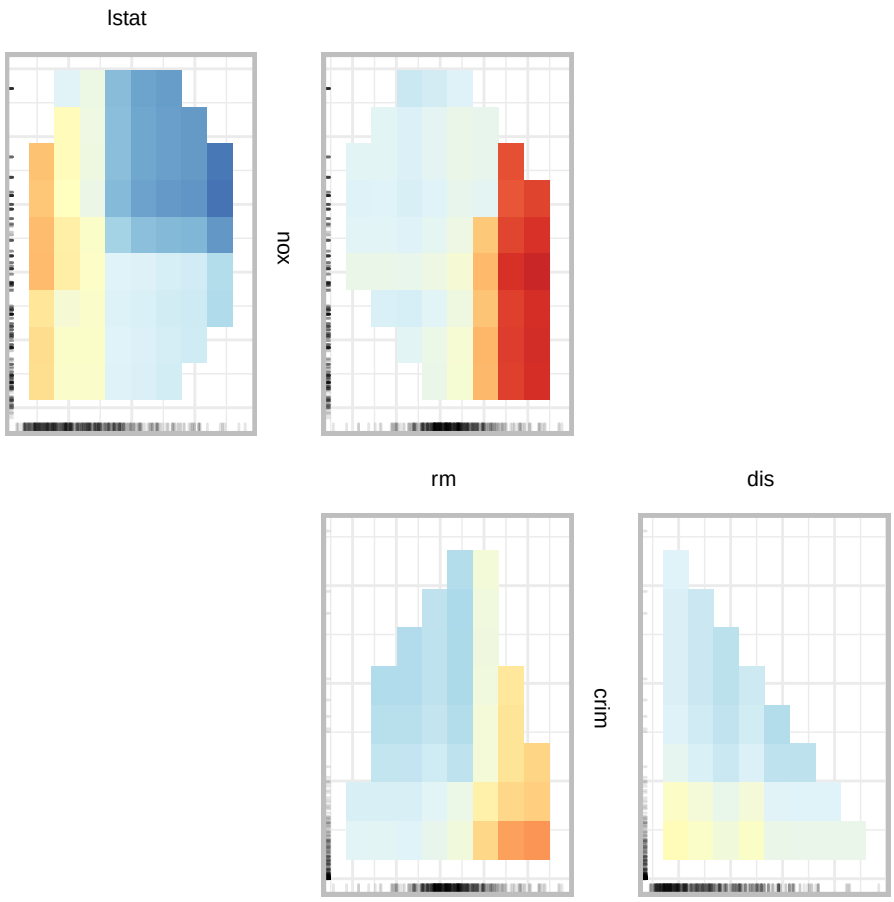
```
# filter matrix:
filteredVars <- colnames(viviGBst)[1:5]

# select rows to plot associated ICE curves:
rmHigh <- sample(which(Boston$rm > mean(Boston$rm)), 25)
lstatLow <- sample(which(Boston$lstat < mean(Boston$lstat)), 25)

# create GPDP for gbm:
set.seed(1701)
pdpPairs(data = Boston,
         fit = gbst,
         response = "medv",
         gridSize = 20,
         nIce = c(rmHigh, lstatLow),
         var = filteredVars,
         convexHull = TRUE,
         fitlims = "pdp",
         predictFun = pFun)
```



ZPDP

Figure 6

```
# create ZPDP for gbm:
pdpZen(data = Boston,
       fit = gbst,
       response = "medv",
       convexHull = TRUE,
       zpath = colnames(viviGBst)[1:5],
       predictFun = pFun)
```

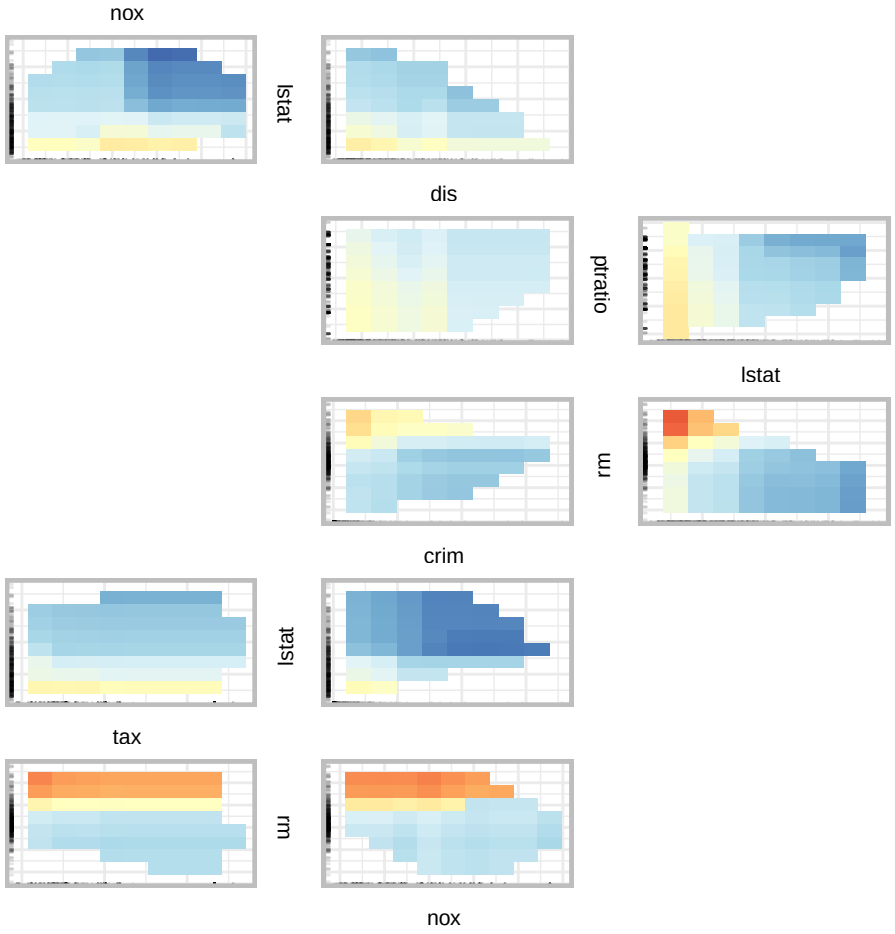


ZPDP

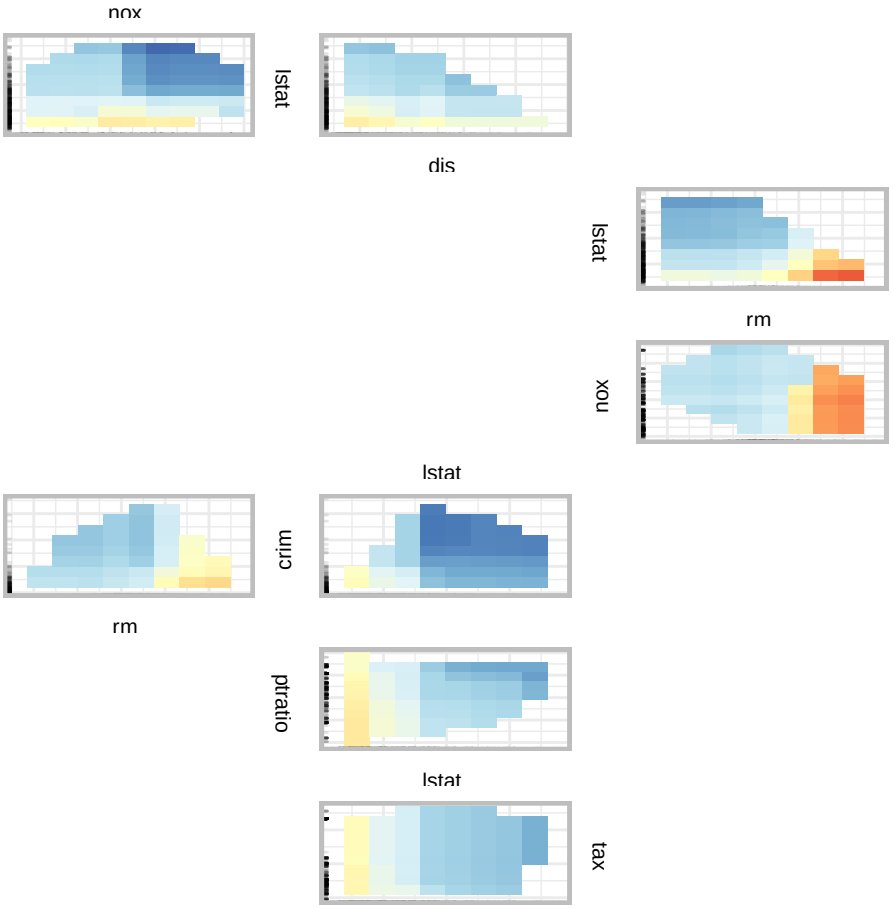
Figure 7

```
# find the 90% quantile of the interactions
qVIntBst <- quantile(intVals, 0.9, na.rm=TRUE)

# set zpaths with different parameters
zpGw  <- zPath(viv = viviGBst, cutoff = qVIntBst, method = 'greedy.weighted')
zpSw  <- zPath(viv = viviGBst, cutoff = qVIntBst, connect = FALSE, method = 'strictly.weighted')

# plots
pdpZen(data = Boston,
```

```
       fit = gbst,
       response = "medv",
       zpath = zpGw,
       convexHull = TRUE,
       predictFun = pFun)
```




```
pdpZen(data = Boston,
       fit = gbst,
       response = "medv",
       zpath = zpSw,
       convexHull = TRUE,
       predictFun = pFun)
```


nox
lstat
dis
lstat
rm
nox
lstat
crim
rm
ptratio
lstat
tax