Cervical data

A. Inglis and C. Hurley

2021-10-25

Packages

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

library(mlr)
library(vivid)
library(dplyr)
library(gbm)
RNGkind(kind="Mersenne-Twister", normal.kind="Inversion", sample.kind="Rejection" ) # defaults
```

Read in and setup data.

```
cervical <- read.csv("https://raw.githubusercontent.com/AlanInglis/vivid/master/paperCodeData/cervicalCa




# Remove cancer tests, similar and constant variables:
# take logs of skewed variables
# Turn dummy variables into factors:
cervical <- dplyr::select(cervical, -Citology, -Schiller, -Hinselmann,
                          -Dx.CIN, -Dx, -Horm_Cont,
                          -Smokes, -IUD, -STDs, -STDs_No_diag, -STDs_AIDS,
                          -STDs.Hep_B, -STDs_cerv_condy, -STDs_Time_first_diag,
                          -STDs_Time_last_diag, -STDs_pel_inf,
                          -STDs_gen_h, -STDs_m_c,
                          -STDs.HPV, -STDs_vag_condy, -STDs_vp_condy) %>%
   mutate(across(Age:IUD_yrs, ~ log(.x+ 1))) %>%
   mutate(Biopsy = factor(Biopsy, levels=0:1,labels=c('Healthy', 'Cancer') )) %>%
   mutate(across(.cols=c("STDs_condy", "STDs_syph", "STDs_HIV",
                         "Dx.HPV", "Dx.Cancer"),   factor))
# set up training and testing, making sure Biopsy is proportionally sampled
set.seed(1701)
```

```
split <- rsample::initial_split(cervical, prop=.7, strata = Biopsy)
cTrain <- rsample::training(split)
cTest <- rsample::testing(split)
```

Model fitting

```
cTask <- makeClassifTask(data = cTrain, target = "Biopsy")
cLrn <- makeLearner("classif.gbm",
                    predict.type = "prob",
                    par.vals = list(
                       interaction.depth = 2,
                       n.trees = 100,
                       shrinkage = 0.15,
                       n.minobsinnode=5
                    ))
set.seed(1701)
cfit <- train(cLrn, cTask)
```

```
## Distribution not specified, assuming bernoulli ...
```

```
# # Test predictions
pred1 <- predict(cfit, newdata = cTest)
# # Evaluate performance accuracy, area under curve and mean misclassification error
performance(pred1, measures = list(acc, auc))
```

```
##       acc       auc
## 0.9341085 0.7349280
```

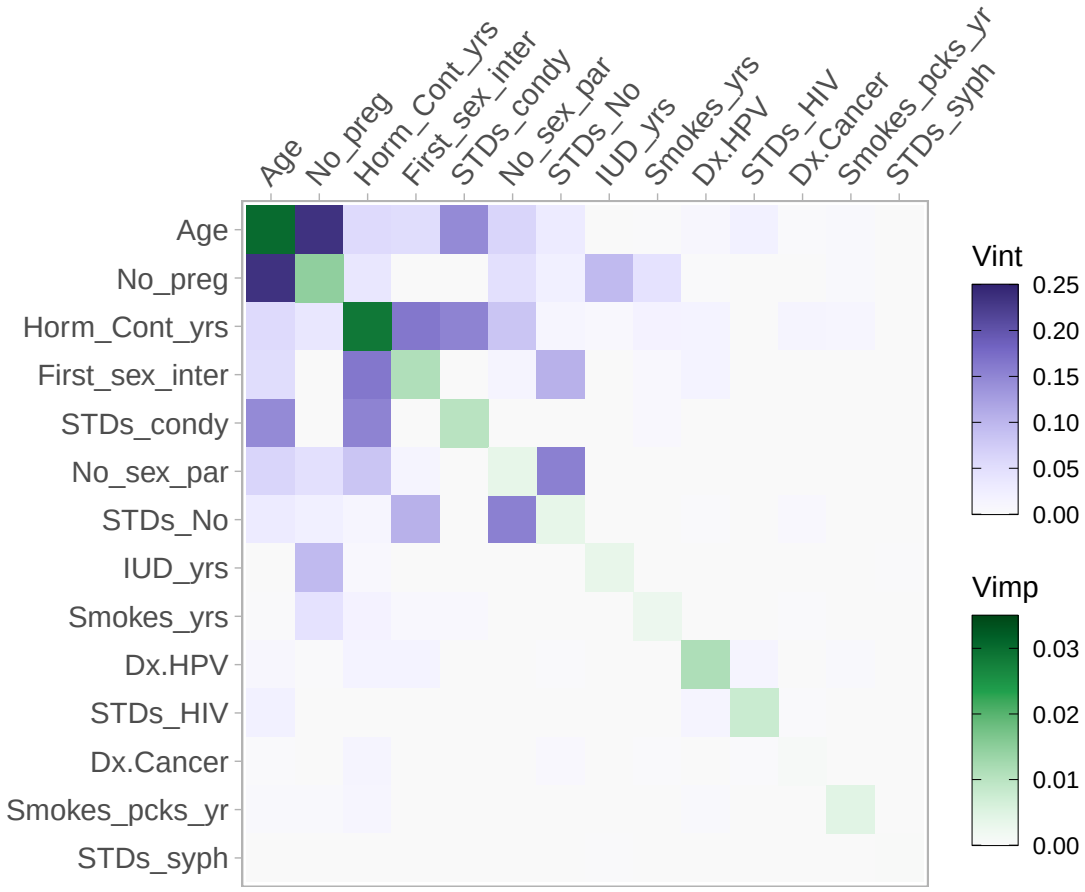
Create vivid matrix

```
set.seed(1701)
viv <- vivi(cTrain, cfit, response = "Biopsy", class = "Cancer",
                gridSize = 30, importanceType = "agnostic")
```

Heatmap

Figure 6:

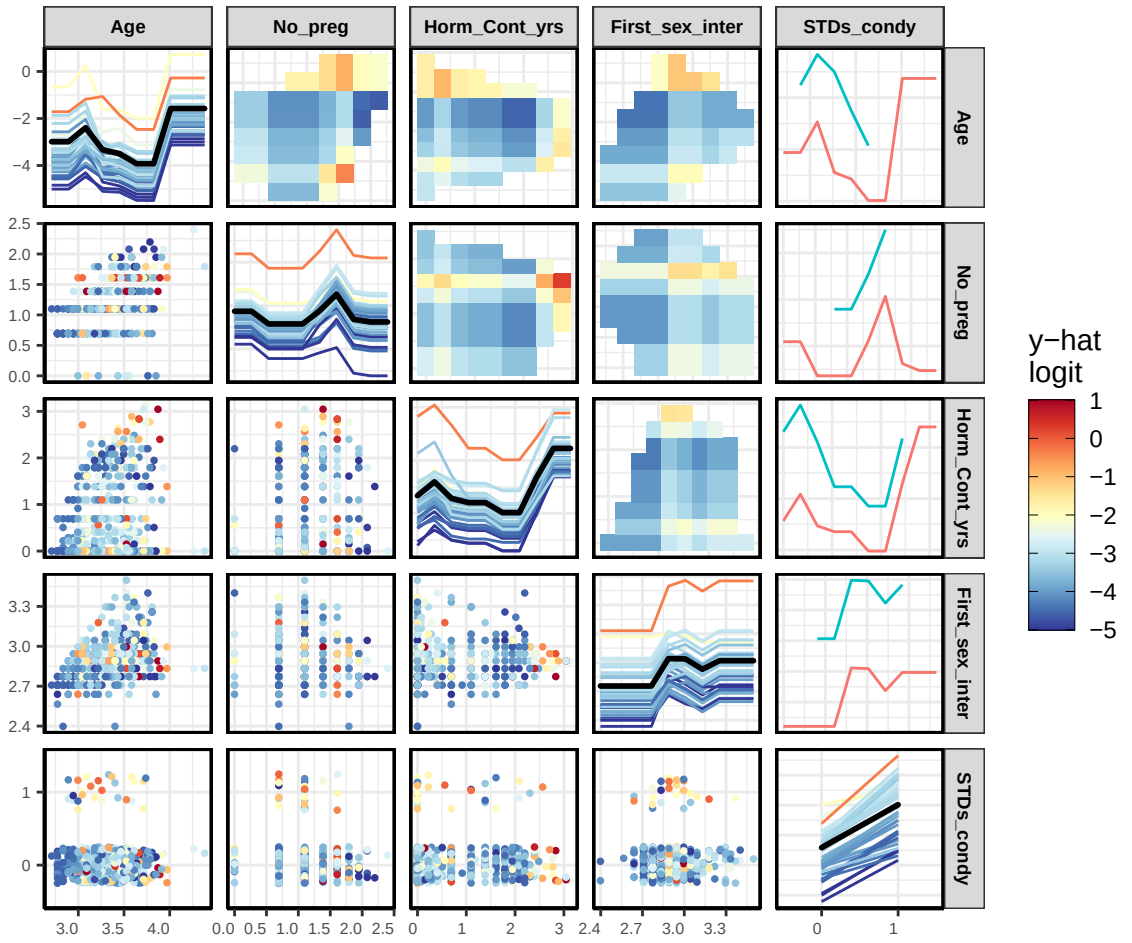
```
viviHeatmap(viv, angle = 50)
```



pdp

Figure 7:

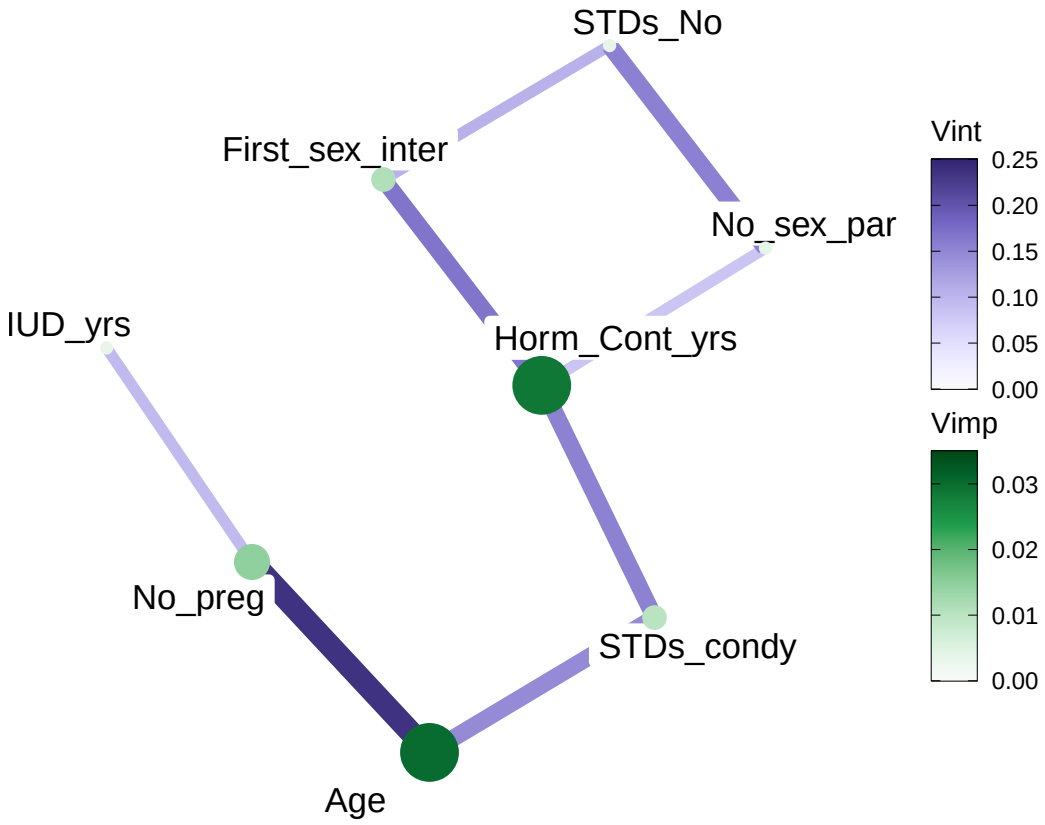
```
# Subsetting top 5 variables:
# sample 50 ice curves - 25 from positive class, 25 from negative class:
set.seed(1701)
yesRows <- sample(which(cTrain$Biopsy == "Cancer"), 25)
noRows <- sample(which(cTrain$Biopsy == "Healthy"), 25)
pdpPairs(data = cTrain,
         fit = cfit,
         response = "Biopsy",
         class = "Cancer",
         nmax = nrow(cTrain),
         nIce = c(yesRows, noRows),
         vars = colnames(viv)[1:5],
         convexHull = TRUE,
         probability = FALSE,
         fitlims = c(-5,1))
```



Network plot

Figure 8: note: layout may differ

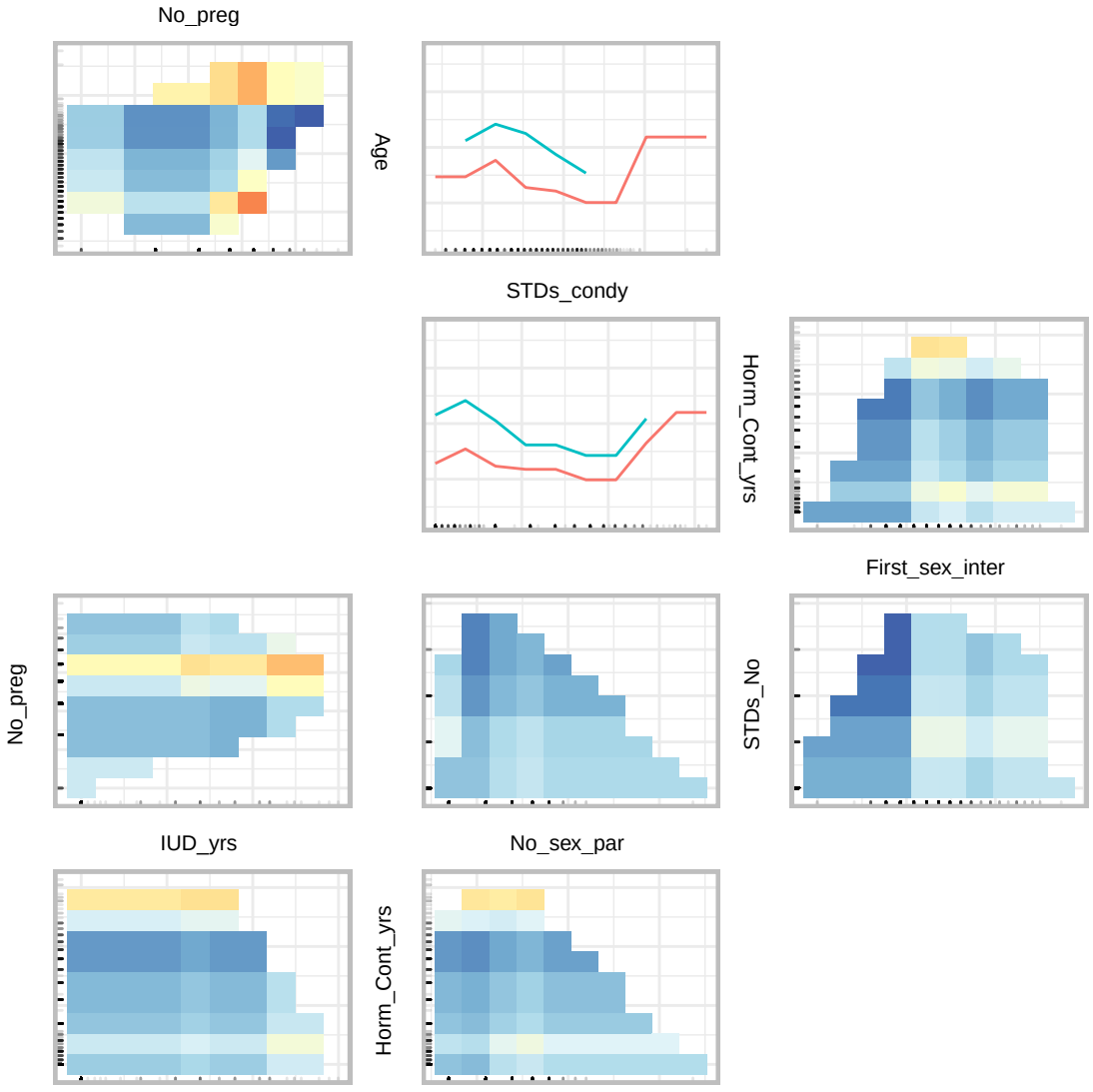
```
viviNetwork(viv, intThreshold = 0.08, removeNode = T,
            layout = igraph::layout_with_lgl
            )
```



Zen plot

Figure 9:

```
zpath <- zPath(viv, cutoff = 0.08) # same as network
set.seed(1701)
pdpZen(data = cTrain,
       fit = cfit,
       response = "Biopsy",
       class = "Cancer",
       zpath = zpath,
       convexHull = TRUE,
       probability = FALSE, fitlims = c(-5,1)
)
```


No_preg
Age
STDs_condy
Horm_Cont_yrs
First_sex_inter
No_preg
STDs_No
IUD_yrs
No_sex_par
Horm_Cont_yrs