

College Data

Alan n. Inglis

2021-10-04

Packages:

```
# install the development version of vivid:
# devtools::install_github("AlanInglis/vivid")

# Load relevant packages:
library("vivid") # for visualisations
library("ISLR") # for data
library("mlr3") # to create model
library("mlr3learners") # to create model
library("randomForest") # to create model
library("condvis2") # for predict function
library("kknn") # to get knn model
library("dplyr")
library("sp")
library("network")
```

Read in and setup data:

```
# Load data:
collegeData <- as.data.frame(College)

# Taking log values of skewed data:
collegeData <- collegeData %>%
  mutate(log(collegeData[,c(2:4,7:12)]))

# Split data into train and test
set.seed(101)
train <- sample(nrow(collegeData), round(.7*nrow(collegeData))) # split 70-30
collegeTrain <- collegeData[train, ]
collegeTest <- collegeData[-train, ]
xTest <- collegeTest[, -4]
yTest <- collegeTest$Enroll
```

Model fitting:

Fit a random forest and k-nearest neighbor models

```

# Fit a random forest model
# Used throughout Section 2:
set.seed(101)
rf <- randomForest(Enroll ~ ., data = collegeTrain)

# Check mse for rf model:
predRf <- predict(rf, newdata = collegeTest)
mspeRf <- mean((yTest - predRf)^2)
Rsqr <- 1 - sum((yTest - predRf)^2)/sum((yTest - mean(yTest))^2)
Rsqr

```

```
## [1] 0.9623167
```

```
mspeRf
```

```
## [1] 0.03424027
```

```

# Fit an mlr3 knn model
# Used in Section 2.3:
knnT <- TaskRegr$new(id = "knn", backend = collegeTrain, target = "Enroll")
knnL <- lrn("regr.kknn")
knnMod <- knnL$train(knnT)

# Check mse for knn model:
pred <- predict(knnMod, newdata = collegeTest)
mspe <- mean((yTest - pred)^2)
mspe

```

```
## [1] 0.1031804
```

Create vivid matrix

```

# Create unsorted vivid matrix for random forest fit:
# Used for Figure 1(a):
set.seed(101)
vividMatrixRF <- vivi(collegeTrain,
                      rf, "Enroll",
                      gridSize = 20,
                      reorder = FALSE)

# Sort matrix:
# Used for Figure 1(b):
vividMatrixRFSorted <- vividReorder(vividMatrixRF)

# Get agnostic VImp values instead of using random forests embedded VImps
# Used for Figure 2(b):
collegeVImps <- vivid::vividImportance.default(rf,
  collegeTrain,
  "Enroll",
  importanceType = "agnostic",
  predictFun = CVpredict
)

```

```

# Update the matrix with the new VImp values and sort:
vividMatrixRFSorted_1 <- viviUpdate(vividMatrixRFSorted, collegeVImps)
vividMatrixRFSorted_1 <- vividReorder(vividMatrixRFSorted_1)

# Create vivid matrix for mlr3 knn fit using agnostic VImp
# Used for Figure 2(a):
set.seed(101)
knnMat <- vivi(
  fit = knnMod,
  data = collegeTrain,
  response = "Enroll",
  gridSize = 20,
  importanceType = "agnostic"
)

```

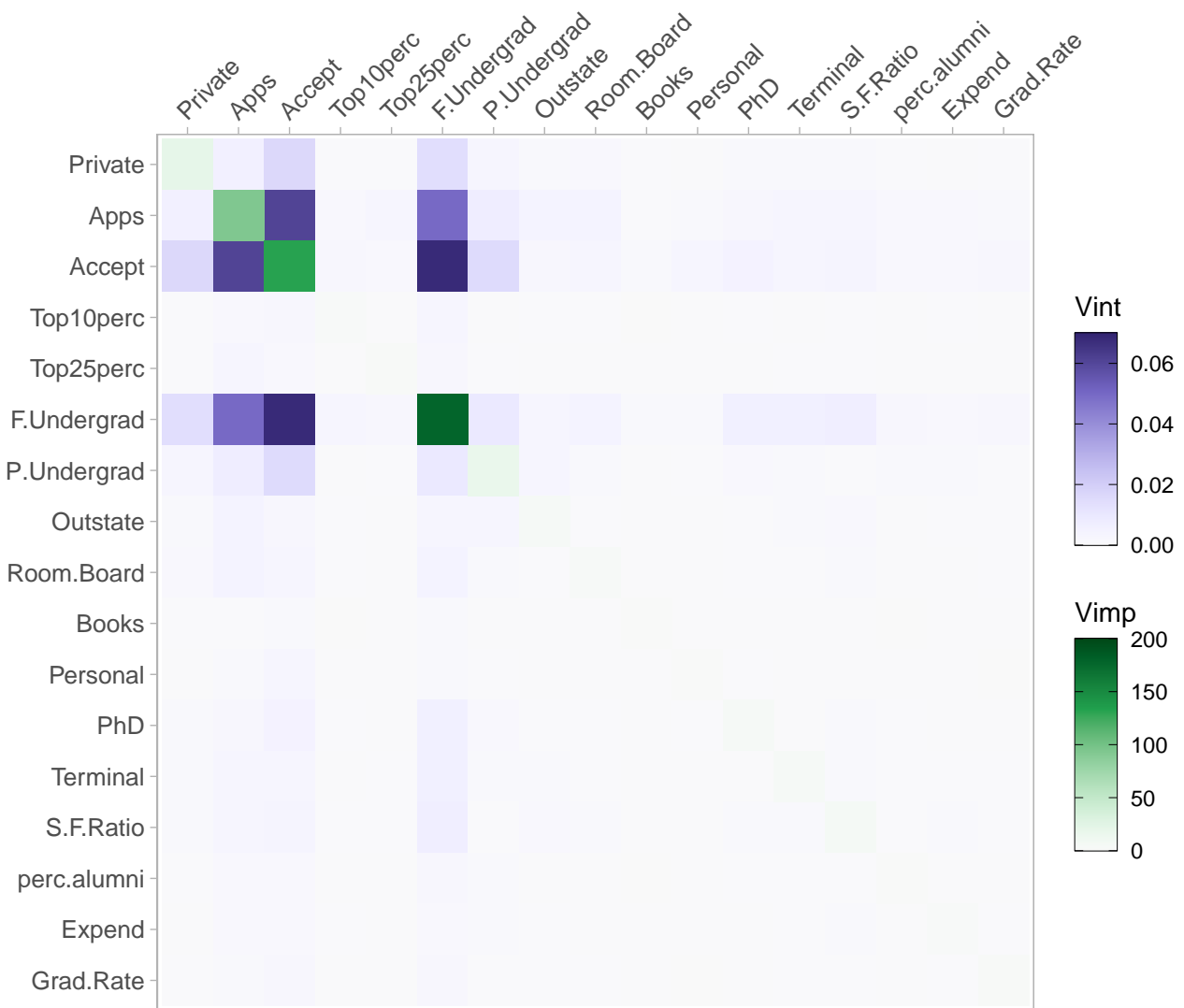
Visualisations for Section 2

Figure 1(a) & 1(b):

```

# Figure 1(a):
viviHeatmap(vividMatrixRF, angle = 45) # unsorted heatmap
# Figure 1(b):
viviHeatmap(vividMatrixRFSorted, angle = 45) # sorted heatmap

```



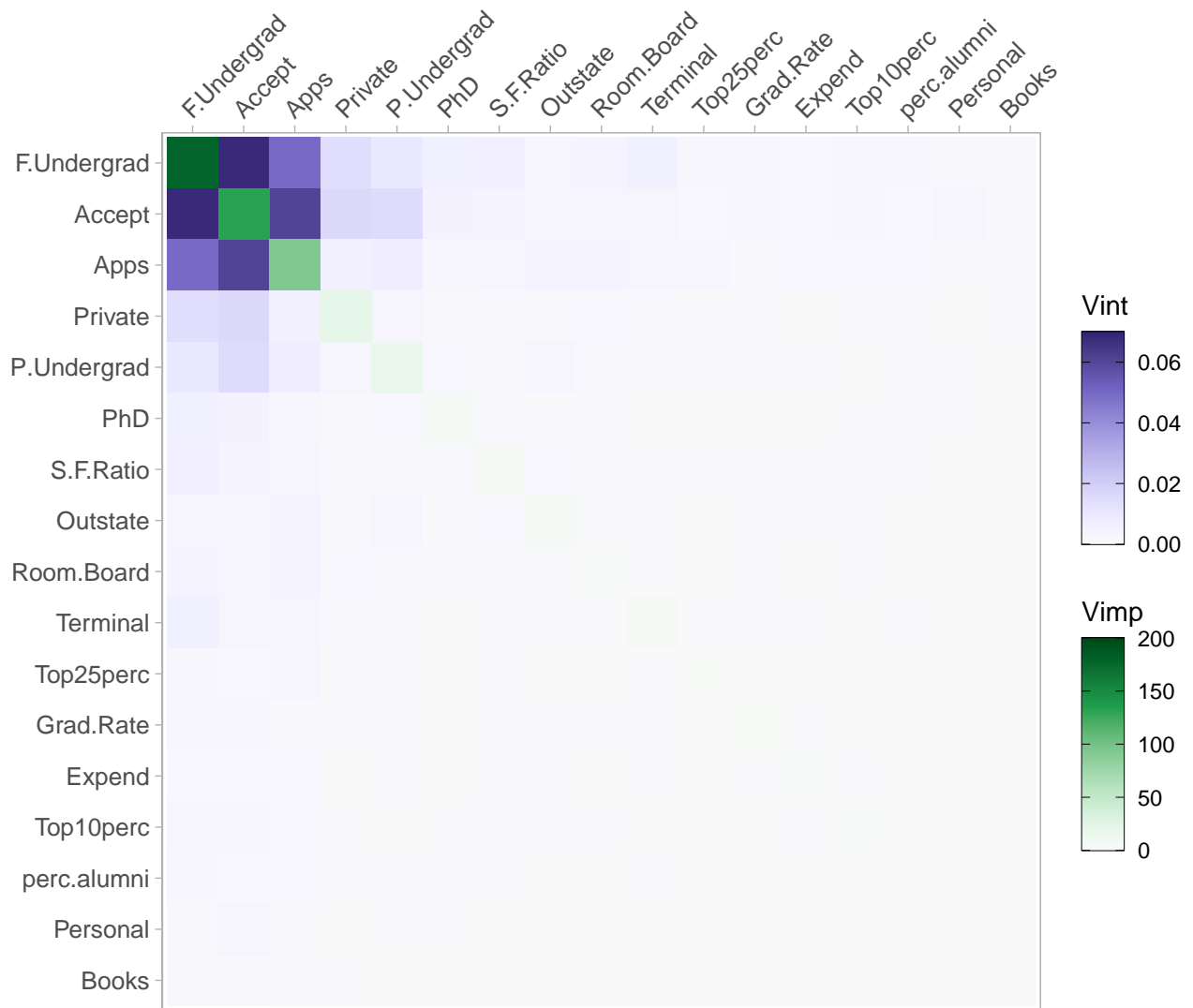
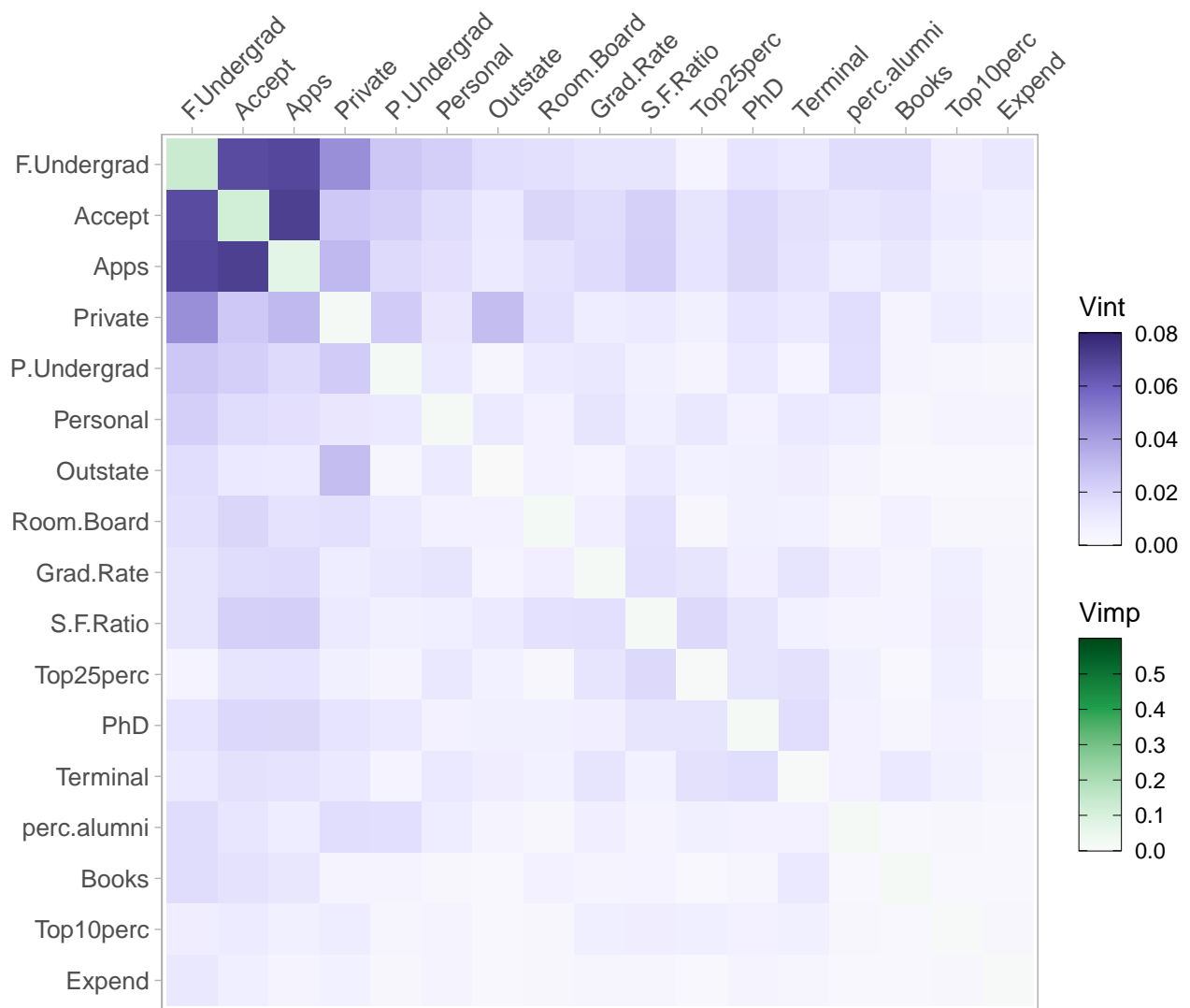


Figure 2(a) & 2(b):

```
# Figure 2(a):
viviHeatmap(knnMat, angle = 45, implims = c(0, 0.6), intLims = c(0, 0.08)) # setting same VImp limits a
# Figure 2(b)
viviHeatmap(vividMatrixRFSorted_1, angle = 45, implims = c(0, 0.6), intLims = c(0, 0.08)) # agnostic VI
```



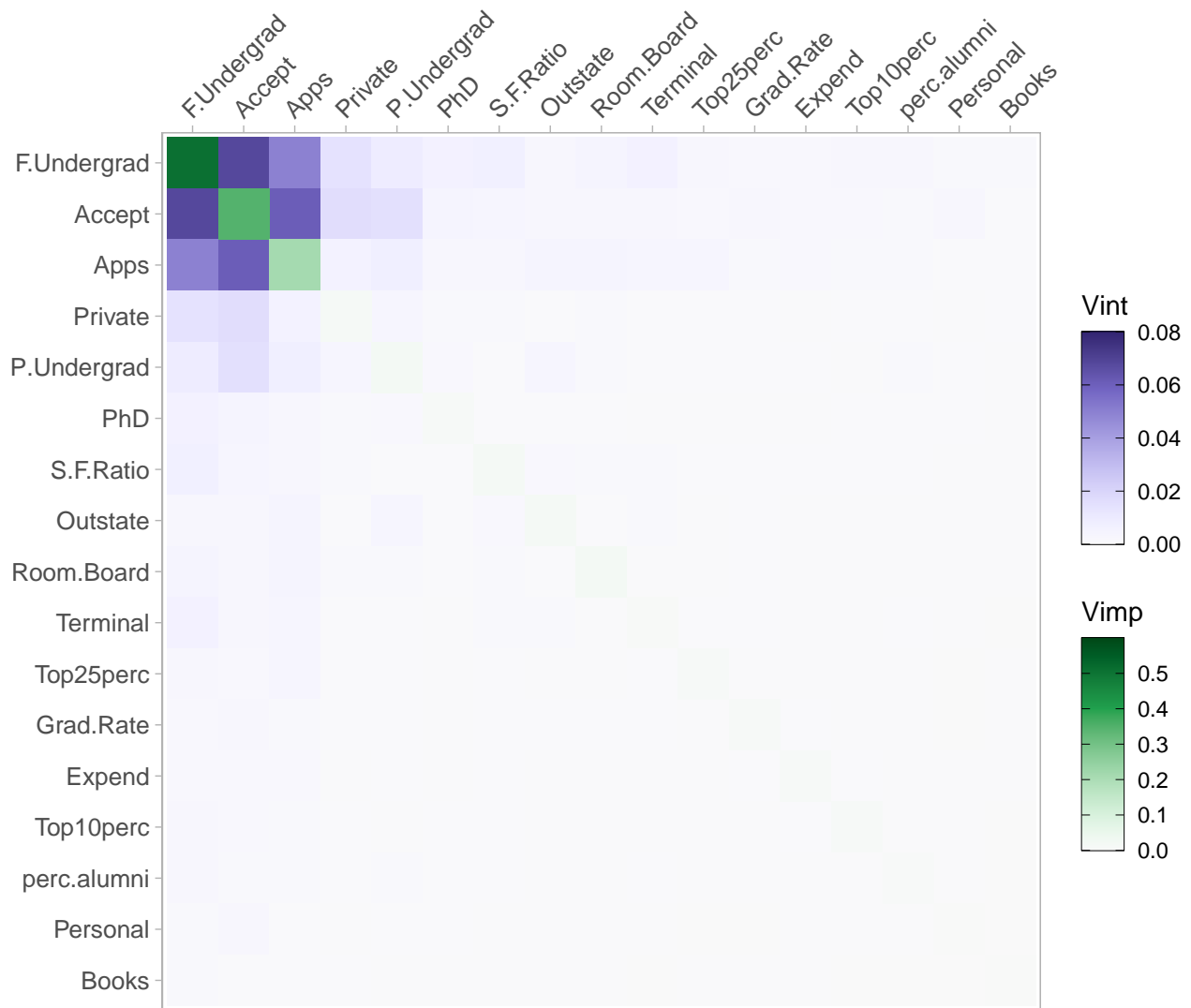
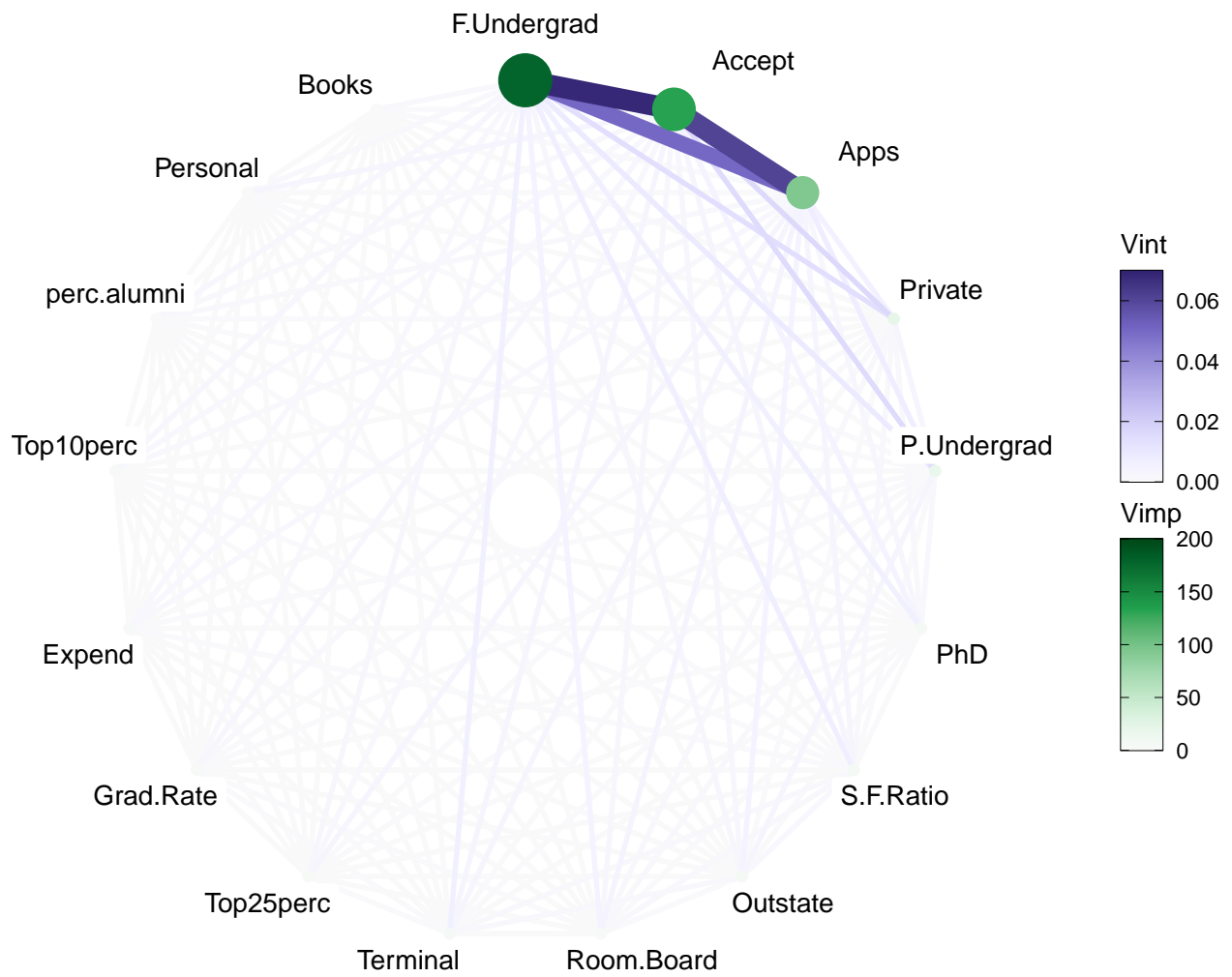
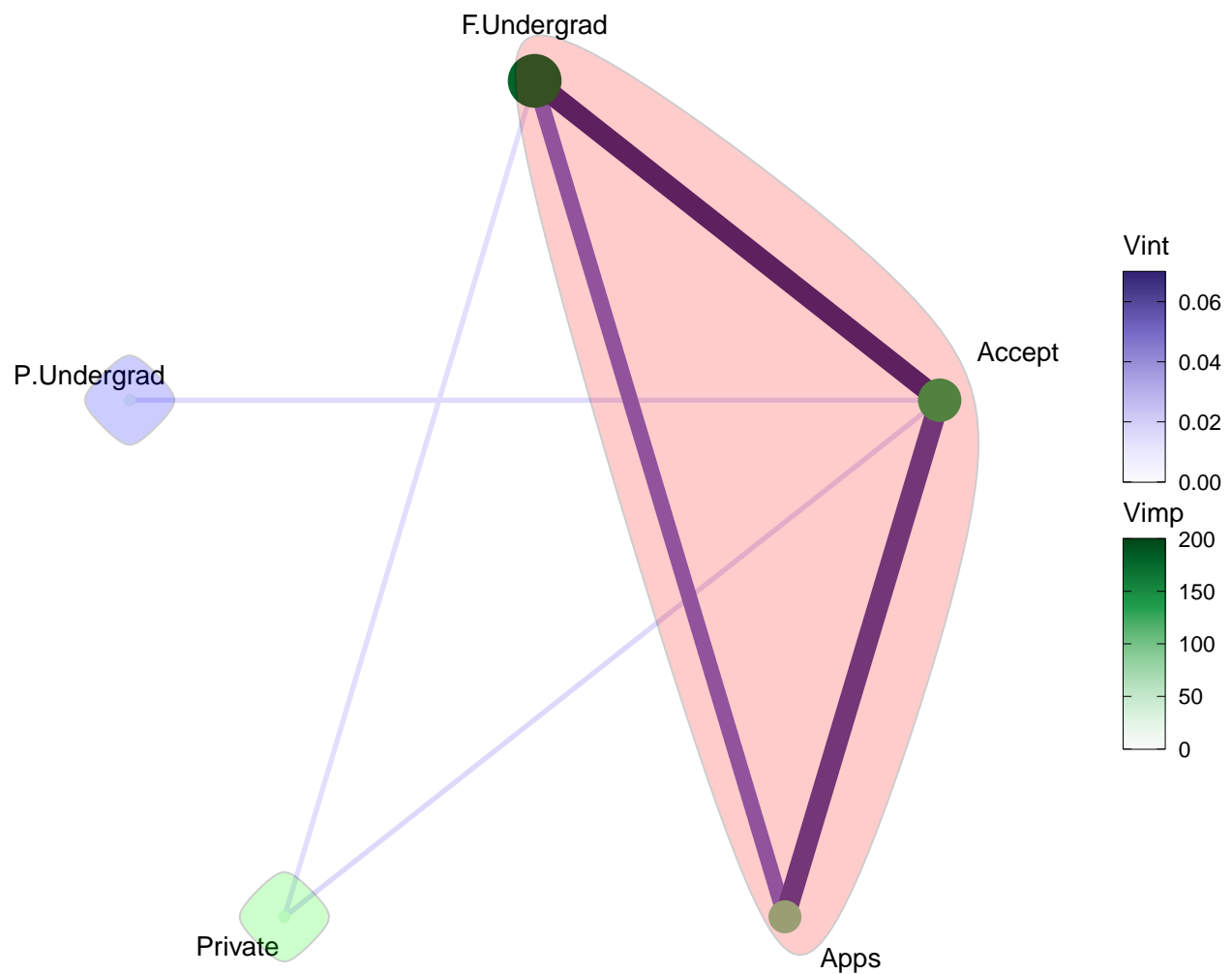


Figure 3(a) & 3(b):

```
# Figure 3(a)
viviNetwork(vividMatrixRFSorted)
# Figure 3(b)
intVals <- as.dist(vividMatrixRFSorted)
intVals <- as.matrix(intVals)
sv <- which(diag(vividMatrixRFSorted) > 50 | apply(intVals, 1,max) > .01)
h <- hclust(-as.dist(vividMatrixRFSorted[sv,sv]), method="single")
viviNetwork(vividMatrixRFSorted[sv,sv],
            intThreshold = 0.01,
            removeNode = T,
            cluster = cutree(h,3))
```



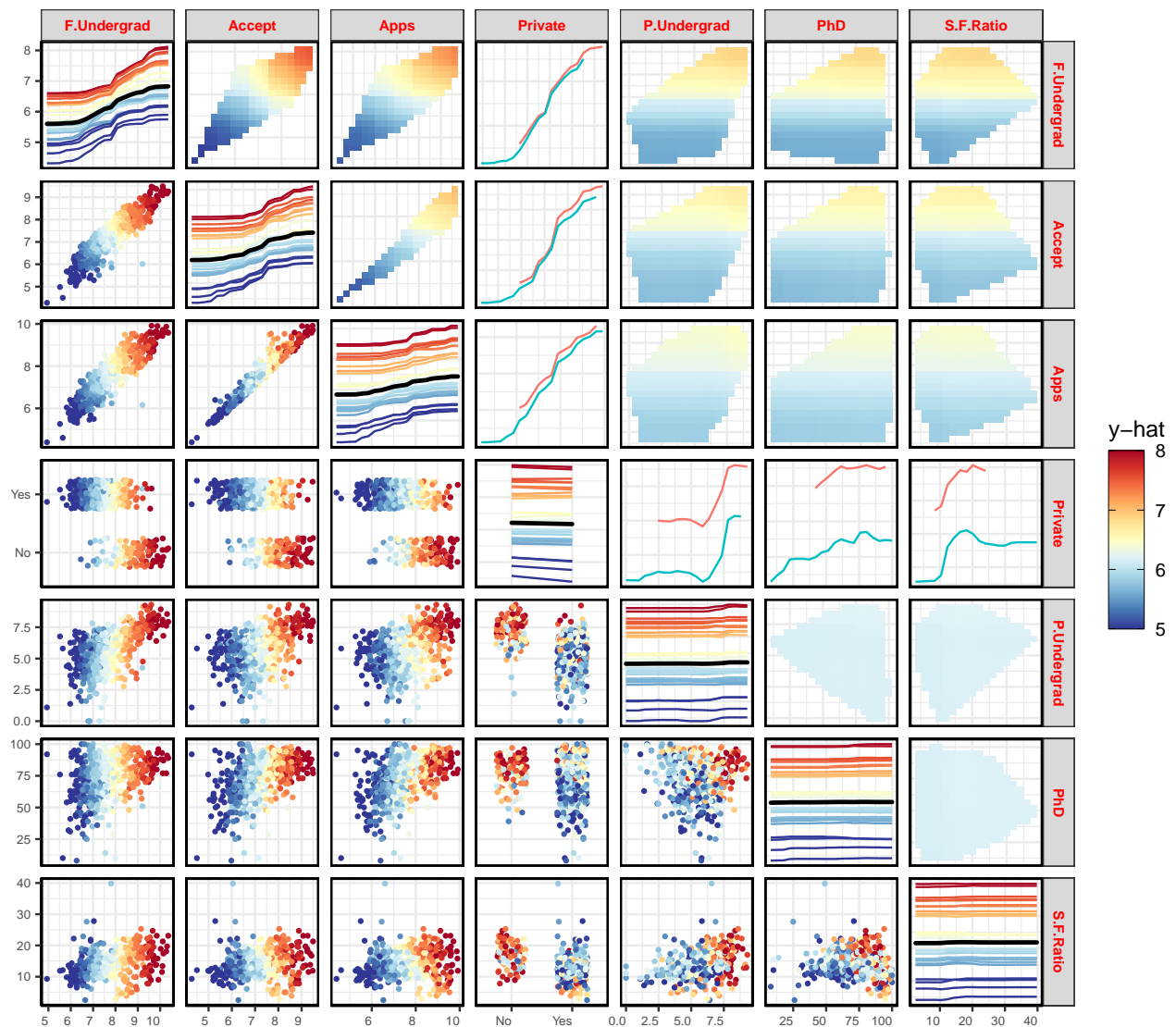


Visualisation for Section 3.2

Figure 4:

```
# Filter matrix:
nam <- colnames(vividMatrixRFSorted) # get names
nam <- nam[1:7] # filter names

# Create GPD for Figure 4:
set.seed(101)
pdpPairs(collegeTrain,
  rf, "Enroll",
  gridSize = 20,
  vars = nam,
  convexHull = TRUE
)
```

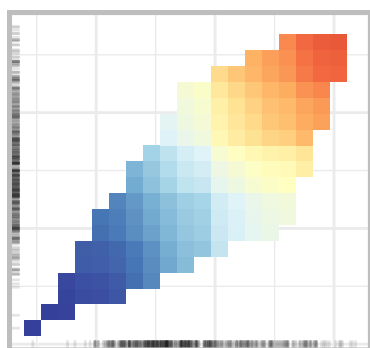


Visualisation for Section 3.3

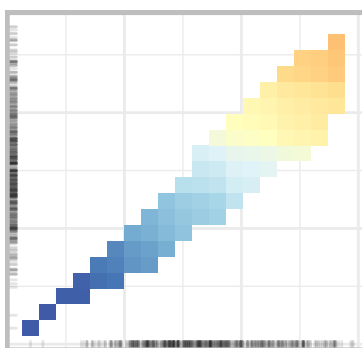
```
# Calculate the zpath using same threshold as Figure 3(b):
zpath <- zPath(vividMatrixRFSorted, 0.01)

# Create ZPDP using zpath for Figure 5:
set.seed(101)
pdpZen(collegeTrain,
  rf,
  "Enroll",
  gridSize = 20,
  zpath = zpath,
  convexHull = T
)
```

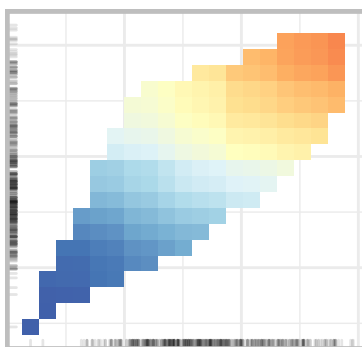
F.Undergrad



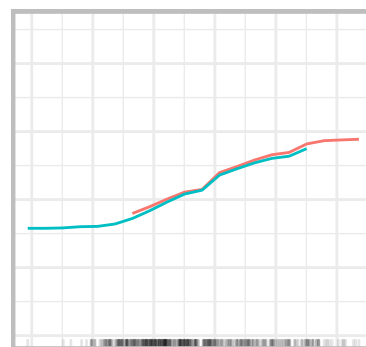
Accept



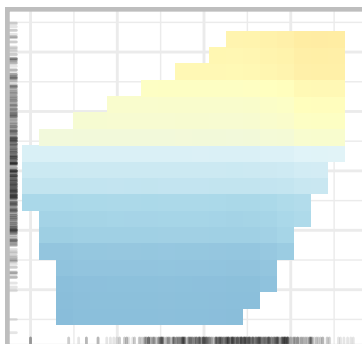
Apps



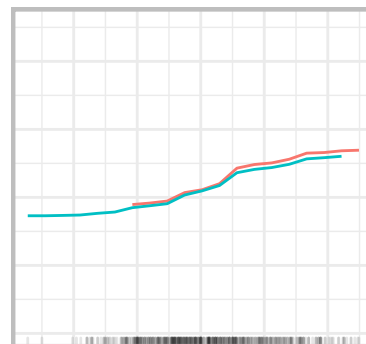
F.Undergrad



Private



Accept



P.Undergrad