STAT 4610

Michael Ghattas

16/11/2022

Chapter 6

```
library(ISLR)
library(caret)
```

```
## Warning: package 'caret' was built under R version 4.1.2
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.1.2
```

```
## Loading required package: lattice
```

```
library(tidyverse)
```

```
## Warning: package 'tidyverse' was built under R version 4.1.2
```

```
## ── Attaching packages
## ─────────────────────────────────────
## tidyverse 1.3.2 ──
```

```
## ✔ tibble  3.1.8      ✔ dplyr   1.0.10
## ✔ tidyr   1.2.1      ✔ stringr 1.4.1
## ✔ readr   2.1.3      ✔ forcats 0.5.2
## ✔ purrr   0.3.5
```

```
## Warning: package 'tibble' was built under R version 4.1.2
```

```
## Warning: package 'tidyr' was built under R version 4.1.2
```

```
## Warning: package 'readr' was built under R version 4.1.2
```

```
## Warning: package 'purrr' was built under R version 4.1.2
```

```
## Warning: package 'dplyr' was built under R version 4.1.2
```

```
## Warning: package 'stringr' was built under R version 4.1.2
```

```
## Warning: package 'forcats' was built under R version 4.1.2
```

```
## ── Conflicts ─────────────────────────────────────────
tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ✖ purrr::lift()   masks caret::lift()
```

Problem - 9

```
data('College')
```

Part (a)

```
set.seed(123)

rawData <- createDataPartition(College$Apps, p = 0.75, list = FALSE)
trainData <- College[rawData, ]
testData <- College[-rawData, ]

cleanData <- preProcess(trainData, method = c('center', 'scale'))
training <- predict(cleanData, trainData)
testing <- predict(cleanData, testData)
trainY <- training$Apps
testY <- testing$Apps

tempMod <- dummyVars(Apps ~ ., data = training)
trainX <- predict(tempMod, training)
testX <- predict(tempMod, testing)
```

Part (b)

```
linMod <- lm(Apps ~ ., data = training)
predMod <- predict(linMod, testing)

linModInfo <- postResample(predMod, testY); linModInfo
```

```
##      RMSE  Rsquared       MAE
## 0.2737462 0.8969092 0.1395384
```

Part (c)

```
lambda = seq(0, 10e2, length.out = 20)
grid = expand.grid(alpha = 0, lambda = lambda)
ridMod <- train(x = trainX, y = trainY, method = 'glmnet', trControl =
trainControl(method = 'cv', number = 10), tuneGrid = grid)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
trainInfo, :
## There were missing values in resampled performance measures.
```

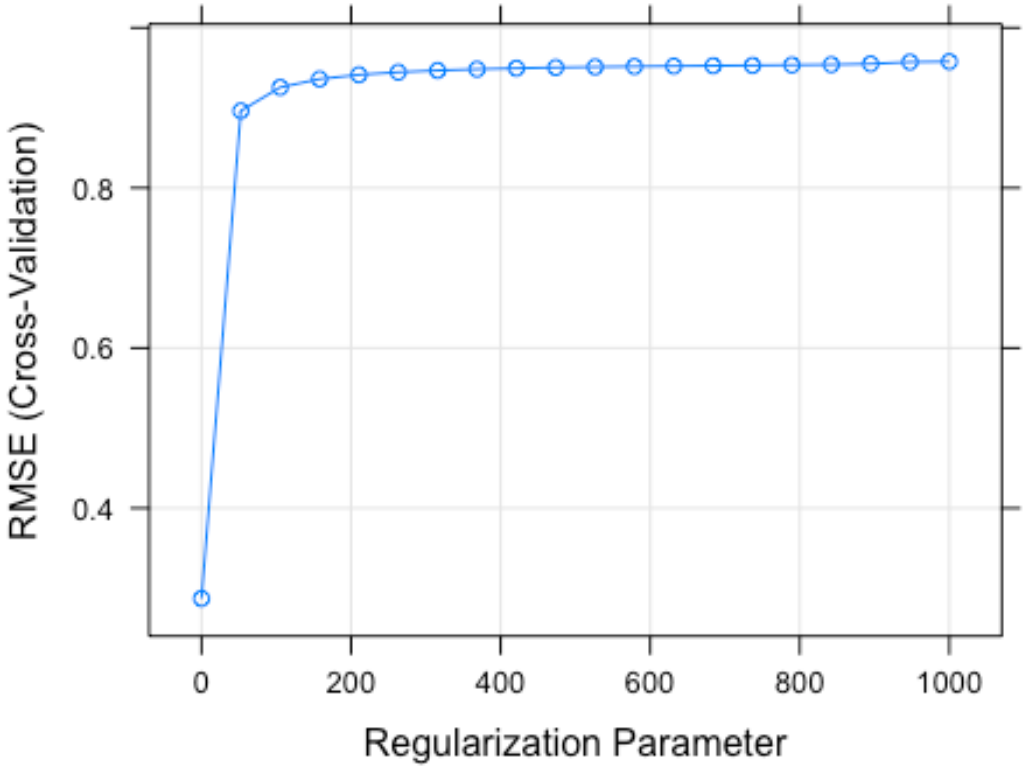
```
ridModInfo <- postResample(predict(ridMod, testX), testY); ridModInfo
```

```
##      RMSE  Rsquared       MAE
## 0.2694149 0.9006750 0.1532929
```

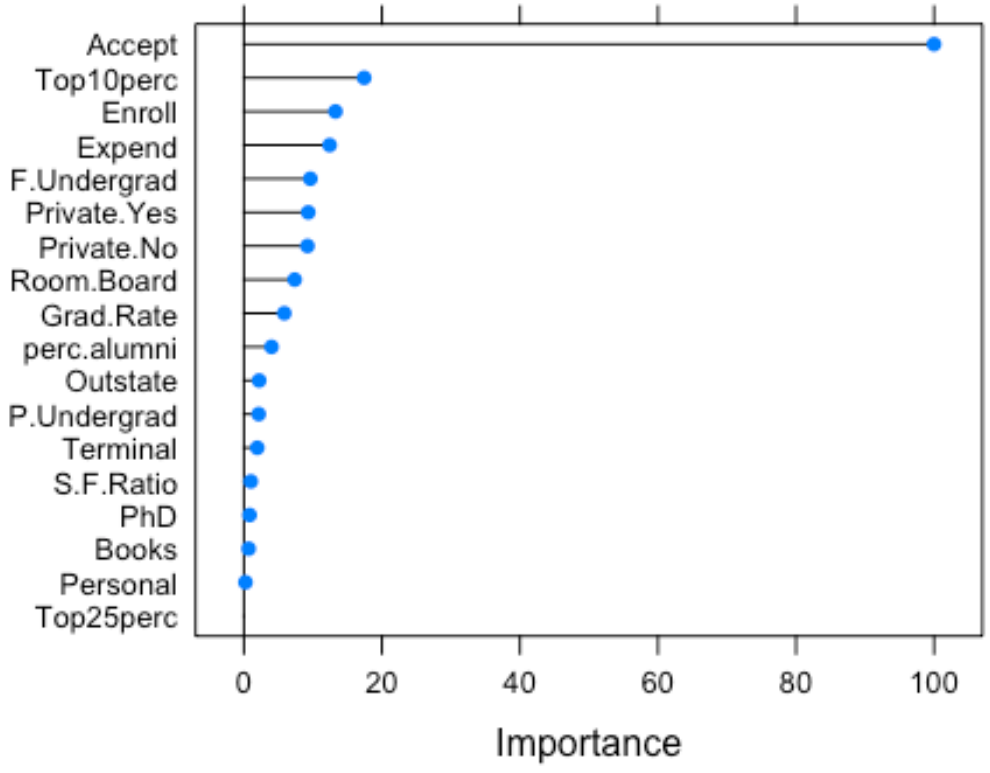
```
coef(ridMod$finalModel, ridMod$bestTune$lambda)
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                        s1
## (Intercept)  0.031972543
## Private.No   0.066368720
## Private.Yes -0.067110721
## Accept       0.665394461
## Enroll       0.093148365
## Top10perc    0.120572580
## Top25perc   -0.005576356
## F.Undergrad  0.069081346
## P.Undergrad  0.019754498
## Outstate    -0.020158852
## Room.Board   0.054132502
## Books        0.010114786
## Personal    -0.007086252
## PhD         -0.011040418
## Terminal    -0.018382693
## S.F.Ratio    0.012212536
## perc.alumni -0.031972333
## Expend       0.087574037
## Grad.Rate    0.044189957
```

```
plot(ridMod)
```




```
plot(varImp(ridMod))
```



Part (d)

```
lambda = seq(0.0001, 1, length.out = 50)
grid = expand.grid(alpha = 1, lambda = lambda)
lasMod <- train(x = trainX, y = trainY, method = 'glmnet', trControl =
trainControl(method = 'cv', number = 10), tuneGrid = grid)
```

```
## Warning in nominalTrainWorkflow(x = x, y = y, wts = weights, info =
trainInfo, :
## There were missing values in resampled performance measures.
```

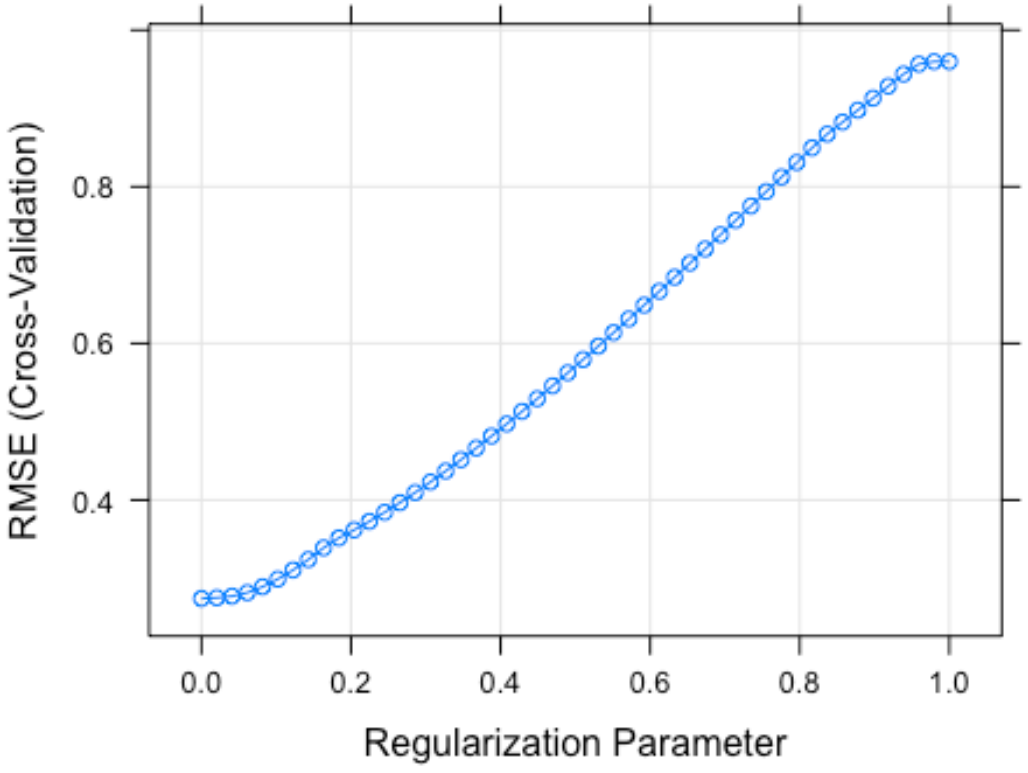
```
lasModInfo <- postResample(predict(lasMod, testX), testY); lasModInfo
```

```
##      RMSE  Rsquared       MAE
## 0.2734627 0.8974709 0.1384661
```

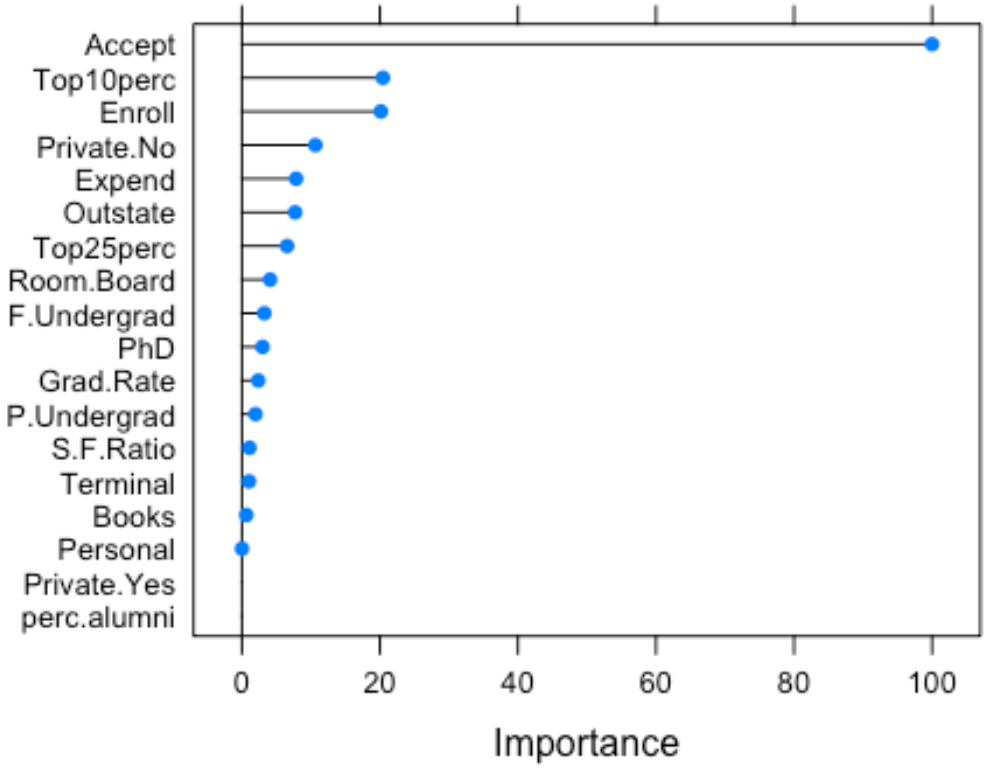
```
coef(lasMod$finalModel, lasMod$bestTune$lambda)
```

```
## 19 x 1 sparse Matrix of class "dgCMatrix"
##                       s1
## (Intercept) -0.029388060
## Private.No   0.111636461
## Private.Yes  .
## Accept       1.048520114
## Enroll      -0.211112391
## Top10perc    0.214212332
## Top25perc   -0.068649498
## F.Undergrad  0.034046186
## P.Undergrad  0.020663840
## Outstate    -0.080920555
## Room.Board   0.042834763
## Books        0.006691422
## Personal    -0.000127994
## PhD         -0.031347004
## Terminal    -0.010781176
## S.F.Ratio    0.011549859
## perc.alumni  .
## Expend       0.082405305
## Grad.Rate    0.024809336
```

```
plot(lasMod)
```




```
plot(varImp(lasMod))
```

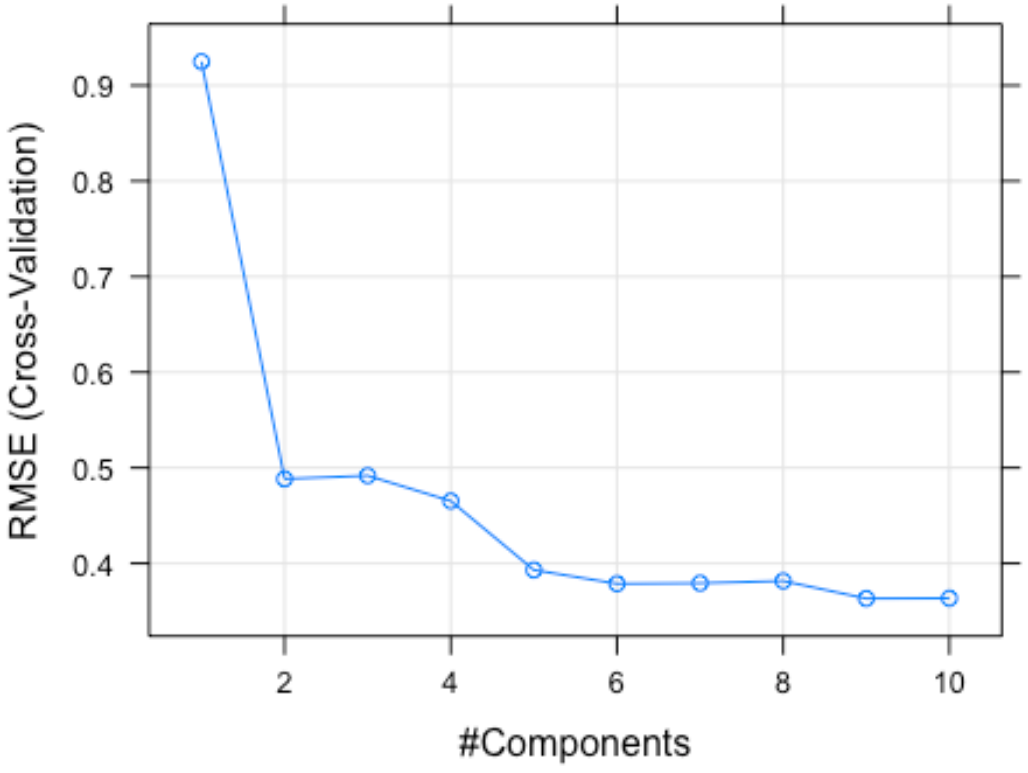


Part (e)

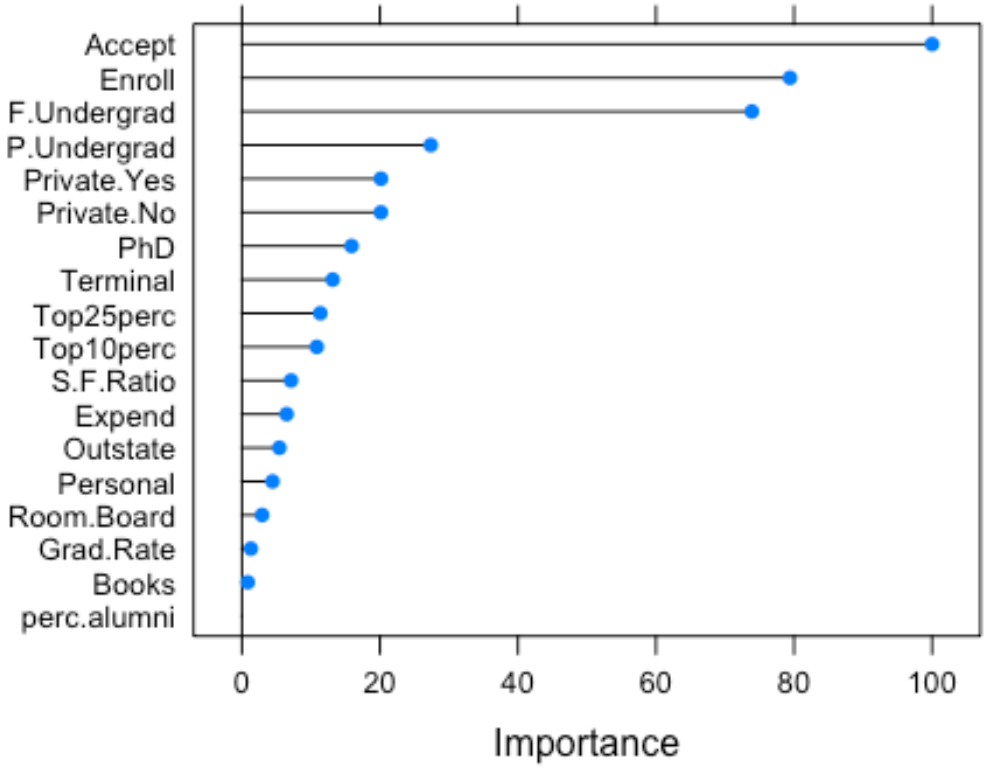
```
pcrMod <- train(x = trainX, y = trainY, method = 'pcr', trControl =
trainControl(method = 'cv', number = 10), tuneGrid = expand.grid(ncomp =
1:10))

(pcrModInfo <- postResample(predict(pcrMod, trainX), trainY))

##      RMSE  Rsquared       MAE
## 0.3838019 0.8524439 0.2003514

coef(pcrMod$finalModel)

## , , 9 comps
##
##                 .outcome
## Private.No   0.037321391
```

```
## Private.Yes -0.037321391
## Accept       0.326733843
## Enroll       0.300402683
## Top10perc    0.044137282
## Top25perc    0.032563667
## F.Undergrad  0.269060485
## P.Undergrad -0.017243124
## Outstate     0.027219287
## Room.Board   0.067745164
## Books        0.008817111
## Personal    -0.031648673
## PhD         -0.014291539
## Terminal    -0.025599219
## S.F.Ratio   -0.017274127
## perc.alumni -0.101282239
## Expend       0.090538343
## Grad.Rate    0.096372068
```

```
plot(pcrMod)
```




```
plot(varImp(pcrMod))
```

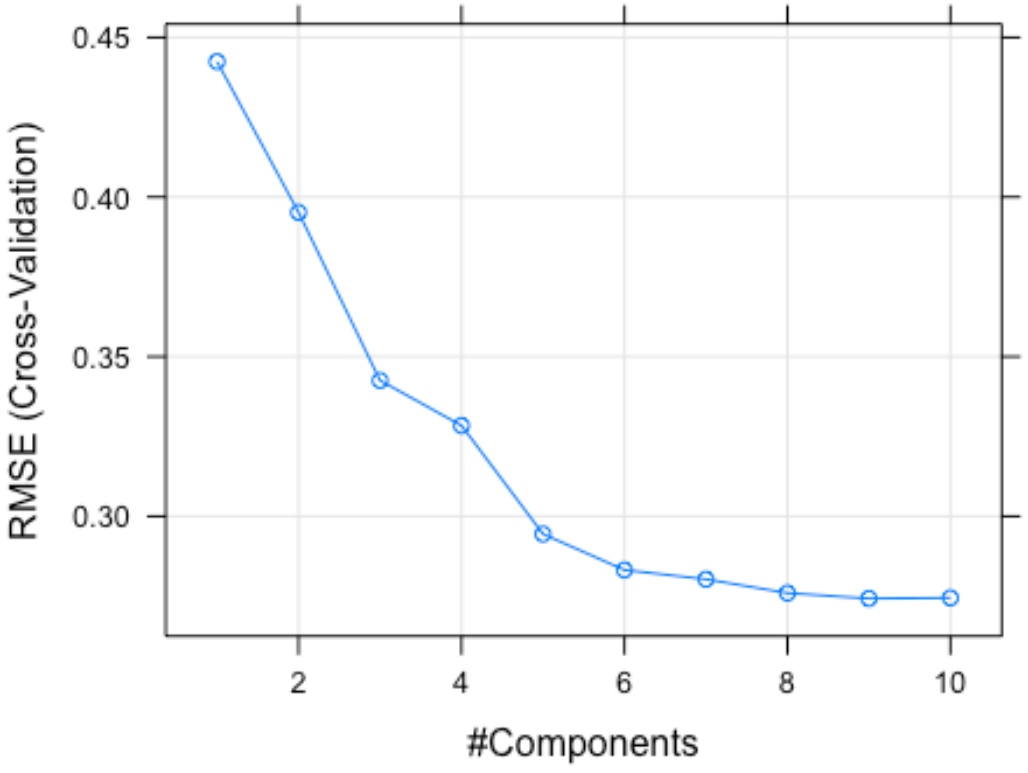


Part (f)

```
plsMod <- train(x = trainX, y = trainY, method = 'pls', trControl =
trainControl(method = 'cv', number = 10), tuneGrid = expand.grid(ncomp =
1:10))

(plsModInfo <- postResample(predict(plsMod, trainX), trainY))

##      RMSE  Rsquared       MAE
## 0.2533850 0.9356861 0.1522649

coef(plsMod$finalModel)

## , , 9 comps
##
##                   .outcome
## Private.No   0.0675989688
```

```
## Private.Yes -0.0675989688
## Accept       1.0388829006
## Enroll      -0.1442721428
## Top10perc    0.2336138321
## Top25perc   -0.0806025836
## F.Undergrad -0.0326477856
## P.Undergrad  0.0220602093
## Outstate    -0.0977014264
## Room.Board   0.0503138399
## Books        0.0097271362
## Personal    -0.0039961166
## PhD         -0.0470487857
## Terminal     0.0056682578
## S.F.Ratio    0.0119700918
## perc.alumni -0.0002917296
## Expend       0.0776694508
## Grad.Rate    0.0332106657
```

```
plot(plsMod)
```



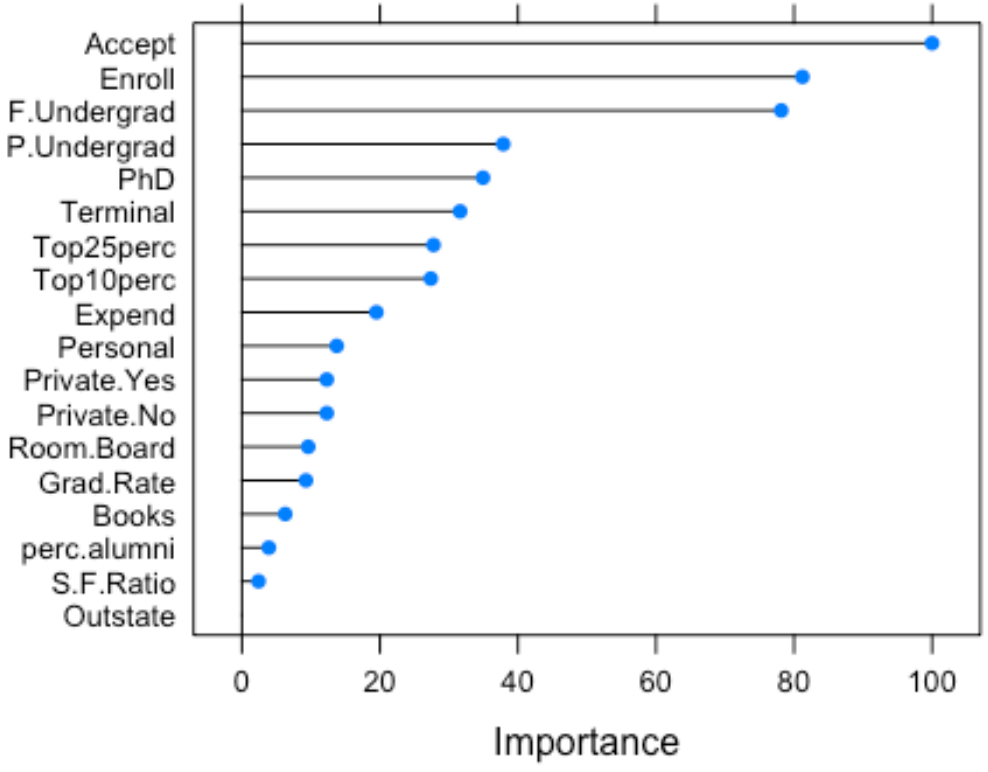

```
plot(varImp(plsMod))
```

```
## Warning: package 'pls' was built under R version 4.1.2
```

```
##
## Attaching package: 'pls'
```

```
## The following object is masked from 'package:caret':
##
##     R2
```

```
## The following object is masked from 'package:stats':
##
##     loadings
```



Part (g)

```
as.data.frame(rbind(linModInfo, ridModInfo, lasModInfo, pcrModInfo,
plsModInfo)) %>% mutate(model = c('2nd Worst', '2nd Best', 'Neutral',
'Worst', 'Best')) %>% select(model, RMSE, Rsquared)

##                model      RMSE  Rsquared
## linModInfo 2nd Worst 0.2737462 0.8969092
## ridModInfo  2nd Best 0.2694149 0.9006750
## lasModInfo   Neutral 0.2734627 0.8974709
## pcrModInfo     Worst 0.3838019 0.8524439
## plsModInfo      Best 0.2533850 0.9356861
```

-> Overall, we can note that almost all the models perform well, with the PLS model being the best and having the lowest RMSE and highest R^2.