N741 Spring 2018 - Homework 7

Homework 7 - DUE WED April 11, 2018

Lacey Gleason

April 10, 2018

Homework 7

Background and Information on HELP Dataset

For homework 7, you will be working with the **HELP** (Health Evaluation and Linkage to Primary Care) Dataset. See complete details posted in Homework 6.

Variables for Homework 7

For Homework 7, you will focus on these variables from the HELP dataset:

Use these variables from HELP dataset for Homework 07

	Variable Label
age	Age at baseline (in years)
female	Gender of respondent
pss_fr	Perceived Social Support - friends
homeless	One or more nights on the street or shelter in past 6 months
pcs	SF36 Physical Composite Score - Baseline
mcs	SF36 Mental Composite Score - Baseline
cesd	CESD total score - Baseline
cesd_gte16	Indicator of Depression
mcs_lt45	Indicator of Poor Mental Health

Homework 7 Assignment

SETUP Download and run the "loadHELP.R" `R` script (included in this Github repo https://github.com/melindahiggins2000/N741Spring2018_Homework7) to read in the HELP Dataset "helpmkh.sav". This script also pulls out the variables you need and creates the dichotomous variable for depression `cesd_gte16` **AND** a dichotomous variable to indicate poor mental health (`mcs_lt45`).

```
# use this script to setup the data subset from
# HELP to use for N741 Spring 2018 Homework 7

# load libraries and dataset
```

```r
library(tidyverse)
library(haven)
helpdata <- haven::read_spss("helpmkh.sav")

# choose variables for Homework 6

h1 <- helpdata %>%
  select(age, female, pss_fr, homeless,
         pcs, mcs, cesd)

# add dichotomous variable
# to indicate depression for
# people with CESD scores >= 16
# and people with mcs scores < 45

h1 <- h1 %>%
  mutate(cesd_gte16 = cesd >= 16) %>%
  mutate(mcs_lt45 = mcs < 45)

# change cesd_gte16 and mcs_lt45 LOGIC variable type
# to numeric coded 1=TRUE and 0=FALSE

h1$cesd_gte16 <- as.numeric(h1$cesd_gte16)
h1$mcs_lt45 <- as.numeric(h1$mcs_lt45)

# check final data subset h1
summary(h1)
```

```
##       age            female           pss_fr          homeless
##  Min.   :19.00   Min.   :0.0000   Min.   : 0.000   Min.   :0.0000
##  1st Qu.:30.00   1st Qu.:0.0000   1st Qu.: 3.000   1st Qu.:0.0000
##  Median :35.00   Median :0.0000   Median : 7.000   Median :0.0000
##  Mean   :35.65   Mean   :0.2362   Mean   : 6.706   Mean   :0.4614
##  3rd Qu.:40.00   3rd Qu.:0.0000   3rd Qu.:10.000   3rd Qu.:1.0000
##  Max.   :60.00   Max.   :1.0000   Max.   :14.000   Max.   :1.0000
##       pcs             mcs              cesd          cesd_gte16
##  Min.   :14.07   Min.   : 6.763   Min.   : 1.00   Min.   :0.0000
##  1st Qu.:40.38   1st Qu.:21.676   1st Qu.:25.00   1st Qu.:1.0000
##  Median :48.88   Median :28.602   Median :34.00   Median :1.0000
##  Mean   :48.05   Mean   :31.677   Mean   :32.85   Mean   :0.8985
##  3rd Qu.:56.95   3rd Qu.:40.941   3rd Qu.:41.00   3rd Qu.:1.0000
##  Max.   :74.81   Max.   :62.175   Max.   :60.00   Max.   :1.0000
##     mcs_lt45
##  Min.   :0.0000
##  1st Qu.:1.0000
##  Median :1.0000
##  Mean   :0.8146
##  3rd Qu.:1.0000
##  Max.   :1.0000
```

For Homework 7, the code is provided here for the regression tree and conditional tree and random forest models looking at depression as given by the continuous measure cesd and the dichotomous indicator of depression cesd_gte16.

You can then use this code and adapt it to run through the models again looking at the mental health composite score (mcs) in these subjects and the dichomotous indicator or poor mental health for people with mcs scores < 45, which is the variable mcs_lt45.

Packages needed for Homework 7

- rpart
- partykit
- party
- tidyverse
- reshape2
- randomForestSRC
- ggRandomForests

```
library(rpart)
library(partykit)
library(reshape2)
library(party)
library(tidyverse)
library(randomForestSRC)
library(ggRandomForests)
```

PROBLEM 1: Regression Tree for MCS

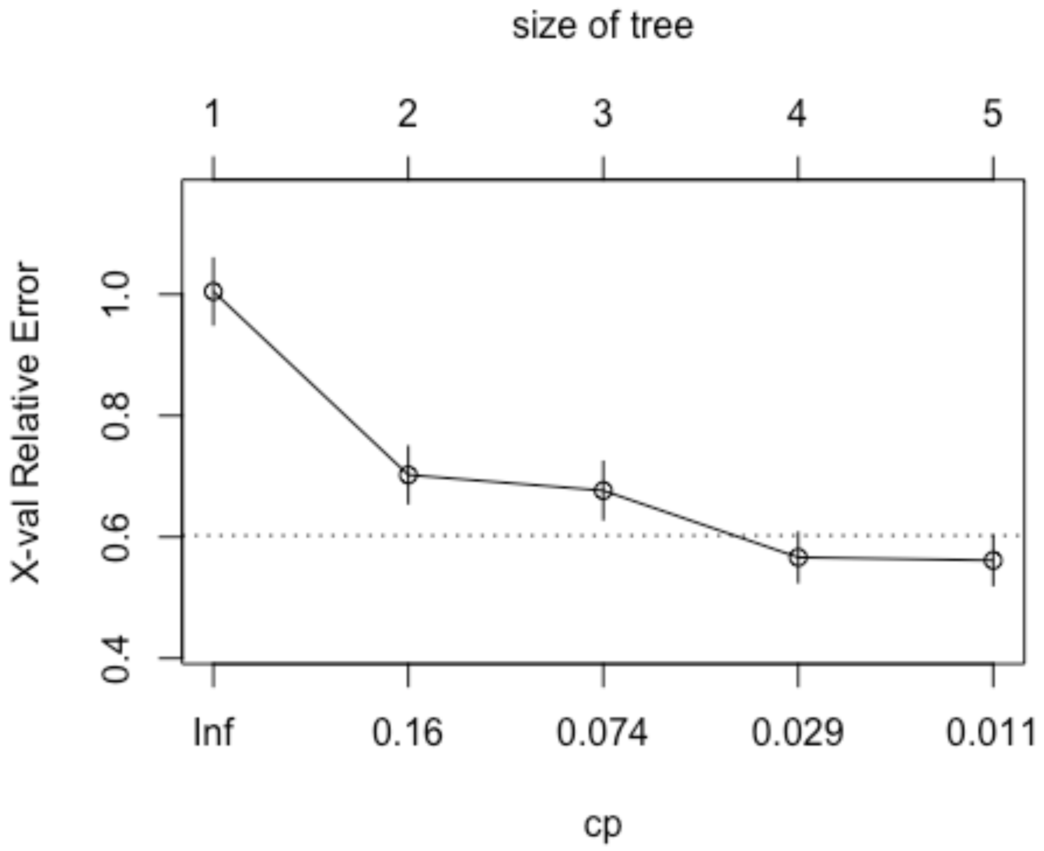
Using the code above, fit a regression tree model where the mcs is the outcome and the cesd is the predictor and complete the following:

- fit a regression tree to the mcs based on only the cesd scores from the h1 dataset;
- display the results
- plot the cross-validated results
- provide a summary of the model fit
- and plot the regression tree

```
# insert code to complete Problem 1 here

# fit a regression tree model to the cesd as the outcome
# and using the mcs as the only predictor
fitcesd <- rpart::rpart(mcs ~ cesd, data = h1)
rpart::printcp(fitcesd) # Display the results

##
## Regression tree:
## rpart::rpart(formula = mcs ~ cesd, data = h1)
```

```
##
## Variables actually used in tree construction:
## [1] cesd
##
## Root node error: 74512/453 = 164.48
##
## n= 453
##
##         CP nsplit rel error  xerror     xstd
## 1 0.325298      0   1.00000 1.00470 0.054569
## 2 0.081349      1   0.67470 0.70231 0.047024
## 3 0.066496      2   0.59335 0.67645 0.047704
## 4 0.012496      3   0.52686 0.56635 0.041113
## 5 0.010000      4   0.51436 0.56143 0.041026
```

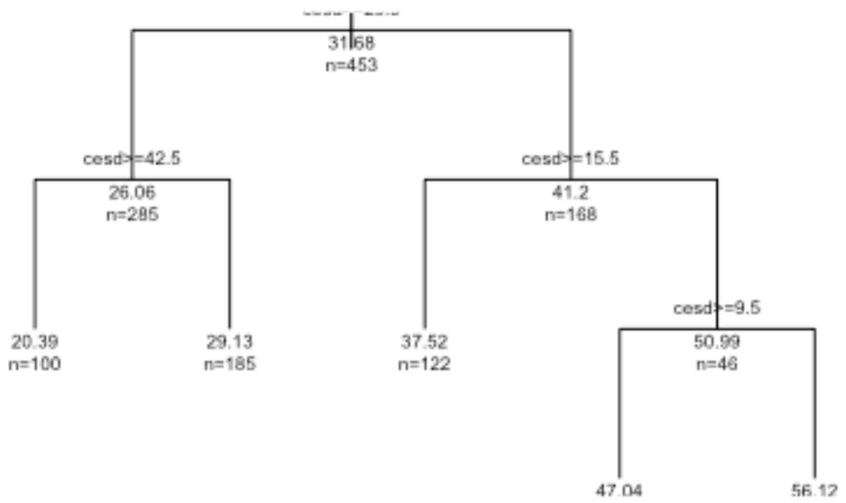
```
rpart::plotcp(fitcesd) # Visualize cross-validation results
```



```
summary(fitcesd) # Detailed summary of fit
```

```
## Call:
## rpart::rpart(formula = mcs ~ cesd, data = h1)
##   n= 453
##
##            CP nsplit rel error    xerror       xstd
## 1 0.32529813      0 1.0000000 1.0046981 0.05456905
## 2 0.08134904      1 0.6747019 0.7023073 0.04702370
## 3 0.06649553      2 0.5933528 0.6764542 0.04770415
## 4 0.01249609      3 0.5268573 0.5663467 0.04111294
## 5 0.01000000      4 0.5143612 0.5614298 0.04102641
##
## Variable importance
## cesd
##  100
##
## Node number 1: 453 observations,    complexity param=0.3252981
##   mean=31.67668, MSE=164.4847
##   left son=2 (285 obs) right son=3 (168 obs)
##   Primary splits:
##       cesd < 29.5 to the right, improve=0.3252981, (0 missing)
##
## Node number 2: 285 observations,    complexity param=0.06649553
##   mean=26.06057, MSE=100.1894
##   left son=4 (100 obs) right son=5 (185 obs)
##   Primary splits:
##       cesd < 42.5 to the right, improve=0.17352, (0 missing)
##
## Node number 3: 168 observations,    complexity param=0.08134904
##   mean=41.20401, MSE=129.2805
##   left son=6 (122 obs) right son=7 (46 obs)
##   Primary splits:
##       cesd < 15.5 to the right, improve=0.2790834, (0 missing)
##
## Node number 4: 100 observations
##   mean=20.38941, MSE=43.95751
##
## Node number 5: 185 observations
##   mean=29.12606, MSE=103.8029
##
## Node number 6: 122 observations
##   mean=37.51566, MSE=103.6988
##
## Node number 7: 46 observations,    complexity param=0.01249609
##   mean=50.98616, MSE=65.35702
##   left son=14 (26 obs) right son=15 (20 obs)
##   Primary splits:
##       cesd < 9.5  to the right, improve=0.3097046, (0 missing)
##
## Node number 14: 26 observations
##   mean=47.04024, MSE=67.29195
##
## Node number 15: 20 observations
##   mean=56.11586, MSE=16.28645
```

```
# plot tree
plot(fitcesd, uniform = TRUE, compress = FALSE)
text(fitcesd, use.n = TRUE, all = TRUE, cex = 0.5)
```

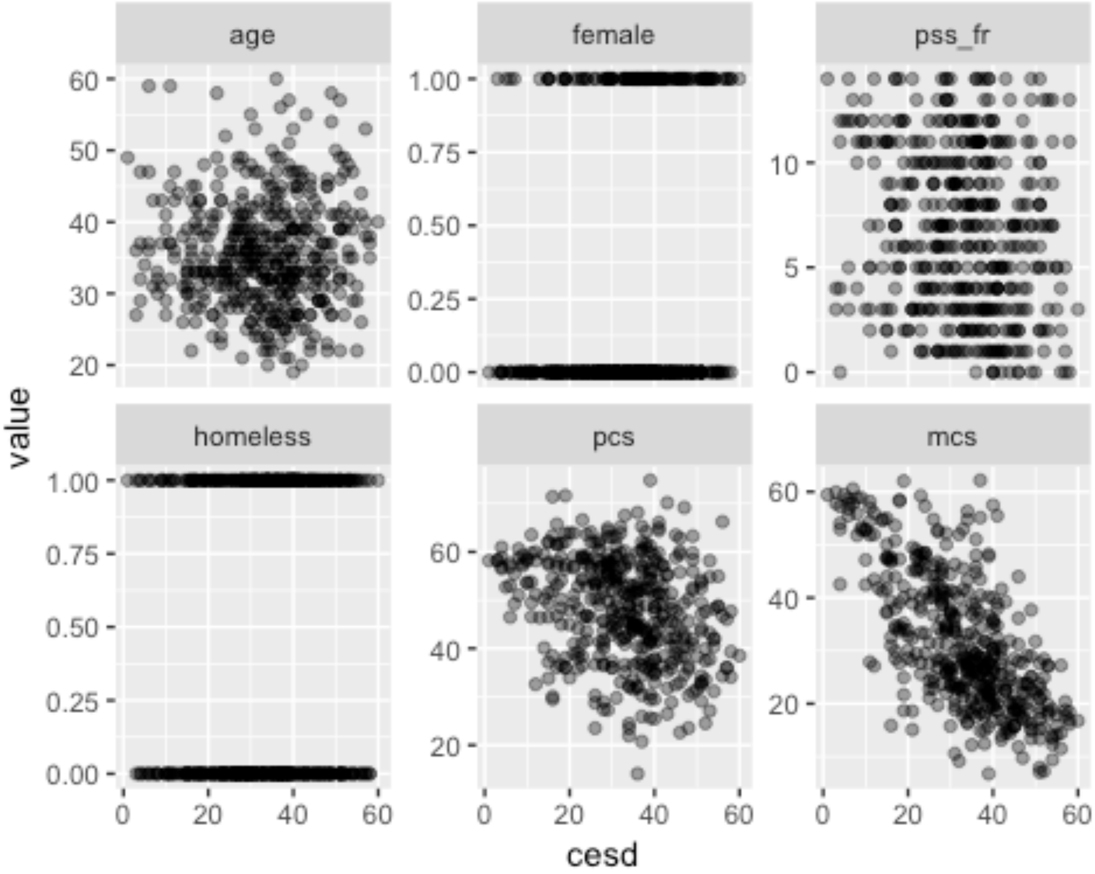


Matrix Scatterplot of Other Variables with CESD

We can use the `reshape2` package to basically stack all of the other variables on top of one another and align them with the `cesd` variable and then use this "melted" dataset with the `facet_wrap` option with `ggplot()` to basically get a matrix of scatterplots showing how all of the other variables are associated with the `cesd`.

I also first remove the variables I don't need for this next step and create the dataset `h1a`.

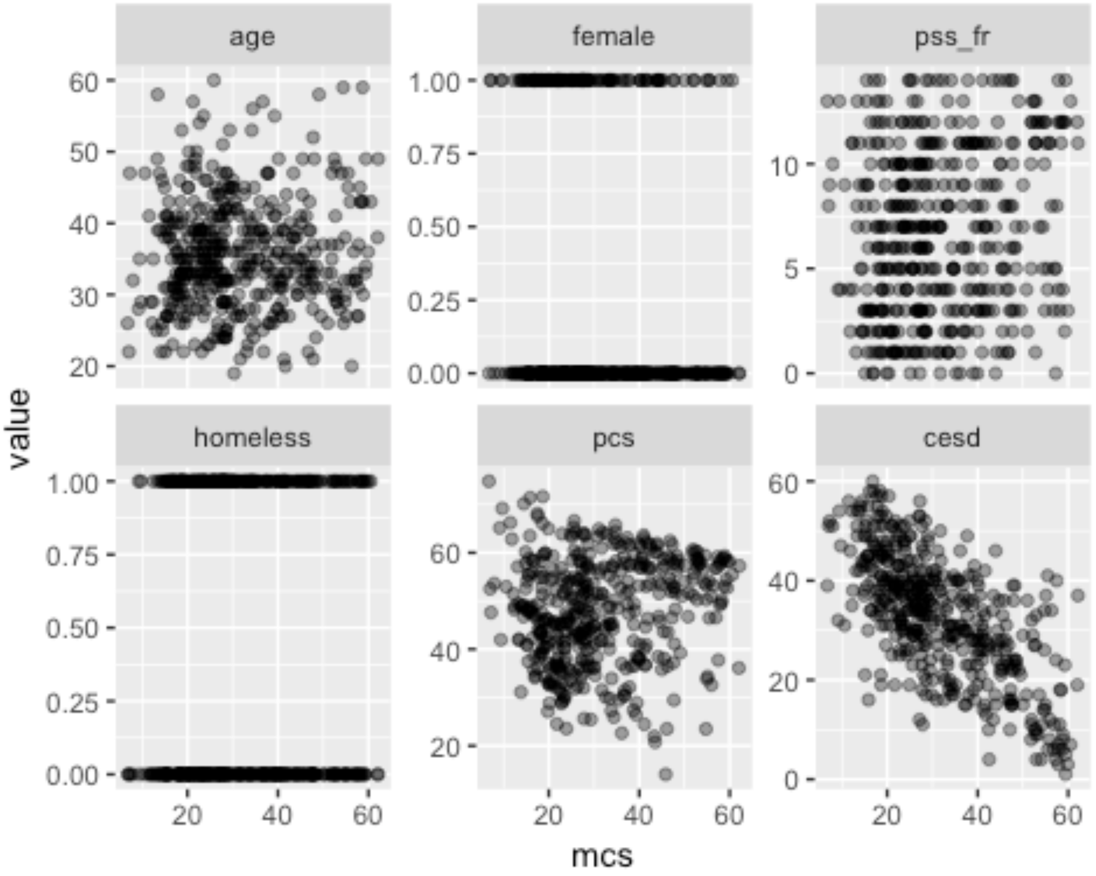
```
# all vars except the dichotomous cesd_gte16 and mcs_lt45
h1a <- h1[,1:7]

# Melt the other variables down and link to cesd
h1m <- reshape2::melt(h1a, id.vars = "cesd")

# Plot panels for each covariate
ggplot(h1m, aes(x=cesd, y=value)) +
  geom_point(alpha=0.4)+
  scale_color_brewer(palette="Set2")+
  facet_wrap(~variable, scales="free_y", ncol=3)
```



PROBLEM 2: Matrix Scatterplot of Other Variables with MCS

Using the code above as a guide,swap out `mcs` for `cesd` and redo the scatterplots compared to the `mcs`. HINT: You can begin with the data subset `h1a`, but you will need to modify the code for `h1m` and for the `ggplot()` code lines.

```
# all vars except the dichotomous cesd_gte16 and mcs_lt45
h1a <- h1[,1:7]

# Melt the other variables down and link to cesd
h1m <- reshape2::melt(h1a, id.vars = "mcs")

# Plot panels for each covariate
ggplot(h1m, aes(x=mcs, y=value)) +
  geom_point(alpha=0.4)+
  scale_color_brewer(palette="Set2")+
  facet_wrap(~variable, scales="free_y", ncol=3)
```



Regression Tree for CESD with the rest of the variables

Now let's see what happens when we include the rest of the variables. A "shorthand" notation used in R that can be handy is to simply put in a period "." indicating use the rest of the variables in the model.

So, the line of code

```
fitall <- rpart::rpart(cesd ~ ., data = h1a)
```

basically says to fit a model for `cesd` from the rest of the variables in the dataset `h1a` which includes:

- `age`
- `female`
- `pss_fr`
- `homeless`
- `pcs`
- `mcs`

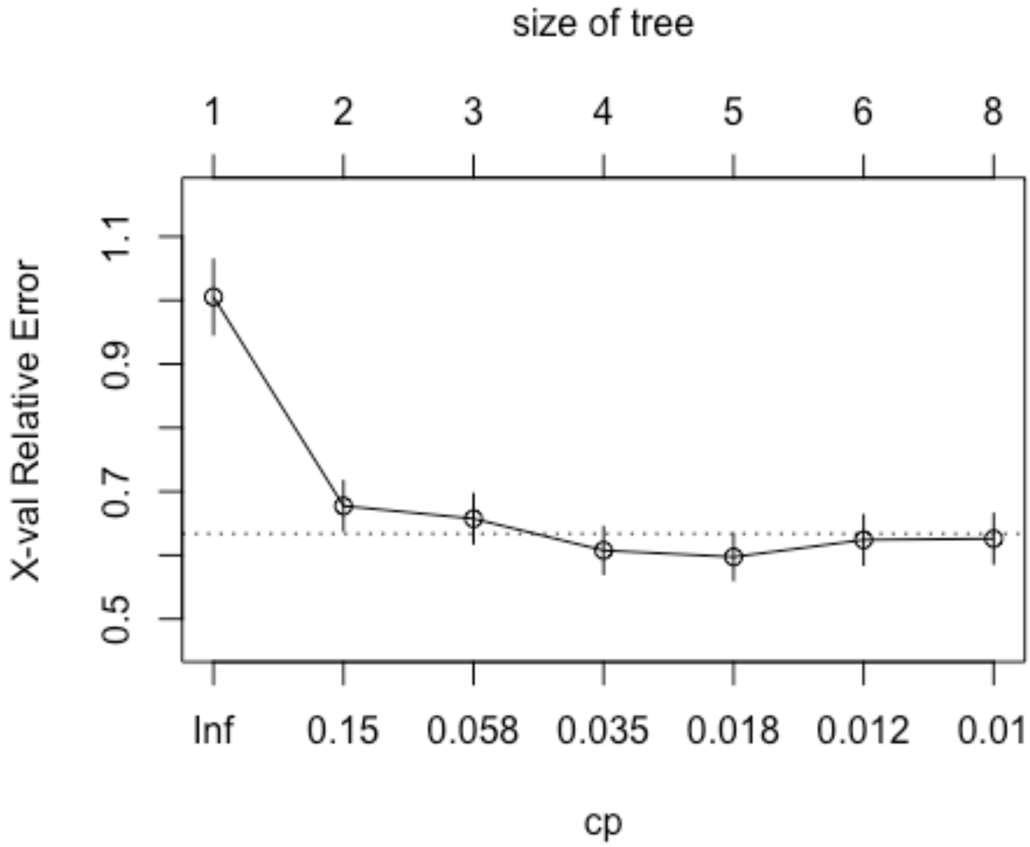
So the period "." in the model formula `cesd ~ .` part of the code above indicates that we're going to put `age`, `female`, `pss_fr`, `homeless`, `pcs`, and `mcs` into the model as predictors.

But the equivalent way to define this model where you list each variable you want in the model is to use the plus + symbol between each variable - so you could also write this code:

```
fitall <- rpart::rpart(cesd ~ age + female + pss_fr +
                              homeless + pcs + mcs,
                              data = h1a)
```

So, let's see what the regression tree for CESD looks like if we try all of these other variables as predictors in the model.

```
# fit a regression tree with all vars
fitall <- rpart::rpart(cesd ~ ., data = h1a)

# equivalent code statement without the shorthand
# using the period for the "rest of the variables"
# this time each variable to be included is listed
# individually putting a plus + in between each
# variable added to the model

fitall <- rpart::rpart(cesd ~ age + female + pss_fr +
                              homeless + pcs + mcs,
                              data = h1a)

# Now let's look at fitall
rpart::printcp(fitall) # Display the results

##
## Regression tree:
## rpart::rpart(formula = cesd ~ age + female + pss_fr + homeless +
##     pcs + mcs, data = h1a)
##
## Variables actually used in tree construction:
## [1] mcs pcs
##
## Root node error: 70788/453 = 156.27
##
## n= 453
##
##         CP nsplit rel error  xerror     xstd
## 1 0.340353      0   1.00000 1.00533 0.058864
## 2 0.063092      1   0.65965 0.67735 0.039151
## 3 0.053626      2   0.59655 0.65732 0.039052
## 4 0.022423      3   0.54293 0.60753 0.037040
## 5 0.013872      4   0.52051 0.59730 0.036529
## 6 0.010032      5   0.50663 0.62412 0.039232
## 7 0.010000      7   0.48657 0.62590 0.039228

rpart::plotcp(fitall) # Visualize cross-validation results
```

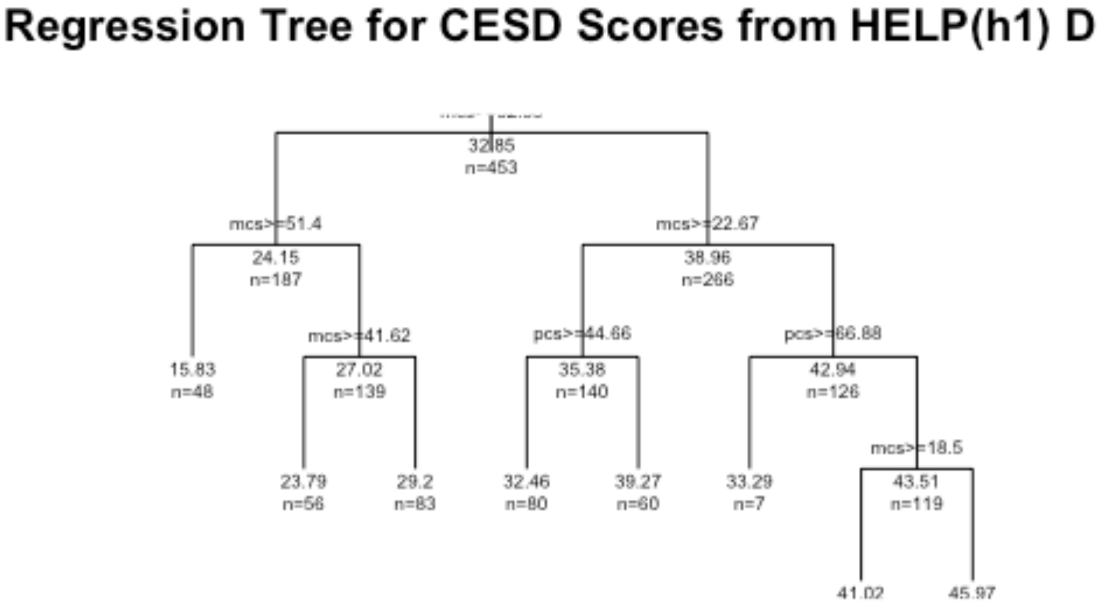


```
summary(fitall) # Detailed summary of fit
```

```
## Call:
## rpart::rpart(formula = cesd ~ age + female + pss_fr + homeless +
##     pcs + mcs, data = h1a)
##   n= 453
##
##           CP nsplit rel error    xerror       xstd
## 1 0.34035277      0 1.0000000 1.0053349 0.05886393
## 2 0.06309226      1 0.6596472 0.6773505 0.03915057
## 3 0.05362563      2 0.5965550 0.6573165 0.03905177
## 4 0.02242335      3 0.5429293 0.6075274 0.03703954
## 5 0.01387215      4 0.5205060 0.5973008 0.03652912
## 6 0.01003176      5 0.5066338 0.6241205 0.03923206
## 7 0.01000000      7 0.4865703 0.6258986 0.03922761
##
## Variable importance
##    mcs    pcs pss_fr    age female
##     78     13      5      3      1
##
## Node number 1: 453 observations,    complexity param=0.3403528
##   mean=32.84768, MSE=156.266
##   left son=2 (187 obs) right son=3 (266 obs)
##   Primary splits:
##       mcs    < 32.52559 to the right, improve=0.340352800, (0 missing)
##       pcs    < 49.19916 to the right, improve=0.104572600, (0 missing)
##       female < 0.5      to the left,  improve=0.032302950, (0 missing)
##       pss_fr < 8.5      to the right, improve=0.029240370, (0 missing)
##       age    < 23.5     to the right, improve=0.007589837, (0 missing)
##   Surrogate splits:
##       pcs    < 56.1551  to the right, agree=0.634, adj=0.112, (0 split)
##       pss_fr < 10.5     to the right, agree=0.609, adj=0.053, (0 split)
##       age    < 21.5     to the left,  agree=0.592, adj=0.011, (0 split)
##
## Node number 2: 187 observations,    complexity param=0.06309226
##   mean=24.14973, MSE=112.5979
##   left son=4 (48 obs) right son=5 (139 obs)
##   Primary splits:
##       mcs    < 51.3962  to the right, improve=0.21211280, (0 missing)
##       pcs    < 46.0814  to the right, improve=0.07616853, (0 missing)
##       pss_fr < 11.5     to the right, improve=0.03161969, (0 missing)
##       age    < 22.5     to the right, improve=0.02449595, (0 missing)
##       female < 0.5      to the left,  improve=0.01088789, (0 missing)
##   Surrogate splits:
##       pss_fr < 11.5     to the right, agree=0.765, adj=0.083, (0 split)
##       age    < 58.5     to the right, agree=0.754, adj=0.042, (0 split)
##
## Node number 3: 266 observations,    complexity param=0.05362563
##   mean=38.96241, MSE=96.38956
##   left son=6 (140 obs) right son=7 (126 obs)
##   Primary splits:
##       mcs    < 22.67163 to the right, improve=0.14805510, (0 missing)
##       pcs    < 40.92127 to the right, improve=0.07769934, (0 missing)
##       pss_fr < 0.5      to the right, improve=0.03572097, (0 missing)
##       female < 0.5      to the left,  improve=0.03455917, (0 missing)
##       age    < 48.5     to the left,  improve=0.01737694, (0 missing)
##   Surrogate splits:
##       pss_fr   < 3.5      to the right, agree=0.583, adj=0.119, (0 split)
##       pcs      < 64.93552 to the left,  agree=0.560, adj=0.071, (0 split)
##       female   < 0.5      to the left,  agree=0.556, adj=0.063, (0 split)
##       age      < 46.5     to the left,  agree=0.553, adj=0.056, (0 split)
##       homeless < 0.5      to the right, agree=0.530, adj=0.008, (0 split)
##
## Node number 4: 48 observations
##   mean=15.83333, MSE=128.0556
##
## Node number 5: 139 observations,    complexity param=0.01387215
##   mean=27.02158, MSE=75.12903
##   left son=10 (56 obs) right son=11 (83 obs)
##   Primary splits:
##       mcs      < 41.62456 to the right, improve=0.09403377, (0 missing)
##       pcs      < 26.8635  to the right, improve=0.07496568, (0 missing)
##       pss_fr   < 3.5      to the right, improve=0.02872252, (0 missing)
##       age      < 33.5     to the left,  improve=0.01948280, (0 missing)
##       homeless < 0.5      to the left,  improve=0.01404178, (0 missing)
##   Surrogate splits:
##       pcs    < 22.26483 to the left,  agree=0.619, adj=0.054, (0 split)
##       pss_fr < 13.5     to the right, agree=0.612, adj=0.036, (0 split)
## 
## Node number 6: 140 observations,    complexity param=0.02242335
##   mean=35.37857, MSE=80.77811 
##   left son=12 (80 obs) right son=13 (60 obs)
##   Primary splits:
##       pcs    < 44.6562  to the right, improve=0.140359400, (0 missing)
##       pss_fr < 8.5      to the right, improve=0.069217610, (0 missing)
##       age    < 38.5     to the left,  improve=0.044384950, (0 missing)
##       mcs    < 27.62416 to the right, improve=0.021316600, (0 missing)
##       female < 0.5      to the left,  improve=0.007874331, (0 missing)
##   Surrogate splits:
##       age      < 36.5     to the left,  agree=0.686, adj=0.267, (0 split)
##       mcs      < 23.7272  to the right, agree=0.621, adj=0.117, (0 split)
##       homeless < 0.5      to the left,  agree=0.593, adj=0.050, (0 split)
##       pss_fr   < 0.5      to the right, agree=0.579, adj=0.017, (0 split)
## 
## Node number 7: 126 observations,    complexity param=0.01003176
##   mean=42.94444, MSE=83.60802 
##   left son=14 (7 obs) right son=15 (119 obs)
##   Primary splits:
##       pcs    < 66.88379 to the right, improve=0.06563616, (0 missing)
##       mcs    < 18.49567 to the right, improve=0.05724195, (0 missing)
##       female < 0.5      to the left,  improve=0.05365277, (0 missing)
##       age    < 48.5     to the left,  improve=0.03224087, (0 missing)
##       pss_fr < 12.5     to the left,  improve=0.02139818, (0 missing)
## 
## Node number 10: 56 observations
##   mean=23.78571, MSE=71.52551 
## 
## Node number 11: 83 observations
##   mean=29.20482, MSE=65.72913 
## 
## Node number 12: 80 observations
##   mean=32.4625, MSE=73.84859 
## 
## Node number 13: 60 observations
##   mean=39.26667, MSE=63.56222 
## 
## Node number 14: 7 observations
##   mean=33.28571, MSE=117.9184 
## 
## Node number 15: 119 observations,    complexity param=0.01003176
##   mean=43.51261, MSE=75.77925 
##   left son=30 (59 obs) right son=31 (60 obs)
##   Primary splits:
```

```
##       mcs    < 18.49567 to the right, improve=0.08082019, (0 missing)
##       female < 0.5      to the left,  improve=0.04062765, (0 missing)
##       pcs    < 35.99184 to the right, improve=0.03933964, (0 missing)
##       age    < 48.5     to the left,  improve=0.03130734, (0 missing)
##       pss_fr < 12.5     to the left,  improve=0.02626137, (0 missing)
##   Surrogate splits:
##       pcs      < 46.51692 to the left,  agree=0.672, adj=0.339, (0 split)
##       pss_fr   < 8.5      to the left,  agree=0.622, adj=0.237, (0 split)
##       age      < 31.5     to the right, agree=0.613, adj=0.220, (0 split)
##       homeless < 0.5      to the right, agree=0.571, adj=0.136, (0 split)
##       female   < 0.5      to the left,  agree=0.521, adj=0.034, (0 split)
## 
## Node number 30: 59 observations
##   mean=41.01695, MSE=69.67768 
## 
## Node number 31: 60 observations
##   mean=45.96667, MSE=69.63222
```

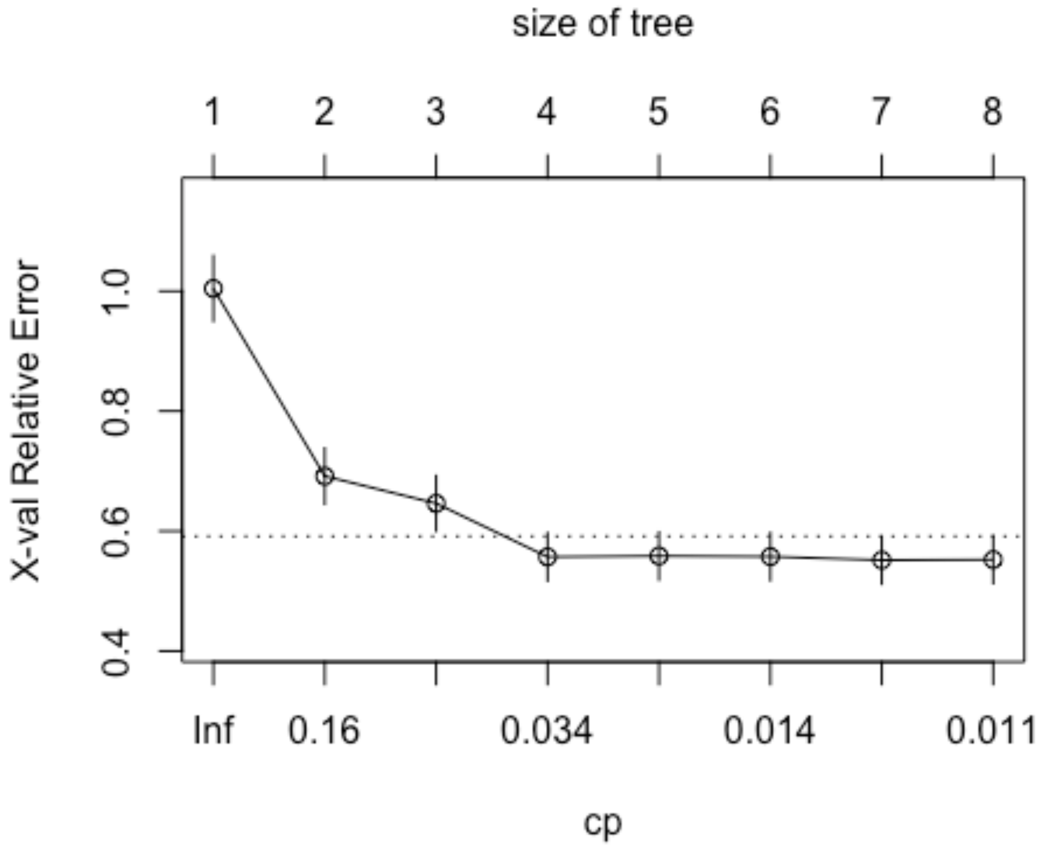
```
plot(fitall, uniform = TRUE, compress = FALSE, main = "Regression Tree for CE
SD Scores from HELP(h1) Data")
text(fitall, use.n = TRUE, all = TRUE, cex = 0.5)
```



PROBLEM 3: Regression Tree for MCS Using Rest of Variables

Using the code above as a guide, swap out `mcs` for `cesd` and redo the regression tree for `mcs` using the rest of the variables in the data subset `h1a`.

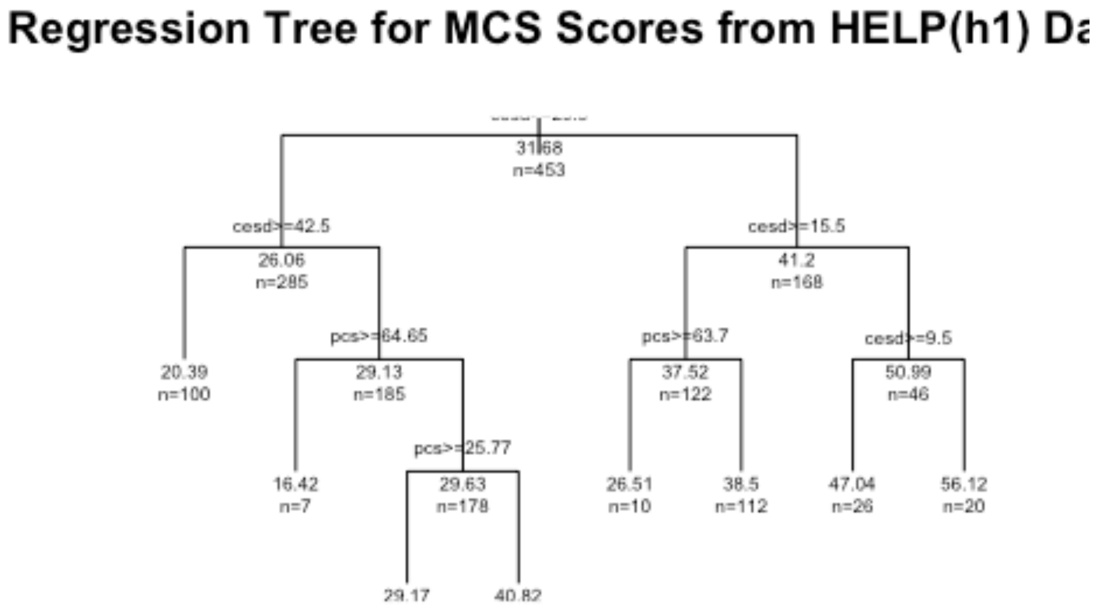
```
# fit a regression tree with all vars
fitall <- rpart::rpart(mcs ~ ., data = h1a)

# Now let's look at fitall
rpart::printcp(fitall) # Display the results

##
## Regression tree:
## rpart::rpart(formula = mcs ~ ., data = h1a)
##
## Variables actually used in tree construction:
## [1] cesd pcs
##
## Root node error: 74512/453 = 164.48
##
## n= 453
##
##         CP nsplit rel error  xerror     xstd
## 1 0.325298      0   1.00000 1.00443 0.054665
## 2 0.081349      1   0.67470 0.69157 0.046764
## 3 0.066496      2   0.59335 0.64611 0.046155
## 4 0.017717      3   0.52686 0.55696 0.040193
## 5 0.015767      4   0.50914 0.55844 0.039865
## 6 0.012496      5   0.49337 0.55739 0.039977
## 7 0.012258      6   0.48088 0.55155 0.039760
## 8 0.010000      7   0.46862 0.55239 0.039745

rpart::plotcp(fitall) # Visualize cross-validation results
```



```
summary(fitall) # Detailed summary of fit

## Call:
## rpart::rpart(formula = mcs ~ ., data = h1a)
##   n= 453
##
##           CP nsplit rel error    xerror       xstd
## 1 0.32529813      0 1.0000000 1.0044325 0.05466545
## 2 0.08134904      1 0.6747019 0.6915654 0.04676416
## 3 0.06649553      2 0.5933528 0.6461123 0.04615506
## 4 0.01771736      3 0.5268573 0.5569624 0.04019269
## 5 0.01576737      4 0.5091399 0.5584427 0.03986457
## 6 0.01249609      5 0.4933726 0.5573873 0.03997658
## 7 0.01225792      6 0.4808765 0.5515523 0.03975960
## 8 0.01000000      7 0.4686186 0.5523851 0.03974499
##
## Variable importance
##   cesd    pcs    age pss_fr
##     83     14      1      1
##
## Node number 1: 453 observations,    complexity param=0.3252981
##   mean=31.67668, MSE=164.4847
##   left son=2 (285 obs) right son=3 (168 obs)
##   Primary splits:
##       cesd   < 29.5     to the right, improve=0.325298100, (0 missing)
##       pcs    < 49.46132 to the left,  improve=0.064711670, (0 missing)
##       pss_fr < 10.5     to the left,  improve=0.039318510, (0 missing)
##       female < 0.5      to the right, improve=0.014091560, (0 missing)
##       age    < 42.5     to the left,  improve=0.005473724, (0 missing)
##   Surrogate splits:
##       pcs < 56.34591 to the left,  agree=0.669, adj=0.107, (0 split)
##       age < 57.5     to the left,  agree=0.631, adj=0.006, (0 split)
##
## Node number 2: 285 observations,    complexity param=0.06649553
##   mean=26.06057, MSE=100.1894
##   left son=4 (100 obs) right son=5 (185 obs)
##   Primary splits:
##       cesd   < 42.5     to the right, improve=0.173520000, (0 missing)
##       pcs    < 24.47511 to the right, improve=0.057879990, (0 missing)
##       pss_fr < 10.5     to the left,  improve=0.015219690, (0 missing)
##       age    < 22.5     to the right, improve=0.005742931, (0 missing)
##       female < 0.5      to the right, improve=0.001903900, (0 missing)
##   Surrogate splits:
##       pss_fr < 0.5      to the left,  agree=0.660, adj=0.03, (0 split)
##       pcs    < 68.64778 to the right, agree=0.653, adj=0.01, (0 split)
##
## Node number 3: 168 observations,    complexity param=0.08134904
##   mean=41.20401, MSE=129.2805
##   left son=6 (122 obs) right son=7 (46 obs)
##   Primary splits:
##       cesd   < 15.5     to the right, improve=0.279083400, (0 missing)
##       pcs    < 62.7532  to the right, improve=0.113215200, (0 missing)
##       pss_fr < 10.5     to the left,  improve=0.053187210, (0 missing)
##       age    < 48.5     to the left,  improve=0.036737610, (0 missing)
##       female < 0.5      to the right, improve=0.007177787, (0 missing)
##   Surrogate splits:
##       age < 58.5     to the left,  agree=0.738, adj=0.043, (0 split)
##
## Node number 4: 100 observations
##   mean=20.38941, MSE=43.95751
##
## Node number 5: 185 observations,    complexity param=0.01576737
##   mean=29.12606, MSE=103.8029
##   left son=10 (7 obs) right son=11 (178 obs)
##   Primary splits:
##       pcs    < 64.65134 to the right, improve=0.061178900, (0 missing)
##       age    < 22.5     to the right, improve=0.031248410, (0 missing)
##       cesd   < 37.5     to the right, improve=0.020833690, (0 missing)
##       pss_fr < 10.5     to the left,  improve=0.015175680, (0 missing)
##       female < 0.5      to the left,  improve=0.004355548, (0 missing)
##
## Node number 6: 122 observations,    complexity param=0.01771736
##   mean=37.51566, MSE=103.6988
##   left son=12 (10 obs) right son=13 (112 obs)
##   Primary splits:
##       pcs    < 63.69606 to the right, improve=0.10434930, (0 missing)
##       age    < 47.5     to the left,  improve=0.02626159, (0 missing)
##       cesd   < 24.5     to the right, improve=0.02348926, (0 missing)
##       female < 0.5      to the right, improve=0.02256241, (0 missing)
##       pss_fr < 2.5      to the right, improve=0.01295167, (0 missing)
##
## Node number 7: 46 observations,    complexity param=0.01249609
##   mean=50.98616, MSE=65.35702
##   left son=14 (26 obs) right son=15 (20 obs)
##   Primary splits:
##       cesd     < 9.5      to the right, improve=0.30970460, (0 missing)
##       pcs      < 59.57495 to the right, improve=0.16249370, (0 missing)
##       pss_fr   < 11.5     to the left,  improve=0.13099300, (0 missing)
##       age      < 40       to the left,  improve=0.06604375, (0 missing)
##       homeless < 0.5      to the left,  improve=0.00873942, (0 missing)
##   Surrogate splits:
##       pss_fr   < 11.5     to the left,  agree=0.674, adj=0.25, (0 split)
##       pcs      < 54.5861  to the left,  agree=0.652, adj=0.20, (0 split)
##       age      < 46       to the left,  agree=0.609, adj=0.10, (0 split)
##       homeless < 0.5      to the left,  agree=0.609, adj=0.10, (0 split)
##
## Node number 10: 7 observations
##   mean=16.41837, MSE=35.31025
##
## Node number 11: 178 observations,    complexity param=0.01225792
##   mean=29.6258, MSE=99.89614
##   left son=22 (171 obs) right son=23 (7 obs)
##   Primary splits:
##       pcs      < 25.77119 to the right, improve=0.051365510, (0 missing)
##       age      < 22.5     to the right, improve=0.029936490, (0 missing)
##       pss_fr   < 10.5     to the left,  improve=0.022699840, (0 missing)
##       cesd     < 37.5     to the right, improve=0.020642200, (0 missing)
##       homeless < 0.5      to the right, improve=0.002448012, (0 missing)
##
## Node number 12: 10 observations
##   mean=26.50685, MSE=30.97799
##
## Node number 13: 112 observations
##   mean=38.49859, MSE=98.40465
##
## Node number 14: 26 observations
##   mean=47.04024, MSE=67.29195
##
## Node number 15: 20 observations
##   mean=56.11586, MSE=16.28645
##
## Node number 22: 171 observations
##   mean=29.16748, MSE=95.51594
```

```
##
## Node number 23: 7 observations
##   mean=40.8217, MSE=76.41866
```

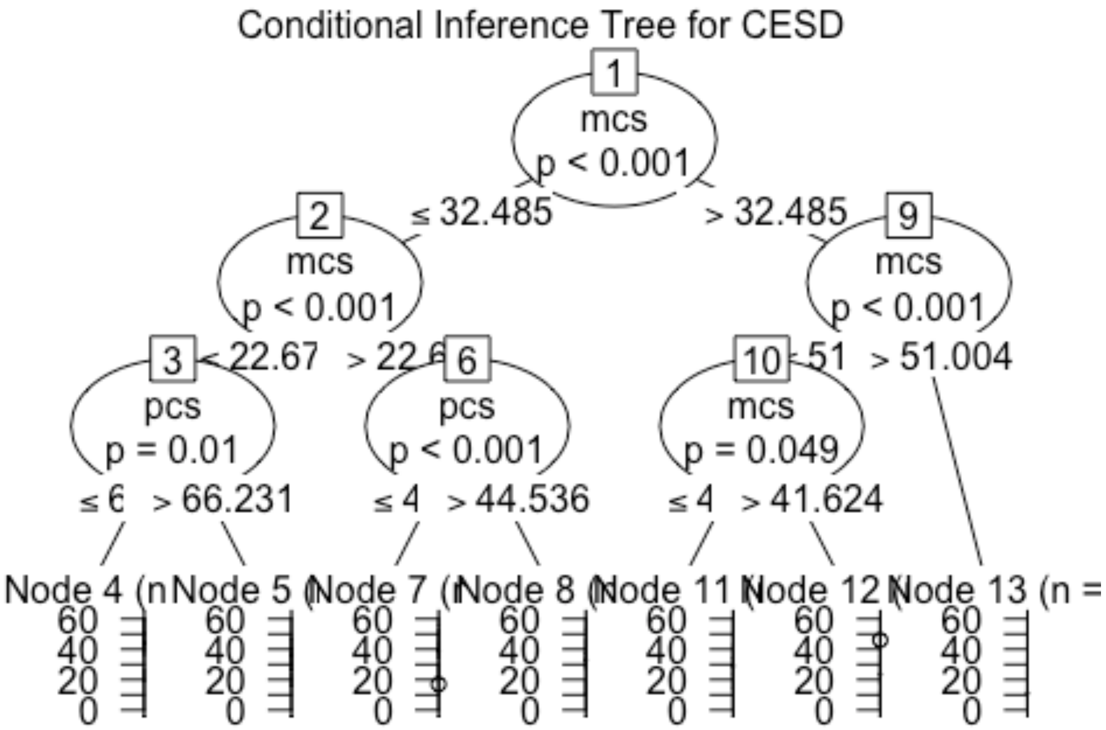
```
plot(fitall, uniform = TRUE, compress = FALSE, main = "Regression Tree for MC
S Scores from HELP(h1) Data")
text(fitall, use.n = TRUE, all = TRUE, cex = 0.5)
```



Regression Tree for CESD Using the `party` package approach

The `party` package has better graphics and fits a "conditional" regression tree using the `ctree()` function. Here is the model approach for the `cesd` using the rest of the variables in the dataset `h1a`.

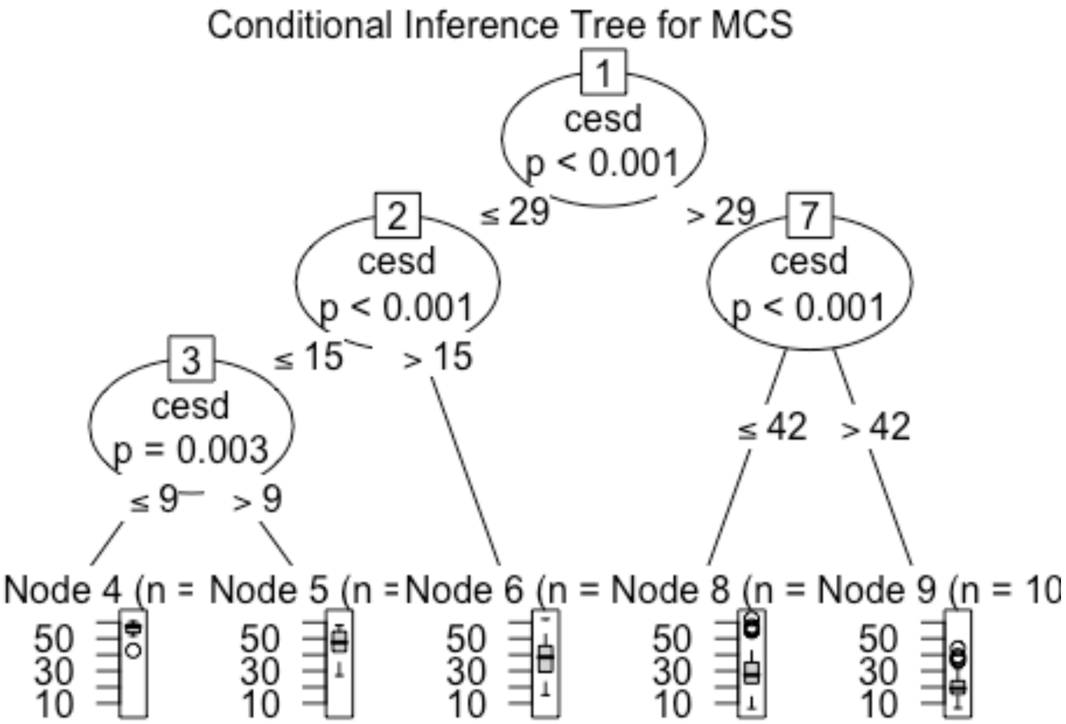
```
fitallp <- party::ctree(cesd ~ ., data = h1a)
plot(fitallp, main = "Conditional Inference Tree for CESD")
```



PROBLEM 4: Fit a Conditional Regression Tree for MCS

Using the code above, swap out `mcs` for `cesd` to fit a confitional regression tree for `mcs` predicted by the other variables in the dataset `h1a`.

```
fitallp <- party::ctree(mcs ~ ., data = h1a)
plot(fitallp, main = "Conditional Inference Tree for MCS")
```



Logistic Regression of CESD => 16

When the outcome is dichotomous or is a categorical outcome, you can fit a "decision tree" or "classification tree". One way you've already learned last week is fitting a logistic regression model. In fact, logistic regression is a supervised classification modeling approach. Let's see what this looks like for predicting depression (indicated by `cesd_gte16` for people with CESD scores => 16). Pay attention to which variables are significant in the resulting logistic regression model.

```
# begin with a logistic regression - depressed or not
glm1 <- glm(cesd_gte16 ~ age + female + pss_fr + homeless +
              pcs + mcs, data = h1)
summary(glm1)

##
## Call:
## glm(formula = cesd_gte16 ~ age + female + pss_fr + homeless +
##     pcs + mcs, data = h1)
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -0.90801  -0.06647   0.02642   0.14484   0.51900
##
```

```
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  1.5529621  0.0948791  16.368  < 2e-16 ***
## age         -0.0018844  0.0016377  -1.151 0.250517    
## female      -0.0309994  0.0294192  -1.054 0.292584    
## pss_fr      -0.0034492  0.0031460  -1.096 0.273509    
## homeless    -0.0045879  0.0250875  -0.183 0.854978    
## pcs         -0.0039722  0.0011870  -3.346 0.000888 ***
## mcs         -0.0114878  0.0009709 -11.832  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for gaussian family taken to be 0.0668409)
## 
##     Null deviance: 41.329  on 452  degrees of freedom
## Residual deviance: 29.811  on 446  degrees of freedom
## AIC: 68.939
## 
## Number of Fisher Scoring iterations: 2
```

PROBLEM 5: Fit a Logistic Regression Model for MCS < 45

The mental component (or composite) scale of the SF36 instrument is a measure of mental health. The scores are created relative to population norms. The population norm for the `mcs` of the SF36 is 50 with a standard deviation of 10. A difference of a "half" of a standard deviation - in other words a difference of 5 points - is considered to be clinically meaningful. So, people with MCS scores greater than 55 are considered to have better than average mental health and those with MCS scores less than 45 are considered to have worse than average mental health scores. So, in the dataset `h1` above, we included an indicator variable called `mcs_lt45` where a value of 1 indicates people with MCS < 45 ("poor mental health") and a value of 0 ("normal or better than normal mental health") is for people with MCS scores => 45.

Use the dataset `h1` and the code above to fit a logistic regression model for `mcs_lt45` based on the predictors of

- `age`
- `female`
- `pss_fr`
- `homeless`
- `pcs`
- `cesd`

Is this model similar to the model for `cesd_gte16` or not - what is similar? what is different?

```
glm1_mcs <- glm(mcs_lt45 ~ age + female + pss_fr + homeless +
              pcs + cesd, data = h1)
summary(glm1_mcs)
```

```
## 
## Call:
## glm(formula = mcs_lt45 ~ age + female + pss_fr + homeless + pcs + 
##     cesd, data = h1)
## 
## Deviance Residuals: 
##      Min        1Q    Median        3Q       Max  
## -0.96035  -0.10332   0.08078   0.21806   0.62498  
## 
## Coefficients:
##               Estimate Std. Error t value Pr(>|t|)    
## (Intercept)  0.3611168  0.1386939   2.604  0.00953 ** 
## age         -0.0023080  0.0021130  -1.092  0.27529    
## female       0.0202380  0.0382212   0.529  0.59672    
## pss_fr      -0.0036606  0.0040882  -0.895  0.37104    
## homeless     0.0172706  0.0323939   0.533  0.59420    
## pcs          0.0005446  0.0015809   0.344  0.73064    
## cesd         0.0158725  0.0013519  11.741  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for gaussian family taken to be 0.1114291)
## 
##     Null deviance: 68.424  on 452  degrees of freedom
## Residual deviance: 49.697  on 446  degrees of freedom
## AIC: 300.46
## 
## Number of Fisher Scoring iterations: 2
```

Fit a Classification Tree for CESD => 16

We can use the rpart package again to fit a classification tree to the depression indicator cesd_gte16.

```
fitk <- rpart::rpart(cesd_gte16 ~ age + female + pss_fr +
                       homeless + pcs + mcs,
                     method = "class", data = h1)
class(fitk)
```

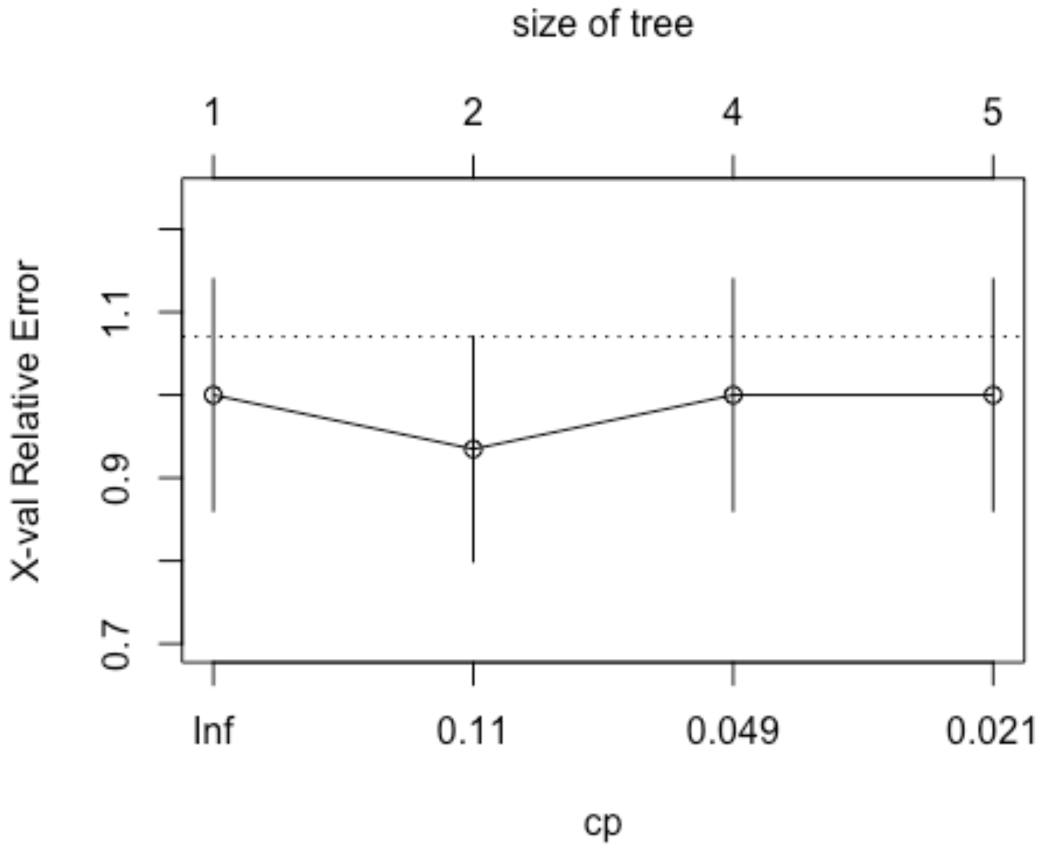
```
## [1] "rpart"
```

```
# Display the results
rpart::printcp(fitk)
```

```
## 
## Classification tree:
## rpart::rpart(formula = cesd_gte16 ~ age + female + pss_fr + homeless + 
##     pcs + mcs, data = h1, method = "class")
## 
## Variables actually used in tree construction:
## [1] age mcs pcs
```

```
##
## Root node error: 46/453 = 0.10155
##
## n= 453
##
##         CP nsplit rel error  xerror    xstd
## 1 0.239130      0   1.00000 1.00000 0.13976
## 2 0.054348      1   0.76087 0.93478 0.13562
## 3 0.043478      3   0.65217 1.00000 0.13976
## 4 0.010000      4   0.60870 1.00000 0.13976
```

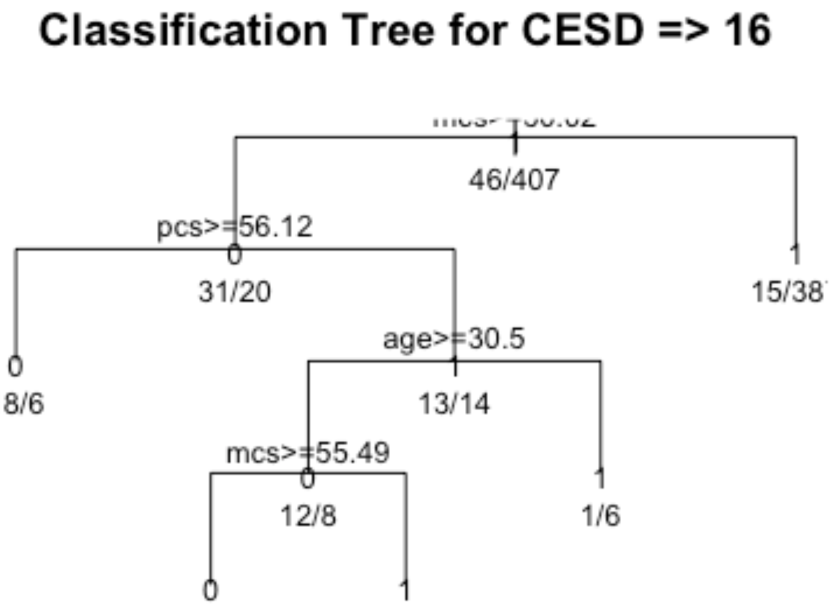
```
#Visualize the cross-validation results
rpart::plotcp(fitk)
```



```
# Get a detailed summary of the splits
summary(fitk)
```

```
## Call:
## rpart::rpart(formula = cesd_gte16 ~ age + female + pss_fr + homeless +
##     pcs + mcs, data = h1, method = "class")
##   n= 453
##
##           CP nsplit rel error    xerror      xstd
## 1 0.23913043      0 1.0000000 1.0000000 0.1397556
## 2 0.05434783      1 0.7608696 0.9347826 0.1356186
## 3 0.04347826      3 0.6521739 1.0000000 0.1397556
## 4 0.01000000      4 0.6086957 1.0000000 0.1397556
## 
## Variable importance
##      mcs      age      pcs   pss_fr homeless 
##       84        8        5        1        1 
## 
## Node number 1: 453 observations,    complexity param=0.2391304
##   predicted class=1  expected loss=0.1015453  P(node) =1
##     class counts:    46   407
##    probabilities: 0.102 0.898 
##   left son=2 (51 obs) right son=3 (402 obs)
##   Primary splits:
##       mcs      < 50.02446 to the right, improve=29.4635100, (0 missing)
##       pcs      < 49.19916 to the right, improve= 4.2774340, (0 missing)
##       pss_fr   < 10.5     to the right, improve= 3.6879600, (0 missing)
##       age      < 25.5     to the right, improve= 0.7580753, (0 missing)
##       homeless < 0.5      to the left,  improve= 0.1845446, (0 missing)
##   Surrogate splits:
##       age < 58.5     to the right, agree=0.89, adj=0.02, (0 split)
## 
## Node number 2: 51 observations,    complexity param=0.05434783
##   predicted class=0  expected loss=0.3921569  P(node) =0.1125828
##     class counts:    31    20
##    probabilities: 0.608 0.392 
##   left son=4 (24 obs) right son=5 (27 obs)
##   Primary splits:
##       pcs      < 56.1216  to the right, improve=1.83224400, (0 missing)
##       age      < 28.5     to the right, improve=1.68385500, (0 missing)
##       mcs      < 52.79105 to the right, improve=0.65918000, (0 missing)
##       homeless < 0.5      to the right, improve=0.11695130, (0 missing)
##       pss_fr   < 8.5      to the right, improve=0.09467787, (0 missing)
##   Surrogate splits:
##       mcs      < 54.23909 to the left,  agree=0.647, adj=0.250, (0 split)
##       homeless < 0.5      to the left,  agree=0.588, adj=0.125, (0 split)
##       age      < 37.5     to the left,  agree=0.569, adj=0.083, (0 split)
##       female   < 0.5      to the left,  agree=0.569, adj=0.083, (0 split)
##       pss_fr   < 12.5     to the right, agree=0.569, adj=0.083, (0 split)
## 
## Node number 3: 402 observations
##   predicted class=1  expected loss=0.03731343  P(node) =0.8874172
##     class counts:    15   387
##    probabilities: 0.037 0.963 
## 
## Node number 4: 24 observations
##   predicted class=0  expected loss=0.25  P(node) =0.05298013
##     class counts:    18     6
##    probabilities: 0.750 0.250 
## 
```

```
## Node number 5: 27 observations,    complexity param=0.05434783
##   predicted class=1  expected loss=0.4814815  P(node) =0.05960265
##     class counts:    13    14
##    probabilities: 0.481 0.519 
##   left son=10 (20 obs) right son=11 (7 obs)
##   Primary splits:
##       age    < 30.5     to the right, improve=2.16719600, (0 missing)
##       mcs    < 54.81272 to the right, improve=1.81481500, (0 missing)
##       pcs    < 53.12609 to the left,  improve=1.21832400, (0 missing)
##       pss_fr < 4.5      to the right, improve=0.72433860, (0 missing)
##       female < 0.5      to the left,  improve=0.05291005, (0 missing)
## 
## Node number 10: 20 observations,    complexity param=0.04347826
##   predicted class=0  expected loss=0.4  P(node) =0.04415011
##     class counts:    12     8
##    probabilities: 0.600 0.400 
##   left son=20 (12 obs) right son=21 (8 obs)
##   Primary splits:
##       mcs      < 55.49419 to the right, improve=1.35000000, (0 missing)
##       pcs      < 50.26239 to the right, improve=0.40000000, (0 missing)
##       pss_fr   < 6        to the right, improve=0.26666670, (0 missing)
##       homeless < 0.5      to the right, improve=0.14545450, (0 missing)
##       age      < 34.5     to the right, improve=0.01758242, (0 missing)
##   Surrogate splits:
##       pss_fr   < 4.5      to the right, agree=0.70, adj=0.250, (0 split)
##       age      < 44       to the left,  agree=0.65, adj=0.125, (0 split)
##       homeless < 0.5      to the left,  agree=0.65, adj=0.125, (0 split)
##       pcs      < 50.20288 to the left,  agree=0.65, adj=0.125, (0 split)
## 
## Node number 11: 7 observations
##   predicted class=1  expected loss=0.1428571  P(node) =0.01545254
##     class counts:     1     6
##    probabilities: 0.143 0.857 
## 
## Node number 20: 12 observations
##   predicted class=0  expected loss=0.25  P(node) =0.02649007
##     class counts:     9     3
##    probabilities: 0.750 0.250 
## 
## Node number 21: 8 observations
##   predicted class=1  expected loss=0.375  P(node) =0.01766004
##     class counts:     3     5
##    probabilities: 0.375 0.625

# Plot the tree
plot(fitk, uniform = TRUE,
     main = "Classification Tree for CESD => 16")
text(fitk, use.n = TRUE, all = TRUE, cex = 0.8)
```



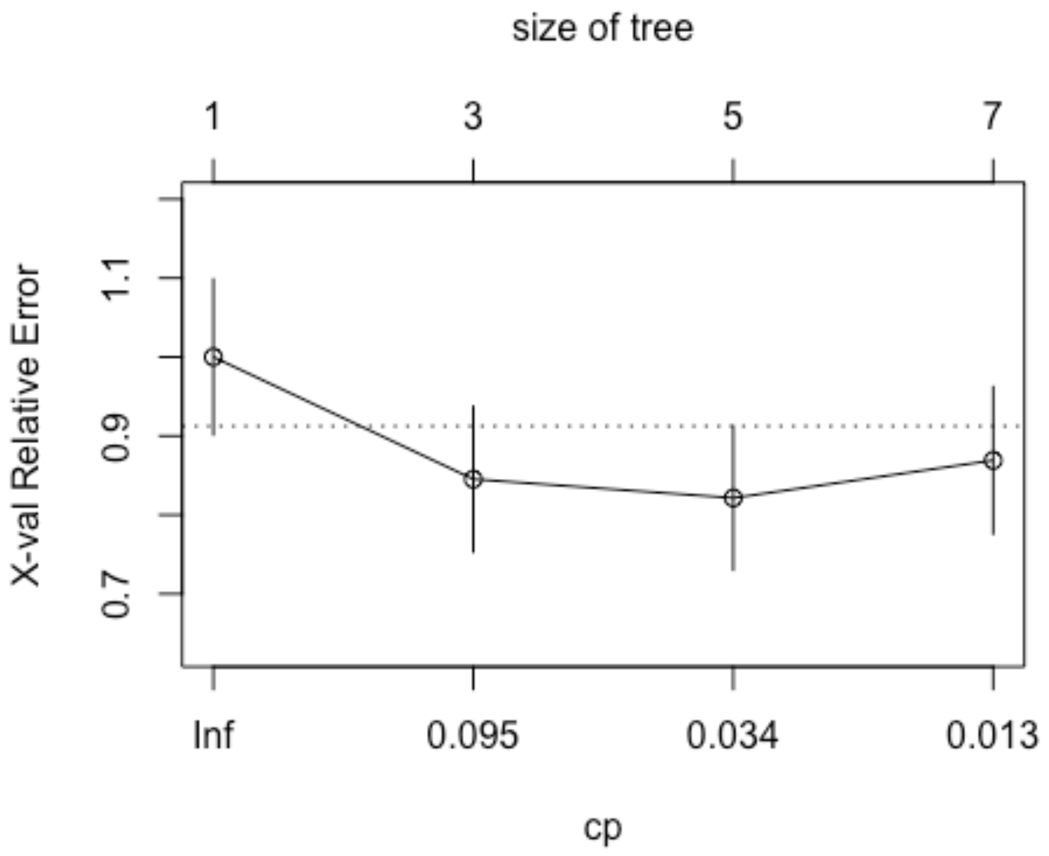
PROBLEM 6: Fit a Classification Tree for MCS < 45

Use the `rpart` package to fit a classification tree to the poor mental health indicator `mcs_lt45`.

```
fitk_mcs <- rpart::rpart(mcs_lt45 ~ age + female + pss_fr +
                           homeless + pcs + cesd,
                         method = "class", data = h1)
class(fitk_mcs)

## [1] "rpart"

# Display the results
rpart::printcp(fitk_mcs)

##
## Classification tree:
## rpart::rpart(formula = mcs_lt45 ~ age + female + pss_fr + homeless +
##     pcs + cesd, data = h1, method = "class")
##
## Variables actually used in tree construction:
## [1] age  cesd pcs
##
## Root node error: 84/453 = 0.18543
```

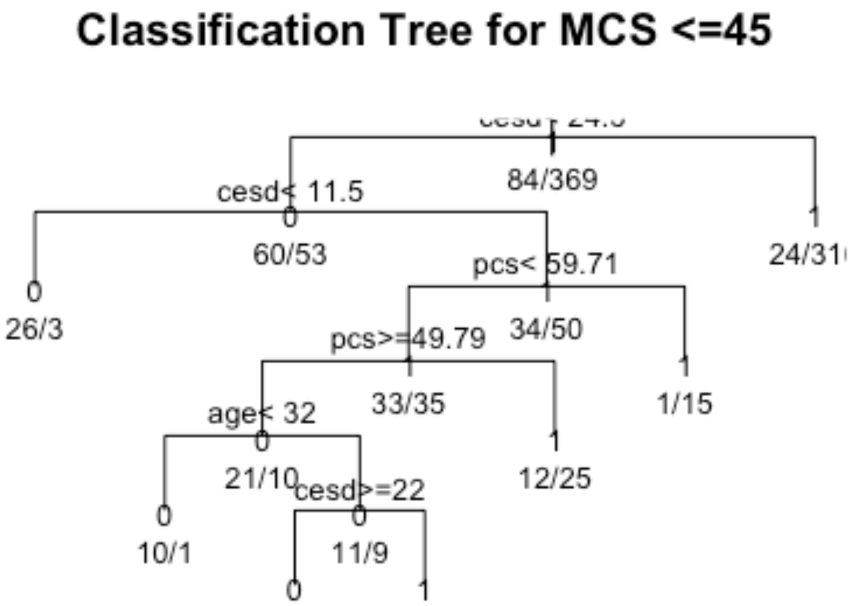
```
##
## n= 453
##
##          CP nsplit rel error  xerror     xstd
## 1 0.136905      0   1.00000 1.00000 0.098475
## 2 0.065476      2   0.72619 0.84524 0.092115
## 3 0.017857      4   0.59524 0.82143 0.091046
## 4 0.010000      6   0.55952 0.86905 0.093159
```

```
#Visualize the cross-validation results
rpart::plotcp(fitk_mcs)
```



```
# Get a detailed summary of the splits
summary(fitk_mcs)
```

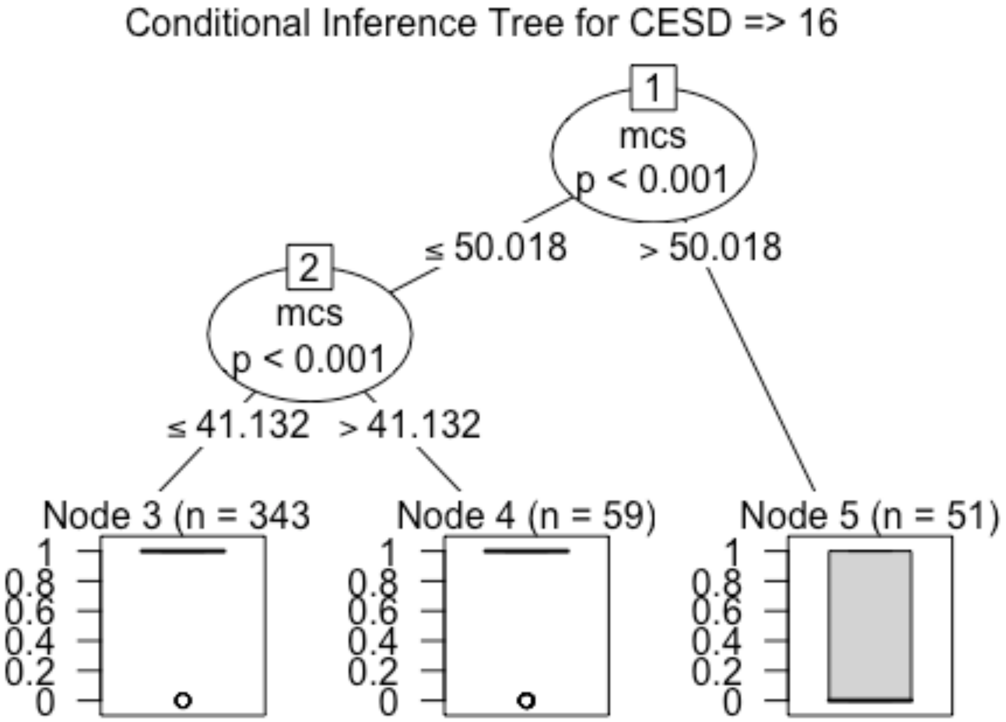
```
## Call:
## rpart::rpart(formula = mcs_lt45 ~ age + female + pss_fr + homeless +
##     pcs + cesd, data = h1, method = "class")
##   n= 453
##
##           CP nsplit rel error    xerror       xstd
## 1 0.13690476      0 1.0000000 1.0000000 0.09847465
## 2 0.06547619      2 0.7261905 0.8452381 0.09211545
## 3 0.01785714      4 0.5952381 0.8214286 0.09104619
## 4 0.01000000      6 0.5595238 0.8690476 0.09315900
## 
## Variable importance
##   cesd    pcs    age pss_fr 
##     76     16      6      2 
## 
## Node number 1: 453 observations,    complexity param=0.1369048
##   predicted class=1  expected loss=0.1854305  P(node) =1
##     class counts:    84   369
##    probabilities: 0.185 0.815 
##   left son=2 (113 obs) right son=3 (340 obs)
##   Primary splits:
##       cesd   < 24.5     to the left,  improve=35.952730, (0 missing)
##       pcs    < 49.46132 to the right, improve= 7.907014, (0 missing)
##       pss_fr < 10.5     to the right, improve= 4.386206, (0 missing)
##       female < 0.5      to the left,  improve= 1.504589, (0 missing)
##       age    < 48.5     to the right, improve= 1.425056, (0 missing)
##   Surrogate splits:
##       age < 57.5     to the right, agree=0.753, adj=0.009, (0 split)
##       pcs < 70.77019 to the right, agree=0.753, adj=0.009, (0 split)
## 
## Node number 2: 113 observations,    complexity param=0.1369048
##   predicted class=0  expected loss=0.4690265  P(node) =0.2494481
##     class counts:    60    53
##    probabilities: 0.531 0.469 
##   left son=4 (29 obs) right son=5 (84 obs)
##   Primary splits:
##       cesd   < 11.5     to the left,  improve=10.427690, (0 missing)
##       pcs    < 60.7539  to the left,  improve= 8.921666, (0 missing)
##       pss_fr < 11.5     to the right, improve= 2.105364, (0 missing)
##       female < 0.5      to the left,  improve= 1.591788, (0 missing)
##       age    < 47.5     to the right, improve= 1.587768, (0 missing)
##   Surrogate splits:
##       age < 58.5     to the right, agree=0.761, adj=0.069, (0 split)
## 
## Node number 3: 340 observations
##   predicted class=1  expected loss=0.07058824  P(node) =0.7505519
##     class counts:    24   316
##    probabilities: 0.071 0.929 
## 
## Node number 4: 29 observations
##   predicted class=0  expected loss=0.1034483  P(node) =0.06401766
##     class counts:    26     3
##    probabilities: 0.897 0.103 
## 
## Node number 5: 84 observations,    complexity param=0.06547619
##   predicted class=1  expected loss=0.4047619  P(node) =0.1854305
##     class counts:    34    50
##    probabilities: 0.405 0.595 
##   left son=10 (68 obs) right son=11 (16 obs)
##   Primary splits:
##       pcs    < 59.71077 to the left,  improve=4.6306020, (0 missing)
##       female < 0.5      to the left,  improve=1.8658960, (0 missing)
##       cesd   < 21.5     to the right, improve=1.7155130, (0 missing)
##       age    < 38.5     to the left,  improve=0.2586838, (0 missing)
##       pss_fr < 11.5     to the right, improve=0.2539683, (0 missing)
##
## Node number 10: 68 observations,    complexity param=0.06547619
##   predicted class=1  expected loss=0.4852941  P(node) =0.1501104
##     class counts:    33    35
##    probabilities: 0.485 0.515
##   left son=20 (31 obs) right son=21 (37 obs)
##   Primary splits:
##       pcs    < 49.7901  to the right, improve=4.2059850, (0 missing)
##       female < 0.5      to the left,  improve=2.0824760, (0 missing)
##       cesd   < 16.5     to the left,  improve=1.1284830, (0 missing)
##       age    < 43.5     to the left,  improve=0.4790628, (0 missing)
##       pss_fr < 7.5      to the left,  improve=0.2761438, (0 missing)
##   Surrogate splits:
##       age      < 27.5     to the left,  agree=0.588, adj=0.097, (0 split)
##       pss_fr   < 13.5     to the right, agree=0.588, adj=0.097, (0 split)
##       homeless < 0.5      to the right, agree=0.559, adj=0.032, (0 split)
##       cesd     < 12.5     to the left,  agree=0.559, adj=0.032, (0 split)
##
## Node number 11: 16 observations
##   predicted class=1  expected loss=0.0625  P(node) =0.03532009
##     class counts:     1    15
##    probabilities: 0.062 0.938
##
## Node number 20: 31 observations,    complexity param=0.01785714
##   predicted class=0  expected loss=0.3225806  P(node) =0.06843267
##     class counts:    21    10
##    probabilities: 0.677 0.323
##   left son=40 (11 obs) right son=41 (20 obs)
##   Primary splits:
##       age      < 32       to the left,  improve=1.830205000, (0 missing)
##       pss_fr   < 8.5      to the left,  improve=1.607211000, (0 missing)
##       cesd     < 21.5     to the right, improve=1.462673000, (0 missing)
##       pcs      < 57.31713 to the right, improve=1.274883000, (0 missing)
##       homeless < 0.5      to the left,  improve=0.004527448, (0 missing)
##   Surrogate splits:
##       pcs  < 59.00035 to the right, agree=0.710, adj=0.182, (0 split)
##       cesd < 16.5     to the left,  agree=0.677, adj=0.091, (0 split)
##
## Node number 21: 37 observations
##   predicted class=1  expected loss=0.3243243  P(node) =0.0816777
##     class counts:    12    25
##    probabilities: 0.324 0.676
##
## Node number 40: 11 observations
##   predicted class=0  expected loss=0.09090909  P(node) =0.02428256
##     class counts:    10     1
##    probabilities: 0.909 0.091 
## 
## Node number 41: 20 observations,    complexity param=0.01785714
##   predicted class=0  expected loss=0.45  P(node) =0.04415011
##     class counts:    11     9
##    probabilities: 0.550 0.450 
##   left son=82 (7 obs) right son=83 (13 obs)
##   Primary splits:
##       cesd     < 22       to the right, improve=2.0318680, (0 missing)
##       age      < 39.5     to the right, improve=0.5813187, (0 missing)
##       pcs      < 57.14784 to the right, improve=0.5813187, (0 missing)
##       pss_fr   < 7        to the left,  improve=0.4454545, (0 missing)
##       homeless < 0.5      to the right, improve=0.1000000, (0 missing)
##   Surrogate splits:
##       pss_fr < 7        to the left,  agree=0.80, adj=0.429, (0 split)
##       age    < 45       to the right, agree=0.75, adj=0.286, (0 split)
##       pcs    < 57.34769 to the right, agree=0.75, adj=0.286, (0 split)
## 
## Node number 82: 7 observations
##   predicted class=0  expected loss=0.1428571  P(node) =0.01545254
##     class counts:     6     1
##    probabilities: 0.857 0.143 
## 
## Node number 83: 13 observations
##   predicted class=1  expected loss=0.3846154  P(node) =0.02869757
##     class counts:     5     8
##    probabilities: 0.385 0.615

# Plot the tree
plot(fitk_mcs, uniform = TRUE,
     main = "Classification Tree for MCS <=45")
text(fitk_mcs, use.n = TRUE, all = TRUE, cex = 0.8)
```



Fit a Conditional Classification Tree for CESD => 16

Using the `party` package, we can fit a conditional classification tree using the `ctree()` function. Let's do one for the indicator of depression `cesd_gte16` given the other variables in the `h1` dataset: `age`, `female`, `pss_fr`, `homeless`, `pcs`, `mcs`.

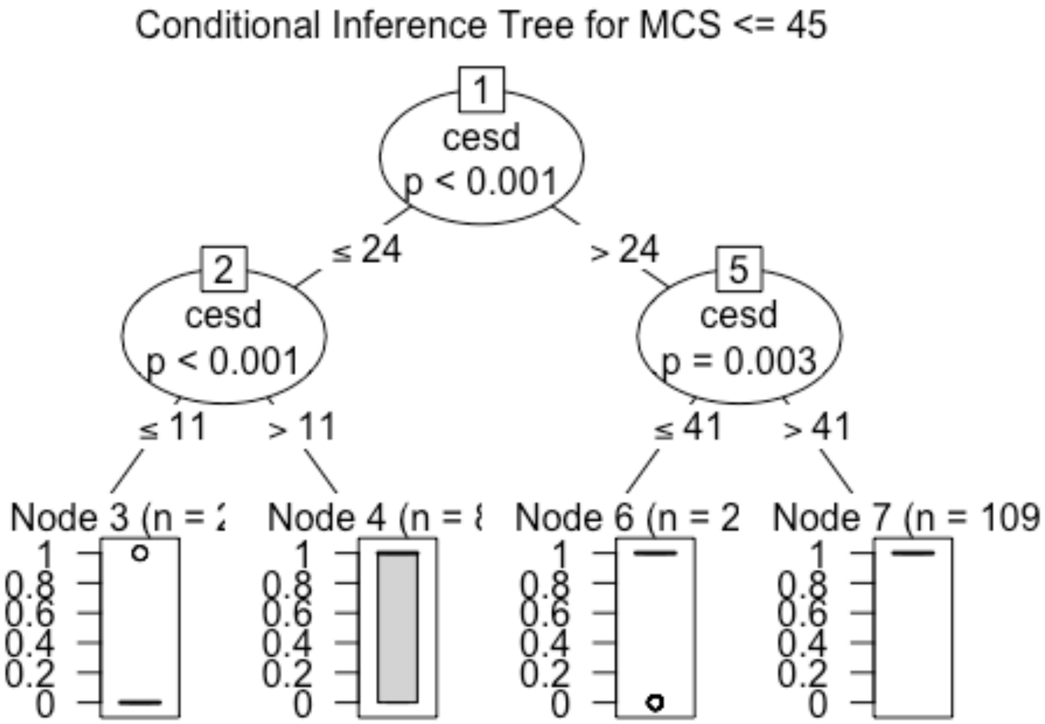
```
# look at cesd_gte16 with ctree from party
fitallpk <- party::ctree(cesd_gte16 ~ age + female + pss_fr +
                           homeless + pcs + mcs, data = h1)
class(fitallpk)

## [1] "BinaryTree"
## attr(,"package")
## [1] "party"

plot(fitallpk, main = "Conditional Inference Tree for CESD => 16")
```



PROBLEM 7: Fit a Conditional Classification Tree for MCS < 45

Using the `party` package, we can fit a conditional classification tree using the `ctree()` function. Let's do one for the indicator of depression `mcs_lt45` given the other variables in the `h1` dataset: `age`, `female`, `pss_fr`, `homeless`, `pcs`, `cesd`.

```
# look at mcs_lt45 with ctree from party
fitallpk_mcs <- party::ctree(mcs_lt45 ~ age + female + pss_fr +
                             homeless + pcs + cesd, data = h1)
class(fitallpk_mcs)

## [1] "BinaryTree"
## attr(,"package")
## [1] "party"

plot(fitallpk_mcs, main = "Conditional Inference Tree for MCS <= 45")
```

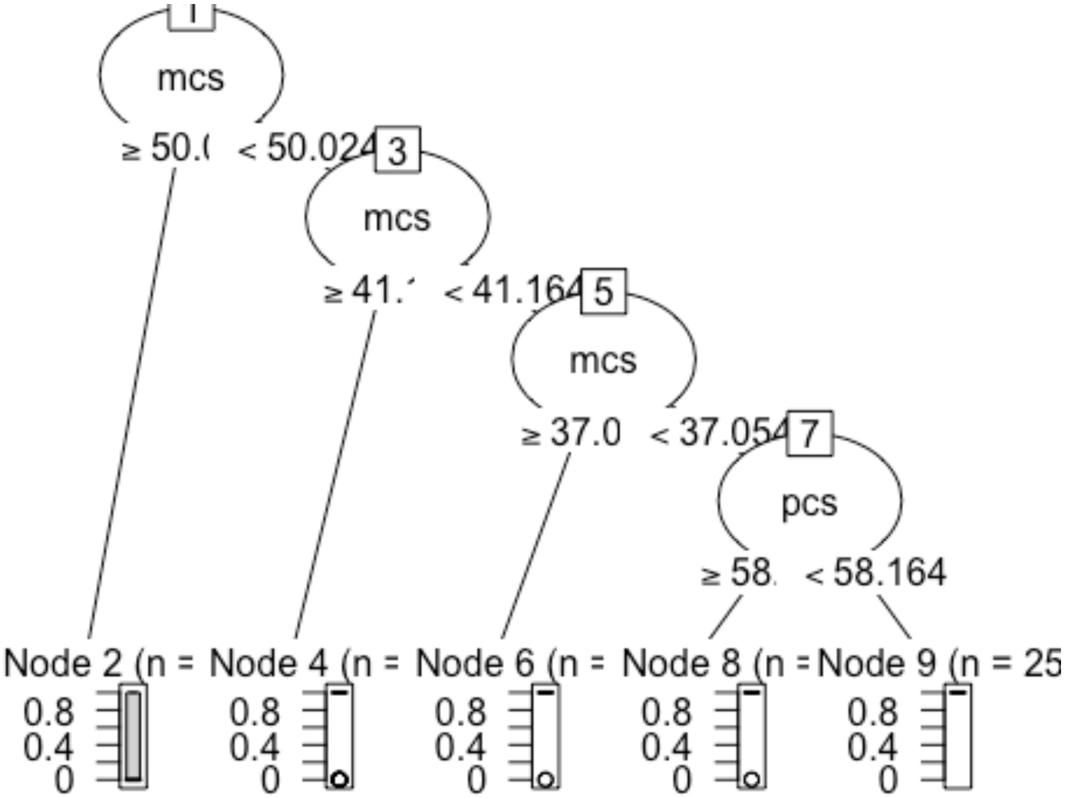


Recursive Partitioning of Classification Tree for CESD => 16

Here is the code doing the recursive partitioning of CESD => 16 on `age`, `female`, `pss_fr`, `homeless`, `pcs`, `mcs`. We're also using the `partykit` package to get prettier graphics for this classification tree.

```
# Recursive partitioning of CESD => 16 on age,
# female, pss_fr, homeless, pcs, mcs
whoIsDepressed <- rpart::rpart(cesd_gte16 ~ age + female +
                                 pss_fr + homeless + pcs + mcs,
                               data = h1,
                               control = rpart.control(cp = 0.001,
                                                       minbucket = 20))

whoIsDepressed

## n= 453
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 453 41.328920 0.8984547
##    2) mcs>=50.02446 51 12.156860 0.3921569 *
##    3) mcs< 50.02446 402 14.440300 0.9626866
##      6) mcs>=41.16363 59  8.949153 0.8135593 *
##      7) mcs< 41.16363 343  3.953353 0.9883382
##       14) mcs>=37.05422 38  1.894737 0.9473684 *
##       15) mcs< 37.05422 305  1.986885 0.9934426
##         30) pcs>=58.16405 51  1.921569 0.9607843 *
##         31) pcs< 58.16405 254  0.000000 1.0000000 *
```

```
library(partykit)
# Plot the tree
plot(partykit::as.party(whoIsDepressed))
```



PROBLEM 8: Recursive Partitioning of Classification Tree for MCS < 45

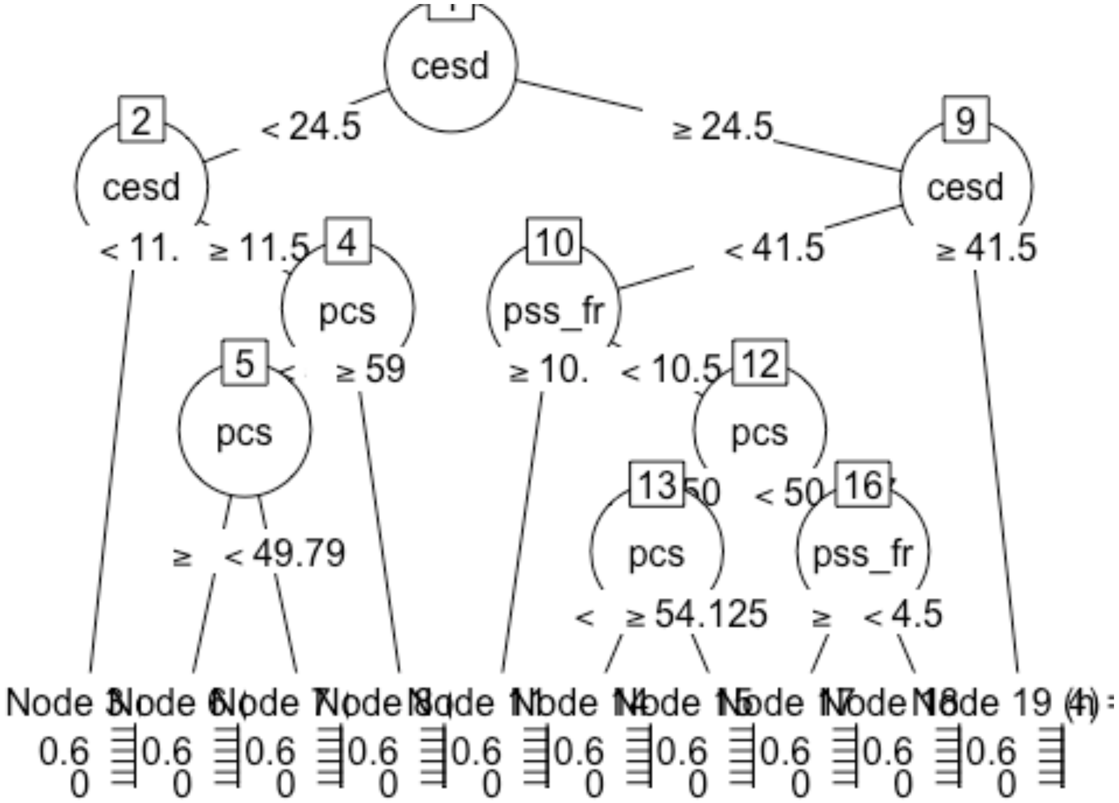
Using the code above to do recursive partitioning of MCS < 45 (`mcs_lt45`) on `age`, `female`, `pss_fr`, `homeless`, `pcs`, `cesd`. Also use the `partykit` package to get prettier graphics for this classification tree.

```
# Recursive partitioning of MCS <= 45 on age,
# female, pss_fr, homeless, pcs, cesd
whoIsDepressed_2 <- rpart::rpart(mcs_lt45 ~ age + female +
                                 pss_fr + homeless + pcs + cesd,
                               data = h1,
                               control = rpart.control(cp = 0.001,
                                                       minbucket = 20))
```

```
whoIsDepressed_2
```

```
## n= 453
##
## node), split, n, deviance, yval
##       * denotes terminal node
##
##  1) root 453 68.423840 0.8145695
##    2) cesd< 24.5 113 28.141590 0.4690265
##      4) cesd< 11.5 29  2.689655 0.1034483 *
##      5) cesd>=11.5 84 20.238100 0.5952381
##       10) pcs< 59.00035 64 15.937500 0.5312500
##         20) pcs>=49.7901 27  6.000000 0.3333333 *
##         21) pcs< 49.7901 37  8.108108 0.6756757 *
##       11) pcs>=59.00035 20  3.200000 0.8000000 *
##    3) cesd>=24.5 340 22.305880 0.9294118
##      6) cesd< 41.5 231 21.506490 0.8961039
##       12) pss_fr>=10.5 52  8.076923 0.8076923 *
##       13) pss_fr< 10.5 179 12.905030 0.9217877
##         26) pcs>=50.23704 80  8.750000 0.8750000
##           52) pcs< 54.12466 25  4.560000 0.7600000 *
##           53) pcs>=54.12466 55  3.709091 0.9272727 *
##         27) pcs< 50.23704 99  3.838384 0.9595960
##           54) pss_fr>=4.5 55  3.709091 0.9272727 *
##           55) pss_fr< 4.5 44  0.000000 1.0000000 *
##      7) cesd>=41.5 109  0.000000 1.0000000 *
```

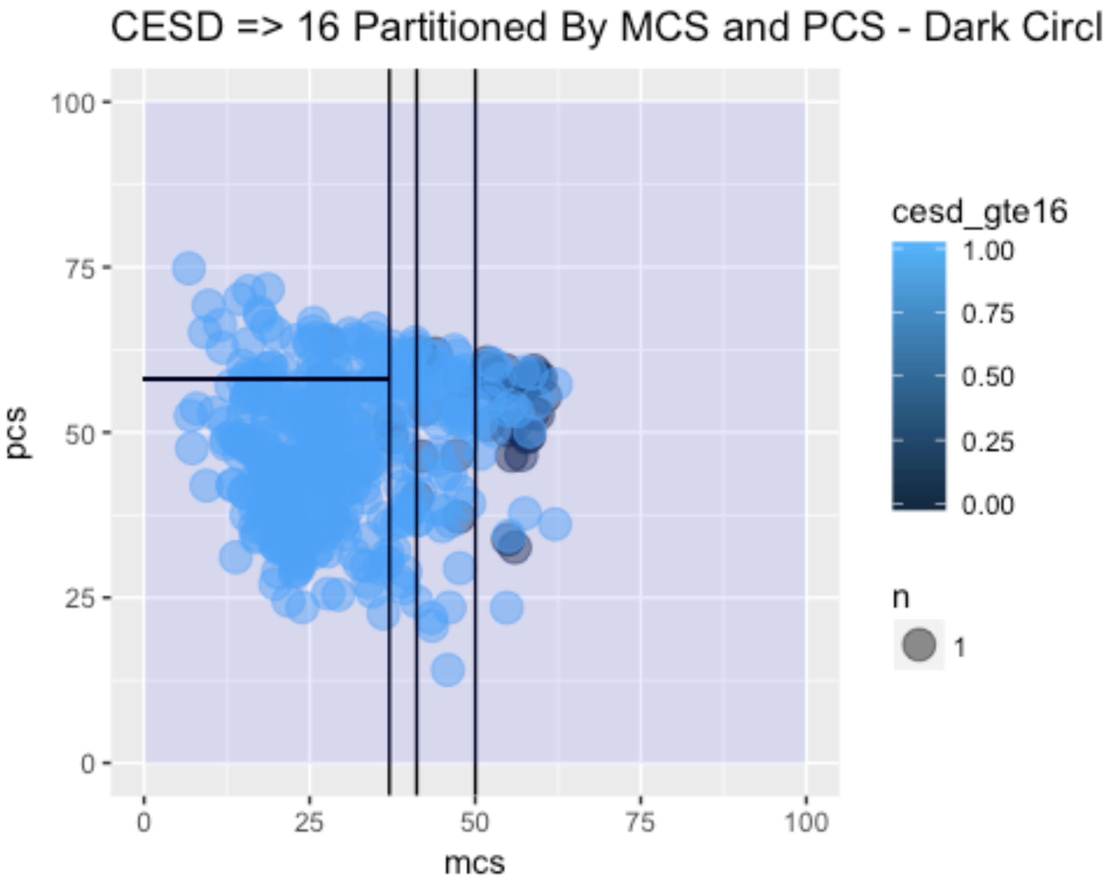
```
library(partykit)
# Plot the tree
plot(partykit::as.party(whoIsDepressed_2))
```



Scatterplot of recursive partitions for CESD => 16 for MCS and PCS

The code below creates a scatterplot of `pcs` and `mcs` where the points are colored by the indication of depression `cesd_gte16`. The lines have been inserted showing the dividing lines that best separate subjects with depression (CESD => 16) from those without depression (CESD < 16).

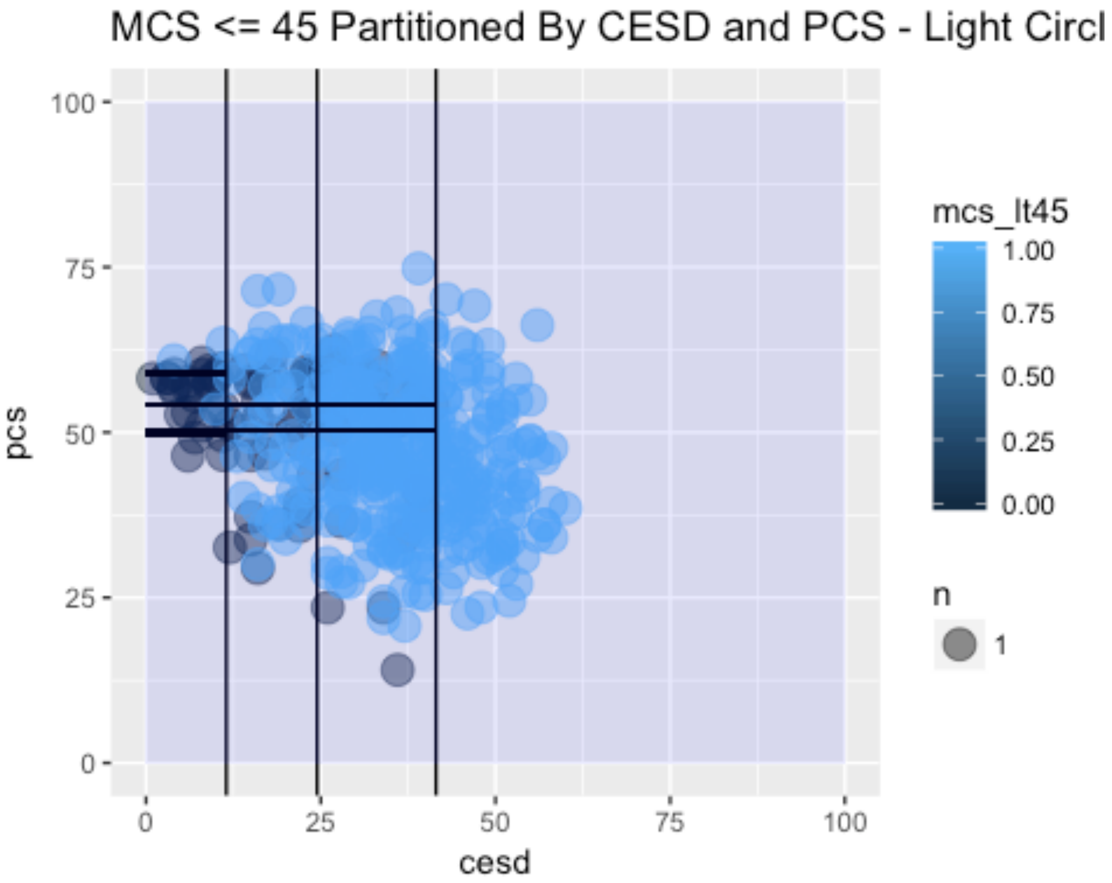
```
# EXTRA CREDIT
# Graph as partition
# using the break points shown from the
# conditional tree
ggplot(data = h1, aes(x = mcs, y = pcs)) +
  geom_count(aes(color = cesd_gte16), alpha = 0.5) +
  geom_vline(xintercept = 50.024) +
  geom_vline(xintercept = 41.164) +
  geom_vline(xintercept = 37.054) +
  geom_segment(x = 37.054, xend = 0, y = 58.164, yend = 58.164) +
  annotate("rect", xmin = 0, xmax = 100, ymin = 0, ymax = 100, fill = "blue",
alpha = 0.1) +
  ggtitle("CESD => 16 Partitioned By MCS and PCS - Dark Circles Not Depressed
")
```



EXTRA CREDIT Scatterplot of recursive partitions for MCS < 45 for PCS and CESD

Using the code above, create a scatterplot of `pcs` and `cesd` where the points are colored by the indication of poor mental health `mcs_lt45`. Play with the `geom_vline()` or `geom_hline()` or `geom_segment()` to insert lines that best separate subjects with poor mental health (MCS < 45) from those with normal to better than average mental health (MCS > 45).

```
ggplot(data = h1, aes(x = cesd, y = pcs)) +
  geom_count(aes(color = mcs_lt45), alpha = 0.5) +
  geom_vline(xintercept = 24.5) +
  geom_vline(xintercept = 11.5) +
  geom_vline(xintercept = 41.5) +
  geom_segment(x = 11.5, xend = 0, y = 59.00035, yend = 59.00035) +
  geom_segment(x = 11.5, xend = 0, y = 49.7901, yend = 49.7901) +
  geom_segment(x = 41.5, xend = 0, y = 50.23704, yend = 50.23704) +
  geom_segment(x = 41.5, xend = 0, y = 54.12466, yend = 54.12466) +
  annotate("rect", xmin = 0, xmax = 100, ymin = 0, ymax = 100, fill = "blue",
alpha = 0.1) +
  ggtitle("MCS <= 45 Partitioned By CESD and PCS - Light Circles Are Depresse
d")
```



Random Forest Model for CESD

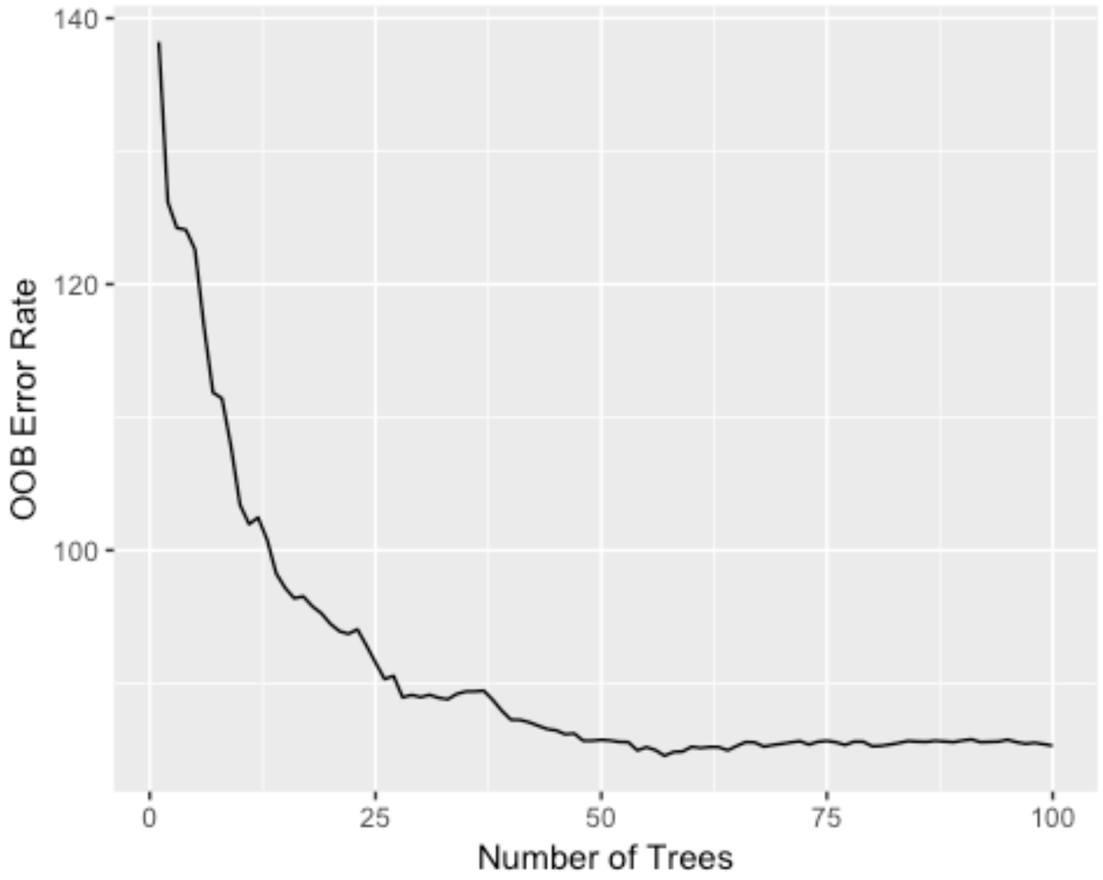
Now let's use a Random Forest approach for modeling the CESD by the other variables in the dataset:

- age
- female
- pss_fr
- homeless
- pcs
- mcs

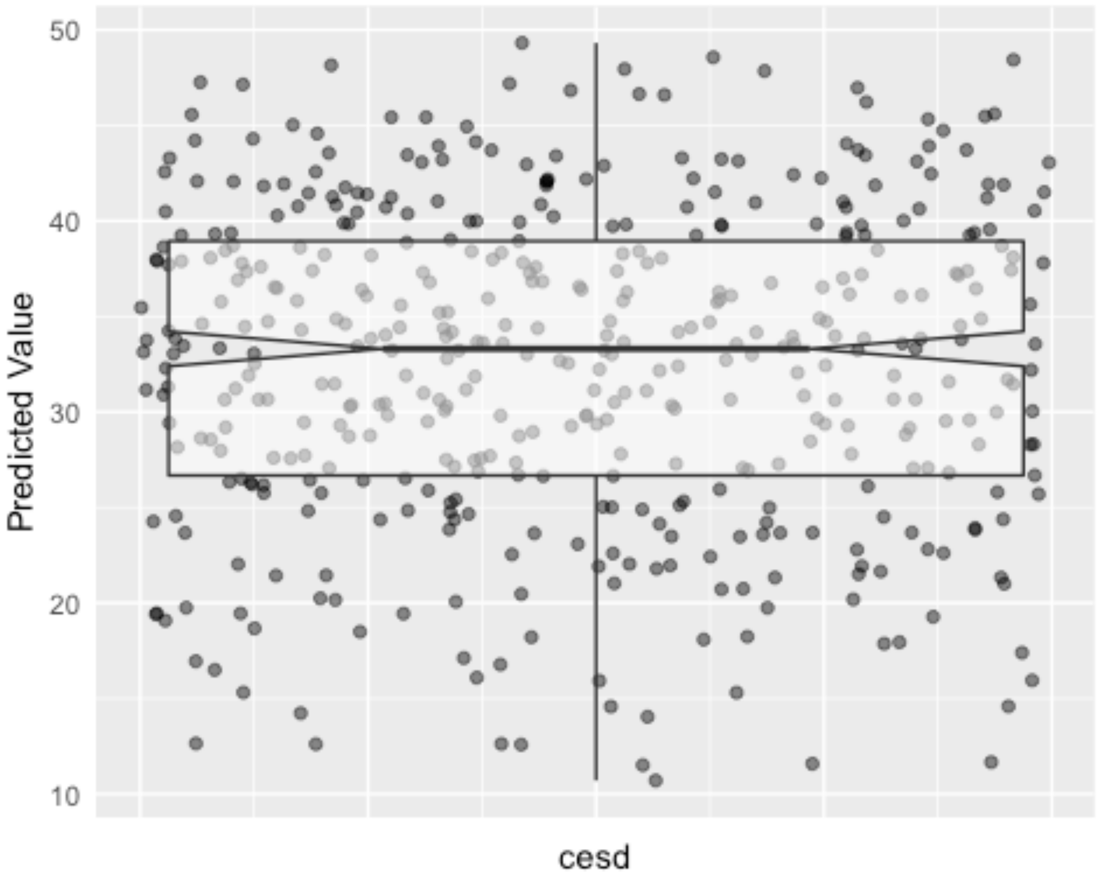
And using the code below, we'll explore how well the model converges and how well it does predicting CESD scores.

```
h1 <- as.data.frame(h1)
set.seed(131)
# Random Forest for the h1 dataset
fitallrf <- randomForestSRC::rfsrc(cesd ~ age + female +
                                     pss_fr + homeless + pcs + mcs,
                                   data = h1, ntree = 100,
                                   tree.err=TRUE)
```

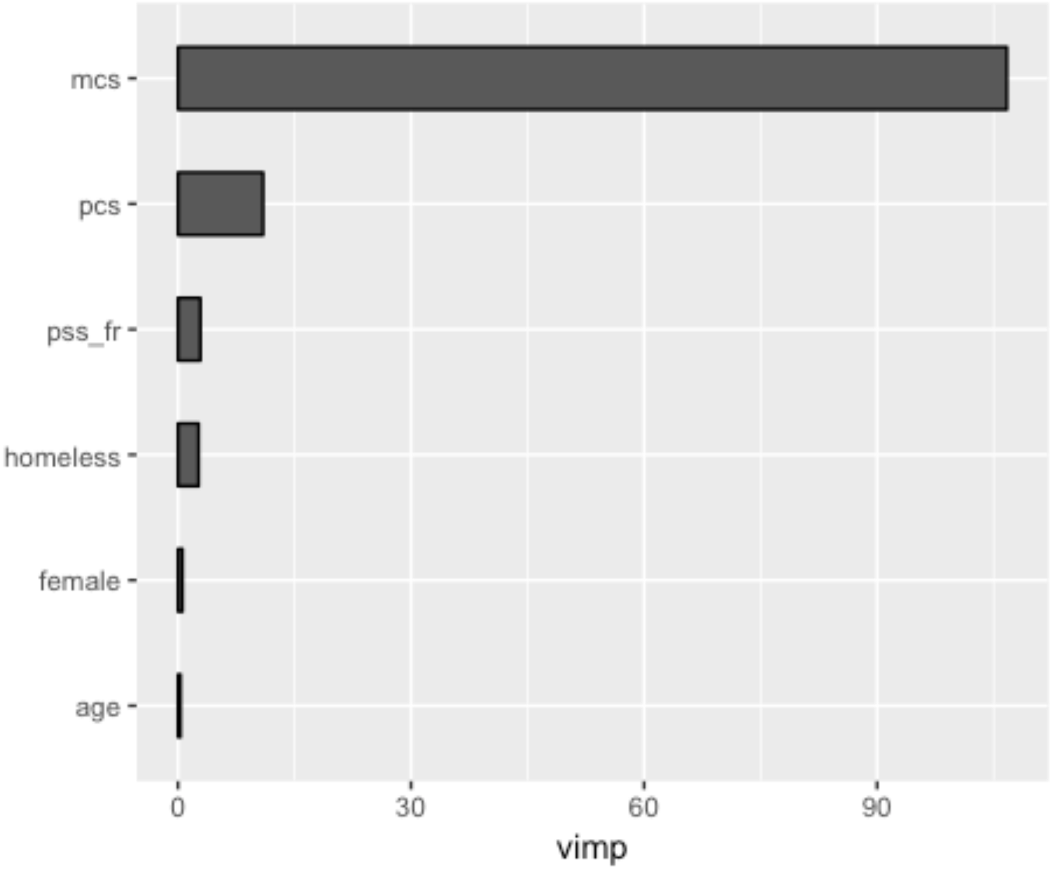
```
# view the results
fitallrf

##                          Sample size: 453
##                      Number of trees: 100
##            Forest terminal node size: 5
##        Average no. of terminal nodes: 91.16
## No. of variables tried at each split: 2
##               Total no. of variables: 6
##                             Analysis: RF-R
##                               Family: regr
##                       Splitting rule: mse
##                 % variance explained: 45.53
##                           Error rate: 85.3

gg_e <- ggRandomForests::gg_error(fitallrf)
plot(gg_e)
```



```
# Plot the predicted cesd values
plot(ggRandomForests::gg_rfsrc(fitallrf), alpha = 0.5)
```



```
# Plot the VIMP rankins of independent variables
plot(ggRandomForests::gg_vimp(fitallrf))
```



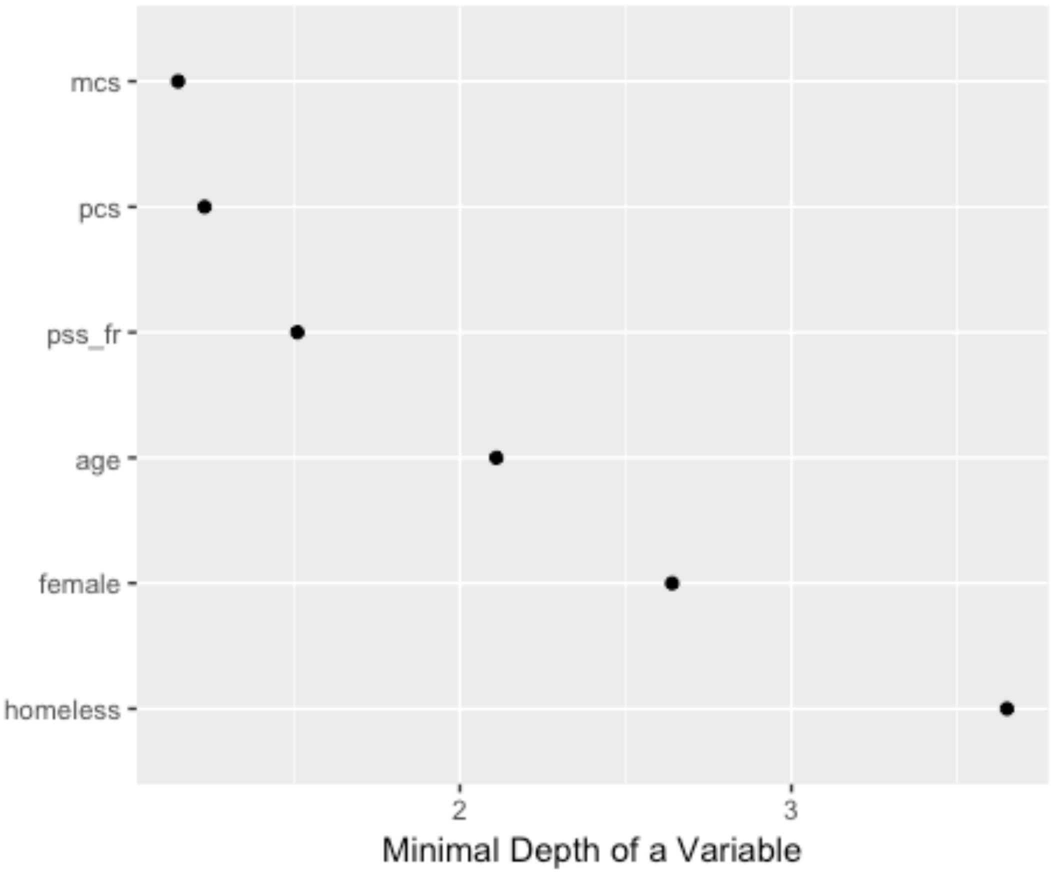
```
# Select the variables
varsel_cesd <- randomForestSRC::var.select(fitallrf)

## minimal depth variable selection ...
##
##
## -----------------------------------------------------------
## family               : regr
## var. selection       : Minimal Depth
## conservativeness     : medium
## x-weighting used?    : TRUE
## dimension            : 6
## sample size          : 453
## ntree                : 100
## nsplit               : 0
## mtry                 : 2
## nodesize             : 5
## refitted forest      : FALSE
## model size           : 6
## depth threshold      : 5.6833
## PE (true OOB)        : 85.3018
##
##
```

```
## Top variables:
##          depth vimp
## mcs       1.15   NA
## pcs       1.23   NA
## pss_fr    1.51   NA
## age       2.11   NA
## female    2.64   NA
## homeless  3.65   NA
## -----------------------------------------------------------
```

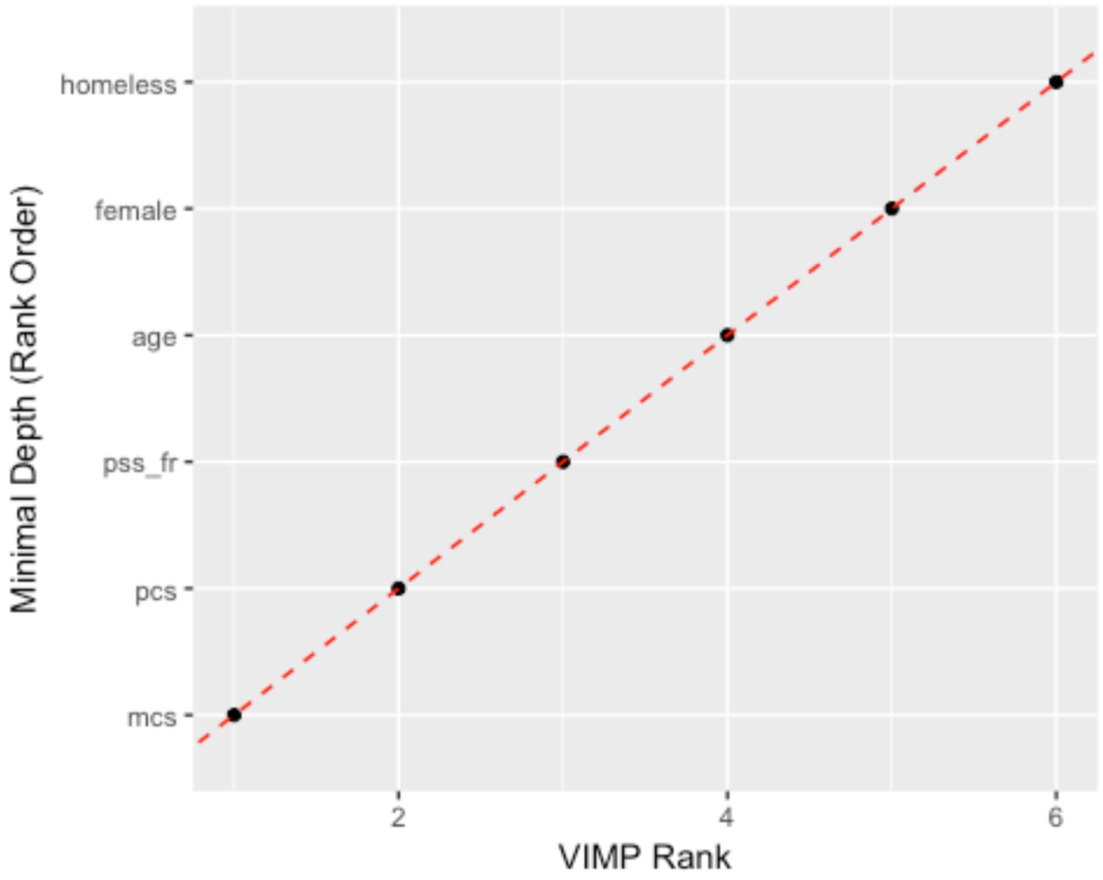
```
glimpse(varsel_cesd)
```

```
## List of 6
##  $ err.rate       : num 85.3
##  $ modelsize      : int 6
##  $ topvars        : chr [1:6] "mcs" "pcs" "pss_fr" "age" ...
##  $ varselect      :'data.frame': 6 obs. of  2 variables:
##   ..$ depth: num [1:6] 1.15 1.23 1.51 2.11 2.64 3.65
##   ..$ vimp : num [1:6] NA NA NA NA NA NA
##  $ rfsrc.refit.obj: NULL
##  $ md.obj         :List of 11
##   ..$ order                 : num [1:6, 1:2] 2.11 2.64 1.51 3.65 1.23 1.15
3.57 6.23 5.37 4.77 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   ..$ count                 : Named num [1:6] 0.1539 0.0816 0.1107 0.1075
0.091 ...
##   .. ..- attr(*, "names")= chr [1:6] "age" "female" "pss_fr" "homeless" ..
.
##   ..$ nodes.at.depth        : num [1:10000, 1:100] 2 4 7 7 10 14 12 11 9 7
...
##   ..$ sub.order             : NULL
##   ..$ threshold             : num 5.68
##   ..$ threshold.1se         : num 5.88
##   ..$ topvars               : chr [1:6] "age" "female" "pss_fr" "homeless"
...
##   ..$ topvars.1se           : chr [1:6] "age" "female" "pss_fr" "homeless"
...
##   ..$ percentile            : Named num [1:6] 0.194 0.26 0.141 0.357 0.12
...
##   .. ..- attr(*, "names")= chr [1:6] "age" "female" "pss_fr" "homeless" ..
.
##   ..$ density               : Named num [1:21] 0.0641 0.0968 0.1314 0.1307
0.1008 ...
##   .. ..- attr(*, "names")= chr [1:21] "0" "1" "2" "3" ...
##   ..$ second.order.threshold: num 10.1
```

```
# Save the gg_minimal_depth object for later use
gg_md <- ggRandomForests::gg_minimal_depth(varsel_cesd)
# Plot the object
plot(gg_md)
```



```
# Plot minimal depth v VIMP
gg_mdVIMP <- ggRandomForests::gg_minimal_vimp(gg_md)
plot(gg_mdVIMP)
```



PROBLEM 9: Fit a Random Forest Model for MCS

Now let's use a Random Forest approach for modeling the MCS by the other variables in the dataset:

- age
- female
- pss_fr
- homeless
- pcs
- cesd

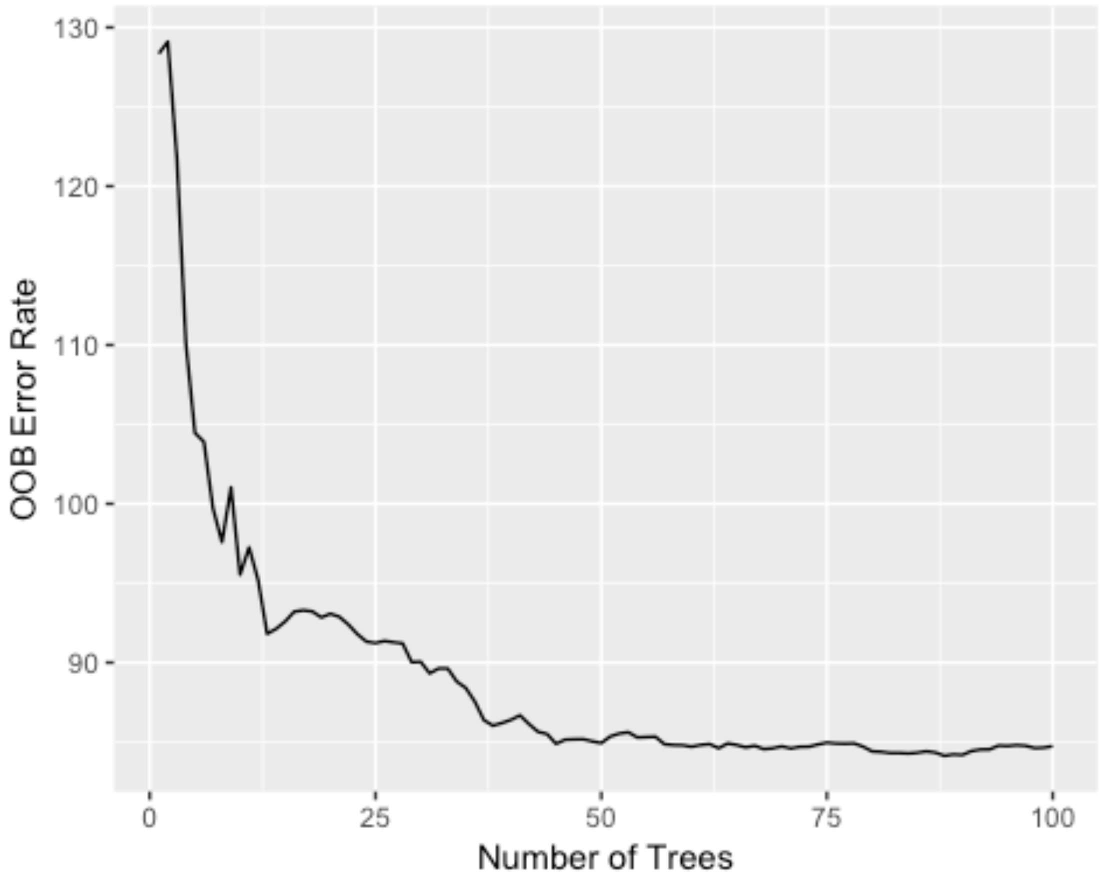
Use the code above to fit the model and explore how well the model converges and how well it does predicting MCS scores.

```
h1 <- as.data.frame(h1)
set.seed(131)
# Random Forest for the h1 dataset
fitallrf_mcs <- randomForestSRC::rfsrc(mcs ~ age + female +
                                          pss_fr + homeless + pcs + cesd,
                                        data = h1, ntree = 100,
                                        tree.err=TRUE)
```

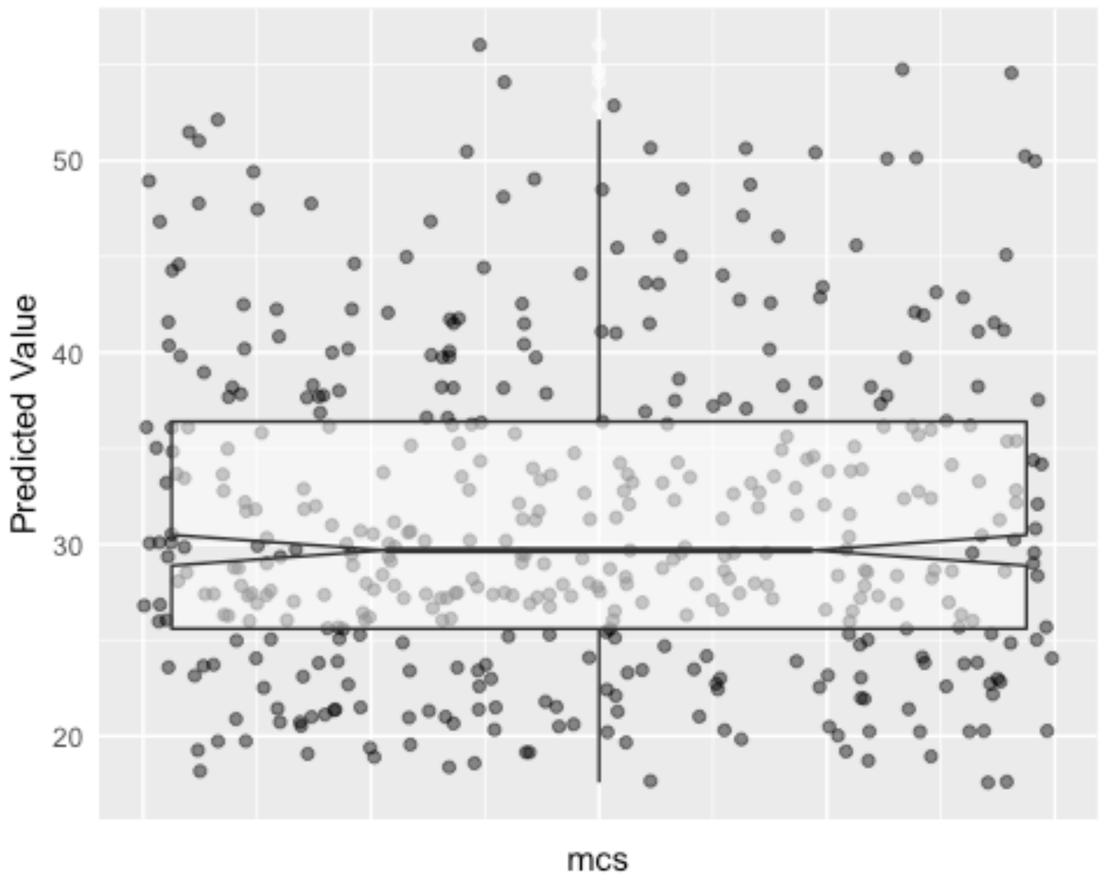
```
# view the results
fitallrf_mcs
```

```
##                          Sample size: 453
##                      Number of trees: 100
##            Forest terminal node size: 5
##        Average no. of terminal nodes: 90.85
## No. of variables tried at each split: 2
##               Total no. of variables: 6
##                             Analysis: RF-R
##                               Family: regr
##                       Splitting rule: mse
##                 % variance explained: 48.6
##                           Error rate: 84.74
```

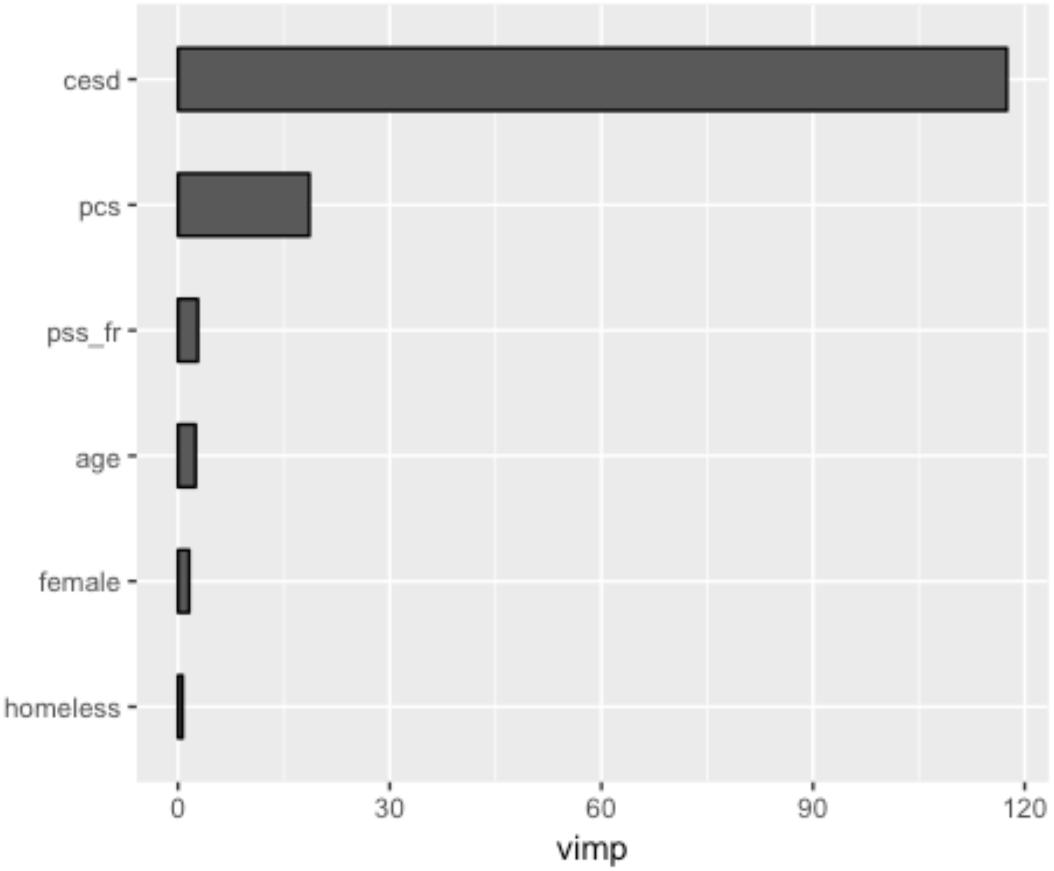
```
gg_e_mcs <- ggRandomForests::gg_error(fitallrf_mcs)
plot(gg_e_mcs)
```



```
# Plot the predicted mcs values
plot(ggRandomForests::gg_rfsrc(fitallrf_mcs), alpha = 0.5)
```



```
# Plot the VIMP rankins of independent variables
plot(ggRandomForests::gg_vimp(fitallrf_mcs))
```



```
# Select the variables
varsel_mcs <- randomForestSRC::var.select(fitallrf_mcs)
```

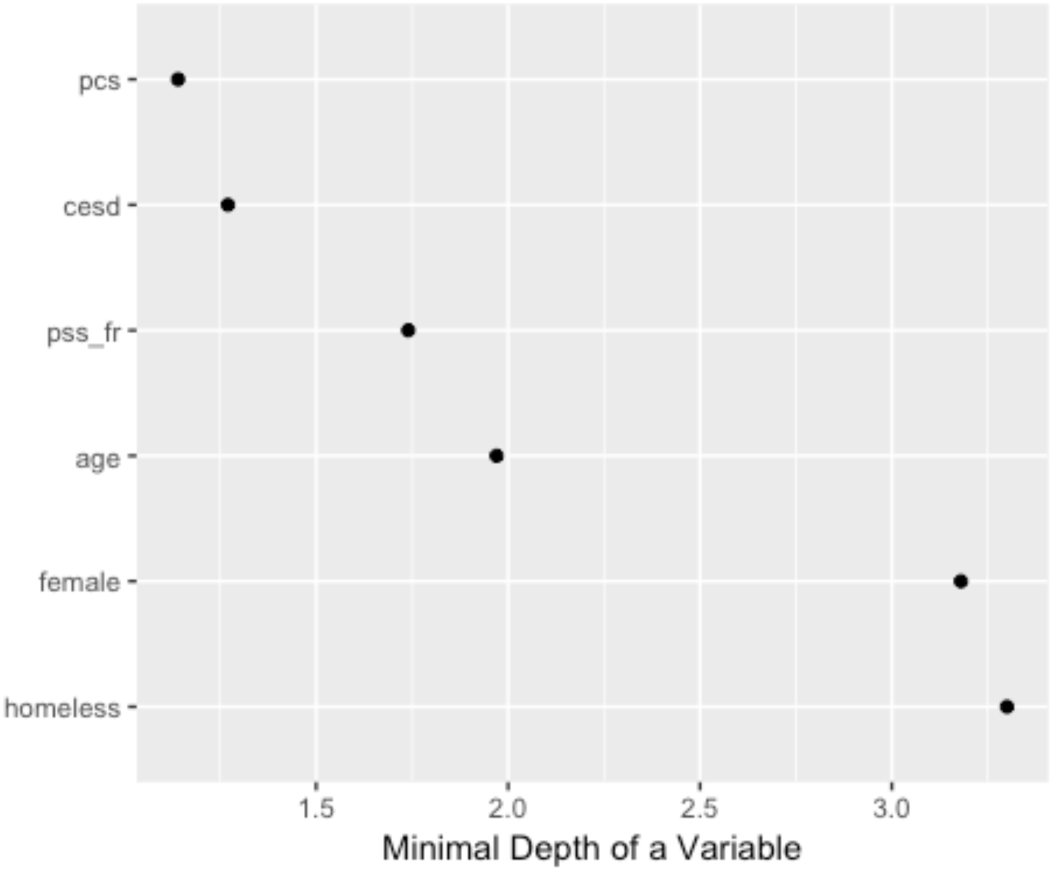
```
## minimal depth variable selection ...
##
##
## -----------------------------------------------------------
## family               : regr
## var. selection       : Minimal Depth
## conservativeness     : medium
## x-weighting used?    : TRUE
## dimension            : 6
## sample size          : 453
## ntree                : 100
## nsplit               : 0
## mtry                 : 2
## nodesize             : 5
## refitted forest      : FALSE
## model size           : 6
## depth threshold      : 5.9024
## PE (true OOB)        : 84.7368
##
##
```

```
## Top variables:
##          depth vimp
## pcs       1.14   NA
## cesd      1.27   NA
## pss_fr    1.74   NA
## age       1.97   NA
## female    3.18   NA
## homeless  3.30   NA
## -----------------------------------------------------------
```

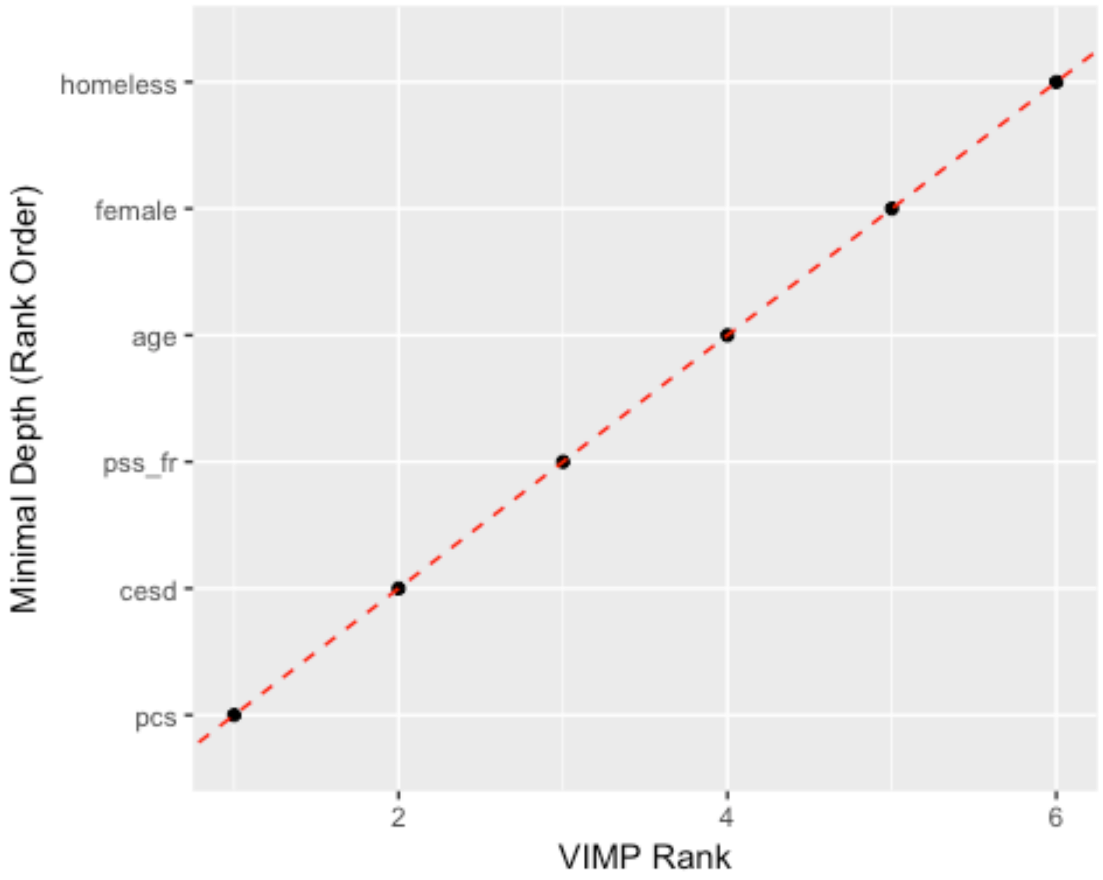
```
glimpse(varsel_mcs)
```

```
## List of 6
##  $ err.rate       : num 84.7
##  $ modelsize      : int 6
##  $ topvars        : chr [1:6] "pcs" "cesd" "pss_fr" "age" ...
##  $ varselect      :'data.frame': 6 obs. of  2 variables:
##   ..$ depth: num [1:6] 1.14 1.27 1.74 1.97 3.18 3.3
##   ..$ vimp : num [1:6] NA NA NA NA NA NA
##  $ rfsrc.refit.obj: NULL
##  $ md.obj         :List of 11
##   ..$ order                  : num [1:6, 1:2] 1.97 3.18 1.74 3.3 1.14 1.27
4.04 5.38 5.76 4.64 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   ..$ count                  : Named num [1:6] 0.1396 0.0918 0.1192 0.0993
0.092 ...
##   .. ..- attr(*, "names")= chr [1:6] "age" "female" "pss_fr" "homeless" ..
.
##   ..$ nodes.at.depth         : num [1:10000, 1:100] 2 4 5 9 10 13 13 13 7 3
...
##   ..$ sub.order              : NULL
##   ..$ threshold              : num 5.9
##   ..$ threshold.1se          : num 6.1
##   ..$ topvars                : chr [1:6] "age" "female" "pss_fr" "homeless"
...
##   ..$ topvars.1se            : chr [1:6] "age" "female" "pss_fr" "homeless"
...
##   ..$ percentile             : Named num [1:6] 0.172 0.299 0.161 0.303 0.10
4 ...
##   .. ..- attr(*, "names")= chr [1:6] "age" "female" "pss_fr" "homeless" ..
.
##   ..$ density                : Named num [1:23] 0.0612 0.0906 0.1222 0.1232
0.0981 ...
##   .. ..- attr(*, "names")= chr [1:23] "0" "1" "2" "3" ...
##   ..$ second.order.threshold: num 10.4
```

```
# Save the gg_minimal_depth object for later use
gg_md_mcs <- ggRandomForests::gg_minimal_depth(varsel_mcs)
# Plot the object
plot(gg_md_mcs)
```



```
# Plot minimal depth v VIMP
gg_mdVIMP_mcs <- ggRandomForests::gg_minimal_vimp(gg_md_mcs)
plot(gg_mdVIMP_mcs)
```

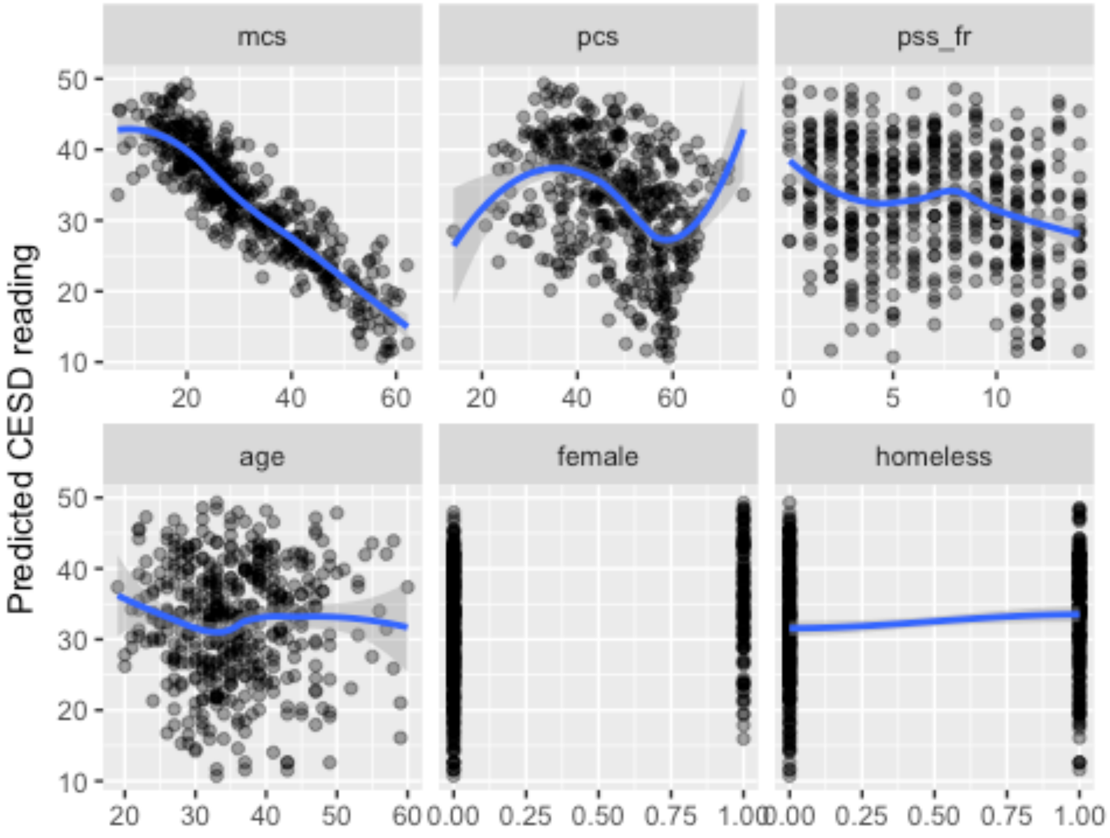


```

## Create Plots of How Well Each Variable Predicts CESD

Using the code below, we can see how well each variable predicts CESD scores.

```
#Create the variable dependence object from the random forest
gg_v <- ggRandomForests::gg_variable(fitallrf)

gg_v <- ggRandomForests::gg_variable(fitallrf)

# Use the top ranked minimal depth variables only, plotted in minimal depth r
ank order
xvar <- gg_md$topvars

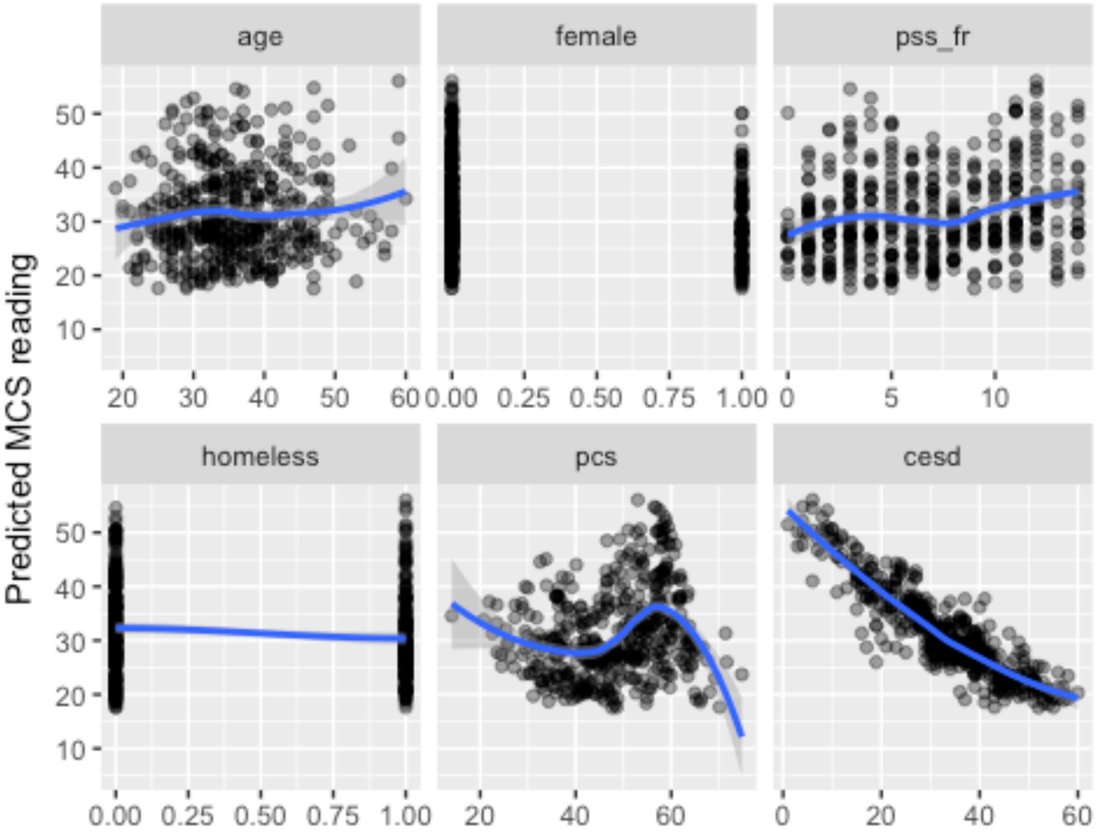
# Plot the variable list in a single panel plot
plot(gg_v, xvar = xvar, panel = TRUE, alpha = 0.4) +
  labs(y="Predicted CESD reading", x="")
```
```



PROBLEM 10: Create Plots of How Well Each Variable Predicts CESD*

Using the code above, see how well each variable predicts MCS scores given the other variables in the dataset h1.

```
#Create the variable dependence object from the random forest
gg_v_mcs <- ggRandomForests::gg_variable(fitallrf_mcs)

# Use the top ranked minimal depth variables only, plotted in minimal depth r
ank order
xvar_mcs <- gg_md_mcs$topvars

# Plot the variable list in a single panel plot
plot(gg_v_mcs, xvar_mcs = xvar_mcs, panel = TRUE, alpha = 0.4) +
  labs(y="Predicted MCS reading", x="")
```



**** Link to Github repo for Homework 7 *****
https://github.com/lpgleason/N741Spring2018_Homework7.git