PA trial #2 - averaged weather stations

Rachel Schattman

April 9, 2019

# load libraries

##   
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':  
##   
## filter, lag

## The following objects are masked from 'package:base':  
##   
## intersect, setdiff, setequal, union

## -------------------------------------------------------------------------

## You have loaded plyr after dplyr - this is likely to cause problems.  
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:  
## library(plyr); library(dplyr)

## -------------------------------------------------------------------------

##   
## Attaching package: 'plyr'

## The following objects are masked from 'package:dplyr':  
##   
## arrange, count, desc, failwith, id, mutate, rename, summarise,  
## summarize

## Warning: package 'randomForest' was built under R version 3.5.3

## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##   
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':  
##   
## combine

## Warning: package 'Metrics' was built under R version 3.5.3

## Warning: package 'randomForestExplainer' was built under R version 3.5.3

# Helpful resources:

## <https://www.r-bloggers.com/how-to-implement-random-forests-in-r/>

## <https://cran.r-project.org/web/packages/randomForest/randomForest.pdf>

## <https://rpubs.com/mbaumer/randomForest>

## <https://cran.rstudio.com/web/packages/randomForestExplainer/vignettes/randomForestExplainer.html>

script\_path <- "C:/Users/rschattman/Documents/Research/RandomForestRMA/data"  
in\_dir <- "C:/Users/rschattman/Documents/Research/RandomForestRMA/data"  
out\_dir <- "C:/Users/rschattman/Documents/Research/RandomForestRMA/output/data"

# Read in data and combine into single dataframe

## Year State WetDollars WetAcres nWet DryDollars DryAcres nDry  
## 1 2001 PA 129986.0 1700.12 57 14721696 216408.04 478  
## 13 2002 PA 625739.0 8657.63 155 54954463 613635.76 619  
## 25 2003 PA 20514682.0 221234.13 992 111925 813.80 21  
## 37 2004 PA 10553409.0 98898.31 839 77650 557.83 14  
## 49 2005 PA 426305.2 7509.00 188 11030151 136717.00 531  
## 61 2006 PA 6227641.6 53338.00 563 5463886 74329.00 286  
## StateCollege\_PRCP.1 Lebanon\_PRCP.1 Selinsgrove\_PRCP.1  
## 1 3.39 6.69 3.71  
## 13 4.26 5.79 4.52  
## 25 5.99 5.69 5.49  
## 37 9.64 5.14 7.07  
## 49 13.35 13.35 11.78  
## 61 10.81 11.53 11.11  
## StateCollege\_PRCP.2 Lebanon\_PRCP.2 Selinsgrove\_PRCP.2  
## 1 1.87 4.08 2.85  
## 13 3.03 0.91 2.44  
## 25 9.94 10.84 8.65  
## 37 6.65 6.28 5.49  
## 49 7.22 4.63 4.19  
## 61 3.31 5.79 4.45  
## StateCollege\_PRCP.3 Lebanon\_PRCP.3 Selinsgrove\_PRCP.3  
## 1 10.70 8.51 12.10  
## 13 10.17 11.54 7.28  
## 25 8.02 9.24 8.30  
## 37 7.39 4.87 5.77  
## 49 9.63 9.94 12.44  
## 61 3.25 1.82 2.75  
## StateCollege\_PRCP.4 Lebanon\_PRCP.4 Selinsgrove\_PRCP.4  
## 1 6.75 6.93 5.65  
## 13 7.04 11.93 9.16  
## 25 6.90 7.93 7.93  
## 37 11.22 12.35 12.21  
## 49 3.79 11.58 11.50  
## 61 6.62 9.33 8.51  
## StateCollege\_PRCP.5 Lebanon\_PRCP.5 Selinsgrove\_PRCP.5  
## 1 3.51 9.61 4.78  
## 13 16.43 13.19 12.81  
## 25 9.29 13.51 7.92  
## 37 9.44 12.11 10.52  
## 49 5.23 2.43 4.00  
## 61 4.98 7.83 6.90  
## StateCollege\_PRCP.6 Lebanon\_PRCP.6 Selinsgrove\_PRCP.6  
## 1 12.76 6.79 16.53  
## 13 19.73 11.23 11.96  
## 25 13.53 20.26 12.89  
## 37 8.69 12.27 9.20  
## 49 4.77 9.56 6.66  
## 61 14.86 33.07 26.70  
## StateCollege\_PRCP.7 Lebanon\_PRCP.7 Selinsgrove\_PRCP.7  
## 1 6.12 9.18 4.40  
## 13 2.76 4.48 3.87  
## 25 13.74 13.54 15.73  
## 37 18.52 18.87 18.21  
## 49 7.57 16.46 12.53  
## 61 8.32 10.24 11.71  
## StateCollege\_PRCP.8 Lebanon\_PRCP.8 Selinsgrove\_PRCP.8  
## 1 9.88 10.02 2.50  
## 13 6.35 5.57 2.72  
## 25 23.11 15.03 16.85  
## 37 18.20 17.77 12.36  
## 49 7.63 7.15 8.52  
## 61 8.05 8.63 16.77  
## StateCollege\_PRCP.9 Lebanon\_PRCP.9 Selinsgrove\_PRCP.9  
## 1 7.89 6.67 16.41  
## 13 8.51 9.71 13.24  
## 25 18.32 12.71 13.29  
## 37 26.84 23.98 25.30  
## 49 2.18 1.63 2.07  
## 61 10.15 10.83 13.56  
## StateCollege\_PRCP.10 Lebanon\_PRCP.10 Selinsgrove\_PRCP.10  
## 1 4.15 2.18 2.77  
## 13 14.11 15.57 16.43  
## 25 7.86 13.45 12.85  
## 37 5.47 6.88 10.31  
## 49 14.35 17.85 18.31  
## 61 13.22 12.15 9.01  
## StateCollege\_PRCP.11 Lebanon\_PRCP.11 Selinsgrove\_PRCP.11  
## 1 3.92 3.45 3.20  
## 13 7.30 10.05 8.51  
## 25 11.68 9.82 9.66  
## 37 7.50 8.39 9.17  
## 49 10.37 7.58 11.19  
## 61 7.20 11.24 11.90  
## StateCollege\_PRCP.12 Lebanon\_PRCP.12 Selinsgrove\_PRCP.12 JAN  
## 1 6.17 5.93 6.28 4.596667  
## 13 8.33 11.77 9.68 4.856667  
## 25 10.85 9.31 11.34 5.723333  
## 37 6.83 10.07 8.96 7.283333  
## 49 5.91 6.37 6.03 12.826667  
## 61 3.48 5.10 8.38 11.150000  
## FEB MAR APR MAY JUN JUL AUG  
## 1 2.933333 10.436667 6.443333 5.966667 12.026667 6.566667 7.466667  
## 13 2.126667 9.663333 9.376667 14.143333 14.306667 3.703333 4.880000  
## 25 9.810000 8.520000 7.586667 10.240000 15.560000 14.336667 18.330000  
## 37 6.140000 6.010000 11.926667 10.690000 10.053333 18.533333 16.110000  
## 49 5.346667 10.670000 8.956667 3.886667 6.996667 12.186667 7.766667  
## 61 4.516667 2.606667 8.153333 6.570000 24.876667 10.090000 11.150000  
## SEP OCT NOV DEC  
## 1 10.32333 3.033333 3.523333 6.126667  
## 13 10.48667 15.370000 8.620000 9.926667  
## 25 14.77333 11.386667 10.386667 10.500000  
## 37 25.37333 7.553333 8.353333 8.620000  
## 49 1.96000 16.836667 9.713333 6.103333  
## 61 11.51333 11.460000 10.113333 5.653333

# Create new data frames with one dependent variable

#head(PAbeta\_wide)  
WetAcres <- PAbeta\_wide[,c(1,3,6:17)] #subset year, dependent variable, and all precip columes  
WetDollars <- PAbeta\_wide[,c(1,2,6:17)]  
DryAcres <- PAbeta\_wide[,c(1,5,6:17)]  
DryDollars <- PAbeta\_wide[,c(1,4,6:17)]

# Review data

# Split into trainning, validation, and test sets

# Create Random Forest Model and test performance metrics

## Wet Acres

Mod1 <- randomForest(WetAcres ~ .,   
 data = Wettrain,   
 ntree = 500,   
 #method = "anova",   
 importance = TRUE)  
  
#print(Mod1) # % of variance expalined is low. Tuning needed  
summary(Mod1)

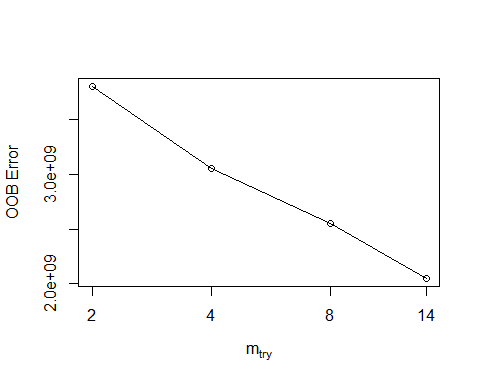
## Length Class Mode   
## call 5 -none- call   
## type 1 -none- character  
## predicted 14 -none- numeric   
## mse 500 -none- numeric   
## rsq 500 -none- numeric   
## oob.times 14 -none- numeric   
## importance 26 -none- numeric   
## importanceSD 13 -none- numeric   
## localImportance 0 -none- NULL   
## proximity 0 -none- NULL   
## ntree 1 -none- numeric   
## mtry 1 -none- numeric   
## forest 11 -none- list   
## coefs 0 -none- NULL   
## y 14 -none- numeric   
## test 0 -none- NULL   
## inbag 0 -none- NULL   
## terms 3 terms call

#plot(Mod1)  
  
pred <- predict(object = Mod1, newdata = Wettest)  
RMSE\_Mod1 <- rmse(actual = Wettest$WetAcres, #actual values  
 predicted = pred) #predicted values  
print(RMSE\_Mod1/mean(Wettest$WetAcres)) #tells us the %of the mean represented by RMSE. AKA "coefficient of variation"

## [1] 0.337439

# Tune mtry using OOB error  
set.seed(25)  
#train\_pred <- predict(object = Mod1, newdata = PAtrain)  
res <- tuneRF(x = Wettrain,  
 y = Wettrain$WetAcre,  
 proximity = TRUE, # includes a proximity matrix (how often 2 variables end up in the same terminal node)  
 ntree = 500,  
 stepfactor = 0.5,  
 doBest=TRUE, # Returns a random forest model with optimal mtry value  
 importance = TRUE)

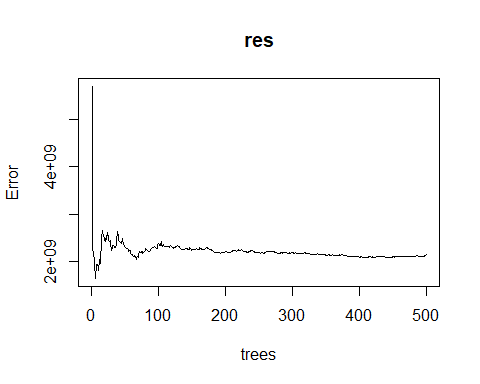
## mtry = 4 OOB error = 3059841972   
## Searching left ...  
## mtry = 2 OOB error = 3806132365   
## -0.2438983 0.05   
## Searching right ...  
## mtry = 8 OOB error = 2550655814   
## 0.1664093 0.05   
## mtry = 14 OOB error = 2050363312   
## 0.1961427 0.05



#localImp = TRUE)  
print(res)

##   
## Call:  
## randomForest(x = x, y = y, mtry = res[which.min(res[, 2]), 1], importance = TRUE, proximity = TRUE, stepfactor = 0.5)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 14  
##   
## Mean of squared residuals: 2140856041  
## % Var explained: 43.18

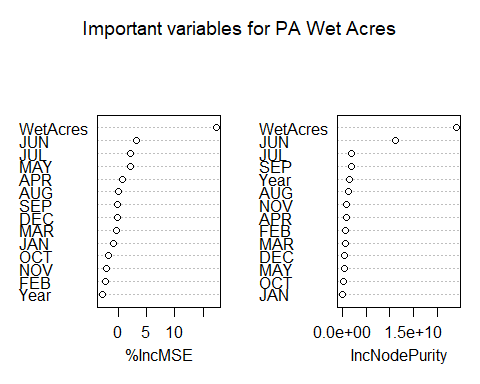
plot(res)



res$importance

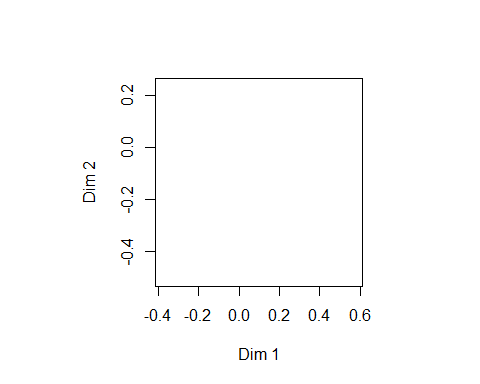
## %IncMSE IncNodePurity  
## Year -144371023 1494735121  
## WetAcres 2359169731 24009246795  
## JAN -5070070 126853083  
## FEB -54596634 667434265  
## MAR -9989064 643606457  
## APR 19345072 769812897  
## MAY 27811413 347278660  
## JUN 264494886 11284975851  
## JUL 68943725 1911118045  
## AUG 2248145 1273209938  
## SEP -1757757 1855819946  
## OCT -21664356 290479820  
## NOV -91874511 861574717  
## DEC -1584655 549036772

varImpPlot(res, main = "Important variables for PA Wet Acres") # Need this for pub



MDSplot(res, fac = Wettrain$WetAcre, k=2, palette = NULL, pch = 20) # Need this for pub

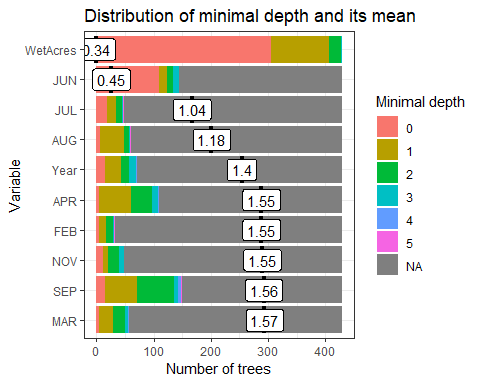
## Warning in RColorBrewer::brewer.pal(nlevs, "Set1"): minimal value for n is 3, returning requested palette with 3 different levels



# using randomForestExplainer  
min\_depth\_frame\_wetacres <- min\_depth\_distribution(res)

## Warning: Factor `split var` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`

plot\_min\_depth\_distribution(min\_depth\_frame\_wetacres, mean\_sample = "relevant\_trees")



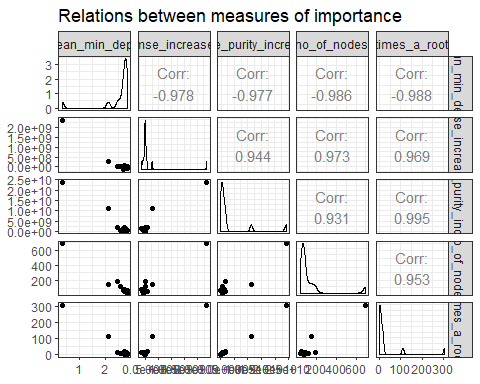
importance\_frame\_wetacres <- measure\_importance(res)

## Warning: Factor `split var` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`  
  
## Warning: Factor `split var` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`

print(importance\_frame\_wetacres)

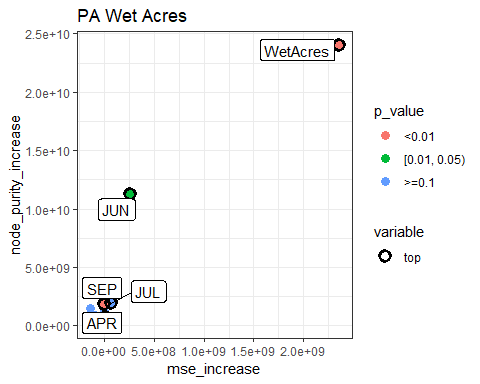
## variable mean\_min\_depth no\_of\_nodes mse\_increase node\_purity\_increase  
## 1 APR 2.5913473 121 19345072 769812897  
## 2 AUG 2.7046340 64 2248145 1273209938  
## 3 DEC 2.8220793 74 -1584655 549036772  
## 4 FEB 2.8438042 34 -54596634 667434265  
## 5 JAN 2.7706853 71 -5070070 126853083  
## 6 JUL 2.7337063 53 68943725 1911118045  
## 7 JUN 2.1126340 152 264494886 11284975851  
## 8 MAR 2.7650163 59 -9989064 643606457  
## 9 MAY 2.8095291 64 27811413 347278660  
## 10 NOV 2.7919814 51 -91874511 861574717  
## 11 OCT 2.7411655 77 -21664356 290479820  
## 12 SEP 2.4698368 188 -1757757 1855819946  
## 13 WetAcres 0.3449883 688 2359169731 24009246795  
## 14 Year 2.6919907 84 -144371023 1494735121  
## no\_of\_trees times\_a\_root p\_value  
## 1 110 5 7.267769e-01  
## 2 60 6 1.000000e+00  
## 3 68 3 9.999999e-01  
## 4 33 5 1.000000e+00  
## 5 67 0 1.000000e+00  
## 6 49 18 1.000000e+00  
## 7 144 110 1.414674e-02  
## 8 58 5 1.000000e+00  
## 9 60 1 1.000000e+00  
## 10 49 11 1.000000e+00  
## 11 74 0 9.999997e-01  
## 12 149 15 8.625107e-08  
## 13 429 305 2.573167e-310  
## 14 72 16 9.999895e-01

#Use ggpairs to determine which informative information from the importance frame  
plot\_importance\_ggpairs(importance\_frame\_wetacres) #the two LEAST negative correlations in top row are the winners (?)



# pvalue used to determine whether the actual number of splits using this variable exceeds what would be expect if the splits were random.  
plot\_multi\_way\_importance(importance\_frame\_wetacres, x\_measure = "mse\_increase", y\_measure =   
 "node\_purity\_increase", size\_measure = "p\_value", no\_of\_labels = 5, main = "PA Wet Acres") #top 5 variables are labeled

## Warning: Using alpha for a discrete variable is not advised.



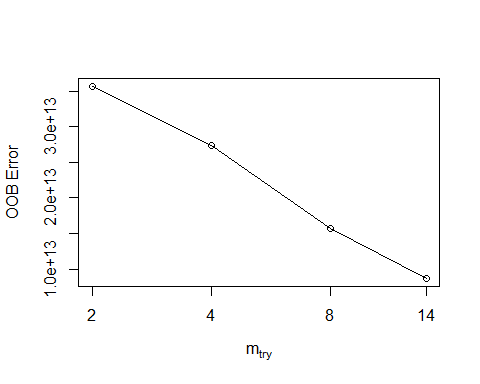
## Wet Dollars

# Split into trainning, validation, and test sets  
set.seed(25)  
assignment <- sample(1:3, size = nrow(WetDollars), prob = c(0.7, 0.15, 0.15), replace = TRUE)  
  
Wettrain2 <- WetDollars[assignment == 1,]  
Wetvalid2 <- WetDollars[assignment == 2,]  
Wettest2 <- WetDollars[assignment == 3,]  
  
#summary(Wettrain2)  
#summary(Wetvalid2)  
#summary(Wettest2)  
  
  
Mod2 <- randomForest(WetDollars ~ .,   
 data = Wettrain2,  
 ntree = 500,   
 #method = "anova",   
 importance = TRUE)  
  
#print(Mod2) # % of variance expalined is low. Tuning needed  
#summary(Mod2)  
#plot(Mod2)  
  
pred2 <- predict(object = Mod2, newdata = Wettest2)  
RMSE\_Mod2 <- rmse(actual = Wettest2$WetDollars, #actual values  
 predicted = pred2) #predicted values  
print(RMSE\_Mod2/mean(Wettest2$WetDollars)) #tells us the %of the mean represented by RMSE. AKA "coefficient of variation"

## [1] 0.3798737

# Tune mtry using OOB error  
set.seed(25)  
#train\_pred <- predict(object = Mod1, newdata = PAtrain)  
res2 <- tuneRF(x = Wettrain2,  
 y = Wettrain2$WetDollars,  
 ntree = 500,  
 proximity = TRUE, # includes a proximity matrix (how often 2 variables end up in the same terminal node)  
 stepfactor = 0.5,  
 doBest=TRUE, # Returns a random forest model with optimal mtry value  
 importance = TRUE)

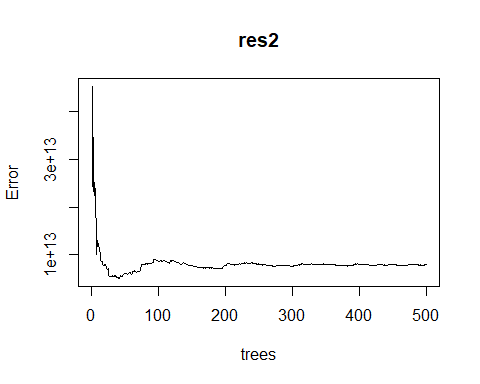
## mtry = 4 OOB error = 2.741795e+13   
## Searching left ...  
## mtry = 2 OOB error = 3.569785e+13   
## -0.3019885 0.05   
## Searching right ...  
## mtry = 8 OOB error = 1.56655e+13   
## 0.4286407 0.05   
## mtry = 14 OOB error = 8.682468e+12   
## 0.4457586 0.05



#localImp = TRUE)  
print(res2)

##   
## Call:  
## randomForest(x = x, y = y, mtry = res[which.min(res[, 2]), 1], importance = TRUE, proximity = TRUE, stepfactor = 0.5)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 14  
##   
## Mean of squared residuals: 7.874921e+12  
## % Var explained: 84.26

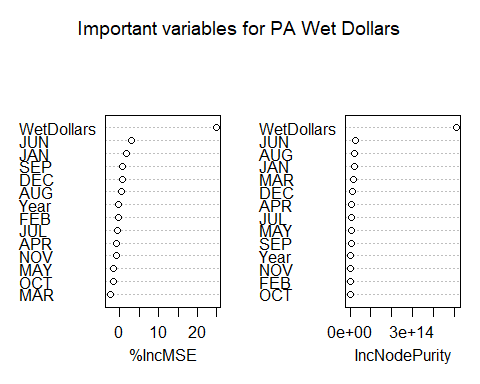
plot(res2)



res2$importance

## %IncMSE IncNodePurity  
## Year -3.230487e+10 4.453218e+12  
## WetDollars 6.110619e+13 5.106766e+14  
## JAN 6.488473e+11 1.893420e+13  
## FEB -2.593717e+10 3.063050e+12  
## MAR -6.879893e+11 1.440424e+13  
## APR -6.209974e+10 7.032181e+12  
## MAY -2.809741e+11 5.497693e+12  
## JUN 1.223908e+12 2.414703e+13  
## JUL -9.406841e+10 6.679720e+12  
## AUG 6.087577e+10 2.094456e+13  
## SEP 1.585961e+10 5.303694e+12  
## OCT -1.713034e+11 2.181112e+12  
## NOV -1.937835e+11 3.064942e+12  
## DEC 1.013271e+11 1.180301e+13

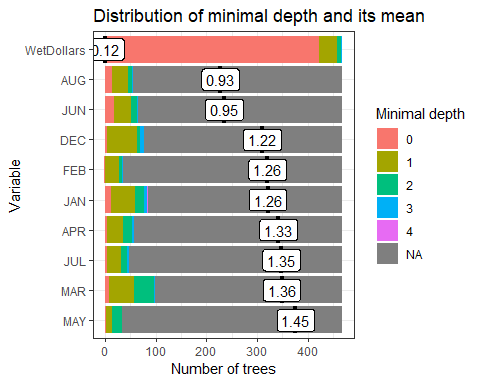
varImpPlot(res2, main = "Important variables for PA Wet Dollars")



# using randomForestExplainer  
min\_depth\_frame\_wetdollars <- min\_depth\_distribution(res2)

## Warning: Factor `split var` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`

plot\_min\_depth\_distribution(min\_depth\_frame\_wetdollars, mean\_sample = "relevant\_trees")



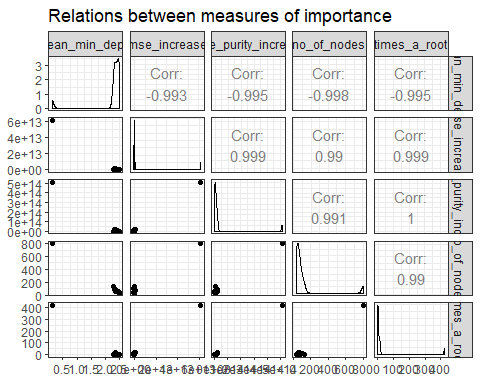
importance\_frame\_wetdollars <- measure\_importance(res2)

## Warning: Factor `split var` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`  
  
## Warning: Factor `split var` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`

print(importance\_frame\_wetdollars)

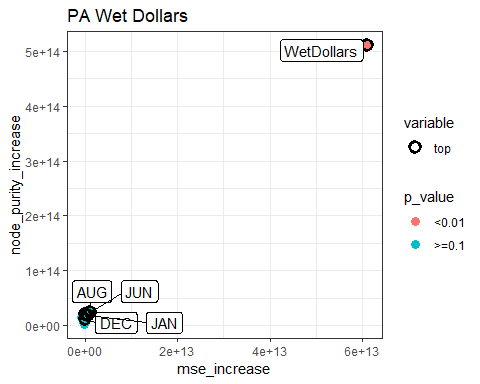
## variable mean\_min\_depth no\_of\_nodes mse\_increase  
## 1 APR 2.4141111 59 -6.209974e+10  
## 2 AUG 2.3683077 60 6.087577e+10  
## 3 DEC 2.3430000 82 1.013271e+11  
## 4 FEB 2.4662650 35 -2.593717e+10  
## 5 JAN 2.3269487 90 6.488473e+11  
## 6 JUL 2.4399145 50 -9.406841e+10  
## 7 JUN 2.3403675 70 1.223908e+12  
## 8 MAR 2.3112821 106 -6.879893e+11  
## 9 MAY 2.4857692 34 -2.809741e+11  
## 10 NOV 2.4953590 42 -1.937835e+11  
## 11 OCT 2.4599145 60 -1.713034e+11  
## 12 SEP 2.4985897 33 1.585961e+10  
## 13 WetDollars 0.1217949 799 6.110619e+13  
## 14 Year 2.2859744 129 -3.230487e+10  
## node\_purity\_increase no\_of\_trees times\_a\_root p\_value  
## 1 7.032181e+12 57 5 1.0000000  
## 2 2.094456e+13 56 15 1.0000000  
## 3 1.180301e+13 77 4 0.9998666  
## 4 3.063050e+12 35 1 1.0000000  
## 5 1.893420e+13 85 12 0.9974609  
## 6 6.679720e+12 48 4 1.0000000  
## 7 2.414703e+13 65 18 0.9999996  
## 8 1.440424e+13 98 8 0.8811044  
## 9 5.497693e+12 33 3 1.0000000  
## 10 3.064942e+12 41 1 1.0000000  
## 11 2.181112e+12 58 0 1.0000000  
## 12 5.303694e+12 33 4 1.0000000  
## 13 5.106766e+14 468 423 0.0000000  
## 14 4.453218e+12 119 2 0.1528882

#Use ggpairs to determine which informative information from the importance frame  
plot\_importance\_ggpairs(importance\_frame\_wetdollars) #the two LEAST negative correlations in top row are the winners (?)



# pvalue used to determine whether the actual number of splits using this variable exceeds what would be expect if the splits were random.  
plot\_multi\_way\_importance(importance\_frame\_wetdollars, x\_measure = "mse\_increase", y\_measure =   
 "node\_purity\_increase", size\_measure = "p\_value", no\_of\_labels = 5, main = "PA Wet Dollars") #top 5 variables are labeled

## Warning: Using alpha for a discrete variable is not advised.

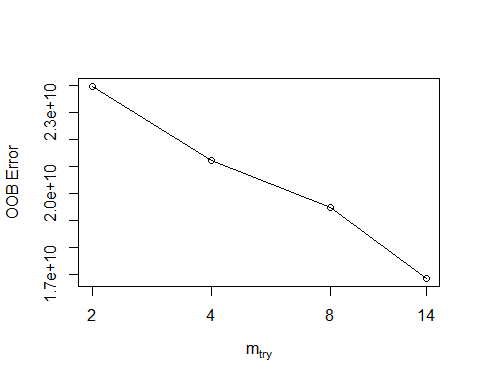
 ## Dry Acres

# Split into trainning, validation, and test sets  
set.seed(25)  
assignment <- sample(1:3, size = nrow(DryAcres), prob = c(0.7, 0.15, 0.15), replace = TRUE)  
  
Drytrain <- DryAcres[assignment == 1,]  
Dryvalid <- DryAcres[assignment == 2,]  
Drytest <- DryAcres[assignment == 3,]  
  
#summary(Drytrain)  
#summary(Dryvalid)  
#summary(Drytest)  
  
  
Mod3 <- randomForest(DryAcres ~ .,   
 data = Drytrain,   
 ntree = 500,   
 #method = "anova",   
 importance = TRUE)  
  
#print(Mod3) # % of variance expalined is low. Tuning needed  
#summary(Mod3)  
#plot(Mod3)  
  
pred3 <- predict(object = Mod3, newdata = Drytest)  
RMSE\_Mod3 <- rmse(actual = Drytest$DryAcres, #actual values  
 predicted = pred3) #predicted values  
print(RMSE\_Mod3/mean(Drytest$DryAcres)) #tells us the %of the mean represented by RMSE. AKA "coefficient of variation"

## [1] 21.28634

# Tune mtry using OOB error  
set.seed(25)  
#train\_pred <- predict(object = Mod1, newdata = PAtrain)  
res3 <- tuneRF(x = Drytrain,  
 y = Drytrain$DryAcres,  
 ntree = 500,  
 stepfactor = 0.5,  
 doBest=TRUE, # Returns a random forest model with optimal mtry value  
 importance = TRUE)

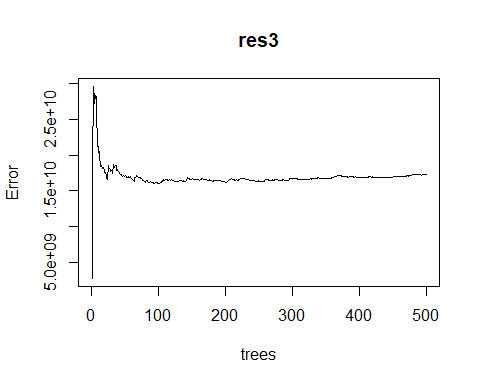
## mtry = 4 OOB error = 21222279369   
## Searching left ...  
## mtry = 2 OOB error = 23958522393   
## -0.1289326 0.05   
## Searching right ...  
## mtry = 8 OOB error = 19481921556   
## 0.08200617 0.05   
## mtry = 14 OOB error = 16839863208   
## 0.1356159 0.05



#localImp = TRUE)  
print(res3)

##   
## Call:  
## randomForest(x = x, y = y, mtry = res[which.min(res[, 2]), 1], importance = TRUE, stepfactor = 0.5)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 14  
##   
## Mean of squared residuals: 17259836561  
## % Var explained: 26

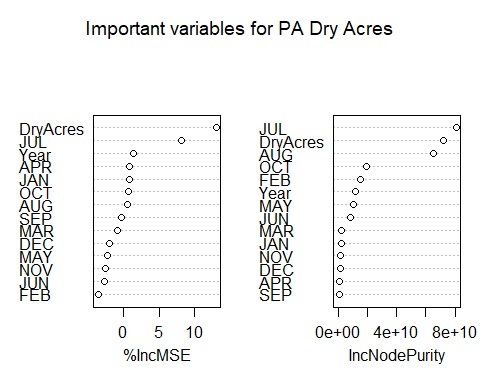
plot(res3) # looks pretty choppy?



res3$importance

## %IncMSE IncNodePurity  
## Year 335052509 11651527305  
## DryAcres 7781871220 72001264204  
## JAN 55217636 2238373424  
## FEB -875998628 15191583429  
## MAR -49582720 2260928707  
## APR 94012095 951436832  
## MAY -518386443 10343846827  
## JUN -582150327 8219079993  
## JUL 3190619183 80691021211  
## AUG 42331475 64913378398  
## SEP -8969663 786624804  
## OCT 127465501 19534069982  
## NOV -141829532 1249809402  
## DEC -103873276 1218211609

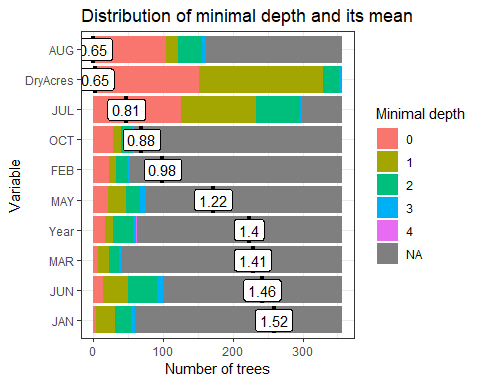
varImpPlot(res3, main = "Important variables for PA Dry Acres")



# using randomForestExplainer  
min\_depth\_frame\_dryacres <- min\_depth\_distribution(res3)

## Warning: Factor `split var` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`

plot\_min\_depth\_distribution(min\_depth\_frame\_dryacres, mean\_sample = "relevant\_trees")



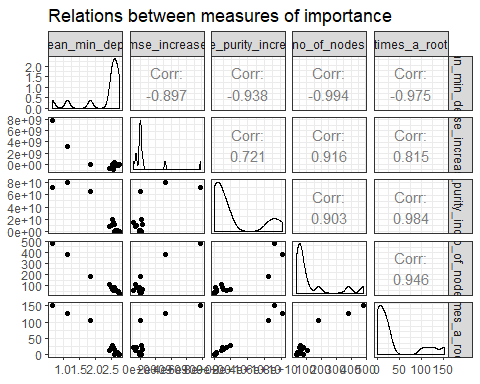
importance\_frame\_dryacres <- measure\_importance(res3)

## Warning: Factor `split var` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`  
  
## Warning: Factor `split var` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`

print(importance\_frame\_dryacres)

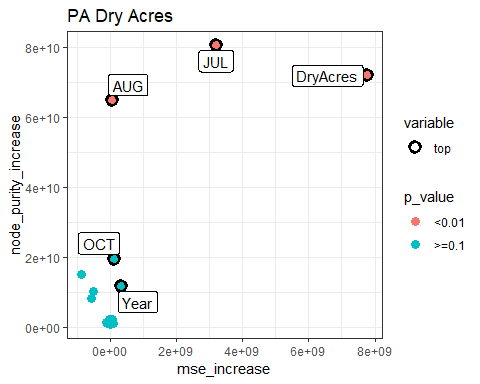
## variable mean\_min\_depth no\_of\_nodes mse\_increase node\_purity\_increase  
## 1 APR 2.7241798 42 94012095 951436832  
## 2 AUG 1.8414944 181 42331475 64913378398  
## 3 DEC 2.7091461 47 -103873276 1218211609  
## 4 DryAcres 0.6544944 479 7781871220 72001264204  
## 5 FEB 2.5667191 56 -875998628 15191583429  
## 6 JAN 2.6153034 61 55217636 2238373424  
## 7 JUL 1.1313652 382 3190619183 80691021211  
## 8 JUN 2.4509213 108 -582150327 8219079993  
## 9 MAR 2.6740730 43 -49582720 2260928707  
## 10 MAY 2.4933708 81 -518386443 10343846827  
## 11 NOV 2.5935843 80 -141829532 1249809402  
## 12 OCT 2.5188876 61 127465501 19534069982  
## 13 SEP 2.7486292 33 -8969663 786624804  
## 14 Year 2.5829607 66 335052509 11651527305  
## no\_of\_trees times\_a\_root p\_value  
## 1 40 1 1.000000e+00  
## 2 162 104 1.868762e-07  
## 3 44 2 1.000000e+00  
## 4 356 152 2.231913e-149  
## 5 52 23 1.000000e+00  
## 6 60 4 1.000000e+00  
## 7 299 126 1.508232e-87  
## 8 100 14 9.269313e-01  
## 9 41 7 1.000000e+00  
## 10 76 21 9.999867e-01  
## 11 74 0 9.999917e-01  
## 12 58 28 1.000000e+00  
## 13 32 1 1.000000e+00  
## 14 63 17 1.000000e+00

#Use ggpairs to determine which informative information from the importance frame  
plot\_importance\_ggpairs(importance\_frame\_dryacres) #the two LEAST negative correlations in top row are the winners (?)



# pvalue used to determine whether the actual number of splits using this variable exceeds what would be expect if the splits were random.  
plot\_multi\_way\_importance(importance\_frame\_dryacres, x\_measure = "mse\_increase", y\_measure =   
 "node\_purity\_increase", size\_measure = "p\_value", no\_of\_labels = 5, main = "PA Dry Acres") #top 5 variables are labeled

## Warning: Using alpha for a discrete variable is not advised.



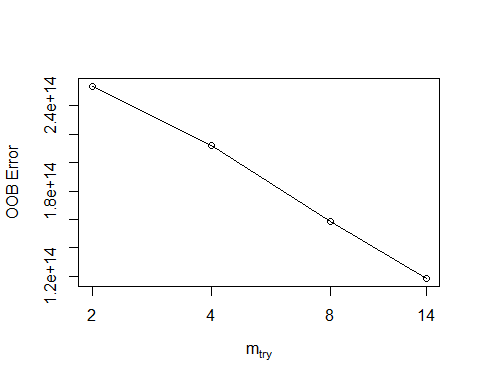
## Dry Dollars

# Split into trainning, validation, and test sets  
set.seed(25)  
assignment <- sample(1:3, size = nrow(DryDollars), prob = c(0.7, 0.15, 0.15), replace = TRUE)  
  
Drytrain2 <- DryDollars[assignment == 1,]  
Dryvalid2 <- DryDollars[assignment == 2,]  
Drytest2 <- DryDollars[assignment == 3,]  
  
#summary(Drytrain2)  
#summary(Dryvalid2)  
#summary(Drytest2)  
  
  
Mod4 <- randomForest(DryDollars ~ .,   
 data = Drytrain2,   
 ntree = 500,   
 #method = "anova",   
 importance = TRUE)  
  
#print(Mod4) # % of variance expalined is low. Tuning needed  
#summary(Mod4)  
#plot(Mod4)  
  
pred4 <- predict(object = Mod4, newdata = Drytest2)  
RMSE\_Mod4 <- rmse(actual = Drytest2$DryDollars, #actual values  
 predicted = pred4) #predicted values  
print(RMSE\_Mod4/mean(Drytest2$DryDollars)) #tells us the %of the mean represented by RMSE. AKA "coefficient of variation"

## [1] 33.4995

# Tune mtry using OOB error  
set.seed(25)  
#train\_pred <- predict(object = Mod1, newdata = PAtrain)  
res4 <- tuneRF(x = Drytrain2,  
 y = Drytrain2$DryDollars,  
 ntree = 500,  
 stepfactor = 0.5,  
 doBest=TRUE, # Returns a random forest model with optimal mtry value  
 importance = TRUE)

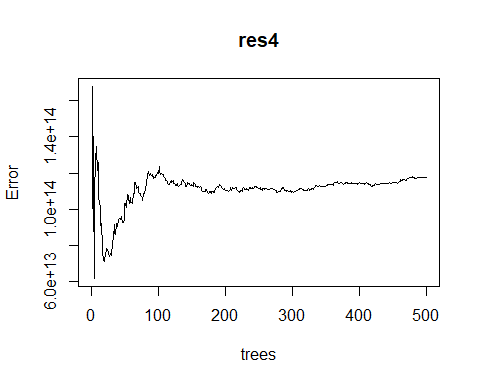
## mtry = 4 OOB error = 2.12258e+14   
## Searching left ...  
## mtry = 2 OOB error = 2.535643e+14   
## -0.1946039 0.05   
## Searching right ...  
## mtry = 8 OOB error = 1.583224e+14   
## 0.254104 0.05   
## mtry = 14 OOB error = 1.185475e+14   
## 0.2512271 0.05



#localImp = TRUE)  
print(res4)

##   
## Call:  
## randomForest(x = x, y = y, mtry = res[which.min(res[, 2]), 1], importance = TRUE, stepfactor = 0.5)   
## Type of random forest: regression  
## Number of trees: 500  
## No. of variables tried at each split: 14  
##   
## Mean of squared residuals: 1.173597e+14  
## % Var explained: 51.09

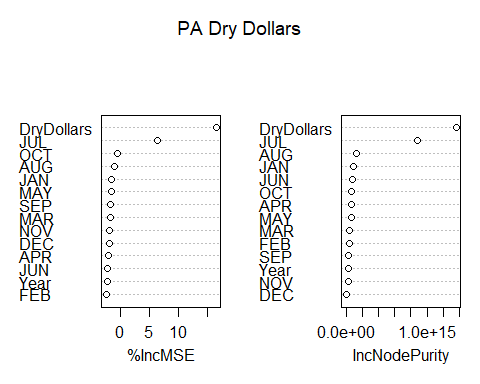
plot(res4) # looks pretty choppy?



res4$importance

## %IncMSE IncNodePurity  
## Year -1.138754e+12 2.680711e+13  
## DryDollars 1.604389e+14 1.362988e+15  
## JAN -2.543257e+12 8.727872e+13  
## FEB -2.939626e+12 3.599853e+13  
## MAR -1.067521e+12 4.274672e+13  
## APR -2.625323e+12 6.829828e+13  
## MAY -2.573209e+12 6.678883e+13  
## JUN -6.380732e+12 7.645203e+13  
## JUL 2.778854e+13 8.828699e+14  
## AUG -1.517649e+12 1.237452e+14  
## SEP -1.137859e+12 2.779162e+13  
## OCT -4.815690e+11 7.021416e+13  
## NOV -9.203381e+11 2.302118e+13  
## DEC -5.387045e+11 9.076889e+12

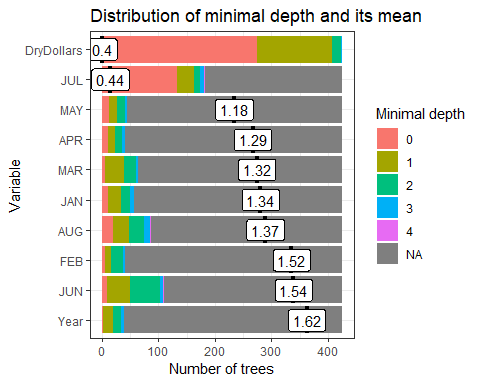
varImpPlot(res4, main = "PA Dry Dollars")



# using randomForestExplainer  
min\_depth\_frame\_drydollars <- min\_depth\_distribution(res4)

## Warning: Factor `split var` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`

plot\_min\_depth\_distribution(min\_depth\_frame\_drydollars, mean\_sample = "relevant\_trees")



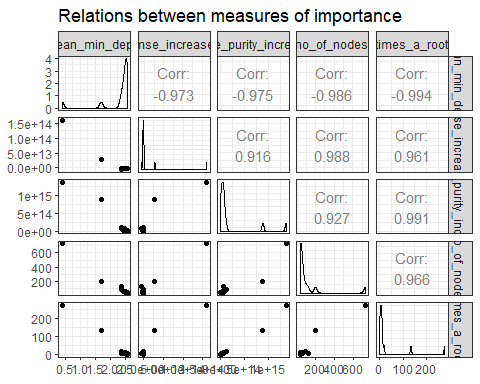
importance\_frame\_drydollars <- measure\_importance(res4)

## Warning: Factor `split var` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`  
  
## Warning: Factor `split var` contains implicit NA, consider using  
## `forcats::fct\_explicit\_na`

print(importance\_frame\_drydollars)

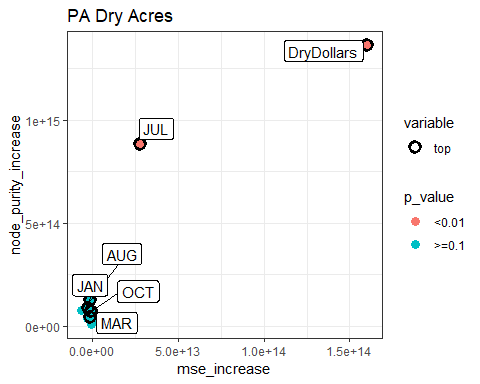
## variable mean\_min\_depth no\_of\_nodes mse\_increase  
## 1 APR 2.5262871 42 -2.625323e+12  
## 2 AUG 2.3938918 92 -1.517649e+12  
## 3 DEC 2.5748329 38 -5.387045e+11  
## 4 DryDollars 0.3952941 730 1.604389e+14  
## 5 FEB 2.5513647 42 -2.939626e+12  
## 6 JAN 2.4842400 57 -2.543257e+12  
## 7 JUL 1.7040847 201 2.778854e+13  
## 8 JUN 2.3676941 122 -6.380732e+12  
## 9 MAR 2.4592847 64 -1.067521e+12  
## 10 MAY 2.5051718 46 -2.573209e+12  
## 11 NOV 2.5834400 37 -9.203381e+11  
## 12 OCT 2.5235576 64 -4.815690e+11  
## 13 SEP 2.5679576 46 -1.137859e+12  
## 14 Year 2.5623247 39 -1.138754e+12  
## node\_purity\_increase no\_of\_trees times\_a\_root p\_value  
## 1 6.829828e+13 41 11 1.000000e+00  
## 2 1.237452e+14 87 20 9.919000e-01  
## 3 9.076889e+12 37 0 1.000000e+00  
## 4 1.362988e+15 425 275 0.000000e+00  
## 5 3.599853e+13 40 5 1.000000e+00  
## 6 8.727872e+13 56 10 1.000000e+00  
## 7 8.828699e+14 183 133 3.796111e-14  
## 8 7.645203e+13 110 9 2.851480e-01  
## 9 4.274672e+13 63 6 1.000000e+00  
## 10 6.678883e+13 44 12 1.000000e+00  
## 11 2.302118e+13 36 4 1.000000e+00  
## 12 7.021416e+13 61 9 1.000000e+00  
## 13 2.779162e+13 46 5 1.000000e+00  
## 14 2.680711e+13 39 1 1.000000e+00

#Use ggpairs to determine which informative information from the importance frame  
plot\_importance\_ggpairs(importance\_frame\_drydollars) #the two LEAST negative correlations in top row are the winners (?)



# pvalue used to determine whether the actual number of splits using this variable exceeds what would be expect if the splits were random.  
plot\_multi\_way\_importance(importance\_frame\_drydollars, x\_measure = "mse\_increase", y\_measure =   
 "node\_purity\_increase", size\_measure = "p\_value", no\_of\_labels = 5, main = "PA Dry Acres") #top 5 variables are labeled

## Warning: Using alpha for a discrete variable is not advised.



End of Script