Module4\_Assignment2

drug <- read\_csv("drug\_data-1.csv")

names(drug) = c("ID", "Age", "Gender", "Education", "Country", "Ethnicity","Nscore", "Escore", "Oscore", "Ascore", "Cscore", "Impulsive","SS", "Alcohol", "Amphet", "Amyl", "Benzos", "Caff", "Cannabis","Choc", "Coke", "Crack", "Ecstasy", "Heroin", "Ketamine", "Legalh","LSD", "Meth", "Mushrooms", "Nicotine", "Semer", "VSA")  
  
#str(drug)

drug[drug == "CL0"] = "No"  
drug[drug == "CL1"] = "No"  
drug[drug == "CL2"] = "Yes"  
drug[drug == "CL3"] = "Yes"  
drug[drug == "CL4"] = "Yes"  
drug[drug == "CL5"] = "Yes"  
drug[drug == "CL6"] = "Yes"

drug\_clean = drug %>%   
 mutate\_at(vars(Age:Ethnicity), funs(as\_factor)) %>%  
 mutate(Age = factor(Age, labels = c("18\_24", "25\_34", "35\_44","45\_54", "55\_64", "65\_"))) %>%  
 mutate(Gender = factor(Gender, labels = c("Male", "Female"))) %>%  
 mutate(Education = factor(Education, labels =c("Under16", "At16", "At17", "At18", "SomeCollege","ProfessionalCert", "Bachelors", "Masters", "Doctorate"))) %>%mutate(Country = factor(Country,labels = c("USA", "NewZealand", "Other", "Australia","Ireland","Canada","UK"))) %>%  
 mutate(Ethnicity = factor(Ethnicity,labels = c("Black", "Asian", "White", "White/Black", "Other","White/Asian", "Black/Asian"))) %>%  
 mutate\_at(vars(Alcohol:VSA), funs(as\_factor)) %>%select(-ID)

#str(drug\_clean)

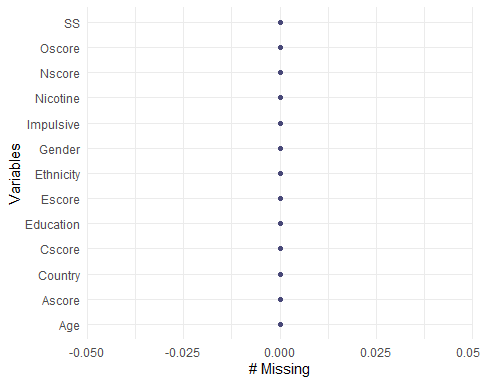
drug\_clean = drug\_clean %>%   
 select(!(Alcohol:Mushrooms)) %>%   
 select(!(Semer:VSA))  
names(drug\_clean)

## [1] "Age" "Gender" "Education" "Country" "Ethnicity" "Nscore"   
## [7] "Escore" "Oscore" "Ascore" "Cscore" "Impulsive" "SS"   
## [13] "Nicotine"

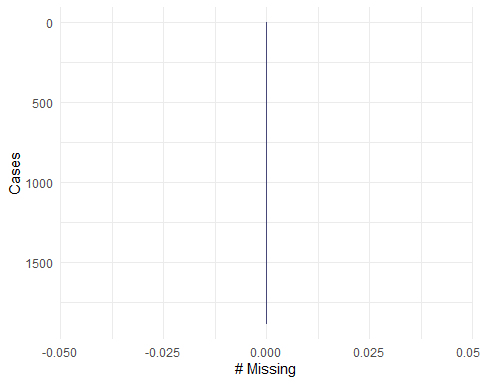
## Task 1

#str(drug\_clean)  
#summary(drug\_clean)  
#skim(drug\_clean)

gg\_miss\_var(drug\_clean)



gg\_miss\_case(drug\_clean)



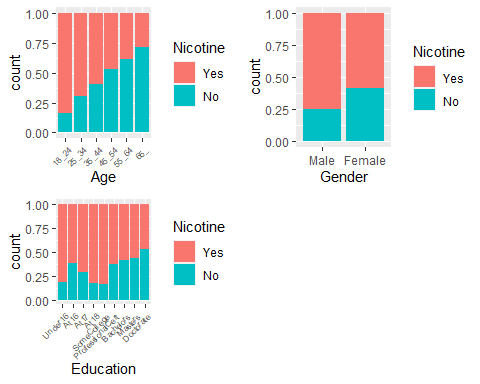
After using several methods to check for missingness in the drug\_clean data set, there does not appear to be any missing data.

## Task 2

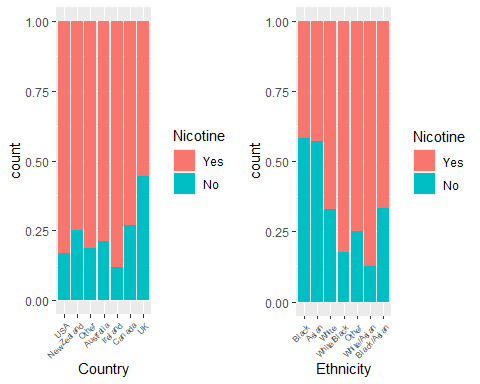
set.seed(1234)  
drug\_clean\_split = initial\_split(drug\_clean, prob = 0.70, strata = Nicotine)  
train = training(drug\_clean\_split)  
test = testing(drug\_clean\_split)

## Task 3

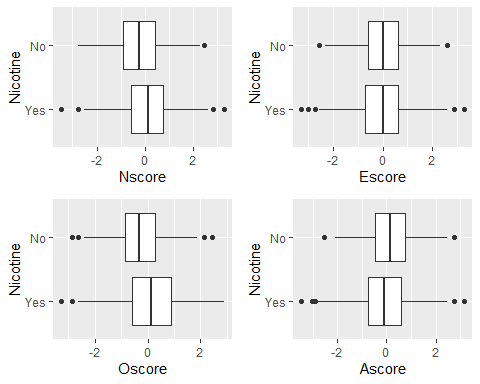
p1 = ggplot(train, aes(x = Age, fill = Nicotine)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(size=6, angle = 45, vjust = 1, hjust=1))  
p2 = ggplot(train, aes(x = Gender, fill = Nicotine)) + geom\_bar(position = "fill")  
p3 = ggplot(train, aes(x = Education, fill = Nicotine)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(size=6, angle = 45, vjust = 1, hjust=1))  
  
grid.arrange(p1,p2,p3, ncol = 2)



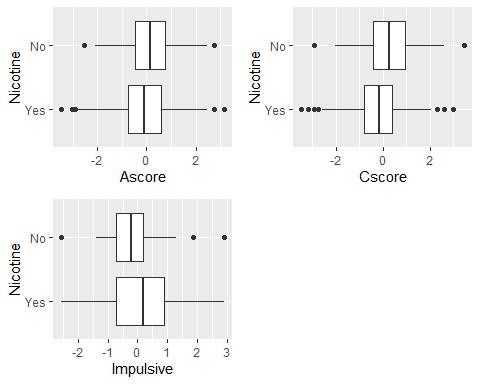
p4 = ggplot(train, aes(x = Country, fill = Nicotine)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(size=6,angle = 45, vjust = 1, hjust=1))  
p5 = ggplot(train, aes(x = Ethnicity, fill = Nicotine)) + geom\_bar(position = "fill") + theme(axis.text.x = element\_text(size=6,angle = 45, vjust = 1, hjust=1))  
  
grid.arrange(p4,p5, ncol = 2)



p6=ggplot(train, aes(x = Nscore, y = Nicotine)) + geom\_boxplot()  
p7=ggplot(train, aes(x = Escore, y = Nicotine)) + geom\_boxplot()  
p8=ggplot(train, aes(x = Oscore, y = Nicotine)) + geom\_boxplot()  
p9=ggplot(train, aes(x = Ascore, y = Nicotine)) + geom\_boxplot()  
  
grid.arrange(p6,p7,p8,p9, ncol = 2)



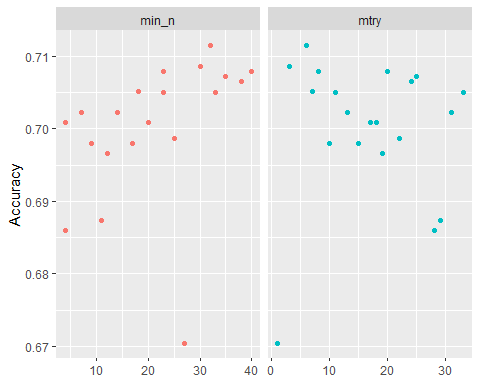
p10=ggplot(train, aes(x = Cscore, y = Nicotine)) + geom\_boxplot()  
p11=ggplot(train, aes(x = Impulsive, y = Nicotine)) + geom\_boxplot()  
p12=ggplot(train, aes(x = SS, y = Nicotine)) + geom\_boxplot()  
  
grid.arrange(p9,p10,p11, ncol = 2)

 ## Task 4

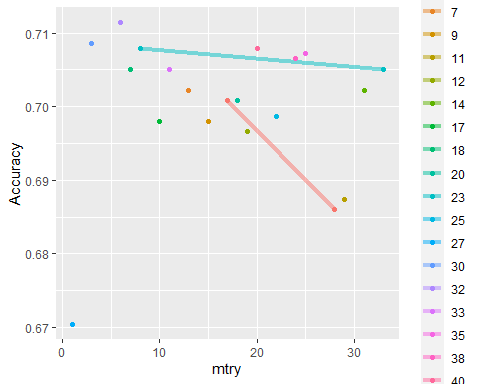
set.seed(123)  
rf\_folds = vfold\_cv(train, v = 5)

drug\_clean\_recipe = recipe(Nicotine ~., train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
rf\_model = rand\_forest(mtry = tune(), min\_n = tune(), trees = 100) %>%   
 set\_engine("ranger", importance = "permutation") %>%   
 set\_mode("classification")  
  
drug\_clean\_wflow =   
 workflow() %>%   
 add\_model(rf\_model) %>%   
 add\_recipe(drug\_clean\_recipe)  
  
rf\_grid = grid\_regular(  
 mtry(range = c(2, 8)),   
 min\_n(range = c(5, 20)),   
 levels = 10)  
  
set.seed(123)  
rf\_res\_tuned = tune\_grid(  
 drug\_clean\_wflow,  
 resamples = rf\_folds,  
 grid = 20)

rf\_res\_tuned %>%  
 collect\_metrics() %>%  
 filter(.metric == "accuracy") %>%  
 select(mean, min\_n, mtry) %>%  
 pivot\_longer(min\_n:mtry,  
 values\_to = "value",  
 names\_to = "parameter"  
 ) %>%  
 ggplot(aes(value, mean, color = parameter)) +  
 geom\_point(show.legend = FALSE) +  
 facet\_wrap(~parameter, scales = "free\_x") +  
 labs(x = NULL, y = "Accuracy")



rf\_res\_tuned %>%  
 collect\_metrics() %>%  
 filter(.metric == "accuracy") %>%  
 mutate(min\_n = factor(min\_n)) %>%  
 ggplot(aes(mtry, mean, color = min\_n)) +  
 geom\_line(alpha = 0.5, size = 1.5) +  
 geom\_point() +  
 labs(y = "Accuracy")

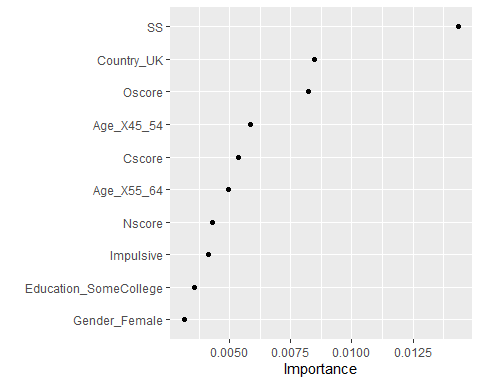


## Task 5

best\_rf = select\_best(rf\_res\_tuned, "accuracy")  
  
final\_rf = finalize\_workflow(drug\_clean\_wflow,  
 best\_rf)  
  
final\_rf

## == Workflow ====================================================================  
## Preprocessor: Recipe  
## Model: rand\_forest()  
##   
## -- Preprocessor ----------------------------------------------------------------  
## 1 Recipe Step  
##   
## \* step\_dummy()  
##   
## -- Model -----------------------------------------------------------------------  
## Random Forest Model Specification (classification)  
##   
## Main Arguments:  
## mtry = 6  
## trees = 100  
## min\_n = 32  
##   
## Engine-Specific Arguments:  
## importance = permutation  
##   
## Computational engine: ranger

final\_rf\_fit = fit(final\_rf, train)  
  
final\_rf\_fit %>% pull\_workflow\_fit() %>% vip(geom = "point")



The two greatest variables that predict nicotine usage are SS (a measure of sensation seeking) and Oscore ( a measure of openess to experiences).

## Task 6

trainpredrf = predict(final\_rf\_fit, train)  
head(trainpredrf)

## # A tibble: 6 x 1  
## .pred\_class  
## <fct>   
## 1 Yes   
## 2 Yes   
## 3 Yes   
## 4 No   
## 5 Yes   
## 6 Yes

confusionMatrix(trainpredrf$.pred\_class, train$Nicotine,   
 positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 908 163  
## No 40 303  
##   
## Accuracy : 0.8564   
## 95% CI : (0.8371, 0.8743)  
## No Information Rate : 0.6704   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.6518   
##   
## Mcnemar's Test P-Value : < 2.2e-16   
##   
## Sensitivity : 0.9578   
## Specificity : 0.6502   
## Pos Pred Value : 0.8478   
## Neg Pred Value : 0.8834   
## Prevalence : 0.6704   
## Detection Rate : 0.6421   
## Detection Prevalence : 0.7574   
## Balanced Accuracy : 0.8040   
##   
## 'Positive' Class : Yes   
##

testpredrf = predict(final\_rf\_fit, test)  
head(testpredrf)

## # A tibble: 6 x 1  
## .pred\_class  
## <fct>   
## 1 Yes   
## 2 No   
## 3 No   
## 4 Yes   
## 5 Yes   
## 6 No

confusionMatrix(testpredrf$.pred\_class, test$Nicotine,   
 positive = "Yes")

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 281 90  
## No 35 65  
##   
## Accuracy : 0.7346   
## 95% CI : (0.6923, 0.774)  
## No Information Rate : 0.6709   
## P-Value [Acc > NIR] : 0.00164   
##   
## Kappa : 0.3393   
##   
## Mcnemar's Test P-Value : 1.366e-06   
##   
## Sensitivity : 0.8892   
## Specificity : 0.4194   
## Pos Pred Value : 0.7574   
## Neg Pred Value : 0.6500   
## Prevalence : 0.6709   
## Detection Rate : 0.5966   
## Detection Prevalence : 0.7877   
## Balanced Accuracy : 0.6543   
##   
## 'Positive' Class : Yes   
##

## Task 7

The model would have real world applications when studying the best way to deter nicotine usage based on the metrics for the various variables such as “sensation seeking” or “openness to new experiences.” In its current state though the accuracy rate between the training set at .8564 and testing set at .7346 is probably too great for use in the rea world. The model should be modified to include a greater number of trees in our random forest to determine a better fit.