drug <- read\_csv("~/Documents/MBA\_Predictive/Week 4/drug\_data-1.csv")

##   
## ── Column specification ────────────────────────────────────────────────────────  
## cols(  
## .default = col\_character(),  
## Column1 = col\_double(),  
## Column2 = col\_double(),  
## Column3 = col\_double(),  
## Column4 = col\_double(),  
## Column5 = col\_double(),  
## Column6 = col\_double(),  
## Column7 = col\_double(),  
## Column8 = col\_double(),  
## Column9 = col\_double(),  
## Column10 = col\_double(),  
## Column11 = col\_double(),  
## Column12 = col\_double(),  
## Column13 = col\_double()  
## )  
## ℹ Use `spec()` for the full column specifications.

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)

## Warning: `funs()` is deprecated as of dplyr 0.8.0.  
## Please use a list of either functions or lambdas:   
##   
## # Simple named list:   
## list(mean = mean, median = median)  
##   
## # Auto named with `tibble::lst()`:   
## tibble::lst(mean, median)  
##   
## # Using lambdas  
## list(~ mean(., trim = .2), ~ median(., na.rm = TRUE))  
## This warning is displayed once every 8 hours.  
## Call `lifecycle::last\_warnings()` to see where this warning was generated.

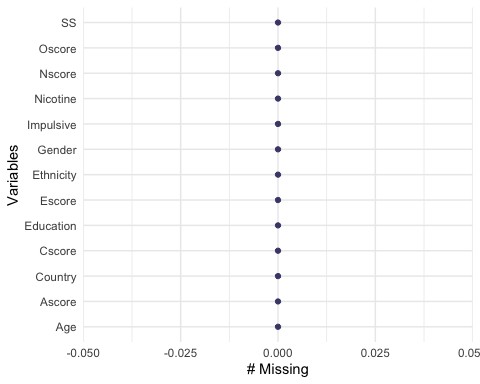
#str(drug\_clean)

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

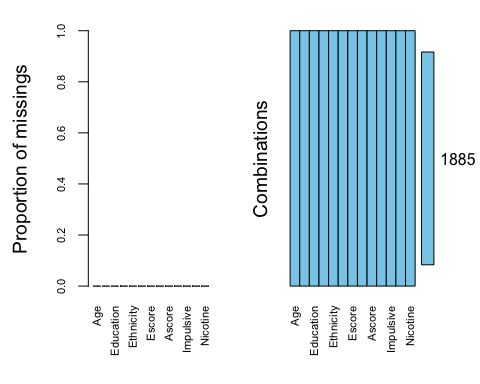
## [1] "Age" "Gender" "Education" "Country" "Ethnicity" "Nscore"   
## [7] "Escore" "Oscore" "Ascore" "Cscore" "Impulsive" "SS"   
## [13] "Nicotine"

# Task 1

gg\_miss\_var(drug\_clean)



vim\_plot = aggr(drug\_clean, numbers = TRUE, prop = c(TRUE, FALSE),cex.axis=.7)

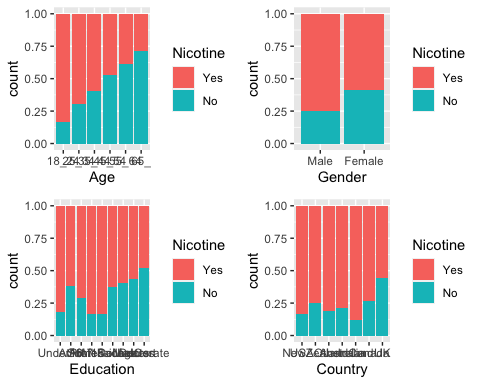


# Task 2

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

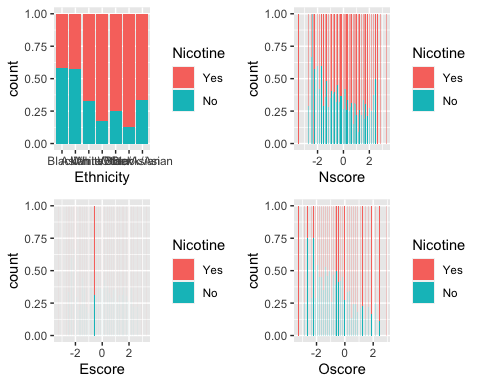
# Task 3

p1 = ggplot(train, aes(x = Age, fill = Nicotine)) + geom\_bar(position = "fill")  
p2 = ggplot(train, aes(x = Gender, fill = Nicotine)) + geom\_bar(position = "fill")  
p3 = ggplot(train, aes(x = Education, fill = Nicotine)) + geom\_bar(position = "fill")  
p4 = ggplot(train, aes(x = Country, fill = Nicotine)) + geom\_bar(position = "fill")  
grid.arrange(p1,p2,p3,p4)



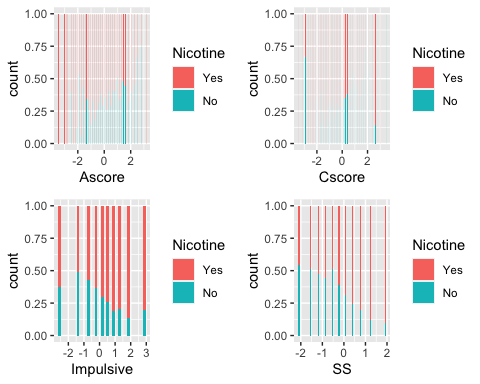
Age- As age increases, Nicotine drecreases  
Gender- More males than females use Nicotine  
Education- There is not a direct correlation between Education and Nicotine  
Country- There is not a direct correlation between Country and Nicotine

p5 = ggplot(train, aes(x = Ethnicity, fill = Nicotine)) + geom\_bar(position = "fill")  
p6 = ggplot(train, aes(x = Nscore, fill = Nicotine)) + geom\_bar(position = "fill")  
p7 = ggplot(train, aes(x = Escore, fill = Nicotine)) + geom\_bar(position = "fill")  
p8 = ggplot(train, aes(x = Oscore, fill = Nicotine)) + geom\_bar(position = "fill")  
grid.arrange(p5,p6,p7,p8)



Ethnicity- Some ethnicities use Nicotine more than others. Ex. the White ethnicity uses more Nicotine than the Black ethnicity  
Nscore- There is no relationship between Nscore and Nicotine  
Escore- As Escore increases, Nicotine increases  
Oscore- As Oscore increases, Nicotine Increases

p9 = ggplot(train, aes(x = Ascore, fill = Nicotine)) + geom\_bar(position = "fill")  
p10 = ggplot(train, aes(x = Cscore, fill = Nicotine)) + geom\_bar(position = "fill")  
p11 = ggplot(train, aes(x = Impulsive, fill = Nicotine)) + geom\_bar(position = "fill")  
p12 = ggplot(train, aes(x = SS, fill = Nicotine)) + geom\_bar(position = "fill")  
grid.arrange(p9,p10,p11,p12)



Ascore- As Ascore increases, Nicotine increases  
Cscore- As Csocre increases, Nicotine decreases  
Impulsive- As impulsiviness increases, Nicotine increases  
SS- As SS increases, Nicotine increases

# Task 4

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

drug\_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\_wflow =   
 workflow() %>%   
 add\_model(rf\_model) %>%   
 add\_recipe(drug\_recipe)  
  
rf\_grid = grid\_regular(  
 mtry(range = c(2, 8)),   
 min\_n(range = c(5, 20)),   
 levels = 10)  
  
set.seed(123)  
rf\_res = tune\_grid(  
 drug\_wflow,  
 resamples = rf\_folds,  
 grid = 10)

## i Creating pre-processing data to finalize unknown parameter: mtry

##   
## Attaching package: 'rlang'

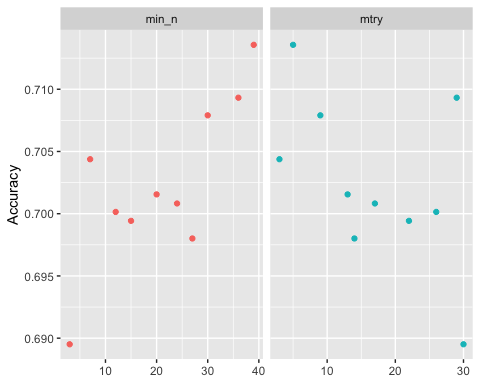
## The following objects are masked from 'package:purrr':  
##   
## %@%, as\_function, flatten, flatten\_chr, flatten\_dbl, flatten\_int,  
## flatten\_lgl, flatten\_raw, invoke, list\_along, modify, prepend,  
## splice

##   
## Attaching package: 'vctrs'

## The following object is masked from 'package:dplyr':  
##   
## data\_frame

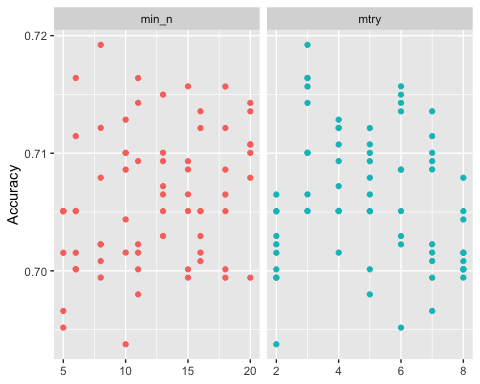
## The following object is masked from 'package:tibble':  
##   
## data\_frame

rf\_res %>%  
 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")

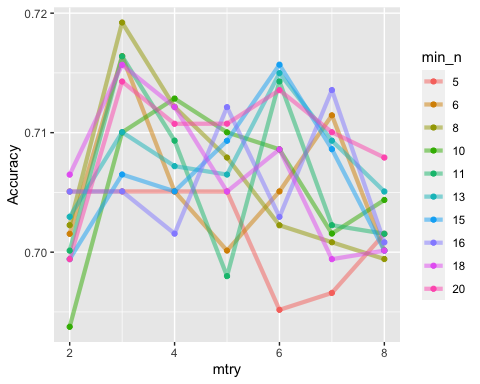


drug\_recipe = recipe(Nicotine ~., train) %>%  
 step\_dummy(all\_nominal(), -all\_outcomes())  
  
rf\_model = rand\_forest(mtry = tune(), min\_n = tune(), trees = 100) %>%  
 set\_engine("ranger", importance = "permutation") %>%   
 set\_mode("classification")  
  
drug\_wflow =   
 workflow() %>%   
 add\_model(rf\_model) %>%   
 add\_recipe(drug\_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\_wflow,  
 resamples = rf\_folds,  
 grid = rf\_grid  
)

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")



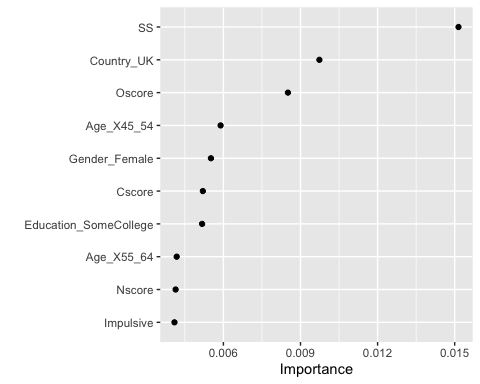
# Task 5

best\_rf = select\_best(rf\_res\_tuned, "accuracy")  
  
final\_rf = finalize\_workflow(  
 drug\_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 = 3  
## trees = 100  
## min\_n = 8  
##   
## Engine-Specific Arguments:  
## importance = permutation  
##   
## Computational engine: ranger

final\_rf\_fit = fit(final\_rf, train)

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



SS scores the highest in importance for this model. This is followed by Country\_UK and Oscore.

# 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 Yes   
## 5 Yes   
## 6 Yes

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

## Confusion Matrix and Statistics  
##   
## Reference  
## Prediction Yes No  
## Yes 933 103  
## No 15 363  
##   
## Accuracy : 0.9165   
## 95% CI : (0.9009, 0.9304)  
## No Information Rate : 0.6704   
## P-Value [Acc > NIR] : < 2.2e-16   
##   
## Kappa : 0.8016   
##   
## Mcnemar's Test P-Value : 1.156e-15   
##   
## Sensitivity : 0.9842   
## Specificity : 0.7790   
## Pos Pred Value : 0.9006   
## Neg Pred Value : 0.9603   
## Prevalence : 0.6704   
## Detection Rate : 0.6598   
## Detection Prevalence : 0.7327   
## Balanced Accuracy : 0.8816   
##   
## '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 283 92  
## No 33 63  
##   
## Accuracy : 0.7346   
## 95% CI : (0.6923, 0.774)  
## No Information Rate : 0.6709   
## P-Value [Acc > NIR] : 0.00164   
##   
## Kappa : 0.3345   
##   
## Mcnemar's Test P-Value : 2.129e-07   
##   
## Sensitivity : 0.8956   
## Specificity : 0.4065   
## Pos Pred Value : 0.7547   
## Neg Pred Value : 0.6562   
## Prevalence : 0.6709   
## Detection Rate : 0.6008   
## Detection Prevalence : 0.7962   
## Balanced Accuracy : 0.6510   
##   
## 'Positive' Class : Yes   
##

The train set is 91.65% accurate. The test set is 73.46% accurate.

# Task 7

This model could be used in schools to educate children on the harsh affects of drugs. The model would help identify students who are more likely to use drugs than others. This could help prevent drug users in the future.I would recommend this model or real-world use. The training set is 91.65% accurate and the test set is 73.46% accurate. Although the sets are not 100% accurate, it does give a good picture of who is using what drugs. My concern with the model is that it does incorrectly classify some groups making it harder to truly classify the data.