

705604096_stats101c_hw4

Jade Gregory

2023-10-30

Question 1

```
kagtrain <- read.csv("TrainSAData2.csv")
kagtest <- read.csv("TestSAData2NoY.csv")
head(kagtrain)
```

```
##   ID    sex age height weight waistline sight_left sight_right hear_left
## 1  1  Male  75   160    NA         NA         NA         0.7   Normal
## 2  2 Female  50   160    60       74.0       1.0       1.2   Normal
## 3  3  Male  65   170    80       95.0       1.0       1.5   Normal
## 4  4  <NA>  65   155    55       81.0       0.3       0.4  Abnormal
## 5  5  Male  35   160    60       85.0       1.0       1.0   Normal
## 6  6 Female  50   160    70       73.2       0.3       0.4   Normal
##   hear_right SBP DBP BLDS tot_chole HDL_chole LDL_chole triglyceride hemoglobin
## 1   Normal  NA  76  136    215      33      143      193      15.0
## 2   Normal 118  70  125    207      85       NA      110      13.3
## 3   Normal 149  83  130    115      48      33      170      16.4
## 4  Abnormal 118  67  97    171      65      67      195      13.9
## 5   Normal  96  62  78    114      42      58       72      16.0
## 6   Normal 119  79  220    178      61      80      181      10.5
##   urine_protein serum_creatinine SGOT_AST SGOT_ALT gamma_GTP      BMI
## 1             3             0.9      28      23      36 23.43750
## 2             1             0.6      28      19      22 23.43750
## 3             1             1.4      41      64      53      NA
## 4             1             0.8      26      25      NA 22.89282
## 5             1             1.0      17      24      34      NA
## 6             1             0.5      36      NA      20 27.34375
##   BMI.Category AGE.Category Smoking.Status Alcoholic.Status
## 1   Healthy      Very Old   Still Smoking                Y
## 2    <NA>      Mid-aged   Never Smoked                  Y
## 3  Overweight      Old     Still Smoking                Y
## 4    <NA>      Old     Never Smoked                   N
## 5   Healthy      Mid-aged   Still Smoking                N
## 6  Overweight      Mid-aged   Never Smoked                  N
```

```
head(kagtest)
```

```
##   ID    sex age height weight waistline sight_left sight_right hear_left
## 1  1  <NA>  40   175    NA         76         1.5       1.2   Normal
```

```
## 2 2 Female 55 150 55 81 1.0 0.9 Normal
## 3 3 Female 35 155 50 73 0.2 0.2 Normal
## 4 4 Female 60 155 50 79 1.0 1.0 Normal
## 5 5 Male 55 165 65 84 NA 0.9 Normal
## 6 6 Male 45 170 55 73 1.5 1.2 Normal
## hear_right SBP DBP BLDS tot_chole HDL_chole LDL_chole triglyceride hemoglobin
## 1 Normal 118 78 89 160 49 75 181 NA
## 2 Normal 89 52 109 240 67 154 95 12.6
## 3 Normal 102 63 86 NA 48 120 63 12.0
## 4 Normal NA 76 97 222 61 140 101 12.9
## 5 Normal 102 63 NA 198 46 112 200 17.1
## 6 Normal 120 80 98 152 NA 55 283 14.5
## urine_protein serum_creatinine SGOT_AST SGOT_ALT gamma_GTP BMI
## 1 1 1.1 18 13 15 22.85714
## 2 1 0.7 47 32 27 24.44444
## 3 1 0.8 14 10 10 20.81165
## 4 1 1.0 33 NA 64 20.81165
## 5 2 0.7 21 33 78 23.87511
## 6 1 1.0 17 25 26 NA
## BMI.Category AGE.Category Smoking.Status
## 1 Healthy Mid-aged Still Smoking
## 2 <NA> Old <NA>
## 3 Healthy Mid-aged <NA>
## 4 Healthy Old Never Smoked
## 5 Healthy Old Never Smoked
## 6 Healthy Mid-aged Still Smoking
```

a)

```
dim(kagtrain)
```

```
## [1] 70000 28
```

```
dim(kagtest)
```

```
## [1] 30000 27
```

The training data set has 28 columns by 70,000 rows. The testing data set has 27 columns by 30,000 rows.

- b) There are 21 numerical predictors. They include the variables ID, age, height, weight, waist-line, sight_left, sight_right, SBP, DBP, BLDS, tot_chole, HDL_chole, LDL_chole, triglyceride, hemoglobin, urine_protein, serum_creatinine, SGOT_AST, SGOT_ALT, gamma_GTP, and BMI.
- c) There are 7 categorical variables. They are sex, hear_left, hear_right, BMI.Category, AGE.Category, Smoking.Status, and Alcoholic.Status.
- d)

```
(sapply(kagtrain, function(x) sum(is.na(x))) / 70000) * 100
```

```
##           ID           sex           age           height
##      0.000000      7.088571      6.967143      7.058571
##      weight      waistline      sight_left      sight_right
##      7.102857      7.057143      6.967143      7.000000
##      hear_left      hear_right      SBP      DBP
##      6.904286      6.981429      7.027143      6.992857
##      BLDS      tot_chole      HDL_chole      LDL_chole
##      6.887143      6.948571      6.880000      7.020000
##      triglyceride      hemoglobin      urine_protein      serum_creatinine
##      6.967143      7.087143      6.998571      6.924286
##      SGOT_AST      SGOT_ALT      gamma_GTP      BMI
##      6.981429      6.990000      7.087143      7.095714
##      BMI.Category      AGE.Category      Smoking.Status      Alcoholic.Status
##      6.962857      11.875714      6.970000      0.000000
```

```
((sapply(kagtest, function(x) sum(is.na(x)))) / 30000) * 100
```

```
##           ID           sex           age           height
##      0.000000      7.296667      7.103333      7.063333
##      weight      waistline      sight_left      sight_right
##      7.003333      7.183333      6.736667      7.146667
##      hear_left      hear_right      SBP      DBP
##      6.840000      6.800000      7.013333      7.056667
##      BLDS      tot_chole      HDL_chole      LDL_chole
##      7.050000      7.203333      7.026667      7.003333
##      triglyceride      hemoglobin      urine_protein      serum_creatinine
##      6.680000      6.896667      6.830000      6.953333
##      SGOT_AST      SGOT_ALT      gamma_GTP      BMI
##      6.933333      7.076667      6.853333      7.200000
##      BMI.Category      AGE.Category      Smoking.Status
##      7.073333      11.730000      7.030000
```

e)

```
length(kagtrain$Alcoholic.Status[kagtrain$Alcoholic.Status == "Y"])
```

```
## [1] 34887
```

```
length(kagtrain$Alcoholic.Status[kagtrain$Alcoholic.Status == "N"])
```

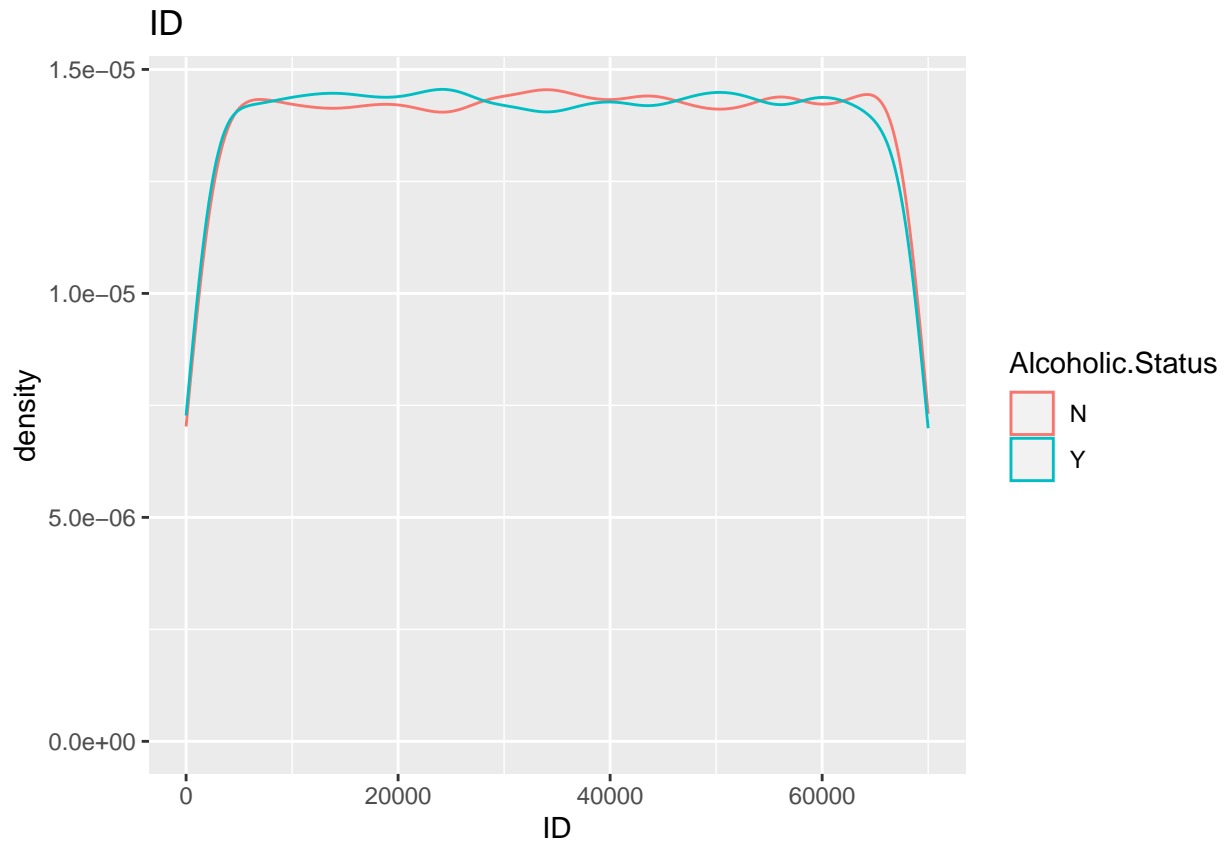
```
## [1] 35113
```

Our response variable is Alcoholic.Status that has two values, yes or no, denotes Y or N. Alcoholic.Status is Y 34887 times out of 70000 observations which is 49.84% and it is N 35113 times out of 70000 observations which is 50.16%. Our max error rate based on our training data is 49.84%.

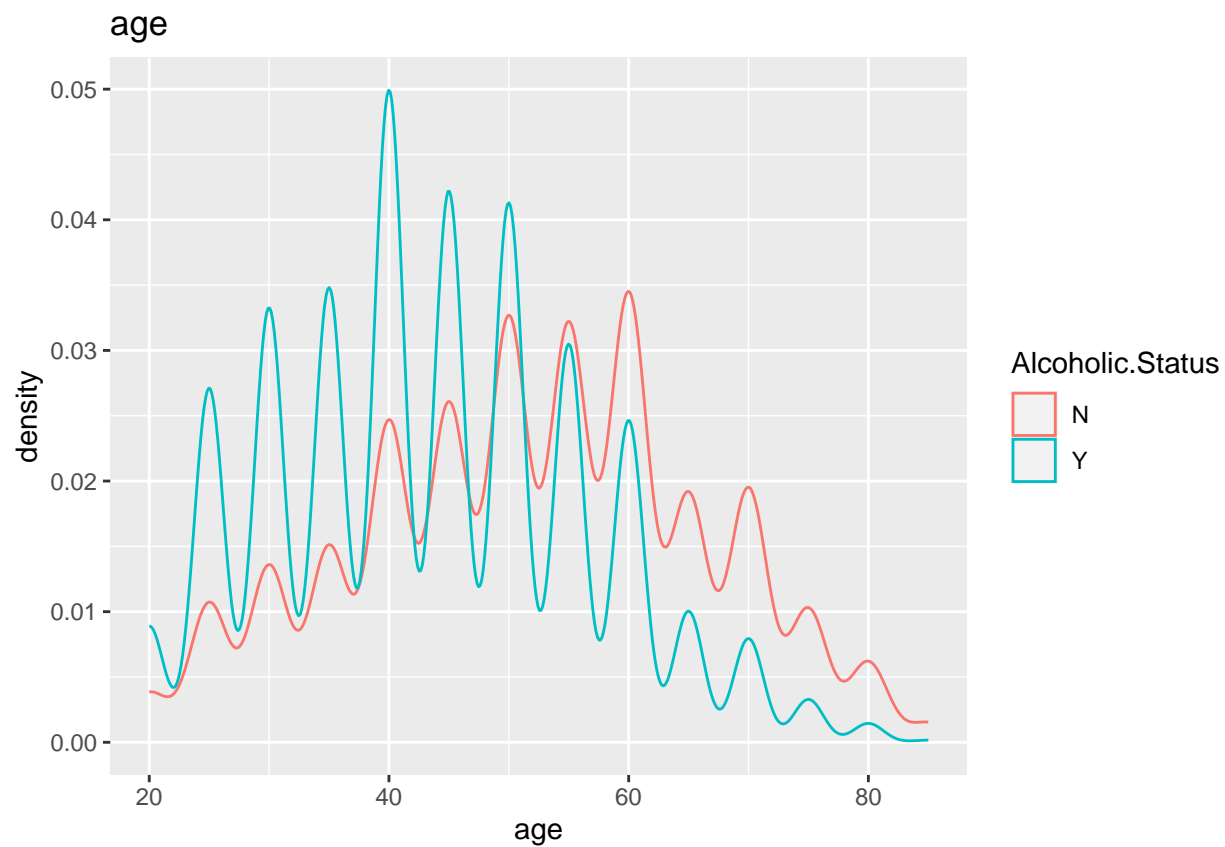
f)

```
num_names <- names(kagtrain[sapply(kagtrain, is.numeric)])
for(variable in num_names){
  plot <- ggplot(kagtrain, aes_string(variable, color = "Alcoholic.Status")) + geom_density() + ggtitle
  print(plot)
}
```

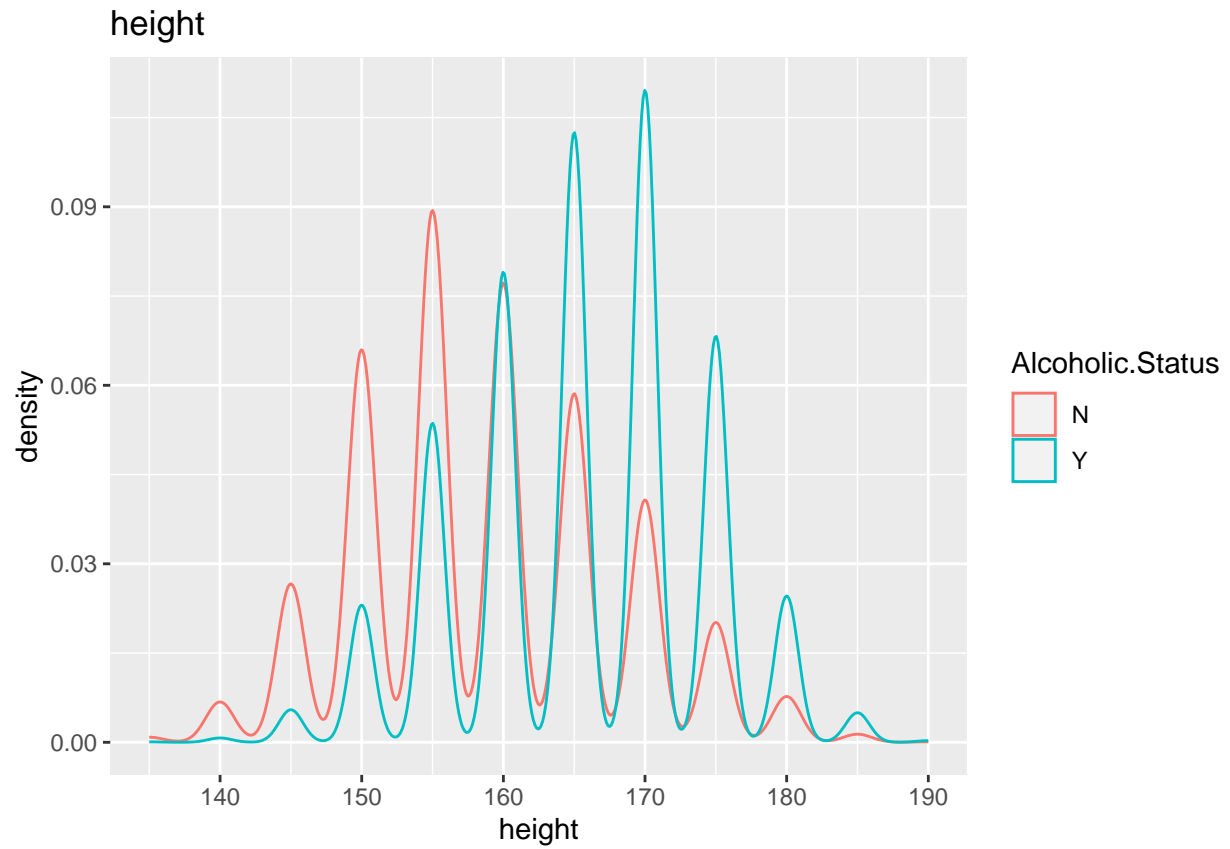
```
## Warning: 'aes_string()' was deprecated in ggplot2 3.0.0.  
## i Please use tidy evaluation idioms with 'aes()'.  
## i See also 'vignette("ggplot2-in-packages")' for more information.  
## This warning is displayed once every 8 hours.  
## Call 'lifecycle::last_lifecycle_warnings()' to see where this warning was  
## generated.
```



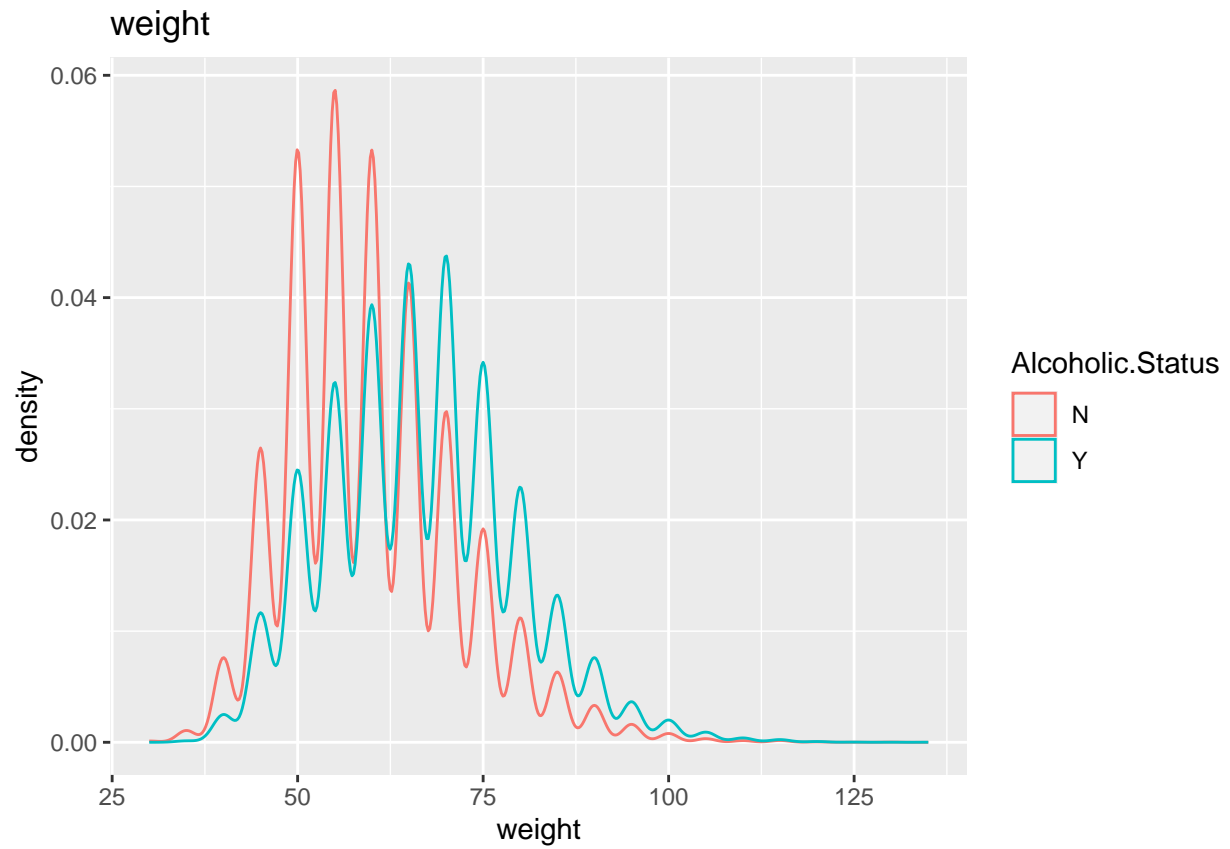
```
## Warning: Removed 4877 rows containing non-finite values ('stat_density()').
```



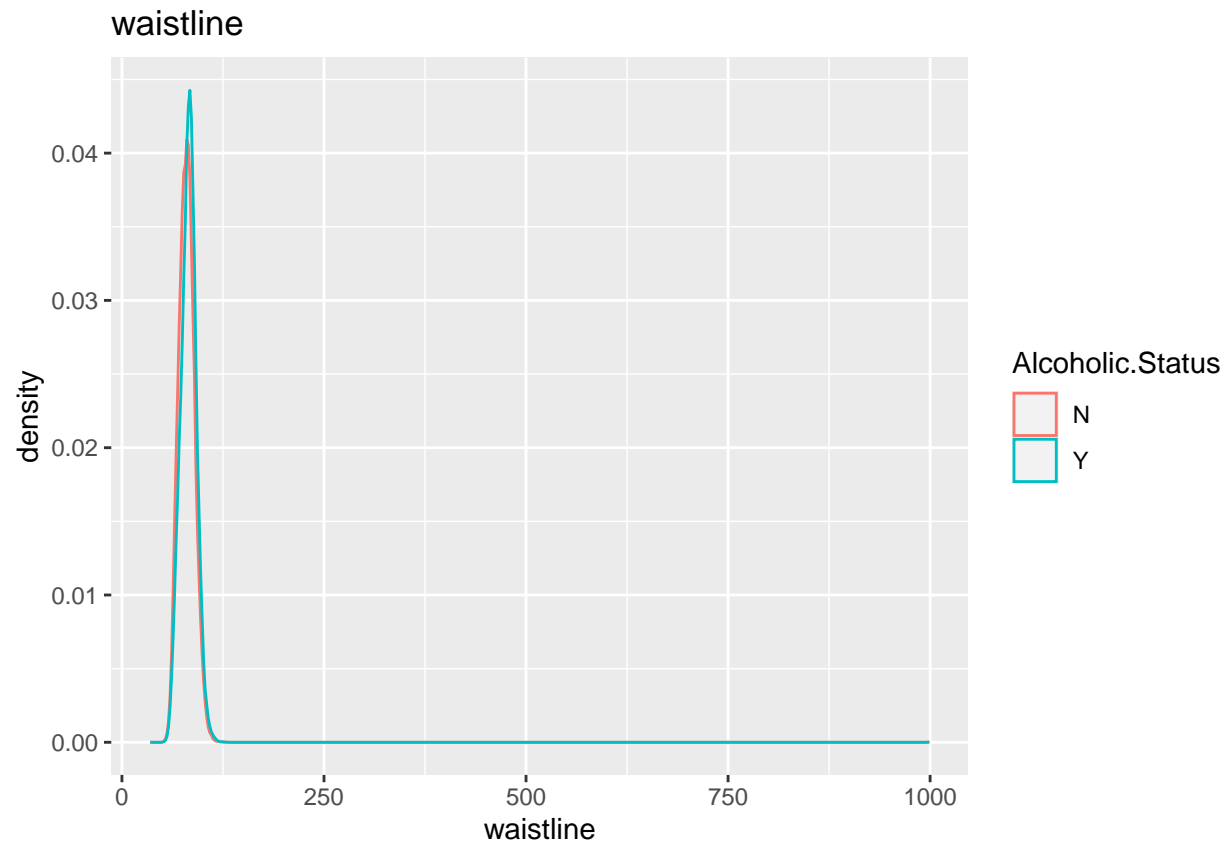
```
## Warning: Removed 4941 rows containing non-finite values ('stat_density()').
```



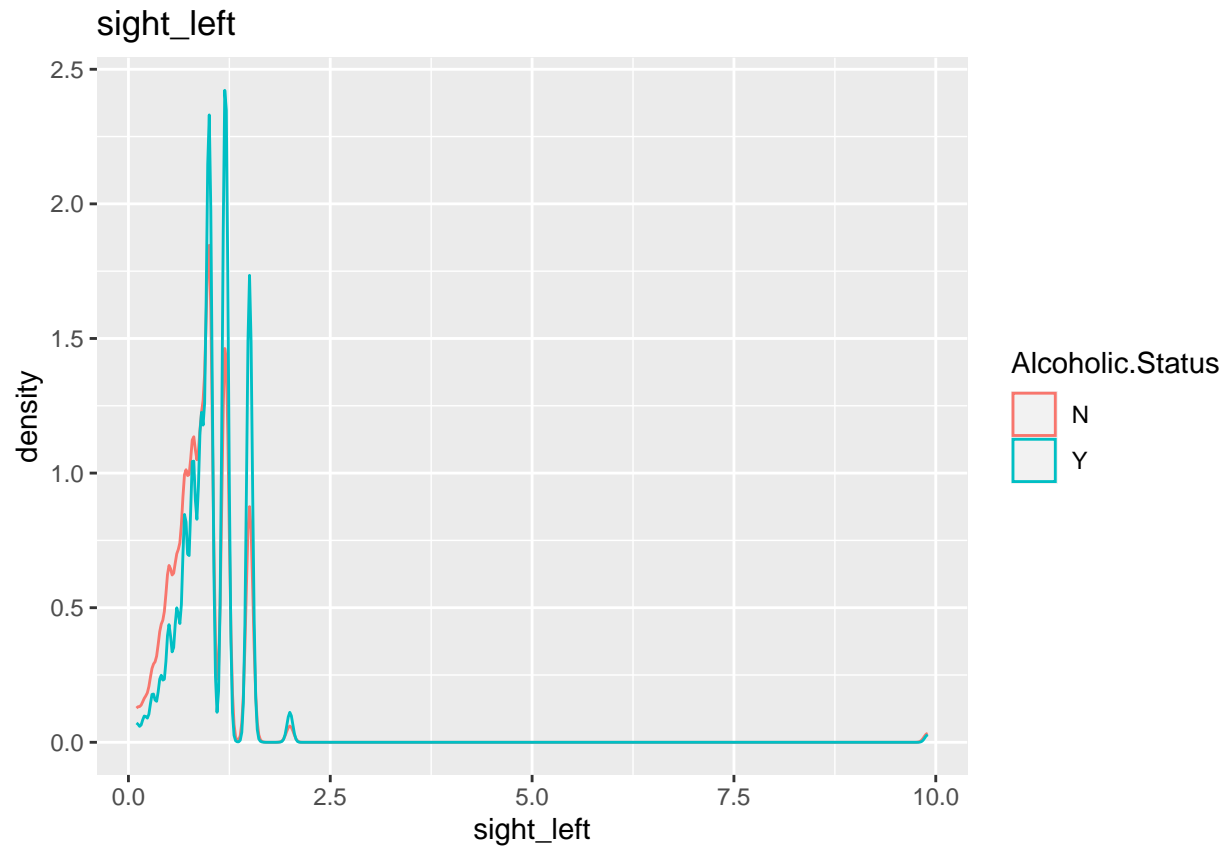
```
## Warning: Removed 4972 rows containing non-finite values ('stat_density()').
```



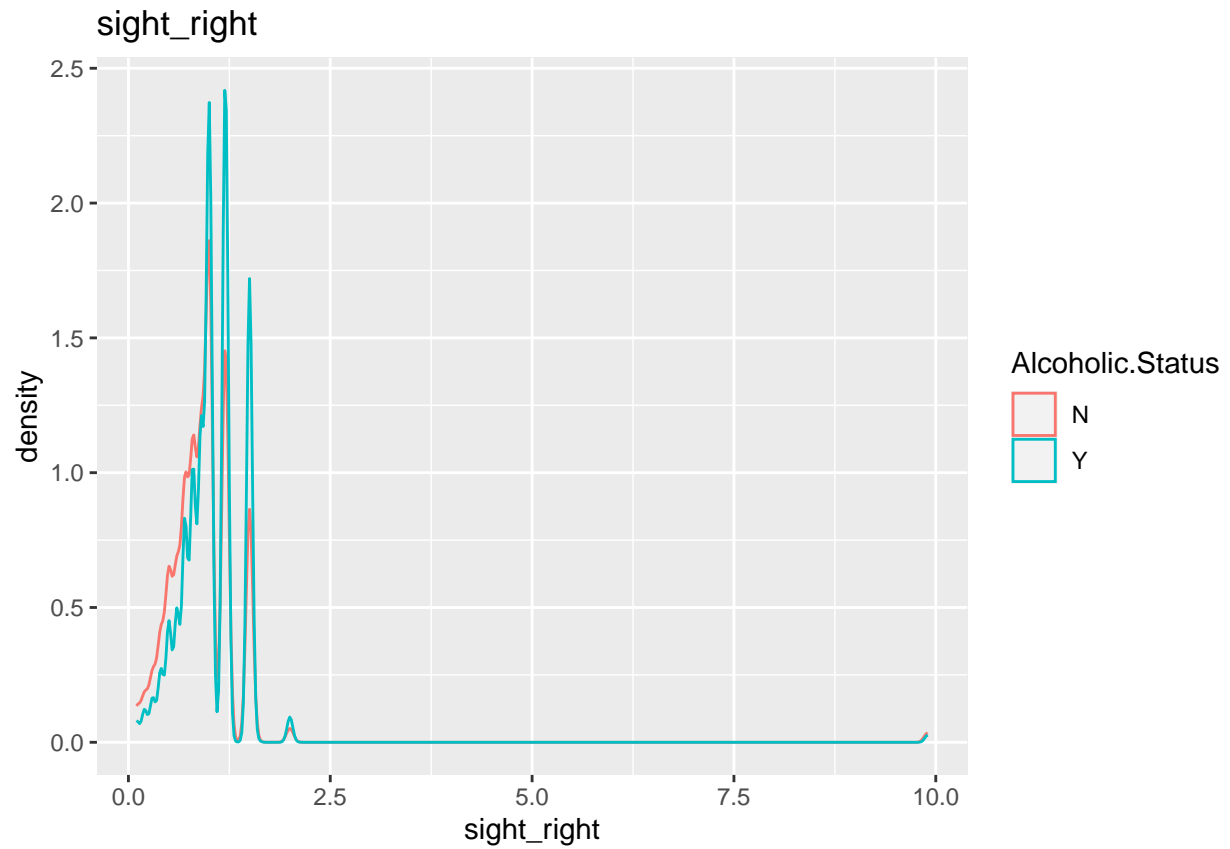
```
## Warning: Removed 4940 rows containing non-finite values ('stat_density()').
```



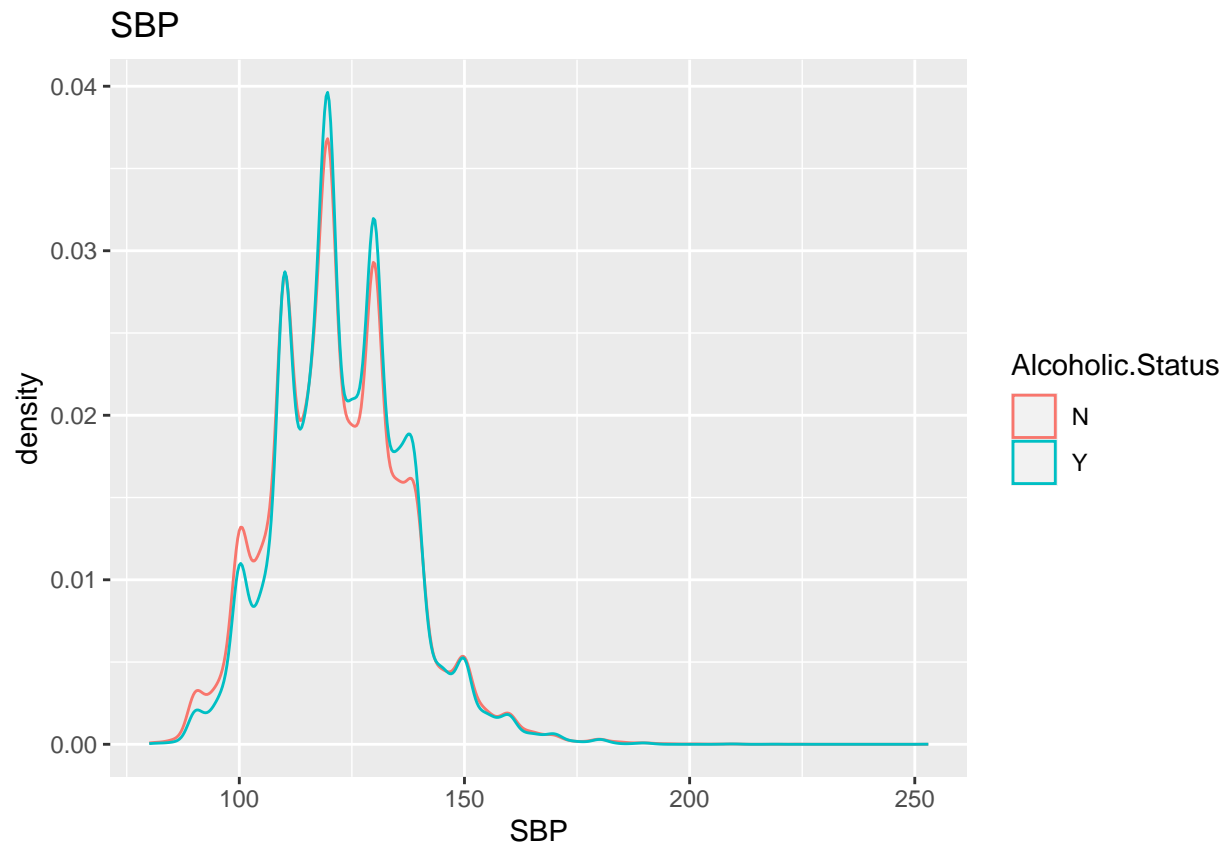
```
## Warning: Removed 4877 rows containing non-finite values ('stat_density()').
```

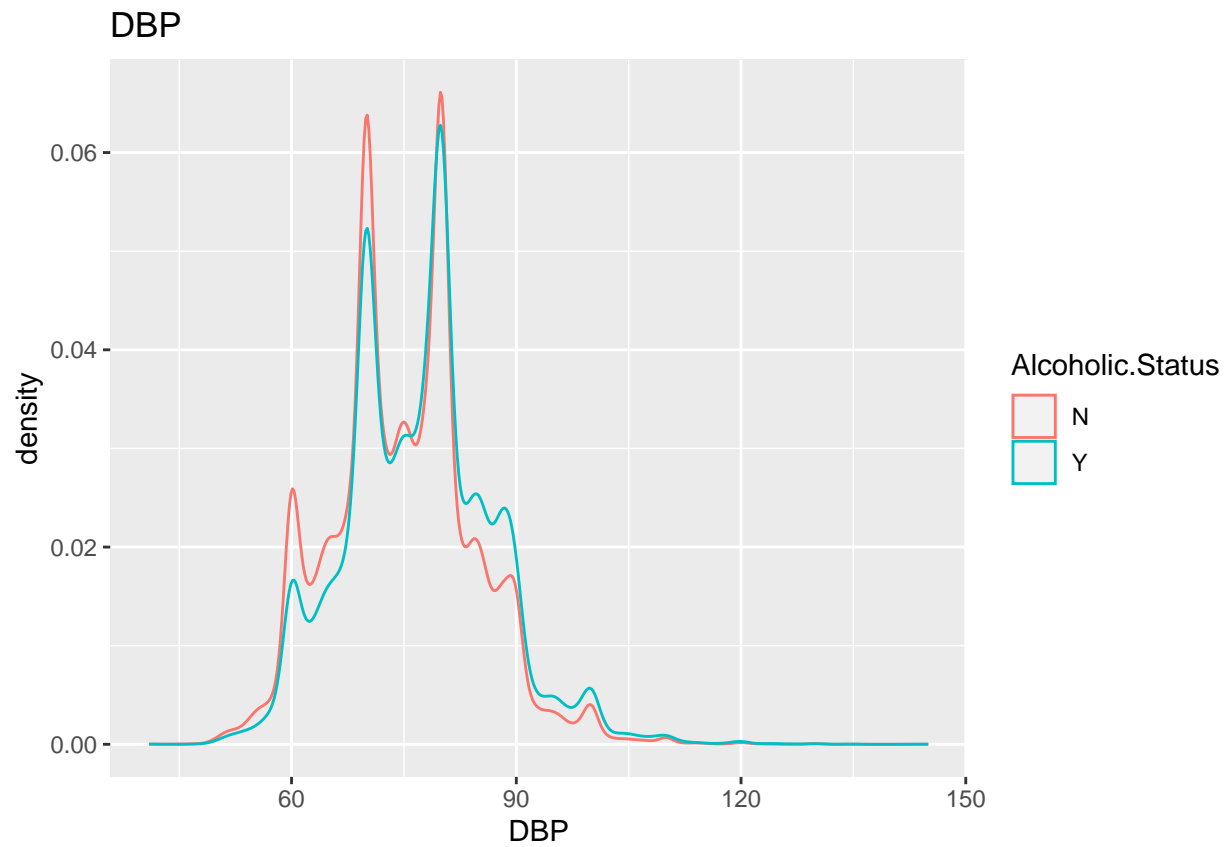
```
## Warning: Removed 4900 rows containing non-finite values ('stat_density()').
```



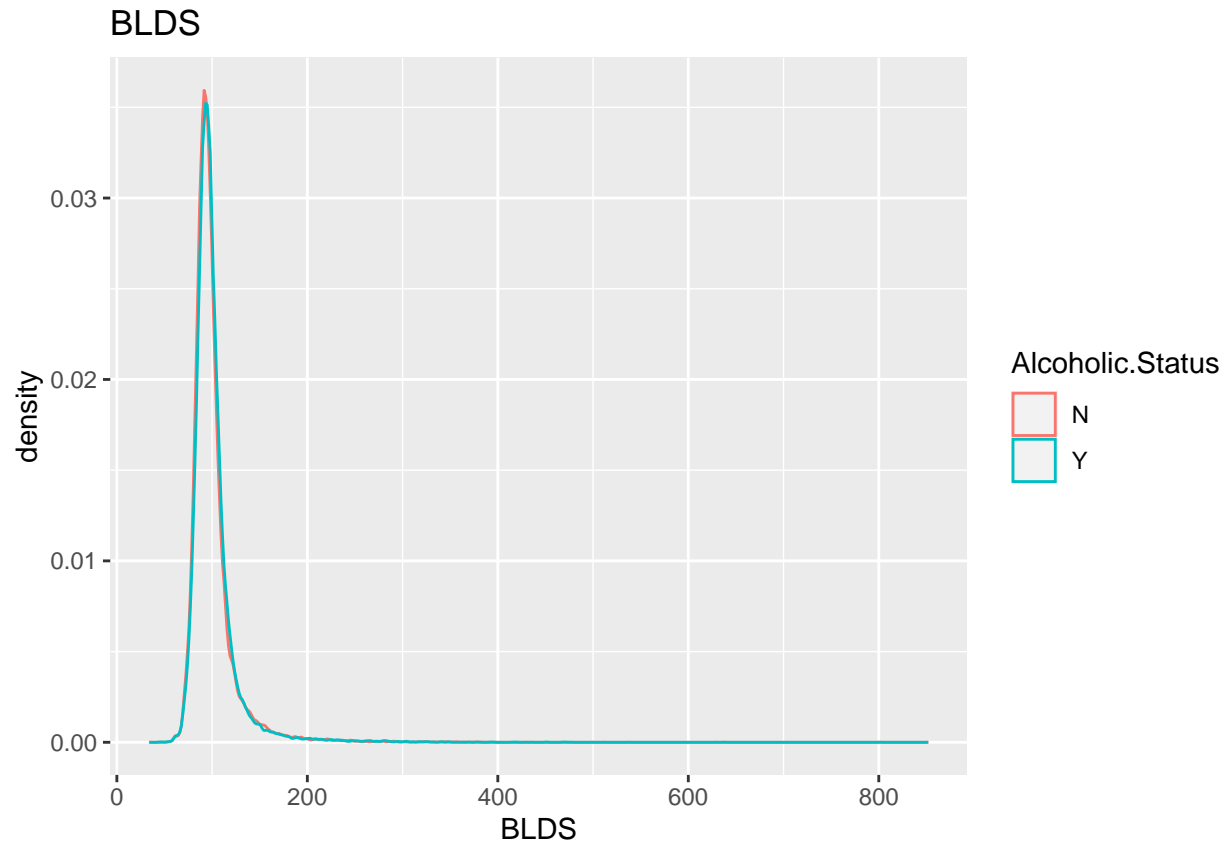
```
## Warning: Removed 4919 rows containing non-finite values ('stat_density()').
```



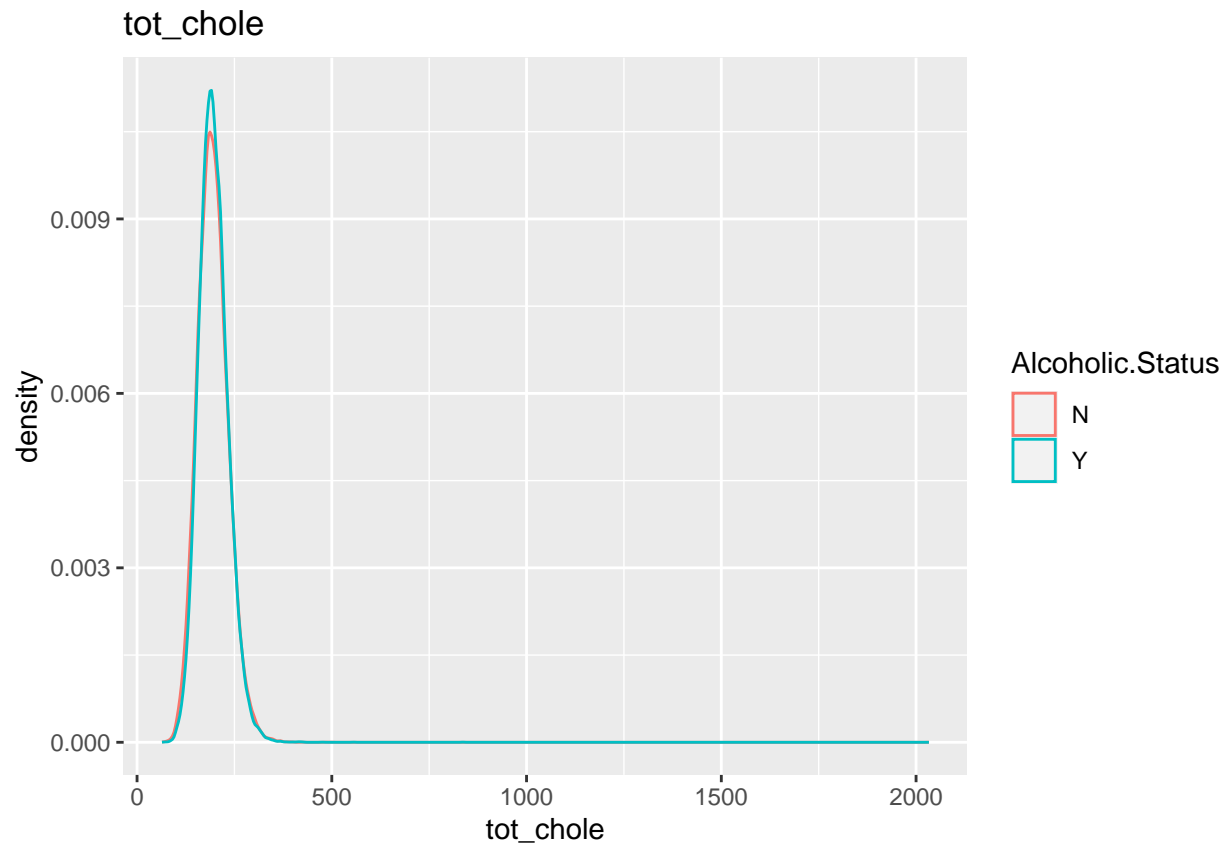
```
## Warning: Removed 4895 rows containing non-finite values ('stat_density()').
```



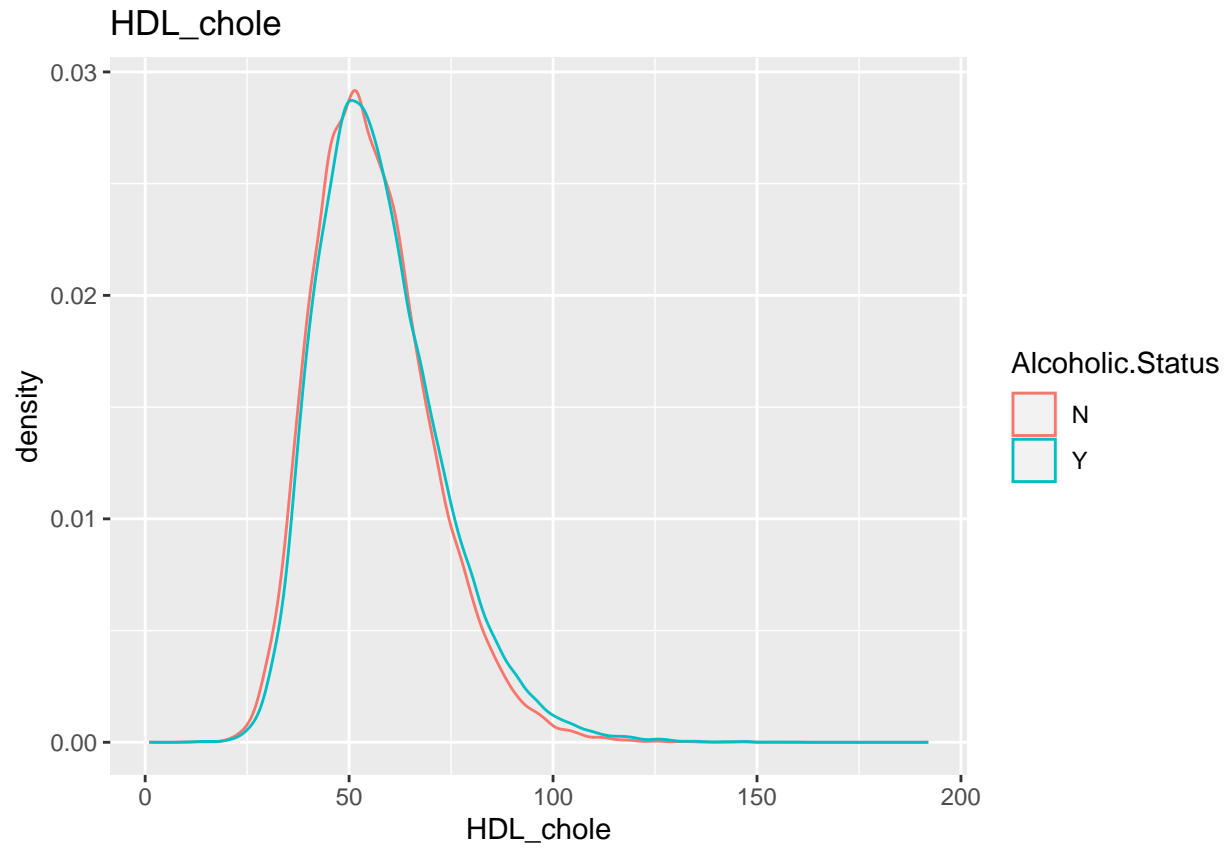
```
## Warning: Removed 4821 rows containing non-finite values ('stat_density()').
```



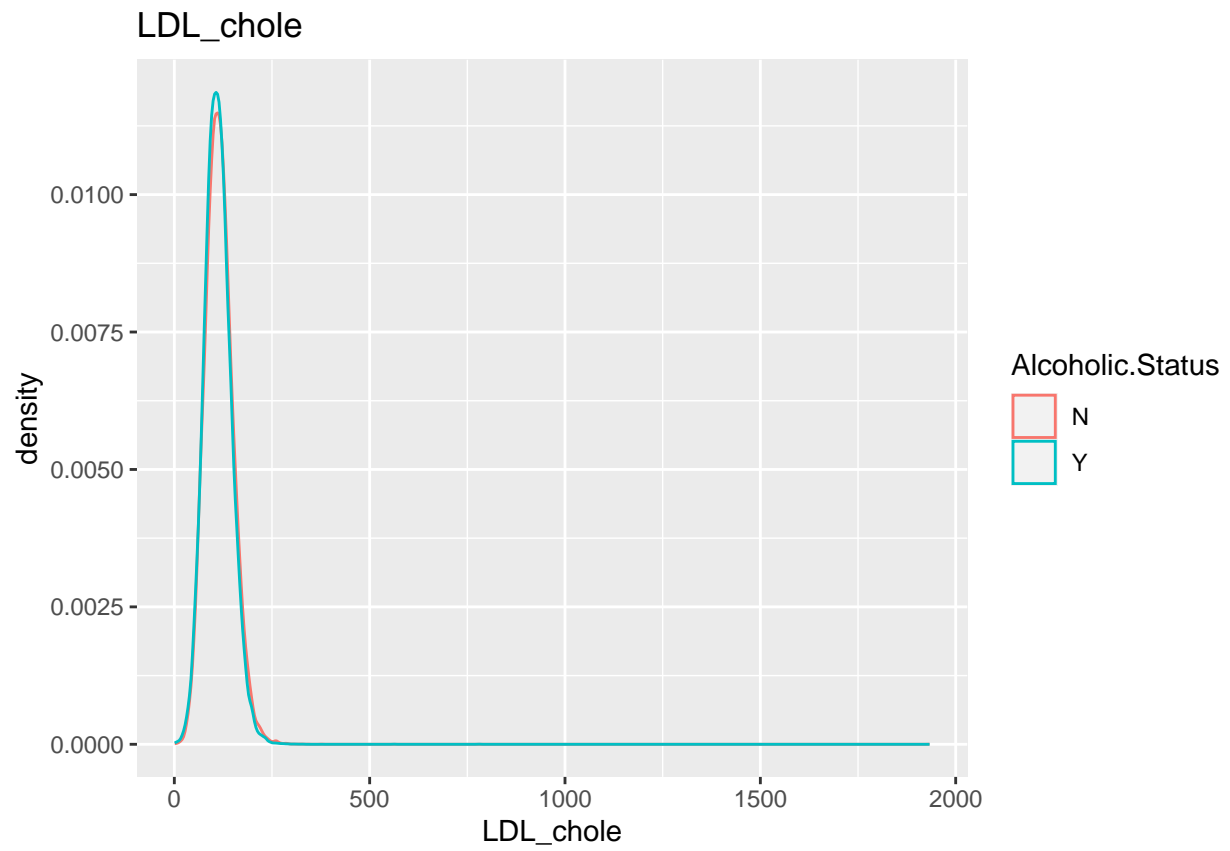
```
## Warning: Removed 4864 rows containing non-finite values ('stat_density()').
```



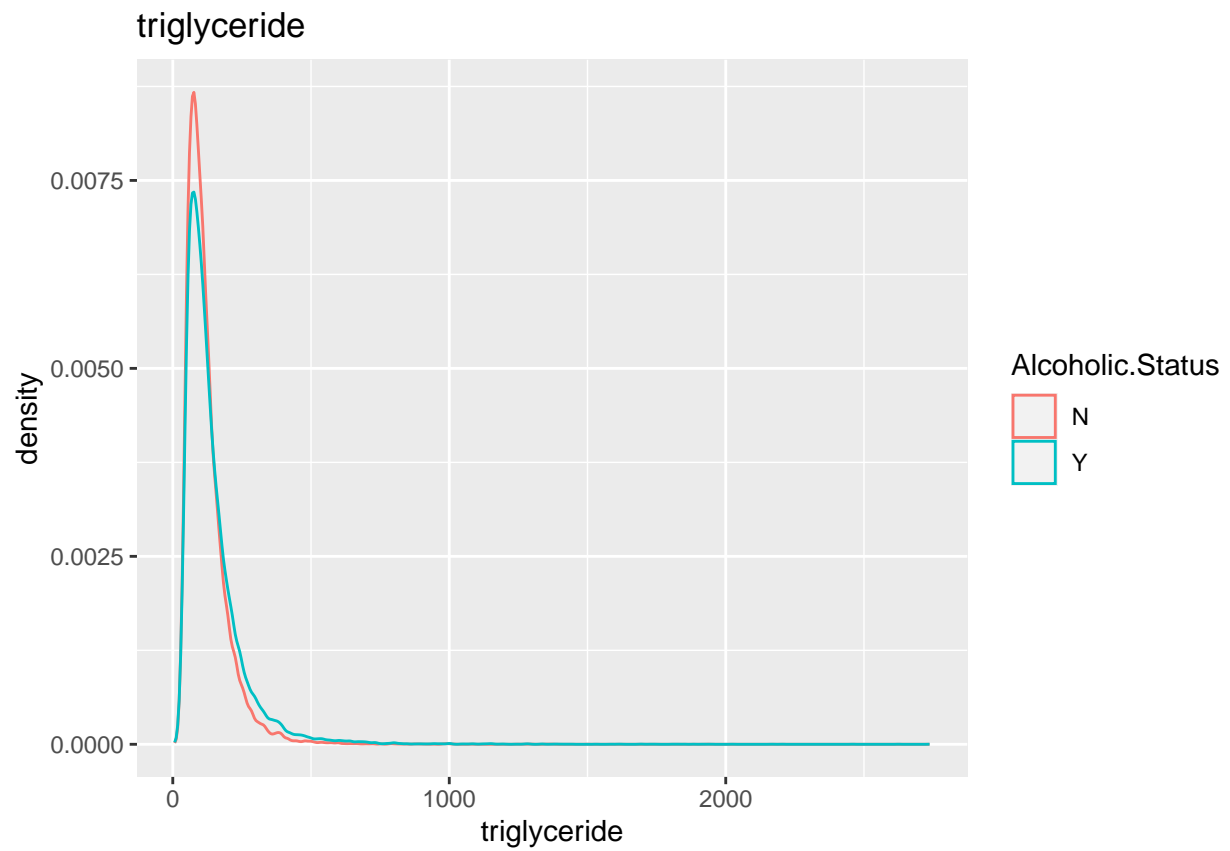
```
## Warning: Removed 4816 rows containing non-finite values ('stat_density()').
```



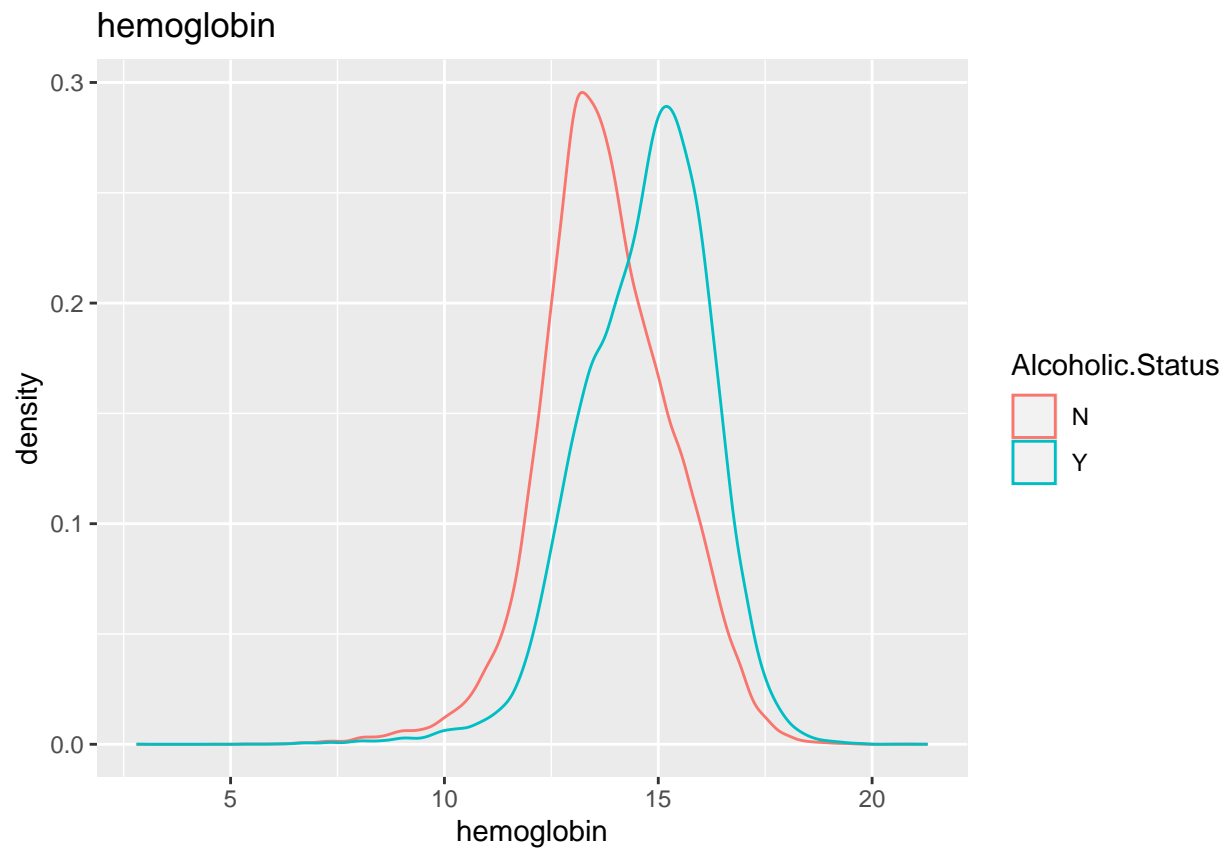
```
## Warning: Removed 4914 rows containing non-finite values ('stat_density()').
```



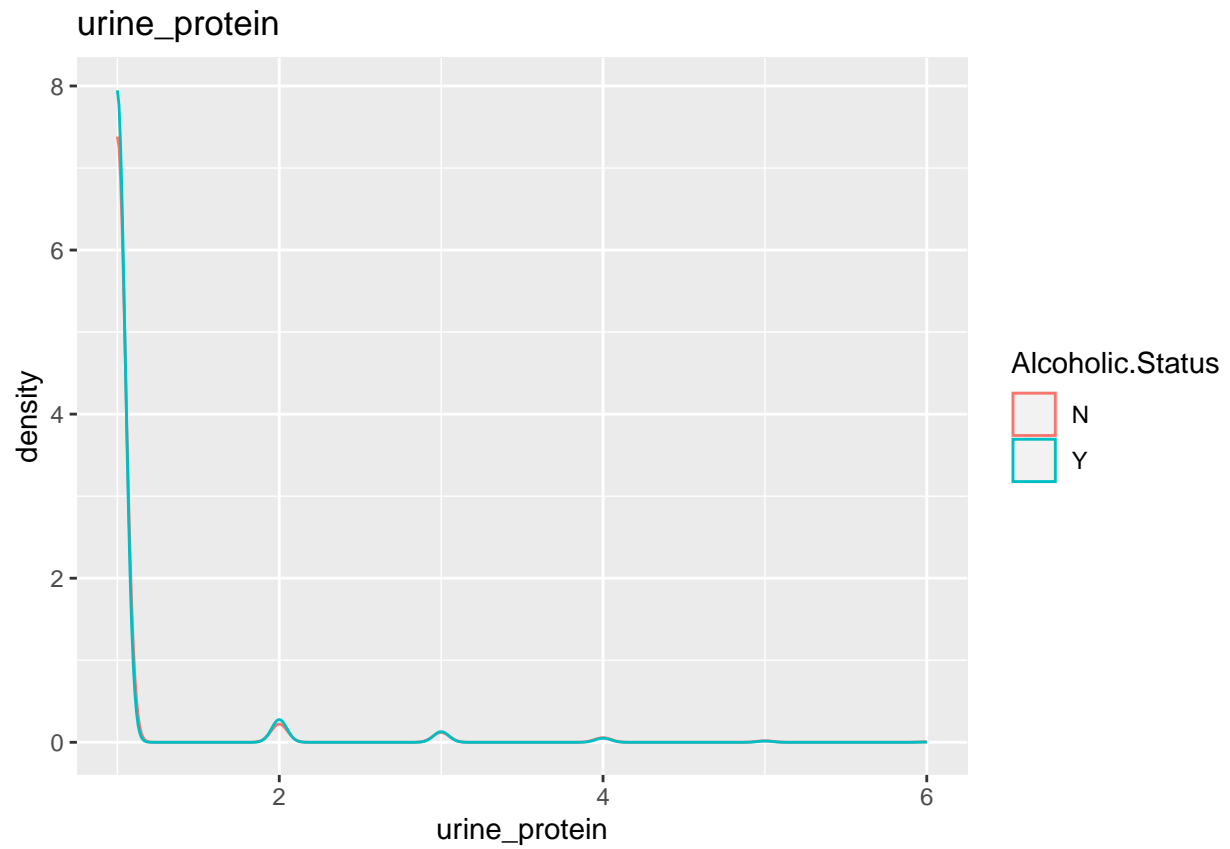
```
## Warning: Removed 4877 rows containing non-finite values ('stat_density()').
```

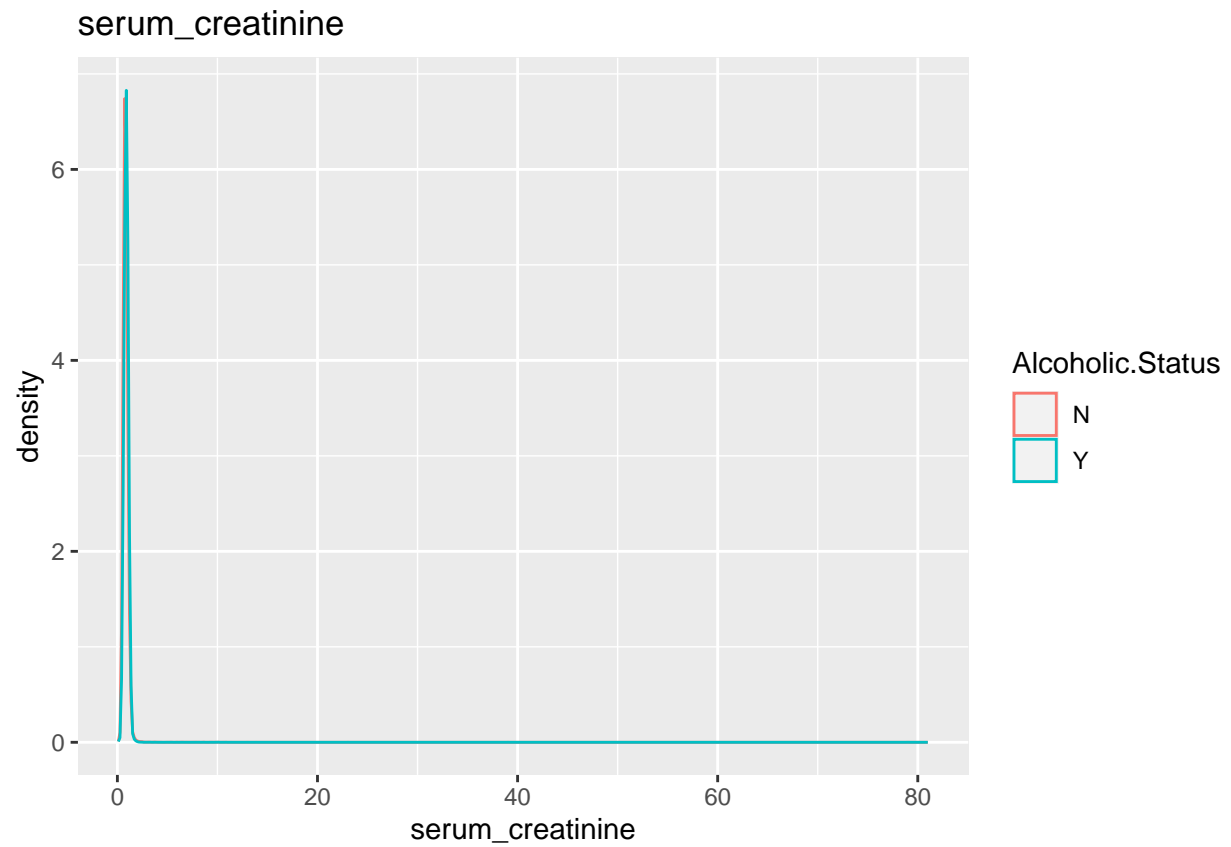
```
## Warning: Removed 4961 rows containing non-finite values ('stat_density()').
```



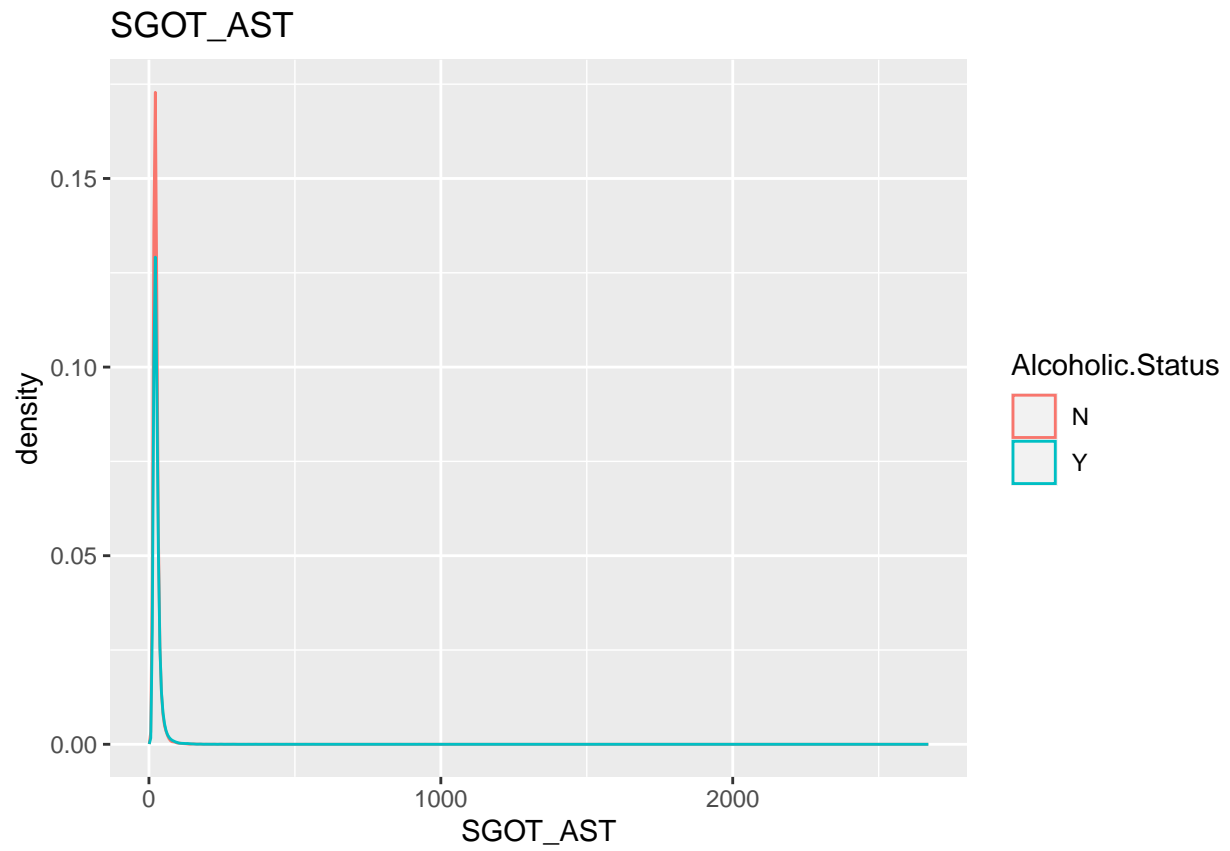
```
## Warning: Removed 4899 rows containing non-finite values ('stat_density()').
```



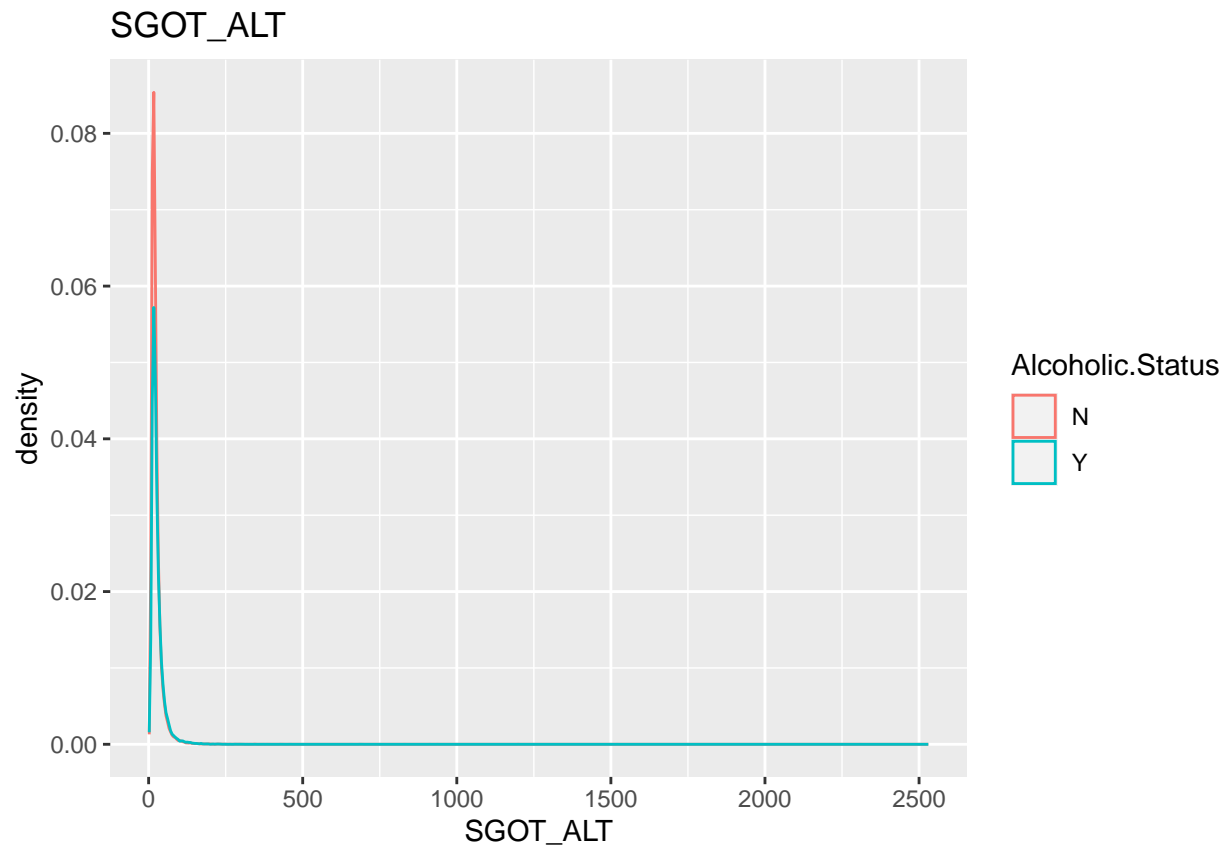
```
## Warning: Removed 4847 rows containing non-finite values ('stat_density()').
```



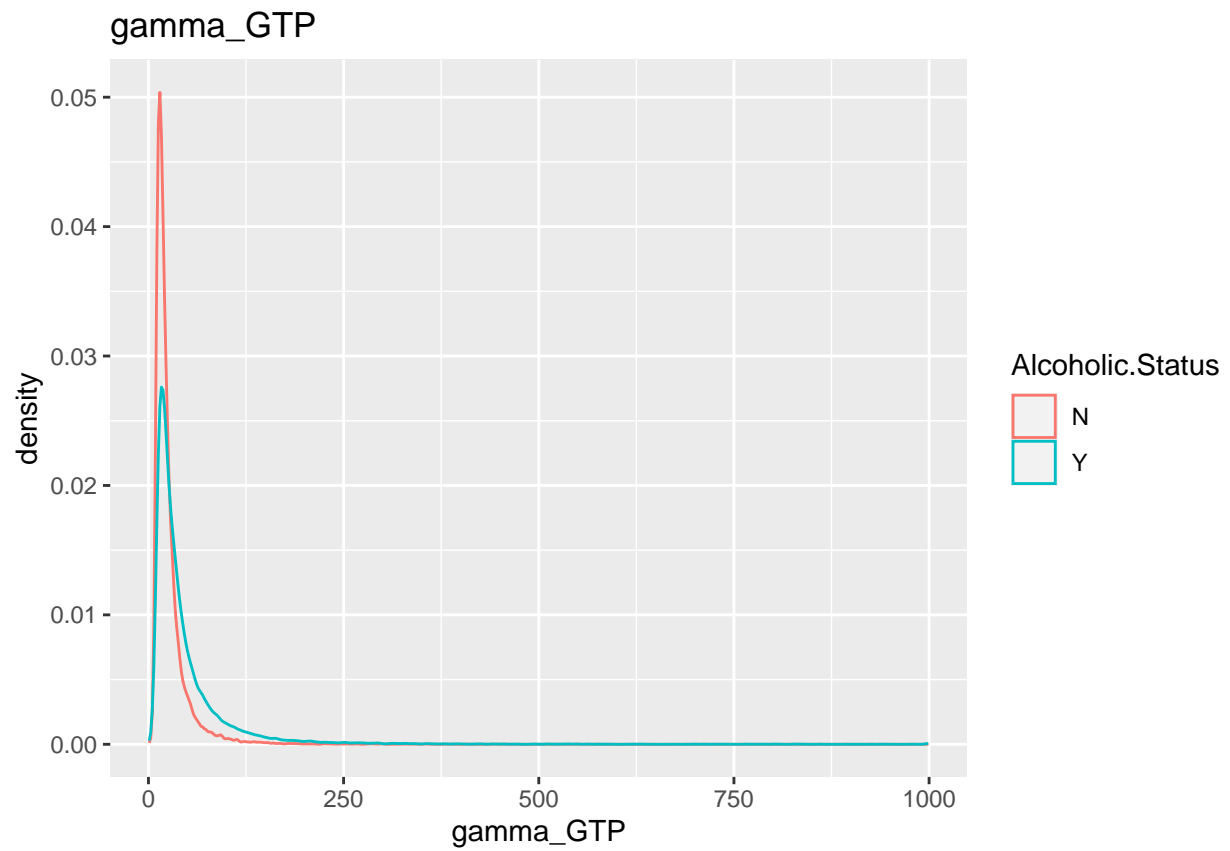
```
## Warning: Removed 4887 rows containing non-finite values ('stat_density()').
```



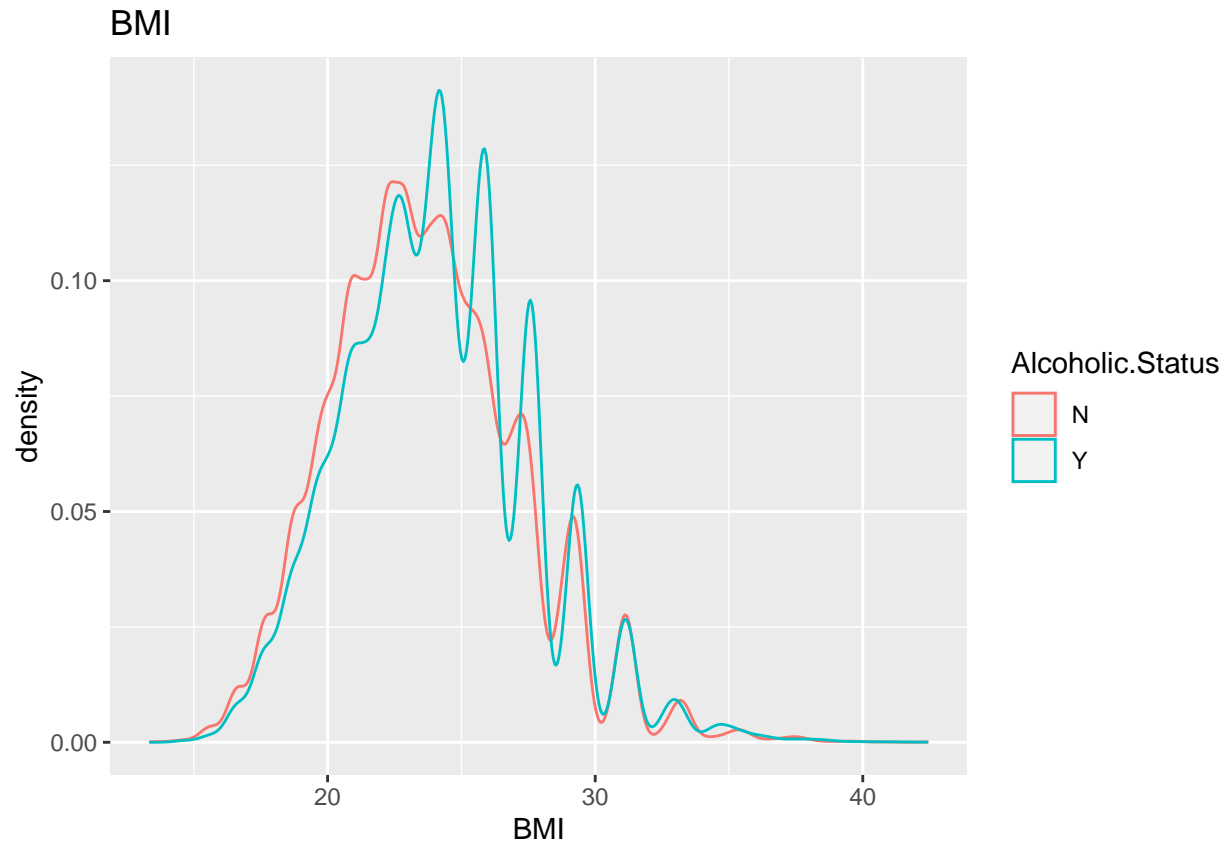
```
## Warning: Removed 4893 rows containing non-finite values ('stat_density()').
```



```
## Warning: Removed 4961 rows containing non-finite values ('stat_density()').
```



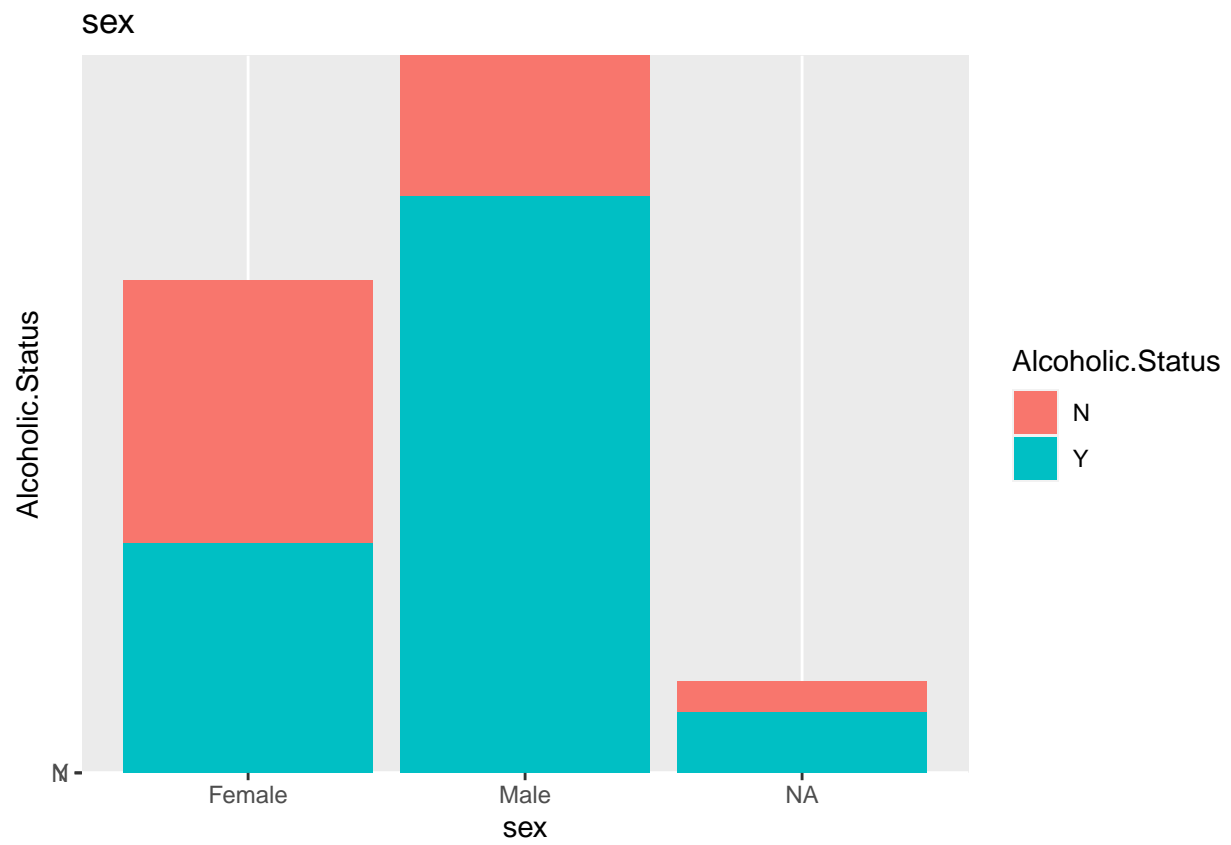
```
## Warning: Removed 4967 rows containing non-finite values ('stat_density()').
```

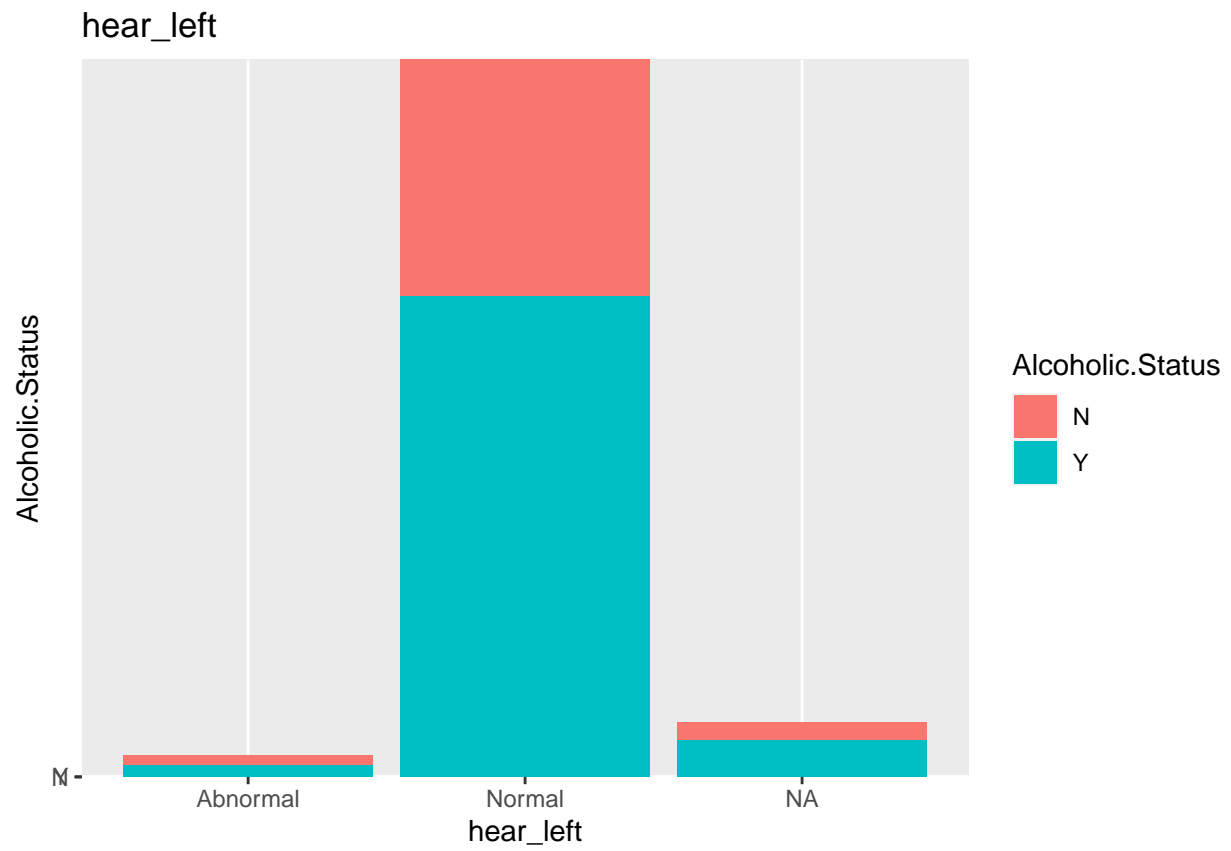


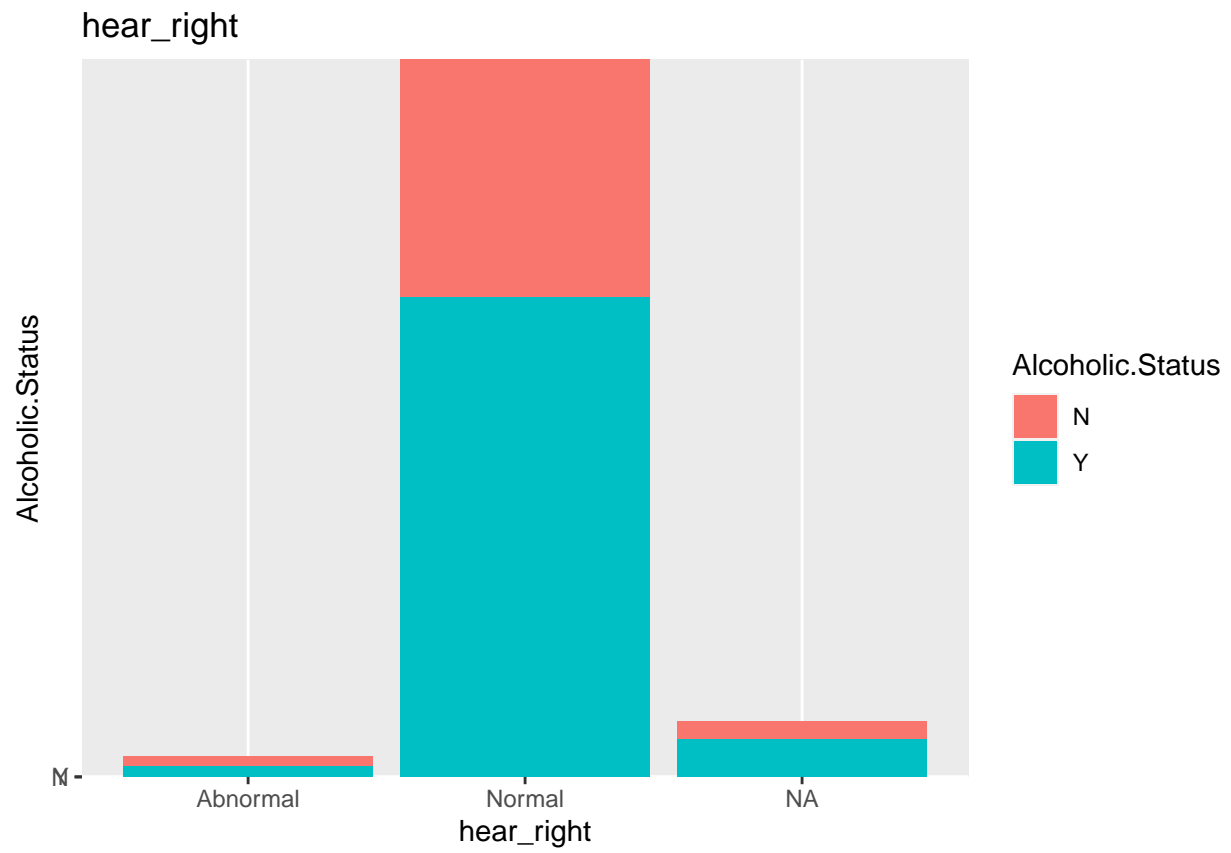
The best four numerical predictors are age, height, hemoglobin and BMI. We can infer this information from their density charts.

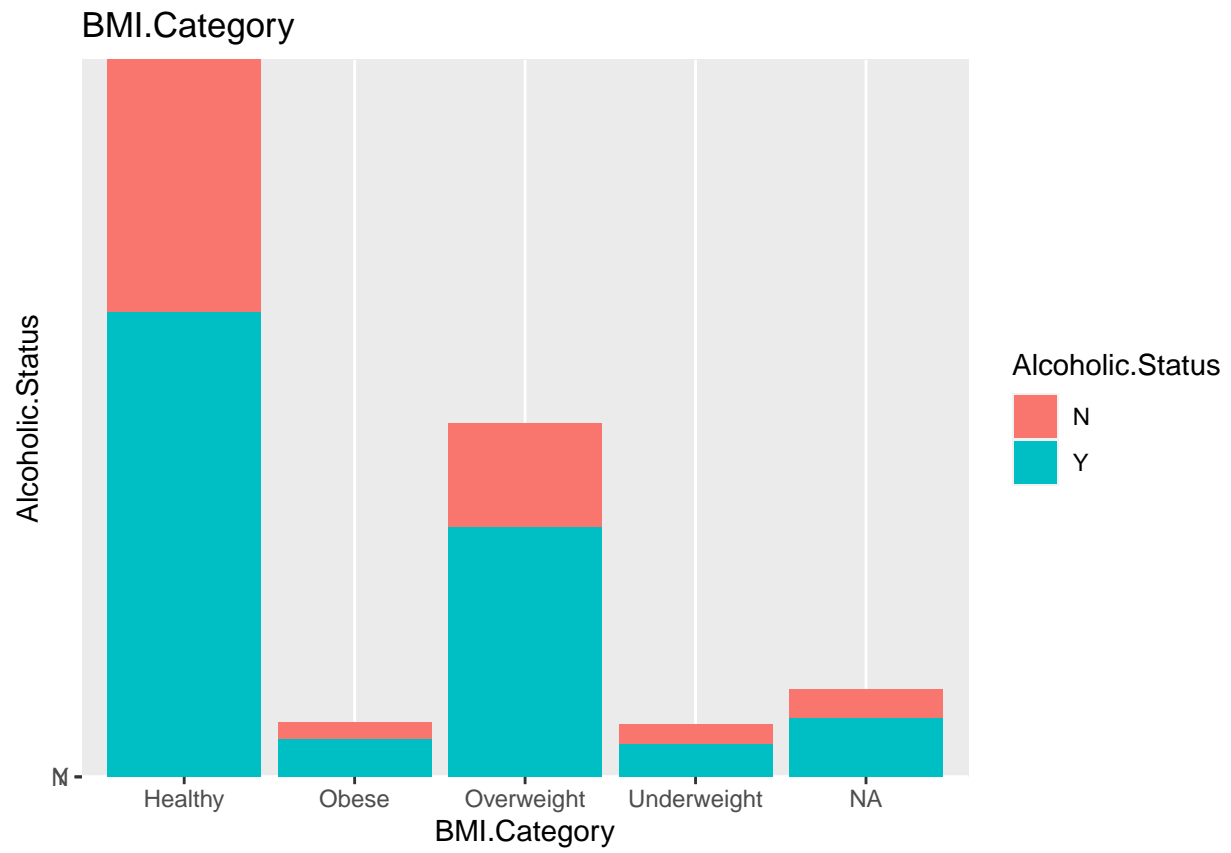
g)

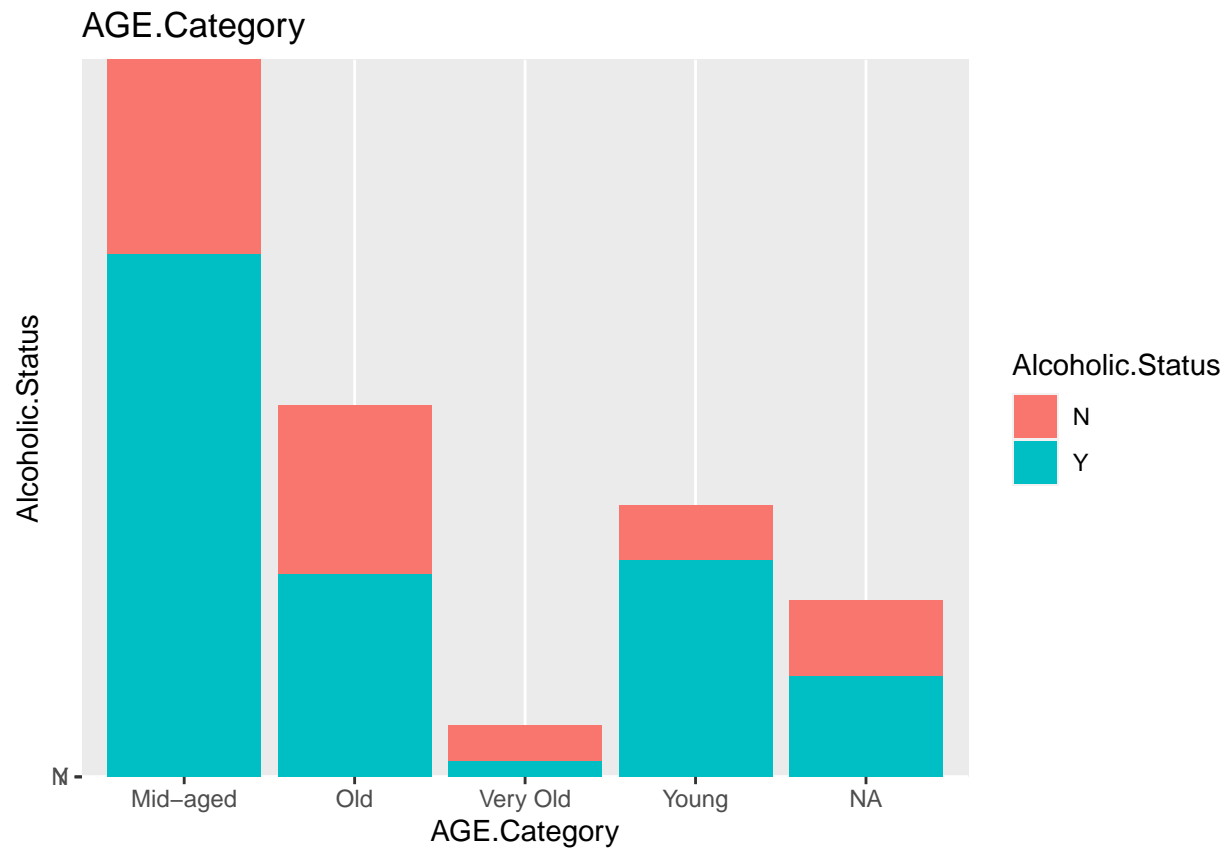
```
cat_names <- names(kagtrain[sapply(kagtrain, is.character)])  
for(variable in cat_names){  
  plot <- ggplot(kagtrain, aes_string(fill = "Alcoholic.Status", y = "Alcoholic.Status", x = variable))  
  print(plot)  
}
```

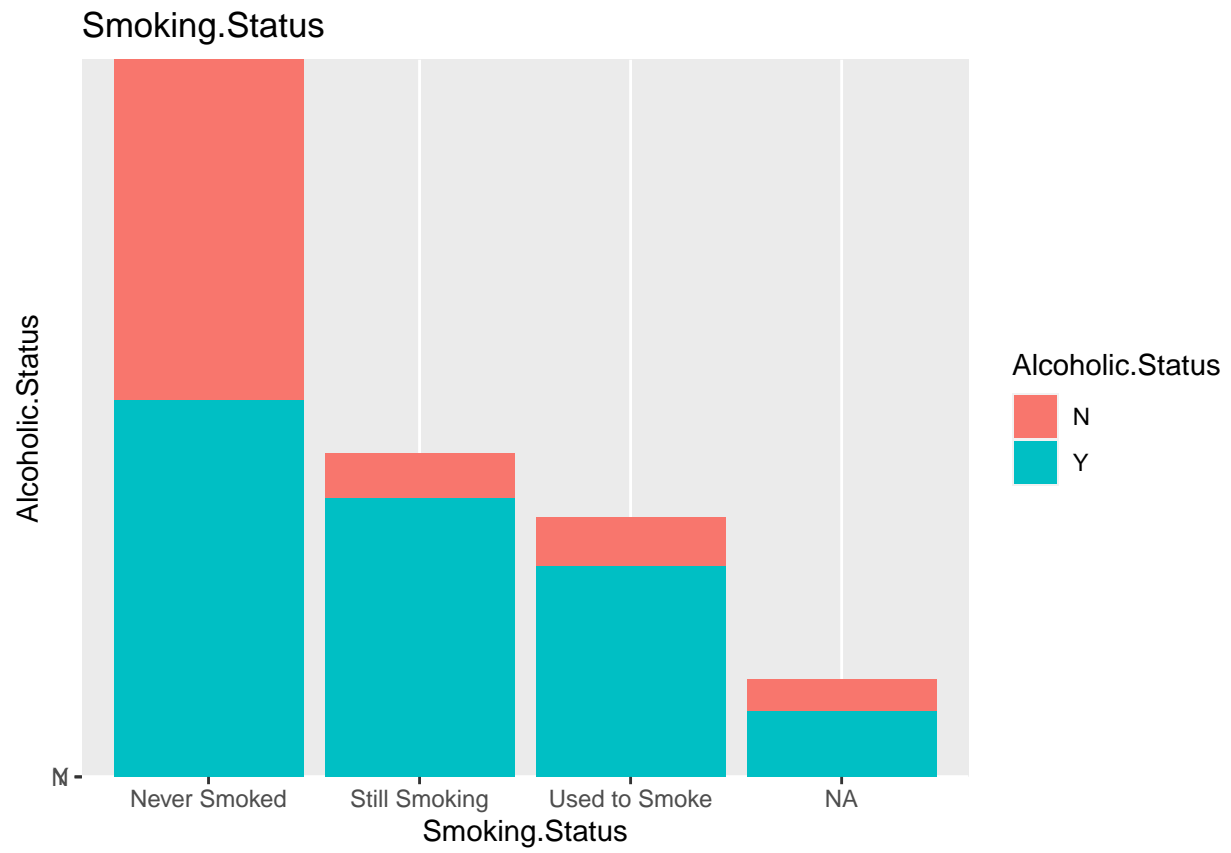



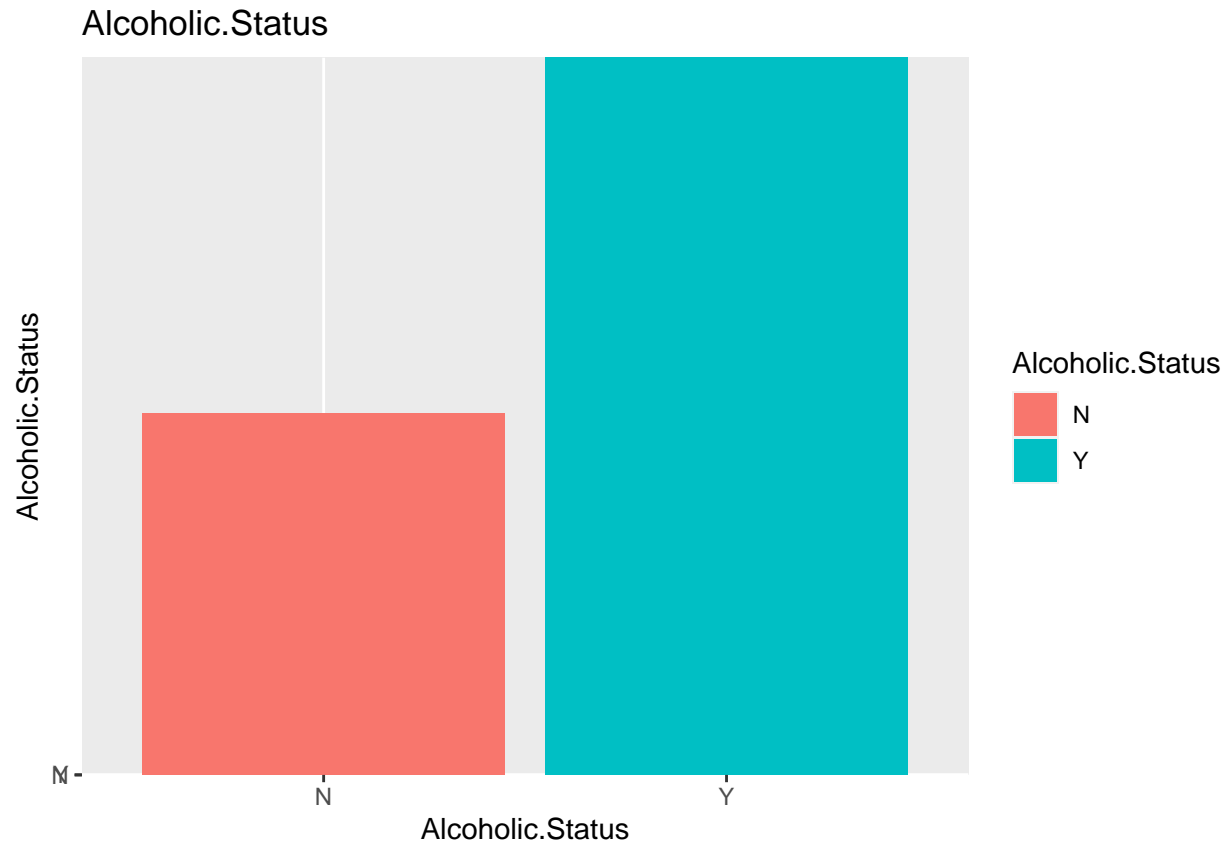












Our best two categorical predictor variables are Smoking.Status and AGE.Category.

Question 2

a)

```
head(kagtrain %>% mutate(across(where(is.numeric), ~replace_na(., median(., na.rm = TRUE)))))
```

```
## ID sex age height weight waistline sight_left sight_right hear_left
## 1 1 Male 75 160 60 81.0 1.0 0.7 Normal
## 2 2 Female 50 160 60 74.0 1.0 1.2 Normal
## 3 3 Male 65 170 80 95.0 1.0 1.5 Normal
## 4 4 <NA> 65 155 55 81.0 0.3 0.4 Abnormal
## 5 5 Male 35 160 60 85.0 1.0 1.0 Normal
## 6 6 Female 50 160 70 73.2 0.3 0.4 Normal
## hear_right SBP DBP BLDS tot_chole HDL_chole LDL_chole triglyceride hemoglobin
## 1 Normal 120 76 136 215 33 143 193 15.0
## 2 Normal 118 70 125 207 85 111 110 13.3
## 3 Normal 149 83 130 115 48 33 170 16.4
## 4 Abnormal 118 67 97 171 65 67 195 13.9
## 5 Normal 96 62 78 114 42 58 72 16.0
## 6 Normal 119 79 220 178 61 80 181 10.5
## urine_protein serum_creatinine SGOT_AST SGOT_ALT gamma_GTP BMI
## 1 3 0.9 28 23 36 23.43750
## 2 1 0.6 28 19 22 23.43750
```

```
## 3      1      1.4      41      64      53 23.87511
## 4      1      0.8      26      25      23 22.89282
## 5      1      1.0      17      24      34 23.87511
## 6      1      0.5      36      20      20 27.34375
## BMI.Category AGE.Category Smoking.Status Alcoholic.Status
## 1      Healthy      Very Old      Still Smoking      Y
## 2      <NA>      Mid-aged      Never Smoked      Y
## 3      Overweight      Old      Still Smoking      Y
## 4      <NA>      Old      Never Smoked      N
## 5      Healthy      Mid-aged      Still Smoking      N
## 6      Overweight      Mid-aged      Never Smoked      N
```

```
kagtrain$sex <- as.factor(kagtrain$sex)
kagtrain$hear_left <- as.factor(kagtrain$hear_left)
kagtrain$hear_right <- as.factor(kagtrain$hear_right)
kagtrain$BMI.Category <- as.factor(kagtrain$BMI.Category)
kagtrain$AGE.Category <- as.factor(kagtrain$AGE.Category)
kagtrain$Smoking.Status <- as.factor(kagtrain$Smoking.Status)
kagtrain$Alcoholic.Status <- as.factor(kagtrain$Alcoholic.Status)
cleankagtrain <- kagtrain[complete.cases(kagtrain), ]
cleankagtrain$Alcoholic.Status <- as.factor(cleankagtrain$Alcoholic.Status)
```

```
glmkg <- glm(Alcoholic.Status ~ . - ID - Alcoholic.Status, data = cleankagtrain, family = binomial())
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(glmkg)
```

```
##
## Call:
## glm(formula = Alcoholic.Status ~ . - ID - Alcoholic.Status, family = binomial(),
##      data = cleankagtrain)
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)   -3.9771959   3.2391986  -1.228  0.21951
## sexMale         1.0663028   0.0945390  11.279 < 2e-16 ***
## age           -0.0379901   0.0052914  -7.180 6.99e-13 ***
## height          0.0108923   0.0197531   0.551  0.58135
## weight          0.0073973   0.0245745   0.301  0.76340
## waistline     -0.0005591   0.0020247  -0.276  0.78242
## sight_left      0.0134764   0.0485802   0.277  0.78147
## sight_right    -0.0247307   0.0453646  -0.545  0.58565
## hear_leftNormal -0.0928936   0.1769397  -0.525  0.59958
## hear_rightNormal 0.0594782   0.1846674   0.322  0.74739
## SBP            0.0025443   0.0026671   0.954  0.34011
## DBP            0.0118107   0.0037365   3.161  0.00157 **
## BLDS           0.0001888   0.0010225   0.185  0.85354
## tot_chole       0.0004400   0.0043060   0.102  0.91861
## HDL_chole       0.0230691   0.0044950   5.132 2.86e-07 ***
## LDL_chole      -0.0019803   0.0043409  -0.456  0.64825
## triglyceride    0.0008135   0.0007352   1.107  0.26849
```



```
## hemoglobin          -0.0044983  0.0211150  -0.213  0.83130
## urine_protein       -0.0437604  0.0548591  -0.798  0.42505
## serum_creatinine    -0.3112565  0.1164993  -2.672  0.00755 **
## SGOT_AST            0.0196459  0.0034722   5.658 1.53e-08 ***
## SGOT_ALT            -0.0249463  0.0023168 -10.768 < 2e-16 ***
## gamma_GTP           0.0134537  0.0010989  12.243 < 2e-16 ***
## BMI                 0.0154784  0.0660724   0.234  0.81478
## BMI.CategoryObese    -0.5100170  0.1930533  -2.642  0.00825 **
## BMI.CategoryOverweight -0.0727624  0.0886899  -0.820  0.41198
## BMI.CategoryUnderweight -0.0980623  0.1306307  -0.751  0.45284
## AGE.CategoryOld      -0.1812045  0.0966219  -1.875  0.06074 .
## AGE.CategoryVery Old -0.4333644  0.2273759  -1.906  0.05666 .
## AGE.CategoryYoung     0.0412159  0.1071919   0.385  0.70060
## Smoking.StatusStill Smoking 0.8290526  0.0711244  11.656 < 2e-16 ***
## Smoking.StatusUsed to Smoke 0.8717824  0.0725326  12.019 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 13906  on 10032  degrees of freedom
## Residual deviance: 10852  on 10001  degrees of freedom
## AIC: 10916
##
## Number of Fisher Scoring iterations: 5
```

b)

```
kagprob <- predict(glmkag, data = cleankagtrain, type = "response")
kagpredlog <- rep("Y", length(kagprob))
kagpredlog[kagprob <= 0.5] <- "N"
table(kagpredlog, cleankagtrain$Alcoholic.Status)
```

```
##
## kagpredlog      N      Y
##              N 3577 1371
##              Y 1364 3721
```

```
mean(kagpredlog != cleankagtrain$Alcoholic.Status)
```

```
## [1] 0.2726004
```

The misclassification rate is 27.26%.

c)

```
kptest <- predict(glmkag, data = cleankagtrain, newdata = kagtest, type = "response")
kgtestpl <- rep("Y", length(kptest))
kgtestpl[kptest <= 0.5] <- "N"
my_kaggle <- data.frame(ID = 1:nrow(kagtest), predictions = kgtestpl)
write.csv(my_kaggle, file = "kagglepredictions.csv", row.names = FALSE)
```

My kaggle public score is 0.52996.

d) My kaggle rank is 64th.

Question 3

```
winetrain <- read.csv("WineTrain copy.csv")
winetest <- read.csv("WineTest copy.csv")
winetrain$Class <- as.factor(winetrain$Class)
winetrain$Wine.Color <- as.factor(winetrain$Wine.Color)
winetest$Class <- as.factor(winetest$Class)
winetest$Wine.Color <- as.factor(winetest$Wine.Color)
winedat <- rbind(winetrain, winetest)
head(winedat)
```

```
##      X Wine.Color fixed.acidity volatile.acidity citric.acid residual.sugar
## 1 1          W           7.3             0.23         0.41          14.6
## 2 2          R          10.0             0.32         0.59           2.2
## 3 3          W           6.2             0.27         0.43           7.8
## 4 4          W           6.6             0.25         0.32           5.6
## 5 5          W           6.9             0.24         0.39           1.3
## 6 6          W           7.1             0.23         0.39           1.6
##      chlorides free.sulfur.dioxide total.sulfur.dioxide density    pH sulphates
## 1      0.048              73              223 0.99863 3.16      0.71
## 2      0.077               3              15 0.99940 3.20      0.78
## 3      0.056              48              244 0.99560 3.10      0.51
## 4      0.039              15              68 0.99163 2.96      0.52
## 5      0.063              18             136 0.99280 3.31      0.48
## 6      0.032              12              65 0.98980 3.25      0.40
##      alcohol Class
## 1      9.4   Bad
## 2      9.6   Bad
## 3      9.0   Bad
## 4     11.1  Good
## 5     10.4  Good
## 6     12.7  Good
```

```
dim(winedat)
```

```
## [1] 4000   14
```

```
library(crossval)
```

```
##
## Attaching package: 'crossval'

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

```
library(boot)
```

```
##
## Attaching package: 'boot'
```

```
## The following object is masked from 'package:lattice':
##
##      melanoma
```

a)

```
# logistic regression
wineglm <- glm(Class ~ . - X - Class, data = winedat, family = binomial())
wineprob <- predict(wineglm, data = winedat, type = "response")
winepredl <- rep("Good", length(wineprob))
winepredl[wineprob <= 0.5] <- "Bad"
table(winepredl, winedat$Class)
```

```
##
## winepredl  Bad Good
##      Bad  1374  755
##      Good   676 1195
```

```
mean(winepredl != winedat$Class)
```

```
## [1] 0.35775
```

We can see our confusion matrix in our output above. Our misclassification rate for this model is 35.775%.

b)

```
# lda model
winelda <- lda(Class ~ . - X - Class, data = winedat, CV = TRUE)
summary(winelda)
```

```
##           Length Class  Mode
## class       4000  factor numeric
## posterior    8000  -none- numeric
## terms         3    terms  call
## call          4    -none- call
## xlevels       1    -none- list
```

```
table(winelda$class, winedat$Class)
```

```
##
##           Bad Good
##      Bad  1374  771
##      Good   676 1179
```

```
mean(winelda$class != winedat$Class)
```

```
## [1] 0.36175
```

Our misclassification rate is 36.175%.

c)

```
# qda model
wineqda <- qda(Class ~ . - X - Class, data = winedat, CV = TRUE)
summary(wineqda)
```

```
##           Length Class  Mode
## class      4000   factor numeric
## posterior  8000   -none- numeric
## terms        3    terms  call
## call         4   -none- call
## xlevels       1   -none- list
```

```
table(wineqda$class, winedat$Class)
```

```
##
##           Bad Good
##    Bad  1058  499
##    Good   992 1451
```

```
mean(wineqda$class != winedat$Class)
```

```
## [1] 0.37275
```

Our misclassification rate for our qda model of the wine data is 37.275%.

d)

```
# knn model with k = 25
wine_knn1 <- train(as.factor(Class) ~ . - X, data = winedat, method = "knn", trControl = trainControl(m
```

- e) Our model with the lowest misclassification rate is our glm model for our data. The highest misclassification rate is from our qda model.

Question 4

a)

```
# logistic regression with 10 fold method
wineglm10f <- cv.glm(winedat, wineglm, K = 10)
summary(wineglm10f)
```

```
##           Length Class  Mode
## call        4   -none- call
## K            1   -none- numeric
## delta        2   -none- numeric
## seed       626   -none- numeric
```

```
cv.err.10 <- wineglm10f$delta
cv.err.10
```

```
## [1] 0.2264399 0.2263708
```

The MSE for the glm of the wine data is 0.2266443 and the second error of 0.2265667 is for the LOOCV.

b)

```
wine_lda <- train(as.factor(Class) ~ . - X, data = winedat, method = "lda", trControl = trainControl(mo
caret::confusionMatrix(wine_lda)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction  Bad Good
##           Bad  34.2 19.4
##           Good 17.1 29.4
##
## Accuracy (average) : 0.6358
```

The misclassification rate is $1 - 0.6413 = 0.3587$.

c)

```
wine_qda <- train(as.factor(Class) ~ . - X, data = winedat, method = "qda", trControl = trainControl(mo
caret::confusionMatrix(wine_qda)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction  Bad Good
##           Bad  26.5 12.8
##           Good 24.8 36.0
##
## Accuracy (average) : 0.624
```

The misclassification rate is $1 - 0.623 = 0.377$.

d)

```
wine_knn <- train(as.factor(Class) ~ . - X, data = winedat, method = "knn", trControl = trainControl(mo
caret::confusionMatrix(wine_knn)
```

```
## Cross-Validated (10 fold) Confusion Matrix
##
## (entries are percentual average cell counts across resamples)
##
##           Reference
## Prediction  Bad Good
##           Bad  31.6 21.6
##           Good 19.7 27.1
##
## Accuracy (average) : 0.587
```

The misclassification rate is $1 - 0.5882 = 0.4118$.

- e) The cv glm model has the lowest misclassification rate amongst all of the cv models. Our highest misclassification rate is from our knn model.