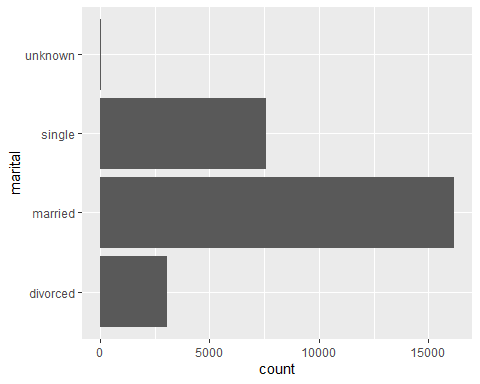
ADS502-Assignment-2.1-R.R

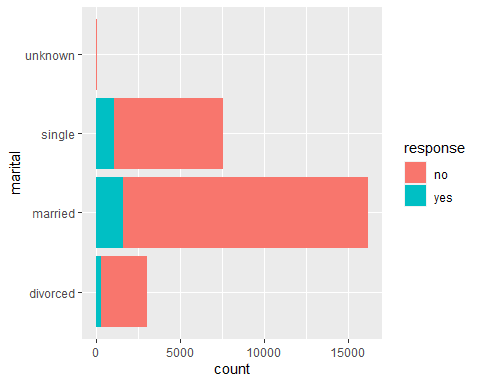
DDY

2021-07-11

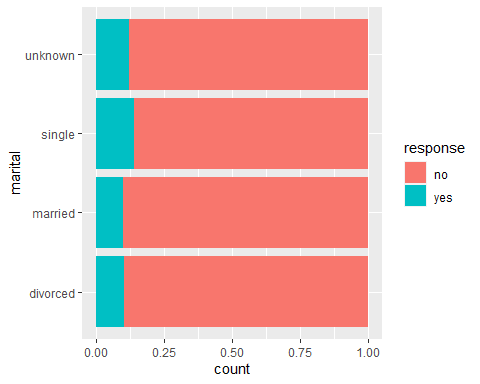
# Assignment 2.1 [R]  
  
# University of San Diego  
  
# ADS 502  
  
# Dingyi Duan  
  
  
# For Exercises 21-30, continue working with the bank\_marketing\_training   
# data set. Use either Python or R to solve each problem.  
  
# 21. Produce the following graphs. What is the strength of each graph? Weakness?  
  
# a. Bar graph of marital.  
  
library(ggplot2)  
  
bank\_train <- read.csv(file = "C:/Users/DDY/Desktop/2021-Spring-textbooks/ADS-502/Module2/Website Data Sets/bank\_marketing\_training.csv")  
  
ggplot(bank\_train, aes(marital)) + geom\_bar() + coord\_flip()



# b. Bar graph of marital, with overlay of response.  
  
ggplot(bank\_train, aes(marital)) + geom\_bar(aes(fill = response)) + coord\_flip()



# c. Normalized bar graph of marital, with overlay of response.  
  
ggplot(bank\_train, aes(marital)) + geom\_bar(aes(fill = response),  
 position = "fill") + coord\_flip()



# 22. Using the graph from Exercise 21c, describe the relationship between marital and response.  
# In divorced and married status, the response of "yes" rate is the same and the lowest among all;  
# For unknown status, the response of "yes" rate is in between single and divorced/married;  
# Response rate of "yes" is the highest for single marital status  
  
  
## 23. Do the following with the variables marital and response.  
  
# a. Build a contingency table, being careful to have the correct variables   
# representing the rows and columns. Report the counts and the column percentages.  
  
t.v1 <- table(bank\_train$response, bank\_train$marital)  
t.v2 <- addmargins(A = t.v1, FUN = list(total = sum),quiet = TRUE)  
  
# table without total  
t.v1

##   
## divorced married single unknown  
## no 2743 14579 6514 50  
## yes 312 1608 1061 7

# table with total  
t.v2

##   
## divorced married single unknown total  
## no 2743 14579 6514 50 23886  
## yes 312 1608 1061 7 2988  
## total 3055 16187 7575 57 26874

t.v1\_pct <- round(prop.table(t.v1, margin = 2)\*100, 1)  
t.v2\_pct <- addmargins(A = t.v1\_pct, FUN = list(total = sum),quiet = TRUE)  
  
# percentage table  
t.v1\_pct

##   
## divorced married single unknown  
## no 89.8 90.1 86.0 87.7  
## yes 10.2 9.9 14.0 12.3

# b. Describe what the contingency table is telling you.  
# For response of "no", 'married' has the most percentage;  
# For response of "yes", 'single' has the most percentage.  
  
# 24. Repeat the previous exercise, this time reporting the row percentages. Explain the  
# difference between the interpretation of this table and the previous contingency table.  
  
# swap cols and rows  
t.v1\_r <- table(bank\_train$marital, bank\_train$response)  
t.v2\_r <- addmargins(A = t.v1\_r, FUN = list(total = sum),quiet = TRUE)  
  
t.v1\_r

##   
## no yes  
## divorced 2743 312  
## married 14579 1608  
## single 6514 1061  
## unknown 50 7

t.v2\_r

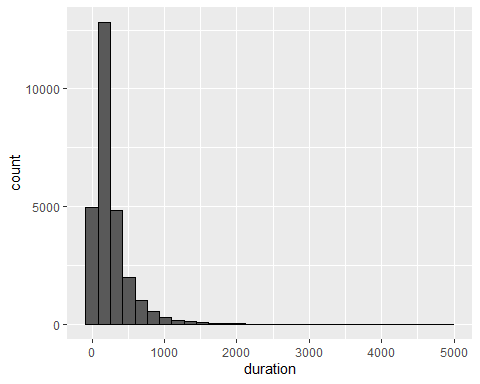
##   
## no yes total  
## divorced 2743 312 3055  
## married 14579 1608 16187  
## single 6514 1061 7575  
## unknown 50 7 57  
## total 23886 2988 26874

t.v1\_r\_pct <- round(prop.table(t.v1\_r, margin = 1)\*100, 1)  
t.v2\_r\_pct <- addmargins(A = t.v1\_r\_pct, FUN = list(total = sum),quiet = TRUE)  
  
t.v1\_r\_pct

##   
## no yes  
## divorced 89.8 10.2  
## married 90.1 9.9  
## single 86.0 14.0  
## unknown 87.7 12.3

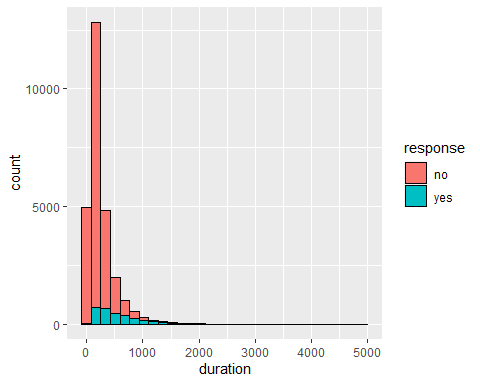
# This time the row percentage shows the ratio in each marital status of response of "yes" and "no";  
# In "divorced", 89.79% responded "no" and 10.21% responded "yes";  
# In "married", 90.07% responded "no" and 9.93% responded "yes";  
# In "single", 85.99% responded "no" and 14.01% responded "yes";  
# In "unknown", 87.72% responded "no" and 12.38% responded "yes";  
# Overall, more people recompensed "no" than "yes".  
  
# The difference between this two tables is one is from the perspective of   
# response while the other is   
# from the perspective of marital status.  
  
  
### 25. Produce the following graphs. What is the strength of each graph? Weakness?  
  
# a. Histogram of duration.  
  
ggplot(bank\_train, aes(duration)) + geom\_histogram(color="black")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



# b. Histogram of duration, with overlay of response.  
  
ggplot(bank\_train, aes(duration)) + geom\_histogram(aes(fill = response), color="black")

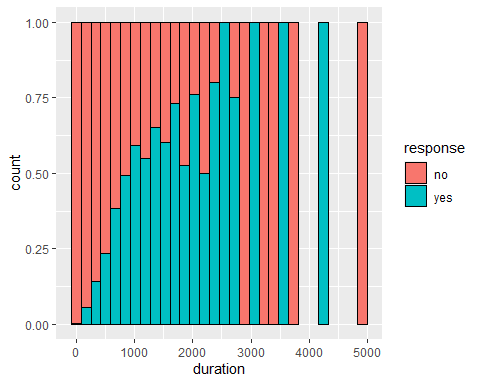
## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.



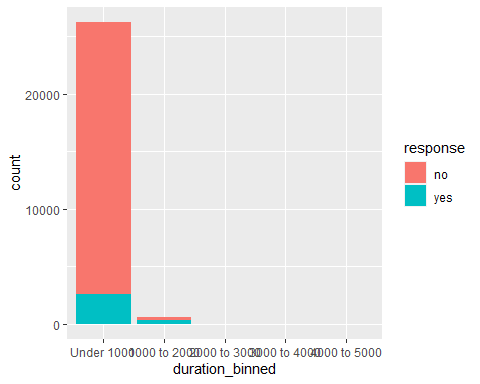
# c. Normalized histogram of duration, with overlay of response.  
  
ggplot(bank\_train, aes(duration)) + geom\_histogram(aes(fill = response), color="black",  
 position = "fill")

## `stat\_bin()` using `bins = 30`. Pick better value with `binwidth`.

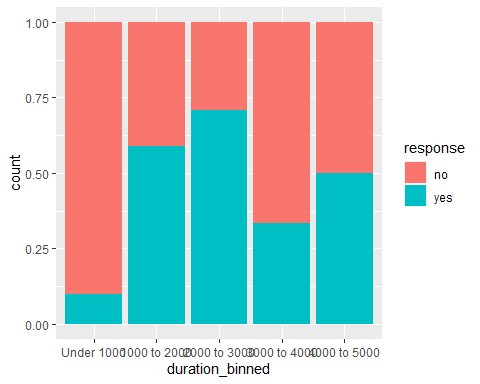
## Warning: Removed 10 rows containing missing values (geom\_bar).



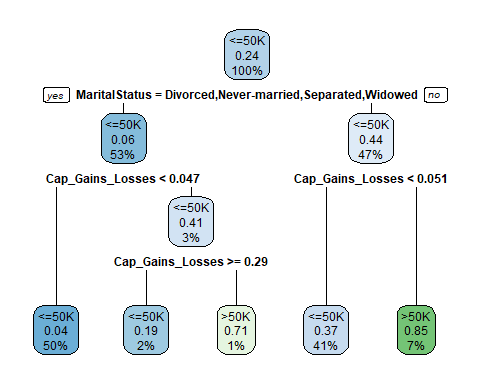
# binned barchart  
  
bank\_train$duration\_binned <- cut(x = bank\_train$duration, breaks = c(0, 1000, 2000, 3000,4000,5000),  
 right = FALSE,  
 labels = c("Under 1000", "1000 to 2000", "2000 to 3000",  
 "3000 to 4000", "4000 to 5000"))  
ggplot(bank\_train, aes(duration\_binned)) + geom\_bar(aes(fill = response))



ggplot(bank\_train, aes(duration\_binned)) + geom\_bar(aes(fill = response), position = 'fill')



# For Exercises 14-20, work with the adult\_ch6\_training and adult\_ch6\_test data  
# sets. Use either Python or R to solve each problem.  
  
# 14. Create a CART model using the training data set that predicts income using  
# marital status and capital gains and losses. Visualize the decision tree   
# (that is, provide the decision tree output). Describe the first few splits in the decision tree.  
  
adult\_training <- read.csv(file = "C:/Users/DDY/Desktop/2021-Spring-textbooks/ADS-502/Module2/Website Data Sets/adult\_ch6\_training")  
  
colnames(adult\_training)[1] <- "MaritalStatus"  
  
# change income and marital status to factors  
adult\_training$Income <- factor(adult\_training$Income)  
adult\_training$MaritalStatus <- factor(adult\_training$MaritalStatus)  
  
  
library(rpart); library(rpart.plot)  
  
# build decision tree  
DT\_CART <- rpart(formula = Income ~ MaritalStatus + Cap\_Gains\_Losses,data =   
 adult\_training, method = "class")  
  
rpart.plot(DT\_CART)

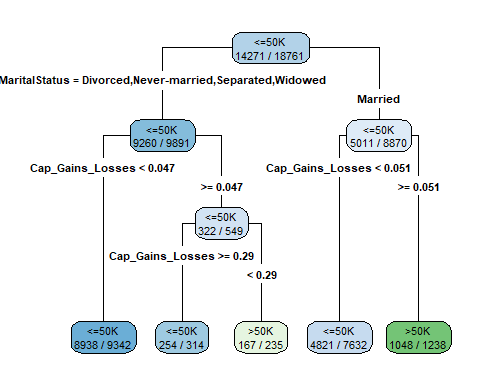


?rpart.plot

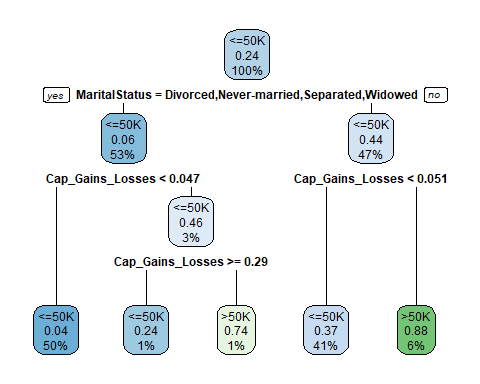
## starting httpd help server ...

## done

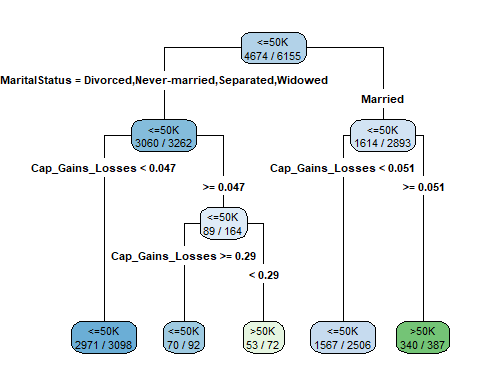
# using type = 4 to label each branch with its specific value, instead of a   
# yes/no at the top of the split  
  
#extra = 2 to add the correct classification proportion to each node.  
  
rpart.plot(DT\_CART, type = 4, extra = 2)



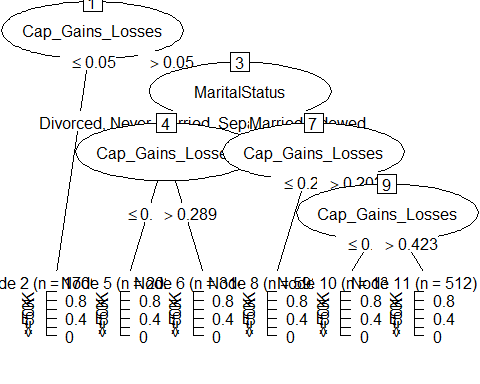
# create a data frame that includes the predictor variables of the records you   
# wish to classify  
X = data.frame(MaritalStatus = adult\_training$MaritalStatus,  
 Cap\_Gains\_Losses =  
 adult\_training$Cap\_Gains\_Losses)  
  
# Once you have the predictor variables you wish to classify, use the predict()  
# command.  
predIncomeCART = predict(object = DT\_CART, newdata = X,type = "class")  
  
  
# 15. Develop a CART model using the test data set that utilizes the same target   
# and predictor variables. Visualize the decision tree. Compare the decision trees.   
# Does the test data result match the training data result?  
  
adult\_test <- read.csv(file = "C:/Users/DDY/Desktop/2021-Spring-textbooks/ADS-502/Module2/Website Data Sets/adult\_ch6\_test")  
  
# run through the same process using test dataset  
colnames(adult\_test)[1] <- "MaritalStatus"  
adult\_test$Income <- factor(adult\_test$Income)  
adult\_test$MaritalStatus <- factor(adult\_test$MaritalStatus)  
DT\_CART\_test <- rpart(formula = Income ~ MaritalStatus + Cap\_Gains\_Losses,data =   
 adult\_test, method = "class")  
  
rpart.plot(DT\_CART\_test)



rpart.plot(DT\_CART\_test, type = 4, extra = 2)



X\_test = data.frame(MaritalStatus = adult\_test$MaritalStatus,  
 Cap\_Gains\_Losses =  
 adult\_test$Cap\_Gains\_Losses)  
  
predIncomeCART\_test = predict(object = DT\_CART\_test, newdata = X\_test,  
 type = "class")  
  
# The decision tree of test dataset matches the one with training dataset.  
  
  
# 16. Use the training data set to build a C5.0 model to predict income using   
# marital status and capital gains and losses. Specify a minimum of 75 cases per  
# terminal node. Visualize the decision tree. Describe the first few splits in the decision tree.  
  
library(C50)  
  
# run c5.0 algo  
C5 <- C5.0(formula = Income ~ MaritalStatus + Cap\_Gains\_Losses,   
 data = adult\_training, control = C5.0Control(minCases=75))  
plot(C5)



#predict(object = C5, newdata = X)  
  
# 17. How does your C5.0 model compare to the CART model? Describe the similarities and differences.  
  
# Similarities: Both CART and C50 follow the similar logic of test conditions;  
# Differences: CART starts the split with marital status and goes on with Cap\_Gains\_Losses  
# while c50 starts with Cap\_Gains\_Losses and goes on with marital status; Different   
# number of nodes and different ways of displaying classes for the leaf nodes.  
  
  
# For the following exercises, work with the bank\_reg\_training and the   
# bank\_reg\_test data sets. Use either Python or R to solve each problem.  
  
# 34. Use the training set to run a regression predicting Credit Score,   
# based on Debt-to-Income Ratio and Request Amount. Obtain a summary of the model.  
# Do both predictors belong in the model?  
  
bank\_reg\_train = read.csv(file ='C:/Users/DDY/Desktop/2021-Spring-textbooks/ADS-502/Module2/Website Data Sets/bank\_reg\_training')  
bank\_reg\_test = read.csv(file ='C:/Users/DDY/Desktop/2021-Spring-textbooks/ADS-502/Module2/Website Data Sets/bank\_reg\_test')  
  
# run the model  
model01 <- lm(formula = Credit.Score ~ Debt.to.Income.Ratio +Request.Amount,  
 data = bank\_reg\_train)  
  
# display the summary table  
summary(model01)

##   
## Call:  
## lm(formula = Credit.Score ~ Debt.to.Income.Ratio + Request.Amount,   
## data = bank\_reg\_train)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -279.13 -25.11 10.87 39.93 175.32   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.685e+02 1.336e+00 500.27 <2e-16 \*\*\*  
## Debt.to.Income.Ratio -4.813e+01 4.785e+00 -10.06 <2e-16 \*\*\*  
## Request.Amount 1.075e-03 6.838e-05 15.73 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 66 on 10690 degrees of freedom  
## Multiple R-squared: 0.02839, Adjusted R-squared: 0.02821   
## F-statistic: 156.2 on 2 and 10690 DF, p-value: < 2.2e-16

# 35. Validate the model from the previous exercise.  
  
model02 <- lm(formula = Credit.Score ~ Debt.to.Income.Ratio + Request.Amount,  
 data = bank\_reg\_test)  
  
summary(model02)

##   
## Call:  
## lm(formula = Credit.Score ~ Debt.to.Income.Ratio + Request.Amount,   
## data = bank\_reg\_test)  
##   
## Residuals:  
## Min 1Q Median 3Q Max   
## -288.16 -24.49 11.08 39.47 199.84   
##   
## Coefficients:  
## Estimate Std. Error t value Pr(>|t|)   
## (Intercept) 6.655e+02 1.328e+00 501.26 <2e-16 \*\*\*  
## Debt.to.Income.Ratio -5.214e+01 4.826e+00 -10.80 <2e-16 \*\*\*  
## Request.Amount 1.302e-03 6.849e-05 19.01 <2e-16 \*\*\*  
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## Residual standard error: 65.78 on 10772 degrees of freedom  
## Multiple R-squared: 0.03845, Adjusted R-squared: 0.03827   
## F-statistic: 215.4 on 2 and 10772 DF, p-value: < 2.2e-16

# Validation complete.  
  
# 36. Use the regression equation to complete this sentence: "The estimated Credit Score equals.."  
# The estimated Credit Score equals y = 668.4562 - 48.1262\* Debt-to-Income Ratio + 0.0011\* Request Amount  
  
# 37. Interpret the coefficient for Debt-to-Income Ratio.  
# The coefficient for Debt-to-Income Ratio is negative which means the lower the  
# Debt-to-Income Ratio, the higher the credit score.  
  
# 38. Interpret the coefficient for Request Amount.  
# The coefficient for Request Amount is positive which means the higher the   
# Request Amount, the higher the credit score.  
  
# 39. Find and interpret the value of s.  
# Residual standard error: 65.78 on 10772 degrees of freedom. The size of model   
# prediction error is 65.8 (66), that is the difference between the actual   
# credit score and of which predicated from the model.  
  
# 40. Find and interpret Radj2 . Comment.  
# The adjusted R squared value is modified version of R-squared that has been   
# adjusted for the number of predictors in the model. It increases when the new   
# term improves the model more than would be expected by chance. It decreases   
# when a predictor improves the model by less than expected. The R-adj^2 is 0.028   
# from the model. This means that 2.8% of the variability in Credit Score is   
# accounted for by the predictors Debt-to-Income Ratio and Request Amount.  
  
# 41. Find MAE\_Baseline and MAE\_Regression, and determine whether the regression   
# model outperformed its baseline model.  
  
# use the predicators from the test dataset to predict  
X\_test <- data.frame(Debt.to.Income.Ratio = bank\_reg\_test$Debt.to.Income.Ratio,   
 Request.Amount = bank\_reg\_test$Request.Amount)  
  
# y predicated using the model from the test dataset  
ypred <- predict(object = model02, newdata = X\_test)  
  
# compare to the actual targets from the test dataset  
ytrue <- bank\_reg\_test$Credit.Score  
  
library(MLmetrics)

##   
## Attaching package: 'MLmetrics'

## The following object is masked from 'package:base':  
##   
## Recall

# mean absolute error from regression  
MAE\_Regression = MAE(y\_pred = ypred, y\_true = ytrue)  
  
# mean absolute error from baseline using the formula  
y\_y\_bar = abs(bank\_reg\_test$Credit.Score - mean(bank\_reg\_test$Credit.Score))  
MAE\_Baseline = sum(y\_y\_bar)/length(y\_y\_bar)  
  
MAE\_Regression

## [1] 48.01625

MAE\_Baseline

## [1] 48.60024

# So the MAE\_Regression is 48.02 and the MAE\_Baseline is 48.60.   
# Since MAE\_Regression < MAE\_Baseline, thus, our regression model outperformed its baseline model.