Module2_R_HW

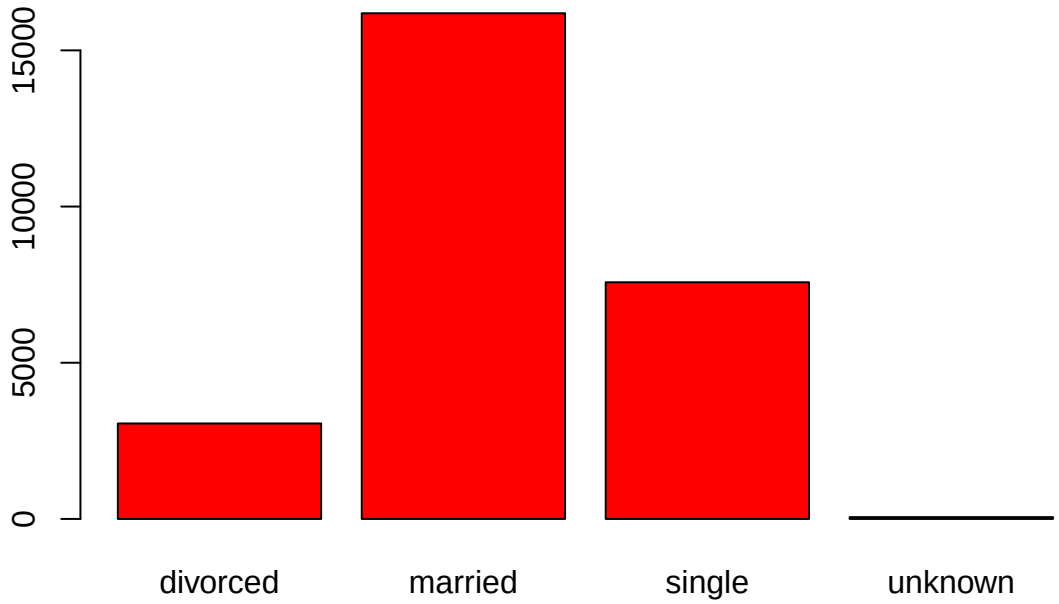
Filipp Krasovsky

3/15/2021

Questions for Chapter 4:#21, 22, 23, 24, & 25 Dataset to use: bank_marketing_training

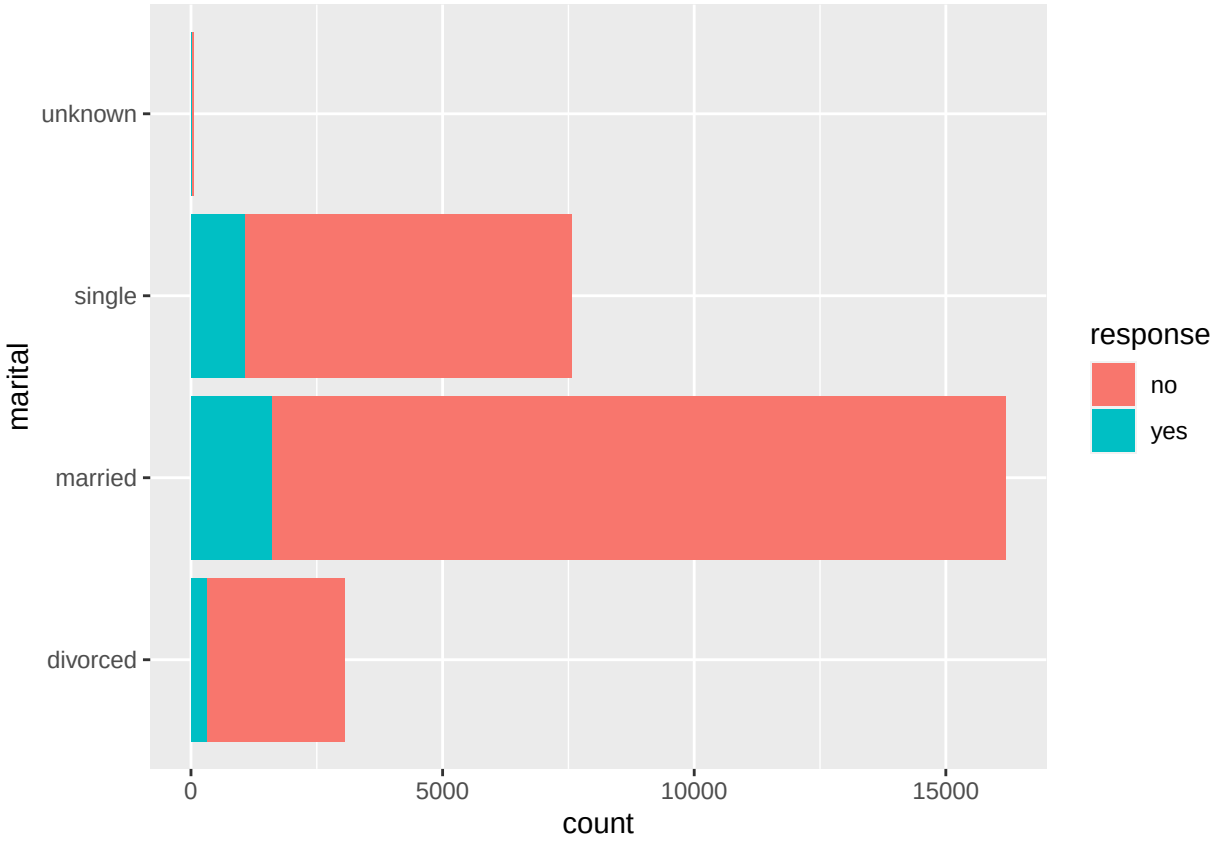
Question 21: a. Bar graph of marital. b. Bar graph of marital, with overlay of response. c. Normalized bar graph of marital, with overlay of response.

```
#a bar graph of marital
barplot(table(df$marital),col = "red")
```



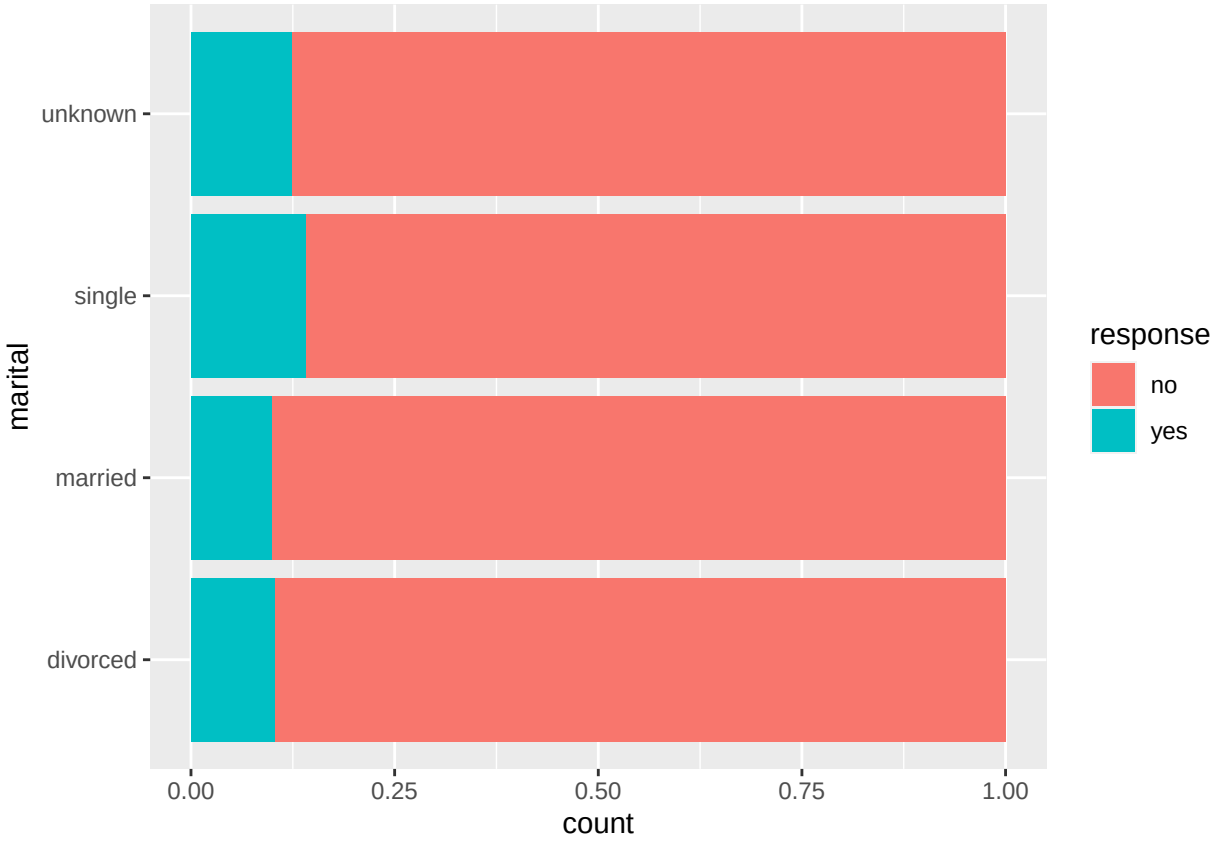
Strength: Shows us raw distribution of variable for futher analysis Weakness: Does not show relationship to target variable.

```
#b bar graph of marital with overlay of response
    ggplot(df, aes(marital)) + geom_bar(aes(fill = response)) + coord_flip()
```



Strength: Shows relationship with target variable Weakness: Lack of normalization obscures relationship

```
#normalized bar graph of marital with overlay of response
          ggplot(df, aes(marital)) + geom_bar(aes(fill = response),position = "fill") + coord_flip()
```



Strength: Normalization makes relationship between response and predictor variables clearer. Weakness: Does not show us distribution of predictor variable.

Question 22: Using the graph from Exercise 21c, describe the relationship between marital and response.

Since the 'unknown' marital status does not provide much insight on face value, we are left comparing single, married, and divorced individuals. In these instances, Single individuals have a significantly higher rate of responding "yes" than married or divorced individuals. That being said, individuals who are not identifiably married or divorced also had a higher response rate of "yes".

Question 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. b. Describe what the contingency table is telling you.

```
        #the addmargins function creates a row and column to table A=t.v1 called "total" which contains
        t.v1 <- table(df$response, df$marital)
        t.v2 <- addmargins(A = t.v1, FUN = list(total = sum),quiet = TRUE)
        #print with percentages and without
        print(t.v1)
```

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

```
        print(round(prop.table(t.v1, margin = 2)*100, 1))
```

```
##
```

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

The contingency table shows us the same story, largely, as the graph - Single people and those with no clear marital status were more likely to respond "yes" than their married and divorced counterparts by 2-5%, depending on the comparrison.

Question 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.

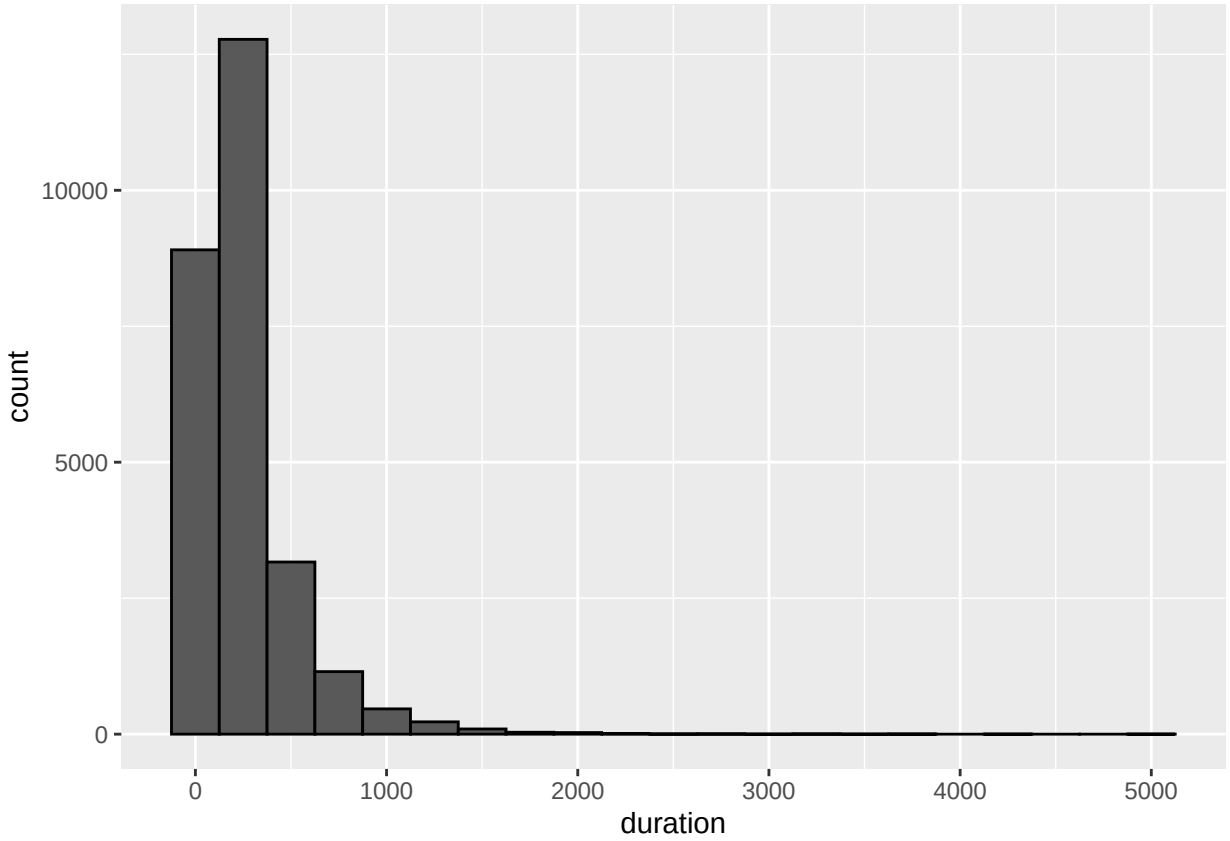
```
t.v1 <- table(df$response,df$marital)
t.v2 <- addmargins(A = t.v1, FUN = list(total = sum),quiet = TRUE)
print(round(prop.table(t.v1, margin = 1)*100, 1))
```

```
##
##       divorced married single unknown
##   no      11.5    61.0   27.3     0.2
##   yes     10.4    53.8   35.5     0.2
```

The difference in interpretation is that here we're examining the demographic composition of those who said "yes" and "no", versus the analysis of the previous table, which was concerned with identifying the portion of each marital group that responded a certain way. Here, we can report that the largest portion of individuals who responded "no" and "yes" were both married people, while in the previous report we identified that single people are more likely to respond "yes".

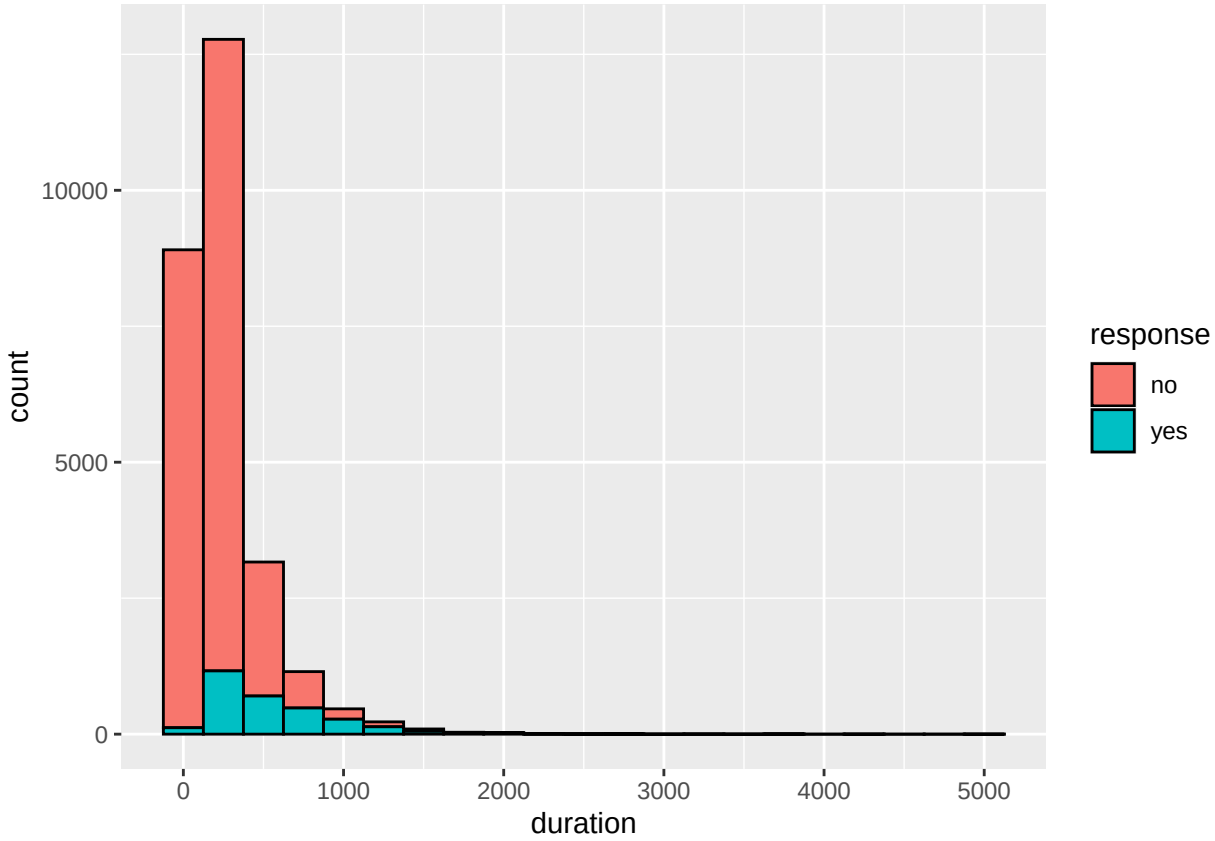
Question 25. Produce the following graphs. What is the strength of each graph? Weakness? a. Histogram of duration. b. Histogram of duration, with overlay of response. c. Normalized histogram of duration, with overlay of response.

```
#histogram of duration
ggplot(df, aes(duration)) +
geom_histogram( color="black",binwidth = 250)
```



Strength: Shows us distribution of raw data weakness: does not show relationship to target variable

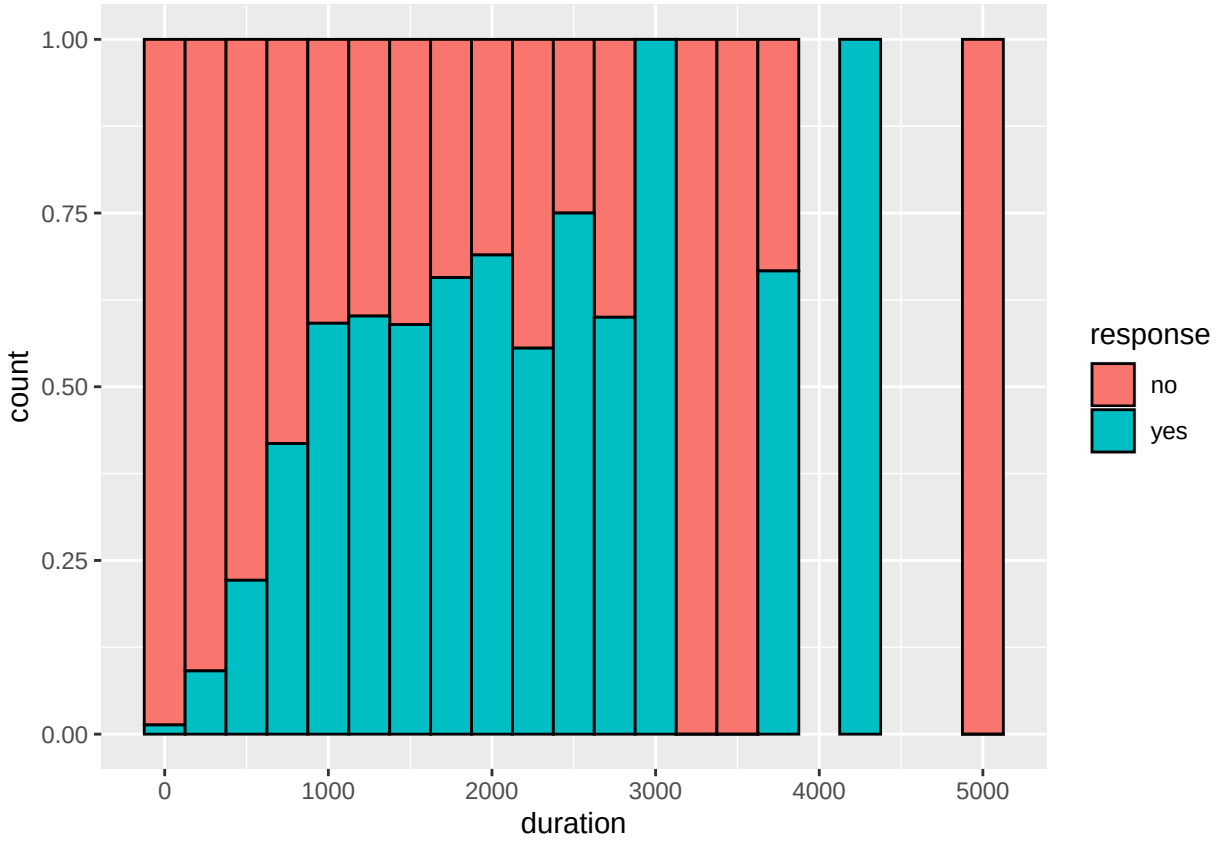
```
#histogram of duration with overlay of response
ggplot(df, aes(duration)) +
geom_histogram(aes(fill = response), color="black",binwidth = 250)
```



Strength: Shows relationship with target variable Weakness: Does not show proportion of each bin belonging to subgroup of the response variable.

```
#normalized histogram of duration with overlay of response.
ggplot(df, aes(duration)) +
geom_histogram(aes(fill = response), color="black",binwidth = 250,position="fill")
```

```
## Warning: Removed 6 rows containing missing values (geom_bar).
```



Strength: Shows relationship between target and predictor variable Weakness: Does not show distribution of predictor variable

Chapter 6 Questions #14, 15, 16, & 17 adult_ch6_training and adult_ch6_test data sets

```
#load in data sets
file_dir = "C:/Users/Filipp/Documents/usd_data_sci/502_data mining/module1/Website Data Sets"
setwd(file_dir)
train = read.csv("adult_ch6_training")
test = read.csv("adult_ch6_test")

library(rpart)
```

```
## Warning: package 'rpart' was built under R version 4.0.4
```

```
library(rpart.plot)
```

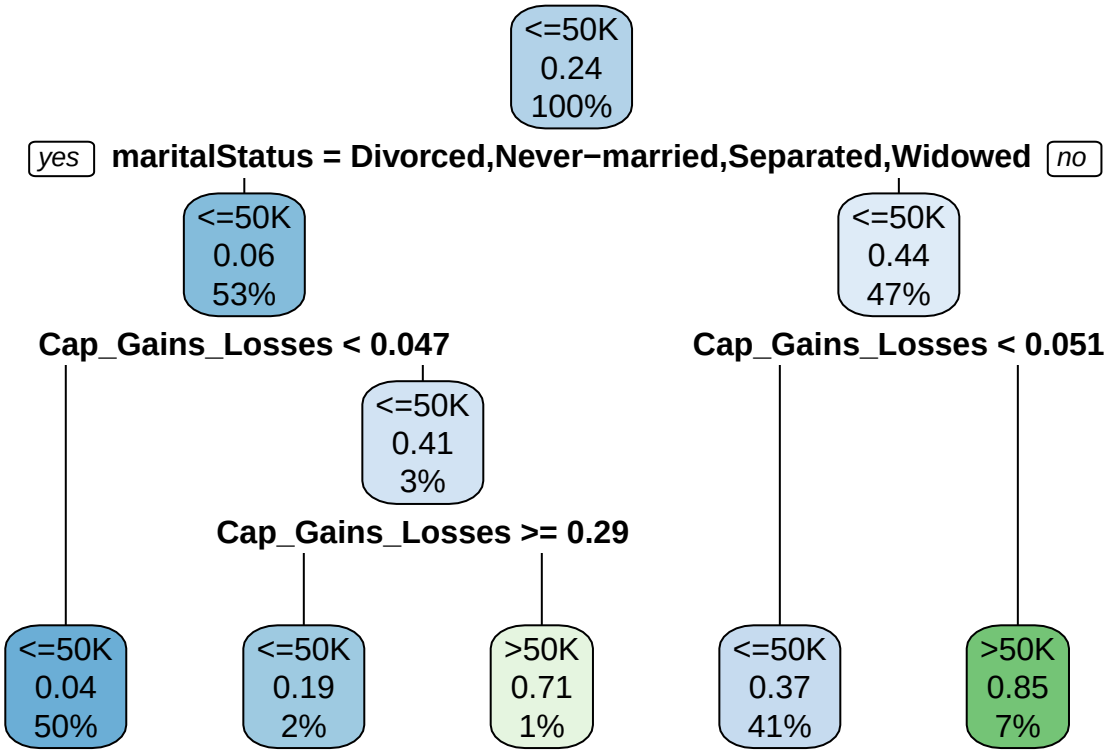
```
## Warning: package 'rpart.plot' was built under R version 4.0.4
```

Question 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.

```
#using marital status and capital gains and losses
#factorize
colnames(train)[1] = "maritalStatus"
```

```
train$maritalStatus = factor(train$maritalStatus)
train$Income = factor(train$Income)

cart01 <- rpart(formula = Income ~ maritalStatus + Cap_Gains_Losses,data = train, method = "class")
rpart.plot(cart01)
```



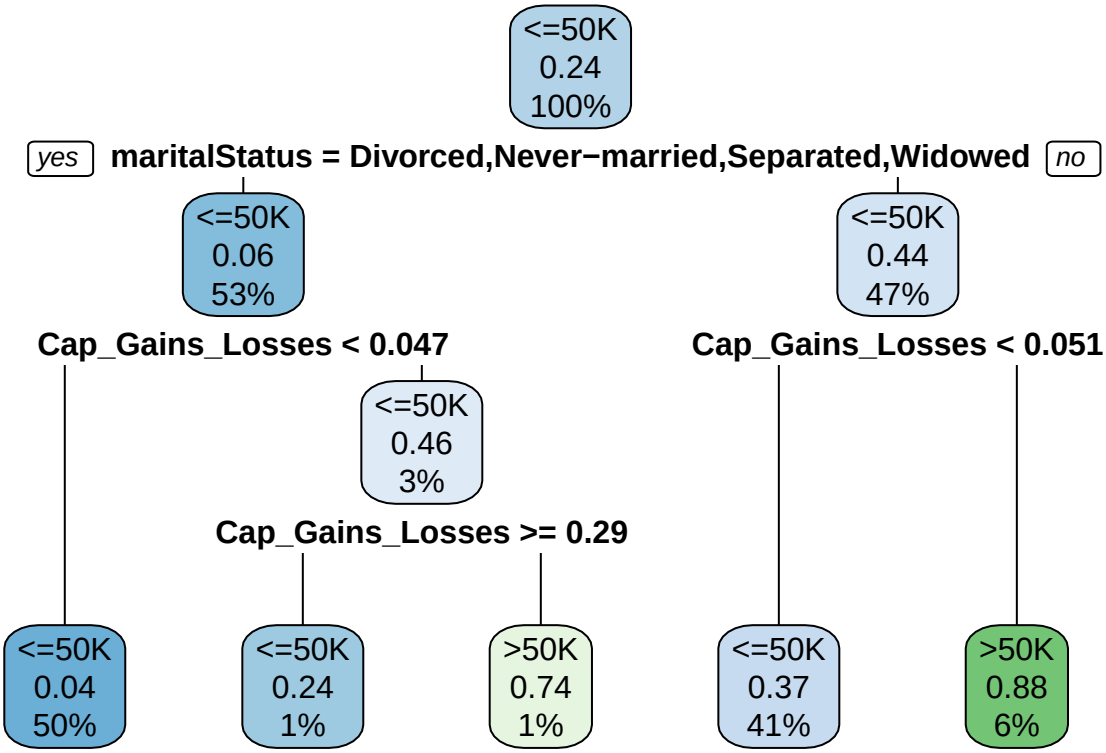
From the root node, we can confirm that about 24% of our dataset has an income of >50k. Those are not married comprise about 53% of our dataaset, and 6% of those individuals have an income of >50k. Similarly, Married people comprise 47% of our data, and 47% of those individuals have an income >50k. Non-married individuals with a capital gains loss of less than 4.7% make up 50% of our total data, and 4% of this subgroup has an income of >50k, which is one of our terminal nodes.

44% of our married group has a high income, while only 6% of our non-married group has a high income (>50k).

Question 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?

```
colnames(test)[1] = "maritalStatus"
test$maritalStatus = factor(test$maritalStatus)
test$Income = factor(test$Income)

cart01 <- rpart(formula = Income ~ maritalStatus + Cap_Gains_Losses,data = test, method = "class")
rpart.plot(cart01)
```



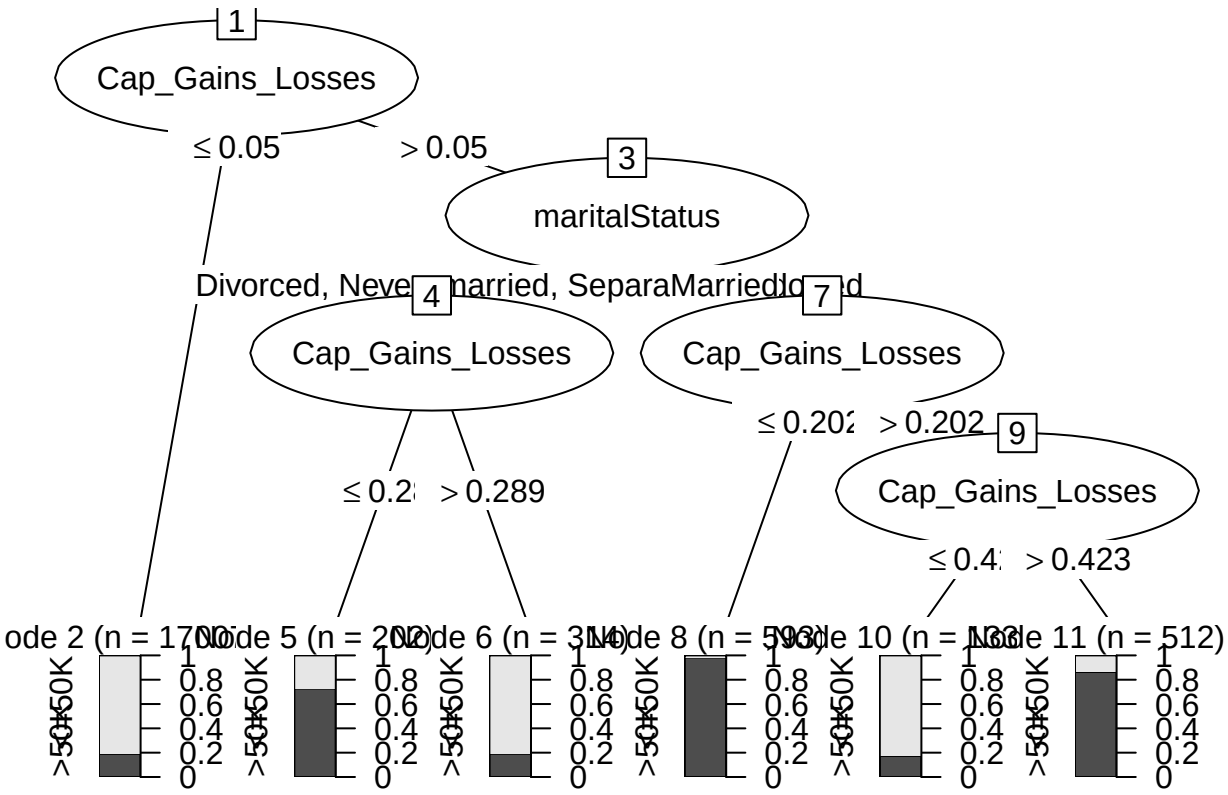
Yes, the decision trees are similar - the only discrepancy occurs at the terminal nodes for non-married instances with a capital gains loss greater than or equal to 4.7% and less than 29%, wherein the proportion of high income individuals is 5 percent greater than in the training set.

Question 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)
```

```
## Warning: package 'C50' was built under R version 4.0.4
```

```
C5 <- C5.0(formula = Income ~ maritalStatus + Cap_Gains_Losses, data = train, control = C5.0Control(min
plot(C5)
```



Observing the root node, we find that individuals who have less than 5% in capital gains losses immediately terminate in a node with a majority low-income bracket. Non-married individuals with a capital gains loss of less than 28% have a noticably high concentration of high-income individuals than those with a capital gains loss of more than 28.9%. Married individuals with a capital gains loss smaller than 20 percent also have a majority high-income composition.

Question 17 How does your C5.0 model compare to the CART model? Describe the similarities and differences.

Similarities: Both have roughly the same number of terminal nodes, and the C5 privileges the marital status variable over capital gains losses in all cases except one. Some of the bins have overlap in the modeling process, such as 4.7% being a split in the first level of our CART model and C5 using 5%.

Differences: C5 bins a large chunk of our dataset into the <0.05 CGL category without doing any further analysis on marital status, ending in a terminal node right way, which our CART model doesn't engage in. Furthermore, our CART model has fewer levels thn the C5 model, which goes up to four intermediary nodes before terminating. Our C5 model also has 3 instances of high-frequency concentration of the >50k income group, whereas our CART model only has two.

Chapter 11 Questions 34-41

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.

```
#load in our datasets
file_dir = "C:/Users/Filipp/Documents/usd_data_sci/502_data mining/module1/Website Data Sets"
setwd(file_dir)
train = read.csv("bank_reg_training")
test = read.csv("bank_reg_test")
```

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?

```
#predict credit score based on debt-to-income and request amount.
#we don't need to any factorization since we're using continuous variables all around.
#train$Request.Amount = scale(x = train$Request.Amount)
train.model = lm(formula = Credit.Score ~ Debt.to.Income.Ratio + Request.Amount, data = train)
summary(train.model)
```

```
##
## Call:
## lm(formula = Credit.Score ~ Debt.to.Income.Ratio + Request.Amount,
##     data = 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
```

While our model performs relatively poorly with an R-squared value of ~0.03, both variables are statistically significant. Scaling down the request amount and standardizing seems to have no effect on perforamnce.

35. Validate the model from the previous exercise.

```
model.validation <- lm(formula = Credit.Score ~ Debt.to.Income.Ratio +Request.Amount, data = test)
summary(model.validation)
```

```
##
## Call:
## lm(formula = Credit.Score ~ Debt.to.Income.Ratio + Request.Amount,
##     data = 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
```

The validation confirms that all parameters are significant and the R squared values are largely similar.

36. Use the regression equation to complete this sentence: "The estimated Credit Score equals. . . ."

```
train.model$coefficients
```

```
##           (Intercept) Debt.to.Income.Ratio       Request.Amount
##        668.456242903        -48.126164151          0.001075439
```

The estimated credit score equals 682.8 minus the debt-to-income ratio times 48 plus 10 times the request amount.

37. Interpret the coefficient for Debt-to-Income Ratio.

a one-unit increase in the debt to income ratio, that is, as an individual takes on more debt relative to their income, are associated with a 48-point drop in their credit score.

38. Interpret the coefficient for Request Amount.

A one-unit increase in the request amount is associated with a 0.001 point increase in the credit score.

39. Find and interpret the value of s.

s, or the square root of the MSE:

```
#get the MSE
sqrt(mean(train.model$residuals^2))
```

```
## [1] 65.99269
```

The standard error is 66, which can be interpret to mean that the typical prediction error around a point estimate is 66 credit score points.

40. Find and interpret adjusted R^2. Comment.

The adjusted R-squared is 0.02821, which isn't particularly different from our Multiple R-squared of 0.02839 since we're not using a large amount of parameters for our multiple regression.

41. Find MAEBaseline and MAERegression, and determine whether the regression model outperformed its baseline model.

```r
#predict with test data
model.predict <- predict(train.model,newdata = test)
error = test$Credit.Score - model.predict
MAE = mean(abs(error))
print(MAE)
```

```
## [1] 47.79067
```

```r
#get baseline MAE
error_base = mean(test$Credit.Score) - test$Credit.Score
MAE_base = mean(abs(error_base))
print(MAE_base)
```

```
## [1] 48.60024
```

There isn't a significant difference between the baseline performance and our model performance, although we technically outperform on a very slight margin.