Bank Marketing Classification

Eliot Jones

6/21/2020

Introduction

The dataset used for this project is entitled "Bank Marketing Data Set," and is found in the UCI Machine Learning Repository. Before getting into exploring the dataset, I will provide some brief background on the dataset, and what the machine learning algorithms will determine. The goal of this project is to create a machine learning algorithm that can predict whether or not a person will make a term deposit, and what factors are most important when making this prediction.

Background

The dataset is taken from a Portuguese bank, that used a phone-based marketing campaign to see if their clients would like to make bank term deposits. Term deposits involve the client depositing their money in the bank for a set period of time ("term"), and then when the term is up they may withdraw the funds. Because the money is non-withdrawable (usually) for the term, banks can then understand how much money they can lend out at any given time, which can prove to be very useful. In this version of the dataset, there are 17 columns, 16 for the input variables of the dataset and one for the outcome of the marketing campaign. The overview of these variables from the dataset location are as follows:

1 - age (numeric) 2 - job : type of job (categorical: 'admin.','blue-collar','entrepreneur','housemaid','management','retired','self-employed','services','student','technician','unemployed','unknown') 3 - marital : marital status (categorical: 'divorced','married','single','unknown'; note: 'divorced' means divorced or widowed) 4 - education (categorical: 'basic.4y','basic.6y','basic.9y','high.school','illiterate','professional.course','university.degree','unknown') 5 - default: has credit in default? (categorical: 'no','yes','unknown') 6 - balance: the balance in the account in question (numeric) 7 - housing: has housing loan? (categorical: 'no','yes','unknown') 8 - loan: has personal loan? (categorical: 'no','yes','unknown')

Related with last contact of current campaign:

9 - contact: contact communication type (categorical: 'cellular','telephone') 10 - day_of_week: last contact day of the week (categorical: 'mon','tue','wed','thu','fri') 11 - month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec') 12 - duration: last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call y is obviously known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.

Other Attributes:

13 - campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact) 14 - pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted) 15 - previous: number of contacts performed before this campaign and for this client (numeric) 16 - poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')

The Output Variable

17 - y - has the client subscribed a term deposit? (binary: 'yes','no')

Exploration

Basics

Getting to know the dataset in the beginning is fairly straightforward, simply finding out some information about the numeric data will help us out in the future.

```
mean(bank_full$age)
```

```
## [1] 40.93621
```

```
sd(bank_full$age)
```

```
## [1] 10.61876
```

```
mean(bank_full$balance)
```

```
## [1] 1362.272
```

```
sd(bank_full$balance)
```

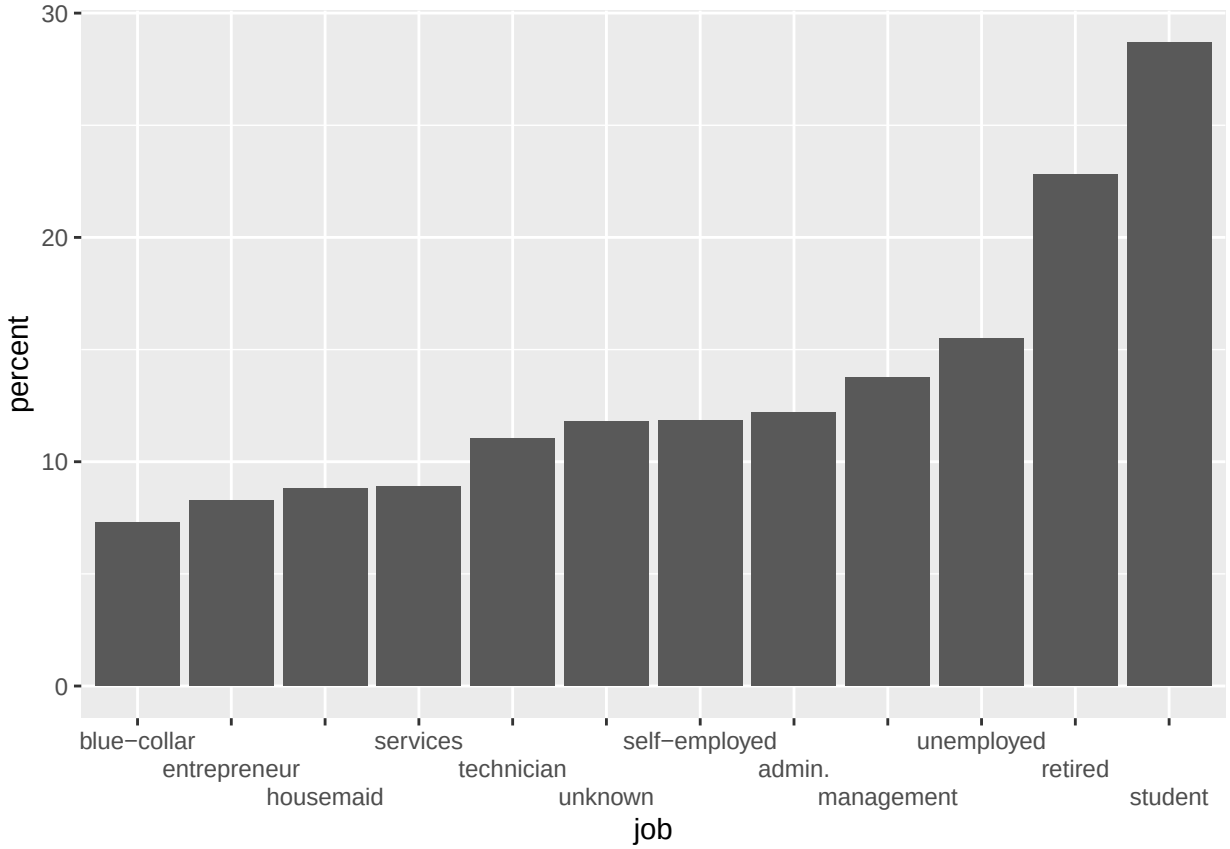
```
## [1] 3044.766
```

These are all insights that will be useful later. Also, as mentioned in the overview of the variables, there is a very high correlation between duration and outcome, which is why it will be dropped when the algorithm is created.

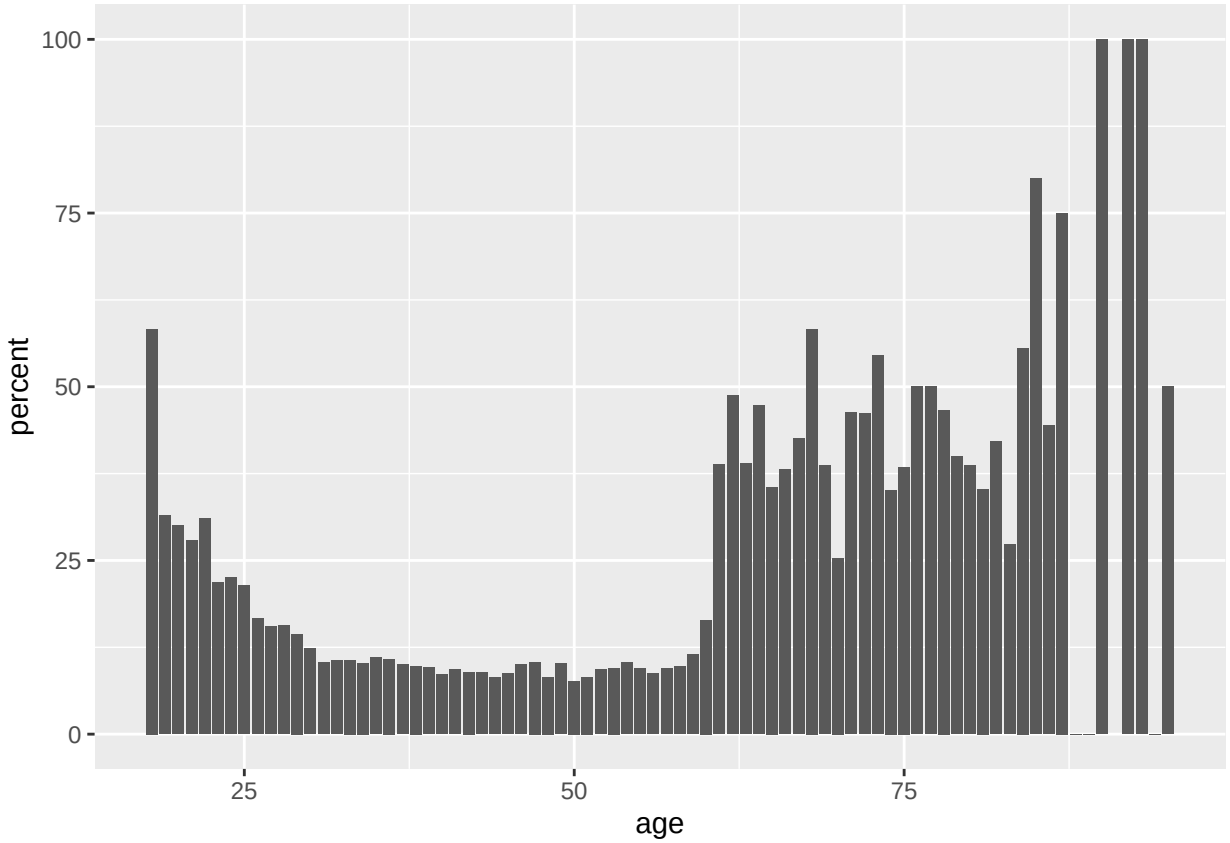
Visualization

Finding visual relationships between the various input variables will be important in determining which of the variables will be most useful in creating the final machine learning algorithm.

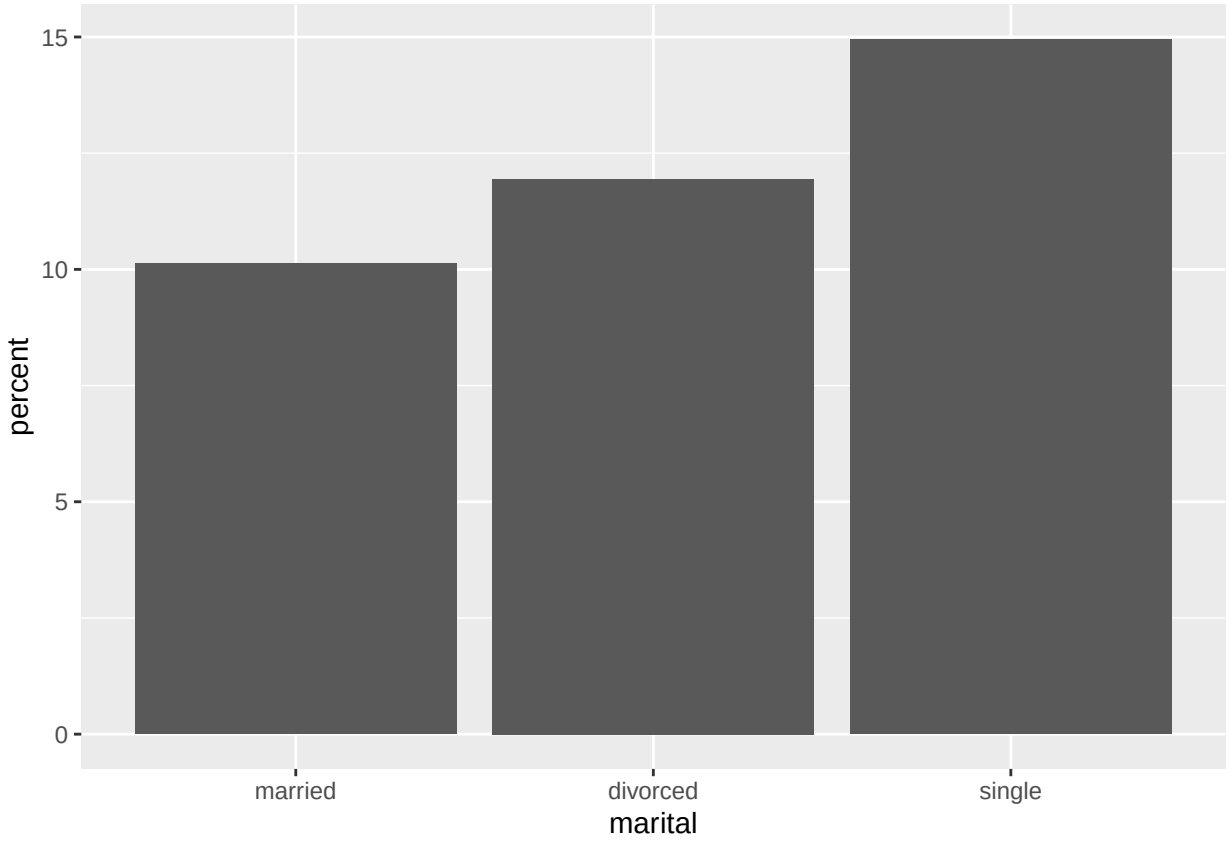
```
## `summarise()` ungrouping output (override with `.groups` argument)
```



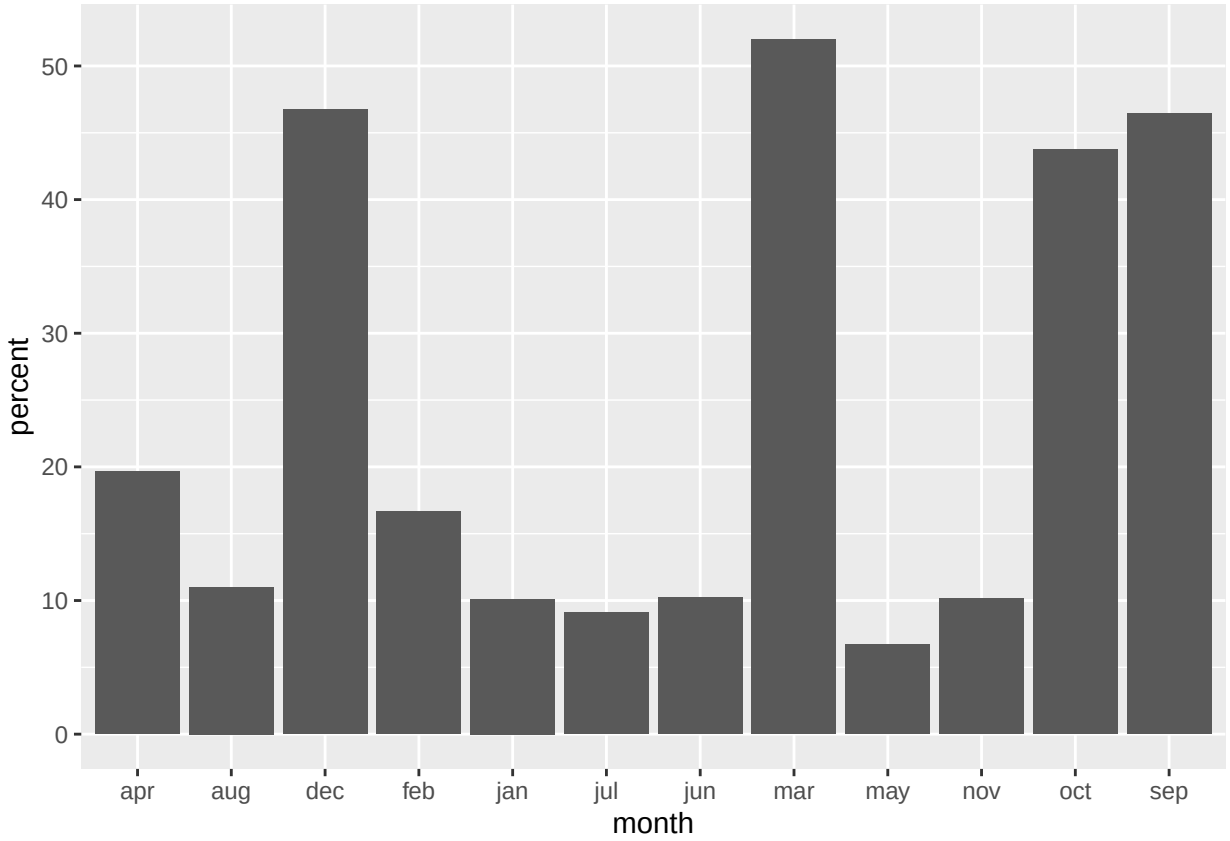

```
## `summarise()` ungrouping output (override with `.groups` argument)
```



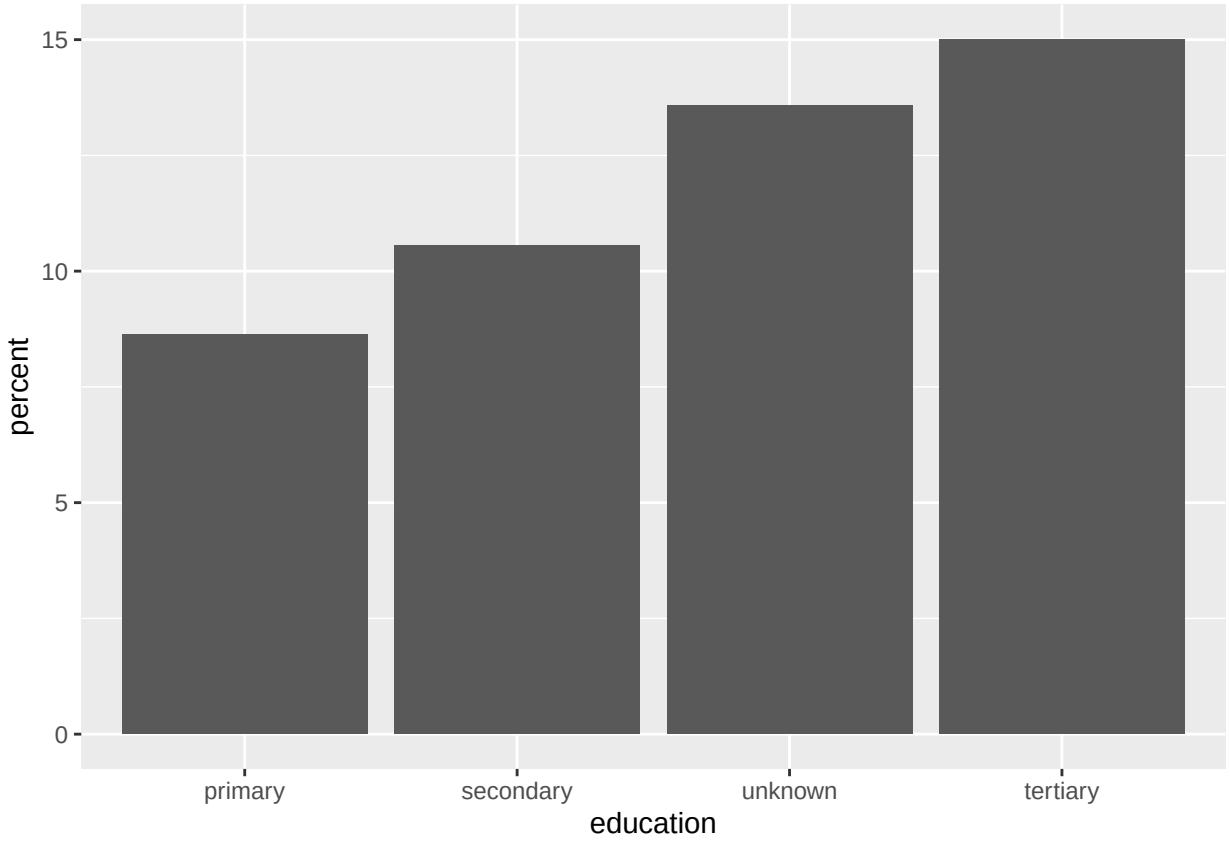

```
## `summarise()` ungrouping output (override with `.groups` argument)
```




```
## `summarise()` ungrouping output (override with `.groups` argument)
```

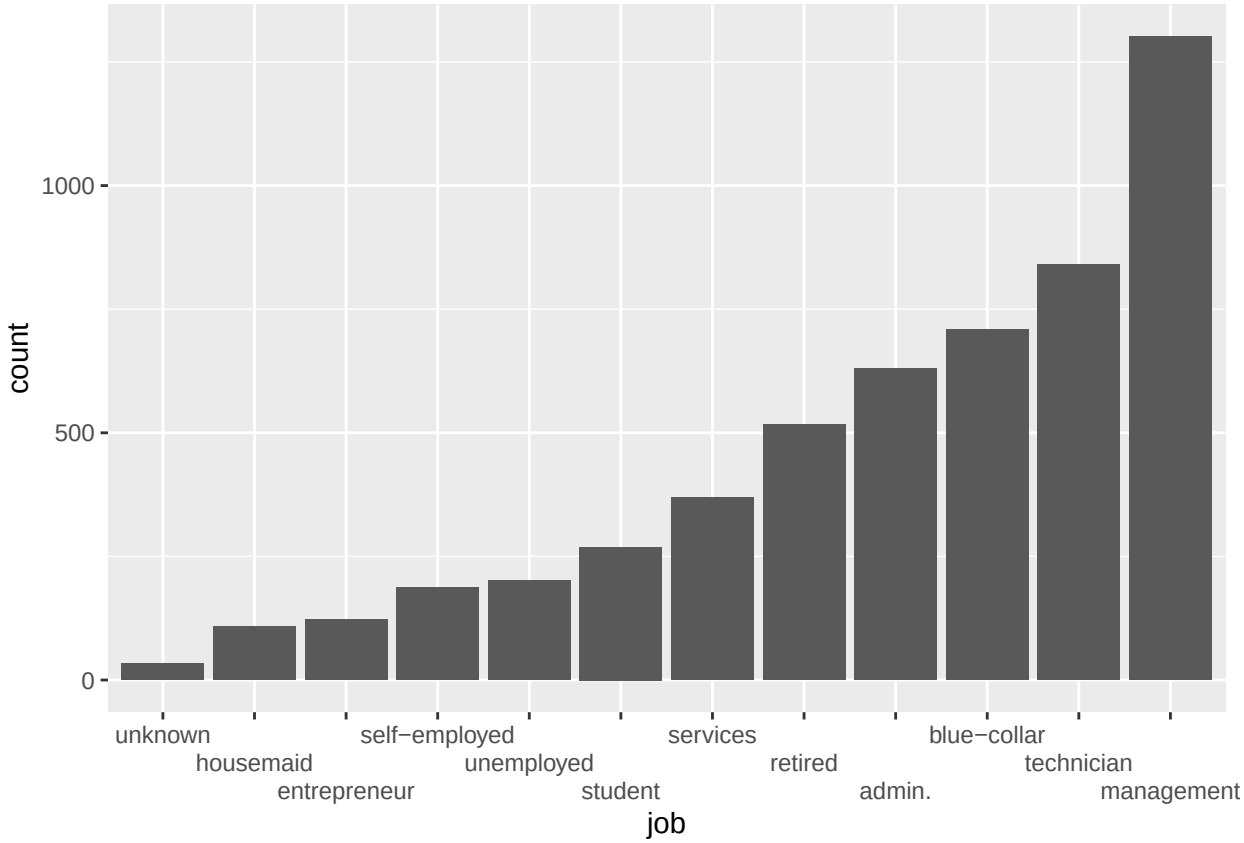



```
## `summarise()` ungrouping output (override with `.groups` argument)
```

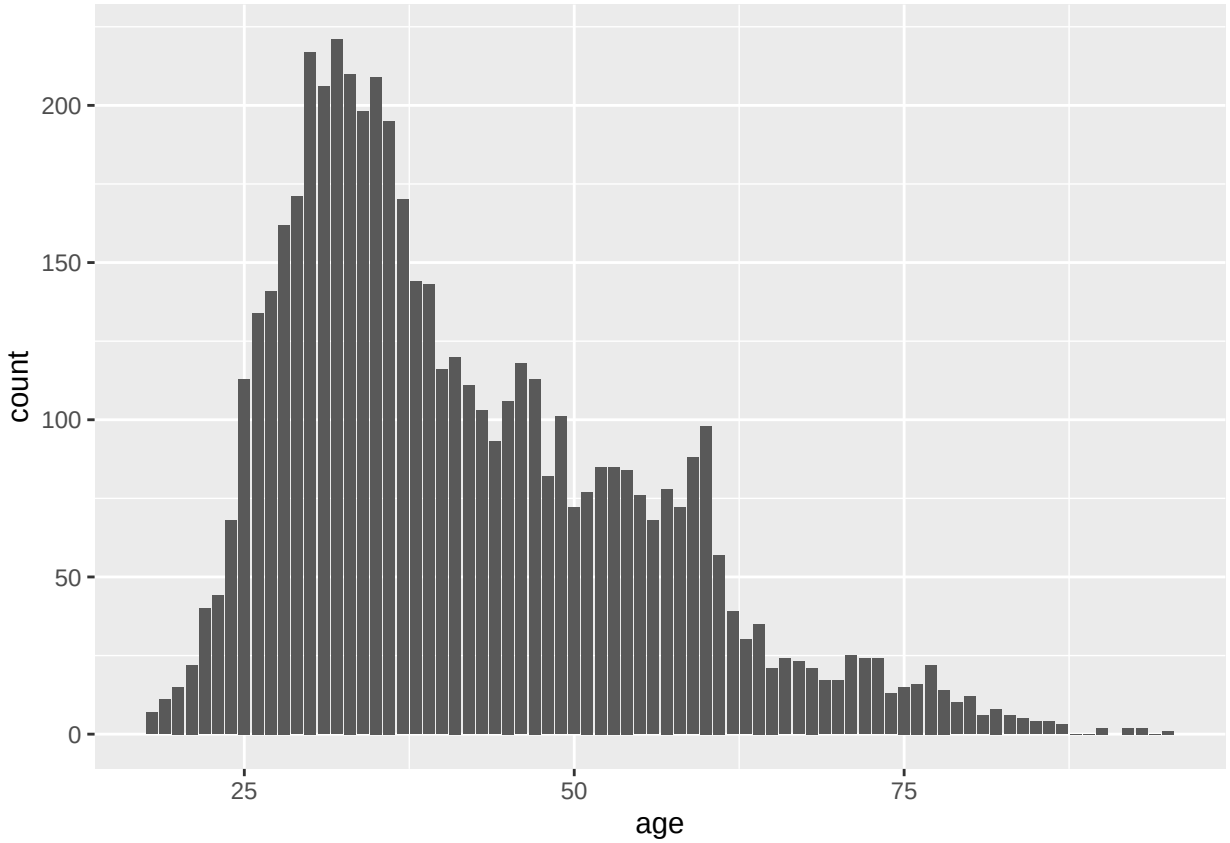


The previous graphs are all percentages, while the following graphs are all in terms of count:

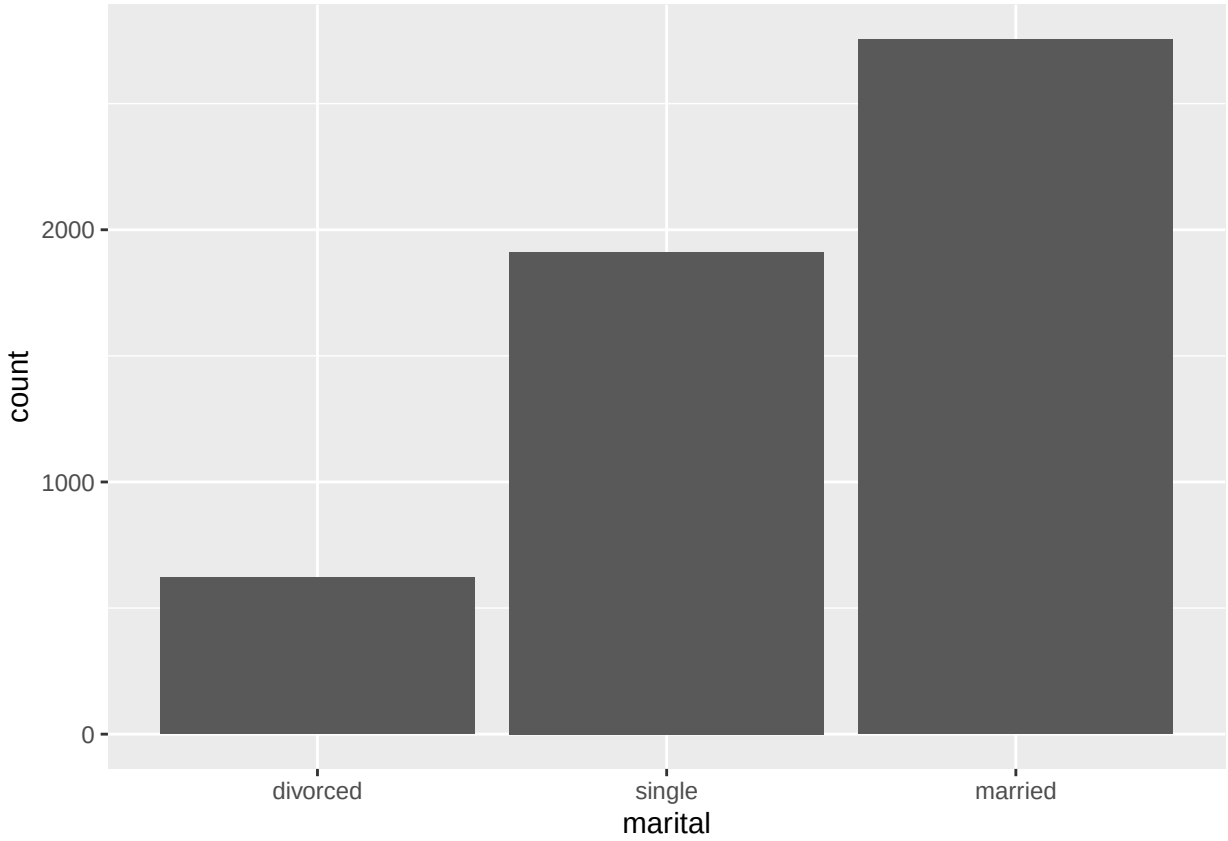
```
## `summarise()` ungrouping output (override with `.groups` argument)
```



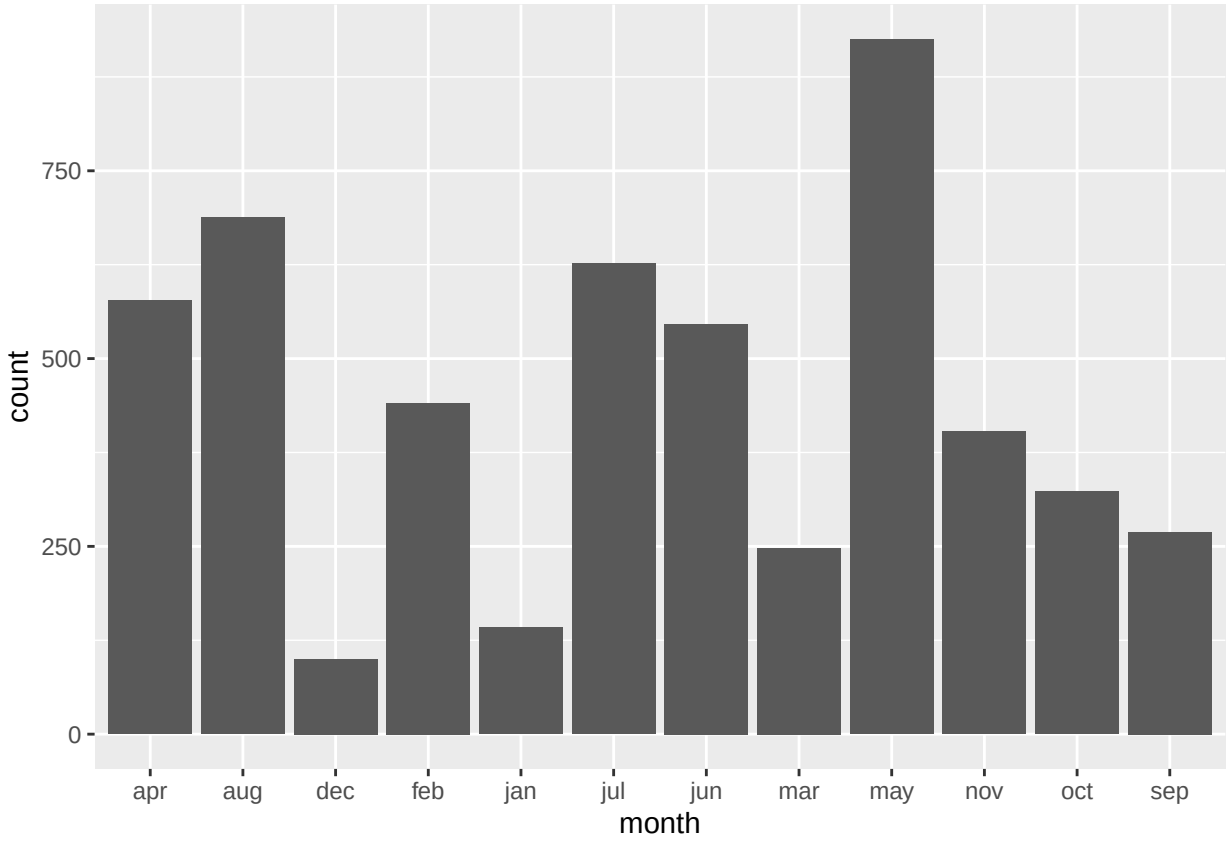

```
## `summarise()` ungrouping output (override with `.groups` argument)
```



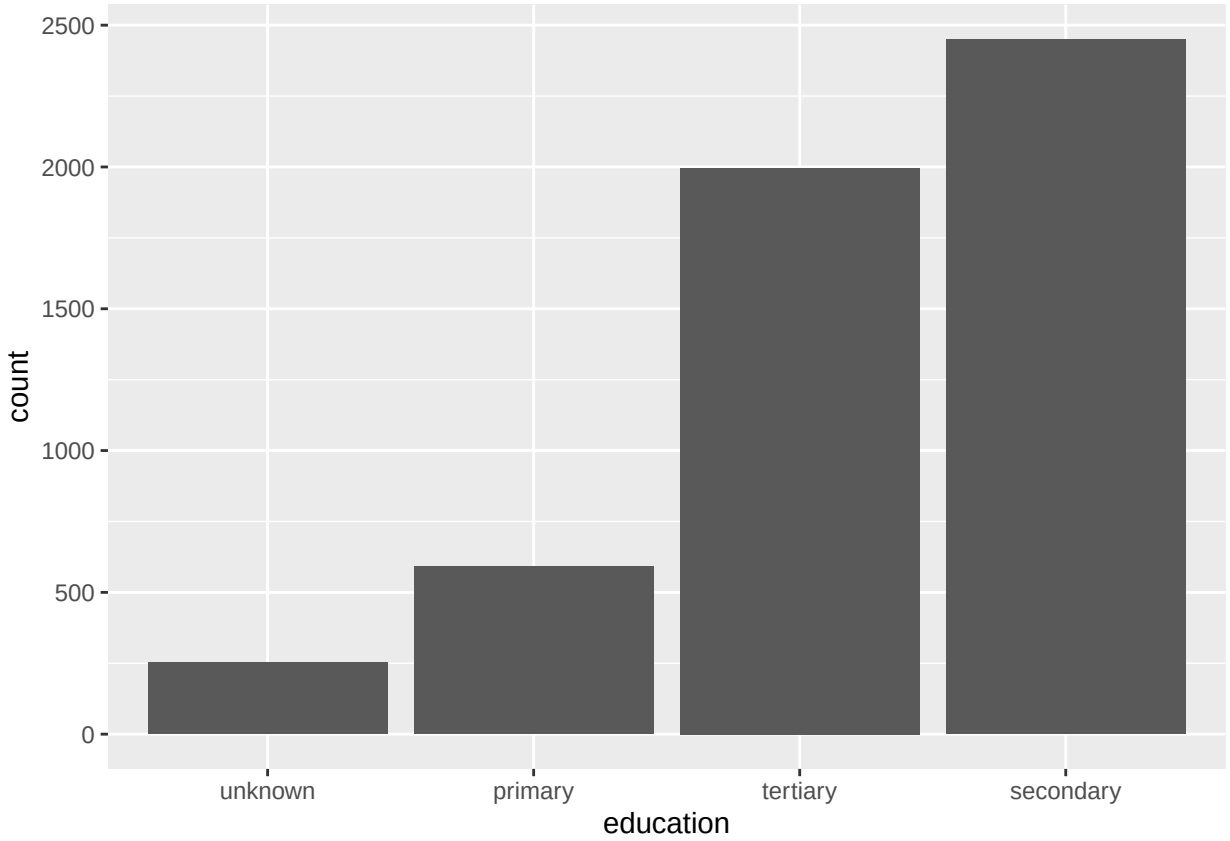

```
## `summarise()` ungrouping output (override with `.groups` argument)
```




```
## `summarise()` ungrouping output (override with `.groups` argument)
```




```
## `summarise()` ungrouping output (override with `.groups` argument)
```



Simply from these two sets of comparisons, we begin to see a few patterns which may have some type of influence on whether or not a client is likely to open a term deposit with the bank. Let's break these observations down by category: job, age, marital status, month, and education.

Job

When looking at the liklihood of people with different occupations to open a term deposit, students, retirees, and the unemployed all have the highest percentages of their population who have opened deposits. Coming in with the fourth highest percentage are those in management, who have the highest count of people who have opened deposits. Because of the high interest rates of term deposits, and the fact that both the retired and students don't have as much need for cash on hand, it makes sense that these two groups of people would be the most likely to open a term deposit with the bank.

Age

The age data is very similar to the job data. When looking at the first set of plots, 18-24 year olds and those from their 60s onward are the most likely to subscribe to a term deposit. This makes sense, because most 18-24 year olds are students, and most people in their 60s and beyond are retired, another data point which supports the conclusion that those that are retired and those who are students are the most likely to open this type of deposit.

Marital Status

Because young people are likelier than middle-aged people to subscribe to a term deposit, it makes sense that singles have a higher percentage of subscription than either married people or divorced people. Simply put, the low number of divorcees that have this type of subscription makes sense, because divorcees often have

less money than younger, single people or married couples. The reverse logic is true for married couples; because they have more money, they can afford to make more term deposits.

Month

March, September, October, and December all have the highest percentages of clients who subscribed to term deposits, but these months all have relatively low counts of subscriptions. The months with the highest amount of people who subscribed to term deposits are April, May, June, and July, but these all have low percentages of people actually subscribing to term deposits. Let's find out if the bank makes more calls in the summer than the fall and winter:

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

```
## # A tibble: 12 x 2
##    month     n
##    <fct> <int>
##  1 apr    2932
##  2 aug    6247
##  3 dec     214
##  4 feb    2649
##  5 jan    1403
##  6 jul    6895
##  7 jun    5341
##  8 mar     477
##  9 may   13766
## 10 nov    3970
## 11 oct     738
## 12 sep     579
```

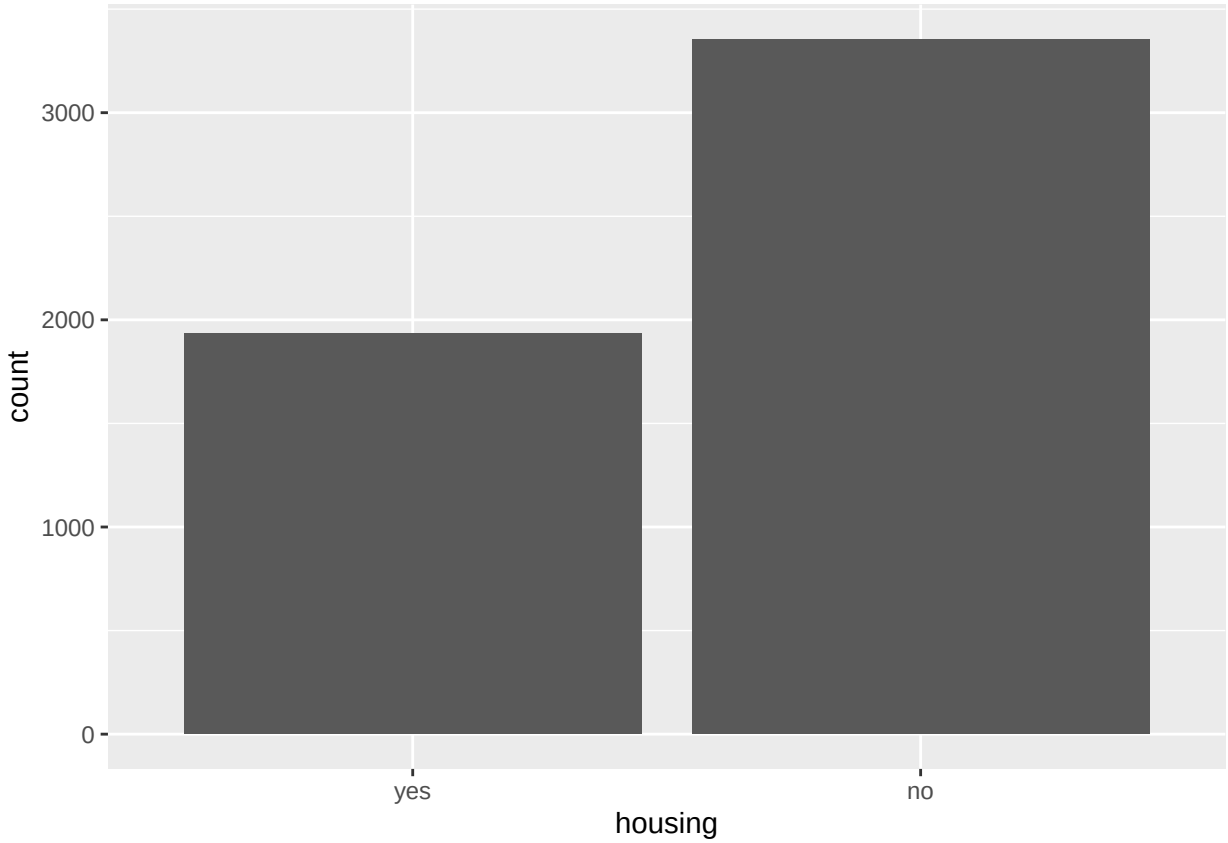
It is clear from the table that the bank makes the most calls during the summer months, and not very many calls in the fall and winter. This could be put down to strategy, because the fall and winter are riddled with money-spending holidays compared to the summer months, but it could also be worthwile making more calls in the fall and winter.

Education

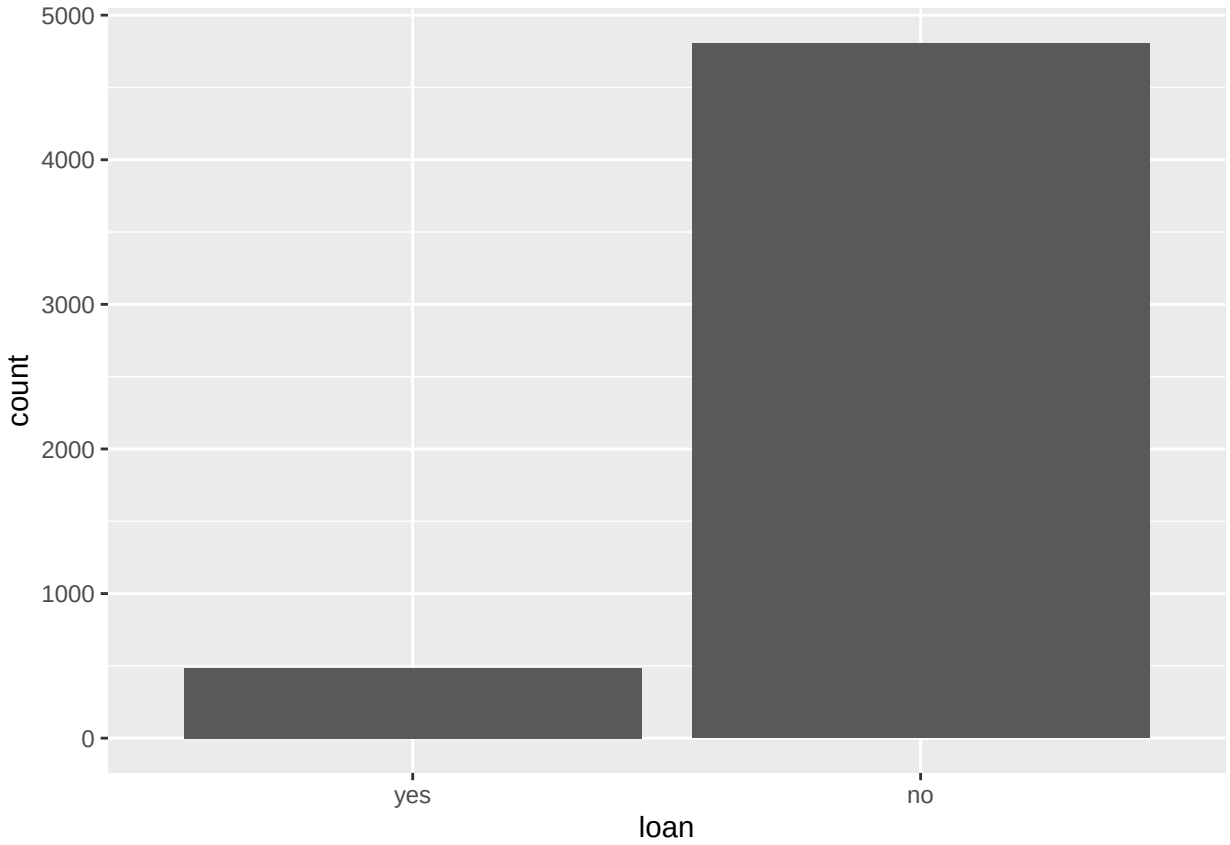
When looking at education, there is another clear pattern. When looking at percentages, those with an unknown education have a high percentage because there are very few people who have subscribed to term deposits with this being the case, suggesting that there are some perticular circumstances at play here. However, those with a tertiary or secondary level of education make the most term deposits and have higher percentages than those with a primary level of education, because they are more educated. The difference between elementary/middle school and high school and beyond in terms of real-world knowledge is massive, which is why it makes sense for more investments to be made by more educated people.

Here are two more relationships worth taking a look at:

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

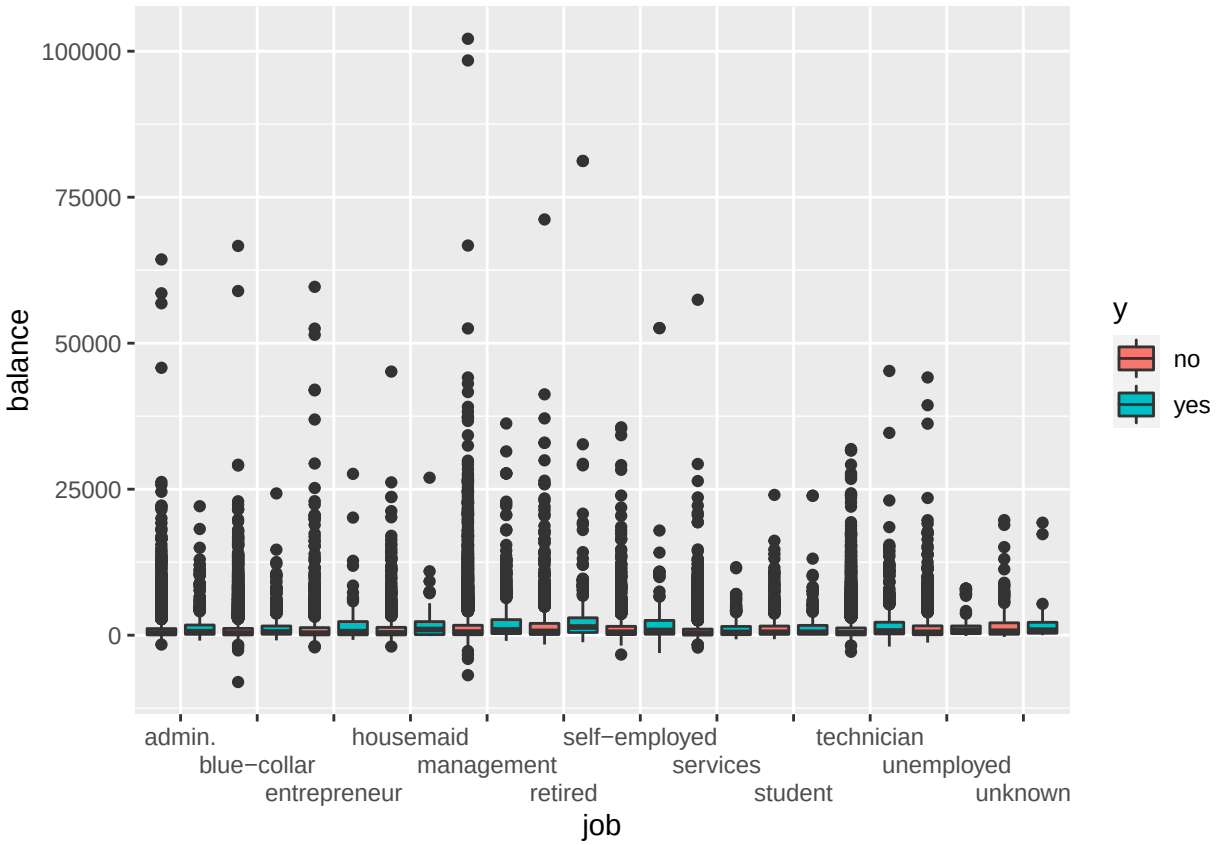


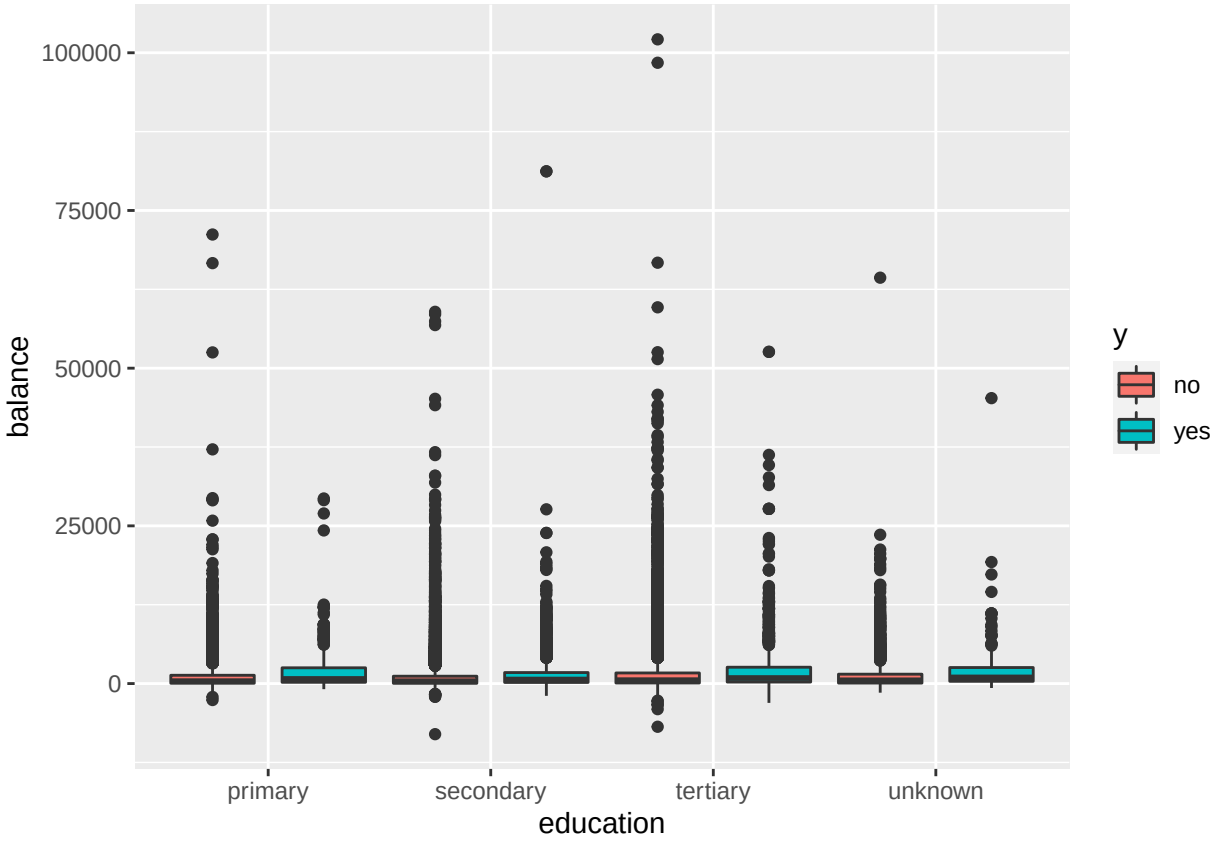

```
## `summarise()` ungrouping output (override with `.groups` argument)
```

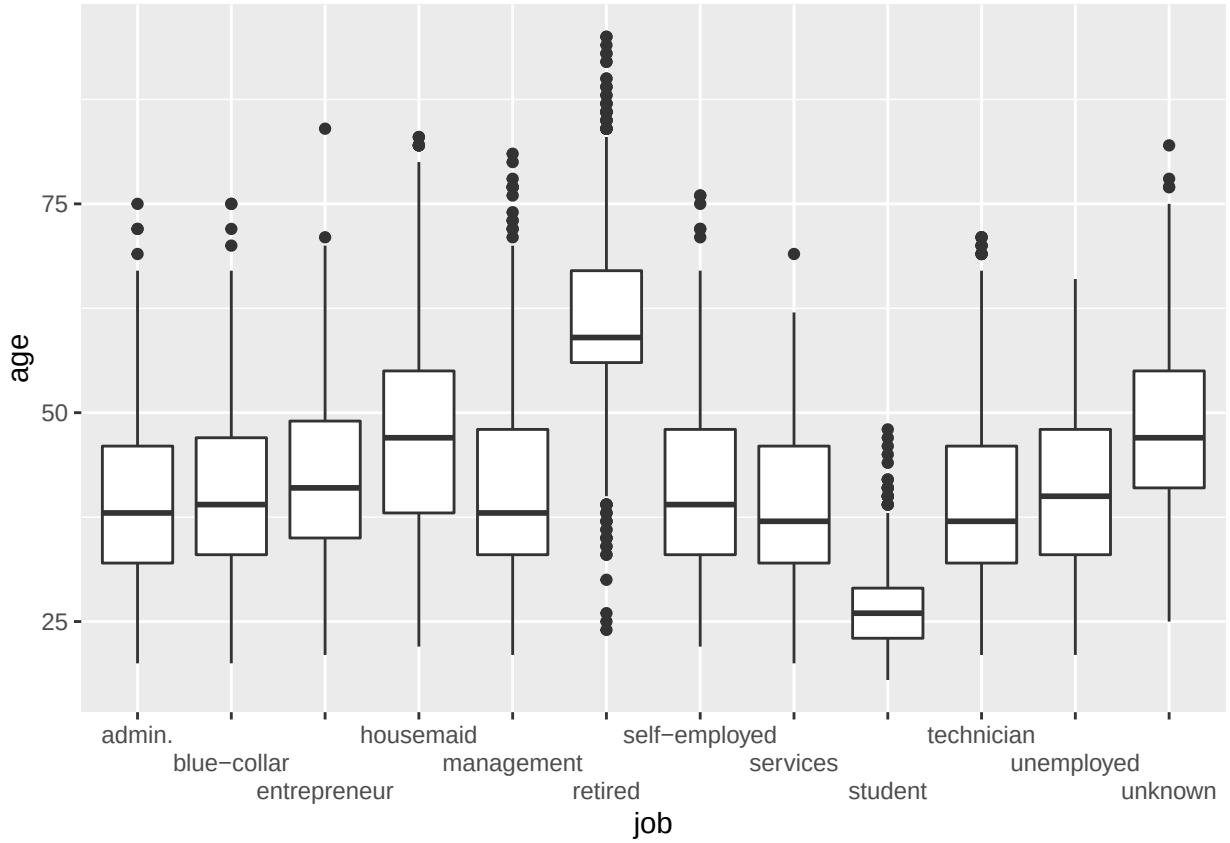


In both of these cases, there is a higher amount of people who have not subscribed to term deposits than those that have, which makes sense. Those that have housing loans are generally more focused on making those payments and need their money to put towards that each month, and those that have personal loans are the same, but in general have much less money, and therefore have a much lower count.

Before moving on to the final model, let's have one last look at the data, particularly when it comes to the balance column.


100000
75000
50000
25000
0
balance
admin.
blue−collar
entrepreneur
housemaid
management
retired
self−employed
services
student
technician
unemployed
unknown
job
y
no
yes


100000
75000
50000
25000
0
balance
primary
secondary
tertiary
unknown
education
y
no
yes



Boxplots

The two boxplots that deal with balance involve age and occupation. While these provide a whole host of information in their own right, they both match the conclusions drawn from the histogram data: Those in management have a higher count of loans because they are the richest group of people, and those with a tertiary education have a higher percentage and count because they make more money (as a result of their higher education). One interesting observation that comes from looking at the age distribution amongst jobs is that those in management aren't older, in fact they are on the younger end.

Creating a Machine Learning Algorithm

The dataset will be split into a 75% training set and a 25% testing set in order to avoid overfitting the algorithm, and then both the training and test sets will be duplicated in training and test sets with the outcome as a 0 or 1 instead of a "no" or "yes". The 75/25 split will be useful because it provides a good balance of training data and enough test data to yield an accurate result.

```
set.seed(2020, sample.kind="Rounding")
```

```
## Warning in set.seed(2020, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
y <- bank_full$y
test_index <- createDataPartition(y, times = 1, p = 0.25, list = FALSE)
test_set <- bank_full[test_index,]
```

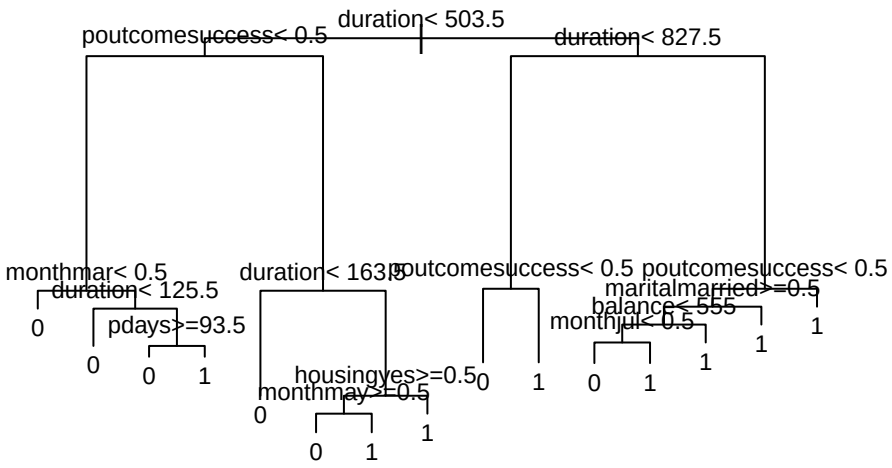
```
train_set <- bank_full[-test_index,]

train_set_numeric <- train_set %>% mutate(y = ifelse(y=="yes",1,0)) %>% mutate(y = factor(y))
test_set_numeric <- test_set %>% mutate(y = ifelse(y=="yes",1,0)) %>% mutate(y = factor(y))
```

Before removing the call duration from the training and test sets, running a decision tree algorithm (using caret's rpart) will show just how influential the duration is when it comes to predicting whether or not a client will make a term deposit:

```
train_rpart1 <- train(y ~ ., method = "rpart", tuneGrid = data.frame(cp = seq(0, 0.05, len = 25)), data=
confusionMatrix(predict(train_rpart1, test_set_numeric), test_set_numeric$y)$overall["Accuracy"]
```

```
##  Accuracy
## 0.8985315
```

```
plot(train_rpart1$finalModel, margin = 0.1)
text(train_rpart1$finalModel, cex = 0.75)
```



As we can see from the decision tree, many decisions rely on the duration. When adding the tuneGrid, the decision tree showed more branches, some of which don't involve the duration, so now the machine learning algorithms will be trained without the duration.

```
#Cleaning Data
train_set_numeric$duration <- NULL
```

```
test_set_numeric$duration <- NULL
train_set_numeric$poutcome <- NULL
test_set_numeric$poutcome <- NULL

# LDA
set.seed(1, sample.kind="Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
 train_lda <- train(y ~ ., method = "lda", data = train_set_numeric)

#QDA
 set.seed(1, sample.kind="Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
 train_qda <- train(y~., method = "qda", data=train_set_numeric)
#GLM
set.seed(1, sample.kind="Rounding")
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```
train_glm <- train(y~., method="glm", data=train_set_numeric)

#Rpart
set.seed(10, sample.kind="Rounding")
```

```
## Warning in set.seed(10, sample.kind = "Rounding"): non-uniform 'Rounding'
## sampler used
```

```
train_rpart2 <- train(y ~ ., method = "rpart",
       tuneGrid = data.frame(cp = seq(0, 0.05, len = 25)), data=train_set_numeric)

#Accuracy
#LDA Accuracy
confusionMatrix(predict(train_lda, test_set_numeric), test_set_numeric$y)$overall["Accuracy"]
```

```
##  Accuracy
## 0.8816348
```

```
#QDA Accuracy
confusionMatrix(predict(train_qda, test_set_numeric), test_set_numeric$y)$overall["Accuracy"]
```

```
##  Accuracy
## 0.8596072
```

```r
#GLM Accuracy
confusionMatrix(predict(train_glm, test_set_numeric), test_set_numeric$y)$overall["Accuracy"]
```

```
##  Accuracy
## 0.8847311
```

```r
#RPART Accuracy
confusionMatrix(predict(train_rpart2, test_set_numeric), test_set_numeric$y)$overall["Accuracy"]
```

```
## Accuracy
## 0.886058
```

With the rpart algorithm yielding the best accuracy, let's have a look at the confusionMatrix from that model:

```r
confusionMatrix(predict(train_rpart2, test_set_numeric), test_set_numeric$y)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 9884 1191
##          1   97  132
##
##                Accuracy : 0.8861
##                  95% CI : (0.8801, 0.8919)
##     No Information Rate : 0.883
##     P-Value [Acc > NIR] : 0.1564
##
##                   Kappa : 0.1404
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.99028
##             Specificity : 0.09977
##          Pos Pred Value : 0.89246
##          Neg Pred Value : 0.57642
##              Prevalence : 0.88296
##          Detection Rate : 0.87438
##    Detection Prevalence : 0.97974
##       Balanced Accuracy : 0.54503
##
##        'Positive' Class : 0
##
```

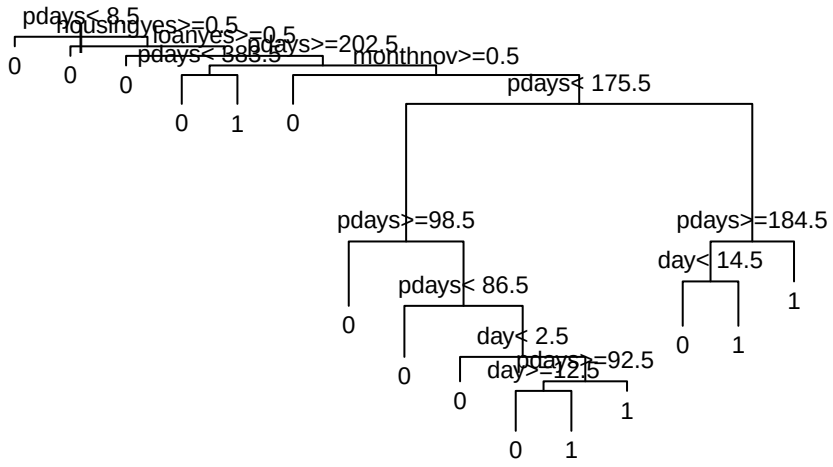
How about the variable importance?

```r
varImp(train_rpart2)
```

```
## rpart variable importance
##
```

```
##   only 20 most important variables shown (out of 39)
##
##                 Overall
## pdays          100.0000
## housingyes      77.9554
## age             54.9281
## previous        48.4007
## contactunknown  33.1192
## monthmay        14.2414
## jobblue-collar  11.7287
## monthnov        11.7127
## campaign         9.8780
## day              8.4708
## balance          7.3622
## loanyes          5.6527
## monthsep         4.1297
## monthjan         2.5447
## monthaug         2.0516
## monthjul         1.6320
## jobhousemaid     1.5189
## jobunemployed    0.8661
## jobstudent       0.4571
## monthmar         0.4278
```

```
plot(train_rpart2$finalModel, margin = 0.1)
text(train_rpart2$finalModel, cex = 0.75)
```



The importance of the pdays variable here is interesting, but doesn't necessarily yield a lot of information when it comes to determining which factors are most important when it comes to predicting when people will subscribe to a term deposit. However, when looking at all 20 of the variables it is clear that many of the important data points that we found in the beginning are repeated here. Whether or not someone has a house loan is one of the most important, which was something discussed earlier. Whether or not contact was made in the month of May was important, and that makes sense because there were many calls made in may. The same is true for the month of September, which has one of the highest percentages of people susbscribing to a term deposit.

Results, Conclusion, and Final Comments

Results

In the end, the rpart model yielded an accuracy of .8861, and the precision can be found as followed:

```
confusionMatrix(predict(train_rpart2, test_set_numeric), test_set_numeric$y)$byClass["Precision"]
```

```
## Precision
## 0.8924605
```

So we can see that the algorithm was successful at being able to determine the subscription of a client, even without the duration being taken into account.

Conclusion

The conclusion has not changed since the evaluation of the graphs generated from the data earlier. It is important to recognize that because there are a lot of people who do not subscribe to a term deposit, a lot of the predictors are ones that might predict a "no" instead of a yes. For example, looking at people who are maids or are unemployed, it is reasonable to conclude that they are very likely to not make a term deposit, compared to students who were found more likely to make one. The existence of a housing loan is also something that we discussed earlier, alongside age, marital status, and education. In order to have a higher success rate, the bank should look at people who are 18-24 or over 60 years old, with a higher level of education, who are either retired or a student, in the winter months, and who doesn't have any personal or house loans. Some characteristics to avoid are a blue-collar job, or being a maid or unemployed, making the calls in the month of may, and calling someone who has loans, all conclusions supported by the machine learning algorithm.

Final Comments

In exploring the data, creating the algorithm, and drawing conclusions I would say that I learned a lot about using r, looking at data, and finding patterns. In my eyes, this project was a success. In terms of final comments, I think that there were only a few obstacles that I wish I was able to overcome. First of all, determining the split between training and test sets, alongside tuning parameters in rpart were ones that I had worked with in the past, and didn't necessarily come from my knowledge of the data. Further, when creating the rpart algorithm, I chose to remove (alongside the duration variable) the poutcome variable, because it took away from some of the variables that are easier to control or determine. However, when trying to delve even deeper beyond the poutcome and pdays, I recieved errors when trying to get a decision tree from the algorithm. Nonetheless, I believe that the variables highlighted in varImp were consistent with the conclusions from earlier and therefore made this project successful.

Thanks

Many thanks to the HarvardX Data Science staff for their instruction and insight. Below is the citation from the dataset used for this project.

[Moro et al., 2014] S. Moro, P. Cortez and P. Rita. A Data-Driven Approach to Predict the Success of Bank Telemarketing. Decision Support Systems, Elsevier, 62:22-31, June 2014